UNIVERSITY OF THESSALY

SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Design Strategies Towards the Enhancement of Short-Term Forecasting in the Energy Sector

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy

Dimitrios Kontogiannis

June 2023

UNIVERSITY OF THESSALY

SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Design Strategies Towards the Enhancement of Short-Term Forecasting in the Energy Sector

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy

Dimitrios Kontogiannis

June 2023

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Στρατηγικές Σχεδιασμού για τη Βελτιστοποίηση Βραχυπρόθεσμων Προβλέψεων στον Τομέα Ενέργειας

Διατριβή η οποία υποβλήθηκε για τη μερική εκπλήρωση των υποχρεώσεων απόκτησης του Διδακτορικού Διπλώματος

Δημήτριος Κοντογιάννης

Ιούνιος 2023

UNIVERSITY OF THESSALY

SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Design Strategies Towards the Enhancement of Short-Term Forecasting in the Energy Sector

Ph.D. Dissertation

Dimitrios Kontogiannis

Advisory Committee

Dimitrios Bargiotas, Professor, University of Thessaly (Supervisor)

Aspassia Daskalopulu, Associate Professor, University of Thessaly

Lefteri H. Tsoukalas, Professor, University of Thessaly

Examination Committee

Dimitrios Bargiotas, Professor, University of Thessaly (Supervisor)

Lefteri H. Tsoukalas, Professor, University of Thessaly

Aspassia Daskalopulu, Associate Professor, University of Thessaly

Alexander Chroneos, Professor, University of Thessaly

Michael Vassilakopoulos, Professor, University of Thessaly

Dimitrios Katsaros, Associate Professor, University of Thessaly

Christos Manasis, Professor, National and Kapodistrian University of Athens

June 2023

DISCLAIMER ON ACADEMIC ETHICS AND INTELLECTUAL PROPERTY RIGHTS

Being fully aware of the implications of copyright laws, I expressly state that this Ph.D. dissertation, as well as the electronic files and source codes developed or modified in the course of this dissertation, are solely the product of my personal work and do not infringe any rights of intellectual property, personality and personal data of third parties, do not contain work / contributions of third parties for which the permission of the authors / beneficiaries is required and are not a product of partial or complete plagiarism, while the sources used are limited to the bibliographic references only and meet the rules of scientific citing. The points where I have used ideas, text, files and / or sources of other authors are clearly mentioned in the text with the appropriate citation and the relevant complete reference is included in the bibliographic references section. I also declare that the results of the work have not been used to obtain another degree. I fully, individually and personally undertake all legal and administrative consequences that may arise in the event that it is proven, in the course of time, that this dissertation or part of it does not belong to me because it is a product of plagiarism.

The Declarant

Dimitrios Kontogiannis

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Στρατηγικές Σχεδιασμού για τη Βελτιστοποίηση Βραχυπρόθεσμων Προβλέψεων στον Τομέα Ενέργειας

Διδακτορική Διατριβή

Δημήτριος Κοντογιάννης

Συμβουλευτική Επιτροπή

Δημήτριος Μπαργιώτας, Καθηγητής, Πανεπιστήμιο Θεσσαλίας (Επιβλέπων)

Ασπασία Δασκαλοπούλου, Αναπληρώτρια Καθηγήτρια, Πανεπιστήμιο Θεσσαλίας

Ελευθέριος Τσουκαλάς, Καθηγητής, Πανεπιστήμιο Θεσσαλίας

Επταμελής εξεταστική επιτροπή

Δημήτριος Μπαργιώτας, Καθηγητής, Πανεπιστήμιο Θεσσαλίας (Επιβλέπων)

Ελευθέριος Τσουκαλάς, Καθηγητής, Πανεπιστήμιο Θεσσαλίας

Ασπασία Δασκαλοπούλου, Αναπληρώτρια Καθηγήτρια, Πανεπιστήμιο Θεσσαλίας

Αλέξανδρος Χροναίος, Καθηγητής, Πανεπιστήμιο Θεσσαλίας

Μιχαήλ Βασιλακόπουλος, Καθηγητής, Πανεπιστήμιο Θεσσαλίας

Δημήτριος Κατσαρός, Αναπληρωτής Καθηγητής, Πανεπιστήμιο Θεσσαλίας

Χρήστος Μανασής, Καθηγητής, Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών

Ιούνιος 2023

ΥΠΕΥΘΥΝΗ ΔΗΛΩΣΗ ΠΕΡΙ ΑΚΑΔΗΜΑΪΚΗΣ ΔΕΟΝΤΟΛΟΓΙΑΣ ΚΑΙ ΠΝΕΥΜΑΤΙΚΩΝ ΔΙΚΑΙΩΜΑΤΩΝ

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ρητά ότι η παρούσα διδακτορική διατριβή, καθώς και τα ηλεκτρονικά αρχεία και πηγαίοι κώδικες που αναπτύχθηκαν ή τροποποιήθηκαν στα πλαίσια αυτής της διατριβής, αποτελούν αποκλειστικά προϊόν προσωπικής μου εργασίας, δεν προσβάλλουν οποιασδήποτε μορφής δικαιώματα διανοητικής ιδιοκτησίας, προσωπικότητας και προσωπικών δεδομένων τρίτων, δεν περιέχουν έργα/εισφορές τρίτων για τα οποία απαιτείται άδεια των δημιουργών/δικαιούχων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον και πληρούν τους κανόνες της επιστημονικής παράθεσης. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο, αρχεία ή/και πηγές άλλων συγγραφέων αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή. Δηλώνω επίσης ότι τα αποτελέσματα της εργασίας δεν έχουν χρησιμοποιηθεί για την απόκτηση άλλου πτυχίου. Αναλαμβάνω πλήρως, ατομικά και προσωπικά, όλες τις νομικές και διοικητικές συνέπειες που δύναται να προκύψουν στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής.

Ο Δηλών

Δημήτριος Κοντογιάννης

Ευχαριστίες

Θα ήθελα να ευχαριστήσω την οικογένειά μου για την αμέριστη υποστήριξη κατά τη διάρκεια των διδακτορικών σπουδών μου. Επίσης, θα ήθελα να ευχαριστήσω τους καθηγητές της τριμελούς συμβουλευτικής επιτροπής για την πολύτιμη καθοδήγηση και βοήθειά τους σε όλη τη διάρκεια των σπουδών μου αλλά και για την εξαιρετική συνεργασία στην ανάπτυξη ερευνητικού έργου που ανταποκρίνεται στα ερευνητικά μου ενδιαφέροντα.

Ph.D. Dissertation

Design Strategies Towards the Enhancement of Short-Term Forecasting in the Energy Sector

Kontogiannis Dimitrios

Abstract

Short-term forecasting processes constitute an integral part of data analysis in the energy sector since their integration in demand response programs, energy management systems, smart grid and energy market applications is valuable towards the examination of complex and volatile time series variables such as load and electricity price. Short-term load forecasting models offer valuable insight towards consumption patterns through the inspection of influential factors and introduce intelligent ways of monitoring electricity demand as well as the occurrence of irregular events in order to improve the decision-making processes of electric utilities and reinforce grid stability respectively. Short-term electricity price forecasting models address the challenge of price volatility and contribute towards the development of robust strategies towards efficient resource management and optimal energy transactions for all types of energy market participants and consumers. It is evident that methods focusing on load and electricity price time series follow a similar structure including preprocessing, forecasting and output modules for the estimation of the target variables after data collection. Therefore, this dissertation acknowledges the shared and overlapping structure of those forecasting processes and addresses prominent challenges and research gaps associated with each component through the development of optimal design strategies that improve the overall model performance. The study of the preprocessing module led to the assessment of robust feature selection and highlighted the role of rule generation for efficient examination of the studied environments. Since prominent challenges in data preprocessing are connected to dataset dimensionality and feature interpretability, a method towards the generation of a compact and interpretable set of rules through hybrid feature selection was proposed. Furthermore, the study of the main forecasting framework denoted challenges with regards to standalone, combinatorial

and meta-modeling design philosophies. In standalone modeling, the uncertainty surrounding estimator selection due to the insufficient exploration of edge cases often hinders research progress and leads to confusion with regards to training behavior. Consequently, a comparative study examining the baseline performance of neural network methodologies for high resolution predictions addresses one of the edge cases where the brevity of the training process and time constraints could provoke this uncertainty. Regarding combinatorial modeling, the uncertainty of estimation member selection in ensemble methods coupled with the challenges of concept and data drift could lead to arbitrary design decisions and suboptimal model combinations. As a result, a novel design strategy focusing on the deterministic selection of estimator members based on the structural characteristics of peak and non-peak indices was proposed in order to generate performant ensemble learning models. Moreover, the examination of meta-modeling approaches highlighted the performance benefits of additional forecasting layers and led to the introduction of a forecasting approach that estimated load consumption through the inspection of similarity and causality for the derivation of alternative time series representations. This approach improved the error metrics compared to the base LSTM ensemble model, denoting the impact of community factors when the quality of the input dataset is far from ideal. Lastly, following this a posteriori design method, the study of the output module identified the need for performance refinement through additional structures that estimate and minimize error values towards increased model stability and improved accuracy. In this scope, an error compensation module was developed towards the performance improvement of a deep learning structure for the task of short-term electricity price forecasting. This approach introduced an autoregressive model for the estimation of residual training error, resulting in more consistent predictions and overall lower error metrics when tested in different training scenarios. Additionally, this method discussed the potential addition of hyperparameters that configure the error compensation module for future applications and benchmarks. The extension of the strategies presented in this dissertation could enable the development of more flexible and adaptive forecasting pipelines that could enhance the capabilities of future energy applications.

Keywords:

Load forecasting, electricity price forecasting, demand response, machine learning, neural networks, artificial intelligence, regression, supervised learning, fuzzy logic, feature engineering, model selection, meta-modeling, short-term forecasting design

Διδακτορική Διατριβή

Στρατηγικές Σχεδιασμού για τη Βελτιστοποίηση Βραχυπρόθεσμων Προβλέψεων στον Τομέα Ενέργειας

Δημήτριος Κοντογιάννης

Περίληψη

Οι διεργασίες βραχυπρόθεσμης πρόβλεψης αποτελούν αναπόσπαστο κομμάτι της ανάλυσης δεδομένων στον τομέα της ενέργειας καθώς η ενσωμάτωσή τους σε προγράμματα ανταπόκρισης ζήτησης, συστήματα διαχείρισης ενέργειας, ευφυή δίκτυα ενέργειας και σε εφαρμογές ενεργειακών αγορών είναι πολύτιμη για την εξέταση πολύπλοκων και ευμετάβλητων χρονοσειρών όπως των μεταβλητών φορτίου και της τιμής ηλεκτρικής ενέργειας. Τα βραχυπρόθεσμα μοντέλα πρόβλεψης φορτίου προσφέρουν πολύτιμες πληροφορίες σχετικά με τα πρότυπα κατανάλωσης μέσω του ελέγχου παραγόντων επιρροής και εισάγουν ευφυείς τρόπους παρακολούθησης της ζήτησης ηλεκτρικής ενέργειας καθώς και της εμφάνισης ακανόνιστων γεγονότων, προκειμένου να βελτιώσουν τις διαδικασίες λήψης αποφάσεων των υπηρεσιών ηλεκτρισμού κοινής ωφέλειας και να ενισχύσουν τη σταθερότητα του δικτύου αντιστοίχως. Τα βραχυπρόθεσμα μοντέλα πρόβλεψης τιμών ηλεκτρικής ενέργειας αντιμετωπίζουν την πρόκληση της αστάθειας των τιμών και συμβάλλουν στην ανάπτυξη ισχυρών στρατηγικών για την αποτελεσματική διαχείριση των ενεργειακών πόρων και τη βελτιστοποίηση ενεργειακών συναλλαγών για όλα τα μέλη των ενεργειακών αγορών και για τους καταναλωτές. Είναι προφανές ότι οι μέθοδοι που εστιάζουν στις χρονοσειρές φορτίου και τιμών ηλεκτρικής ενέργειας ακολουθούν παρόμοια δομή, συμπεριλαμβάνοντας στοιχεία προεπεξεργασίας, πρόβλεψης και εξόδου για την εκτίμηση των μεταβλητών-στόχων μετά τη συλλογή δεδομένων. Επομένως, αυτή η διατριβή αναγνωρίζει την κοινή και επικαλυπτόμενη δομή αυτών των διεργασιών πρόβλεψης και απευθύνεται στην αντιμετώπιση εμφανών προκλήσεων και ερευνητικών κενών που σχετίζονται με κάθε δομικό στοιχείο μέσω της ανάπτυξης βέλτιστων στρατηγικών σχεδιασμού που βελτιώνουν τη συνολική απόδοση των μοντέλων. Η μελέτη του δομικού

στοιχείου προεπεξεργασίας οδήγησε στην αξιολόγηση της ισχυρής επιλογής χαρακτηριστικών και ανέδειξε το ρόλο της παραγωγής κανόνων για την αποτελεσματική εξέταση των υπό μελέτη περιβαλλόντων. Καθώς εξέχουσες προκλήσεις στην προεπεξεργασία δεδομένων συνδέονται με τις διαστάσεις των συνόλων δεδομένων και την ερμηνευσιμότητα των χαρακτηριστικών, προτάθηκε μία μέθοδος για τη δημιουργία ενός συμπαγούς και ερμηνεύσιμου συνόλου κανόνων μέσω υβριδικής επιλογής χαρακτηριστικών. Επιπροσθέτως, η μελέτη του δομικού στοιχείου που αποτελεί το κύριο πλαίσιο πρόβλεψης εμφάνισε προκλήσεις που σχετίζονται με τις αυτόνομες, συνδυαστικές και μετα-μοντελοποιητικές φιλοσοφίες σχεδιασμού. Στην αυτόνομη μοντελοποίηση, η αβεβαιότητα που περιβάλλει την επιλογή εκτιμητών λόγω της ανεπαρκούς εξερεύνησης ακραίων περιπτώσεων συχνά παρεμποδίζει την πρόοδο της έρευνας και οδηγεί σε σύγχυση που σχετίζεται με τη συμπεριφορά εκπαίδευσης. Συνεπώς, μία συγκριτική μελέτη που εξετάζει την απόδοση αναφοράς των μεθοδολογιών νευρωνικών δικτύων για προβλέψεις υψηλής ανάλυσης αναφέρεται σε μία από τις ακραίες περιπτώσεις όπου η συνοπτικότητα της διαδικασίας εκπαίδευσης και οι χρονικοί περιορισμοί θα μπορούσαν να προκαλέσουν αυτή την αβεβαιότητα. Στη συνδυαστική μοντελοποίηση, η αβεβαιότητα της επιλογής των εκτιμητών-μελών για μεθόδους συνόλου σε συνδυασμό με τις προκλήσεις αποκλίνουσας αντίληψης μοντέλου και απόκλισης δεδομένων θα μπορούσαν να οδηγήσουν σε αυθαίρετες σχεδιαστικές αποφάσεις και ανεπαρκείς συνδυασμούς μοντέλων. Κατά συνέπεια, προτάθηκε μία νέα στρατηγική σχεδιασμού που επικεντρώνεται στην ντετερμινιστική επιλογή των μελών του εκτιμητή με βάση τα δομικά χαρακτηριστικά των δεικτών κορύφωσης και μη-κορύφωσης ώστε να δημιουργηθούν αποδοτικά μοντέλα μάθησης συνόλου. Επίσης, η εξέταση των προσεγγίσεων μετα-μοντελοποίησης ανέδειξε τα οφέλη απόδοσης που προκύπτουν από τη χρήση περισσότερων επιπέδων πρόβλεψης και οδήγησε στην εισαγωγή μίας προσέγγισης πρόβλεψης που υπολόγιζε την κατανάλωση φορτίου μέσω της επισκόπησης της ομοιότητας και της αιτιότητας για τη δημιουργία εναλλακτικών αναπαραστάσεων χρονοσειρών. Αυτή η προσέγγιση βελτίωσε τις μετρήσεις σφάλματος σε σύγκριση με το βασικό μοντέλο συνόλου LSTM, υποδηλώνοντας την επίδραση των παραγόντων κοινότητας όταν η ποιότητα του συνόλου δεδομένων εισόδου απέχει αρκετά από την ιδανική. Τέλος, ακολουθώντας αυτή τη μέθοδο της εκ των υστέρων σχεδίασης, κατά τη μελέτη του δομικού στοιχείου εξόδου εντοπίστηκε η ανάγκη για βελτίωση απόδοσης μέσω

πρόσθετων δομών που εκτιμούν και ελαχιστοποιούν τις τιμές σφάλματος για την αυξημένη σταθερότητα και βελτιωμένη ακρίβεια του μοντέλου. Σε αυτό το πεδίο, αναπτύχθηκε ένα δομικό στοιχείο αντιστάθμισης σφαλμάτων για τη βελτίωση της απόδοσης μίας δομής βαθιάς μάθησης για τη βραχυπρόθεσμη πρόβλεψη τιμών ηλεκτρικής ενέργειας. Αυτή η προσέγγιση εισήγαγε ένα αυτοπαλινδρομικό μοντέλο για την εκτίμηση του υπολειπόμενου σφάλματος εκπαίδευσης, οδηγώντας σε πιο συνεπείς προβλέψεις και σε συνολικά χαμηλότερες μετρήσεις σφάλματος μετά από δοκιμές σε διαφορετικά σενάρια εκπαίδευσης. Επιπροσθέτως, αυτή η μέθοδος εξέτασε την πιθανή προσθήκη υπερπαραμέτρων για τη διαμόρφωση του στοιχείου αντιστάθμισης σφαλμάτων σε μελλοντικές εφαρμογές και μοντέλα αναφοράς. Η επέκταση των στρατηγικών που παρουσιάζονται σε αυτή τη διατριβή θα μπορούσε να επιτρέψει την ανάπτυξη πιο ευέλικτων και ευπροσάρμοστων διεργασιών πρόβλεψης που θα ενίσχυαν τις δυνατότητες μελλοντικών ενεργειακών εφαρμογών.

Λέξεις-κλειδιά:

Πρόβλεψη φορτίου, πρόβλεψη τιμής ηλεκτρικής ενέργειας, ανταπόκριση ζήτησης, μηχανική μάθηση, νευρωνικά δίκτυα, τεχνητή νοημοσύνη, παλινδρόμηση, εποπτευόμενη μάθηση, ασαφής λογική, μηχανική χαρακτηριστικών, επιλογή μοντέλου, μετα-μοντελοποίηση, σχεδίαση βραχυπρόθεσμης πρόβλεψης

Contents

List of Figures

List of Tables

List of Abbreviations

ACF	Autocorrelation Function Threshold
ADWIN	Adaptive Windowing
AIC	Akaike Information Criterion
AR	Autoregression
ARIMA	Autoregressive Integrated Moving Average
ARMA	Autoregressive Moving Average
BIC	Bayesian Information Criterion
CNN	Convolutional Neural Network
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DNN	Deep Neural Network
DR	Demand Response
DTW	Dynamic Time Warping
ERC-DNN	Error Compensation-Deep Neural Network
GBRT	Gradient Boosting Regression Trees
GRU	Gated Recurrent Unit
HQIC	Hannan-Quinn Information Criterion
HWES	Holt Winter’s Exponential Smoothing
ISO	International Standards Organization
KNN	K-Nearest Neighbors
LARS	Least-angle Regression
LSTM	Long Short-Term Memory
MA	Moving Average
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MLP	Multi-Layer Perceptron
MSE	Mean Squared Error
NEPOOL	New England Power Pool
OLS	Ordinary Least Squares
PCA	Principal Component Analysis
PSI	Population Stability Index
ReLU	Rectified Linear Unit
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
SEATS	Seasonal Extraction in ARIMA Time Series
SES	Simple Exponential Smoothing
SGD	Stochastic Gradient Descent
STL	Seasonal and Trend decomposition using Loess
SVD	Singular Value Decomposition
SVM	Support Vector Machine
SVR	Support Vector Regression
TFT	Temporal Fusion Transformer
t-SNE	t-Stochastic Neighbor Embedding

UMAP	Uniform Manifold Approximation and Projection
VAR	Vector Autoregression
VARMA	Vector Autoregressive Moving Average
XGBoost	Extreme Gradient Boosting

Chapter 1 Introduction

1.1 Motivation

The evolution of power systems coupled with the growth and the increased complexity of electricity markets introduce a plethora of challenges as well as interesting research questions often connected to the study and processing of time series data such as load and electricity price. Modern power grid design focuses on the development of robust data-driven strategies that could control the bidirectional flow of information between electricity providers and consumers since the penetration of renewable energy sources and the increase in energy demand could lead to unstable operation, poor resource management and inefficient scheduling, resulting in imbalanced demand response and consumer dissatisfaction [1]. Additionally, modern electricity markets often adopt sophisticated data-driven methods for the design of smart energy policies due to the phenomenon of price volatility in order to perform efficient electricity trading [2]. Short-term forecasting tasks involving load and electricity price time series add immense value to those data-driven approaches since the ability to predict these values accurately over a prediction horizon of several minutes or hours provides the necessary knowledge for optimal decision-making. Accurate short-term load forecasting contributes towards the effective planning and reliable operation of modern power grids since irregular events could be avoided and demand response flexibility could be improved. On a consumer level, short-term load forecasting could indirectly influence the rescheduling of daily tasks through intelligent analytics for the optimization of electricity consumption and the optimal response to financial incentives. Moreover, load forecasting enables the development of cost-effective consumption strategies that could assist in flattening the demand curve [3]. Accurate short-term price forecasting contributes to the efficiency of energy transactions due to the minimization of uncertainty, giving market participants the opportunity to react to changes in price appropriately and follow price trends [4]. Both categories of energy forecasting tasks are valuable to the development of real-time applications and energy management systems.

Short-term forecasting tasks for the prediction of load or electricity price typically follow a regression analysis in order to interpret the relationships between the dependent time series which expresses the target variable and several independent influencing factors such as temperature, fuel cost or historical load and price values derived from previous timesteps. Recent research on the field shows that prominent methods for those regression tasks stem from a statistical and artificial intelligence background. Statistical methods follow a more traditional path towards the discovery of linear and nonlinear relationships between the data based on assumptions that lead to the construction of a mathematical model which best fits the dataset. These forecasting approaches often follow a simple and easily interpretable structure, requiring less computing power for data processing. However, the resulting models are often limited due to those initial assumptions about the dataset that could impact the discoverability of patterns and trends negatively. Furthermore, this simplicity of structure could hinder the predictive potency of statistical methods since the resulting mathematical models may not be capable of explaining all data dependencies equally well as the dataset becomes larger and the relationships between features become increasingly complex. Methods such as linear regression and autoregressive moving average are commonly utilized in those time series forecasting tasks in order to predict future values of load and price through the interpretation of trends and the examination of influencing factors [5].

On the other side of the spectrum, artificial intelligence methods approach function approximation in a more flexible way, through the development of free-form models that adapt to the input and iteratively learn the relationships between the variables. This category of models often has a more complex structure with computations becoming increasingly difficult to follow and interpret as the scale and the complexity of the forecasting problems increase, essentially rendering them as black-box approaches. A major set of artificial intelligence methods in this research space consists of machine learning models. Machine learning approaches featuring prominent supervised learning algorithms such as random forest and gradient boosted decision trees offer scalable and performant solutions to regression tasks in the energy sector. Moreover, neural network models such as multi-layer perceptron (MLP) and long short-term memory network (LSTM) greatly contribute towards the development of dynamic and adaptive forecasting structures that process time dependencies efficiently and are capable of identifying all possible interactions between independent variables as they follow an implicit detection process for complex nonlinear relationships. The main disadvantages of machine learning methods that could be encountered more frequently in neural network models are the lack of interpretability, the increased computational burden and the proneness to overfitting. These disadvantages could result in suboptimal prediction accuracy and generalization issues as these models are integrated in real world applications. Complementary to machine learning methods, fuzzy logic approaches aim to reinforce the interpretability of forecasting models by introducing a set of rules that expresses the relationships between features, rendering feature selection a manageable task for most artificial intelligence algorithms [6].

Lastly, it is worth mentioning that while the categorization of the most prominent forecasting methods highlights the contributions of linear and nonlinear statistical models as well as machine learning algorithms and fuzzy systems, these approaches are not necessarily utilized as standalone estimators for every load or price forecasting task. Therefore, we have to acknowledge the broad set of design philosophies that lead to hybrid modeling [7]. Hybrid modeling focuses on the combination of multiple estimators from the previously discussed categories in order to develop robust structures that process the input simultaneously or sequentially. As an example, fuzzy neural networks merge elements from fuzzy inference systems and neural network design in order to utilize those principles cooperatively or as a fully fused entity in time series forecasting. In addition to those combinatorial approaches, hybrid modeling includes the plethora of ad hoc methods that focus on the transformation or decomposition of the output for the purposes of a new model which may function as an additional processing layer. This subcategory of hybrid forecasting models form the set of meta-modeling approaches and offer significant value to short-term forecasting research as they represent the extra step in combinatorial design that could further improve prediction accuracy [8]. Figure 1.1 shows this categorization in short-term load and price forecasting, denoting the most prominent types of methods in this research space that substantially influenced the content of this dissertation.

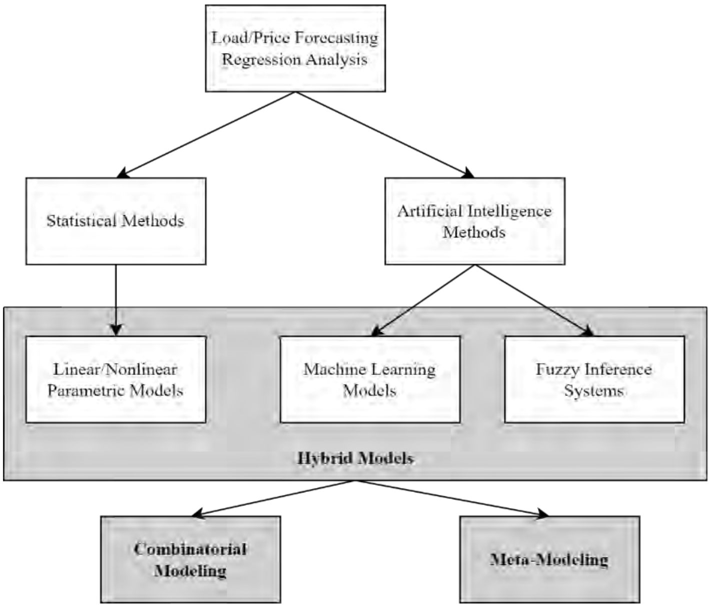


Figure 1.1: Categorization of prominent short-term load and price forecasting methods and modeling methodologies.

1.2 Overview of Challenges and Research Gaps

This section outlines several prominent challenges and research gaps in the design and implementation of short-term forecasting models in the energy sector, denoting the core research directions followed in this dissertation. First, it is evident that since load and price are influenced by a plethora of factors, the number of relevant features included in datasets is large, forming a high-dimensional space where the sparsity and dissimilarity of some feature groups could hinder the accurate generalization of the model and increase the overall complexity of the forecasting structure. Consequently, the emergence of the dimensionality challenge could contribute towards the emergence of interpretability issues since a large amount of training data would be required for efficient learning and the relationships between the studied variables could become too difficult to follow. The challenges of dimensionality and interpretability show a degree of codependence and could affect the performance of statistical as well as artificial intelligence methods [9]. In statistical methods, interpretability issues are often strongly connected to dimensionality since the initial set of assumptions for the dataset and the mathematical models utilized are relatively simple. In artificial intelligence methods, the challenge of interpretability could affect models utilizing low-dimensional feature spaces independently, when the forecasting method follows a black-box approach for parameter tuning as it is commonly observed in neural network structures. It is evident that fuzzy inference systems could reinforce the interpretability of a model since the decomposition of variables in linguistic terms and the extraction of an accurate set of rules could clearly explain the complete relationships between features. However, the algorithmic discovery of minimal rule sets that maintain high levels of accuracy remains an open research question since most rule bases on high-dimensional spaces either consider all possible rules, resulting in inefficient inference systems that ignore the challenge of dimensionality, or utilize expert knowledge which inherently does not define a deterministic and easily interpretable process.

Second, some research gaps related to the participation of estimators in forecasting frameworks could be identified. In standalone modeling, research works often present state of the art models as parts of a novel forecasting pipeline but there are not enough research projects aimed at a comprehensive performance overview of a specific state of the art approach in fundamental supervised learning tasks that utilize load or price time series. Additionally, studies do not sufficiently cover forecasting tasks that could be considered as edge cases in terms of training and convergence time. Consequently, uncertainty often surrounds the selection of a specific model configuration when the most prominent models such as the long short-term memory network have several performant structures that could be applied on the same forecasting tasks. Therefore, without the guidance of research works exploring this space, extensive and repeated testing could delay the development of useful forecasting approaches. In combinatorial modeling and especially in the development of ensemble forecasting methods, uncertainty surrounds the selection of the estimator members as these are often included arbitrarily due to their relevance or due to their prominence in recent research work. As a result, load and price forecasting models that utilize diverse feature sets which follow different distributions often fail to adapt to the input. The research space of deterministic strategies that generate optimal estimator sets is not sufficiently explored and there are still several steps that need to be taken towards the development of more modular, adaptive and generative processes. Moreover, meta-modeling in short-term forecasting is an active and evolving research topic since the intricacies of the data collection process reinforce the need for models that generate and process different interpretations of the target variables in order to efficiently capture patterns in non-ideal data structures.

Third, the value of prediction refinement in combinatorial modeling should be acknowledged since recent reviews and benchmarks do not often include post-processing techniques in the forecasting pipeline. It is important to note that feedback systems capable of processing residual error values could introduce several useful hyperparameters for model selection as well as improved prediction accuracy. The inclusion of those systems in enhanced and modular forecasting pipelines could reinforce prediction stability within the specified prediction horizon, providing an additional tool against noisy forecasts. Furthermore, the study or error compensation systems could address the inconsistencies in the performance evaluation of estimators as error fluctuations in hourly or minutely predictions could be reduced.

1.3 Dissertation Outline and Contributions

This dissertation aims to address the previously discussed challenges and research gaps in the design and development of short-term load and price forecasting methodologies through the analytical presentation of research projects that contribute towards the enhancement of widely used forecasting processes in the energy sector. Therefore, this dissertation is structured as follows:

In Chapter 2, a thorough examination of the short-term forecasting structure for time series in the energy sector is presented and the main modules and processes utilized in most recent research efforts conducting regression analysis for the prediction of load and electricity price are discussed. The individual inspection of the processes that form forecasting models in this research space enables the analysis of the challenges that could potentially hinder the performance of each module and denote specific areas where our research contributions could be applicable. This chapter highlights the roles of the data collection module, the preprocessing module, the forecasting framework and the output module, forming the baseline forecasting structure. Through the study of each process, suitable approaches from the literature are highlighted for several prominent short-term forecasting scenarios where the intricacies of each task are detailed. Furthermore, the most prominent prediction evaluation techniques are outlined through the definition of widely used error metrics. Lastly, an enhanced version of the forecasting process pipeline is presented, suggesting proposed adjustments that could benefit the initial structure and improve the forecasting performance of several forecasting tasks when one or more of those adjustments are applied. This enhanced version of the forecasting pipeline is directly connected to the research works presented in the following chapters.

In Chapter 3, the enhancement of the preprocessing module is studied through the design and implementation of a fuzzy system that features a hybrid feature selector towards the reinforcement of interpretability and the reduction of the initial fuzzy dataset. This system follows the forward chaining Mamdani approach and utilizes an improved decision tree linearization for the generation of a small and accurate rule base. Additionally, the hybrid feature selector combines metrics from extreme gradient boosting and decision tree structures in order to derive a concise set of important features. Since several machine learning methods utilize neural networks and neuro-fuzzy systems for consumer load predictions and recommendations in energy management systems, the efficient processing of additional fuzzy parameters such as weather data is an important step towards the complete and accurate discovery of relationships between independent and dependent variables. This discovery boosts the overall transparency of the model as the fuzzy rules that connect the remaining features could clearly explain the values of neural network parameters without the need to retrace every step of the computation process. Additionally, an algorithmic approach for rule base generation often speeds up data processing methods in this scope as expert knowledge and brute-force approaches could no longer be viable for high-dimensional fuzzy datasets. Furthermore, robust feature selection strategies are valuable tools towards dimensionality reduction, finding wide application in most short-term load and price forecasting tasks. In this study, the fuzzification of weather parameters enables the usage of the hybrid feature selector for the discovery of the most impactful feature states that are strongly connected to consumer load values. As a result, the performance evaluation of this fuzzy system showed that a drastically smaller and slightly more accurate rule base could be generated at a lower time frame when compared to the baseline decision tree linearization due to the integration of the hybrid feature selector.

In Chapters 4, 5 and 6, an in-depth study of the forecasting layer addresses challenges and research gaps in standalone, combinatorial and meta-modeling approaches through the analysis of several research works. First, in Chapter 4, a study presenting a comprehensive performance comparison between multi-layer perceptron, convolutional neural network and several long short-term network structures on the task of minutely active power forecasting aims to present the behavior of widely used neural network models on a fundamental short-term forecasting approach. This research work attempts to guide and motivate research on similar tasks in the energy sector through the presentation of prominent methods and the examination of error metrics as well as training time. This project shows that while the forecasting performance of neural network structures on a baseline configuration does not exhibit drastic differences in terms of error metrics, the training time and the complexity of each architecture play an important role in model selection since load and price predictions in the short-term horizon need to be derived within strict intervals of minutes or hours as the models get recalibrated in order to include newly recorded samples. Therefore, this study denotes that several important decisions that need to be taken in standalone modeling when model complexity, prediction horizon and computing resource availability are considered.

Second, in Chapter 5, a research project presenting a novel estimator selection strategy for ensemble learning models addresses the overall uncertain and often arbitrary inclusion of base estimators in combinatorial modeling. Since accurate short-term electricity demand forecasting is vital to the evolution of smart grids and the development of robust demand side management strategies, the selection of estimators that are most compatible to the given input is an important task. Moreover, it is evident that as the scale of the forecasting problem increases and time series from a diverse set of consumers are utilized, the need to transition from static and centralized standalone predictors to more adaptive and generative approaches that could manage the intricacies of those diverse data distributions becomes accrescent. Therefore, this research project is motivated by the cluster-based aggregate framework and introduces a flexible structural ensemble approach where the base estimators are selected through the cross-examination of error metrics from the evaluation of peak and non-peak indices. The use case presented in this study shows the intended behavior of this strategy since this implementation enables the generation of ensemble models that achieve the expected performance boost in a deterministic way.

Third, in Chapter 6, the impact of meta-modeling techniques towards the reduction of error metrics is explored through a research project that introduces a short-term forecasting model utilizing the combined effects of similarity and causality for the robust estimation of load. Data abnormalities stemming from a non-ideal data collection process could hinder the accuracy of estimators and the inherent diversity of client time series could complicate the interpretation of relationships throughout the training process. Therefore, this novel approach shows that the utilization of different input datasets for the estimation of additional load time series components through long short-term memory ensembles could derive similar and causal data representations. These components are passed to a multi-layer perceptron which functions as a meta-processing estimation layer that derives the target output. Our experiments indicated that the inclusion of this meta-modeling structure in the forecasting pipeline and the combined processing of similarity and causality features resulted in more performant models when compared to neural network ensembles utilizing only one output data representation.

In Chapter 7, a research work presenting a novel a posteriori processing methodology for short-term electricity price forecasting based on residual error estimation addresses the research gaps derived from the underutilization of error compensation systems in combinatorial modeling and the lack of related hyperparameters in recent reviews and benchmarks. The improvement of the output time series is the decisive final step towards robust estimation in the energy sector since it enables the derivation of more stable error profiles and the emergence of useful evaluation parameters that could be used in optimization processes. The proposed methodology utilizes a benchmark deep neural network structure for the prediction of day-ahead electricity prices and enhances the output module with the development of an autoregressive process tuned by several information criteria for the reduction of the error component in the final price prediction. Our experiments indicated that this approach yields improved error metrics when compared to the baseline deep learning structure in several training scenarios and the refined predictions shared increased stability throughout the forecasting horizon.

In Chapter 8, a comprehensive summary of the contributions is presented with additional comments based on the results of our experiments that highlight the advantages and disadvantages of the proposed methodologies. The integration of those methods in future energy applications and benchmarks is discussed and the overall enhancement of the forecasting pipeline with the inclusion of one or more of those methods is addressed.

Furthermore, the motivation for future research work is included in this chapter and future directions towards the expansion and the combination of the proposed methods are analyzed with examples and use cases relevant to the research areas of short-term forecasting and demand response in the energy sector.

Chapter 2 Short-term Time Series Forecasting Structure in the Energy Sector

Short-term forecasting tasks in the energy sector focus on the processing of time series data for the derivation of estimated values for the prediction of load and electricity price. This forecasting horizon covers predictions for up to one week ahead with more prominent tasks targeting minutely, hourly and daily predictions. The predicted values for load and price could reflect the expected value in that timeframe or the probability that summarizes future events, expressed as a set of different outcomes. The estimation of the expected values of load and price is commonly derived from tasks involving the approximation of a mapping function which aims to interpret the relationships between input and output variables. This forecasting approach is known as regression predictive modeling [10]. On the other side of the spectrum, models predicting the occurrence of specific outcomes and categorizing time series in groups are typically designed as classification and clustering tasks respectively. Classification tasks take advantage of labeled time series features in order to assign a label to a new and unlabeled time series based on common patterns. These tasks could contribute towards the efficient association of consumer time series to specific categories, formed by load and price policies, through the examination of historical data and customer characteristics [11]. Clustering time series tasks focus on the separation of unlabeled time series and the discovery of distinct groups based on patterns, distance and similarity metrics. Load and price forecasting models often utilize clustering in order to discover groups of similar consumers. Additionally, time series clustering approaches are utilized for anomaly detection in power grids as well as energy markets [12]. In this dissertation, we focus on the study and interpretation of relationships between the core energy time series variables of load and price and the independent influencing factors that affect them for the accurate prediction of target values. Therefore, we select regression predictive modeling as the base design philosophy for the presentation and examination of the time series forecasting structure. Classification and clustering methodologies support this structure indirectly as optional processing tasks or supplementary forecasting tasks when the time series data represent the consumption of a diverse consumer base.

2.1 Pipeline Overview

A high-level examination of the processes that contribute towards the development of predictive regression models for load and electricity price time series leads to the distinction of several modules that form the path from the construction of the input to the prediction of the output. The data collection module utilizes a set of methodologies for the collection of raw data and organization of values into time series features. The output of this module is the initial dataset made available for regression research and development. This initial dataset consists of data points indexed in time order and includes the target variables as well as a plethora of influencing factors. The values of those factors could either be connected to the target time series based on specific timestamps or they could independently characterize the entire time series. The features obtained from the data collection module typically include most time series characteristics such as trends, seasonal and nonseasonal cycles, pulses, steps and outliers [13]. Additionally, depending on the quality of the data collection process, the initial dataset could have missing values, noisy data and features that may not be strongly connected to the variables of load or price [14].

Some time series characteristics such as trends and seasonal patterns are valuable for the development of robust estimators as their detection is an integral part of most models. Sudden temporary or permanent shifts in the series level resulting in pulses and steps respectively could lead to uncertainty and poor model fitting when the underlying events are not properly explained. Furthermore, it is evident that the existence of missing values as well as noisy and insignificant features could increase forecasting error, resulting in poor load and price estimation. It is also worth mentioning that the initial dataset may not meet several compatibility criteria that satisfy the fundamental assumptions of a model such as data distribution, sample size and dataset dimensions, resulting in poor training performance. Additionally, the scope of application for short-term forecasting tasks in the energy sector is closely connected to the studied data structure. For example, models that predict load values from a diverse set of consumers follow a different data structure when compared to individual consumption predictions. Models aimed at larger groups of consumers typically include load features for each consumer, increasing the dimensions of the dataset and often requiring the application of clustering and classification methods for optimal feature division. Therefore, the initial dataset is passed to the preprocessing module where a set of data transformations, decompositions and feature engineering techniques could be applied in order to finalize the dataset structure and provide a compatible data representation for each forecasting task. The output of the preprocessing module contains the input and output features of the model. Time series splitting strategies are utilized before any data transformation and feature engineering task is executed in order to split the input and output features into training, validation and test sets given a specified ratio. The training set typically contains the larger percentage of samples and is used for the discovery of patterns and relationships between features during the learning process. The validation set includes a smaller percentage of samples and is often utilized for parameter tuning during training and for the prevention of overfitting. Lastly, the test set often contains a percentage of samples comparable to the validation set and is used to evaluate the performance of the forecasting models on unknown data after training. This performance evaluation aims to derive unbiased metrics that denote the generalization capabilities of the model as well as the magnitude of error. Data processing methodologies are typically applied based on the data available in the training samples in order to ensure that information from the sets used for model evaluation do not influence feature engineering decisions, hence eliminating the bias of involving samples that are supposed to remain unknown [15].

Following the data preprocessing, the resulting training and validation sets for input and output features are passed to the forecasting framework. The forecasting framework includes the estimation models as well as the supporting heuristics and algorithms that could reinforce model selection, hyperparameter tuning and model fitting. Several forecasting frameworks in short-term load and electricity price forecasting utilize a single estimator due to recalibration and computation power constraints. It is evident that newly proposed models in this research space are rarely compared in terms of their computational requirements and their deployment in real world applications could be uncertain due to the tradeoff between computation time and cost [16]. It is worth considering that complex standalone estimator structures and robust combinatorial approaches may offer marginally better forecasts but the overall benefits from this application may be lower than the execution cost of the model on more powerful systems when appropriate computational power is available. Additionally, recalibration constraints may have an impact on model selection in this forecasting horizon since new data could be sampled within short time intervals. Therefore, models need to be executed fast in order to reflect the changes in the output based on newly received batches of samples. On the other side of the spectrum, combinatorial modeling typically utilizes copies of the same estimator structure or several different estimators in order to provide more accurate forecasts. Combinatorial modeling is a core component of short-term load and price forecasting approaches as novel approaches in this space could adapt better to a wider range of scenarios and appropriately answer more complex research questions as the learning process becomes more intricate. Given the previously discussed constraints and challenges, combinatorial approaches offer performant alternatives that reinforce prediction accuracy through the simultaneous processing of the same dataset or the partial processing of different data segments [17].

The output of the forecasting framework could denote the estimated values of load or price time series which are then visualized and compared to the actual values in terms of several error metrics. However, recent load and price forecasting research does not strictly utilize the output of the forecasting model as the final estimate since the performance of estimators could be improved *a posteriori* with the implementation of meta-modeling techniques and error refinement algorithms. In this scope, the meta-modeling module often receives the outputs of the forecasting framework and generates a subsequent forecasting task on the same problem formulation. When the meta-modeling technique shares the same structure as the forecasting framework, meta-modeling approaches function as an additional processing layer for prediction refinement. However, the integration of meta-modeling techniques is usually coupled with several changes to the forecasting framework. In this scenario, the main forecasting model may utilize different versions of the given dataset in order to derive intermediate predictions of time series features, enabling the creation of the meta-modeling input dataset. This new input dataset may not reflect the estimated values of load or price directly, but it may contain time series features that express clear patterns that are strongly connected to the target output variable. Therefore, the goal of the meta-modeling approach becomes the approximation of the target output based on the features extracted from the intermediate forecasting stage [18].

Lastly, the output module includes the visualization and prediction evaluation tasks that measure the overall performance and accuracy of the model. Researchers, examine those results in order to make insightful comments towards the enhancement of future forecasting approaches in the energy sector [19]. Furthermore, this module could be considered as the second stage of *a posteriori* processing as subsequent models could utilize the estimated time series to extract and analyze the error component through feedback mechanisms. After the completion of the above processes the model could be deployed on real world applications and integrated into energy management systems. Figure 2.1 presents the diagram of the short-term forecasting pipeline in the energy sector after the distinction of the main modules and tasks presented in this section. The following sections analyze each module one by one and provide a thorough presentation of the prominent methods as well as a direct association of several challenges that could have an impact on the performance of each process.

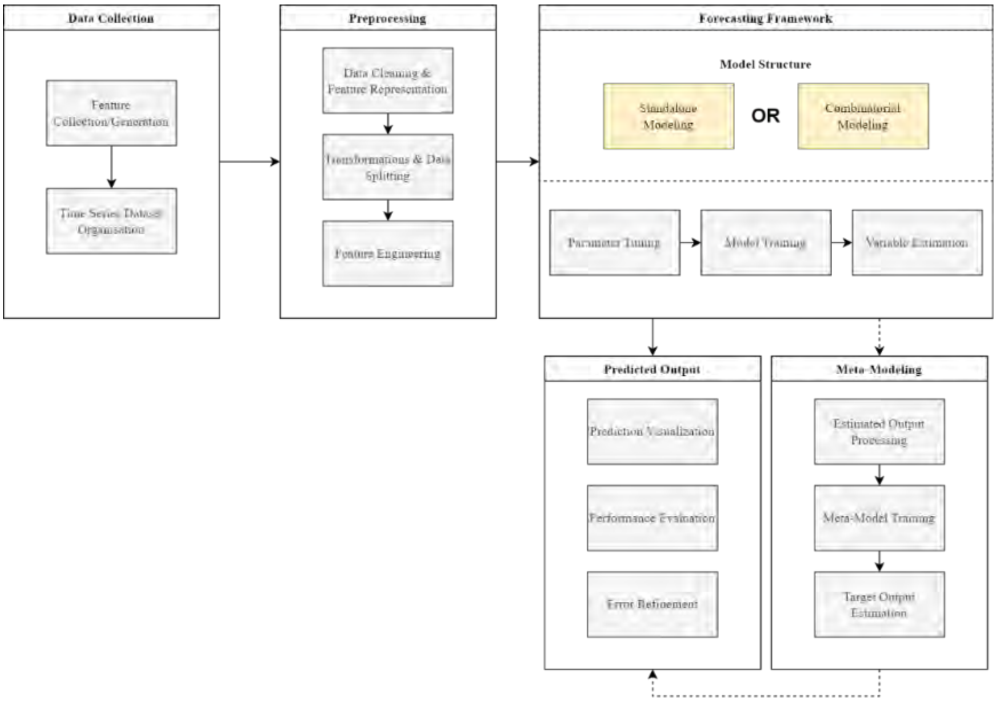


Figure 2.1: Process pipeline for short-term time series forecasting in the energy sector. Solid arrow lines denote the typical flow of information from the data collection module to the predicted output and dashed arror lines denote the meta-modeling direction often followed in novel research projects.

2.2 Data Collection

The emergence of new and digital technologies has enabled the development of several energy data collection strategies for the registration and management of essential forecasting variables such as load and price as well as several influencing factors such as weather and geospatial data. Recent short-term forecasting models typically utilize energy end-use time series data obtained from administrative sources, surveys, metering techniques and generative models. These strategies could provide large amounts of data to researchers for the study and development of effective energy policies as well as the optimal control of energy demand. Consequently, the data collection module is an integral part of short-term forecasting methodologies in the energy sector since the quality of the available dataset often determines the level of processing that will follow and the level of accuracy that is expected from a model. Therefore, it is important to highlight the most prominent sources and processes involved in energy end-use data collection for research purposes, denote the main advantages and disadvantages for each one and evaluate the impact of these approaches on the output datasets. Figure 2.2 presents the categorization of data collection approaches and the following subsections analyze each strategy, providing an overview of widely used practices [20].

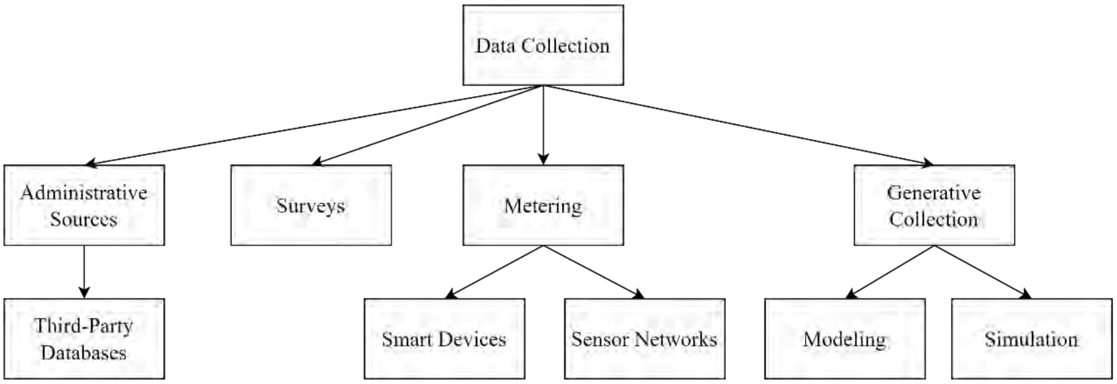


Figure 2.2: Data Collection Methods

2.2.1 Administrative Sources

Time series features collected from administrative sources are often provided by governmental entities, agencies operating at a national, state and local level, energy utilities and energy market participants. These datasets often include detailed statistics for energy consumption, price and several macroeconomic parameters at high volumes and flexible sampling rates. The data is typically stored in databases and made available through websites and published reports. Therefore, the datasets utilized in forecasting methodologies are often obtained through direct access, data mining techniques and web crawling approaches.

It is evident that there are several significant advantages in the collection of energy data through administrative sources. First, the storage of data in databases and the inclusion of a wide range of parameters for each record enables fast and cost-effective retrieval while ensuring higher quality standards. Time series samples are protected against duplicates and the overall higher data granularity enables more complex querying, hence boosting the interpretability of features. Second, time series collected from administrative sources could provide better population coverage since the participating entities have the resources to monitor the energy activity of a wide range of consumers and the usage of several building types. The advanced monitoring capabilities of those organizational units coupled with the potential for real-time simultaneous data collection from multiple sources through the use of modern web crawling tools often result in the extraction of feature-rich datasets that capture complex relationships within the client bases. These datasets are suitable for a wide range of forecasting tasks depending on the sampling rate and could be utilized by several different studies for the examination of a plethora of events in the energy sector. Moreover, since that data is provided by established sources in the energy domain, the validity and integrity of the data is reinforced when compared to other open access alternatives and synthetic datasets.

On the other side of the spectrum, several disadvantages and challenges could be associated with data collection based on administrative sources. It is made clear that the resulting datasets may not always satisfy the needs of a specific research question directly as the features and the records may be defined differently in order to meet the needs of the providers. Therefore, several processing steps may be necessary for the data to be rendered suitable for forecasting models. This process may slow down research output significantly as a thorough data exploration is needed. Client and building identification as well as feature association could be difficult challenges since the diversity of registration formats leads to inconsistencies in data linking. Furthermore, access to those datasets could be difficult and more time-consuming since bureaucratic procedures may be involved for the approval to utilize and modify the data for research purposes. Additionally, third party data providers may consent to the use of datasets but the presentation and access of that data through published work may still be limited or restricted. Consequently, research utilizing those datasets could lead to irreproducible results and conclusions that may not be easily traceable, hence imposing more restrictions on future research efforts due to that lack of transparency [21].

2.2.2 Surveys

Surveys could be conducted for the collection of energy data in order to capture consumer behaviors and patterns valuable to energy demand management. Survey-based methodologies could be divided into two commonly used types, the production and consumption surveys. Production surveys primarily focus on energy supply and gather information about fuel receipt, generation, production and shipment. Consumption surveys gather end-use energy consumption data from different types of clients and buildings in order to cover several use cases, such as the examination of residential or industrial load patterns throughout the year. Survey design focuses on the designation of the optimal type and frequency of data as well as the selection of an appropriate target group in order to extract unique and unbiased samples for each use case. Additionally, the selection of an effective sampling method coupled with a robust validation and dissemination strategy could be considered for the development of an insightful survey in the energy sector. Surveys could be conducted through the traditional paper-based questionnaires and interviews or through more modern methods such as website forms and smart phone applications.

The utilization of surveys could be beneficial towards the collection of energy-related features and the organization of robust short-term forecasting datasets since there are several arguments that denote the positive impact of surveys in this research space. The flexibility of structure and the increased availability of survey methods result in cost-effective data collection processes that could target specific research questions and follow the research scope closely. Since the overall scope of surveys tends to be relatively narrow, data selection becomes more efficient. Moreover, the types of questions answered in surveys lead to a more natural interpretation of energy data that could boost the predictive performance of forecasting tasks after additional processing. Features extracted in the form of linguistic variables or numerical values that could be mapped to different feature states enable the utilization of fuzzy logic approaches for the interpretation of relationships through the derivation of fuzzy rules. Therefore, survey features could be used directly as the input of certain forecasting approaches or serve as complementary data that boosts the interpretability of more complex models.

Surveys add significant value to the data collection module but there are several drawbacks that need to be outlined in order to fully understand the role of these methods. First, surveys could require a higher amount of available resources and well-trained staff in order to guarantee high quality data. Since those prerequisites are not always disclosed in the final endpoint where survey data becomes available to researchers, this uncertainty could impact the overall interpretability of research efforts. Second, there is uncertainty surrounding the content of survey responses as some survey questions may remain unanswered or receive incomplete and biased responses from the population. This results in datasets that may contain missing values and sometimes noisy data, requiring further processing in order to extract suitable features. Furthermore, sampling errors such as duplicate records are more likely to occur in survey data since the first layer of record registration and storage may not be as robust as the one utilized by administrative sources. Lastly, survey data may have access restrictions based on data policies regarding data protection, hence resulting in limited data availability for future research tasks [22].

2.2.3 Metering

Metering methodologies are becoming increasingly popular approaches for data collection in the energy sector as technological advances enable the use of sophisticated infrastructures that are capable of measuring large volumes of energy data. Metering approaches rely primarily on smart devices such as smart meters, sensors, lighting and plugs in order to extract features related to electricity consumption, consumer patterns as well as several influencing factors such as environmental and weather data. Metering data is utilized in a plethora of forecasting studies since the integration of these types of equipment results in granular measurements that are suitable for direct use in short-term and very short-term forecasting tasks. These methodologies could contribute towards the enrichment of survey and modeling datasets, providing a detailed view of dynamic and evolving environments through the extraction of time series.

The contribution of metering methodologies and their role in energy end-use data collection processes vary depending on the type of metering equipment. The integration of smart devices such as smart appliances, reinforces the concept of the Internet of Things and leads to the extraction of real time information, contributing towards the thorough understanding of consumer patterns. Smart devices provide direct feedback to energy applications and management systems as well as indirect feedback for the development of billing strategies and energy audits. Furthermore, smart meters enable the measurement of electricity and gas related features, resulting in the efficient aggregated tracking of energy demand. It is evident that smart meters record large volumes of data and often expose user characteristics that could be exploited by certain models. Therefore, storage, security and privacy risks need to be addressed for optimal smart meter data collection. Moreover, wireless sensor networks, smart thermostats and smart lighting contribute towards direct metering methodologies that collect environmental features and track heating and cooling parameters. The examination of influential variables such as temperature, humidity and light intensity is crucial for the development of robust forecasting models since they support the estimation of load and contribute towards improved decision-making. Lastly, smart plugs provide a simple data collection pipeline that involves consumption and voltage measurement through hardware and data organization through a management platform.

Moreover, it is worth mentioning that there are several challenges and drawbacks in metering data collection methods. First, the high cost of equipment and maintenance often limits the implementation of large-scale infrastructures. Consequently, the resulting datasets often target the consumption patterns of smaller groups of clients and monitor a limited set of buildings. Additionally, the datasets produced by sensors and smart meters often have high storage and processing requirements, rendering some short-term data analysis tasks infeasible due to the lack of computing power and the difficulty of deriving results within short time intervals. However, technological advances in the energy sector should lead to more cost-effective solutions for infrastructures that utilize metering devices, resulting in the large-scale utilization of smart meters and the availability of resources for big data processing. Furthermore, advances in distributed computing could address the computational burden of sensor data through the development of decentralized models. Second, the quality of the resulting dataset is not guaranteed as there is a possibility to encounter missing values and noisy features due to data interruption, corruption and interception risks. Therefore, quality assurance criteria need to be examined before the datasets are made available for research and energy applications in order to reinforce reliability [23].

2.2.4 Generative Data Collection

Short-term forecasting tasks in the energy sector often require large amounts of data in order to study specific use cases and analyze the complex dynamics of energy systems and energy markets. Since the required features that address a specific research question need to be strictly defined within the research scope, the data provided by most well-known data collection methodologies needs to be suitable for the formulation of the research problem and contain a sufficient number of samples for the development and validation of robust forecasting models. However, several third-party data collection processes follow policies that may limit or restrict data access, contributing towards the scarcity of suitable datasets. Additionally, the available data provided to researchers in the energy sector may not always follow ideal data collection processes, resulting in poorly structured datasets that contain a low number of samples. It is also worth noting that when smaller datasets are considered for specific forecasting tasks due to their high compatibility with the research scope, the need to perform larger scale tests for the examination of scalability often requires dataset expansion. Therefore, generative data collection methodologies are utilized in order to address the challenges surrounding the overall difficulty of obtaining high volumes of suitable high-quality data.

Generative approaches mainly rely on models and simulations in order to increase the number of samples from an existing dataset, combine smaller datasets cohesively and generate data approximations for a given research task when no data points are provided. Modeling methodologies typically receive an input dataset and based on a set of assumptions; multiple processing cycles generate output samples. This type of generative data collection is often utilized for the expansion or combination of smaller datasets since the required set of assumptions associated with the data distribution and time series characteristics could be easily derived from the smaller sets of samples [24]. Simulation processes typically generate datasets based on a set of parameters that denote the initial state of the environment under study as well as its evolution over a specified period of time [25]. It is evident that simulations add significant value to data collection since they could produce high volumes of data without the need of an initial input dataset and offer more flexibility in energy research. Researchers could explore a plethora of scenarios through simulations given the deterministic process that sets the parameter values, resulting in faster and more robust experiments. Moreover, hybrid approaches combining modeling and simulation structures could provide increased flexibility in the selection of input and the finalization of environmental parameters.

A hierarchical categorization of generative approaches based on the type of input data utilized in these data collection processes distinguishes two types of methodologies, the top-down and bottom-up methods. The top-down methods utilize aggregate features in order to produce more samples through the estimation of energy variables such as the energy demand, whereas bottom-up methods utilize disaggregated input [26]. Since aggregate input features may not express the behavior of the individual components of a system accurately and disaggregated input data may not always comply to all general restrictions of the system simultaneously, the generated output samples may contain inconsistencies as the target generated feature values may be overestimated or underestimated.

The generation of samples for the creation, extension and combination of datasets involves several risks that could have an impact on the performance and integrity of forecasting models. First, it is clear that modeling and simulation approaches operate through the execution of several processing stages in order to provide readily available data that could easily be integrated in relevant research tasks. Therefore, the quality of the data is directly dependent on the accuracy of the processing tasks. Additionally, the availability of the generated samples is dependent on the complexity and the response time of the models. These dependencies indicate that suboptimal and poorly designed processing tasks could negatively impact the generated samples, compromising the quality of the output dataset. Furthermore, it is worth noting that the development and implementation of a robust processing pipeline is a time-consuming task that could delay research output. Second, the dependency of the modeling approaches on data assumptions and the dependency of simulations on environmental parameters require expert knowledge and a thorough examination of edge cases in order to reinforce reliability. These dependencies introduce a level of uncertainty since the studied environments in the energy sector are dynamic and rapidly evolving. Lastly, transparency and traceability risks could emerge and have an impact on the integrity and reproducibility of research projects due to the intricacies of processing tasks. Therefore, research on the field should include a comprehensive overview of the processing tasks utilized for data generation in order to reinforce the clarity of the presented work.

2.3 Processing

The processing module is one of the most important components of short-term time series forecasting approaches in the energy sector since the dataset needs to be appropriately prepared for model training. The dataset derived from data collection methodologies needs to be compatible with the data format of a studied model in terms of structure and address all model requirements for optimal performance. Additionally, the dataset needs to include useful features that are relevant to the studied research questions and could boost forecasting accuracy. Therefore, in this section we examine the primary tasks involved in time series processing for forecasting tasks and discuss about their impact on the research pipeline and the respective challenges that may arise. The following subsections present an overview of data cleaning, feature representation, data splitting, transformations and feature engineering approaches since these are the prominent tasks executed in this module.

2.3.1 Data Cleaning and Feature Representation

The datasets derived from data collection methodologies may include missing values, duplicate data points and errors depending on the quality standards of the data collection process. It is often observed that the datasets made available for research in the energy sector have undergone a data cleaning process at the source in order to boost data quality and reduce the effects of erroneous samples, however this process could also be executed by the researchers if the dataset is still poorly structured.

2.3.1.1 Missing Values

Missing data and missing components of aggregate features could be detected and omitted from the initial dataset for simplicity or could be replaced with more useful samples through data imputation. When the missing rate is low, typically below 5%, the deletion of missing data may not have a significant impact on dataset quality [27]. On the other side of the spectrum, data imputation is preferred when the missing rate is high since the datasets would be incomplete and the task of learning consumption or electricity price patterns would not be possible. Therefore, several data imputation methods were developed and used frequently towards the improvement of energy time series datasets. One of the simplest categories of data imputation methods attempts to replace the missing samples based on information available in neighboring samples. These approaches could utilize simple duplication strategies in order to carry the last observation forward or the next observation backward. Alternatively, when ranges of past and future neighboring data regions are considered, interpolation techniques based on linear and nonlinear structures could be utilized to impute missing values. Furthermore, robust neighbor-based imputation methods take into consideration the aspect of local similarity and based on extensive examination of similar data points through well-known clustering approaches such as K-nearest neighbors and DBSCAN they update missing values with the mean value of similar neighbors. Moreover, constraint-based imputation methods attempt to discover dependencies and rules between samples in order to form a set of constraints that could regulate sample replacement. These constraints could be derived from similarity and distance metrics as well as graph structures and networks. This category of methods could be accurate and time-efficient but could also be restrictive at a larger scale since the constraints may not reflect the entirety of dynamics found in real world consumption and energy market datasets [28].

Efficient data imputation could also be achieved through learning-based methods since several subcategories could be identified featuring robust models. First, the subcategory of statistical methods often utilizes traditional data fitting approaches as well as rolling statistics and the use of mean value in order to derive suitable data points considering historical and future data regions. Second, regression models utilize historical data and neighboring data points in order to formulate forecasting models for the prediction of missing values. More recent learning-based approaches utilize neural networks for the development of more sophisticated data imputation models. The most prominent neural network structures used for this task are the multi-layer perceptron, generative adversarial networks and several recurrent neural network architectures coupled with the gated recurrent unit model (GRU) for the processing of long-term data dependencies [29].

Alternatively, methodologies utilizing collaborative filtering could also be useful for data imputation since matrix decomposition models could extract features from the original dataset and based on correlation metrics reconstruct the original data matrix by completing the missing values [30]. Furthermore, expectation-maximization methods could be applicable to data imputation tasks since new data samples that fit the original data distribution could be derived from the iterative tuning of model parameters at the maximization step [31]. It could easily be observed that more powerful data imputation processes could lead to the development of time-consuming strategies that may include the formulation of subsequent forecasting tasks. Therefore, execution time should be an additional concern for short-term and very short-term forecasting pipelines as model recalibration could become slower when new samples that include missing values pass through more complex data imputation structures such as deep recurrent neural networks.

2.3.1.2 Erroneous Data

Apart from missing values, data cleaning methods address the challenge of erroneous time series data through several types of error correction algorithms in order to derive less noisy series with fewer outliers. The selection of error correction method depends on the type of erroneous data and it is possible that multiple methods could be utilized simultaneously towards the improvement of the dataset. Time series datasets typically include continuous, single point and translational errors. Continuous errors refer to abnormal values in multiple consecutive data points. This type of error typically occurs due to noise or malfunction of metering equipment. Additionally, supporting features such as geospatial data could exhibit continuous errors due to interruptions in data transmission or partial data corruption. Single point errors refer to isolated data points that have a small or large distance from the true value. These errored samples could be identified as outliers since they may not follow the patterns in the time series. The existence of continuous and single point errors is common in energy time series surrounding the study of load and electricity price since the study of consumption reflects the potential instabilities of the smart grid as well as the dynamic client habits and the study of electricity price reflects the volatile nature of energy pricing due to several phenomena such as renewable energy penetration and energy market dynamics. Lastly, translational errors could occur due to the poor alignment of timestamps, resulting in a suboptimal arrangement of features since sample values within the same row may not correspond to the same timestep for every feature.

There is a significant overlap between missing value data imputation methods and erroneous data processing approaches since the improvement of the time series data depends on the optimal replacement of samples. Therefore, prominent error value cleaning methods could be classified as smoothing-based, constraint-based, statistics-based or in the wide category of anomaly detection algorithms. Smoothing-based methods attempt to reduce noise through low frequency filtering, moving average and autoregressive processes. However, the application of these methods is not extensive since the risk of data distortion after smoothing could lead to increased confusion and uncertainty in model formulation. Constraint-based methods focus on the detection of several types of dependencies in order to derive rules that could refine the values of samples. These dependencies are typically detected from the order of samples, the value difference in consecutive data samples, the speed of value changes and the temporal structure denoting causative and dependent behaviors. Furthermore, statistical approaches often include maximum likelihood estimation, Markov models, binomial sampling and probabilistic models for the discovery and examination of patterns in historical data, resulting in the estimation of values that could optimally replace specific samples. Moreover, anomaly detection algorithms utilize a plethora of learning structures such as long short-term memory networks, generative adversarial networks and autoregressive moving average models in order to identify and repair data abnormalities of sequences as well as standalone samples. Lastly, dynamic programming and distance-based clustering approaches are utilized towards the mitigation of translational errors and the optimal alignment of features [28].

2.3.1.3 Feature Representation

The structure of features and the way they are presented within a dataset could provide significant benefits to data exploration, research problem formulation and forecasting model design since the clear representation of time series variables and the detailed description of influencing factors reinforce interpretability and could contribute towards the efficient discovery of patterns. Optimal feature representation techniques shift the complexity from the forecasting framework and the individual model to the examination of features. In the energy sector, this is an important task for most forecasting models due to the complex dynamics that exist in smart grids and energy markets as well as the wide set of influencing factors that affect client consumption. Consequently, two main categories of feature representation methods could be identified. The first category refers to contextual representation methods that attempt to alter existing features and introduce new ones in order to enrich or compress the contents of a dataset, following the specifications of the research task closely. The second category refers to structural representation methods that mainly alter existing features in order to accommodate the assumptions and the computational path of specific forecasting models. Both categories are valuable for data processing and it is evident that a combination of techniques from those representation method sets is typically utilized before further feature processing tasks are executed.

Contextual feature representation methods operate as preliminary feature engineering and selection layers in order to expand or shrink the available feature space in ways that increase the compatibility of features with the scope and goals of the research task. Feature representation methods focusing on feature space expansion typically include disaggregation, fuzzification, statistical and temporal enrichment. Disaggregation techniques are primarily utilized for the decomposition of existing features into more detailed components. These methods commonly apply unsupervised learning methods as well as edge detection to general consumption data in order to isolate appliance features and denote events in the studied environment. Additionally, fuzzification techniques could be applied in order to generate a new set of linguistic variables from an existing feature. Fuzzification is suitable for features that describe nondeterministic quantities with uncertainty such as influential weather variables. The resulting fuzzified features describe the degree of membership that maps the original value to the linguistic variables, boosting the overall interpretability of the dataset [32]. Furthermore, statistical and temporal enrichment describes the process of feature space expansion through the inclusion of additional statistical and temporal variables extracted from the original dataset. This simple process typically involves the calculation of rolling statistics such as rolling mean and the inclusion of lagged variables that could describe load or price historical data at different time steps. It is worth noting that temporal enrichment could also refer to the inclusion of simple time variables extracted from timestamps such as the hour or the day [33].

On the other side of the spectrum, feature space shrinkage may be necessary in some forecasting tasks since low dimensional datasets containing fewer features that summarize the factors affecting the target variables concisely may result in simpler and easily interpretable models. Contextual feature space shrinkage typically includes aggregation tasks. Aggregation approaches utilize simple mathematical models, statistical methods and unsupervised learning algorithms in order to summarize features given a specific research direction [34]. For example, total demand and price forecasting tasks as well as client group consumption analysis often require the summation of load or price features from multiple sources and the organization of clients into distinct groups based on their common characteristics.

Structural feature representation methods focus on alternative dataset organization approaches such as time series encoding and some dimensionality reduction techniques in order to derive an equivalent dataset structure that describes existing variables differently based on assumptions and observations. These methods attempt to increase the compatibility of the available dataset to the studied forecasting structure without significant compromises in data quality. Encoding techniques exploit existing structural characteristics of some feature types in order to derive more detailed representations that express a more accurate mapping of those features to time series data. Prominent approaches in time series encoding are one-hot, cyclical and radial basis function encoding. One-hot encoding typically targets categorical influencing factors and time-related information for the introduction of dummy variables that have specific values only in the rows where the mapping of the sample to the variable is valid. Consequently, the introduction of those sparse features may lead to reinforced interpretability and robustness when compared to the original non-sparse representation [35]. Cyclical encoding methods acknowledge the continuity of some variables and transform them utilizing trigonometric functions such as sine and cosine. This transformation could clearly expose periodic patterns in an easily interpretable way that could simplify the training of some forecasting models [36]. Radial basis function encoding methods utilize distance metrics in order to derive curves that denote the closeness of each sample to a specific value, providing the model with clear relationships of the features to crucial reference points [37]. The role of dimensionality reduction techniques in the feature representation stage remains simplistic since complex transformations that reduce the shape of features typically need to be applied after data splitting in order to use only the training data as reference. Therefore, dimensionality reduction methods in this scope extend the main principles of aggregation tasks through the inclusion of compression and vectorization strategies that could boost pattern visibility [38].

2.3.2 Data Splitting

An important step in dataset processing is the implementation of data splitting strategies since there are several benefits contributing towards forecasting model robustness and evaluation fairness. Two main data splitting directions can be identified in forecasting tasks in the energy sector. The first direction refers to data splitting for the purposes of model training and evaluation given general design guidelines for optimal forecasting. The second direction refers to problem-specific data splitting in order to derive several datasets that could be processed separately from the same forecasting framework.

Training and evaluation oriented data splitting approaches are mandatory in most forecasting tasks in the energy sector since they express the general learning procedure where a model receives a specific set of samples that are considered as known data in order to tune its parameters in a way that when new samples considered as unknown data are given as input, the predicted output of the model is close to the actual values in that unknown data segment. Therefore, data is typically split in a training set, a validation set and a test set. The training set represents the known data segment given to the forecasting model for pattern discovery and initial parameter learning. The validation set represents the unknown data segment used for the optimization of estimator parameters. Lastly, the test set represents the unknown data segment utilized for estimator performance evaluation. This data splitting process is often executed before any data transformations and feature engineering techniques are applied. Consequently, feature processing methods that aim to alter the properties of a dataset are applied based on training data samples in order to eliminate bias. Short-term time series forecasting tasks in the energy sector often require several years of historical data for model training and utilize the most recent years of data samples as unknown data for parameter tuning and performance evaluation. The dataset is split based on a specified splitting ratio. The most commonly used ratios in the literature split the data 80%-10%-10%, 70%-15%-15% or 60%-20%-20% for training validation and test sets respectively. It is worth mentioning that many methodologies in this research space unify the validation and test sets. This unification leads to the identification of distinct model evaluation strategies such as holdout validation and cross-validation that will be analyzed further in the examination of the forecasting framework and the output module since the processing module focuses on the role and structure of the training module [39].

Problem-specific data splitting approaches are applied on the training, validation and test set equivalently in order to derive several smaller datasets of explicitly specified dimensions. Data splitting approaches in this scope often address research tasks that utilize load data from different types of clients and electricity prices from different sources. The main goals of those methods are to create well-separated datasets that could be passed to the same model or to several different models within a forecasting framework in order to estimate the values of target variables partially or to provide different output representations depending on the structure of the input. Consequently, clustering algorithms such as k-means are utilized for the segmentation of the dataset based on distance metrics. This data segmentation could enable diverse and localized data processing and feature enrichment techniques as the unique characteristics of each data segment could be exploited for performance improvement [40].

2.3.3 Data Transformations

The separation of data into a known training segment and unknown validation and test segments enables a series of impactful transformations that optimize and rescale the input based on the manipulation of time series properties for efficient processing in the forecasting framework. These transformations are applied to the training data segment and the unknown data segments are subsequently transformed based on the transformation principles of the training set in order to avoid bias and data leakage. However, transformed input data samples lead to the derivation of transformed predicted output samples that need to be reverted back through the application of the inverse transformation for performance analysis. Additionally, it is evident that some research efforts may utilize only a small number of data transformations based on model requirements or no transformations in order to minimize the impact of processing performance benefits while isolating the forecasting structure for performance evaluation. Therefore, while there is a large set of data transformations for time series data that could be suitable for forecasting tasks in the energy sector, there are only a few prominent methods that are utilized situationally. The most widely used methods discussed in this section include power transformations, differencing techniques, standardization and normalization.

It can be observed that due to the complexity of consumption patterns and the occurrence of seasonal trends in load and electricity price observations, energy datasets may include non-stationary features as the mean and variance shift over time. Consequently, the available data may not follow a normal distribution and the overall instability caused by the increased variance values could affect the performance of some statistical and machine learning forecasting models. As a result, power transformations could be utilized in order to stabilize variance and reinforce clarity in feature correlation analysis [41]. These transformations apply a set of power functions that attempt to nullify the effects of the trend based on the function that best explains the shift in variance. For example, the effects of quadratic trends could be stabilized through a square root transformation and exponential trends could be reduced or removed through a logarithmic transformation. In this scope, it is worth mentioning that the box-cox power transformation utilizes the exponent lambda (λ) as a decision variable in order to detect the appropriate power transformation for a given time series y, resulting in an optimal approximation of a normal data distribution through the formula:

$$y(\lambda) = \begin{cases} \frac{y^{\lambda} - 1}{\lambda} & if\ \lambda \neq 0 \\ \log(y) & if\ \lambda = 0 \end{cases} \tag{2.1}$$

The impact of trends and seasonality that render time series data non-stationary could also be addressed through the utilization of differencing transformations that help stabilize the mean values. Differencing methods calculate the differenced time series by subtracting an observation at a past time step $t - n$ from the current observation at timestep t. Since the timestep denoting the previous sampling interval is typically utilized for the calculation of the differenced series, n usually takes the value of 1. However, the value of n depends on the temporal structure as well as the problem formulation. Furthermore, differencing could be applied multiple times when the trends are nonlinear in order to eliminate any instabilities that may still persist. The inverse operation involving the addition of the previous observation is applied when the predicted series needs to be converted to the original scale for performance evaluation [42]. Given the observed value of the time series at time t denoted as $y(t)$ and a previous observation $y(t - n)$, the differencing term at time t denoted as $diff(t)$ can be defined by the formula:

$$diff(t) = y(t) - y(t - n) \tag{2.2}$$

Energy datasets usually contain time series features of different units with values at different scales. Therefore, these independent variables may not contribute equally to regression analysis tasks and could lead to biased predictions as they may follow different distributions. As a result, the performance of models that assume a normal feature distribution such as linear regression and support vector machines could be affected negatively. Standardization is the suitable data transformation method that could provide a solution to this problem since the time series features are modified to have a mean value of 0 and a standard deviation of 1, following the behavior of a standard normal distribution [43]. Given the mean value μ and standard deviation value σ of time series features with values denoted as $x(t)$ at timestep t, the standardized time series values denoted as $stx(t)$ could be calculated through the formula:

$$stx(t) = \frac{x(t) - \mu}{\sigma} \tag{2.3}$$

Moreover, in scenarios where the data distribution may not be known and the contribution of features is affected by their value range, the scale of time series could be adjusted in order to accommodate the assumptions of some machine learning methodologies such as neural networks that require an appropriate data scaling strategy for the effective usage of activation functions. Consequently, the normalization task is applied in order to transform the data and bring all samples within a common value range. Typically, the preferred range for data scaling is between 0 and 1. This process is applied to the training set and based on that scaler, the remaining validation and test sets are transformed accordingly [44]. Given the values of time series observations denoted as $x(t)$ at timestep t as well as the minimum and maximum values of the features denoted as $min(x)$ and $max(x)$ respectively the normalized values $nrx(t)$ could be calculated through the formula:

$$nrx(t) = \frac{x(t) - min(x)}{max(x) - min(x)} \tag{2.4}$$

2.3.4 Feature Engineering

Feature engineering tasks expand on the principles of feature representation and feature transformation processes in order to derive new features, select the most appropriate ones for a given model or modify existing ones in an attempt at finalizing the input dataset based on the properties of the training data. These tasks focus more on the improvement of prediction accuracy and convergence time of the forecasting model and could be utilized for the development of a research project based on the low-level inspection of the forecasting framework. Three categories of prominent feature engineering tasks are presented in this section. The first category refers to feature decomposition approaches since the component-wise analysis may benefit the forecasting performance of some models. The second category refers to feature projection techniques since the mapping of the feature set could expose specific characteristics that may be valuable for training or provide solutions towards dimensionality reduction, resulting in faster convergence times. Lastly, the third category refers to feature selection tasks that evaluate the significance of features and derive a set of the most important ones for the prediction of the target variable.

2.3.4.1 Feature Decomposition

Feature decomposition approaches focus on the extraction of recurrent and non-recurrent time series components. Recurrent time series components in the energy sector mainly consist of the average value of load and price, increasing and decreasing trends as well as short-term cycles repeating throughout the year that denote a seasonal pattern in consumption, electricity transactions or other influencing factors such as weather data. The primary non-recurrent component considered in this decomposition refers to noise that could be present due to irregular events and poor data quality. Alternatively, the average value of the time series including any random fluctuations that could occur due to the effect of noise could be treated as the residual component which denotes the values that remain after the extraction of the trend and seasonal components. These components could be processed individually for robust estimation or utilized for parameter tuning and model selection. Decomposition approaches mainly consider an additive or multiplicative relationship between the components. Additive decomposition is suitable for datasets where a change in the average value of the series for a specific time period is not proportional to the variation exhibited in the trend and seasonal components. When the variation of trend and cycle are proportional to the time series level, a multiplicative decomposition is preferred. The formulation of a purely additive or multiplicative time series decomposition method is usually preferred for simplicity. Given a specific time period t, a time series y_t, a seasonal component S_t, a trend component T_t and the residual series R_t, classical additive and multiplicative decomposition could be formulated through the respective equations:

$$y_t = S_t + T_t + R_t \tag{2.5}$$

$$y_t = S_t \times T_t \times R_t \tag{2.6}$$

However, additive and multiplicative relationships between components could coexist. This phenomenon coupled with the need to control how fast each component changes and handle outliers efficiently lead to more decomposition methods that extend the knowledge of the classical additive and multiplicative approaches such as the Seasonal Extraction in ARIMA Time Series (SEATS), the X11 and the Seasonal and Trend decomposition using Loess (STL) methods [45].

2.3.4.2 Feature Projection

Feature projection techniques focus on the derivation of alternative time series representations that contribute towards dimensionality reduction and efficient context-dependent decomposition while improving the efficiency of time series clustering approaches and enabling the application of sophisticated feature selection methods at a subsequent step. One of the most prominent methods in this research space is principal component analysis (PCA) which is utilized for the extraction of low-dimensional dataset of uncorrelated components that maintain a high percentage of the variance found in the original data. This method utilizes an orthogonal transformation in order to map a high-dimensional dataset to a smaller set of components through the examination of the variance-covariance matrix. The algorithm of PCA was modified to accommodate several time series forecasting tasks in this research field and several other alternative methods focusing on different aspects of feature engineering were subsequently developed [46]. Piecewise Vector Quantized Approximation is an equally important technique in time series dimensionality reduction as it provides a symbolic representation of time series through the mapping of sequence segments based on distance metrics. This method could also enhance similarity analysis and clustering tasks for robust feature selection [47]. Furthermore, methods such as the t-stochastic neighbor embedding (t-SNE) are adapted to time series in order to visualize datasets containing consumption or price data from multiple customers at a low dimensional space and validate the efficiency of time series clustering before the data is passed to the forecasting model [48]. Time series clustering results could also be enhanced through the Uniform Manifold Approximation and Projection method (UMAP) by providing a topological data representation strategy [49]. Moreover, Singular Value Decomposition (SVD) could be utilized for dimensionality reduction as well as separation of random effects that could cause noise for further examination [50].

2.3.4.3 Feature Selection

Feature selection methods utilize importance metrics and visualization techniques in order to reduce the total number of features and derive the set of the most significant ones. These techniques reduce the dimensions of the dataset and could lead to improved forecasting performance as features that could have a negative impact on the training process and the generalization capabilities of a model are not included. For example, short-term load predictions could benefit from the detection of important environmental features such as temperature and electricity price predictions could be improved if the exogenous variables directly connected to price volatility such as fuel costs and system load are included in the input dataset.

It is evident that several dimensionality reduction techniques discussed in the previous section as well as clustering methodologies contribute towards the selection of impactful features. Methods that focus on the mapping of features to a low dimensional space such as PCA could provide insightful information towards the identification of important features through the examination of coefficients used to combine the dataset columns. Higher coefficient values denote increased importance of the candidate features. Additionally, clustering methodologies reinforce feature similarity through distance metrics and focus on the inclusion of features that match a set of criteria while removing less relevant columns. These methods provide an indirect quantification of feature importance that depends heavily on data structure.

On the other side of the spectrum, modeling methodologies and importance scores provide a more direct quantification of feature significance and result in the straightforward and simplified understanding of the data that could subsequently lead to a better understanding of the forecasting model. Importance thresholds are defined based on research assumptions and performance expectations. The features that are connected to weights or importance scores below the specified thresholds are eliminated. Since short-term forecasting tasks in the energy sector utilize supervised machine learning regression techniques, the coefficients of those models could be used as direct indicators of feature importance. Therefore, widely used feature selection strategies involve fitting a simple model such as linear regression on training time series samples in order to derive the weights that denote the significance of each feature. Additionally, decision tree-based tasks adapted to regression methodologies could be utilized for importance evaluation and feature selection since importance scores could be extracted based on the reduction of the criterion that evaluates the quality of decision rules leading to splits in the structure. Consequently, simple decision trees, random forest models or more sophisticated stochastic gradient boosting algorithms such as XGBoost could retrieve feature importance scores. Generalizing the process of deriving importance scores leads to the implementation of permutation techniques that could utilize any model as the base structure and given a variable combination of features, the importance scores are derived from the iterative performance evaluation of the model. Permutation approaches enable the development of highly complex model structures but the size of the feature set and the training time of the model impose restrictions in terms of time complexity that could render these approaches infeasible for short-term forecasting pipelines. More robust feature importance methods typically utilize a combination of simple models and derive the optimal set of features based on the cross-examination of importance scores [51].

Furthermore, influential features could be selected through the examination of causality. Statistical hypothesis tests such as the Granger causality test take into consideration the evolution of time series variables and attempt to express a causal relationship between features under the principles that a causal series happens before the feature that expresses the effect and the causal series contains unique and useful information about the effect series. The examination of causality as a feature selection technique is valuable to short-term forecasting methodologies in the energy sector since the interpretation of events that could cause fluctuations in the target variables of load and price could be clearly identified through a smaller set of features [52]. Additionally, an equally important criterion that contributes to optimal feature selection is the quantification of similarity. Since the regression tasks that are based on data extracted from diverse customer sets and different building types is often clustered before forecasting, distance metrics could support the identification of the features that are more closely connected to the target variable [53].

Time series datasets in the energy sector may contain influential factors that are not derived from the target series as well as features that represent the target series at a previous time step, known as lags. The previously discussed feature selection tools could have general applications on all types of features. However, the examination of correlation is often considered as one of the most prominent processes for time series feature selection when lagged series are included in the dataset. Correlation denotes the type of association between two variables as positive, neutral or negative, depending on how the values of those variables change. Positive correlation denotes a change in the same direction for both features, neutral correlation denotes the absence of relationship as the values of the variables change and negative correlation indicates that the values of the variables change in the opposite direction. The detection of significant correlation between an influential feature and the target variable could be valuable for the improvement of model performance as the inclusion of this feature could strengthen the expression of patterns and relationships in the data. However, correlation between independent influential factors could be an indicator of multicollinearity, leading to unreliable forecasting performance. Consequently, some features may need to be eliminated in order to avoid this risk. Correlation could be used to study linear relationships in time series features under the assumption of a normal distribution and the absence of outliers through the calculation of Pearson correlation coefficients. These coefficients could be calculated based on the covariance cov and the standard deviation σ of the features [54]. Therefore, given variables X and Y, the Pearson coefficient p_{coeff} is calculated as follows:

$$p_{coeff} = \frac{cov(X,Y)}{\sigma_X \times \sigma_Y} \tag{2.7}$$

Moreover, the study of nonlinear relationships based on data that may not follow a normal distribution could be conducted through the calculation of the Spearman correlation coefficient s_{coeff} that considers the covariance and standard deviation of the rank r of values in each feature [55]. This alternative correlation coefficient is calculated based on the formula:

$$s_{coeff} = \frac{cov(r_X, r_Y)}{\sigma_{r_X} \times \sigma_{r_Y}} \tag{2.8}$$

The values of the coefficients range from -1 to 1 with -1 denoting perfect negative correlation and 1 denoting perfect positive correlation. The study of correlation could provide significant insights in the examination of lagged time series features. The autocorrelation plot is utilized in order to present the correlation values for the time series at different time steps and a confidence interval is specified, denoting that correlation values outside of its boundaries are statistically significant. The autocorrelation denoting the relationship between a time series feature and a shifted version of this series at a prior time step contains the direct correlation between them as well as the indirect correlations that could occur due to intermediate time steps. This information could be complex and for the purposes of feature selection the isolated correlation between the series and the lagged version may be needed in order to form the decision to keep or eliminate the lagged series from the feature set. Therefore, the partial correlation could be calculated and plotted in order to examine this relationship [56]. These computations and visualization techniques contribute towards more robust statistical and machine learning models since regression tasks in the energy sector typically include lagged series for the target variables of load and price in the feature set.

2.4 Forecasting Framework

The forecasting framework is the core module of every forecasting process since it is the main component for model design and development. This module includes the selection and implementation of forecasting algorithms as well as several processes for the tuning of hyperparameters and the optimal training of the models based on known observations. The estimated target variables are provided as the output of this module for interpretation, performance evaluation and subsequent processing. The forecasting framework structure depends primarily on the decision regarding the forecasting philosophy followed for the resolution of a specific research task. This decision influences the model selection as well as the tuning and the training process. In the energy sector, several approaches utilize a single forecasting algorithm in order to derive short-term predictions of load and price and continually attempt to explore different “what if” scenarios. Through this exploration, researchers seek to improve the performance of this standalone structure through several modifications that are connected to parameter values and changes in modeling assumptions. Alternatively, research efforts could utilize a combination of forecasting methodologies in order to derive more reliable and highly performant models at an increased structural complexity. As a result, a plethora of standalone and combinatorial models that have a statistical and artificial intelligence background are applied to a wide variety of research tasks contributing towards the evolution of this research area and providing efficient solutions for the prediction of important energy related variables. Therefore, in this section an analytical discussion of standalone and combinatorial modeling approaches presents the prominent statistical and artificial intelligence methods, the main goals of each structure as well as their respective challenges and limitations. Additionally, an overview of prominent hyperparameter tuning techniques follows this analysis since it is necessary to present the methodologies that could optimally handle the parameters of a forecasting structure towards the exploration of alternative scenarios and subsequent error reduction.

2.4.1 Standalone Modeling

Standalone modeling approaches focus on the design and implementation of a centralized forecasting structure that features a single estimator which processes the input dataset. This structure could include a set of parameters that may be tuned in order to optimize the performance of the forecasting structure globally and the output typically describes the estimated target variable since there is a direct flow of information that connects the training and evaluation process to the estimated values. Short-term forecasting in the energy sector utilizes standalone modeling in order to derive baseline models for benchmarking or optimized models for applications that involve more versatile and modular feature processing methods. Since standalone models follow a single forecasting algorithm, they are more reliable in terms of training time, rendering them suitable for applications with strict time constraints. Additionally, standalone models often follow a simple structure that could reinforce the interpretability of a forecasting task as well as the reproducibility of research results since implementations and comprehensive experiments could be easily available through several research efforts. Standalone forecasting models could be organized into the two main categories of statistical and artificial intelligence models which could be further analyzed in the following subsections.

2.4.1.1 Statistical Forecasting Models

Statistical forecasting models refer to traditional parametric and non-parametric methodologies that attempt to capture and interpret linear and nonlinear relationships through averaging techniques, time series decomposition and regression analysis. These approaches were developed based on simple yet powerful mathematical concepts that could lead to satisfactory short-term time series predictions without requiring a restrictive amount of computing resources. The most prominent sets of methods utilized in short-term energy forecasting are exponential smoothing, moving average and regression estimators.

Exponential smoothing methods primarily rely on the formulation of weighted averaging techniques that utilize an exponential decay mechanism in order to convey the decreased impact of time series lags as the time window between the studied series increases. This category of methods often explores core time series components such as trend and seasonality in order to derive a suitable smoothing technique. When trend and seasonal components cannot be easily detected, a naïve approach is usually adopted where future observations at timesteps $t + n$ have the same value as the current observation at timestep t and the model focuses on the most recent observation as past observations do not provide any information about the future. Therefore, any forecasted future observation could be expressed as an average of the current observation where every term has equal weights. This is an edge case that sets the basis for exponential smoothing approaches. Expanding on this concept, the smoothing parameter a could be defined within the range of 0 and 1, denoting the level of smoothing increase when the value of the parameter is small and smoothing reduction when the value is large. [57] Following this step, the exponential decrease of the weights that could express the decreased impact of past observations for the point forecast of the next observation $\hat{y}_{t+1|T}$, given the smoothing parameter, the total number or observations T as well as the first fitted value l_0 could be calculated through the formula:

$$\hat{y}_{t+1|T} = \sum_{j=0}^{T-1} a(1-a)^j y_{T-j} + (1-a)^T l_0 \tag{2.9}$$

While simple exponential smoothing could be effective when no clear trend or seasonal patterns are detected, several extensions of this model cover different scenarios where a specific type of trend or seasonal component could be detected. Consequently, the Holt linear trend method could be utilized when only an additive trend could be detected and the additive as well as the multiplicative variants of the Holt-Winters methods could be utilized when an additive or multiplicative seasonality is present given the detection of an additive trend [58]. These methods introduced trend and seasonality equations featuring additional smoothing parameters β, γ in order to regulate the level of smoothing for those components. Additionally, damped versions of those methods were developed in order to stop the indefinite increase or decrease of the trend for future observations. In the energy sector, an exponential smoothing approach utilized for load or electricity price forecasting is typically modelled as a linear or quadratic curve. The values of the smoothing parameters as well as the initial fitted observation could be tuned for performance optimization in order to derive lower error metrics. Exponential smoothing methods perform well for point forecasts but depending on the forecasting task, these algorithms could be extended in order to output estimated intervals. Therefore, state space models could be developed from the smoothing equations after the definition of a measurement equation that represents each observation as the addition of the previous smoothing level with an error term, the formulation of a state equation that represents the adjustment to the smoothing level and the derivation of the probability distribution associated with the error. The performance optimization of those state space models could be achieved through error metric minimization as well as likelihood maximization techniques. Since the state space models are configured differently based on the type of trend and seasonality exhibited in the time series, information criteria such as the Akaike information criterion (AIC) and the Bayesian information criterion (BIC) could be utilized for the selection of a suitable configuration [59].

An equally important set of statistical methodologies refers to statistical regression approaches where the target variables could be expressed as a linear or nonlinear combination of features. Linear models constitute the simplest regression approaches that attempt to find the line of best fit given training data points through the derivation of optimal coefficients that describe this feature combination. The method that sets the basis for those approaches is linear regression [60]. According to this method, the predicted value $\hat{y}(w, x)$ could be expressed through $p + 1$ coefficients w_i and p number of features x_i with the formula:

$$\hat{y}(w, x) = w_0 + w_1 x_1 + \cdots + w_p x_p \tag{2.10}$$

The coefficients are calculated with the goal to minimize the residual sum of squares between the actual and predicted values, solving an ordinary least squares (OLS) task that has the following objective function:

$$\min_{w} \|Xw - y\|_2^2 \tag{2.11}$$

It is evident that feature independence is crucial for linear regression forecasting since feature dependences could result in higher error sensitivity due to the effect of multicollinearity. Several variations of the baseline linear regression model were developed in order to address a wide spectrum of challenges towards optimal forecasting. Ridge regression attempts to provide a more resilient solution towards the challenge of multicollinearity through the introduction of the penalty term $a\|w\|_2^2$ on the size of the coefficients adjusted by a positive complexity parameter a that regulates the level of shrinkage. This penalty term is added to the main objective function and reinforces the robustness of the algorithm. As a result, the value space of the coefficients is restricted and extreme values occur less frequently. Furthermore, lasso regression attempts to derive more sparse solutions through the derivation of fewer non-zero coefficients. This approach reduces the number of influential features utilized in the derivation of estimated values. A penalty factor based on a constant a denoting the degree of sparsity and the l_1-norm of the coefficient vector $\|w\|_1$ transforms the main objective function as follows:

$$\min_{w} \frac{1}{2n_{samples}} \|Xw - y\|_2^2 + a\|w\|_1 \tag{2.12}$$

Moreover, Elastic Net regression is an equally important linear regression method that shares the coefficient sparsity of lasso as well as the regularization properties of ridge regression. This method applies l_1 and l_2-norm regularization for the calculation of coefficients in order to derive a more stable model that could select multiple correlated features for the estimation of the output values. Consequently, given the degree of sparsity parameter a as well as the ratio parameter ρ that controls the convex combination of l_1 and l_2-norm regularization, the objective function is transformed as follows:

$$\min_{w} \frac{1}{2n_{samples}} \|Xw - y\|_2^2 + a\rho\|w\|_1 + \frac{a(1-\rho)}{2} \|w\|_2^2 \tag{2.13}$$

Since short-term forecasting in the energy sector often utilizes high-dimensional data, Least Angle regression could be utilized in order to provide robust estimations through the iterative refitting of residual values, producing a piecewise linear solution path. This method follows the intuitive approach of adjusting the coefficients in order to reflect correlation equality between influential factors when it occurs and performs similar to forward selection regression methods [61]. However, noisy output could occur as the residual values are fit due to carrying an error term at each iteration. It is worth noting that probabilistic methods such as Bayesian regression could contribute significantly to the development of statistical forecasting techniques since more adaptive models that consider the distribution of the data and the effect of regularization parameters could lead to robust predictions [62]. Consequently, variations of previously discussed models such as ridge regression could adopt the Bayesian framework in an attempt at improving prediction accuracy.

From the formulation of linear regression, it could be observed that the features x_i could represent independent influential factors as well as past values of the target variable. Therefore, the subcategory of regression models utilizing the shifted load or electricity price time series as input features refers to autoregressive methods. These models could set constraints on the values of coefficients in order to perform well with stationary data. Replacing the general feature notation with the given lagged time series and including a white noise term ε_t, autoregressive models of order p could be expressed as:

$$y_t = w_0 + w_1 y_{t-1} + \cdots + w_p y_{t-p} + \varepsilon_t \tag{2.14}$$

Alternatively, some regression models could utilize past forecast errors as input features in order to predict future values of energy related variables. These models are described in the literature as moving average models and the formula denoting the structure of a moving average model of order q could be easily derived from the replacement of the lagged time series features from the previous equation with error features ε_i as follows:

$$y_t = w_0 + w_1 \varepsilon_{t-1} + \cdots + w_q \varepsilon_{t-q} + \varepsilon_t \tag{2.15}$$

Depending on the processing and feature selection techniques utilized, the regression model could combine differenced past forecast features y'_{t-i} and error features ε_{t-i} in order to derive an autoregressive integrated moving average structure (ARIMA) that predicts the differenced time series for the target variable. The formulation of the ARIMA model could be derived from the separation of coefficients into the autoregressive weights φ_i and the moving average weights θ_i. Since the resulting model integrates the concepts of differencing, autoregression and moving average, the definition of the model depends on the parameters denoting the autoregressive order p, the degree of differencing d and the moving average order q. As a result, the $ARIMA(p, d, q)$ model is defined by the formula:

$$y'_t = w_0 + \varphi_1 y'_{t-1} + \cdots + \varphi_p y'_{t-p} + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t \tag{2.16}$$

Since time series data in the energy sector exhibit seasonal patterns, ARIMA models could be extended to include seasonal terms involving lagged time series for the seasonal period. The seasonal terms are typically multiplied by the non-seasonal terms. The definition of a seasonal ARIMA model extends the previous notation by adding a set of autoregression, differencing, and moving average parameters for the seasonal terms with capital letters as well as a parameter m denoting the number of observations per year. Therefore, a seasonal ARIMA model is defined as $ARIMA(p, d, q)(P, D, Q)_m$ [63].

2.4.1.2 Artificial Intelligence Models

It is evident that statistical approaches may become limited in terms of scope as the datasets become richer and the patterns within a studied environment become more complex. Additionally, the performance of more traditional estimation processes is often limited by initial assumptions and constraints on the behavior of the data. Therefore, more intuitive and flexible approaches for short-term forecasting in the energy sector could result in the thorough understanding of patterns as well as higher prediction accuracy. These models stem from the field of artificial intelligence and belong to the classes of machine learning and deep learning. Machine learning estimators introduce an algorithmic computational structure for the prediction of core energy time series and deep learning estimators expand on that structure through the integration of multiple additional computation layers that could process larger datasets more efficiently. Machine learning and deep learning estimators utilize sophisticated learning processes that focus on the efficient calibration of a model to the input and the approximation of functions that define relationships between the input features and the target variables. These types of models focus on the evolution of regression tasks and typically belong to the supervised learning subcategory since labeled datasets are processed for the estimation of load or electricity price. It is worth mentioning that the supervised learning subcategory includes some structures suitable for classification tasks such as decision trees and support vector machines that could also be adapted to regression problems.

A simple and efficient approach for regression tasks utilizes the stochastic gradient descent (SGD) algorithm that attempts to learn the linear regression function through the minimization of the regularized training error, given a loss function L, a regularization term R that functions as a penalty factor with regards to model complexity and a positive hyperparameter a that denotes the strength of regularization [64]. This is a versatile estimation method since the parameters utilized in the calculation of the regularized training error could be changed in order to express different mathematical structures. Consequently, depending on the regression task, there are different options for the loss function that are connected to other regression methodologies such as squared error for linear regression, epsilon-insensitive for support vector machines and modified Huber. Additionally, several options for the regularization parameter are available including the $l1$ and $l2$ norm as well as the elastic net regularization. The SGD regression algorithm is iterative and often utilizes an inverse scaling schedule determined by a learning rate parameter $\eta^{(t)}$ that could be calculated given the initial learning rate eta_0 and the exponent $power\_t$ as follows:

$$\eta^{(t)} = \frac{eta_0}{t^{power\_t}} \tag{2.17}$$

Alternatively, the algorithm could utilize a constant learning schedule considering only the initial learning rate or an adaptive schedule that gradually decreases the learning rate when the stopping criterion is reached, until it becomes lower than a specified threshold value.

Support vector machines (SVM) are prominent short-term load and price forecasting methods since they enable efficient high dimensional data processing and provide a flexible nonparametric structure. Models utilizing the SVM algorithm attempt to map data points to a high-dimensional space through the use of kernel functions in order to search for the hyperplanes that separate them optimally. The data points or vectors closest to those hyperplanes are known as support vectors. Therefore, the main goal of this approach is to maximize the distance between the support vectors and the hyperplane. It is evident that this objective is mostly suitable for classification tasks. However, SVM could be adapted to predict continuous output through the search for the line of best fit within the threshold set by the distance between the data boundary line and the hyperplane. The optimization problem of SVM methodologies could become simpler through the Lagrange dual formulation, providing a lower bound to the initial problem. Given N input observations x_n as well as the nonnegative Lagrange multipliers a_n and a_n^*, the function $f(x)$ utilized for predictions in the linear SVR regression approach could be defined through the formula:

$$f(x) = \sum_{n=1}^{N} (a_n - a_n^*)\,(x_n \cdot x) + b \tag{2.18}$$

The nonlinear version of the SVR regressor considers the nonlinear kernel function $G(x_n, x)$ that defines the transformation which maps observations to a high-dimensional space [65]. The prediction formula for nonlinear SVR regression is the following:

$$f(x) = \sum_{n=1}^{N} (a_n - a_n^*)\,G(x_n, x) + b \tag{2.19}$$

Moreover, an equally important set of classification methodologies adapted to regression tasks utilize decision trees as the main forecasting structure. The tree structure contains the root node that denotes the best predictor variable, decision nodes that correspond to each feature and leaf nodes. Decision nodes could contain several branches that represent the values of the feature. Leaf nodes represent the decision output. The decision tree algorithm considers the entire set of observations at the root node and attempts to split the datasets into smaller segments through a top-down greedy search approach that focuses on standard deviation reduction. The goal of this approach is to find the features that return the highest standard deviation reduction, resulting in the most homogeneous branches. The first step of this algorithm calculates the standard deviation of the target variable. Following this step, the initial dataset is split on different features and the standard deviation for each branch is calculated and subtracted from the standard deviation before the split to derive the reduction value. The feature with the highest standard deviation reduction value is chosen for the decision node and a segmentation of the dataset occurs based on the values of this feature. This process continues recursively for all branches until the leaf nodes are formed. It is evident that loss functions such as squared error and mean absolute error could be utilized for the evaluation of split quality [66]. Additionally, it is worth mentioning that the performance of this method depends on several structural parameters such as the maximum depth of the tree, the minimum number of observations for node splitting, the minimum number of observations at a leaf node and the maximum number of leaf nodes. Depending on the dataset dimensions and the complexity of feature relationships, the resulting size of the decision tree could include longer computation paths, resulting in longer total execution time. Several robust estimation approaches utilize a number of decision trees in a unified model in order to improve forecasting accuracy. One of the most prominent decision tree-based algorithms in short-term load and price forecasting is the extreme gradient boosting method (XGBoost). This algorithm utilizes several regression trees, following an iterative training process where new trees are predicting the residual errors of previous ones. The combined output of those structures forms a standalone differentiable loss function that could be minimized through the gradient descent algorithm [67].

Lastly, neural network models form one of the most flexible and robust categories of standalone artificial intelligence estimators since they follow parametric approaches in order to execute complex computations for function approximation. This class of estimators perform a series of computations on the input features in order to determine core learning parameters such as weights and biases. The computation path is typically split into several stages where the transition from the output of one stage to the input of the next one is controlled by activation functions. Neural network approaches follow several adaptive structure types and contribute in different ways when short-term forecasting tasks in the energy sector are considered.

The first type of neural network structure features computation paths that utilize sets of neurons organized in layers while forming directed acyclic graphs. Each neuron receives a set of inputs and translates them into the output through a series of computations and passing the resulting data through the activation function. These computations typically describe the output as a weighted sum of the inputs. The class of feedforward neural networks utilizes this structure in order to process linearly and non-linearly separable data and detect useful patterns for robust estimation. One of the most prominent methods in this neural network class is the multilayer perceptron (MLP). This fully connected neural network structure is a powerful estimator that utilizes an input layer, several hidden computation layers and an output layer. Multilayer perceptron could be utilized as a standalone machine learning estimator for smaller datasets when only a small number of hidden layers is selected or as a deep learning model for larger datasets when more hidden layers are included. This approach is flexible since the vectorized input of MLP does not have restrictive requirements with regards to data representation. Therefore, data representation and computation flow are problem specific and highly customizable, rendering the MLP as the template model for many forecasting tasks in the energy sector. However, the fully connected structure of MLP could result in training issues such as overfitting and the lack of explicit methods for pattern simplification could result in structures that are difficult to interpret as the complexity of the studied environment increases. Short-term time series forecasting tasks in the energy sector could study dynamic and evolving environments where a wide set of parameters may be needed for effective modeling. As a result, the global examination of the environment through the MLP structure would consider the full set of parameters, forming a shallow network that is generally difficult to interpret [68].

Convolutional neural networks (CNN) are often applied to short-term load and price forecasting in order to address some of those challenges and provide simpler pattern interpretations through a hierarchical processing approach. This category of feed forward neural networks was primarily utilized for image processing tasks, but the potential of processing data sequences as one-dimensional arrays led to the adaptation of the CNN structure for time series forecasting. The structure of CNN focuses on the local examination of data regions since each neuron is connected to an input segment. The receptive field of each neuron denotes the spatial extent of this connection that is expressed through convolution. It is evident that CNNs could be considered as the regularized version of MLPs since they introduce more compact methods towards pattern simplification that are directly connected to a hierarchical data representation. The structure of CNNs introduces several layers and concepts that contribute towards the efficient regularization and processing of time series data. First, the utilization of convolution layers as the main computational approach enables more flexible management of influential data points.

These layers could be dilated, resulting in a sparsely populated receptive field that could allow the processing of more historical time series samples. Additionally, CNN models could utilize filters in order to discard time series properties that could hinder forecasting performance such as noise and create meaningful a data mapping strategy that maintains important patterns while reducing data dimensions. Moreover, the CNN structure includes some mechanisms that could prevent overfitting such as weight decay and dropout [69].

The second type of neural network structure features computation paths that form loops in order to process data sequences based on information stored in previous states. These approaches utilize feedback mechanisms and their internal memory in order to exhibit a temporal dynamic behavior. The class of recurrent neural networks (RNN) utilizes this structure in order to make decisions based on the current input as well as information from previous training steps. One of the most prominent methods in the RNN class is the long short-term memory network (LSTM). This model extends the concepts introduced in the RNN structure in order to describe long-term time dependencies and address training challenges such as the problem of vanishing gradient that could prevent the weights of the network from changing values. LSTM architectures follow a block structure that represents the computational units used to derive each state of the network. Each block includes a set of gates that control the information that enters a computation block, denoting the data that will be omitted, the necessary data updates to the current state and the data that will be passed to the output and subsequently used as the input of the next block. This gated block structure improves upon the base RNN principles since it introduces explicit and robust ways of data control. LSTM networks add value to short-term load and forecasting since the target interval may often be influenced by time series lags that describe long-term dependencies [70].

Moreover, the wide application of the LSTM architecture in sequence prediction led to the development of several variants that highlighted different aspects of this neural network approach and reinforced the flexibility of those models in time series forecasting. The gated recurrent unit (GRU) is an alternative gated block structure that operates similarly to the LSTM. However, GRU models feature fewer gates since the processing tasks only consider the information that needs to be transferred to the next state and the information that needs to be neglected based on the importance of the previous block. Consequently, GRU models could result in faster computation when compared to LSTM due to the simpler gated structures, but they do not include an internal memory and may be less effective as longer sequences are processed. Furthermore, alternative structures could emerge from the adoption of different information flow strategies and the implementation of structural modifications that address more complex forecasting tasks. Therefore, the implementation of a forward and a backward processing layer that leads to the development of a bidirectional LSTM estimator could contribute towards the thorough understanding of complex seasonal patterns. Additionally, the inclusion of several layers of LSTM blocks results in deeper stacked LSTM estimators that could derive improved forecasting accuracy when deep learning tasks in the energy sector are considered. The LSTM and GRU architectures are suitable for the design of deep learning estimators since the hyperparameters that determine the amount of processing blocks and the types of activation functions for each gate could be configured in order to develop deeper neural network models [71].

2.4.2 Combinatorial Modeling

Standalone estimation approaches usually offer fast and satisfactorily accurate load and price predictions within short forecasting intervals, rendering them as suitable components for benchmarking and baseline formulation. However, model assumptions, structural limitations and irregular parameter behavior could impact the processing of complex energy datasets negatively, resulting in unstable, suboptimal and less interpretable predictions in several short-term forecasting scenarios. Therefore, more robust estimators need to be introduced as the evolution of the research area focuses on the development of sophisticated approaches that could result in improved accuracy, flexibility and modularity. These approaches typically combine standalone estimators as building blocks and often enhance the forecasting structure with the inclusion of traditional machine learning concepts such as fuzzy logic and more recent methodologies such as the attention mechanism while maintaining the same goal of predicting the values of the target variable in the output. Therefore, this section details the role of the most prominent categories of combinatorial modeling in the energy sector through an overview of ensemble learning methodologies, fuzzy-supported regression approaches and encoder decoder structures.

2.4.2.1 Ensemble Learning

The class of ensemble learning algorithms contributes towards the development of robust estimators that combine several standalone statistical and artificial intelligence learners in order to derive lower error metrics and improved prediction stability. The development process of an ensemble learning method includes the model selection and model integration subprocesses. In the model selection subprocess, a set of estimators is generated or selected based on arbitrary or deterministic strategies. These estimators are members of the ensemble that are utilized in order to predict the target variables from the given energy data. At this stage, an optional model elimination subprocess could be applied in order to reduce the estimator set and include the most impactful models for the forecasting task. In the model integration process, the set of base estimators follows a combination strategy in order to optimally derive improved forecasts. The set of estimator members could be diverse, utilizing different types of estimators or homogeneous, utilizing similar models or the same estimator structure. Diverse ensemble sets focus on the performance improvement achieved through the complete discovery and interpretation of patterns and relationships, while homogeneous ensemble sets focus on performance benefits derived from the correction of erroneous predictions through the utilization of input, output, parameter and induction manipulation techniques.

The most prominent ensemble methodologies utilized in short-term time series forecasting in the energy sector refer to stacking, bagging and boosting models. Stacking estimators pass the output of ensemble members to a subsequent regression model in order to learn the optimal combination of forecasts. This approach evaluates the impact of each participating estimator and derives a prediction that reflects the joint contribution of the ensemble set, resulting in robust generalization [72]. Bagging and boosting ensembles find wide application in research tasks where the standalone models such as neural networks and decision tress would yield unstable performance due to irregular events or structural intricacies in the dataset. Bagging methodologies fit ensemble members on random data subsets that encapsulate all the characteristics of the original series and subsequently aggregate the results through voting or averaging strategies. These methods focus on the reduction of variance and could be easily parallelizable for efficient computation [73]. Lastly, boosting methods build an ensemble model incrementally through the sequential training of estimator members in order to reduce the bias of the participating models [74]. As a result, the focus of boosting methods is shifted towards the iterative performance improvement in the prediction of observations that exhibited higher error in previous instances. Ensemble learning sets the foundation for meta-modeling approaches since it introduces the concept of multi-stage regression.

2.4.2.2 Fuzzy-Supported Regression

Fuzzy logic principles and, by extension, fuzzy inference systems could be integrated in short-term load and price estimation models in order to reinforce interpretability and improve training performance in terms of convergence time and accuracy. The most prominent contributions of fuzzy logic in this forecasting framework are the analysis of influential features that exhibit a degree of uncertainty in linguistic terms, the extraction of a rule base that thoroughly describes feature relationship within the studied environment and the fuzzification of model parameters. Feature fuzzification and the development of fuzzy rules offer enriched a priori knowledge to forecasting models for more informed decisions. Additionally, the fuzzification of model parameters could address the uncertainty that occurs during the training process when fewer historical observations are available or when the datasets contain missing values. Therefore, the preliminary task of feature fuzzification in the forecasting pipeline is often coupled with a rule base generator in order to form a combinatorial fuzzy regression structure.

Combinatorial fuzzy regression structures could integrate statistical as well as neural network models for the development of robust estimators. Statistical approaches such as ARIMA, typically utilize fuzzy logic principles in order to fuzzify the estimated coefficients that describe the contribution of features [75]. Neural network models often adopt several fuzzy system characteristics in order to enhance the learning process and utilize the knowledge provided by the rule base. It is evident that since neural networks often follow a black-box approach that is difficult to interpret, the fuzzification of influential features as well as the utilization of a compact set of rules before, during and after the learning process could simplify function approximation. This complementary role of fuzzy rules supports the relationships of historical observations and serves as a prototype training template. Neuro-fuzzy systems typically follow a fully connected layered structure that includes three types of layers. The first layer receives the input features, the second layer proceeds with the extraction of fuzzy rules and the third layer focuses on the representation of the estimated values for the target variables.

Two categories of fuzzy neural network structures are usually utilized for regression tasks in the energy sector. First, the cooperative fuzzy neural network includes a neural network structure and a fuzzy system that operate independently [76]. In this architecture, the neural network learns the necessary parameters from the fuzzy system through an offline or online learning process depending on the methods utilized for fuzzy set definition and fuzzy rule integration. Prominent learning processes based on fuzzy rules utilize clustering on self-organizing maps and weight strategies for rule importance. Second, hybrid neuro-fuzzy networks utilize a homogeneous structure involving a neural network and a fuzzy system where the units of the network represent the fuzzy rules and the weights in the learning process are modeled as fuzzy sets. This structure operates under the principles of fuzzy controllers since the resulting neural network is regarded as a fuzzy knowledge base. Additionally, several hybrid approaches utilize the previously described systems to form more complex forecasting pipelines where the cooperative or concurrent operation of this structure is determined based on heuristics [77]. The main point of focus in fuzzy regression modeling is the construction of an optimal rule base. Since the models are deployed for short-term load and price forecasting tasks, the rule base needs to be accurate in order to address the intricacies of the studied environment and relatively small in order to ensure faster training and model recalibration.

2.4.2.3 Encoder-Decoder Estimation

The class of recurrent neural network models offers some robust standalone LSTM and GRU estimators that predict data sequences efficiently through the processing of time dependencies. An extension of this class introduces the combinatorial encoder-decoder structure for performant time series predictions in deep learning tasks. This model consists of three main components, the encoder, the context vector and the decoder. The encoder could be represented as a recurrent neural network that receives the input sequences in order to learn a mapping that converts the sequence to the context vector. The context vector contains encoded time series information derived from the final hidden state of the encoder. This information could also represent the initial hidden state of the decoder for accurate decision making. The decoder could be represented as another recurrent neural network that derives the final forecast from the analysis of the context vector. This model could be a useful tool in the processing of series that have variable length and alignment. Consequently, the consumption features of several types of clients could be studied more efficiently as the client base evolves through time [78].

Furthermore, it is evident that the target variables of load and electricity price are often described by long input sequences containing influence factors and lagged values from several previous time steps. The processing of long input sequences could be resource intensive for most neural network architectures and could result in slow convergence time depending on the complexity of the network structure. The encoder-decoder structure addresses those issues through the integration of attention mechanisms that enable the decoder to utilize encoder information selectively, based on a weight strategy. This is achieved through the assignment of importance weights to different input sequence values and the subsequent derivation of context vectors for every time step of the decoding process. These context vectors reflect this weighted importance mechanism since they are calculated based on all the hidden states of the encoder.

There are several attention mechanisms that could be applied to encoder-decoder modeling depending on the importance evaluation strategy followed by the model. The generalized attention mechanism forms a query between each element of the input sequence, comparing it to the output. This comparison leads to the calculation of scores that reflect the relative importance of each input element and is utilized in order to derive the attention weights and scale the input values accordingly. The self-attention mechanism selects different parts of the input sequence and compares them with each other in order to modify the output sequence. Multi-head attention considers the parallel layered structure of attention heads formed by the iterative computation of attention weights. Each layer processes the input and output sequence elements and derive a combinatorial score. It is worth mentioning that powerful deep learning architectures such as the Temporal Fusion Transformer (TFT) utilize multi-head attention in a structure consisting of an LSTM encoder-decoder layer, a variable selection network and a gated residual network for robust multi-horizon forecasting of heterogeneous time series [79]. Furthermore, Bandanau attention generates a set of annotations for the input sequence at the encoder and passes them to an additive alignment model with the previous hidden decoder state for the calculation scores which are subsequently normalized into attention weights. The annotations and the weights form the context vector that could help the decision-making process at the decoder when coupled with the previous hidden decoder state. Lastly, Luong attention considers a multiplicative model where all the hidden states of the encoder are considered in order to derive the context vector [80].

2.4.3 Meta-Modeling

Meta-modeling approaches focus on the improvement of generalization, stability and accuracy through the integration of a subsequent model that processes the output of the main standalone or combinatorial structure for the derivation of the estimated target variables. These methodologies could utilize ensemble learning principles in order to combine multiple output sequences through an additional forecasting strategy or introduce feedback mechanisms aimed at prediction refinement. The main requirements in the development of meta-modeling methods involve the shift in the scope of the core forecasting structure and the extraction of additional information that could expose additional properties of the output series. Consequently, the main estimator could utilize the base feature set or different variations of the base dataset in order to derive different representations of the output. These representations could be influenced by several metrics and concepts such as similarity and causality. Alternatively, the main estimator could provide a preliminary forecast that is split into different components. The components extracted from the estimated series such as the error, often exhibit a degree of volatility that could negatively impact the performance of the model over time. This phenomenon could be easily visible in long-term forecasting horizons since the prediction error is often larger but the impact of unstable estimated components should be considered as equally important for short-term horizons since real time energy applications that present irregular changes in consumption or price could result in uncertainty, leading to poor decision making and lack of reliability. The estimated values of those components are passed to a meta-estimator in order to form a feedback mechanism that derives more stable isolated component predictions and adds them back to the original series [81].

It is evident that the research area of meta-modeling approaches is vast and evolving since most prominent estimators could be repurposed through experimentation in order to operate under different research assumptions and the same structures could be utilized in the development of the surrogate model. However, there are several challenges and restrictions associated with the formulation of efficient meta-modeling approaches in short-term forecasting. First, the overall complexity of model structure should be considered since a sophisticated meta-modeling layer that shares a similar number of parameters with the main estimator could result in slower convergence times and render model recalibration infeasible for shorter time intervals. Therefore, simpler statistical models such as autoregressive processes and fundamental neural network models such as the multilayer perceptron are usually preferred. Additionally, meta-modeling design is a balancing act that exposes the tradeoffs with regards to model selection for the main and meta-processing layer. Research in this area should evaluate the performance impact of each individual estimation layer and determine the need for a more complex architecture when it is appropriate. It is commonly observed that techniques aiming at prediction refinement utilize comparatively simpler meta-processing models since the core estimation structure needs to have the appropriate complexity in order to learn from environment dynamics. However, this standard practice may not always be effective as the dimensions of the initial dataset increase. In this scenario, there is the possibility that a robust model operating on a smaller set of estimated series derived from a simpler core model may yield satisfactory performance metrics. Second, the total number of model parameters could increase, rendering tuning and model selection strategies more computationally expensive.

2.4.4 Model Tuning

Most models utilized in short-term time series forecasting tasks introduce a set of parameters that could be adjusted in order to derive optimal performance. Several research approaches and proof of concept applications could perform a baseline model analysis utilizing parameter values that follow the default configuration provided by the application programming interface or a set of values derived from trial and error. However, when more robust combinatorial and meta-modeling approaches are considered, hyperparameter optimization ensures that the components of this structure do not exhibit a divergent behavior on the given dataset.

Prominent hyperparameter optimization methods form search strategies that examine the parameter space and algorithmically denote the best candidate solutions based on a set of rules. Grid Search is one of the most common optimization strategies that exhaustively searches value subsets based on the performance metrics provided by time series cross validation or holdout validation methods. Since this approach tests all the combinations within those subsets, there could be performance issues associated with the high dimensionality of the search space. Alternatively, Random Search could be utilized in order to examine a random sample of parameter values derived from specified distributions [82]. This search strategy could outperform Grid Search since it operates on the reduced dimensions of the sampled values and could be easily parallelized. Furthermore, the global search algorithm of Simulated Annealing could be utilized as an iterative stochastic approach for the selection of optimal parameter values since it introduces customizable criteria and functions that could control the convergence process of machine learning estimators [83]. Equivalently, Bayesian optimization could be considered as another impactful global search algorithm that iteratively evaluates different model configurations based on the probabilistic mapping that determines the transition from one candidate configuration to the next, resulting in fewer evaluations when compared to other search methods [84]. Moreover, evolutionary algorithms contribute towards the global optimization of neural network models through fitness ranking and the iterative replacement of suboptimal hyperparameters generated through the genetic operators denoting crossover and mutation [85]. Lastly, gradient-based optimization methods could be utilized for the selection of optimal parameters in neural network structures through the application of the gradient descent algorithm and the definition of a hypernetwork that generates weights for the main neural network estimator and learns the configuration that yield optimal output [86].

2.5 Output and Performance Evaluation

The output of the forecasting structure contains estimated values for the target energy variables examined in short-term prediction horizons. The evaluation of those values is typically executed in three stages. At the first stage input training samples are utilized in order to derive the output at each instance of the learning process based on known data. At the second stage, the framework derives the estimated output from the input values of the validation set for the purposes of parameter refinement as an internal procedure. Lastly, at the third stage, the input values of the test set derive the test output samples for generalization to unknown data and performance evaluation. The main strategies applied to those three stages may differ due to the roles of the input datasets in the forecasting framework for training and validation as well as the output module for testing. However, a shared set of performance metrics is usually utilized in order to denote the divergence of the estimated from the actual values. Therefore, in this section the prominent methodologies involved in the evaluation of the output for all stages of the learning process are examined as well as the associated performance metrics utilized in this research area. Furthermore, the processes involved in the training of the models are examined and significant risks such as overfitting, underfitting and the concept of drift are outlined.

2.5.1 Performance Metrics

Functions that express the comparison between the actual time series values to the estimated output could be considered useful tools towards the monitoring of the training process, parameter optimization, result interpretation and performance quantification of short-term forecasting models in the energy sector. These functions are mainly statistical measures that describe the magnitude of error, presented as a numerical value or as a percentage. Error direction, scale dependence and interpretability are some of the most crucial factors that influence the categorization and selection of those performance metrics. Research efforts utilizing energy data for short-term load and price predictions typically attempt to minimize those metrics at the forecasting framework and the output module since lower error metric values denote more accurate forecasts. Additionally, these metrics are widely used for model comparisons given a specific input dataset. It is worth noting that since the quality of the data influences those metrics, the comparison of models needs to be conducted on the same input in order to ensure fairness. The most prominent error functions utilized in this research space are the mean absolute error (MAE), mean absolute percentage error (MAPE), mean squared error (MSE), and root mean squared error (RMSE).

Mean absolute error provides an easily interpretable and natural error metric that is indifferent to the direction of errors [87]. This metric is commonly used as a loss function for the training of machine learning models and as a simple performance evaluation indicator for the output. The numerical values of this metric follow the original units of the estimated variables. This function does not interpret the impact of the relative error size as there is a difficulty in the differentiation of error magnitude. Since optimal forecasts require the minimization of error, a value of 0 MAE denotes that the forecast is perfect and exhibits no error. Given the predicted values y_i and real values x_i in a set of n samples, the mean absolute error is computed by the formula:

$$MAE = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} \tag{2.20}$$

Furthermore, mean absolute percentage error [88] is mainly used as a scale independent performance metric since it expresses the percentage of the average of absolute differences between estimated and actual values. This metric could provide a generalized percentage score for forecasting models. Given the same parameters for the calculation of MAE, MAPE is computed by the formula:

$$MAPE = \frac{100}{n} \sum_{i=1}^{n} \left| \frac{y_i - x_i}{x_i} \right| \tag{2.21}$$

Similar to MAE, this error metric does not emphasize on the impact of large errors exhibited due to value spikes. Additionally, since the denominator in this formula contains the actual time series value, this metric is suitable for datasets where the values of the target variables are nonzero. It is easily observed that extreme actual values may impact the consistency of this metric.

Moreover, the error metrics of mean squared error [89] and root mean squared error [90] provide quadratic loss functions that measure the forecasting uncertainty while focusing on the impact of large errors. The values of MSE could express the sum of the variance and square value of bias, further contributing to the performance analysis of a model while penalizing large errors more than small errors. Additionally, the values of RMSE increase with the variance of the frequency distribution of error magnitudes, resulting in larger values when large error values are present. Similar to MAE, RMSE values could be easily interpretable since they share the same unit as the estimated variables. Furthermore, the simultaneous inspection of MAE and RMSE could provide a thorough examination of error variance. When there is a great difference between the values of MAE and RMSE, variance in the magnitude of errors could be detected, denoting the occurrence of large errors in the forecast. Given the same parameters used for the computation of the previously described error functions, the formulae for MSE and RMSE are the following:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - x_i)^2 \tag{2.22}$$

$$RMSE = \sqrt{\sum_{i=1}^{n}\frac{(y_i - x_i)^2}{n}} \tag{2.23}$$

For the purposes of this dissertation, the previously discussed performance metrics will be utilized in the training and evaluation of the strategies proposed towards the improvement of the short-term forecasting pipeline in the energy sector given specific scenarios where regression is applicable. As a result, training, validation and test loss will be calculated through those metrics for the three output stages. However, it is worth noting that while these are the most prominent metrics, weighted variations of those functions could be utilized in order to cover the edge case where significantly smaller datasets are selected. Additionally, problem specific metrics could be defined in order to enhance the knowledge extracted from the data based on the associated field, such as econometric functions and directional performance metrics for energy price forecasts.

2.5.2 Output Evaluation Stages

2.5.2.1 Model Training

The training process at the forecasting framework could be considered as the first stage in the forecasting pipeline where estimated output samples are generated based on known data points. These observations are derived iteratively based on training algorithms such as gradient descent for simple linear regression or backpropagation for neural network structures. Since these processes are iterative, at every iteration the loss function expressed as one of the previously discussed performance metrics is calculated. A naïve approach considers a large number of iterations for the training process with the expectation that the loss function will converge to a small value close to zero. However, there are several challenges associated with the training process that could have a considerable impact on future output evaluation stages. First, the resulting model after the completion of the training process may not be able to capture the patterns in the data adequately due to small training dataset size, poor data quality or low model complexity. In this case, the model is underfitting and could be characterized by high bias and low variance. In order to mitigate this challenge, an increase in the number of training iterations, an increase in the number of features or an increase of the overall model complexity could be applied. Second, the resulting model could overfit as it may learn details and patterns associated to noise that could limit its generalization capabilities. This behavior often occurs due to high model complexity, an increased number of training iterations or due to the small size of the training dataset. In this scenario, high variance and low bias are exhibited. The challenge of overfitting could be mitigated through the reduction of model complexity, the increase in training observations and the application of regularization strategies [91]. Additionally, early stopping mechanisms could be considered in order to track the progress of the loss functions and stop the training process as the error starts to increase [92].

2.5.2.2 Model Validation

The calculation of performance metrics with regards to the estimated output generated from the validation set could be considered as the second stage of output evaluation in the forecasting pipeline. Validation loss plays a significant role in hyperparameter optimization since there are several strategies that influence model selection based on the repeated calculation of this metric. The most prominent approaches in this research area include holdout validation and time series cross-validation. Holdout validation is a simple approach primarily used for the development of baseline models for performance comparisons or for the evaluation of models that utilize large datasets [93]. This method is often utilized in order to derive an unknown data segment that could be considered as the validation and the test set simultaneously, resulting in the same value for validation and test loss. Alternatively, this segment could be separate from the test set and contribute towards model selection through the minimization of validation loss. Time series cross validation considers the temporal data structure and the autocorrelation of observations by iteratively splitting the dataset into n segments, where the first $n-1$ segments belong to the training set and the last, most recent set, in terms of temporal order, belongs to the validation set [94].

The combined inspection of validation and training loss could provide insight towards the detection of irregular training behaviors such as overfitting and underfitting. When the value of the validation loss increases and greatly surpasses the training loss, model overfitting could be assumed. Additionally, underfitting could be observed when the values of validation and training loss remain high, and their respective curves denote irregular peaks and valleys. An optimal fit could be observed when validation and training loss remain low, sharing similar values. In this scenario, validation loss could be slightly higher than the training loss as the number of training epochs increase as the updates to model weights become less significant. Alternatively, validation loss could be slightly lower than the training loss when regularization and dropout mechanisms are integrated. The same effect could be observed when the training process considers a small number of epochs and the training algorithms perform significant updates to model weights during those first steps. When validation loss is drastically lower than the training loss or fluctuates above and below the training curve, representativeness issues in the validation set could be detected [95].

2.5.2.3 Output Module

The final stage of output and performance evaluation is performed at the output module of the forecasting pipeline where the test set is examined and test loss is calculated through the previously discussed performance metrics. At this stage, test loss provides an indicator for the generalization capabilities of the model on unknown data. Additionally, if a meta-modelling strategy is applied for prediction refinement, the final output of the feedback mechanism that improves the error is evaluated and examined in this module. Data visualization techniques that plot the actual and predicted data points within specific time intervals complement the error metrics and reinforce interpretability since the errors could be directly connected to the observed differences between the two curves.

Test loss is expected to share a similar error profile to validation loss since both validation and test sets refer to observations of future values collected after the timestamps of the training set. Since short-term forecasting in the energy sector processes diverse time series that may correspond to different types of clients and buildings or may express different price fluctuations in studies that examine combined energy market dynamics, the test loss could increase as the model receives new data samples for prediction, leading to unstable and suboptimal performance. This phenomenon occurs due to the challenges of data and concept drift. These challenges denote the gradual performance degradation of forecasting models as the distributions of the input datasets and the statistical properties of the target variables change over time. It is evident that issues related to drift could be easily detected when new samples are added for the recalibration of the model since the evolution of the historical dataset could be continuously monitored through the calculation of error metrics [96]. The detection of those challenges becomes more difficult in more complex combinatorial models when several different datasets need to be processed by a central estimation structure. In both scenarios, these issues could be mitigated through lagged drift detection. The most prominent methodologies for lagged drift detection utilize hypothesis testing, distribution comparisons, machine learning models and sliding window approaches. Nonparametric tests such as the Kolmogorov-Smirnov test compare the cumulative distributions of datasets and evaluate the null hypothesis denoting that the distributions are the same. Furthermore, the population stability index (PSI) could be used for distribution comparison as a single value indicator that denotes a small drift when PSI is lower than 0.1, a moderate drift when PSI values are between 0.1 and 0.25 and significant drift when PSI values are greater than 0.25. Alternatively, the repeated training of machine learning models for the prediction of the target variable could monitor data drift through accuracy fluctuations in the output. Adaptive windowing (ADWIN) could detect concept drift through the definition of a fixed-size sliding window. This algorithm computes statistically significant values for the time series such as the mean in the regions of the sliding window and compares their difference to a specified threshold value in order to examine potential changes in the statistical properties of the series. Lastly, drift detection methods could calculate statistical metrics as new data is made available to the model in order to provide a real-time drift monitoring strategy. The Page-Hinkley method follows this approach by calculating the mean of the observed values and comparing them to a threshold [97].

Chapter 3 Integration of Fuzzy Logic and Hybrid Feature Importance on the Preprocessing Module

3.1 Motivation

The design and development of the preprocessing module in short-term forecasting tasks involves the implementation of methods that efficiently process and transform the input dataset in order to derive the most significant features in an interpretable form. Research in this area contributed to the introduction of methods that reduce the dimensions of the initial dataset, focusing on the selection of the most influential factors. Additionally, feature representation methods such as feature fuzzification and rule base construction were outlined for the reinforcement of the decision-making capabilities in the forecasting framework since the extraction of a set of rules could enable more intuitive time series forecasting strategies. However, several challenges and research gaps were detected in the development of feature selection strategies and the construction of performant rule bases. First, it could be observed that while several hybrid feature selection approaches were developed for the reduction of energy datasets, the space that defines combinatorial feature selectors remains insufficiently explored since more novel strategies that perform the cross-examination of feature importance metrics from classification and regression algorithms could be proposed. Consequently, the introduction of more hybrid feature selection strategies could reinforce the flexibility of the preprocessing module as the examination of more combinations contributes towards the design and application of preprocessing tools that could be readily available for further research and real-world forecasting applications. Second, fuzzy rule bases may provide enhanced interpretability and decision-making to the forecasting models but this is done at the cost of scalability, since the dimensionality of the fuzzified features could render the extraction of rules infeasible for short-term tasks that examine complex environments. Short-term horizons set strict training and recalibration time requirements and the increased complexity of the environment often results in an extended space of fuzzified influential factors where feature processing becomes computationally expensive. Naive approaches neglect the challenge of dimensionality and attempt to include all possible rule combinations generated from the fuzzified features. More robust approaches propose the simplification of the rule base through expert knowledge, model assumptions and rule filtering, implying that the initial set of rules will be generated and subsequent methods will be applied for rule elimination or reevaluation. Therefore, recent research efforts do not sufficiently cover the need for a deterministic approach that focuses on the most important influential factors within the scope of a short-term forecasting task in the energy sector in order to build a small and interpretable rule base that maintains the accuracy of the expanded space.

These observations could be combined in order to highlight the need for substantial contributions in the design of neurofuzzy systems and forecasting models that utilize a fuzzy controller for information extraction through a rule base in order to further understand environment dynamics. It is evident that fuzzification expands the feature set, introducing dimensionality issues in complex environments, while robust feature selectors shrink the feature set, mitigating performance hinderances and preserving the most important features. Therefore, strategies that evaluate the importance of the fuzzy antecedents with regards to the target variables could assist in the derivation of compact rule sets, providing feasible solutions for energy applications in short-term horizons. In this chapter, the design strategy for the development of a fuzzy controller with regards to the electricity consumption of a residential building is presented. This research project focuses on the improvement of the rule generation process through the integration of a hybrid feature selector, resulting in a smaller set of rules that accurately define the studied environment. The following sections correspond to the introduction, methodology, results and discussion of this published work.

3.2 Introduction to Fuzzy Control System for Smart Energy Management in Residential Buildings Based on Environmental Data

Modern energy applications often use load profiles resulting from time-series data of electricity usage to monitor and manage the power consumption of customers efficiently and reliably [98]. In an attempt to maintain the balance between power supply and demand, energy consumption patterns are further processed and as a result, a plethora of models aiming at the adjustment of customer behavior are developed. The insights extracted from the energy data convey more interpretable trends and patterns, which can be used by the energy provider as a management tool for the control of distribution and pricing. Additionally, the output of such models can be useful to customers as a recommendation engine, helping them make more informed decisions and reschedule their daily tasks when opportunities arise for them to participate in more dynamic pricing plans [99]. In the past, simpler prediction and recommendation models were linear and faced many challenges such as data dimensionality, trend detection, and uncertainty. Since the study of residential and industrial environments requires a more detailed definition of all the variables that contribute toward energy consumption, the energy datasets used in modern applications often contain many important measurements ranging from appliance consumption values to weather parameters. Consequently, the dimensions of the inputs and outputs grow, and this could hinder the computational performance of more traditional models, rendering the resulting energy applications less efficient [100]. Furthermore, linear models sometimes fail to capture the trends that can be observed from the data, and the mathematical models used to describe them do not express the dynamic and complex nature of those environments as they evolve over time. Hence, linear forecasting models and decision-making applications yield less accurate and suboptimal results, respectively [101]. Moreover, many input parameters used to define those environments often have a more imprecise and uncertain meaning that is often associated with human perception and expert knowledge. Therefore, it is difficult to fit those crisp values in a strict mathematical model without further interpretation and achieve decent performance [102].

Solutions to some of the challenges mentioned above can be found in the fields of machine learning and fuzzy logic. Traditional machine learning methods, such as decision tree classification, are capable of achieving highly accurate and interpretable results, while more advanced techniques such as artificial neural networks preserve the complex and dynamic nature of those environments and assist in constructing more adaptive models with impressive performance. Fuzzy logic methods tackle the challenges of uncertainty and partial truth in decision-making systems, since the environments are processed in a more interpretable way with the introduction of linguistic terms that express the vagueness of human perception for input and output parameters. Since fuzzy systems are defined by sets of rules that are close to real world expert rules, decision-making models based on fuzzy logic are popular due to their computational efficiency and overall simplicity [103]. The main practical advantages of using fuzzy theory can be observed from several successful Enterprise Resource Planning (ERP) and power system control applications. Fuzzy logic can handle the ambiguities and vagueness of qualitative factors covered by ERP software [104]. Additionally, the stability problems of multi-area interconnected power systems caused by nonlinearities can be resolved through fuzzy logic approaches by approximating nonlinear models into linear sub-models [105]. Hybrid techniques utilizing concepts from both fields such as fuzzy neural networks are proven valuable in the development of robust energy applications due to their adaptability and their black-box behavior [106].

However, it is worth mentioning that there are still questions, challenges, and research gaps that arise with the evolution of those fields. Firstly, the challenge of dimensionality is a recurring threat to the performance and interpretability of those applications and design philosophies around feature engineering should be applied in order to isolate the features that are more relevant and important in a particular environment. In general, modern energy applications based on those models need to yield results within specific time intervals with the upper limit being the time that new data would normally be measured by smart meters in order to be considered relevant and acceptable. Therefore, systems using highly dimensional input data could yield slower performance outside of the acceptable time intervals. Secondly, there is a level of ambiguity that surrounds the design process of each energy application, which is mostly related to the available knowledge and information about the environment as well as the intended behavior of the finalized model. For example, residential environments could be clustered together, and available expert knowledge could extract a more generalized set of rules that is applicable to that group but on an individual basis, expert knowledge could not always be readily available, and the historical data as well as the behavior of each occupant could be more important in the extraction of meaningful rules. Additionally, fuzzy logic models and machine learning models often need to be retrained to reflect major changes in some vital parameters such as occupancy and number of appliances. Since the environments evolve over time, respective models need to adapt to the new data easily, because decisions and recommendations based on outdated rules could hinder customer satisfaction.

A thorough examination of the literature shows that there exists relevant research work highlighting aspects of fuzzy logic and machine learning in the development of systems that offer optimizations, management solutions, and forecasting potential in the energy sector. In 2008, Azadeh et al. [107] presented a framework that combines fuzzy logic and a data mining approach in order to predict electricity demand. In their work, they briefly outline different methods of rule extraction from decision trees and offer other meaningful comparisons of their work with modern machine learning methods such as artificial neural networks. The same year, Lau et al. [108] presented a case study of a fuzzy logic forecasting system in a clothing manufacturing plant, drawing optimal strategies for efficient energy consumption forecasts in that environment. In 2015, Suganthi et al. [109] published a useful review of fuzzy logic applications in renewable energy systems and concluded that these models provide realistic estimates. In 2017, Emagbetere et al. [110] developed a fuzzy prediction system for power consumption forecasts following the Mamdani approach. Their system utilized a small set of predefined rules, and their work offered a concise error comparison between different membership functions. Javaid et al. [111] used Mamdani and Sugeno fuzzy systems in order to evaluate their adaptive thermostat. In their work, the simplicity and flexibility of fuzzy inference systems is highlighted. Zhang et al. [112] presented a fuzzy forecasting method utilizing historical data found in time series through link prediction. Furthermore, Bissey et al. [113] developed a fuzzy logic method for the optimization of electricity consumption in an individual residential environment, thus allowing for the better management of appliances and for the flexibility to reshape the load profile should that be desirable. This work is particularly important for our project, since it shares a similar scope. In 2018, Krishna et al. [114] proposed a smart home energy management system based on fuzzy logic with a hardware implementation that renders it ready for installation and deployment. The impact of fuzzy reasoning on energy applications developed for residential environments can be clearly seen in the work of Nebot and Mugica [115], published in 2020, where a side-by-side comparison of two fuzzy logic methodologies shows the importance of feature selection and correct identification of the most relevant building parameters.

Machine learning and fuzzy logic methods are strongly interrelated, and relevant research on the field reinforces the notion that one approach can benefit from the integration of the other. Sophisticated machine learning methods such as neural networks follow a data-driven modeling approach that utilizes a numerical representation in order to prepare the data for relationship induction and model inference. Since relationships between data points are often presented as complex computational graphs, the interpretability and flexibility of those models is poor due to the lack of a human–machine interface. Therefore, it is easy to understand that machine learning succeeds in the statistical induction of models from observations and data, but there are considerable difficulties when attempting to derive conclusions from premises, models, and assumptions. Fuzzy logic extends existing machine learning models through concepts, tools, and techniques that introduce knowledge-based design elements and a symbolic representation of data that is more interpretable. As a result, the logical deduction of conclusions is a significant contribution of fuzzy logic to machine learning methods. Additionally, fuzzy systems can be significantly improved with the integration of data-driven approaches. The development and implementation of machine learning methods in state-of-the-art fuzzy systems could address the potential sparsity of expert knowledge. Furthermore, the insights and data processing techniques used in machine learning models could lead to the generation of smaller and more accurate sets of rules while enabling future changes as the data evolves without the continuous supervision of an expert [116].

In this study, we focus on fuzzy control systems for individual residential environments without the contribution of expert knowledge. We believe that many interesting design approaches can be discussed in an attempt to tackle the challenges mentioned in order to develop intelligent systems that merge aspects of fuzzy logic and machine learning effectively. The main purpose of this work is to present the design and implementation process of a fuzzy energy system for an individual residential environment; the system discovers and generates rules based on a decision tree model that integrates a hybrid feature selection method for the choice of the most important linguistic variables. The proposed system should be viewed as a contribution to the development of intelligent decision-making, recommendation, and management tools in the energy sector, since the expected output denotes the optimal energy consumption value based on environmental parameters such as weather data. This system could be integrated into client-side applications in order to derive recommendations that could help reschedule the daily tasks of consumers and minimize energy consumption within short intervals. Additionally, electricity providers could utilize this system as a secondary management and control tool for regulation and electricity pricing in more customizable and dynamic models that apply to individual customers indirectly. Classification methods and load profile monitoring could be powerful tools that contribute toward the creation of electricity plans, but the realization that these plans are usually formed from generalized consumer patterns greatly reinforces the need of having localized models that could help the adjustment of those existing plans at a greater detail in an attempt to increase customer satisfaction and plan flexibility. To the best of our knowledge, the combination of machine learning methods and feature engineering techniques explored in this paper has not been discussed before in the context of individual energy consumption recommendations without the availability of expert knowledge. Therefore, we believe that our project presents a novel and intuitive fuzzy system structure that addresses the challenges and the complexity of the residential environment while maintaining simplicity. Section 3.3 presents a concise overview of the design process used in the development of a fuzzy control system, and the core structure is expanded by outlining the components of the proposed model. Section 3.4 presents the results by providing a sample response of the fuzzy system and listing the most important improvements when compared to a simpler variant that does not utilize a hybrid feature selector. Finally, Section 3.5 offers a discussion of the results obtained from the design and implementation process and identifies directions for the utilization of the system and future work.

3.3 Materials and Methods

3.3.1 Fuzzy Control System Design

3.3.1.1 Core Structure

According to the Mamdani inference method [117] and fuzzy logic principles [118], the fuzzy control system includes several components that form a pipeline that is used to derive crisp output values from a given set of crisp inputs. Uncertainty and imprecision are present and often impact on the decision-making process considerably, since people use non-numerical information to evaluate and interpret real world scenarios. To understand the entire design process, we explain each component of our proposed model in turn and present the resulting algorithm of the base Mamdani system.

In the first step of the fuzzy control system design process, the input and output variables are selected, and fuzzy sets need to be constructed. Intuitively, fuzzy sets are regions of data points that, to some degree, belong to a certain linguistic interpretation of a variable given a range of values. For example, if we selected the temperature of a room as our input variable and decided to recognize the linguistic terms “cold”, “warm”, and “hot”, a trapezoid-shaped curve could be defined to describe the fuzzy set that corresponds to the linguistic term “warm”. Hence, there is the need to map each crisp input value to the fuzzy sets and receive the corresponding degrees of membership. Continuing the example above, a specific room temperature value could yield the set of membership degrees [0.8, 0.2, 0] denoting the real world equivalent of asking 100 people about their perception of the room temperature and 80% of them responding with “cold” while 20% would respond with warm. This assignment of values to membership degrees is achieved through the membership function defined for each linguistic term, and this process is executed by the fuzzification module of the control system. The number and types of the various membership functions used in the system structure are chosen by the designer based on experimentation, expert knowledge, or clustering. It is important to note that fuzzy systems that are designed to manage complex environments focus on having a low execution time, and consequently, the choice of three or five membership functions for a given variable is very common [119].

The second component of fuzzy control systems is the decision-making unit, which uses a set of fuzzy rules in order to map the input truth values to the desired output truth values. Fuzzy rules are IF–THEN statements between antecedents and are consequently expressed in linguistic terms. These rules utilize fuzzy operators [120] and are evaluated in parallel using fuzzy reasoning. The evaluation of each fuzzy rule entails the assignment of rule weights denoting their importance and the application of an implication method such as the minimum and product, which scale the output fuzzy set accordingly. The number of rules for a particular system heavily relies on the selection methods used, the intended usage of the fuzzy system, and the complexity of the environment. Since the rules constitute the basis for pattern identification, the number of rules should cover every possible result in the output. Fuzzy systems designed to produce predictions often use a larger set of rules to maintain high accuracy, whereas systems that focus on the regulation of a specific behavior or the extraction of recommendations and insights focus on the most important subset of rules that will be applicable in each case. Furthermore, rules can be manually constructed or generated based on the availability of expert knowledge, the variable dimensions, and the dependencies within a system. Simpler systems that remain static and explore a smaller input–output space usually work well with rules created by the designers in cooperation with experts on the field. On the other hand, dynamic systems that change and evolve over time as well as systems that handle highly dimensional datasets use rule discovery and generation techniques. Modern fuzzy systems use a variety of methods from the fields of artificial intelligence and machine learning such as grid partitioning, genetic algorithms, decision trees, and fuzzy neural networks in order to generate interpretable sets of rules [121–124].

The third and final component of fuzzy control systems using the Mamdani approach is the defuzzification unit, where the results of the rules are combined and distilled. The aggregate output fuzzy set of the rule evaluation step is now mapped back to a crisp set. There are a wide variety of methods used in the defuzzification process, which can be organized in distinct groups based on their properties. Maxima methods such as the mean of maxima are often used in fuzzy reasoning systems in order to calculate the most plausible result, whereas distribution methods and area methods such as the center of gravity are increasingly popular in fuzzy controllers due to the property of continuity [125]. The simulation and calculation of the crisp output using those methods is made easy due to various programming interfaces and libraries in Matlab (R2020b, The Mathworks, Natick, MA, USA) and Scikit-Fuzzy that carry out these operations efficiently. Figure 3.1 presents the core structure of a fuzzy system that contains the components analyzed above and serves as the basis upon which we shall expand for our proposed model.

These components form the standard Mamdani fuzzy system, which will be structurally modified to address the challenges of the use case examined in this work. The algorithm of the standard Mamdani system used to compute the crisp output y from the crisp numerical input $X = x$ given a rule base of statements in the form of "IF X is A_k THEN Y is B_k" where A_k and B_k are fuzzy sets appearing in the antecedent and consequent respectively that consist of four steps. In the first step, the degree of membership of input x in the fuzzy set A is computed as $\mu_{A_k}(x)$ and the corresponding rules with positive degrees of membership are activated. In the second step, the fuzzy set in the consequent of each rule is truncated at the level of the previously calculated degree of membership, forming the output fuzzy set $\mu_{output\ k|x}$, which follows the equation:

$$\mu_{output\ k|x}(y) = \min\left(\mu_{B_k}(y), \mu_{A_k}(x)\right) \tag{3.1}$$

In the third step of the algorithm, all the truncated fuzzy sets are aggregated to provide a single set $\mu_{Mamdani|x}$, which can be defined by the membership function:

$$\mu_{Mamdani|x}(y) = \max_k \left[\min\left(\mu_{B_k}(y), \mu_{A_k}(x)\right)\right] \tag{3.2}$$

Lastly, the crisp output is calculated from the defuzzification of the fuzzy set using the horizontal axis projection of the center of gravity of the region under the membership function $\mu_{Mamdani|x}$ in the final step.

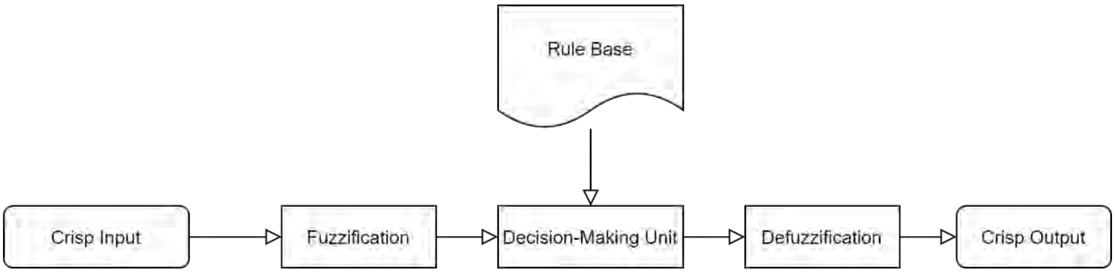


Figure 3.1: Base Fuzzy System

3.3.1.2 Proposed Model

Following the base fuzzy system design of the previous subsection, the design of our system, which features a decision-making unit that is enhanced by machine learning methods, is presented. Since the target environments of our system lie within the energy sector, and specifically the automatic regulation and management of electricity consumption at an individual level, certain aspects of the decision-making process need to be explored further in order to suggest fast and easily interpretable solutions. Energy data and environmental parameters such as weather variables form time series with complex patterns that create complex datasets that cannot be easily expressed by expert rules. It is easy to see that different consumers living in separate buildings have different needs and therefore generate different load profiles based on their individual schedules and their perception of the environment. Moreover, for the construction of the optimal recommended consumption response to a set of weather parameters, rules need to be discovered by a method that could easily be retrained on new datasets when drastic changes occur in the load profiles due to schedule or major appliance changes. The increased complexity and dynamic nature of these environments often result in larger sets of rules due to the high number of input features. Consequently, one of the main appeals of fuzzy logic methods, namely computational efficiency, could be hindered if no extra processing is performed on the input features.

In order to tackle the challenges mentioned above, we divided the decision-making unit into a feature engineering and a rule generation process, which proceed to organize rules and feed them to the inference engine of the Scikit–Fuzzy application programming interface (API) for evaluation. The feature engineering process focuses on reducing the number of distinct inputs while maintaining the most important linguistic terms associated with each input variable. One-hot encoding [126] is used in order to denote the presence or absence of a specific linguistic term based on the most dominant fuzzy labels produced by the membership function evaluation. The resulting state-based features are ranked based on their importance in a hybrid feature selection system including XGBoost (1.2.1, The XGBoost Contributors, Seattle, WA, USA) and decision tree metrics. The linguistic terms with scores above certain thresholds are appended to a list and passed down to the rule generation process as inputs. In this process, a decision tree classifier is constructed, and each branch of the resulting tree is linearized recursively into a relatively small set of IF–THEN rules. The crisp output is derived after the rule evaluation and defuzzification of results following the Mamdani approach. In Figure 3.2, we present a diagram of our proposed model outlining each step used to construct the rule base, and in Figure 3.3, we include a diagram of the main use cases that could take advantage of this fuzzy system as it was discussed in a previous section. In the following subsections, we apply this model design on a real-world energy dataset of a building and analyze each step in more detail while explaining all the decisions formed in order to handle that data efficiently. For the following case study, Pandas 0.25.3 and Numpy 1.17.3 were used for data manipulation, Matplotlib was used for visualization, and XGBoost 1.2.1 and Scikit-learn 0.24 were used for the rule generation and the hybrid feature selection. Scikit-Fuzzy 0.4.2 was used for the construction of the fuzzy system. The project was written in Python 3.7.5, and the simulation was executed on a desktop computer with an AMD Ryzen 1700X processor, 8 gigabytes of RAM, and an Nvidia 1080Ti graphics processor. The code of this project is available on Github [127].

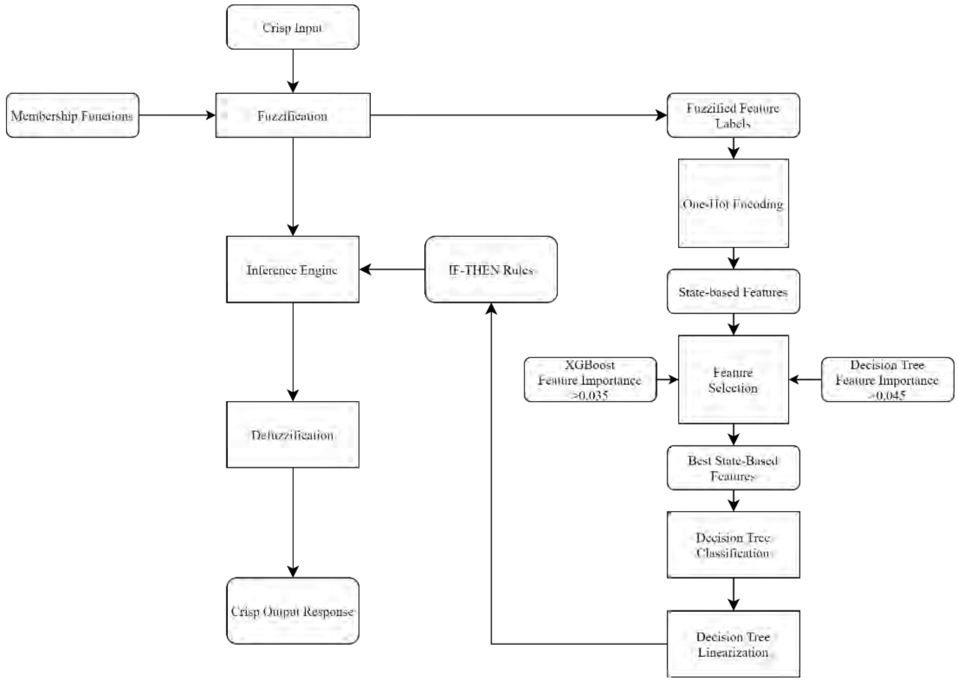


Figure 3.2: Fuzzy system design for optimal consumption recommendations based on the load profile and weather data.

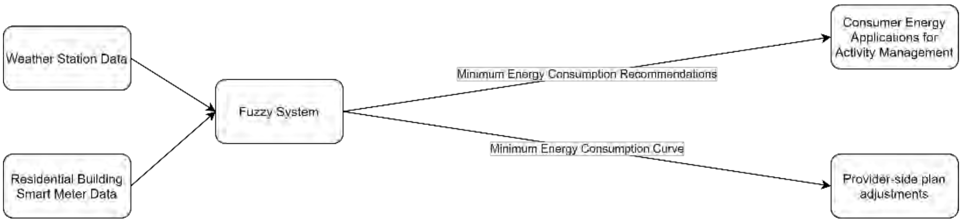

Weather Station Data
Residential Building Smart Meter Data
Fuzzy System
Minimum Energy Consumption Recommendations
Minimum Energy Consumption Curve
Consumer Energy Applications for Activity Management
Provider-side plan adjustments

Figure 3.3: Potential integration of the fuzzy system as a minimum energy consumption recommendation tool for consumer applications or as a secondary analysis tool for provider-side adjustments complementing the load curve.

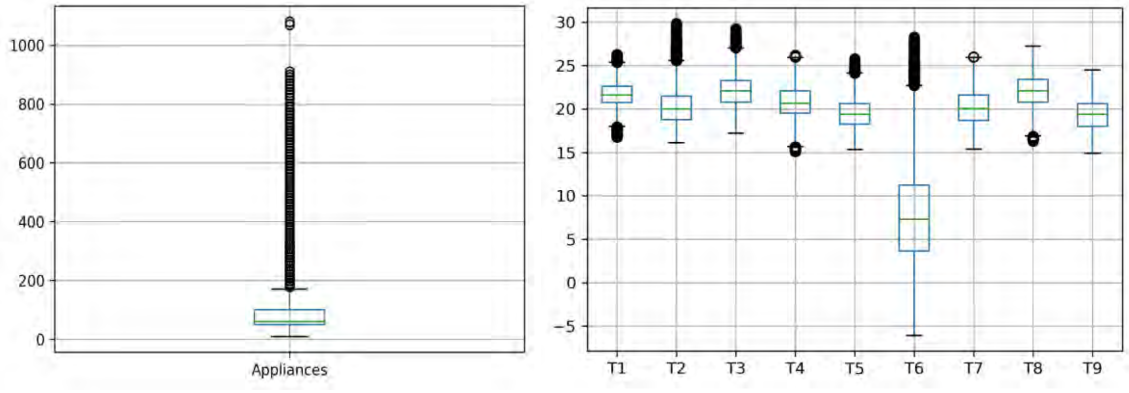
3.3.2 Dataset Overview

In order to construct a complete simulation of the proposed system using Scikit-Fuzzy, we utilized the energy data found in [128]. This dataset contains a time series of energy consumption and weather data of a low-energy house designed according to the passive house certification [129] in Stambruges with a total floor area of 280 m^2 and a total heated area of 220 m^2. The house has four occupants: two teenagers and two adults. The data variables collected in this dataset consist of the appliance and lighting energy consumption, temperature and humidity values of nine different areas inside and outside the residence, wind speed, pressure, visibility, two random variables introduced in the original paper for the study of regression tasks as well as date and time-related features such as number of seconds from midnight, week status, day of the week, and a date timestamp. The energy consumption values as well as the values for humidity and temperature were recorded by sensors at ten-minute intervals and transmitted via Xbee radio. Weather parameters such as wind speed, pressure, and visibility were collected from the weather station in Chièvres at an hourly sampling rate and were interpolated to produce 10-min measurements. The dataset contains records of a 137-day time span and further exploratory analysis of trends, feature correlation, and importance were carried out in the original paper.

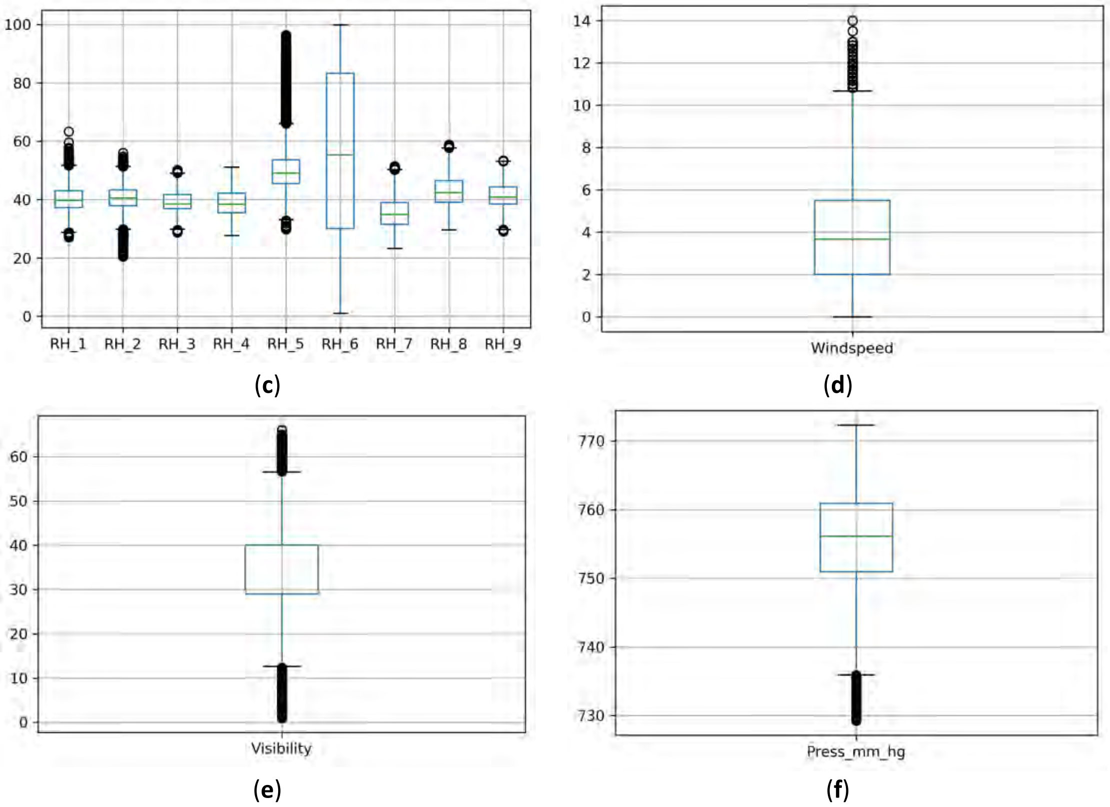
For the purpose of our project, we selected the appliance energy consumption as the output variable, since the desired behavior of our fuzzy system was the generation of optimal energy consumption recommendations for the occupants based on environmental parameters. As for input, we selected the local temperature and humidity measurements for the nine areas as well as the weather variables of wind speed, visibility, and pressure, since the perception of each feature could vary between occupants, therefore making such features suitable for fuzzification. Since the input consists of a total of 21 columns, we can already observe that in the ensuing step of fuzzification, the feature space expands, and refinements are needed in order to deal with its size efficiently.

3.3.3 Fuzzification

In this subsection, we analyze the fuzzification process in which the crisp values of input and output variables are converted into fuzzy sets. In order to achieve that, we generate box plots, as presented in Figure 3.4, and further inspect the exploratory data analysis of the original paper. As a result, we infer the ranges and the universe of discourse for each variable, and we are able to define sets of linguistic terms as well as membership functions. In order to maintain the computational simplicity and interpretability of the system, we select to assign 3 linguistic terms and the associated membership functions for pressure, visibility, wind speed, and humidity while appliance consumption and area temperature are assigned 4 and 5 linguistic terms, respectively. A range of 3 to 5 terms and functions is very common in the literature and could adequately capture the human perception of a fuzzy variable. Furthermore, common membership function shapes are selected such as the triangular, trapezoidal, and sigmoid through the generators of Scikit-Fuzzy in order to contribute to the overall simplicity of the system. In Table 3.1, we list the linguistic terms assigned to each variable, and in Figure 3.5, we present the graphs of the associated membership functions. Since the human perception of temperature and humidity in any given area is universal and the different upper and lower bounds for each area individually would not alter the human decision in the characterization of those parameters, all nine temperature and humidity features share the same membership functions for temperature and humidity, respectively. However, the temperature and humidity of each area is defined as a different fuzzy input variable on the system in order to match the complexity of the environment we study. Intuitively, a human would make nine different decisions for each area of the building and aggregate those in order to make a deduction. It is worth noting that since the ranges for each variable are derived from dataset analysis, the input and output of our system can easily be parameterized to fit the load profiles of other buildings given a history dataset. Finally, the degrees of membership for each crisp record are calculated with the interp_membership method of Scikit-Fuzzy, forming fuzzy sets for each input and output value.



(**a**) (**b**)



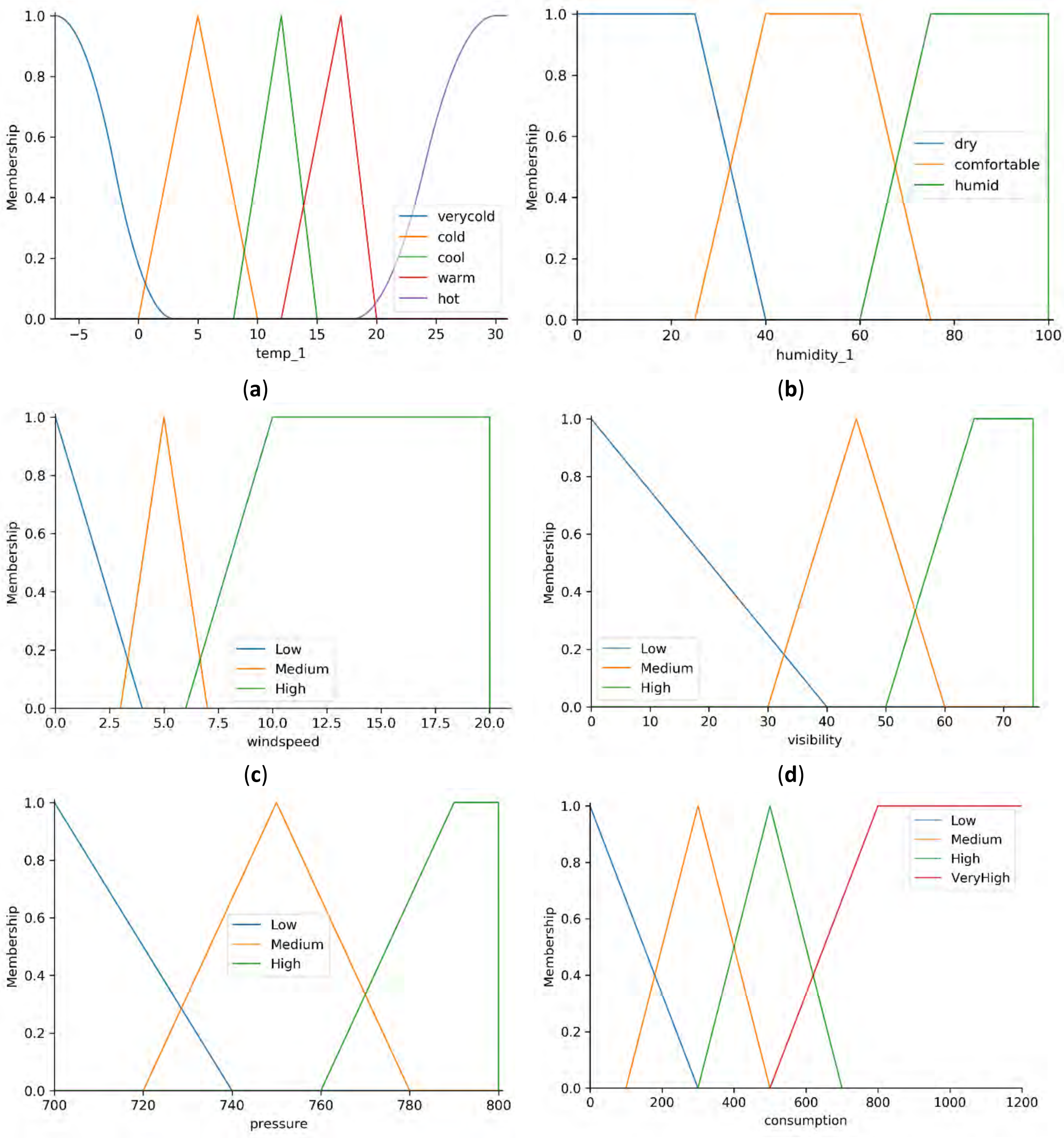
(**c**) (**d**)

(**e**) (**f**)

Figure 3.4: Box plots of dataset features showing the ranges of values for each variable in order to define membership functions. The features used in this fuzzy system case study are: (**a**) Appliance Consumption in Wh; (**b**) Temperature of nine rooms in °C; (**c**) Humidity percentage of nine rooms; (**d**) Wind speed in m/s; (**e**) Visibility in km; (**f**) Pressure in mm Hg.

Table 3.1: Linguistic terms for input and output fuzzy variables.

Variable	**Linguistic Terms**
Temperature	Very Cold, Cold, Cool, Warm, Hot
Humidity	Dry, Comfortable, Humid
Wind Speed	Low, Medium, High
Visibility	Low, Medium, High
Pressure	Low, Medium, High
Consumption	Low, Medium, High, Very High



(**a**) (**b**) (**c**) (**d**) (**e**) (**f**)

Figure 3.5: Membership functions for each input and output variable with different color coding for each linguistic term: (**a**) Temperature; (**b**) Humidity; (**c**) Wind speed; (**d**) Visibility; (**e**) Pressure; (**f**) Appliance Consumption.

3.3.4 Decision-Making Unit

In this subsection, we follow the results of the fuzzification process and analyze the feature engineering and rule generation processes needed to construct the decision-making unit for our fuzzy system. Since the environment we study is based on a historical energy dataset of a building and there are many different parameters involved in the induction of the recommended appliance consumption values, we need to be able to extract rules that are general enough to address the most dominant states of each parameter and at the same time specific enough to include the most important states of each parameter that contribute the most to the construction of a rule. Furthermore, as the environment changes and evolves, we need to ensure that an easily interpretable model is in place that can be conveniently retrained to reflect the updated set of rules in case there are major changes in the occupancy, the appliance setup, and the general operation of the building.

The fuzzification process yielded membership scores for a crisp value on the corresponding set of linguistic terms. In order to derive the most dominant linguistic term that will be useful for rule extraction, we select the maximum membership score for each crisp value and construct a new dataset that consists of the dominant label for each input and output variable. For example, if the value for visibility yielded the highest membership value for the linguistic term “Medium”, we set that as the dominant state of that record on the new dataset. Additionally, due to its simplicity, versatility, and interpretability, we selected the decision tree classifier as our base model for rule extraction. Since the new dataset of dominant terms contains categorical input and output variables, we apply one-hot encoding on the input and use the output terms as classes in order to enable the decision tree to process the data effectively. Therefore, the original entry of the above example is replaced with the appropriate three columns for low, medium, and high visibility while having the value 1 for medium visibility and 0 for all other terms. This data transformation introduces the challenge of dimensionality, since the combined total of 22 input and output feature columns is now increased to 85. One-hot encoding contributes to the desired behavior of the model, because all possible decision paths are represented in the branches of the decision tree. However, a large amount of decision paths could lead to a substantially large set of rules that not only hinders the interpretability of the decision-making model but also the computational performance of the fuzzy system.

In order to tackle the challenge mentioned above, we shift the focus to the pursuit of the most important terms that influence appliance consumption through the process of feature engineering. Since we now have state-based features for each variable, we no longer need to ask the question, “Does the temperature in the kitchen area have a significant impact on appliance energy consumption?” but rather ask, “How important is the state of feeling hot in the kitchen area for appliance energy consumption?” The difference between the above questions reflects the quality difference between feature engineering approaches in a fuzzified input space. Choosing to answer the second question is equivalent to examining the possible antecedents of a rule one by one without significant information loss. On the other hand, the first question could eliminate the entire feature of temperature, thus rendering the rules more general and sometimes less applicable to input sequences where an antecedent related to temperature would activate a specific rule for computation.

Therefore, for our fuzzy system, we select to apply a hybrid feature selector, inspired by the feature selection method proposed in [130] and based on the feature importance values derived from an XGBoost classifier and a decision tree classification model on their default configuration. The one-hot encoded dataset was split into a training and validation set with 70% of the data allocated to the former and 30% of the data reserved for the latter. These models were constructed with the expectation of retraining the decision-making unit in the future; thus, choosing the simple hold-out validation would be less computationally expensive than the other methods. The importance scores are extracted using the built-in methods of the Scikit-learn and XGBoost packages, and they are presented in Figure 3.6.

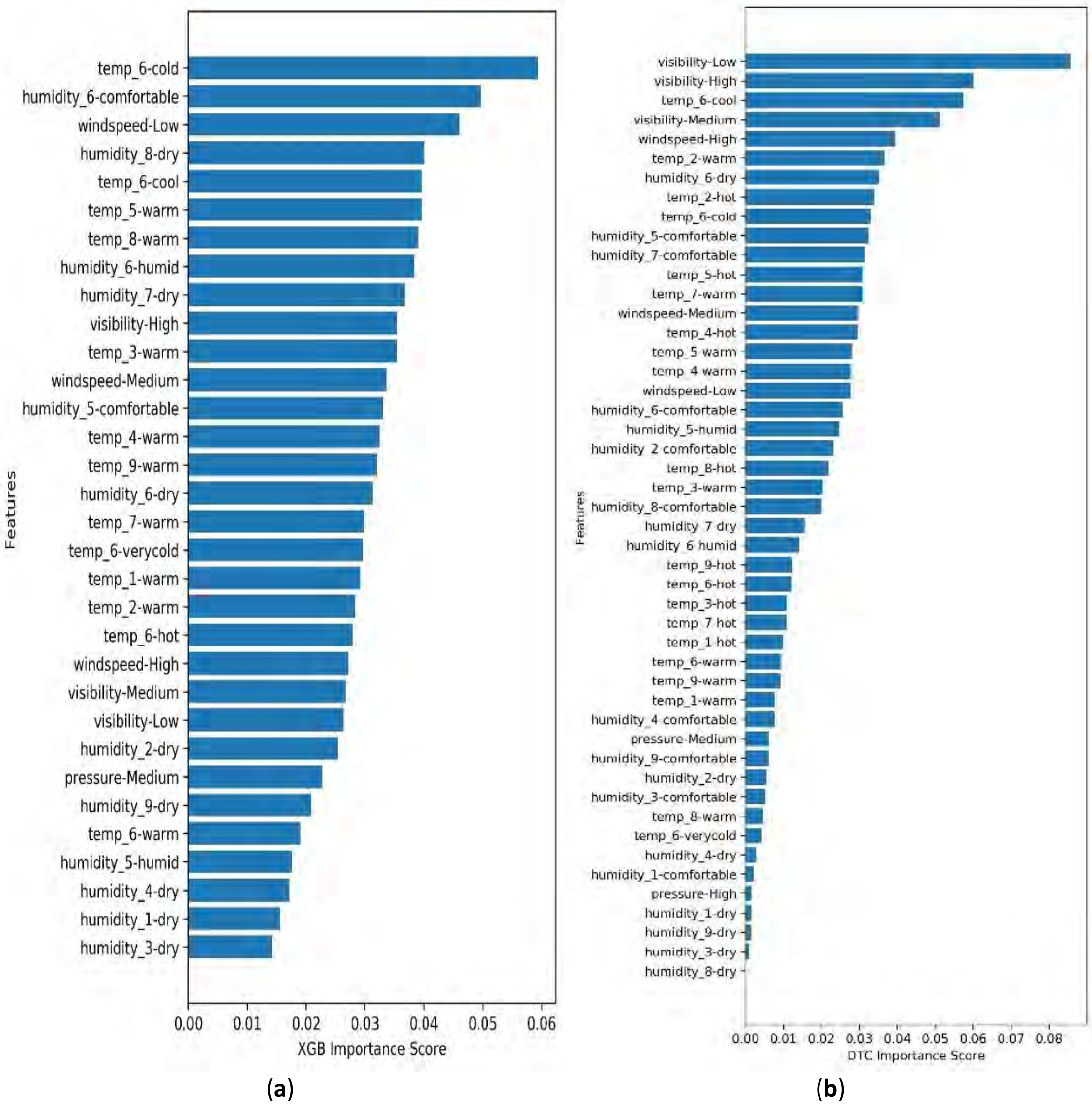


Figure 3.6: Importance scores of state-based features. (**a**) XGBoost feature importance, showing the importance of a feature on the performance of a trained model using this algorithm; (**b**) Decision Tree Classifier feature importance, showing the normalized total reduction of the criterion brought by that feature. More important features receive higher scores.

It can be easily observed that since each feature was split into several linguistic terms, the individual importance score of each term as a rule antecedent yields relatively low values in both cases. The feature selector uses a threshold for each classifier to append the most important state-based features into a list followed by duplicate elimination. The following formulas clarify the process of appending a feature to the list:

$$f(s) = \begin{cases} appendif, & IXG[f] \geq t1 \\ dropif, & IXG[f] < t1 \end{cases} \tag{3.3}$$

$$g(s) = \begin{cases} appendif, & IDT[g] \geq t2 \\ dropif, & IDT[g] < t2 \end{cases} \tag{3.4}$$

Symbols f and g denote the candidate feature groups to be appended to the list, and the variables $IXG[i]$ and $IDT[i]$ refer to the feature importance values derived from the XGBoost and Decision Tree classifiers, respectively. The symbols $t1$ and $t2$ represent the selection threshold of each method and are set to 0.035 and 0.045, respectively. Each threshold was selected after the inspection of each individual feature score. The values represent the middle points of each scale, shifted by 0.005 considering the rounded maximum and minimum importance of the variables.

The resulting list of features is used as the input of a new decision tree classifier, where the output classes are the linguistic labels that characterize consumption. Decision trees are suitable for rule extraction, since they can be linearized to if–then statements [131]. Consequently, we inspect every path of the decision tree recursively and parse the corresponding rule based on the features appearing in that path. Each non-leaf tree node contains a state-based feature, which is selected as an antecedent for the rule. If the feature follows the left branch of a decision path, it is used with the negation operator, since the value for that term is 0. Alternatively, if the feature follows the right branch, it is included in the antecedent as is. The antecedents in each rule are connected in logical conjunction. Leaf nodes denote the consequents of each rule, since they are the linguistic terms that characterize appliance consumption. The rules are written in a text file, which is then processed and parsed to generate an executable Python code that can be used by the Scikit-Fuzzy API to perform simulations of the fuzzy logic system.

3.4 Results

In this section, we demonstrate the output response of the fuzzy logic system after the simulation of an input sample. We outline the performance and interpretability features of the system by highlighting the effectiveness of the changes made in the decision-making unit. Such changes affect the way input and input is handled during rule generation and shorten the response time of the computation process. Since the crisp output results from the aggregation of rules that get activated (that is, their antecedents are satisfied), using a reduced rule base consisting of the most important antecedents leads to fewer antecedent checks. Moreover, due to the nature of some defuzzification methods and the disjunctive effect of multiple rules as seen in [117], the output may not satisfy the consequent of any rule to any extent, because it would be the result of a compromise between different extreme regions on the consequent. Using a decision tree structure combined with a feature refinement technique should decrease the likelihood of those compromises, since the resulting branches are expected to be smaller and distinctly different on the variables that represent the antecedents. Therefore, rules that could point to different extreme regions, causing a compromise in the aggregation, are expected to include a higher number of different antecedents that need to be activated. The effect could still be present, but this expectation sets the requirement of having different antecedents and contributes to the interpretability of the system.

For our example, we assign as input values the crisp values of the first dataset record. Since this system does not predict energy consumption but is aimed at giving advice on the desired minimum consumption based on past operation, the selection of dataset records for demonstration purposes is a fast and convenient way of providing a realistic set of input values. Arbitrary input values for each dataset feature could still yield a response from the system, but the process of determining the probability of their occurrence for this building would be time consuming and lies outside the scope of this work. The fuzzy system was initialized with the integration of 281 rules derived by our decision-making unit. Since the record passed in the input may contain data fields that are not present as antecedents in that set of rules, we implemented conditions to check for their occurrence in the rule base and exclude the columns when those antecedents are not present. In this example, we observed that 11 out of the 21 input variables were not present in the final set of rules, hence excluding five temperature values, five humidity values, and the pressure value. After 1.27 s of computation time, the system yielded a response of 209.89 Wh for appliance consumption, which can be interpreted as the optimally typical consumption value based on the given environmental data and the history of operation of the building. In Figure 3.7, we present the resulting area that is used to calculate the crisp output value based on the

Mamdani approach. Additionally, in Figure 3.8, we present the response of the fuzzy system for 500 10-min intervals, denoting the minimum energy consumption for 500 dataset records. While the inspection of an individual data point in Figure 3.7 provides significant details on the two fuzzy sets involved in the computation as well as their membership, the simulation in Figure 3.8 shows that the minimum energy consumption of the building could be characterized as mostly "Medium" for those timesteps. However, the crisp output values vary, showing the potential influence of fuzzy sets related to different linguistic terms. For example, for timesteps where the minimum energy consumption is below 140 Wh, we can assume that there could be a significant past contribution of several instances where "Low" consumption could occur given the environmental data in the input.

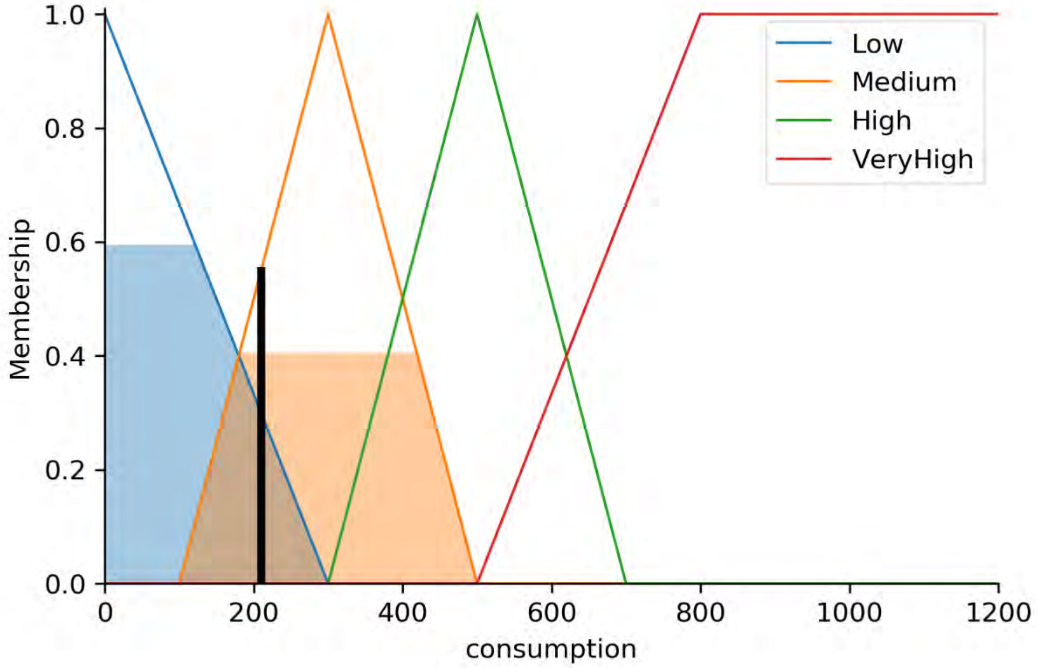


Figure 3.7: Sample response of the fuzzy control system for the computation of energy consumption. The resulting fuzzy sets are derived from the highlighted blue and orange surface areas and the bold black line denotes the optimally typical energy consumption value after the defuzzification process.

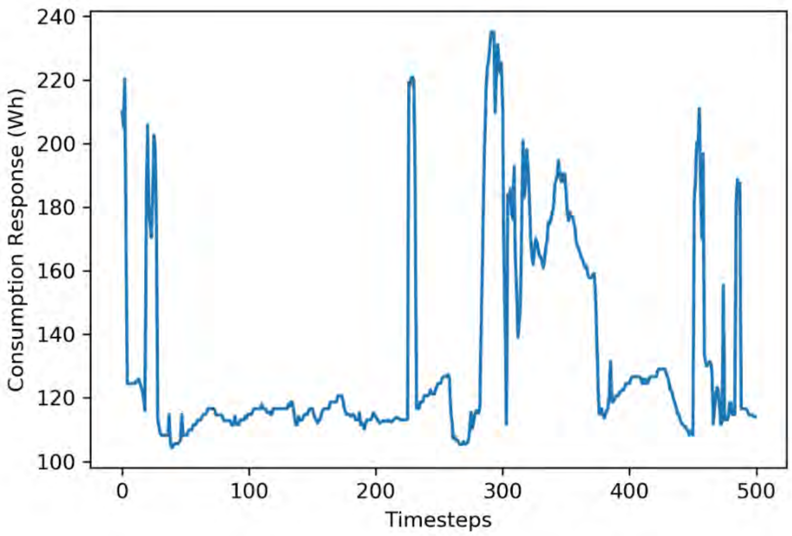


Figure 3.8: Minimum energy consumption response values for 500 timesteps. Each timestep is a 10-min interval corresponding to a dataset measurement.

This control system features several improvements over the base ID3 model for this environment due to the implementation of the decision-making unit. Table 3.2 presents the accuracy scores and the number of resulting rules after the linearization. Through our experiments, we observed that the feature engineering process contributed to a slightly higher classification accuracy while considerably decreasing the number of input features and the number of the resulting set of rules. Consequently, the fuzzy system was capable of computing crisp values fast, despite the complexity and initial number of the linguistic variables. Moreover, it is important to mention that since the time interval of the measurements in the dataset is 10 min, we set that time as the upper limit for a fuzzy system response; this should be the maximum amount of time so that the computed optimal typical value would be the most valuable for applications. The base decision-making unit produced a significantly larger set of rules, and the fuzzy system did not yield a response during that time.

Table 3.2: Linguistic terms for input and output fuzzy variables.

Model	Features	Accuracy (%)	Rules
Base Decision Tree	85	88.8	802
Refined Decision Tree	14	89.2	281

3.5 Discussion

This research work explored a fuzzy system design approach for a residential building based on weather parameters in order to derive recommendations for the minimum energy consumption values based on environmental data. Since the rules of the system are unknown and the nonlinearity of the recorded time series data increases the overall complexity of the environment, a machine learning model was constructed and the decision-making unit of the fuzzy system was modified in an effort to generate accurate rules based on the past operation of the building. Compared to the more traditional decision tree rule generation model, our structure managed to shrink the set of rules by 65% while achieving slightly better classification accuracy. Dimensionality proved to be another challenge for this system, since a total of 85 features would result in a large decision tree that would be hard to interpret, and the generated set of rules would slow down computation time. Therefore, the decision to implement a hybrid feature selector in an attempt to find the most important linguistic terms led to a significant structural optimization [132], since the remaining set of features was 84% smaller than the initial one, and crisp input values were essentially filtered against the rule base to eliminate redundant features—i.e., features that do not contribute to the conditions of any rule. Consequently, the computational performance is acceptable, since the response of the system is within the time interval of recording an energy consumption measurement through smart meters. The base linearized decision tree structure featuring all available variables resulted in a larger and less accurate set of rules. Therefore, there was no output for the base system within the 10-min intervals. For the purposes of this work, we are satisfied with an acceptable computation time within the measurement interval because the fuzzy rationale is not constantly exact, and the output of fuzzy systems may not be generally acknowledged [133]. Shifting the focus toward faster computation times could be detrimental to the stability of the system due to refinements that could be more impactful than feature importance, resulting in an insufficient amount of rule checks. Thus, we focused on the structure and the quality of the features in order to ensure proper knowledge representation.

Additionally, the decision-making module could be easily retrained to accommodate future changes in occupancy and appliance operation. The resulting energy consumption values represent the optimal consumption under the specified weather conditions and could be used by applications in order to inform the consumers, encouraging them to maintain or change their consumption habits, thus introducing fewer irregular patterns in their load profiles. Alternatively, the response of this fuzzy system could be utilized in demand response applications on the provider side in order to drive indirect adjustments to consumer behavior through varying pricing schemes. Since we believe that a direct adjustment targeting the load profile curve could lead to consumer dissatisfaction, an indirect adjustment based on the recommended consumption could provide an incentive to consumers to manage and plan their activities voluntarily. The integration of the proposed structure in consumer or provider applications could be overall user-friendly, since environmental measurements and smart metering information could be provided automatically, without the contribution of an expert for the extraction of knowledge in a particular residential building. Moreover, depending on the parameterization used in the configuration of membership functions for each use case, this system could be suitable for any residential building. Since we use fuzzy logic to map input and output to linguistic terms through an application programming interface, it could be convenient for developers to use those linguistic terms as an additional tag when referring to the output response, thus characterizing the minimum energy consumption in a more interpretable way.

However, it is worth noting that maintaining the transparency of the system and the simplicity in our approach could be regarded as an adaptability and performance hurdle under specific circumstances. The decision tree structure used in the rule generation process can be sensitive to changes in the data. Since the input and output are tied to linguistic terms, there is a level of protection tied to the range of values that corresponds to the same linguistic term but more extreme data variations that could result from significant changes in the appliances, the activities of the occupants, or extreme weather conditions; the model may need to be trained again to reflect the changes on the rules appropriately. Fortunately, in the localized residential environment, retraining the model would not be detrimental to the real-world performance of the system considering measurements recorded at 10-min intervals, but we can expect that a rule generation module based on a neural network and evolutionary algorithms would be more efficient under those extreme conditions while sacrificing interpretability.

In the future, comparisons between this decision-making model and other modern rule generation approaches such as fuzzy neural networks and genetic algorithms would be beneficial to the overall exploration of interpretable and computationally efficient solutions for similar datasets under the same assumptions. Additionally, the integration of similar fuzzy system designs featuring comparable feature engineering approaches would be an interesting area to explore, as automation solutions and demand response applications evolve with the help of machine learning. Finally, the extension of the existing system with the inclusion of a feedback module capable of regulating the desired behavior of the residential buildings based on specific thresholds set by the electricity providers would enhance the proposed structure.

Chapter 4 Neural Network Modeling Towards Granular Standalone Load Estimation

4.1 Motivation

Standalone estimation models add significant value to short-term forecasting in the energy sector since they provide performant solutions through a single centralized structure for the prediction of load and electricity price. A plethora of methods ranging from traditional statistical models to more robust machine learning approaches could be applied to those tasks, enabling the development of flexible demand response applications and energy management systems. The complexity of those models could be easily determined through the analysis of relatively small sets of hyperparameters and satisfactory predictions could be derived within short time intervals. Therefore, the performance evaluation of those models could provide insight towards the individual behavior of estimators in short-term horizons and the greater role of each structure as a building block in combinatorial methods.

Recent research efforts focus on machine learning estimators and the class of neural network models since these structures could adapt to more complex time series patterns through more robust and highly customizable training processes. Additionally, since these methods typically operate as black box approaches for function approximation, the resulting models are difficult to interpret and a more thorough examination of the output is often required. Lastly, it is evident that most standalone statistical models follow a simpler structure based on dataset assumptions. As a result, generalization issues could occur and the standalone statistical models may not be able to offer scalable solutions as the dimensions of the studied dataset increase and the time series include more diverse patterns. Therefore, this observation renders the study of standalone neural network structures increasingly interesting as energy research and applications focus more on the design of flexible real-time automation systems.

Standalone neural network structures such as MLP, CNN and LSTM offer a wide range of powerful tools for short-term load and price estimation since complex patterns could be identified within a small amount of training epochs. These models are widely used in the design of robust forecasting frameworks coupled with novel preprocessing or optimization techniques and often could be combined to form more sophisticated models. However, the selection process for the utilization of a specific neural network structure is not always clear since the baseline performance evaluation of those structure does not sufficiently cover all short-term and very short-term forecasting scenarios. Consequently, there is confusion surrounding the integration of those methods that typically leads to arbitrary decisions and extensive trial and error experimentation. As the forecasting horizon becomes shorter, it is observed that fewer studies consider the performance comparison of those models and tasks involving highly granular load and price measurements are insufficiently covered.

In response to those observations and the overall uncertainty surrounding high resolution energy forecasting through neural network design, our contribution focuses on the comprehensive comparison of baseline neural network performance for the forecasting task of minutely active power predictions. This study highlights the performance of MLP, CNN and LSTM variants in their default configuration through the simple and interpretable metric of MAE and examines the training behavior of those structures through graphs that monitor the training process. Additionally, the average training time per epoch was measured for each baseline configuration in order to denote the most efficient architecture in terms of convergence.

This contribution covers several perspectives of the minutely sampled point forecasting tasks, resulting in the thorough understanding of this edge case for future research. First, the examination of error metrics allows researchers to observe the performance characteristics of the studied structures as the deployment of those powerful architectures is expected to yield low error values in high resolution point forecasting processes. Second, the examination of the loss curves provides useful information with regards to the training process in this edge case. It is evident that high resolution data could present patterns at a higher level of detail and neural network structures could learn those patterns easily through a smaller number of iterations when compared to other forecasting horizons. Since impactful changes to weights and parameter values typically occur during the early training stages it would be interesting to monitor the effect of those changes in a use case where the first training iterations could represent the majority of training steps. Third, the evaluation of training time in this research scope is a crucial factor towards model selection since higher training time could render the deployment and recalibration of a model in this forecasting horizon infeasible. Alternatively, these measurements could also highlight the need for more computing power, denoting that some neural network structures could be more computationally expensive. As a result, future research works focusing on energy applications could consider those three perspectives and based on the results of this study, they could establish a base for more informed model selection, speeding up experimental throughput in short-term and very short-term tasks. The following sections correspond to the introduction, methodology, results and discussion of this published work.

4.2 Introduction to Minutely Active Power Forecasting Models using Neural Networks

The evolution of the smart grid and smart metering technology has enabled electricity providers to develop more sophisticated Demand Response (DR) programs in order to influence the consumption patterns of their customers by adjusting pricing signals. In the modern grid, Demand Response programs exploit the dependencies of the information streams that flow between customers and suppliers. Customers allow for their load profiles to be created and scrutinized, by providing smart meter data that reflect their consumption patterns; the data are derived simply from the daily operation of their devices. Suppliers are then able to interpret that data, and after identifying the demand trends, they can reflect them on supply expectations via price signal alterations that, in turn, can shift or change consumption patterns. In this way, electricity demand may be handled in a dynamic environment. In search of greater Demand Response flexibility and optimization as well as better third-party support through automation there is a lot of ongoing research in the field that is focused on the development of more precise load forecasting techniques, in order to obtain even more dynamic price signal adjustments. Hence, there is a considerable contribution from the areas of artificial intelligence and machine learning to the energy sector by way of various models and techniques aimed at managing and predicting real-time price and load fluctuations [134].

Since the data extracted from smart meters is in the form of time series, many statistical methods and classical machine learning models have relatively difficult implementations due to the temporal difference of the data points and the limitations concerning missing values, data dependencies, and dimensionality. The problem of missing values refers to the complete absence of some samples or the existence of non-interpretable data entries in a dataset [135]. Missing values introduce a level of uncertainty and bias which degrades the performance of classical models. Therefore, the reasons behind the existence of missing data need to be identified and imputation techniques in the preprocessing of the data have to be considered in order to create more robust classical models. On the other side of the spectrum, neural network models often omit missing values without a significant loss of quality in the results. Furthermore, data dependencies refer to the hidden relationships and patterns, such as trends, which could provide useful insights about the time series [136]. Traditional autoregressive integrated moving average (ARIMA) models are based on linear relationships and not on the joint distribution of random variables. Hence, nonlinear trends are not fully explored. Additionally, the limitation of dimensionality refers to the ability of the model to process a large number of input variables derived from different time series efficiently, while yielding meaningful results [100]. Traditional models focus primarily on univariate input data, considerably limiting the potential insights derived from richer time series datasets [137]. Neural networks are more suitable for handling complex relationships within the data and for developing robust forecasting models that are tolerant to noise: long short-term memory (LSTM) networks [138] are capable of identifying the long-term dependencies between data points and convolutional neural networks (CNNs) [139] can extract features from the raw input sequence and encode them in a low-dimensional space. The multi-layer perceptron (MLP) [140] can model non-linear trends and is able to handle missing values in the datasets well.

In 2015, Alamaniotis and Tsoukalas [141] presented a data-driven method for minutely active power forecasting based on Gaussian processes. This research project highlighted the importance of minute predictions in the residential setting due to the volatile nature of household consumption and examined machine learning models that outperformed the more traditional autoregressive moving average approach. In 2017, Singh et al. [142] trained an artificial neural network comprising 20 neurons in order to conduct short-term load forecasting of the NEPOOL region of ISO New England and yielded a decent Mean Absolute Percentage Error (M.A.P.E) performance while training on weekday data points. In 2018, Kuo and Huang [143] proposed the Deep Energy neural network structure, which consisted of an input layer, a feature extraction module, and a forecasting module. The tuning of the parameters in the convolution layers of the feature extraction module and the data flattening layer in the forecasting module resulted in relatively high precision short-term load predictions. Hossen et al. [144] examined deep neural network architectures in order to accurately forecast residential load consumption for a single user with one-minute resolution based on one year of historical datasets. Zhang et al. [145] reviewed machine learning methods in smart grids and outlined state-of-the-art approaches in the field of load forecasting. Kampelis et al. [146] used the genetic algorithms and neural networks to evaluate day-ahead load shifting techniques. Koponen et al. [147] presented physical- and data-driven models for Demand Response. Their work presented a very useful comparison of a support vector machine and a multi-layer perceptron for power forecasting. In a more recent work, Ahmad et al. [148] proposed a modular neural network model for load forecasting which consisted of a pre-processing module for the input time series, a forecast module where the artificial neural networks were trained, and an optimization module which helped minimize the forecast error. Walther et al. [149] utilized machine learning processing techniques such as feature engineering and hyperparameter tuning in order to optimize a Gradient Boosting Regression Trees (GBRT) algorithm which performs very short-term load forecasts with a 15-minute horizon based on minutely sampled data. Zhu et al. [150] presented a comparative study of deep learning techniques using minute-level real-world data of a plug-in electric vehicle charging station in order to evaluate the performance of those approaches on a variety of timesteps. The results of this study are valuable to machine learning researchers in the energy sector due to the examination of many different configurations in the deep learning space. Gasparin et al. [151] assessed the performance of deep recurrent neural networks on minutely sampled datasets of individual household electric power consumption in order to pave the way for standardized evaluation of the most optimal forecasting solutions in the field. Susan Li [152], in an article about time series prediction using LSTM, highlighted the minutely sampled data of a residential dataset provided by the University of California at Irvine (UCI). Cheekoty [153] presented the main advantages of neural network techniques over classical machine learning in time series forecasting, and, finally, Orac [154] constructed an LSTM model in order to predict trading data.

In this study, we focus on the minutely active power forecasting for residential electricity consumption, since we believe that, despite their overall complexity, accurate high granularity models can lead to fine-grained price signal adjustments. In the development of those models we use the types of neural networks mentioned above on the individual household electric power consumption dataset found in the UCI machine learning repository [155]. The main purpose of this work is to compare the baseline performance of each network on the same dataset and provide useful remarks on the training process of each model. There is little work in the area of minute power forecasting and our study is the first concise comparison of the core neural network types on this prediction horizon with experiments conducted on residential active power data. In Section 4.3, we explain the methodology and the concepts that were followed to conduct the experiments. In Section 4.4 we present the results of our experiments through evaluation metrics relevant to the training process and the prediction quality of each network. Finally, in Section 4.5 we discuss the results obtained and suggest some directions for future work.

4.3 Materials and Methods

4.3.1 Neural Networks and Performance Metrics

In this subsection it is important to provide a concise introduction to the neural networks and the performance metrics we used for our experiments in order to outline their primary behavior prior to presenting the configurations of our machine learning models.

4.3.1.1 Multi-Layer Perceptron

The multi-layer perceptron extends the perceptron learning algorithm [156] and uses neurons arranged in layers in order to form a feedforward artificial neural network that approximates a function. This type of neural network uses a non-linear transformation on the input, which is learnt through the adjustment of weights and biases in the intermediate layers of the network. For simplicity, we considered a multi-layer perceptron with one hidden layer. This MLP approximated a function $f: R^D \rightarrow R^L$, where D is the size of the input vector x and L is the size of the output vector $f(x)$. Following the matrix notation, the MLP, which consisted of an input layer, a hidden layer, and an output layer, can be expressed by the following formula:

$$f(x) = G\left(b^{(2)} + W^{(2)}\left(s\left(b^{(1)} + W^{(1)}x\right)\right)\right) \tag{4.1}$$

In this formula, $b^{(1)}$ and $W^{(1)}$ are the bias vector and weight matrix from the input vector to the hidden layer, $b^{(2)}$ and $W^{(2)}$ constitute the bias vector and weight matrix from the hidden layer to the output. Activation functions s and G define the output of the hidden layer and the output layer, respectively, given a set of inputs. The MLP was trained through backpropagation in order to minimize the error in the output, while approaching the expected result. The change in each weight was calculated with gradient descent [157].

4.3.1.2 Convolutional Neural Network

Convolutional neural networks share the same principles as other artificial neural networks, such as MLP, since they also consist of neurons arranged in layers and utilize iterative weight and bias updates to learn a function. The main differences with other types of neural networks lie in the operations performed, the architecture, and the areas of application. Convolutional neural networks perform kernel convolution by passing matrices of numbers, the kernels or filters, over the input in order to detect features. The base architecture of a CNN consists of the convolutional layer, the pooling layer, and the fully connected layer. The convolutional layer performs the kernel operation described above in order to produce a feature map. Pooling layers use the sliding window method in order to downsample the feature map, reducing its dimensions. The inclusion of pooling layers helps the networks train faster and provides an extra layer of safety against overfitting. Since convolution and pooling layers follow a 3D arrangement of neurons, data need to be flattened in order to produce 1D vectors in the output. Furthermore, fully connected layers are used on flattened input in order to produce the output of the CNN model. Figure 4.1 illustrates the structure of a convolutional neural network. The training process of CNNs shares the same concepts as MLPs but the formulas used throughout this process are modified to accommodate the differences in neuron arrangement and the usage of convolution. This type of neural network is particularly popular in image recognition, since image data can be segmented appropriately [158]. For the purposes of our study, we conducted some experiments on the 1D CNN, since this variant handles data with low dimensionality and is suitable for time series and sensor data analysis.

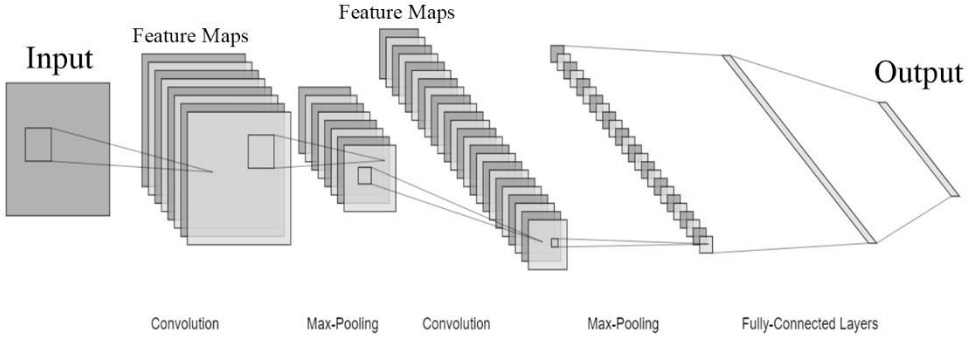


Figure 4.1: Convolutional neural network architecture illustration created using the NN-SVG online schematics tool found in [159].

4.3.1.3 Long Short-Term Memory Network

Long short-term memory networks constitute a variation of recurrent neural networks (RNN) [160] primarily designed to handle long-term data dependencies. Similar to RNN, the LSTM follows a structure consisting of blocks, the LSTM cells. Each cell has its own state C_t, which is passed down to all the blocks in the network. Since the cell state passes through all the LSTM cells, each cell can adjust the state by removing or adding information. Information flowing through the cell state can be regulated with the forget, input, and output gates of every cell. The forget gate of a cell at the timestamp t helps with information removal and can be expressed with the following formula:

$$f_t = \sigma\left(w_f[h_{t-1}, x_t] + b_f\right) \tag{4.2}$$

where x_t is the information at the current timestamp, h_{t-1} is the output of the previous LSTM block, w_f is the weight of the gate, b_f is the bias, and σ is the sigmoid function. Similarly, in Equations (4.3) and (4.4) the input and output gate are expressed with b_i and b_o being the respective biases and w_i and w_o the respective weights.

$$i_t = \sigma(w_i[h_{t-1}, x_t] + b_i) \tag{4.3}$$

$$o_t = \sigma(w_o[h_{t-1}, x_t] + b_o) \tag{4.4}$$

The input gate indicates which values will be updated and stored in the cell state. Furthermore, the output gate indicates the parts of the cell state that will be moved to the output. The information that could possibly be stored to the cell state at the timestamp t is expressed as the candidate d_t and is formulated as:

$$d_t = tanh(w_c[h_{t-1}, x_t] + b_c) \tag{4.5}$$

where w_c and b_c are the respective weights and biases. The current cell state that reflects the adjustments made at timestamp t can be expressed as:

$$c_t = f_t * c_{t-1} + i_t * d_t \tag{4.6}$$

where c_{t-1} denotes the cell state at the previous timestamp. Lastly, the output of the cell h_t is expressed as:

$$h_t = o_t * tanh(c_t) \tag{4.7}$$

LSTM networks have been used extensively in the field of load forecasting since the ability to capture long temporal dependencies efficiently is an important characteristic for time series analysis. In this study we decided to test the base LSTM and two more variants, the stacked LSTM and the bidirectional LSTM, in order to make the comparison more complete. The stacked LSTM variant contained more than one hidden layer of cells and the bidirectional LSTM duplicated the first recurrent layer in order to process an input sequence in both time directions simultaneously. Lastly, the LSTM networks were trained with backpropagation through time and gradient descent [161].

4.3.1.4 Performance Metrics

Since our machine learning task was the prediction of residential active power, regression metrics were considered in order to capture the error in our predictions. Throughout the literature, popular regression metrics such as RMSE, MAE, and MAPE are commonly used to denote the loss in the predicted results of neural networks [162]. In this work we selected MAE as our loss function since it is a linear scoring function of equally weighted differences, rendering this metric easy to understand and interpret.

Our definition of baseline performance for each neural network model examined in this study was the set of MAE scores on the train and test set as well as the average training time per epoch under the assumptions that the configuration parameters were the same and the networks were tested on the same dataset with the same preprocessing adjustments. Seasonal dependencies and special days would certainly improve the performance of our models, but in order to maintain simplicity we examined the univariate case of active power prediction in this work.

4.3.2 Tools and specifications

In order to conduct this comparative study, we used Python 3.7.5, Pandas 0.25.3, and Numpy 1.17.3 for data manipulation, SkLearn 0.21.3 for preprocessing, and Matplotlib for visualization. Furthermore, we used Keras 2.2.4 with the Tensorflow 1.15.0 backend in order to build our neural networks and train our models. The forecasting models were compiled and executed on a desktop with an AMD Ryzen 1700X processor, 8 gigabytes of RAM, and an Nvidia 1080Ti graphics processor. Finally, the code of this study is available on Github [163].

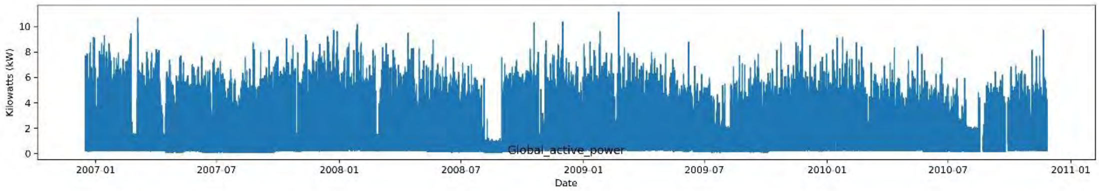
4.3.3 Dataset and Configuration

For this study we used the individual household electrical power consumption dataset from the UCI machine learning repository, which contains 2,075,259 measurements gathered from a house located in the French commune of Sceaux between December 2006 and November 2010. The dataset contains minutely sampled time series for global active power, global reactive power, voltage, global household current intensity, and sub-metering measurements for certain rooms and devices. This dataset was selected as the core input of our models primarily due to the sampling frequency of the data points. Minutely sampled time series matched the prediction horizon that we wanted to target, in order to produce an output at the same level of detail. It is important to note that a lower sampling frequency, for the targeted prediction horizon, would make the input less useful due to the loss of meaningful information. On the other side of the spectrum, an even higher sampling frequency would yield more accurate predictions, but since a sampling frequency of 1Hz would be considered fast for a smart meter and impractical for most applications [164], we opted for minutely sampled data. Furthermore, we conducted all our experiments on the same dataset in order to preserve consistency. After inspecting the data, we deduced that the time series associated with power, voltage, and current intensity was representative of the typical residential behavior and similar datasets would only differ in the data preparation process.

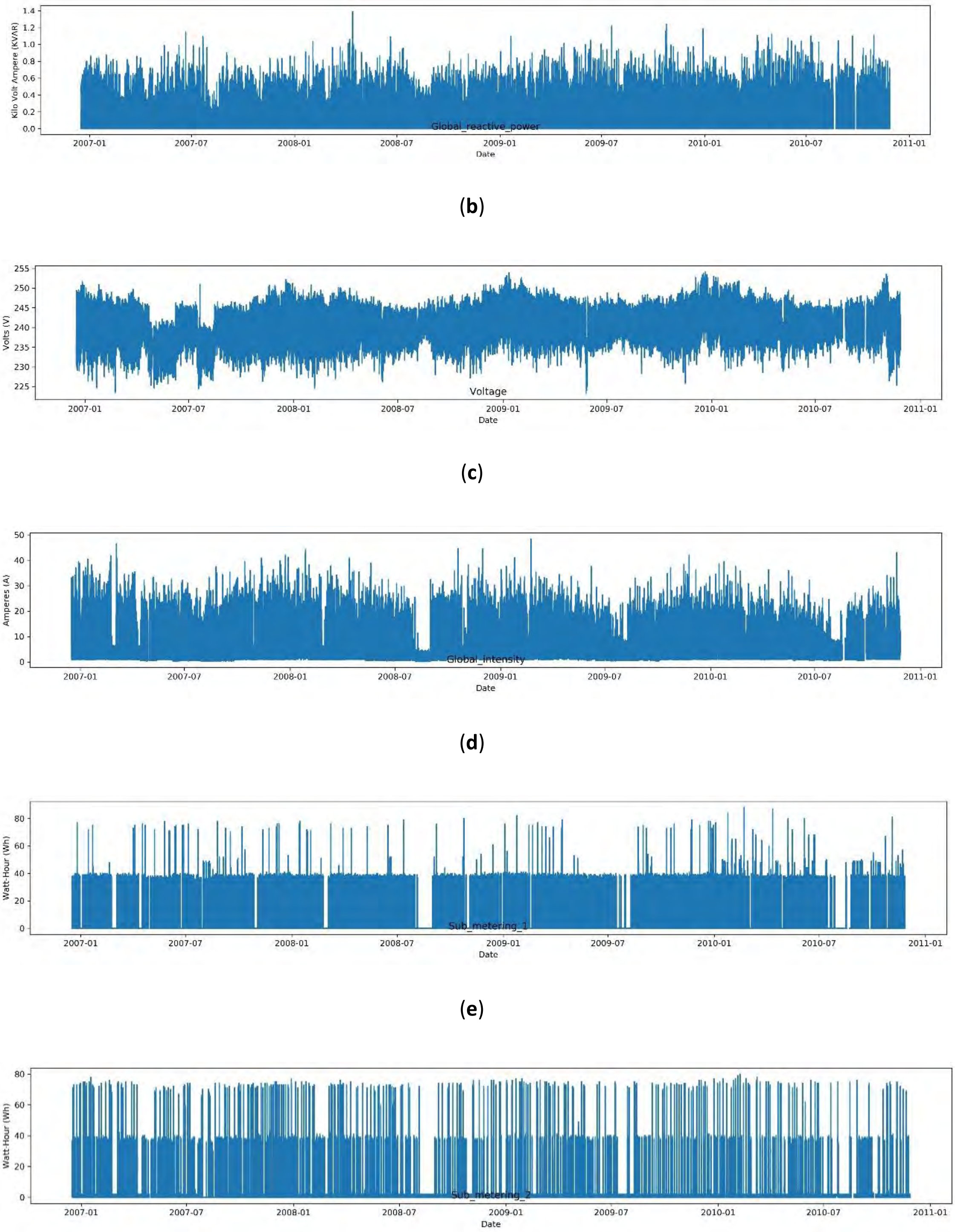
Since we wanted to predict the global active power, we used this time series as our main feature variable, and we examine additional possible useful feature variables in the “Data Exploration” section below. In order to make the data more readable and the records more concise we concatenated date and time information in a single field per record and replaced any missing values marked as “?” with NaN, denoting that these values are not numbers.

4.3.4 Data Exploration

In order to understand the data better, we used line plots for each feature (Figure 4.2) variable of the dataset and we consulted the augmented Dickey–Fuller test of [152] to deduce that the global active power is a stationary time series and therefore is not affected by seasonality. Afterwards, we examined the frequency distributions of the feature variables (Figure 4.3) by plotting histograms and concluded that global active power follows a bimodal distribution. It is interesting to note that the yearly distribution of global active power shows that active power is consistently bimodal (Figure 4.4) each year and as a result, a validation split based on that information would yield a test set that adequately represents the entirety of the data. Finally, we used the Pearson correlation metric (Figure 4.5) in order to check for the property of synchrony between the time series and we observed that global active power is synchronous with global intensity. As a result, we were able to test the impact of global intensity as an extra input variable in our neural networks.



(**a**)


Kilo Volt Ampere (KVAR)
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Global_reactive_power
2007-01
2007-07
2008-01
2008-07
2009-01
2009-07
2010-01
2010-07
2011-01
Date
Volts (V)
255
250
245
240
235
230
225
Voltage
Amperes (A)
50
40
30
20
10
0
Global_intensity
Watt-Hour (Wh)
80
60
40
20
0
Sub_metering_1
Watt-Hour (Wh)
80
60
40
20
0
Sub_metering_2

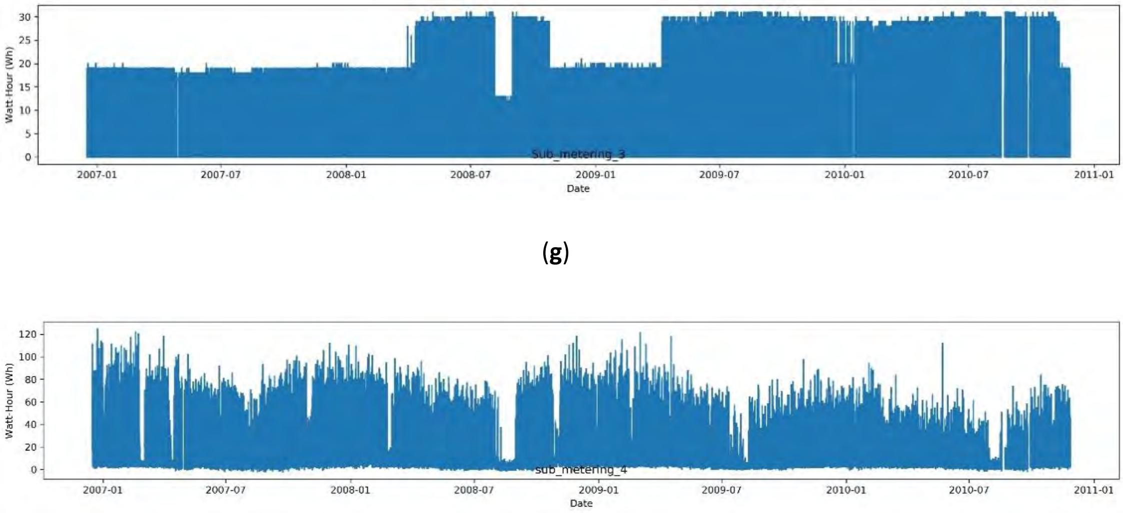
(**b**)

(**c**)

(**d**)

(**e**)

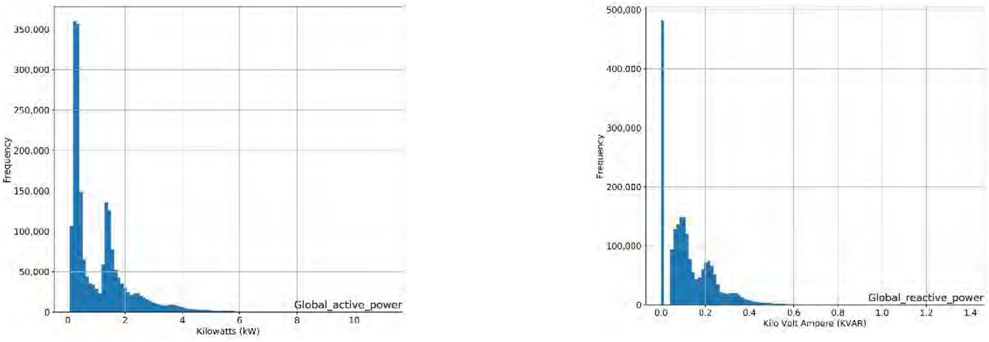
(**f**)



(**g**)

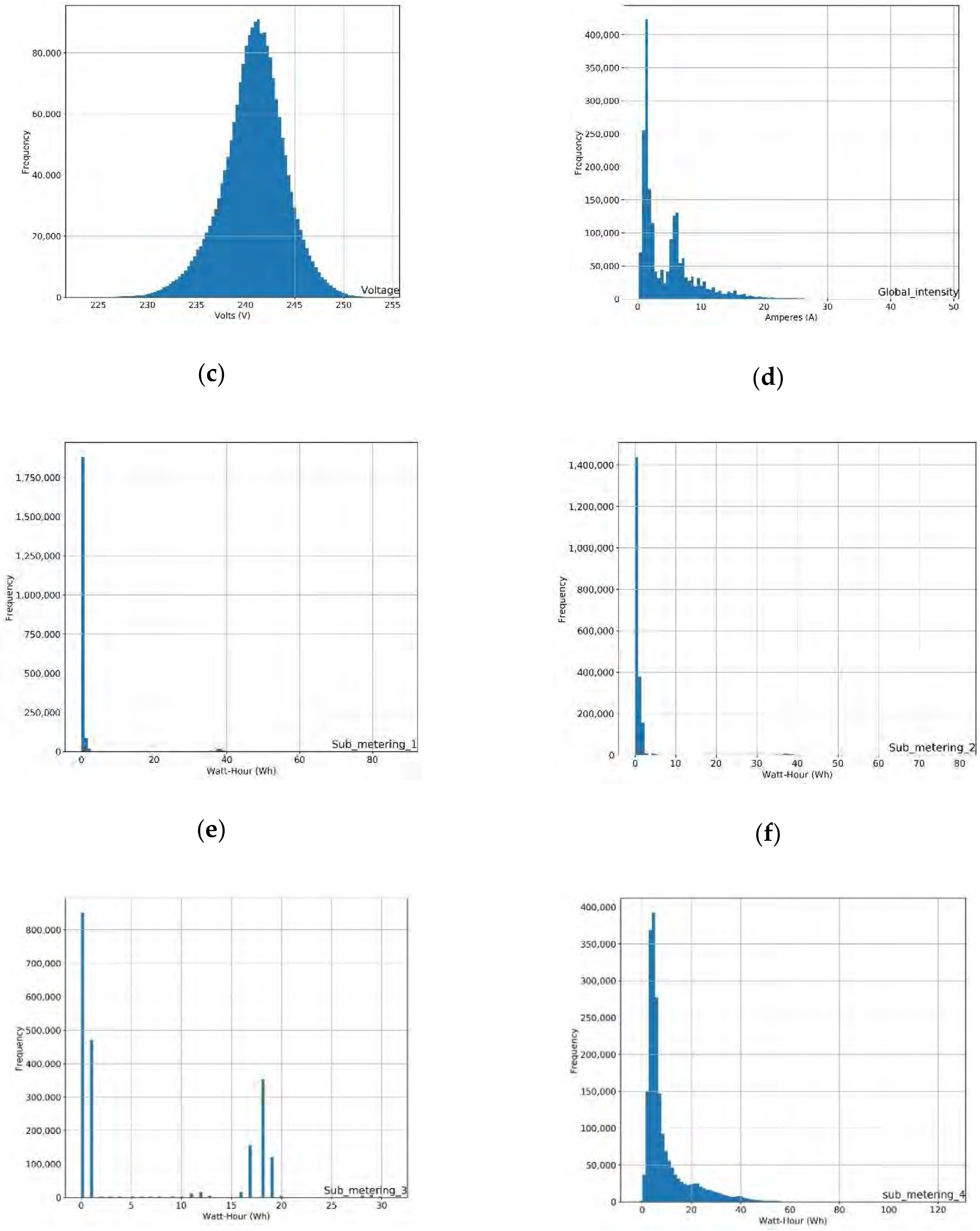
(**h**)

Figure 4.2: Line plots of dataset features. Since each feature is a time series, the y-axis represents the unit of each feature, and x-axis represents the date. (**a**) Global household active power. (**b**) Global household reactive power. (**c**) Household voltage. (**d**) Global household current intensity. (**e**) Kitchen energy consumption. (**f**) Laundry room energy consumption. (**g**) Consumption of electric water-heater and air-conditioner. (**h**) Remaining energy consumption measurements not covered by the sub-metering information.



(**a**)

(**b**)

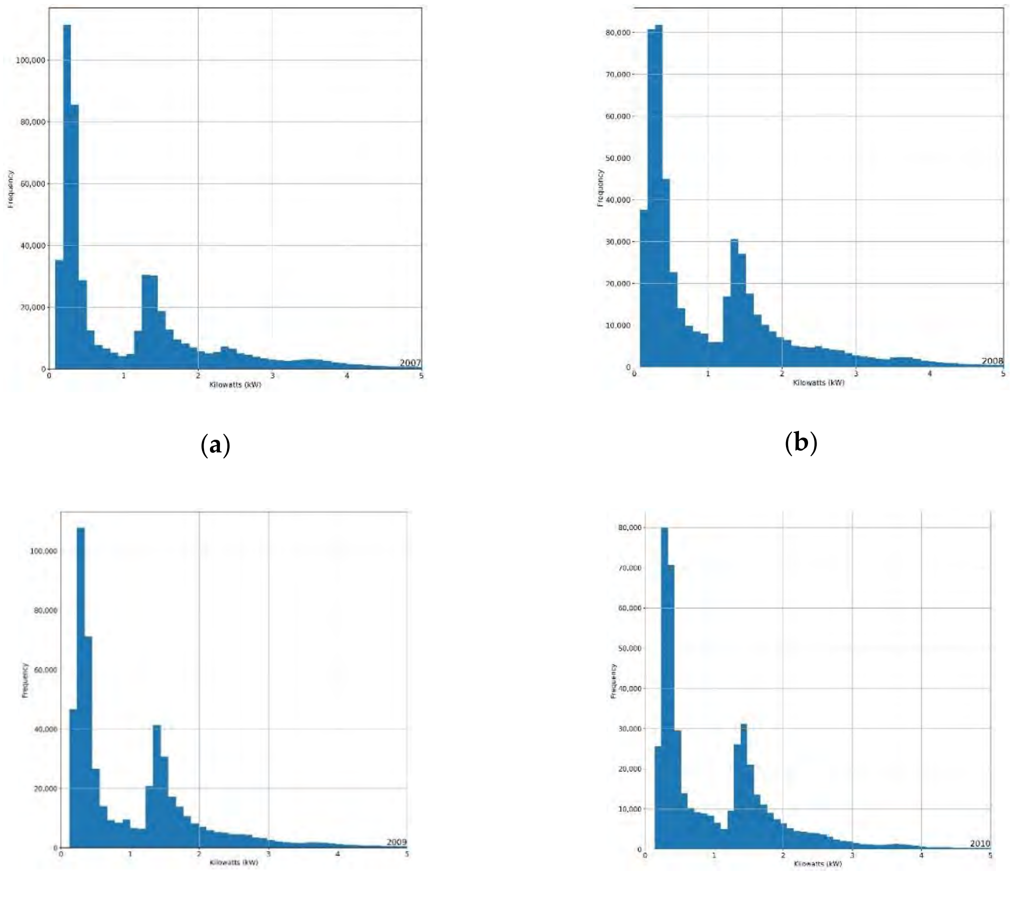

Frequency
80,000
60,000
40,000
20,000
0
225
230
235
240
245
250
255
Volts (V)
Voltage
Frequency
400,000
350,000
300,000
250,000
200,000
150,000
100,000
50,000
0
0
10
20
30
40
50
Amperes (A)
Global_intensity
Frequency
1,750,000
1,500,000
1,250,000
1,000,000
750,000
500,000
250,000
0
0
20
40
60
80
Watt-Hour (Wh)
Sub_metering_1
Frequency
1,400,000
1,200,000
1,000,000
800,000
600,000
400,000
200,000
0
0
10
20
30
40
50
60
70
80
Watt-Hour (Wh)
Sub_metering_2
Frequency
800,000
700,000
600,000
500,000
400,000
300,000
200,000
100,000
0
0
5
10
15
20
25
30
Watt-Hour (Wh)
Sub_metering_3
Frequency
400,000
350,000
300,000
250,000
200,000
150,000
100,000
50,000
0
0
20
40
60
80
100
120
Watt-Hour (Wh)
sub_metering_4

(**c**) (**d**)

(**e**) (**f**)

(**g**) (**h**)

Figure 4.3: Histogram of feature distributions (100 bins). The y-axis represents the number of occurrences and the x-axis represents the unit of each feature. (**a**) Global household active power. (**b**) Global household reactive power. (**c**) Household voltage. (**d**) Global household current intensity. (**e**) Kitchen energy consumption. (**f**) Laundry room energy consumption. (**g**) Consumption of electric water-heater and air-conditioner. (**h**) Remaining energy consumption measurements not covered by the sub-metering information.



(**a**) (**b**)

(**c**) (**d**)

Figure 4.4: Yearly distribution of global active power. The y-axis represents the number of occurrences and the x-axis represents the unit of global active power (kW). (**a**) Global active power distribution in 2007 (**b**) Global active power distribution in 2008 (**c**) Global active power distribution in 2009 (**d**) Global active power distribution in 2010.

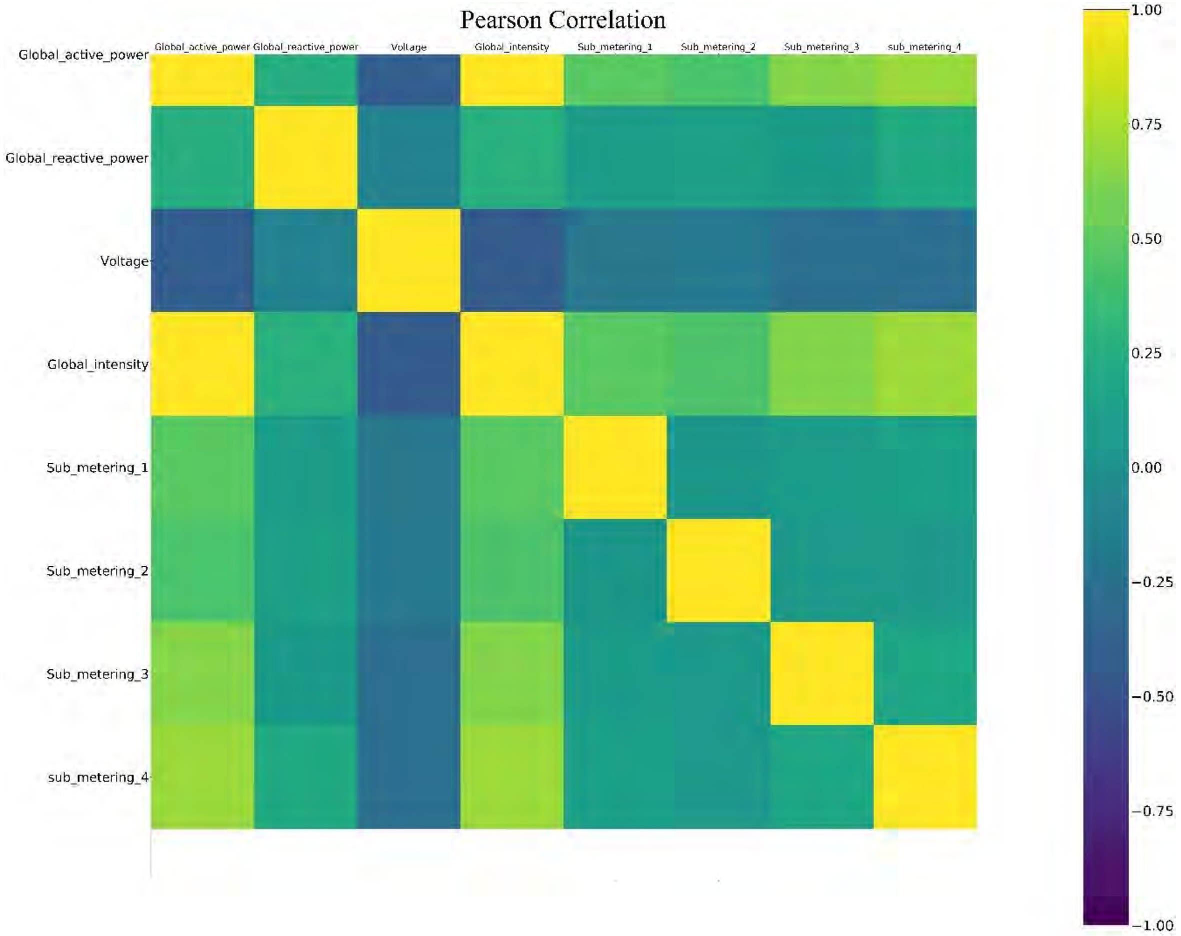


Figure 4.5: Pearson correlation between dataset features. The labels of this 2D array are the names of the dataset features. A positive correlation of 1 is represented by yellow and a negative correlation of −1 is represented by blue.

4.3.5 Problem Formulation and Preprocessing

The first step in this comparative study was to define our research problem and make some initial hypotheses in order to frame the problem space properly. We selected to predict the next minute in the future from the minutely sampled data of the residential UCI dataset. We decided to frame this problem as a supervised learning task by implementing the sliding window method, as proposed by [165], so that our neural networks would be trained to learn a function that maps the input data points to an output. As an initial hypothesis, we considered this formulation in a real world setting where our trained models would be able to benefit an application which could take advantage of minutely predictions. In this scenario, we needed to select a window of lagged observations small enough so as not be affected by cold-start discrepancies. Moreover, the selection of a small window would also cover the main issue of high frequency predictions, which is the interpretation of relationships between data during the training phase of a neural network model. Since data points were sampled at a high rate, we needed to choose critical subsets of data points that clearly depict an upcoming peak, a valley or a more stable active power behavior on the next time step. A big set of data points at the input could lead to a wrong interpretation, since some of them could be interdependent or irrelevant to the predicted output.

Intuitively, we needed to select the smallest window of meaningful data points that would maximize temporal relevance. Since the selection of a single data point in the input may not provide valuable information about the behavior of the time series in the future, a set of two data points could help the neural networks identify patterns from the slope of the line that connects the data or from the level of fluctuations that occur between individual values. Therefore, we decided to set a window of two data points prior to the one that was about to be predicted as our input, on the assumption that these should be the most relevant data points giving us a clear connection to the output. It is certainly possible to conduct the same experiments with a larger input window. Consequently, the input of our neural network models consisted of two columns, the global active power at time $t-1$ and $t-2$. Since the global current intensity had high Pearson correlation with global active power, we added the global intensity at time $t-1$ and $t-2$ as input features on the base LSTM model in order to test the impact of synchrony as an additional experiment. The predicted output was the global active power at time t.

Before configuring the neural networks and building our models, we applied the following preprocessing transformations to the data:

- Values of input features were scaled between [0,1]. Min-max normalization was used in that interval through the MinMaxScaler class of SkLearn since neural networks handle input features well when they are on the same scale and the interval remains small;
- Two input columns were created based on the sliding window method.
- The dataset was split by allocating the first three years of observations to the training set and the last year to the test set. Since the distribution of global active power remains consistently bimodal, we believe that this is a proper holdout validation split;

- The training and test sets were split in input and output columns reshaped as the 3D format (samples, timesteps, features) for CNN and LSTM and (samples, features) for MLP, since it expects a 2D format.

4.3.6 Neural Network Configurations

In this study we configured the most prominent neural networks in time series forecasting and kept the activation and loss functions the same, as well as the compilation and training parameters, in order to derive a baseline performance. Therefore, we examined the behavior of the base LSTM network as well as the stacked and bidirectional variants. Furthermore, we compiled models for a 1D CNN and an MLP in order to compare more neural network architectures. As proposed in [152], neurons on the input layer of every model followed the dimensions of the feature columns and were equal to the window size we selected above. Every model had one hidden layer of 100 neurons and 1 neuron on the output layer. We selected 100 neurons as the hidden layer size because we wanted every network to have sufficient processing capacity. The size of the output layer was determined by the expected result. Therefore, one neuron in the output layer would predict the global active power at time t. Since the stacked LSTM architecture should contain more hidden layers, we decided to use one extra hidden layer with 100 neurons for our stacked LSTM model. Additional experiments were conducted on the stacked LSTM architectures where neurons would be dropped out randomly with a probability of 0.2 in order to test the impact of regularization on more complex neural network structures.

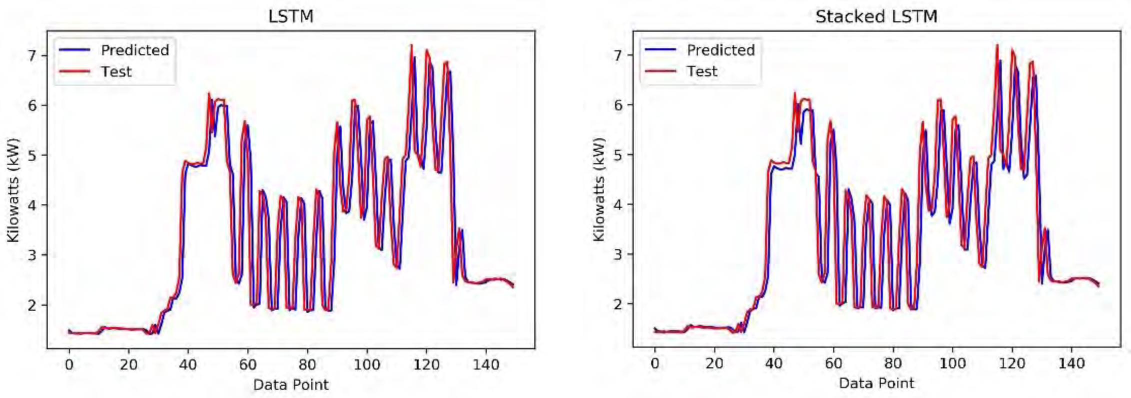
The 1D convolutional network model contained a convolutional layer with 64 filters, a kernel size of 2, and the input was padded accordingly in order to obtain an output of the same length. Additionally, in our CNN architecture we added a pooling layer with a pool size of two in order to downsample the detected features. The pooling layer shared the same padding configuration as the convolutional layer. We proceeded to flatten the output of the pooling layer and use it as the input to a fully connected layer of 100 neurons. The predictions derived from the CNN model were obtained in the output layer. Finally, the MLP model had a simple structure of one input layer, one hidden layer, and one output layer, following the general configuration mentioned above.

The activation function used between the layers of every model was the rectified linear unit (ReLU). Moreover, we used the Mean Absolute Error (MAE) function to measure the loss, since this is a simple metric which denotes the absolute difference between the data and the model predictions. In addition, the Adam optimizer was used for adaptive learning rate optimization [166]. The training parameters for every model were considered with regard to optimal MAE values on the test set. Hence, we used holdout validation in order to use the first three years of data as the training set and the last year of data as the test set. This split was selected because we did not intend to tune the parameters of each model after training. In addition, more sophisticated validation techniques applicable to time series, such as nested cross-validation, would add unnecessary complexity to the comparison we attempted to examine. We acknowledge that the increased processing capacity of the hidden layers could cause increased total training times and a possible risk of overfitting. Therefore, in order to avoid overfitting, we stopped the training process and saved the models when the validation loss scores could not improve further, while their metrics were observed simultaneously in the notion of logical conjunction on the same epoch interval. After the eighth epoch the validation loss of any individual model was at least slightly worse than the best loss observed in that eight-epoch interval for that same model. As a result, the neural network models were trained after eight epochs with a batch size of 72 training samples.

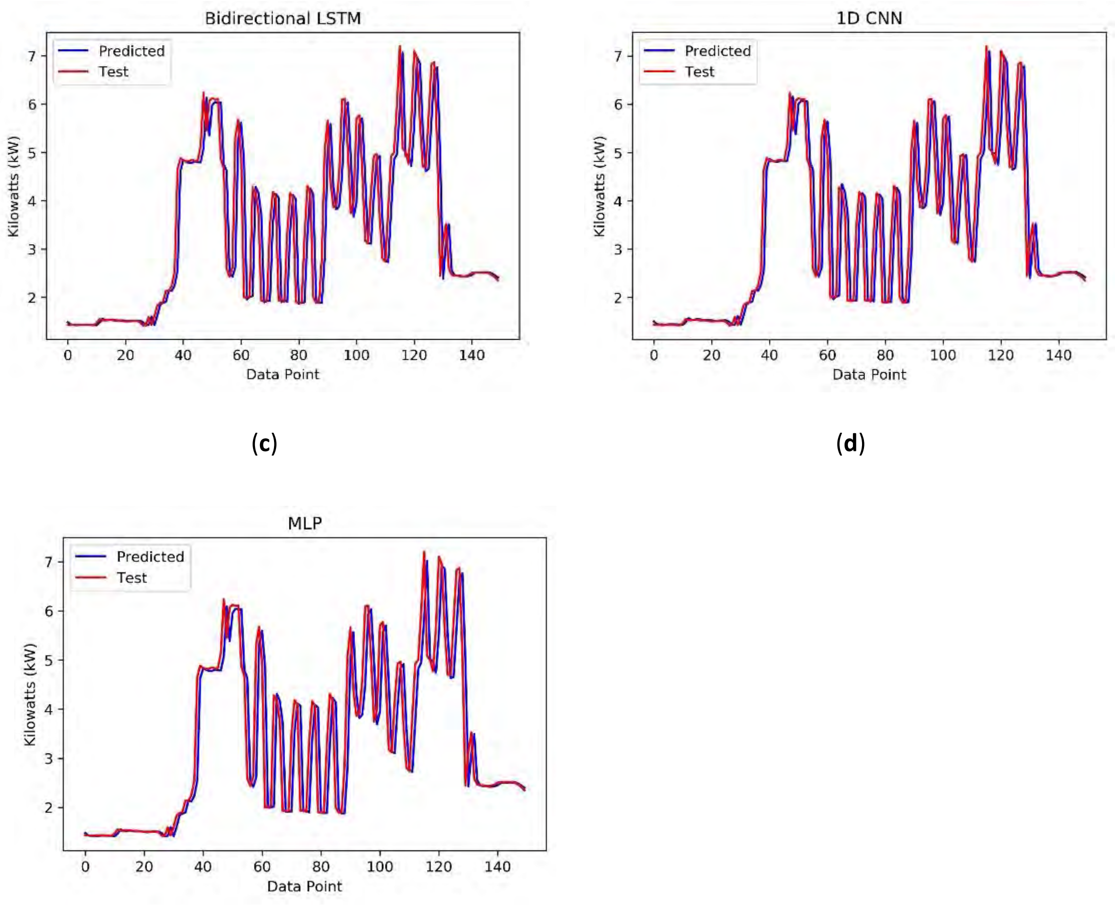
4.4 Results

As a result of the above configurations we evaluated the performance of our models from the values of the loss function on the training and test set. In Figure 4.6, we present the graphs containing a sample of 150 global active power data points from the test set and 150 predicted data points generated by each neural network model, in order to visualize the robustness of our predictions on each time step. Moreover, in Figure 4.7, we present the graph panel containing the loss function values throughout the training process of each neural network model. The examined models performed well when trying to predict the global active power from the unknown data in the test set since the data points at the same time step neither diverged drastically nor did they follow an unusual pattern. However, we can observe that when a peak or a valley occurred in the test data, the models made a less accurate prediction, resulting in greater deviation from the actual active power data points in the same time interval. A distinct example of this phenomenon can be detected in the region between the 45th and 55th data points, where the predicted value for the power peak was lower than the actual test value. Through this example, we are also able to notice the difference in the quality of predictions among our neural network models. The stacked LSTM model made the least accurate peak prediction in that region, while other models, such as the 1D CNN and MLP, were able to achieve approximations closer to the actual peak value. These observations can be generalized to the entirety of the dataset, since it is indeed difficult to anticipate peaks and valleys in such a short prediction horizon when only the most recent and most relevant data points of the same feature are considered.

Furthermore, the graphs of the training and test loss show that we avoided overfitting and underfitting with that configuration, since the curve of validation loss was always below the curve of training loss. The curves of the training loss function in each graph show that there was a steady improvement of the prediction quality throughout the training process, whereas the validation loss curves show that after a small number of epochs no further improvement could be achieved when testing the models on unknown data, given those configuration parameters. It is interesting to note that models with inferior training loss results, such as the 1D CNN, had a decent performance on unknown data and were able to fit the data reasonably well in regions where peaks and valleys occurred.



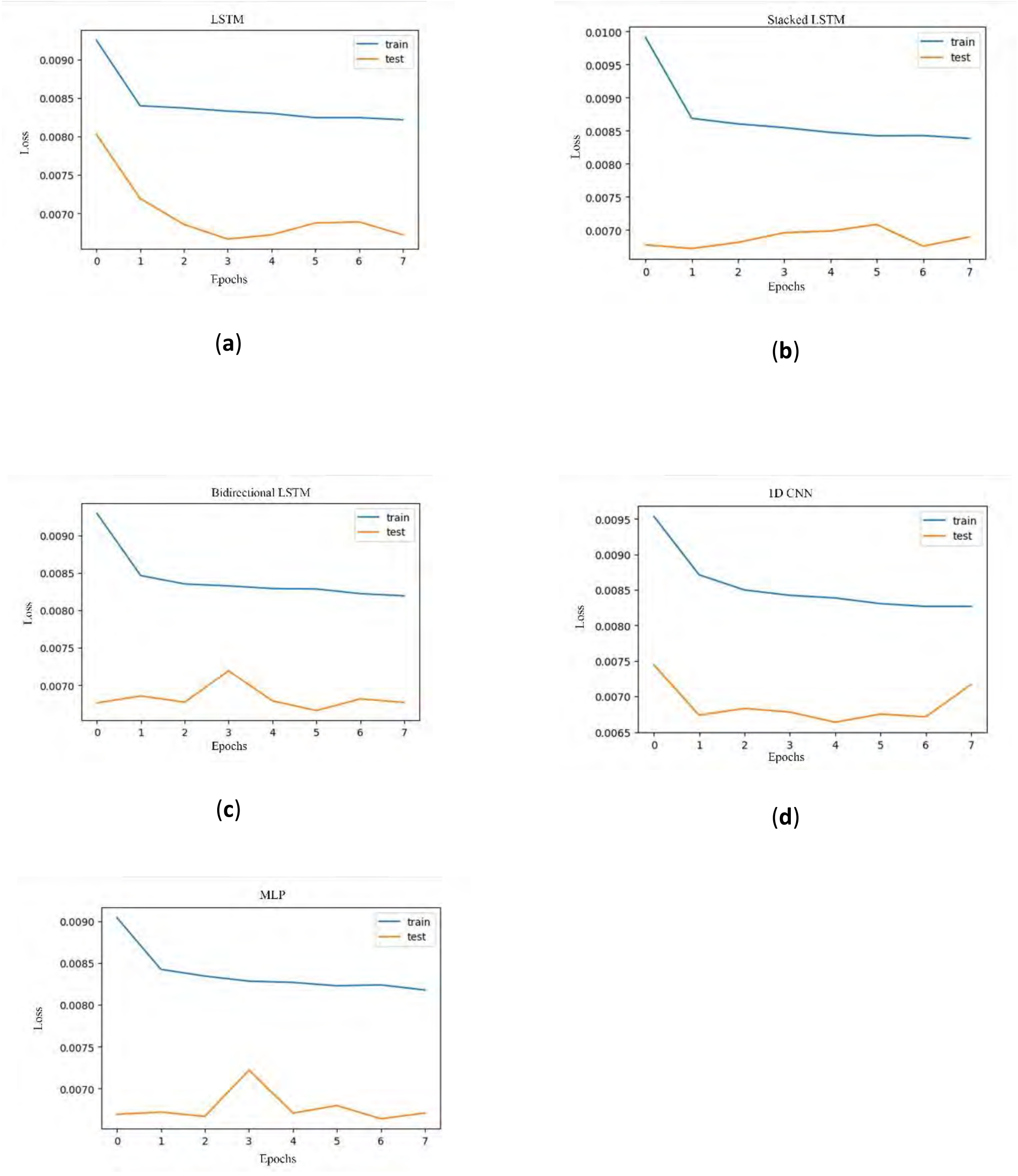
(**a**) (**b**)



(**c**) (**d**)

(**e**)

Figure 4.6**:** Prediction evaluation on 150 data points for (**a**) Long Short-Term Memory (LSTM) network, (**b**) Stacked LSTM, (**c**) Bidirectional LSTM, (**d**) 1-Dimensional Convolutional Neural Network (1D CNN), (**e**) Multilayer Perceptron (MLP). The red line represents the data points of the test set and the blue line represents the data points predicted by each model on the same time step. On the y-axis we set the unit of global active power (kW) and on the x-axis we enumerated the data points.



(a)

(b)

(c)

(d)

(e)

Figure 4.7**:** Loss function on train and test set for eight epochs of training on (**a**) LSTM, (**b**) Stacked LSTM, (**c**) Bidirectional LSTM, (**d**) 1D CNN, (**e**) MLP. The blue line represents the loss function during training and the orange line represents the loss function during validation. On the y-axis we set the values of the loss function and on the x-axis, we enumerated the epochs.

Table 4.1 presents the evaluation metrics for every neural network type on the training and test sets. After examining the overall error loss function scores of each model, we deduced that the top performing ones were the MLP and the Baseline LSTM. Since the LSTM network is more suitable for handling temporal relationships, we re-ran the experiment by adding the extra feature of global current intensity in order to test the impact of synchrony and we observed a decrease in training time. Furthermore, we ran tests by adding dropout to the stacked LSTM variant, since its initial performance on unknown data was the worst when compared to other LSTM models, but that change did not yield better results. Finally, Table 4.2 shows the average training time of each model and we observed that MLP was able to converge to an acceptable mapping of the input to the output faster than the other network models. It is, therefore, interesting to note that some of the more complex LSTM variants went through a distinctly slower training process.

Table 4.1: Training and testing scores of neural network models.

Scores	**MLP**	**1D CNN**	**Baseline LSTM**	**Baseline LSTM (Synchrony)**	**Stacked LSTM (No Dropout)**	**Stacked LSTM (Dropout 0.2)**	**Bidirectional LSTM**
Train Score (Loss)	0.00797	0.00846	0.00798	0.00826	0.00824	0.00839	0.00808
Test Score (Loss)	0.00670	0.00716	0.00672	0.00701	0.00689	0.00711	0.00677

Table 4.2: Average training time per epoch for each neural network configuration.

Neural Network Type	**Average Training Time per Epoch (seconds)**
MLP	20
1D CNN	42
Baseline LSTM (Synchrony)	98
Baseline LSTM	120.25
Stacked LSTM	165.25
Bidirectional LSTM	269.5

4.5 Discussion

This work presented the baseline performance comparison of neural network models for minutely active power forecasts derived from residential data. In our supervised learning formulation of this high frequency forecasting problem we observed that the multi-layer perceptron performed best in terms of loss and average training time. Since MLP follows a simpler structure and recognizes a 2D data format we can deduce that due to the selection of a small and relevant set of input data points, the network was able to converge fast, producing fairly accurate predictions in the output. The long short-term memory network and its variants converged slower, possibly due to their computational complexity, since the data relationships that we wanted to identify did not refer to data points far into the past. We expect changes to the training and test scores when the window size and the number of input variables increase and anticipate that LSTMs will be able to produce accurate predictions at a higher complexity.

Our study could be useful in approaching high granularity forecasts with machine learning methods. Since the baseline performance of neural networks was evaluated and minutely sampled energy data was explored, we now have a starting point for further parameter tuning and experimentation. Future work could apply grid search techniques [167] for hyperparameter optimization in order to improve our baseline models. It would also be useful to explore the potential of modular solutions combining the neural networks we studied here in a pipeline-like structure, in order to investigate whether other important aspects of the highly granular time series forecasting in the energy sector emerge. For example, more complex neural network architectures could utilize these models in order to enrich a dataset at the input module or in order to derive partial predictions based on different criteria in parallelly working modules.

Chapter 5 Structural Forecasting Framework Towards Generative Combinatorial Modeling

5.1 Motivation

Combinatorial modeling is at the forefront of short-term energy data forecasting research since most recent research efforts present more complex unified estimator structures consisting of multiple models. The value of combinatorial approaches is immense since the integration of multiple estimators could provide several improvements to the forecasting framework. First, combinatorial estimators address the assumptions, challenges and limitations of standalone models and develop architectures that are more resilient to the individual weaknesses that could hinder the performance of each structure. As a result, combinatorial approaches typically yield lower error metrics due to the discovery of more complex time series patterns and handle outliers more efficiently. Second, these models address energy time series diversity since different estimators could be utilized for the interpretation of data gathered from different types of clients, buildings or energy markets. Consequently, combinatorial estimators are more flexible and have the potential to adapt to more demanding forecasting tasks. These observations highlight the appeal of combinatorial modeling and are a primary source of motivation towards the introduction of novel and performant combinatorial estimators.

On the other side of the spectrum, motivation towards the enhancement of the forecasting framework with regards to combinatorial estimation methodologies stems from the risks and performance challenges in the design process of those models. First, it is evident that this research area is vast and contains a plethora of standalone models, resulting in an expansive set of estimator combinations. Therefore, extensive experimentation and testing need to be conducted since more performant structures could still be discovered. Following this observation, combinatorial model design becomes an increasingly intricate task since the selection of estimator members could be a difficult process. This difficulty could be directly connected to the uncertainty that surrounds the inclusion of estimator members in the combinatorial structure since the search for optimal estimators that fit a specific forecasting task does not commonly follow a deterministic and structured approach. The selection process is typically conducted based on expert knowledge through the performance evaluation of the methodologies on similar tasks or presented arbitrarily. As a result, the focus is shifted towards time consuming experimentation for the validation of model structure. Performance risks are prevalent in this design approach and the resulting models lack interpretability as the basis of those experiments remains vague. Second, the data volume and the degree of diversity present in energy time series could be linked to the challenges of dimensionality and drift respectively. Consequently, combinatorial design approaches need to respect the restrictions imposed by those challenges and derive resilient solutions that preserve adequate performance.

The previously discussed combinatorial design benefits coupled with the challenges, risks and research gaps surrounding the estimator selection process led to the examination of an interesting combinatorial modeling scenario and the introduction of a novel deterministic strategy for estimator selection. This scenario considers the task of total demand forecasting in the cluster-based aggregate forecasting framework given the consumption data of a diverse client base and highlights most of the challenging aspects in combinatorial design. Since this task processes data from several types of clients anonymously through this framework it is clear that a static standalone estimator would not be able to capture all consumption patterns sufficiently. Additionally, the construction of a combinatorial structure would not benefit from a non-deterministic or arbitrary approach since trial-and-error experimentation as well as efficient model recalibration would be infeasible as the client base evolves. Therefore, a structured approach for the generation of estimator sets that could adapt to the data optimally would be beneficial in this case.

For the purposes of this study, ensemble learning methods were utilized as the main combinatorial methodologies since forecasting performance is more predictable and easier to monitor in this framework. Since the optimal combination of estimators in ensemble learning methods typically ensures a small performance improvement when compared to standalone approaches, the evaluation of the generated estimator sets could be easily tracked and interpreted. The deterministic selection strategy presented in this study is influenced by the relative performance examination of the training process on peak and non-peak indices, forming an algorithm that respects the shape of each input time series. The proposed methodology leads to the automatic selection of optimal estimator combinations given an initial estimator superset and the testing of the ensemble models on different client groups validated that this approach could achieve the expected lower error metrics of optimal ensembles as the partial aggregate demand is predicted more accurately for each consumer group. The following sections correspond to the introduction, methodology, results and discussion of this published work.

5.2 Introduction to Structural Ensemble Regression for Cluster-Based Aggregate Electricity Demand Forecasting

Smart grid technologies and applications are at the forefront of modern electricity network research and development due to the increasing number of challenges that hinder the performance of the traditional power grid as well as the accrescent need to transition towards a digital ecosystem where the bidirectional flow of information between the electricity provider and consumers is simplified. Since the penetration of renewable energy sources introduces additional volatility that could compromise the reliability of the grid and the increasing electricity demand from a growing number of consumers could lead to the occurrence of irregular events such as blackouts, the centralized structure of the traditional grid has limited control over these phenomena [168,169]. Therefore, the development of smart grids that rely on the wide deployment of smart meters is necessary for the efficient, adaptive and autonomous management of consumer loads in a distributed framework. Consequently, a large volume of high dimensional sensor data are extracted from smart meters and the efficient processing as well as prediction of electricity load are crucial tasks that reinforce advanced transmission, distribution, monitoring and billing strategies [170]. Load forecasting tasks could be developed for different time horizons depending on the focus of each smart grid application. In the context of real-time load monitoring, demand response and smart energy pricing, accurate short-term predictions and point forecasts could support energy management systems as well as decision-making models in shaping load allocation and pricing strategies for consumer groups that share similar load profile characteristics. Additionally, high-resolution predictions of total electricity demand could assist in the stability of the grid through the real-time detection of irregular events, enabling online scheduling at a higher level while preserving consumer privacy. It is equally important to note that high-frequency demand forecasts could result in the optimization of energy resources through the examination of total load fluctuations at a higher granularity as well as the optimization of bidding strategies when utility companies purchase electricity from energy markets, enabling short-term flexibility and more efficient market balancing [171].

Artificial intelligence and machine learning contributed significantly towards the accurate estimation of total demand through the supervised learning task of regression analysis. Firstly, simple linear models such as ordinary least squares linear regression [172], ridge [173], lasso [174], stochastic gradient descent (SGD) [175] and Huber [176] estimators search for the line of best fit that optimally describes the relationship between the dependent and independent variables. Linear models are commonly used in large-scale forecasting tasks due to their low computational cost and interpretability. However, these models do not interpret complex nonlinear relationships and the impact of outliers within the data could hinder the forecasting accuracy. Therefore, more robust methods were developed such as the generalized median Theil-Sen estimator [177], gradient boosting models based on decision trees such as XGBoost [178], the least angle regressor (LARS) [179] and efficient unsupervised learning models were adapted such as k-nearest neighbor (KNN) [180] and support vector machine models for regression (SVR) [181] in order to achieve higher accuracy in high dimensional spaces and ensure resilience against multivariate outliers. Secondly, neural network models such as the multilayer perceptron [182–184] and long short-term memory network [185] could be applied to this forecasting task in order to capture nonlinear relationships as well as time dependencies adaptively, operating as function approximators in a black-box approach. It is important to mention that while the standalone performance of these models could result in predictions with low error metrics, combinatorial and hybrid approaches such as ensemble learning could be considered for further performance improvement when a suitable combination of models is discovered through arbitrary selection, informed selection based on expert knowledge and experimentation or criteria examination. Time series estimator output could be combined in a meta-modeling framework for stacked generalization, averaged in a voting framework or used to improve another set of estimators sequentially through boosting [186,187].

It is evident that since consumer load profiles are organized in high dimensional time series, forecasting total electricity demand through the direct use of regression analysis would be computationally expensive and the resulting estimators would exhibit diminishing accuracy as more load data from different types of consumers is collected. Consequently, in order to provide solutions to the challenges of dimensionality and scalability, load forecasting approaches in this sector utilize clustering and aggregation strategies as a preprocessing step, altering the shape of the data before it is used for the training of estimators. Cluster-based approaches mainly focus on the segmentation of the consumers into groups based on similar characteristics or by utilizing heuristic algorithms. Predictions for each cluster are extracted and summed to derive the total demand forecast. This approach may become computationally expensive when the consumer base is large and the optimal number of clusters remains small. However, clustering approaches are valuable to demand forecasting since they preserve load patterns within each consumer group. Furthermore, advances in distributed computing attempt to develop more efficient parallelizable models to offset that computational cost [188]. Aggregation approaches attempt to develop a single prediction model where the time series dataset is typically derived from the summation of all consumer load profiles. This approach offers substantial benefits in terms of data compression at the cost of prediction accuracy since the impact of the patterns found in individual consumer time series as well as the behaviors exhibited in different clusters could be reduced greatly in the resulting time series [189]. Combining the clustering and aggregation methods led to the development of the cluster-based aggregate framework where the time series for each consumer group can be aggregated before the prediction in order to derive the estimated partial sum of total demand. This approach attempts to balance accuracy and computational cost and presents a scalable alternative that improves the performance of estimators as the size of the customer base increases.

In the modern power grid, the evolution of the increasingly diverse customer base coupled with the overall complexity of the data collection process often result in datasets that include missing values, outliers and typically exhibit structural issues due to variations in monitoring periods and differences in the quality of the available equipment. Therefore, the performance of load estimators depends on the dataset structure as well as the ability of data-driven models to adapt to the given input. Consequently, a static load estimation model may not maintain optimal performance across multiple forecasting tasks since some components may underperform due to the unique characteristics of the input. This phenomenon could be easily observed in the processing of clustered time series for the prediction of total electricity demand. The utilization of clustered time series results in several structurally different datasets derived from different consumer groups. When the datasets pass through a single type of estimator or a static combinatorial structure, divergent performance metrics between partial demand predictions could be observed, resulting in suboptimal overall performance when the values are aggregated for the estimation of total demand. The potential failure to adapt to an individual dataset could be more impactful in short-term and very short-term forecasting tasks since lagged features at higher resolutions would require a higher volume of information in order to properly capture meaningful temporal dependencies between samples. These load forecasting issues could be connected to the challenges of data drift and concept drift in machine learning modeling. The challenge of data drift indicates the deterioration of model performance as the distribution of input data changes and the challenge of concept drift denotes the difficulty of the model to adapt to the data as the mapping between the input and the target variable changes [190,191]. These challenges could arise when load time series are considered for the prediction of total demand since data distributions could vary between different client types and the relationship between input and output could change as the size of the customer base and the complexity of observed patterns increase. Furthermore, the impact of those challenges could affect the performance of combinatorial approaches such as ensemble learning significantly, since potential concept or data drift across multiple datasets could result in inefficient estimator combinations that may yield suboptimal performance when compared to standalone models due to underperforming components. As a result, the focus should be shifted towards modular estimator structures that utilize well-defined, criteria-based strategies in order to select estimation components that would not underperform given a specific input, thereby reinforcing consistency. Moreover, the implementation of estimator selection strategies would lead to less arbitrary and less ambiguous combinatorial structures since estimator members would be directly connected to the input data.

Several recent research projects presented interesting demand forecasting approaches utilizing a plethora of regression estimators for centralized analysis as well as distributed modeling in clustering and aggregation frameworks. Ceperic et al. [192] proposed a model input selection strategy for SVR-based load forecasting, outperforming state of the art short-term forecasting approaches in terms of accuracy. Wijaya et al. [193] examined the performance of linear regression, multilayer perceptron and support vector regression on several clustering strategies for short-term load forecasting, highlighting the dependence of the cluster-based aggregate forecasting approach on the number of clusters as well as the size of the customer base for optimal performance. Karthika et al. [194] proposed a hybrid model based on the autoregressive moving average and support vector machine algorithms for hourly demand forecasting, showing reduced error metrics and increased convergence speed through the efficient merging of those machine learning methods. Laurinec and Lucká [195] studied the impact of unsupervised ensemble learning models on clustered and aggregated load forecasting tasks and deduced that the adaptation of those methods could lead to improved performance. Fu et al. [196] developed an adaptive cluster-based method for residential load forecasting through the utilization of self-organizing fuzzy neural networks, harnessing the unique characteristics of each cluster. Li et al. [197] utilized subsampled SVR ensembles coupled with a swarm optimization strategy, resulting in a deterministic and interpretable forecasting model that efficiently combines the output of multiple predictors. Bian et al. [198] proposed a similarity-based approach and implemented K-means clustering and fuzzy C-mean clustering for the derivation of features based on locally similar consumer data for the training of a back-propagation neural network. Sarajcev et al. [199] presented a stacking regressor that combined gradient boosting, support vector machine and random forest learners for clustered total load forecasting, signifying that the robust estimation of electricity consumption can be achieved when a suitable model combination is discovered. Cini et al. [200] examined the performance of the cluster-based aggregate framework on deep neural network architectures and highlighted the suitability of this clustering approach for short-term load forecasting. Additionally, this project raises awareness about the complex and challenging nature of implementations involving multiple predictors in this framework for future research. Kontogiannis et al. [201] presented a meta-modeling technique combining long short-term memory network ensembles and a multilayer perceptron to forecast power consumption and examine the impact of causality and similarity information extracted from client load profiles. This project presented a novel strategy for the decomposition of load data into causal and similar components, resulting in a combinatorial structure that outperformed the standalone load representation. Stratigakos et al. [202] proposed a hybrid model combining time series decomposition and artificial neural networks for efficient short-term net load forecasting. The approach presented in this work reduced the error metrics of multi-layer perceptron and long-short term memory network and highlighted the impact of trend, seasonal and noise time series components. Zafeiropoulou et al. [203] proposed a pilot project that addressed the challenges of congestion and balancing management in energy systems and provided robust solutions that could improve resource flexibility and power system stability. Phyo et al. [204] developed a voting regression model including decision tree, gradient boosting and nearest neighbor estimators, resulting in improved performance when compared to the baseline standalone predictors. This symmetrical forecasting approach achieved the expected performance boost that is often observed in optimal ensemble models and when compared to the autoregressive moving average model, the proposed estimator yielded lower error metrics due to the highly performant components included in this ensemble structure.

In this study, we focused on the high-frequency point prediction of total electricity demand on the cluster-based aggregate framework for the development and evaluation of adaptive and structurally flexible stacking and voting ensemble models. This very short-term forecasting approach addresses the challenges in combinatorial forecasting models through the processing of diverse clustered time series and the introduction of a well-defined member selection strategy. The ensemble estimator considers several peak detection perspectives for member selection. The membership of base learners is determined based on the performance examination from a set of 11 candidate estimators on subsets of training observations from the actual as well as the predicted clustered time series, detected as peaks and non-peaks. The proposed ensemble regressors were evaluated in a case study utilizing smart meter data from a dataset of 370 Portuguese electricity consumers for a period of 4 years. The goal of this project is to examine the impact of this criteria-influenced member selection strategy on the cluster-based aggregate framework and propose alternative adaptive ensemble models that combine knowledge extracted from different estimators based on core time series characteristics. Since recent research efforts have deployed training performance indicators and feature-based criteria for member selection on centralized ensemble models, our contribution aims to expand on this approach through the implementation of flexible ensemble estimators constructed from different base learners on each consumer cluster. Additionally, several adaptive hybrid modeling and meta-modeling approaches on clustered and aggregated frameworks typically include the most prominent estimators for model fusion based on expert knowledge or arbitrary selection. Consequently, the effect of criteria-based ensemble structures for cluster-based aggregate load forecasting is not thoroughly explored. Our study aims to provide meaningful insights while addressing this research gap. Case studies and model comparisons in the literature show that a static ensemble structure or a standalone estimator may not always yield the same level of performance stability on all types of consumer load time series. This observation holds true in the examination of clustered time series since each cluster needs to be processed differently in order to capture the patterns of a specific client group efficiently. Therefore, our project considers the fundamental characteristic of peak and non-peak detection in time series and attempts to adjust the ensemble structure for each cluster locally, reinforcing the idea that more modular and dynamic estimation strategies should be developed for those distributed frameworks. The deployment of our proposed approach in real-world applications could support advanced energy management systems and contribute towards the development of more robust bidding strategies through the extraction of more precise total demand analytics in short time intervals.

In Section 5.3, we present the main methodologies involved in the implementation of our proposed models, including the ensemble learning structure for stacking and voting regression, an overview of the cluster-based aggregate framework for total demand forecasting, an inspection of well-known clustering evaluation methodologies and the structure of our proposed ensemble regressors. Additionally, information about the dataset and the definitions of error metrics are provided in this section for completeness. In Section 5.4, we analyze the results of our experiments and evaluate the performance of our models, comparing them to baseline standalone estimators. In Section 5.5, we discuss the impact of the experimental results and outline the advantages and the potential challenges of the proposed models. Furthermore, we provide insights on future research directions that could expand on our forecasting approach and possibly enhance model performance for similar applications in the energy sector. Finally, in Section 5.6, we present the conclusions derived from the experiments and the analysis of the results.

5.3 Materials and Methods

5.3.1 Stacking and Voting Ensemble Regression

Time series forecasting estimators attempt to capture linear and non-linear patterns from the training data in order to fit a model that is able to generalize well when new observations are tested. However, due to the coexistence of those two types of patterns, a single estimator may not be able to achieve both good interpretation and optimal forecasting performance. The suboptimal accuracy could be attributed to high bias, resulting in limited approximation flexibility, or high variance, leading to larger fluctuations in the estimated time series when value changes occur in the training data. Therefore, models with a high bias could be prone to underfitting, resulting in poor performance on the training and test set. Additionally, models with high variance are prone to overfitting, resulting in optimal performance on the training set and suboptimal accuracy on the test set. Ensemble learning methods acknowledge those potential model instabilities and contribute to the implementation of more robust estimators that are more resilient to noise through the combination of multiple regression models [205]. In this project, we develop the forecasting model structure and investigate the impact of stacking and voting ensembles on clustered aggregate load time series.

The stacking ensemble regression approach combines multiple estimators in order to construct a meta-model that consists of multiple layers responsible for processing estimated time series as features for the training of a new estimator. For this study, we consider the simple two-layer stacking ensemble structure for time series regression tasks. Layer 0 trains several diverse estimators commonly known as base learners and produces a feature set of estimated time series, denoting different representations of the target variable, forming the stacked dataset. Layer 1 usually consists of a simple model such as linear regression that is trained on the stacked dataset in order to derive the final predictions. Figure 5.1 presents this two-layer structure for N base learners. Multilayer stacking extends this structure through the derivation of multiple meta-model time series that are utilized for the training of a subsequent estimator, following the process of the first two layers [206,207].

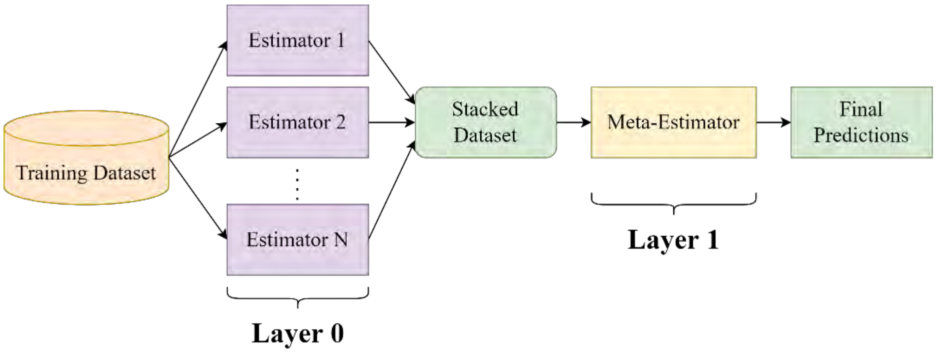


Figure 5.1: Stacked ensemble regressor structure with two layers and N base estimators.

Voting ensemble models attempt to correct highly divergent estimated time series values through the averaging of multiple estimators. Firstly, a set of similarly performant models is selected for the prediction of the target variable. The members of voting regression typically share similar error metrics during training in order to preserve stable performance after the averaging process. Secondly, a weighting strategy is applied in order to denote the significance of each estimated time series in the final prediction. Uniform weights are commonly considered as the default averaging strategy but more sophisticated strategies based on the process of member selection could be explored for performance evaluation. Figure 5.2 presents the structure of a voting regression model of N members [208,209].

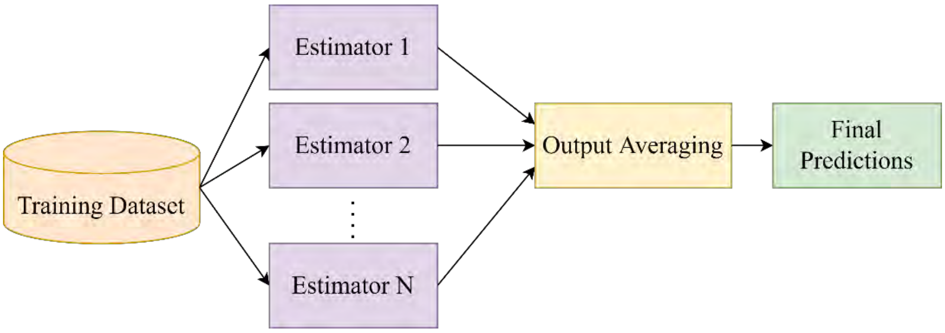


Figure 5.2: Voting regressor structure for the averaging of N estimators.

Stacking and voting ensembles could result in improved performance when compared to standalone estimators since the simultaneous reduction in bias and variance could derive estimated values that are closer to the actual values of the target variable. Additionally, the combination of ensemble members that are able to independently interpret linear or non-linear patterns leads to more robust estimators that could process more complex high-dimensional time series data efficiently. However, the performance benefits stemming from the implementation of an ensemble model are not guaranteed and it is commonly observed that the wrong or arbitrary selection of ensemble members leads to suboptimal performance. Therefore, studies that propose ensemble models based on arbitrary membership usually undergo extensive experimentation in order to verify the results. This work proposes a deterministic approach for member selection based on fundamental time series components, aiming to outperform the standalone base estimators on both ensemble approaches for clustered aggregate forecasting.

5.3.2 Cluster-Based Aggregate Forecasting Framework

Smart meter data processing is a challenging task in the development of load forecasting models since the dimensionality of the datasets and the plethora of different consumer types increase model complexity, resulting in a suboptimal prediction accuracy and convergence time for several centralized approaches relying on a single estimator structure. Therefore, cluster-based approaches attempt to divide the consumer base into groups based on distinct time series characteristics or geographical features in order to leverage trends within similar sets of consumers and reduce the noise by processing consumers with different load patterns separately. This work considers the cluster-based aggregate forecasting framework outlined in [193,200] since this method attempts to balance the effects of data compression from aggregation models and the fine-grained distributed prediction of clustered time series, resulting in a scalable strategy that could lower the forecasting error as the size of the consumer base increases. Firstly, load profile time series are clustered into k groups based on similarity distance metrics. It is important to note that the number of clusters affects the forecasting performance of the model since a suboptimal division of consumers could result in noisy and unbalanced datasets that could overfit or underfit the estimators. Therefore, cluster evaluation strategies such as the elbow method [210] and silhouette analysis [211] are often applied in this step, in order to determine the optimal value of k and ensure that the clusters are well-separated. Secondly, the load consumption time series in each cluster are aggregated in a single time series, resulting in drastically reduced dimensions and increased pattern regularity. The aggregated time series train k estimators that output aggregate load predictions for each cluster. Lastly, the summation of clustered predictions derives the total demand forecast and error metrics for model evaluation are calculated based on this time series. Figure 5.3 presents the cluster-based aggregate forecasting strategy.

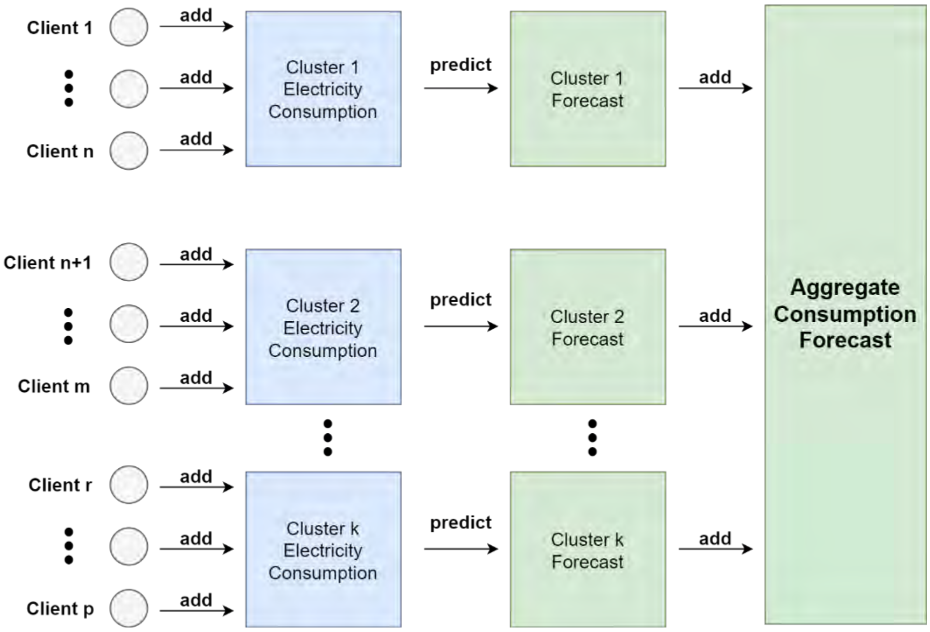

Client 1
Client n
Client n+1
Client m
Client r
Client p
add
Cluster 1 Electricity Consumption
Cluster 2 Electricity Consumption
Cluster k Electricity Consumption
predict
Cluster 1 Forecast
Cluster 2 Forecast
Cluster k Forecast
Aggregate Consumption Forecast

Figure 5.3: Cluster-based aggregate forecasting approach separating the consumer base into k clusters of variable sizes for the prediction of total electricity consumption.

5.3.3 Cluster Evaluation Methods

Clustering approaches in load forecasting such as the cluster-based aggregate framework utilize several evaluation methods in order to determine optimal data segmentation, resulting in groups of similar time series. The increased homogeneity of time series reinforces the presence of patterns in the aggregate data, reducing the noise that could be observed when load profiles of consumers exhibiting drastically different behaviors would be aggregated for the prediction of total demand. Additionally, energy applications based on the processing of load features as well as projects that utilize anonymous consumer data often face the challenge of separating the consumer set into distinct groups, since this would help the predictive performance of forecasting models, leading to meaningful deductions. Therefore, it is important to include some of the commonly used clustering evaluation methods in this project such as the elbow method and the silhouette method in order to properly divide the client base into clusters and potentially avoid irregularities in performance that could result in unstable error metrics.

The elbow method is an iterative process used for the selection of the optimal number of clusters through the search for the point where an increase in the number of clusters would not yield substantial data modeling benefits. This point is considered a threshold for clustering algorithms since the diminishing returns from the inclusion of additional clusters may not improve model performance. The commonly used metric in the elbow method is the sum of squared distances between the samples in each cluster and the cluster center. The value of this metric is calculated as the number of clusters increases and it is usually found that the sum of squared distances decreases in every iteration. The curve formed by those values is examined for the selection of the point after which the metric decreases slowly, exhibiting a linear pattern [212].

The silhouette method aims to quantify the cohesion as well as separation of samples by measuring the similarity of data points within the same cluster and the degree of dissociation of samples from other data points found in neighboring clusters. The silhouette coefficient is the metric calculated for the selection of the optimal number of clusters. Given the average distance of sample i to all other samples in the same cluster denoted as $a(i)$ and the average distance of sample i to all the points in the closest neighboring clusters, denoted as $b(i)$ the silhouette coefficient is computed with the following formula:

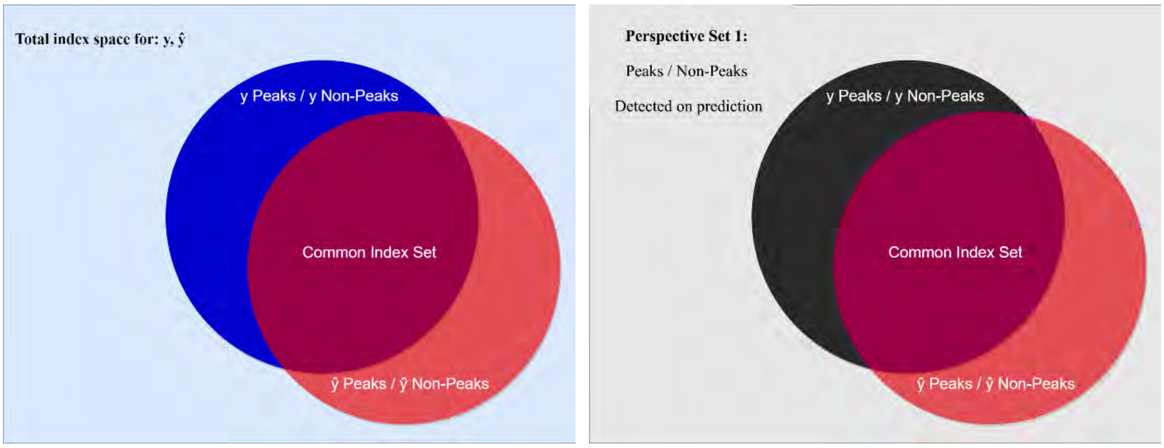
$$s(i) = \frac{b(i) - a(i)}{max\big(b(i), a(i)\big)} \tag{5.1}$$

The silhouette score derived from the averaging of the silhouette coefficient for each data point is utilized for the iterative analysis of each number of clusters. The computation and visualization of the silhouette score provide a robust cluster assessment, summarized in values ranging from −1 to 1. Positive silhouette scores closer to 1 indicate sufficient separation of samples into distinct and well-defined clusters. When the silhouette score is close to 0, the examined samples are usually close to the decision boundary between two neighboring clusters, denoting the ambiguity of the resulting data segmentation. Furthermore, negative silhouette scores closer to −1 often denote incorrect cluster assignment or the presence of outliers. Consequently, the optimal number of clusters corresponds to the number that resulted in the highest silhouette score [213]. However, it is worth noting that a thorough examination of model performance should consider both evaluation metrics and investigate the edge case of the minimum number of clusters when the silhouette score is positive since execution time and parallelization availability could be important factors in the deployment of forecasting applications.

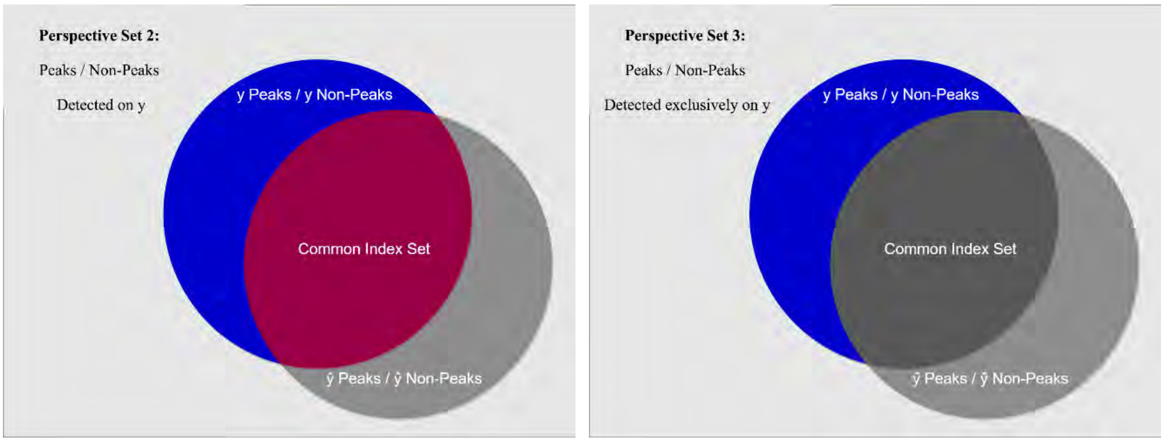
5.3.4 Proposed Forecasting Model

This study examines a combinatorial forecasting approach utilizing the cluster-based aggregate framework as the main structure for customer base segmentation and a model selection method for the development of flexible ensemble estimators that are able to efficiently derive total demand predictions. At the first step, the dataset containing client load profiles is clustered using the K-means algorithm [214] based on the dynamic time warping metric [215] in order to reinforce optimal time series similarity when client data are collected from different start periods. A silhouette analysis and the inertia-based elbow method were applied in order to determine the optimal number of clusters. Following the cluster-based aggregate framework, client time series within each cluster were aggregated to form the input dataset for the ensemble model. At the second step, ensemble membership is determined using peak and non-peak performance evaluation. A peak detection algorithm [216] is applied to the training set in order to detect local maxima by the comparison of neighboring values. A subdivision of the training set is used to train a set of estimators and evaluate their predictive potency on peak and non-peak indices.

The evaluation of peaks and non-peaks is quantified based on an error metric following the examination of three sets of indices denoting three different perspectives where peak and non-peak values are observed. The first set examines the performance of peak and non-peak indices as they were detected by the estimated time series for each candidate ensemble member, the second set examines the performance of peak and non-peak indices observed in the actual time series and the third set considers the performance of peaks and non-peaks detected exclusively in the actual time series. These sets of indices were selected based on the intuitive assumption that peaks and non-peaks should be detected from a relatively large and well-defined set of observations in order to derive robust performance metrics. Therefore, in the extreme case of poor time series estimation, the common index set for the actual and estimated load could result in a small sample that would provide insignificant insight towards the overall peak and non-peak performance of the candidate ensemble members. Additionally, when this extreme scenario is considered, the candidate estimator could be automatically eliminated since the potential inclusion of a prediction model that yields exceedingly poor performance in the ensemble model does not benefit the combinatorial approach. Similarly, uncertainty surrounds the consideration of peak and non-peak values detected exclusively for each estimator since this set may not share a strong connection to the actual time series and result in unreliable deductions. However, in the edge case where the candidate estimators perform extremely well and there is a great overlap of peak and non-peak positions between the actual and estimated time series due to the optimal match of the data points, the evaluation of the remaining indices exclusively detected in the actual time series is significant for the extraction of additional insights that could support informed decisions for model selection since the examination of this small region could be the deciding factor when multiple models are highly performant. The estimator scoring the lowest error metric for each perspective set is added to a list. consequently, lists of peak and non-peak influenced estimators are formed, including the most performant estimators for each case. Figure 5.4 presents the total observation space and highlights the sets of indices selected for this strategy. Furthermore, Figure 5.5 presents the derivation of the membership lists.



(a) (b)



(**c**) (**d**)

Figure 5.4**:** Set representation for the peak and non-peak model selection strategy based on training performance. (**a**) Total index space of peak and non-peak observations. The blue circle in the background denotes the total set of indices of peak and non-peak values for the actual load time series. The red circle in the foreground denotes the total set of indices of peak and non-peak values for the predicted load time series. (**b**) The highlighted red circle denotes the first evaluation set of peaks and non-peaks detected in each estimated time series. (**c**) Denotes the second evaluation set of peaks and non-peaks detected in the actual time series. (**d**) Evaluation set denoting the peaks and non-peaks exclusively detected in the actual time series.

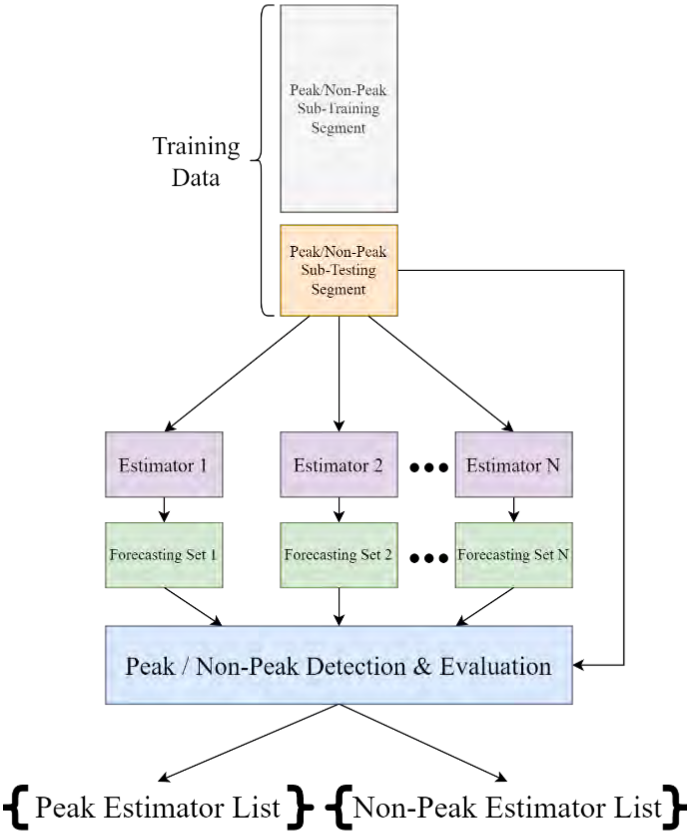

Training Data
Peak/Non-Peak Sub-Training Segment
Peak/Non-Peak Sub-Testing Segment
Estimator 1
Estimator 2
Estimator N
Forecasting Set 1
Forecasting Set 2
Forecasting Set N
Peak / Non-Peak Detection & Evaluation
{Peak Estimator List}
{Non-Peak Estimator List}

Figure 5.5: Ensemble membership selection process.

At the final step of this approach, ensemble regression structures adapted to each cluster due to the membership selection strategy, derive the predicted cluster load. The ensemble estimators could utilize the stacking or the voting paradigm in order to combine the output of the selected ensemble members. When a stacking ensemble is utilized, the lists derived from the member selection strategy could determine the base learners of the first level. Therefore, three models featuring the most performant estimators from sets of peaks, non-peaks and the joint set of indices can be evaluated. Figure 5.6 presents the structure of the stacking ensemble when the information from the sets of indices is available. Alternatively, the consideration of a voting ensemble could result in the development of more models since the member selection strategy could affect the base predictors as well as the weight strategy for the averaging of the estimated output. Consequently, six models could be examined in this case, since each of the previously mentioned sets of indices could follow a uniform or occurrence-based weight strategy. Figure 5.7 presents the structure of the voting ensemble models. Lastly, Figure 5.8 presents the process pipeline of this combinatorial forecasting approach.

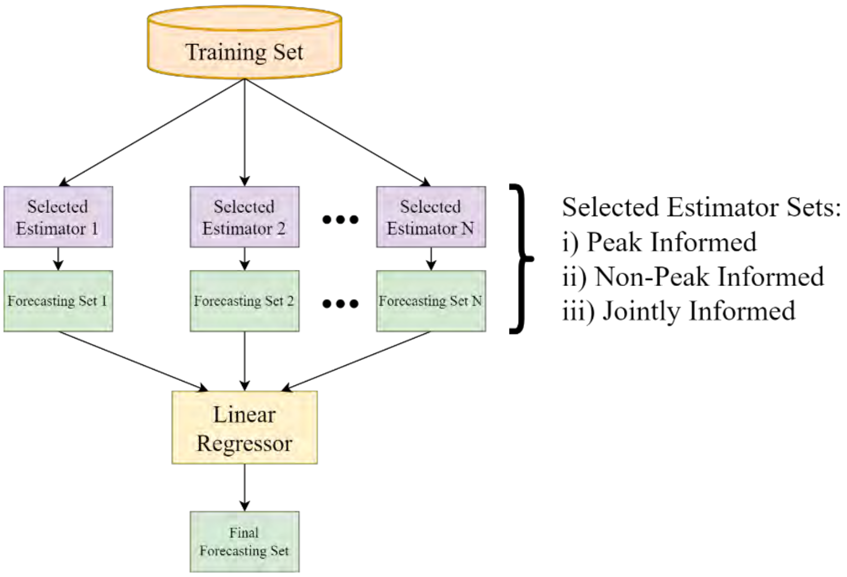


Figure 5.6: Stacking ensemble structure based on the peak and non-peak member selection strategy.

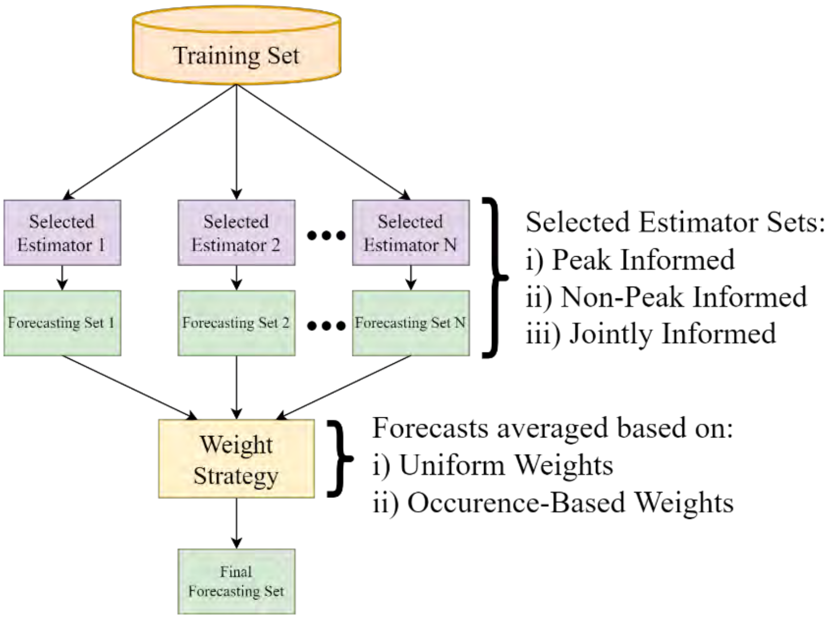


Figure 5.7: Voting ensemble structure based on the peak and non-peak member selection strategy.

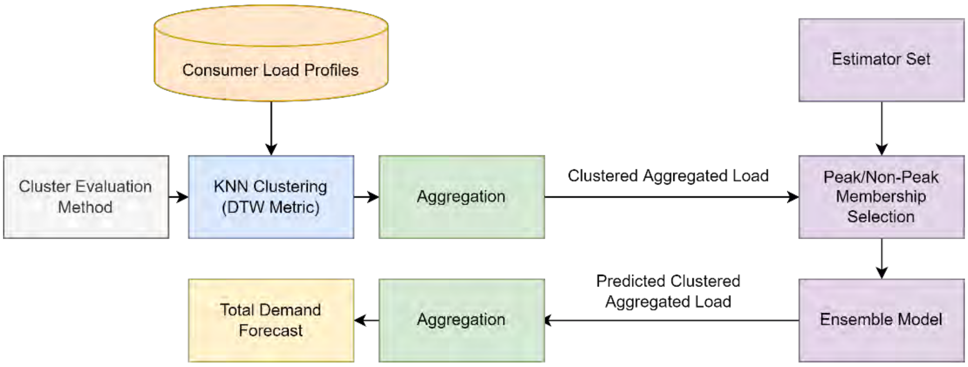

Consumer Load Profiles
Estimator Set
Cluster Evaluation Method
KNN Clustering (DTW Metric)
Aggregation
Clustered Aggregated Load
Peak/Non-Peak Membership Selection
Predicted Clustered Aggregated Load
Total Demand Forecast
Aggregation
Ensemble Model

Figure 5.8: Process pipeline for structural ensemble regression on the cluster-based aggregate framework.

5.3.5 Case Study

5.3.5.1 Data Overview

The proposed model was evaluated on a publicly available dataset [217] containing load measurements for 370 Portuguese clients extracted from smart meters in 15 min intervals for 4 years from January 2011 to December 2014, including a total of 140,256 observations. Since some clients were monitored after 2011, load measurements were considered as zeros. The dataset did not contain any missing values and client measurements were converted from kW to kWh for the purposes of this study. Additionally, the time labels follow the Portuguese time zone and at the start of daylight saving in March values between 1:00 a.m. and 2:00 a.m. are zeros. At the end of daylight saving in October, values between 1:00 a.m. and 2:00 a.m. aggregate the consumption of two hours. The load profiles included in this dataset belong to different types of clients such as industrial and residential, exhibiting different consumption patterns that could lead to the fine-grained classification of several subcategories. Since the dataset focuses solely on load features, the anonymity of clients is preserved. Consequently, the segmentation of the client base through clustering is important to the efficiency of the aggregate forecasting model since the processing of clients exhibiting similar consumption patterns could reduce the potential noise and contribute towards faster convergence during training.

The inspection of total demand in Figure 5.9 as well as the yearly boxplot presented in Figure 5.10 show that the aggregation of different consumer types coupled with the difference in monitoring periods result in peaks and valleys that could be difficult to interpret in short-term and very short-term prediction horizons. These effects become less impactful as the number of actively monitored consumers remains the same and the forecasting horizon is extended since seasonal patterns can be easily discovered. However, the requirement of a static consumer set in the modern power grid would be unrealistic due to the continuous expansion of the client base as well as the increased diversity in client behavior. Therefore, the examination of very short-term forecasting tasks for the prediction of aggregate load through frameworks that aim to address these challenges could lead to the implementation of more robust design strategies.

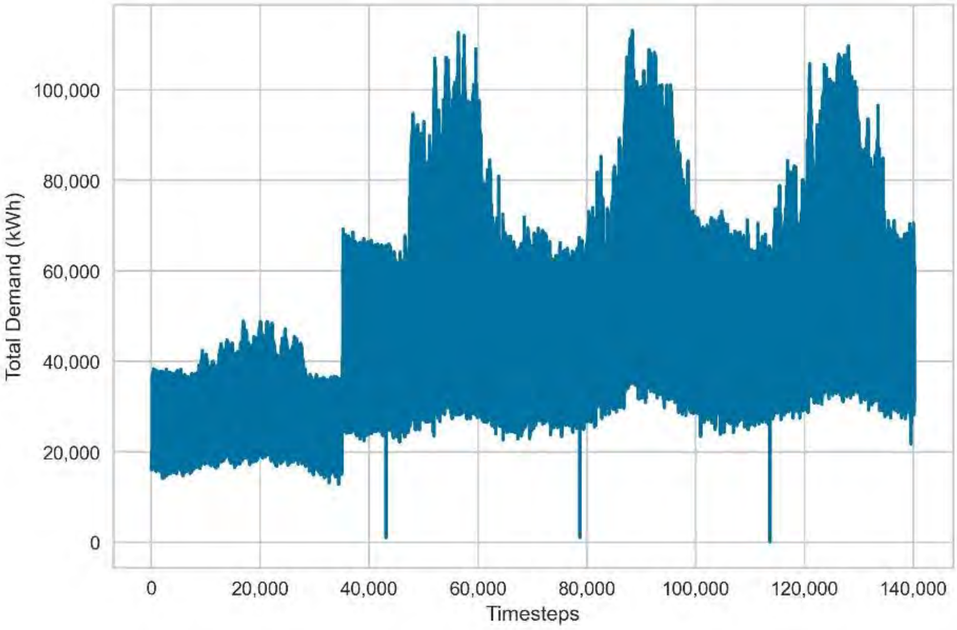


Figure 5.9: Total electricity demand in kWh for 370 clients.

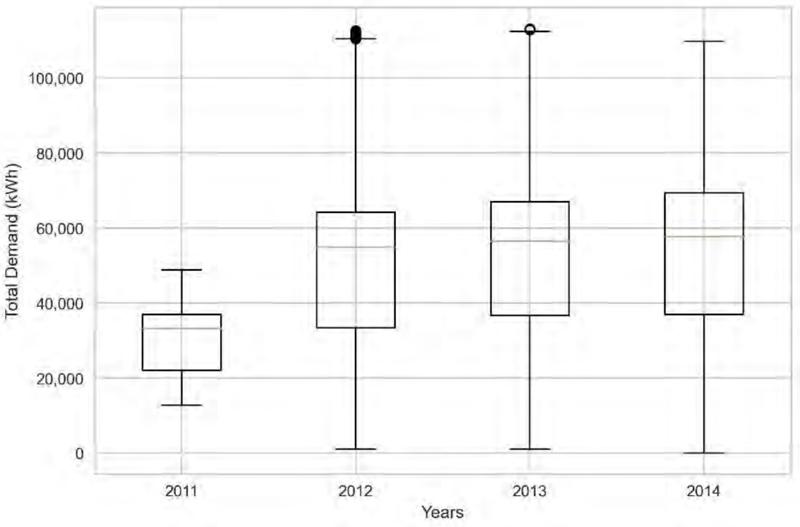
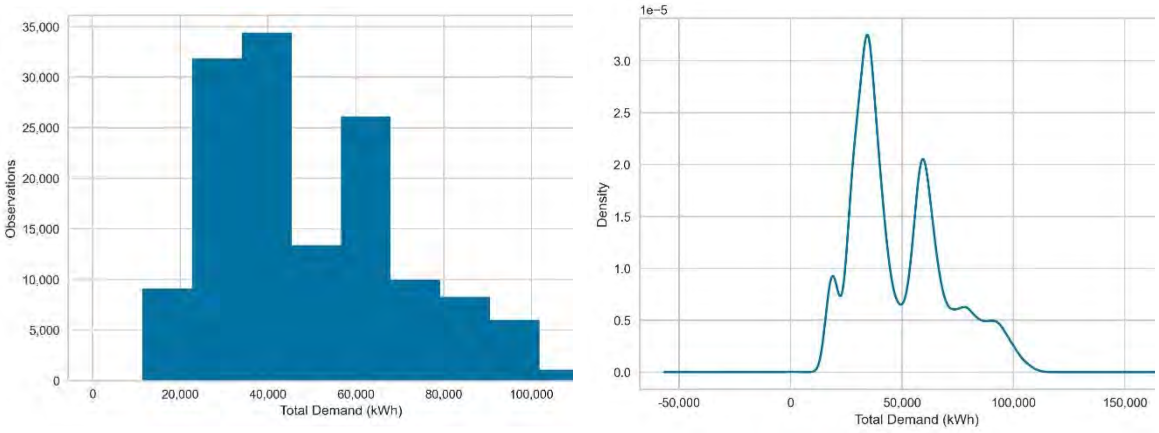


Figure 5.10: Boxplot of total yearly electricity demand.

The examination of the histogram and the inspection of the density plot for the aggregate load in Figure 5.11 indicate a bimodal distribution that could be interpreted as the broader classification of clients into residential and industrial groups. Alternatively, this distribution could denote the sinusoidal shape of observations as the number of actively monitored clients becomes more stable. The presence of several peaks in the density plot and the general imbalance of samples in the bins of the histogram could confirm that this is a challenging task for some linear forecasting models that assume a Gaussian distribution. The implementation of a clustering algorithm could lead to more easily interpretable data distributions, resulting in the accurate prediction of partial aggregate load. However, the selection of estimators for the prediction of the partial aggregate load should not be arbitrary due to irregular data distributions that might persist after the clustering step. Therefore, the adoption of membership strategies is important for the development of combinatorial forecasting approaches; additionally, the utilization of fundamental methods tied to the data distribution such as peak detection could be useful in the refinement of error metrics.



(**a**) (**b**)

Figure 5.11**:** Non-temporal representation of total electricity demand samples. (**a**) Total electricity demand histogram. (**b**) Density plot denoting bimodal distribution.

Lastly, the observation of the first order lag plot denotes relatively high autocorrelation due to the high concentration of samples on the diagonal. Figure 5.12 suggests a positive correlation between the time series $y(t+1)$ and the lag $y(t)$ due to the positive slope of the line formed in the graph. Therefore, autoregressive approaches could be suitable for the prediction of total electricity demand in a very short-term time horizon since most data points are densely concentrated in this linear shape. This could be useful information in research projects that primarily include load features as proof of concept or due to data availability issues.

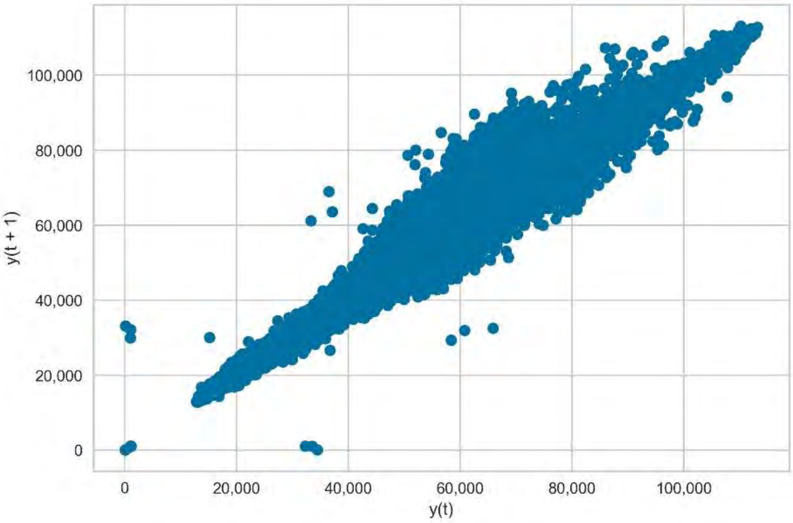


Figure 5.12**:** First order lag plot comparing total electricity demand time series $y(t+1)$ to the lagged total demand $y(t)$.

5.3.5.2 Implementation and Experiments

The forecasting problem examined in this study can be formulated as the point prediction of total demand for the next 15 min interval based on 4 lagged observations for the previous hour in order to define a simple and interpretable supervised learning task. This task minimizes the impact of feature engineering and data preprocessing on the overall performance of estimators and could allow us to focus on the performance of the cluster-based structure through the dedicated usage of load features as the membership of estimators changes for each cluster. The training set contained 80% of observations and the test set 20%, following common practice for similar forecasting tasks. Since the model focuses on very short-term predictions, the execution time for clustering needs to be fast in order to reserve time for the tuning and recalibration of the ensemble predictors at later steps. Consequently, monthly down-sampled load profiles were considered for the assignment of clients into clusters. As a result, the clustering procedure could be executed in seconds instead of several minutes when compared to weekly and daily down-sampling. Furthermore, the utilization of K-means clustering based on dynamic time warping was beneficial to the optimal alignment of the time series since some clients were monitored after 2011 and prior data entries were zeros. Following this step, silhouette and elbow methods were utilized for the selection of the optimal number of clusters. The silhouette score is the main metric examined in the silhouette method and ranges from −1 to 1, denoting poor cluster assignment when the clustering method achieves a negative silhouette score and satisfactory data separation when the value of that score is positive. The silhouette analysis showed that the assignment of clients into clusters ranging from 2 to 10 resulted in acceptable data separation since the silhouette scores were above 0.6, verifying the consensus of selecting an optimal number of clusters that falls within this range and reaching a global maximum at $k = 2$. The elbow method based on the sum of squared distances of the samples to their closest cluster center denotes that the selection of a number of clusters higher than 7 for the assignment of clients would not yield significant data modeling benefits since after that point, inertia decreases linearly at a slow rate. We observed that clustering derived from other candidate elbow points such as $k = 6$ and $k = 8$ did not yield a significant difference in terms of error metrics in this forecasting task when compared to $k = 7$. However, the significantly lower silhouette score of $k = 8$ could indicate data separation issues, discouraging the selection of this value for the elbow method and reinforcing the selection of $k = 7$ since this is a pivotal point after which a significant decrease in the silhouette score occurs when this region of the inertia curve is examined. Therefore, the performance of the ensemble learning models was examined in the representative points of each method for 2 and 7 clusters, respectively. Figures 5.13 and 5.14 present the clustering evaluation of the silhouette and elbow methods, respectively.

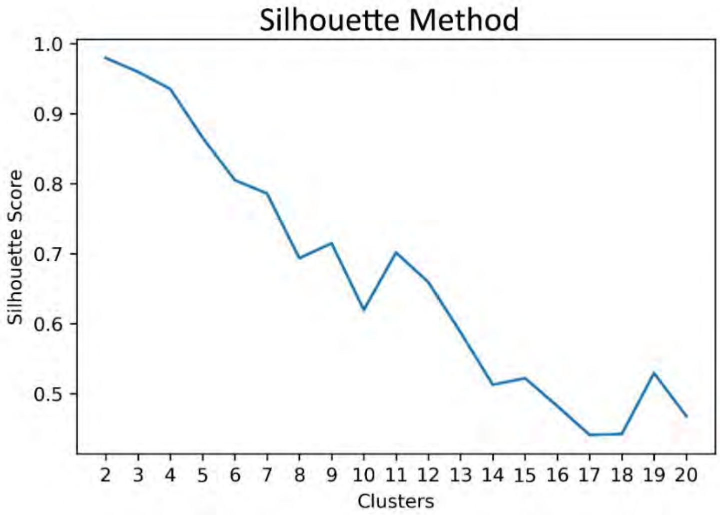


Figure 5.13: Silhouette method for clusters ranging from 2 to 20 using K-means clustering of client load profiles based on dynamic time warping.

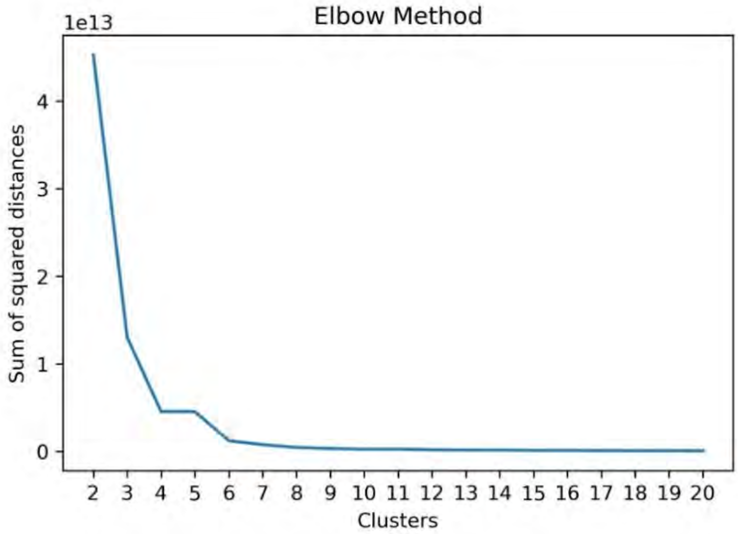


Figure 5.14: Inertia-based elbow method for clusters ranging from 2 to 20 using K-means clustering of client load profiles based on dynamic time warping.

The next process of this forecasting model considers a set of 11 base regression estimators as candidate members of the ensemble structure for each clustered aggregated load. The estimators were tuned based on cross-validated random search [218] on the training data, utilizing a 5-fold time series split that returns the first $f - 1$ folds as the training set and the last fold f as the evaluation set, maintaining the temporal order of observations. Consequently, the models were configured in a way that reflects the average performance of the best selected hyperparameters. Table 5.1 presents the methods utilized for our experiments as well as their respective parameters based on the implementations found on scikit-learn and xgboost packages [219,220].

Table 5.1: Base estimators and hyperparameters.

Model	**Hyperparameters**
XGBoost	learning rate, maximum depth, minimum child weight, number of estimators, columns sampled by tree
Linear Regression	-
Linear SVR	tolerance, regularization parameter C
SGD	learning rate, initial learning rate, alpha regularization strength, maximum iterations, loss, tolerance, penalty parameter for regularizer selection
Huber Regression	maximum iterations, alpha regularization parameter, epsilon outlier resilience, tolerance
LARS	non-zero coefficients
Lasso	maximum iterations, alpha regularization parameter, tolerance
Ridge	alpha regularization parameter
Theil-Sen Regression	-
Bayesian Ridge	lambda weight precision, alpha noise precision

K-Neighbors Regression	leaf size, number of neighbors, power parameter of Minkowski metric

The training set was segmented for the peak and non-peak detection and evaluation following the previously mentioned time series split and mean absolute error was selected as the error metric for the quantification of predictive potency since the dataset contains zeros for the time periods where some clients were not monitored. After the examination of three different perspectives corresponding to three different sets of indices as described in the previous subsection, three lists of estimators were formed for each cluster. The first list contained the three most performant estimators on the sets of observations related to peak indices, the second list included the most performant estimators on the sets of observations associated with non-peak indices and the third list was the concatenated list of the previous two after the removal of duplicate estimator entries. Three stacking ensemble models were developed based on the estimators of each list with linear regression being the second level estimator. Additionally, six voting regression models were developed, featuring uniform and occurrence-based weighting strategies based on the concatenated membership list. All models featuring the tuned ensemble members were trained on the full training set of observations and evaluated on the holdout test set.

This project was developed in Python 3.8.8 using the packages pandas 1.2.3, numpy 1.21.5 and scipy 1.7.3 for data processing, tslearn 0.5.2 for clustering, scikit-learn 1.0.2 and xgboost 1.3.3 for predictive modeling and matplotlib 3.5.1 for visualization. The model implementation and the experiments were executed on a desktop computer with an AMD Ryzen 1700X processor, 8 gigabytes of RAM, and an NVIDIA 1080Ti graphics processor. Additionally, the code of this forecasting approach and case study is publicly available on GitHub [221].

5.3.6 Performance Metrics

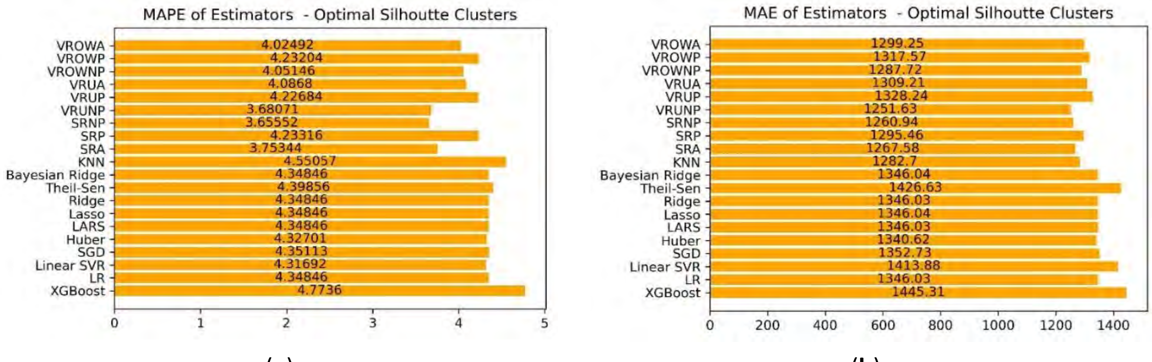
In this section, we outline the main performance metrics utilized for the evaluation of all nine ensemble estimators in the cluster-based aggregate framework. Firstly, MAE [87] is utilized for the peak and non-peak influenced member selection as well as the final ensemble evaluation since it is a common and simple loss function that measures the average error of continuous variables without considering error direction.

Furthermore, MAPE [88] is included for the generalized measurement of relative error since it is an interpretable scale-independent metric. The usage of MAPE is restricted to the evaluation of the total demand for the final ensemble models due to the existence of zeros in some of the clustered time series. Secondly, MSE [222] and RMSE [223] are included as quadratic scale-dependent loss functions that could denote the impact of large errors since errors are squared before they are averaged. Additionally, the simultaneous examination of MAE and RMSE could determine the variation of errors for the ensemble models since a large difference between the values of those metrics could denote great variance in the individual errors of the test sample, indicating the occurrence of large errors.

5.4 Results

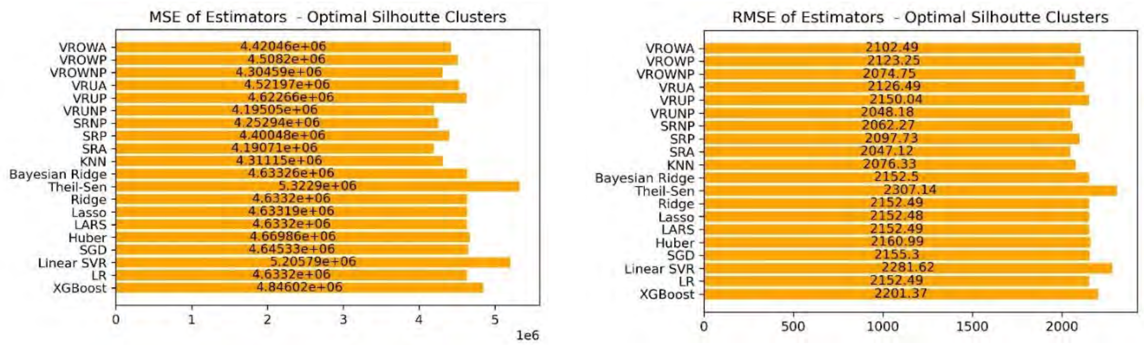
In this section, we analyze the performance of the ensemble models by providing an overview of the error metrics based on the data available in this case study. Since this project focuses on the implementation of a deterministic membership selection technique on stacking and voting ensembles, all nine ensemble estimators discussed in the experiments presented earlier are compared to the standalone estimators in the cluster-based aggregate framework in order to distinguish the most efficient ensemble structures and outline the potential performance benefits of this approach. The main motivation for the development and subsequent comparison of those models stems from the uncertainty that some values could introduce during the training of estimators, resulting in regions where suboptimal fitting could occur. Intuitively, unstable estimator performance could be observed in regions where local maxima could be detected due to the sudden change in the value of electricity consumption or due to the irregularity of the consumption pattern, resulting in large errors. Therefore, the prioritization of points or regions where peaks are not observed would be considered as a safer starting point for the fair performance comparison of base learners and the examination of optimization benefits through the combination of multiple estimated time series. Since the discovery of base learner combinations that reduce the forecasting error in a given machine learning task is a challenging process and a given ensemble structure does not guarantee improved performance when applied to different datasets, adaptive ensembles could result in more robust estimation and the examination of fundamental time series characteristics such as peak and non-peak points could lead to flexible ensemble structures that yield performance benefits when diverse time series are processed, such as the clustered load of different client types. The performance comparison includes the computation of MAPE, MAE, MSE and RMSE for all models. The stacking ensembles utilizing the list of best peak estimators, the list of best non-peak estimators and the merged list containing a single instance of all members from both lists, are labeled as SRP, SRNP and SRA, respectively. Similarly, the voting ensemble structures featuring a uniform weight strategy are labeled as VRUP, VRUNP and VRUA. Lastly, the voting ensemble models featuring an occurrence-based weight strategy derived from the frequencies of estimators in the merged list before duplicate removal are labeled as VROWP, VROWNP and VROWA, respectively.

Figure 5.15 presents the error metrics of the standalone models as well as the ensemble structures on the optimal assignment of clients into two clusters based on the silhouette analysis. The examination of MAPE and MSE shows that the ensemble methods following this membership selection strategy yielded improved forecasting performance when compared to the standalone estimators. Additionally, the simultaneous examination of MAE and RMSE indicates that there is a small variation in the magnitude of the errors in standalone models and each ensemble structure but the occurrence of large errors is unlikely. The stacking and voting regressors utilizing the membership list derived from performant non-peak estimators yielded the most distinct improvement and relatively smaller benefits can be observed from the ensembles based on peak membership. Furthermore, the implementation of a uniform and occurrence-based weight strategy resulted in similar forecasting performances for voting ensembles that utilized the peak as well as the merged membership lists. However, a more substantial difference in error metrics can be observed in the comparison of the voting estimators utilizing the non-peak membership list, where uniform weights resulted in lower metrics.



(a)

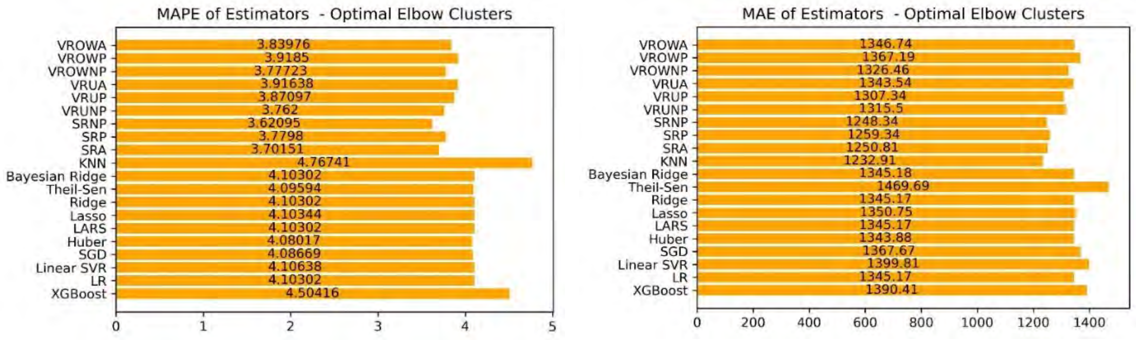
(b)



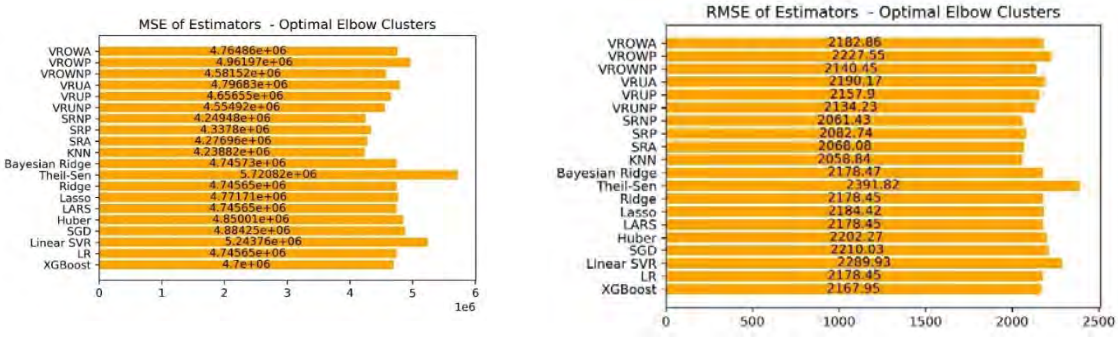
(**c**) (**d**)

Figure 5.15**:** Error metric comparison for standalone estimators and structural ensemble models given the optimal clustering denoted by the silhouette method. The panels present the following metrics: (**a**) Mean absolute percentage error. (**b**) Mean absolute error. (**c**) Mean squared error. (**d**) Root mean squared error.

Figure 5.16 provides an overview of the error metrics derived from the inertia-based elbow method for optimal clustering. Similar to the examination of the silhouette optimal cluster selection, it is evident that the stacking and voting ensembles based on the non-peak membership list yield improved performance in this forecasting task, resulting in lower MAPE values. The values of MAE, MSE and RMSE for those models remain close to the lowest value of the KNN regressor, denoting the overall stability of the ensemble models. However, this observation does not hold true for all ensemble models since voting ensembles following an occurrence-based weight strategy yielded MAE, MSE and RMSE values closer to the average standalone predictors while yielding a smaller improvement of MAPE, denoting fewer substantial benefits derived from the model fusion in this case.



(**a**) (**b**)



(**c**) (**d**)

Figure 5.16**:** Error metric comparison for standalone estimators and structural ensemble models given the optimal clustering denoted by the elbow method. The panels present the following metrics: (**a**) Mean absolute percentage error; (**b**) Mean absolute error; (**c**) Mean squared error; (**d**) Root mean squared error.

Consequently, the inspection of both optimal clustering strategies shows that the implementation of flexible ensemble models in the cluster-based framework could improve the overall load forecasting performance when considering ensemble members that performed well on the prediction of non-peak observations during training. This deduction partly verifies the intuitive assumption that regions with sudden peaks in the clustered data may introduce a level of uncertainty which could result in unstable estimator behavior, leading to the unfair performance evaluation of base learners for membership selection. The uniformly weighted voting regressor based on non-peak influenced membership achieved, approximately, a 16.5% improvement over the average MAPE value of standalone estimators while utilizing the silhouette analysis for optimal clustering. Similarly, the stacking non-peak influenced regressor achieved a 17.2% improvement in the experiment. Furthermore, the experiment utilizing the elbow method for the selection of the optimal number of clusters showed that the previously examined models yielded a 10.4% and 13.8% MAPE improvement over the average of the standalone values, respectively. It is worth noting that in this second experiment the stacking regressor considering the merged list of peak and non-peak influenced membership yielded an 11.9% MAPE improvement, showing slightly better performance when compared to the VRUNP model. The examination of those metrics denotes an overall reduction in MAPE, comparable to the average reduction observed in the implementation of ensemble learning for short-term forecasting over different sets of estimators in recent research results presented in [186] as well as [224,225]. Since the successful implementation of an ensemble model typically yields a small improvement when compared to the best base estimator, a similar behavior can be observed in our study, achieving approximately the same level of error metric reduction when compared to relevant studies. The main difference highlighted in our approach is related to the discovery and examination of optimal base estimator sets from a wider estimator space in an attempt at eliminating the uncertainty of the initial ensemble member selection process. Therefore, our work aims to shift the focus from the individual proposal of specific ensemble structures to member selection strategies that generate appropriate sets of estimators for the training of a given time series.

5.5 Discussion

This research project examined the performance of structurally flexible ensemble estimators on the cluster-based aggregate framework for the improvement of short-term total demand predictions. The proposed approach implemented a membership selection strategy focusing on the evaluation of peak and non-peak data points given different perspectives that consider sets of observations on the actual as well as the estimated time series derived from segments of the training set. This process resulted in the development of nine ensemble models consisting of three stacking and six voting regression structures that covered several ensemble member combinations. Consequently, a case study was carried out for the evaluation of those models on a dataset including the load profiles of 370 clients. The research findings indicated that the ensemble models were able to improve the forecasting accuracy for clustered load estimation, resulting in more robust combinatorial structures. The experiments showed that voting and stacking ensembles influenced by the membership set of non-peak performant base learners could provide more significant forecasting improvements, yielding MAPE scores of 3.68 and 3.65, respectively, when silhouette analysis is used for optimal clustering. Similarly, those models achieved MAPE scores of 3.76 and 3.62, respectively, when an inertia-based elbow method was utilized for optimal clustering and the stacking ensemble including peak as well as non-peak performant base learners resulted in adequate performance, achieving a MAPE value of 3.7.

Since the discovery of efficient base learner combinations is not a straightforward process and one specific ensemble structure may not guarantee the reduction in error in a given forecasting task, we believe that this adaptive approach contributes towards the deterministic member selection through the inspection of fundamental time series characteristics. Additionally, it is evident that a standalone estimator may not perform well when processing time series that exhibit different patterns, resulting in unstable overall metrics for the aggregate values. The average performance of some robust and optimally tuned standalone estimators could be drastically affected by the input data as well as the data collection process. Different electricity consumer types and various data collection characteristics such as the start of the load monitoring period could impact the prediction accuracy and the recalibration process of the forecasting models. Consequently, it could be observed that some estimators may outperform others with minimal context related to the justification of the difference in performance, leading to less interpretable implementations that follow arbitrary model selection processes. Therefore, the main advantage of this proposed approach is the efficient combination of base learners through a simple and well-defined process that could be seamlessly integrated in ensemble regression tasks for the energy sector. The performance hinderances introduced by the extreme cases where the response of a standalone estimator yields irregularly high error metrics on certain data points are diminished through the consideration of multiple estimated time series. Moreover, the focus is shifted towards the inspection of data points where the estimators are expected to perform optimally, reinforcing the fairness of comparison and setting additional criteria towards member selection in ensemble learning.

On the other side of the spectrum, there are a few disadvantages in the application of this method that should be mentioned for completeness. Since cluster-based frameworks often lead to computationally expensive models, the integration of flexible ensemble learners in this paradigm could increase the computational cost due to the training and processing of multiple estimators. Therefore, the complexity of each candidate base learner could be restricted since the tuning, training and processing of several deep neural network architectures and hybrid structures would increase the execution time substantially due to the increased number of hyperparameters as well as the overall latency encountered when loading and storing data during training, rendering them inefficient for short-term forecasting tasks and real time applications. However, advances in distributed computing could remedy this issue through the parallelization of data processing tasks. It is evident that the proposed approach could be implemented in multi-threaded distributed systems since there is a clear distinction between standalone and aggregate tasks. Consequently, the inspection of each base learner and the membership evaluation process for each cluster could be easily parallelized, resulting in a scalable hybrid structure.

Future research projects could explore different time series characteristics and combine them in order to extend the current membership evaluation strategy, resulting in the discovery of additional ensemble structures. Since this study primarily focused on load features, isolating their impact for the inspection of base learners in an environment containing only the load profiles from different types of customers anonymously, the inspection of time series elements derived from different types of features could provide significant insights towards the development of more robust ensemble estimators, depending on data availability. Furthermore, the proposed strategy could be applied to multiple unclustered time series or load profiles processed in different clustering or aggregation frameworks in order to examine the performance of adaptive peak and non-peak ensemble learning through more diverse experiments. Lastly, the impact of several vital parameters to the definition of the forecasting tasks could be studied, such as the forecasting horizon, and the customer base size could be studied in an attempt at quantifying the scalability of this approach in different client groups as well as the versatility of the method.

5.6 Concluding Remarks

The intricacies of very short-term total electricity demand forecasting tasks add a layer of ambiguity to combinatorial modeling since the challenges derived from increasingly diverse and rapidly growing client groups could hinder the efficiency of robust estimators. Additionally, the inclusion of estimators in hybrid and combinatorial approaches is often influenced by expert knowledge and general performance indicators in similar forecasting tasks. Therefore, the criteria for the selection of base estimators are not explicitly linked to the shape and the individual characteristics of a given dataset, resulting in a seemingly arbitrary estimator selection process. This phenomenon could be easily observed in the implementation of ensemble learning models where small refinements to the error metrics are expected when several estimators are optimally combined. Since this optimal combination differs depending on the available data and the problem formulation, this performance boost is not guaranteed and is often derived from extensive experimentation.

In this project, we presented an estimator selection strategy that generates base estimator sets capable of achieving this performance boost through the examination of peak and non-peak observations from multiple evaluation perspectives during training. This membership strategy aims to adapt to different shapes of time series and output estimator groups that outperform the standalone estimators when combined in a stacking or voting ensemble structure. The case study presented in this work focused on the effect of load features and utilized the cluster-based aggregate framework since the clustered time series derived from a diverse set of clients monitored from different starting points would introduce a degree of unpredictability between consecutive samples that would intuitively cause certain models to underperform as the shape of the time series could be drastically different between clusters. As a result, three stacking models and six voting models were evaluated on a group of clustered time series for the prediction of total demand based on the most important numbers of clusters derived from the silhouette and elbow methods. Through our experiments, we observed that base estimator sets generated from the proposed strategy led to consistently more performant ensemble models when the criteria influencing the selection of estimators involved the examination of non-peak observations. It is worth noting that in some ensemble structures the merged set of estimators selected from the examination of peak and non-peak observations performed adequately well. In conclusion, this work attempts to reinforce the basis of ensemble and hybrid modeling through a well-defined and easily interpretable criteria-based approach which is tuned based on the input time series in order to boost predictive performance.

Chapter 6 Development and Application of a Meta-Modeling Architecture Towards Estimation Stability and Generalization

6.1 Motivation

The extension of the forecasting framework through the inclusion of meta-modeling processes adds significant value to short-term forecasting in the energy sector since it enables the development of more robust estimators that could drastically improve the generalization capabilities and the stability of predictions within the studied horizon. The main appeal of meta-modeling approaches stems from the enhanced flexibility in the development of the primary forecasting module. Since the role of the meta-modeling approaches is to combine and process output data from the main forecasting module for the derivation of the estimated target variables, the forecasting module could be restructured and repurposed in order to capture and analyze different aspects of the time series features. Consequently, several meta-features or alternative interpretations of the target variables could be derived in order to form more compact datasets that comprehensively explain the patterns of the target series.

The application of meta-modeling approaches could be useful in more complex environments where the efficient feature dynamics could improve performance and the omission of influential factors could introduce performance hinderances. One of the most prominent scenarios where meta-modeling techniques could be impactful is the research of short-term consumer load forecasts. It is evident that the prediction of consumer load curves could be sufficiently accurate when enough influential factors are examined and when the input dataset is derived from a high-quality data collection process. However, in real-world applications, these requirements may not always be fulfilled, resulting in erroneous and unreliable predictions. Therefore, the forecasting model could benefit from additional learning stages that expose more characteristics of the time series during training. Meta-modeling techniques could be applied in this scenario, in order to introduce a multi-stage pipeline that initially shifts the focus towards the discovery of compact data representations. In this scenario, data representations that address community dynamics are not sufficiently covered through research efforts focusing on meta-modeling. The impact of time series similarity is typically helpful in the early stages of the forecasting pipeline for the purposes of feature selection. Similar consumption patterns from different clients could reinforce the generalization of the model and explain irregular time series alignment issues that may occur during data collection. Furthermore, the examination of causality could be used to further explore the influence that other time series might have in the prediction of the target, denoting a more general measure of the degree of predictive potency that is not severely penalizing data abnormalities. Both influential factors are not sufficiently examined within the scope of meta-modeling design for the exploration of community dynamics that may affect individual load forecasts. This research gap leads to the underutilization of available knowledge, since the role of a consumer within a community and data that could indirectly be related to a specific load curve may have significant value to the overall performance of the forecasting pipeline.

The contribution analyzed in this chapter attempts to address this research gap through the design and development of a novel meta-modeling approach that considers the factors of similarity and causality in order to derive alternative target time series components that could be combined in order to boost accuracy and stability of the target prediction within short time intervals. This case study considers real-world consumer data where poor data collection quality could result in higher error metrics, rendering the convergence of regression models more difficult. The following sections correspond to the introduction, methodology, results and discussion of this meta-modeling approach.

6.2 Introduction to a Meta-Modeling Power Consumption Forecasting Approach Combining Client Similarity and Causality

Data analysis and forecasting models are the cornerstones of research in the energy sector since they enable the development of sophisticated applications and strategies that optimize the flow of energy on the grid and improve the quality of life of electricity consumers. Modern data-driven approaches rely on the collection and processing of client information regarding their power consumption, socio-demographic features, and various external factors, such as weather variables, in order to examine consumption patterns and make accurate predictions. Forecasting models focused on the prediction of power consumption provide meaningful insights that can be utilized by electricity providers in order to monitor and control the demand efficiently, while being able to detect and avoid irregular events. On a larger scale, power forecasting models allow the providers to construct load profiles of buildings for months in advance and, based on that estimate, to validate energy meter readings and cluster buildings into groups, contributing towards more intelligent regional planning approaches. Additionally, smart pricing strategies can be implemented in an attempt to adjust electricity tariffs dynamically, based on client behavior. Price forecasting techniques complement electricity consumption models in that spectrum, with significant contributions towards the efficient execution of regression tasks [226–228]. Furthermore, electricity forecasts can benefit each consumer individually due to the development of applications that allow clients to monitor and reschedule their daily tasks flexibly in order to gain additional control over the billing process. Consequently, there is a growing interest for the development of accurate and robust power forecasting models that are able to extract useful information from the underlying patterns and relationships of the collected energy data [229–231].

Energy data used in the design of forecasting models is commonly structured in the form of time series, where records consist of relevant features indexed in time order. Classical time series forecasting methods such as autoregression (AR) [232], moving-average (MA) [233], and autoregressive moving average (ARMA) are often used to predict the next time step in a univariate sequence modeled as a linear function of information extracted from previous time steps. Moreover, the autoregressive integrated moving average (ARIMA) method and its extensions [234] utilize differencing in the observations of previous time steps. Vector autoregression (VAR) models constitute a generalization of AR models since they support multivariate time series. A similar generalization is observed in vector autoregression moving-average models (VARMA) [235]. Additionally, simple exponential smoothing (SES) [236] and Holt Winter’s exponential smoothing (HWES) [58] model the next time step as an exponentially weighted linear function of past observations. Traditional methods, such as AR, MA, ARMA, VAR, VARMA, and SES, usually do not utilize the trends and seasonal patterns of the input sequence and, while their extensions and variants can integrate those elements to construct more sophisticated models, there are more limitations associated with those methods. The limitations of those statistical methods mainly revolve around the structure of the available data, the relationship between input and output variables, and the ability to support highly dimensional time series [237,238]. Since classical forecasting models do not support missing values, data imputation techniques need to be implemented, thus altering the original dataset. Furthermore, traditional models often generate predictions based on the assumption of a linear relationship between the input and output variables, thereby omitting the more complex non-linear patterns and trends. Lastly, these classical methods are observed to be more suitable on univariate sequences in terms of performance, rendering the design and generalization process for more complex environments more difficult. It is clear that due to the complexity, availability, and structure of many energy datasets, most traditional approaches would not suffice for the derivation of accurate predictions.

On the other side of the spectrum, advances in artificial intelligence and machine learning led to the development of more robust models, which are capable of discovering complex relationships between input and output features. Many different architectures involving neural networks, such as the multilayer perceptron (MLP) [140] and long short-term memory (LSTM) network [239], were used successfully in many time series forecasting tasks, achieving impressive performance [185]. These neural network models follow a black-box approach in the approximation of nonlinear functions. The multilayer perceptron finds frequent application in regression, classification, and fitness approximation tasks with an emphasis on learning to map the set of inputs to the set of outputs. Long short-term memory networks take advantage of the temporal data characteristics in order to extract insights from the order dependencies that could be present in a sequence. The suitability of machine learning methods for energy data processing is evident since these models are able to capture more complex patterns from highly dimensional data without the requirement of having an optimally structured dataset. However, there are still many challenges that limit the performance of these models, and ongoing research attempts to address them. The lack of data needed to train a model successfully in combination with potentially missing values could hinder the performance of neural networks due to overfitting [240], since the model would not have an adequate number of training examples in order to perform well when new data is tested. Additionally, feature engineering is crucial in the design of a machine learning model, since the inclusion or exclusion of certain variables and data transformations can have a great impact in the learning process. Therefore, some forecasting tasks in the energy sector can have poor performance due to a suboptimal data collection process or limited data availability given the forecasting horizon and the expected output. Ongoing research in the field focuses on the introduction of novel methods and hybrid models that utilize a combination of feature engineering techniques, architectural changes, and mechanisms that optimize the training process, thus rendering neural network models more resilient to data abnormalities. Additionally, there is an ongoing effort towards creating more well-structured datasets, while minimizing data distortion and noise for energy forecasts [241,242].

There are several recent projects addressing forecasting and classification tasks with the use of data-driven methods that often utilize neural networks and feature engineering techniques. Choi and Lee [243] proposed a framework based on an LSTM ensemble and a weighted combination of predictions for time series forecasting, showing that combinatorial approaches that utilize the output of multiple neural networks can achieve better performance compared to other popular forecasting methods. Tian et al. [244] presented a hybrid architecture based on the combination of a LSTM and a convolutional neural network (CNN) for short term load forecasting, improving prediction stability for that forecasting horizon. Mujeeb et al. [245] used a deep LSTM network to create a new load forecasting scheme for big data in smart cities, showing the capabilities of deep neural networks on highly dimensional historic load and price data. Markovič et al. [246] reinforced the importance of optimal data aggregation by presenting a data-driven method for the classification of energy consumption patterns based on functional connectivity networks. Jin et al. [247] proposed an encoder-decoder model utilizing an attention mechanism in order to learn long data dependencies from the input sequence efficiently. Tian et al. [248] developed a forecasting model based on transfer learning, using the outputs of already trained models for the estimation of building consumption according to similarity measures. This project provided substantial motivation towards research on meta-modeling techniques that could improve the accuracy of the predicted smart meter readings. Chen et al. [249] proposed a time series forecasting model that explored the impact of Granger causality for stock index predictions. This work presented interesting ideas on the use of causality in prediction models and could be extended to the field of energy forecasts. Boersma [250] studied the correlation and impact of internal and external factors on the prediction of household consumption using an MLP network. This project highlighted the importance of feature engineering as well as time resolution in the derivation of accurate predictions. Emamian et al. [251] implemented a solar power forecasting model using an LSTM ensemble to aggregate the predicted output of each network, demonstrating that ensemble models can achieve higher accuracy and more reliable results than single neural network models. Guo et al. [252] combined energy consumption and environmental data in the development of an LSTM forecasting model. Their study suggested that decent forecasting performance can be achieved when a good quality dataset is available. Lastly, Tao et al. [253] proposed a hybrid short-term forecasting model using an LSTM network for photovoltaic power predictions in conjunction with a bias compensation LSTM in an attempt to improve the predictions based on the residual error. This project highlighted the more positive effects of meta-modeling in neural network design and showed that there is more useful information to be extracted from the predicted output.

In this study, we focused on the prediction of power consumption extracted from monthly energy meter readings for electricity clients. Since the energy meter data is often collected monthly for each household or building, and the data collection process is dependent on the policies of the electricity provider, it is common for the resulting dataset structure to be problematic for most modern forecasting models. Insights and predictions are commonly based on patterns and trends extracted from recent years of consumption. Therefore, it is expected that a dataset containing monthly measurements might not have sufficient records for the training process of neural networks. Furthermore, due to different provider policies and the possibility of having a manual registration of the meter readings, the resulting datasets often contain missing or estimated values for clients that do not have any electric energy meters installed. Consequently, machine learning models trained on such data would probably overfit or exhibit poor performance on both training and test sets. The main goals of this study were: to develop a combinatorial neural network model that manages to outperform the standard single network forecasting approach, while avoiding overfitting; and to demonstrate the impact of feature engineering in the implementation of a meta-modeling technique. The proposed model examined the impact of similarity and causality among clients in an LSTM ensemble architecture in order to derive the base, similar, and causal representations of the predicted output based on changes of the input feature set. Following this step, a multilayer perceptron was used to aggregate the predicted results, in order to discover the optimal combination of those representations that could be used to predict the actual power consumption more accurately, by formulating a meta-model for stacked generalization. This project aimed to stimulate further research in the design of models that do not rely on well-structured datasets, but rather explore the inclusion of potentially helpful features that express relationships between client time series, in order to improve the base performance of models that would otherwise be considered suboptimal. Our work contributes towards the study of influential features and the discovery of patterns within the communities of electricity clients. Additionally, the examination of combinatorial forecasting approaches, similar to the ones presented in this paper, help in the presentation of more complex ideas and greatly expand research knowledge through the investigation of alternative models.

In Section 6.3, we analyze the main methods utilized in the implementation of the proposed model and proceed to provide the forecasting problem framing, with appropriate references to the dataset and performance metrics used as a case study in order to test the performance of this approach. In Section 6.4, we discuss the results of our experiments and evaluate the performance of our model. Finally, in Section 6.5, we highlight the impact of the experimental results and address the advantages as well as the challenges of this approach. Additionally, we outline some ideas for further testing and improvement of this method for future research projects.

6.3 Materials and Methods

6.3.1 Structural Presentation of Long Short-Term Memory Networks

Long short-term memory networks [161] are a class of recurrent neural networks (RNN) that can identify long-term dependencies among the input features. LSTM networks are valuable tools for time series forecasting tasks, since they can perform well, when the duration between time lags of a given sequence is unknown. Additionally, LSTMs manage to preserve gradients throughout the computation, solving one of the main issues of RNNs, where the gradients would vanish during the training process. The structure of an LSTM consists of units known as the LSTM cells. Each cell contains a set of gates that can adjust the current cell state by adding or removing information at a given time step. The cell state is transferred from one unit to the next, where further adjustments occur. The input gate at time step t determines which values will be updated and stored in the cell state. Additionally, the output gate determines which parts of the current cell state will be transferred to the output of the cell, leading to the next unit. An overview of the LSTM cell is presented in Figure 6.1 where each symbol corresponds to the respective symbols explained in the formulae of section 4.3.1.3 and the multiplication as well as the addition blocks connect the terms of each formula in this diagram. LSTM networks are trained with back propagation through time and gradient descent [254].

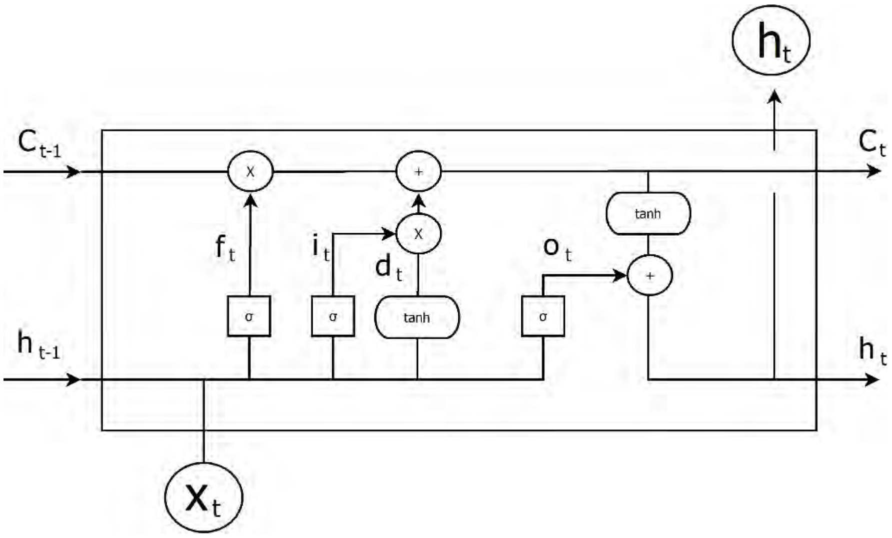


Figure 6.1: LSTM Cell Structure

In the literature, several experiments were conducted with different LSTM variants, including a variable number of units and hidden layers as well as custom training loops for sequence forecasting. However, it is evident that, while changes in the structural parameters of an LSTM can boost model performance and achieve faster training time through faster convergence, stable and reliable results are derived from the aggregation of multiple LSTMs and the construction of ensemble models [255]. Therefore, for the purposes of this study, an LSTM ensemble was considered for the forecasting experiments and the weighted average of the ensemble members was used for each representation of the predicted output. It is worth mentioning that ensemble models can also yield small performance boosts compared to the standalone LSTM, but in this project, we focused more on the stability and reproducibility that ensembles can ensure. Figure 6.2 presents the general ensemble LSTM structure.

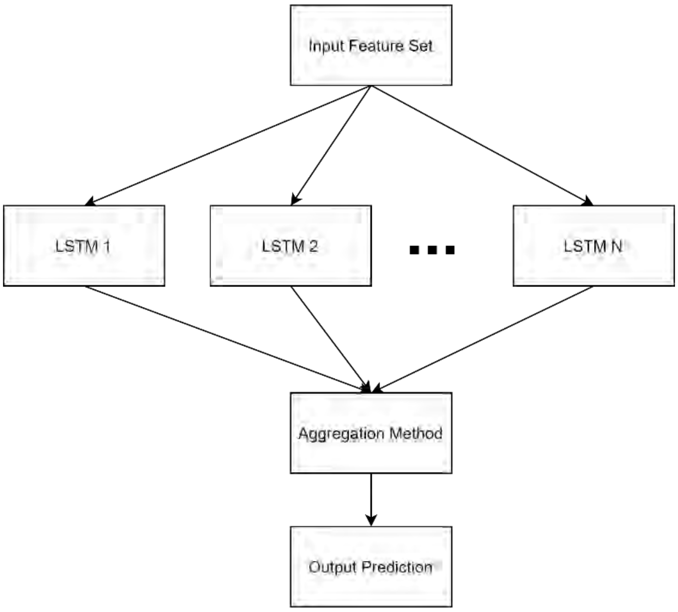


Figure 6.2: General LSTM Ensemble Structure

6.3.2 Structural Presentation of Multi-Layer Perceptron

The multi-layer perceptron is a neural network structure that follows the principles presented in section 4.3.1.1. Time series models and applications that handle energy data often utilize MLPs for univariate and multivariate regression tasks. Alternatively, MLP networks can classify load profiles as well as other variables that could group clients into distinct categories. For the purposes of this study, multi-layer perceptron was used as a meta-modeling prediction approximator that aggregates the results of LSTM ensembles and learns to predict the expected output through stacked generalization, in the spirit of [206]. Figure 6.3 presents the simple MLP structure featuring an input layer, one hidden layer and an output layer designed for univariate predictions [256].

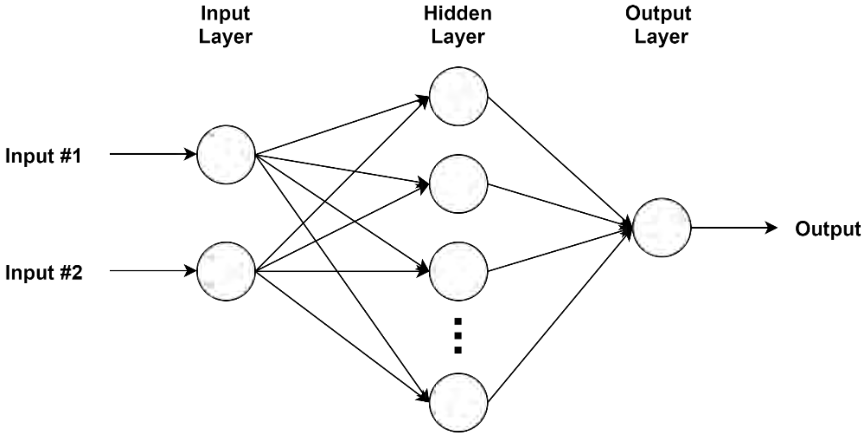


Figure 6.3: MLP with a single hidden layer.

6.3.3 Influential Community Factors

Feature engineering techniques [257] are useful because they contribute to the discovery of relationships and patterns between input features. Additionally, those methods assist towards an insightful ranking of features derived from the results of metrics and algorithms, leading to the inclusion of the most impactful features or exclusion of the least beneficial data records. Consequently, it is evident that the role of feature engineering techniques is crucial in the design of performant models for most machine learning tasks. In this study, we focused on the development of a forecasting model that utilizes the power consumption data of clients. In the literature, studies involving external variables, such as temperature and price, are common in this class of forecasting models and the focus is shifted towards the impact of additional data on a specific load profile through time. While the inspection of external variables is beneficial in the development of accurate forecasting models, we should also consider the discovery of interrelationships among the load profiles of clients and the overall community impact when selecting features that are extracted from a wider pool of consumers. The exploration of this approach could lead to the design of performant models after the investigation of features that discover associations between the power consumption of buildings. These associations can be useful when the data collection process is not ideal and external variables are not available. Additionally, models based on influential community features show the relative evolution of power consumption patterns, which is worth monitoring when electricity providers, as well as customers, want to estimate electricity tariffs and ensure that energy meters function properly. For the purposes of this study, we explored the effect of two influential community elements, namely similarity and statistical causality.

6.3.3.1 Similarity

Similarity metrics [258] quantify the structural closeness of features and rank them in order to find the ones most similar to the given input. In power forecasting tasks, where the similarity between the power consumption time series of clients is considered, the main goal is to create client associations by finding the most similar power consumption time series within the community, given the time series of one client. Since power consumption time series can vary in length or have missing values due to irregularities that occur during the data collection process, we can easily observe that conventional distance metrics that assume optimal time series alignment, such as Euclidian distance, could produce a pessimistic dissimilarity measure due to the absence of a symmetrical point-by-point match of the time series or, in some cases, misinterpret the similarity of some time series. Therefore, we chose to examine the soft dynamic time warping (soft-DTW) algorithm [259] for this project. Soft-DTW is a differentiable loss function for time series and constitutes an extension of the dynamic time warping algorithm [260] for the computation of the best time series alignment through a dynamic programming approach. As a similarity measure, soft-DTW considers all alignment matrices of two time series and produces a score that encapsulates the soft-minimum of the distribution of all costs spanned by all possible alignments. This method yields decent performance in classification and regression tasks involving time series and is considered a useful metric that can serve different purposes in the design and training of a neural network model. In detail, for the comparison of two time series x and y with respective lengths n and m, given the cost matrix $\Delta(x, y)$, the inner product of Δ, with an alignment matrix A as $\langle A, \Delta(x, y)\rangle$, and the proposed generalized operator min^{γ} with the non-negative smoothing parameter γ, soft-DTW is computed with the formula:

$$softDTW(x, y) := min^{\gamma}\left\{\langle A, \varDelta(x, y)\rangle, A \in A_{n,m}\right\} \tag{6.1}$$

6.3.3.2 Statistical Causality

Causality [261] usually refers to the abstract concept that defines a relationship between two variables, where the influence of one can partially justify the value of the other.

Typically, when causality is present there are covert dependencies between those variables and discovering them can be useful in the construction of forecasting models. When variables have a simple structure and include descriptive labels, it can be easy to distinguish the causality between them through intuition and logical reasoning, but this is not the case for more complex data. Time series data are usually collected from complex and dynamic systems and due to their structure, the detection, quantification, and interpretation of causality are challenging tasks. The relationship of cause and effect in time series could describe partial dependencies of values on the same time step as well as changes caused to the values of one sequence due to the effect of past observations of another.

Statistical causality methods, such as Granger causality [262], attempt to determine the forecasting potency of one series with regards to another. The derivation of this predictive causality is a useful tool that could complement similarity measures and feature correlations when data analysis is performed. In the scenario of power consumption forecasting the role of statistical causality is twofold. First, models that rely on lagged observations of consumption can distinguish the most influential lags for prediction by eliminating the observations that fail the statistical causality tests. Second, in the scope of a wider client pool, the forecasting potency of a lagged observation that belongs to one client with regards to future consumption of a different client can enable data augmentation due to the significance of the underlying patterns that led to this causal relationship.

In this project, we utilized the Granger causality test to infer the predictive potency of power consumption time series. The Granger causality test is a bottom-up process, where the null hypothesis states that lagged values of a variable x do not explain the variation in variable y, hence x does not Granger-cause y. The p values of the chi-square and F distributions are compared to the desired statistical significance and the results can be interpreted with the following formula:

$$result = \begin{cases} Reject\ Null\ Hypothesis, & p < 0.05 \\ Accept\ Null\ Hypothesis, & p \geq 0.05 \end{cases} \tag{6.2}$$

6.3.4 Problem Framing and Proposed Design

This study focused on the design and implementation of a power consumption forecasting model for monthly consumption predictions based on the collection of energy meter readings from a set of clients. This model attempted to integrate neural network architectures, feature engineering techniques, and a meta-modeling process in order to address the main challenges of this prediction horizon, as well as the difficulties that could arise due to a non-ideal data collection process.

Neural network models processing monthly client data with the common sliding window approach [263] often have insufficient observations for training, resulting in models that either overfit or have poor performance. Additionally, difficulties in the data collection process can result in unbalanced datasets with missing values that can affect model performance. It is also worth noting that changes in the original dataset addressing these problems could lead to the introduction of unnecessary noise, resulting in the misinterpretation of certain data patterns. Despite all the challenges mentioned above, research should not be limited to good quality datasets because data availability cannot be guaranteed for all machine learning tasks. Furthermore, the investigation of alternative approaches that could boost model performance in a non-ideal setting is interesting since those contributions shift the emphasis towards more robust structures that overcome data limitations. Our proposed approach maintained the original dataset following the common sliding window approach for predictions, while it introduced a revised model structure that can improve model performance under non-ideal conditions.

Following the sliding window approach using an LSTM network, the client dataset underwent the preprocessing phase, where the consumption dataset was clustered by client and the data of the client whose consumption was to be predicted was selected. The consumption of the client was organized into different columns representing lagged observations of the consumption at time $t-1, t-2, \dots, t-n$ derived from the original time series shifted in time. The prediction of the next month was denoted as the next column at time t, which is the target output variable. The preprocessed dataset was split into a training and validation set, and the data was scaled appropriately based on the distribution of each feature, using standardization when the distribution was normal or using normalization otherwise. The scaled features were reshaped in the form of

$[samples, timesteps, features]$ and passed to the LSTM network for training. Once the model was trained, the performance of the model was evaluated on new data from the test set and an error metric was used to determine the divergence of predictions from the actual consumption values. Figure 6.4 presents the standard design for a forecasting model utilizing an LSTM network and this common sliding window method that usually underperforms due to the challenges mentioned above.

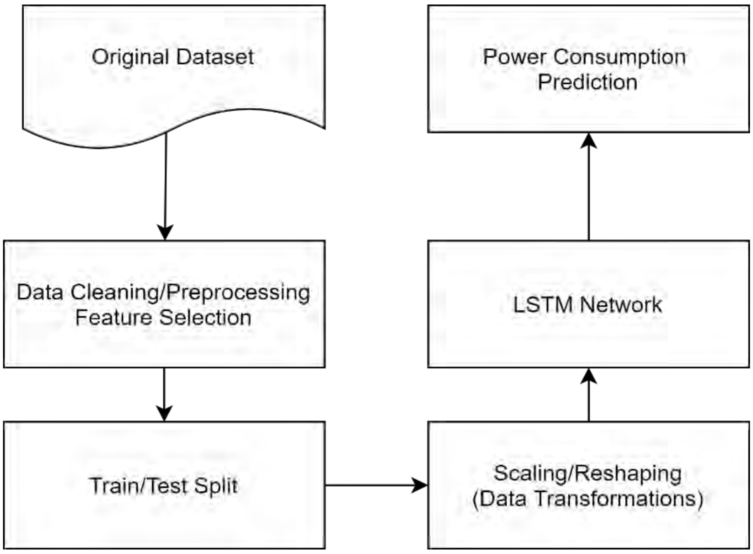

Original Dataset
Data Cleaning/Preprocessing Feature Selection
Train/Test Split
Scaling/Reshaping (Data Transformations)
LSTM Network
Power Consumption Prediction

Figure 6.4: Standard design of power forecasting model using an LSTM network.

We extended the previously described method by introducing several modifications and new components aiming at a performance boost: instead of a standalone LSTM network, an LSTM network ensemble of n members was considered in order to derive more reliable predictions. Each member of the LSTM ensemble features an early stopping mechanism [264] that effectively stops the training process when the validation loss of the model stops decreasing. This mechanism prevents overfitting and selects the epoch where the model would achieve the best predictive performance on unseen data. The prediction of the LSTM ensemble is the aggregate prediction of the members derived from their weighted average, where the weights are determined using grid search [265]. This process was used on the original feature set of lagged observations to derive the base representation of the predicted consumption. Following this step, two different feature sets were constructed based on the influential community factors of similarity and causality. The first feature set contained the original lagged observations, as well as lagged observations of other clients at the same time steps, determined by their similarity ranking based on the previously described soft-DTW method. The second feature set contained the original lagged observations and lagged observations of other clients at the same time steps based on their predictive potency as determined by a Granger causality test when the targeted effect was the monthly consumption at time t of the original client in the training set. Those two feature sets passed through the LSTM ensemble and produced the similar and causal representations of the predicted consumption, respectively. Since the effects of causality and similarity are not strictly predetermined to be positive, we implement a meta-modeling technique that aimed to aggregate the three representations in order to create a model that discovered a weighted combination of those representations. Intuitively, this combinatorial model used an MLP as a meta-learner used for stacked generalization [266], thus creating an ensemble of ensembles that was expected to yield improved performance when compared to single layer LSTM ensembles. Figure 6.5 presents the design of the proposed approach.

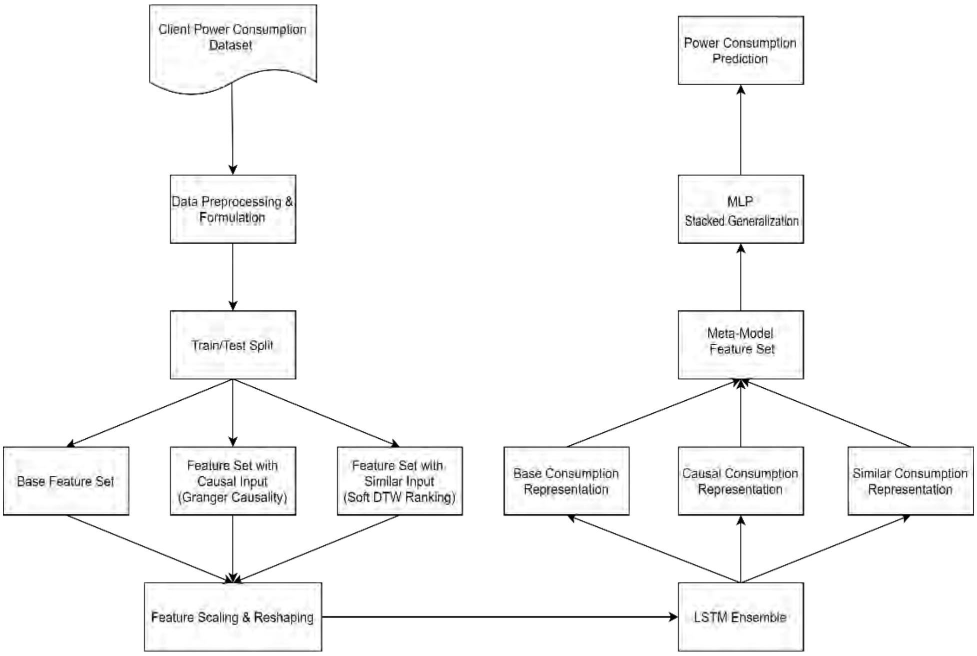

Client Power Consumption Dataset
Data Preprocessing & Formulation
Train/Test Split
Base Feature Set
Feature Set with Causal Input (Granger Causality)
Feature Set with Similar Input (Soft DTW Ranking)
Feature Scaling & Reshaping
LSTM Ensemble
Base Consumption Representation
Causal Consumption Representation
Similar Consumption Representation
Meta-Model Feature Set
MLP Stacked Generalization
Power Consumption Prediction

Figure 6.5**:** Proposed model design utilizing three LSTM ensemble sub-models in the development of a meta-model based on an MLP network.

6.3.5 Case Study and Experiments

In this section, we present a case study used to test our proposed approach. The dataset used for our experiments contained monthly power consumption data of clients located in seven municipalities of Nariño, Colombia from December 2010 to May 2016, and it is freely and publicly available [267]. The data was collected and registered by workers of the company Centrales Eléctricas de Nariño (CEDENAR) after manual inspection of electric energy meters installed at the building of each individual client. The consumption measurements were obtained from the monthly readings of those meters in kWh. The only exception to this manual procedure happened in the case where a client did not have an energy meter installed. In this situation, the estimated consumption derived from the installed electric load of the connected appliances was used. Additionally, the dataset contains socio-demographic features such as area, municipality, use, and stratum that further describe each client. The key index that uniquely identified each client is a code that includes a concatenation of the socio-demographic characteristics. According to the authors of the paper that introduces the dataset, the data was processed and ready for direct use in the implementation and testing of forecasting models. Furthermore, time series data for each client can be extracted when the observations are clustered by the unique code identifier. After further inspection, we deduced that this dataset was suitable for testing since the pool of clients is sufficiently diverse, containing clients that live in rural and urban areas, while using electricity in different environments ranging from residential to industrial and special. Furthermore, the feature of power consumption values is equally diverse, ranging from 1.009 to 305,687.4 kWh. Therefore, the exploration of influential community factors, such as similarity and causality, for the individual power consumption forecasts could be interesting as the dataset includes clients that satisfy a wider spectrum of consumption scenarios.

Firstly, further inspection of the dataset was conducted and additional preprocessing was necessary for the extraction of the consumption time series for each client. Clients were clustered by code with the requirement that each date index contains one consumption measurement for that month. Consequently, 90 clients were detected and formed a new time series dataset. The resulting dataset fit the non-ideal scenario we wished to explore in this project, since it contains several missing values, possibly due to the manual registration process. Additionally, in terms of data shape, each user does not have more than 65 consumption observations associated to the corresponding months of data collection. Therefore, the possibility of having poor performance when training neural network models on this dataset was high. Initial testing was conducted on single layer LSTM networks containing 100 units and using the Adam optimizer in the prediction of the next monthly consumption value based on three previous months as input. The training set consisted of the first four years of data and the test set contained the remaining months. The result of the initial simple model, previously presented in Figure 6.4, exhibited overfitting, thus confirming our intuition to implement the extensions that we proposed in order to stabilize the model and improve its performance.

The problem formulation using lagged observations remained the same in order to have a fair comparison of the modified models. The first modification was the implementation of an early stopping mechanism that attempted to stop training the network when the error metrics derived from the evaluation of the model on validation data after each epoch stop decreasing. The initial number of epochs was set to 4000 and after continuous monitoring of the error metrics from consecutive executions the patience interval, which determines the number of epochs after no improvement to the loss function was detected, was set at 170 epochs, preventing overfitting. This patience interval remained proportionately small when compared to the total training epochs and provided a sufficient window that allowed the improvement of the model. However, due to the decreased training epochs, the model yielded suboptimal performance. Therefore, the replacement of the single LSTM network with an LSTM ensemble of n members yielded more stable and reproducible training results and minor performance improvements. For the purposes of this study, the ensemble contained two members in order to balance execution time and stability benefit. The iterative increase of ensemble members only increased the execution time of our experiments, hence the choice of two ensemble members was appropriate for this dataset. Since the number of ensemble members was a parameter that depended on the dataset and the model structure, future research is encouraged to perform similar experimentation in order to establish the benefit of a larger ensemble before finalizing the model. The output prediction of the ensemble LSTM was the weighted average prediction of the members using grid search.

Taking this approach one step further, we explored the effects of similarity and causality among clients by forming two additional models utilizing the same LSTM ensemble structure as the base model. The first model focused on similarity and contained a modified feature set, where lagged observations from the most similar clients were included alongside the base client. A soft-DTW ranking was used to determine the top three closest clients that had the lowest distance scores. Intuitively, the number of the most similar clients should remain small when compared to the total number of clients in order to strengthen the similarity between the features used in the model. Our experiments indicated that three clients were sufficient in the construction of a feature set that includes power consumption values with a high likelihood of corresponding to the same electricity usage type. Generally, the number of the most similar clients is selected based on the dataset, with an emphasis on creating a small set of similar clients, reinforcing cohesion between the members of the set. The second model extended the base feature set by including lagged observations of power consumption after inspecting all other clients and selecting the columns of lags that satisfied the previously discussed predictive efficacy criterion, by rejecting the null hypothesis of a Granger causality test when that column was tested against the targeted output consumption of the training set for the main client. Each feature in every model was normalized or standardized based on the Shapiro-Wilk statistical test [268] before training.

Since all three LSTM ensemble models share those performance hurdles due to data limitations and the implementation of early stopping, the investigation of a combinatorial approach was interesting due to the variety of feature sets. Therefore, a meta-learner was developed, utilizing a single hidden layer MLP network with 100 neurons. The activation function was the rectified linear unit (ReLU) and the optimizer was Adam. Moreover, 4000 was the selected number of epochs for training and the same early stopping mechanism was utilized in order to prevent overfitting. The meta-learner used the output predictions of the three LSTM ensemble models in order to discover the best weighted combination and predict power consumption more accurately. Experiments for the comparison of those models focused on the prediction of the power consumption of a client for the next 14 months. The comparison considered the performance of each standalone LSTM ensemble using the base, causal, and similar feature set, respectively, as well as combinatorial models utilizing the meta-learner for the pairwise stacked generalization of the ensembles. Finally, the combinatorial model that utilized all three LSTM ensembles was examined and the results are presented in the following section.

The experiments presented in this study were implemented in Python 3.8.8, using the packages pandas 1.2.3, numpy 1.19.2, scikit-learn 0.24.1, tensorflow 2.3.0, keras 2.4.3, statsmodels 0.12.2 and matplotlib 3.3.4. It is worth mentioning that any model parameters not mentioned in this section follow the default values of those packages. The models and experiments were executed on a desktop computer with an AMD Ryzen 1700X processor, 8 gigabytes of RAM, and a NVIDIA 1080Ti graphics processor. The code of this study, containing the implementation of this power consumption forecasting approach, is publicly available on GitHub [269].

6.3.6 Performance Metrics

In this section, we present an overview of the performance metrics used in the evaluation of the neural network models in order to explain their intended usage in our experiments. The metric of MAE was utilized as the loss function for the training of our neural network models since it is a simple measure that we can use to monitor how the divergence of predicted values from the real values decreased after every epoch [87]. Additionally, this metric was utilized in the final performance evaluation of the estimated time series components in order to capture a natural measure of average error. Furthermore, MAPE and RMSE were utilized in the performance evaluation of this meta-modeling approach in order to examine different aspects of error in the predicted time series. The metric of MAPE was utilized in order to provide a scale independent measurement of relative error and the metric of RMSE was used as a secondary scale dependent indicator with attention given only to the relative decrease of the value denoting the improved performance of the model [88,90].

6.4 Results

In this section, we present an overview of the experimental results through figures and error metrics that are based on the findings of the case study in order to evaluate the combinatorial model described in this project. The experiments consisted of the random selection of clients and the construction of individual forecasting models utilizing the base feature set of lagged consumption observations, all pairwise combinations of the base feature set, and the additional columns from the exploration of similarity and causality, as well as the final combinatorial model, which utilizes all three sub-models for stacked generalization. Furthermore, for the clear and concise demonstration of the results, we provide the representative comparison of those models for the predictions of 14 months of power comparison for an individual client. It is worth mentioning that the relative boost in performance following this method was maintained when different clients were selected from the dataset, following the same behavior for standalone models, sub-model pairs, as well as the combinatorial model. Additionally, the error metrics were derived as averages from 10 consecutive executions. Since the changes in the error metric values were miniscule, we found that 10 iterative executions were sufficient in the consolidation of measurements.

First, in Table 6.1 we list the values of MAPE and RMSE for all the models considered in this comparison. We can observe from the values of MAPE that, while the standalone models exhibited fair, but not optimal results given the dataset structure and the implementation of early stopping, the sub-model pairs contributed towards a more accurate meta-model. Moreover, the meta-model that utilized the base, similar, and causal sub-models performed better than all other models in this comparison, showing that the combination of many different models based on varying feature sets can result in a performance boost. The secondary performance metric values of RMSE showed a considerable decreasing trend when we transitioned from the standalone models to pairs of sub-models and, finally, to the three-component meta-model. The values of RMSE were justified due to the range of power consumption values in the dataset and we mainly focused on the decreasing trend in order to determine the improvement. Table 6.1 labels the standalone LSTM ensemble models as base, causal, and similar depending on the feature sets used. The meta-models utilizing pairs of LSTM ensembles are labeled as base-causal, base-similar, and causal-similar. The final combinatorial model using all sub-model ensembles is labeled as base-causal-similar.

Table 6.1: Performance comparison of standalone models, sub-model pairs, and combinatorial meta-model.

Model	MAPE	RMSE	MAE
Base	15.62	8485.73	5865.11
Causal	20.37	9749.18	7465.28

Similar	18.06	8984.84	6595.78
Base-Causal	6.36	6333.37	2739.73
Base-Similar	6.28	3635.15	1915.17
Causal-Similar	8.62	4595.11	2747.87
Base-Causal-Similar	3.49	1697.14	1122.30

Second, Figure 6.6 presents a direct comparison of the actual and predicted values of power consumption for the targeted output of 14 months between the standalone LSTM ensemble models and the final combinatorial meta-model utilizing an MLP. Through this comparison it is clear that no standalone model could get accurate predictions when consumption values show sudden valleys and peaks, such as the areas between data points 3 and 5, as well as data points 8 and 12. The standalone models managed to capture the decreasing and increasing patterns later in time, producing an outcome that seems to be shifted, distorting the result. Additionally, Figure 6.7 presents a direct comparison between the meta-models created by the combination of LSTM ensemble pairs and the meta-model that utilized all three LSTM ensembles. The inspection of this figure could lead to some interesting assumptions since the involvement of the base LSTM ensemble resulted in meta-models that could adapt better to sudden decrease in consumption. Similarly, the involvement of the component of similarity led to models that could capture the sudden increase in consumption. While this behavior could be situational to each model for each individual client, it shows that the combination of sub-models utilizing influential community characteristics could lead to a better fit in the regions where simpler standalone models would not be able to adapt that well. It is evident that the involvement of all three sub-models led to the development of the most accurate meta-model.

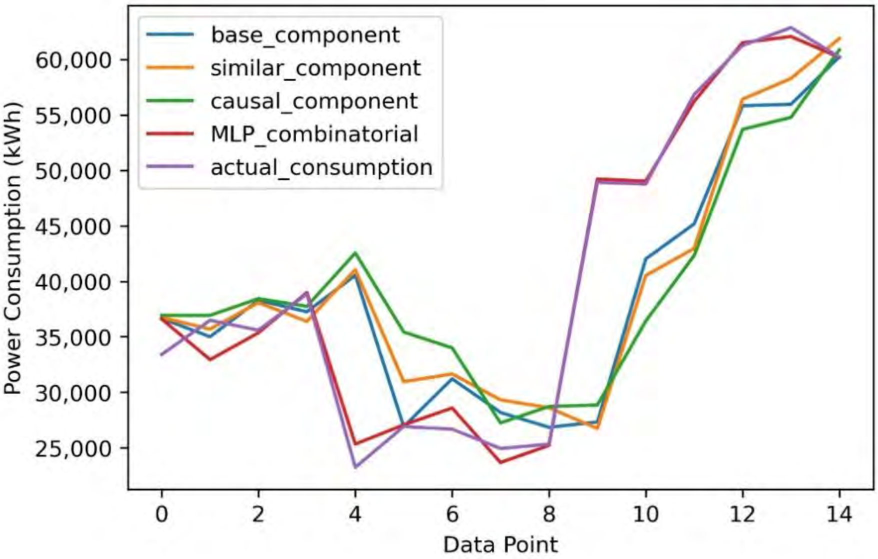


Figure 6.6**:** Comparison of the predicted values between the standalone models featuring one component and the final meta-model.

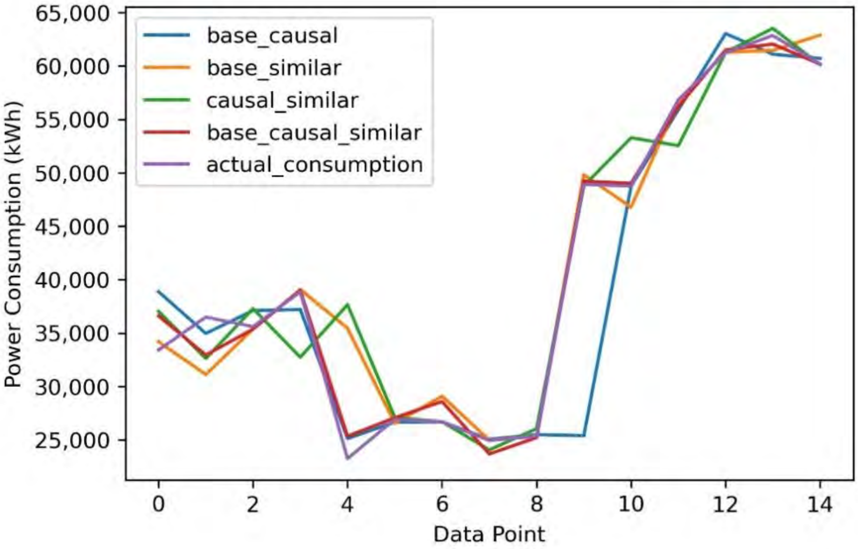


Figure 6.7**:** Comparison of the predicted values between the sub-model pairs and the final meta-model.

Finally, for completeness, we present the graph that shows the training history of the final meta-model in Figure 6.8. In this graph, we observe that the loss function MAE kept decreasing for the training and validation set. The initial training epochs were set to 4000, but the model stopped training after 3500 epochs due to an early-stopping mechanism that prevented overfitting.

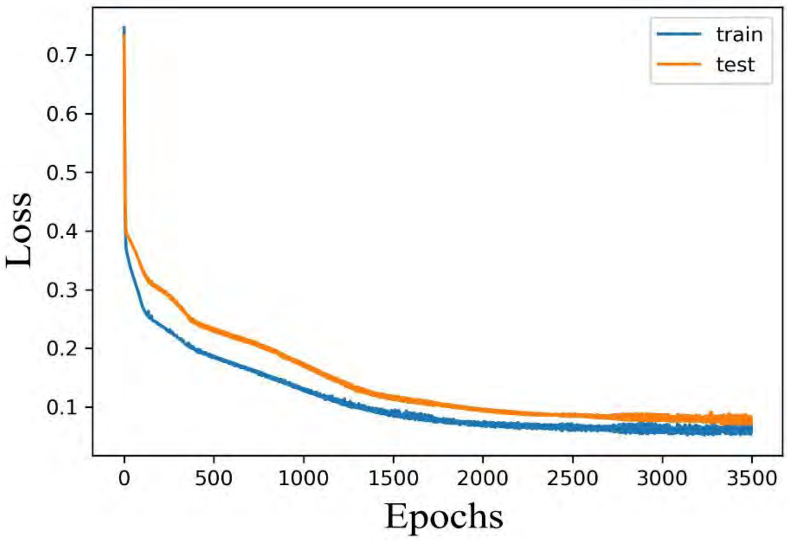


Figure 6.8: Training history showing the loss function of MAE for the final meta-model utilizing similarity and causality.

6.5 Discussion

This work explored the impact of similarity and causality in the development of a combinatorial power consumption forecasting model for electricity clients based on neural networks. Since the proposed model focused on a more realistic approach that addressed the main challenges in neural network model design, a case study was carried out using a dataset that was derived from a non-ideal data collection process. The research findings showed that, while the standard LSTM network, which only utilized lagged observations of the main client, could overfit and exhibit suboptimal performance, the development of meta-models based on combinations of feature sets that were influenced by the similarity and causality could achieve a better and more stable performance. In detail, the LSTM ensemble model utilizing only the lagged observations of the client had a MAPE of 15.62 and was outperformed by the meta-models, which utilized pairs of LSTM ensemble sub-models. In those experiments, the meta-model that utilized the output of the LSTM ensembles with the base and similar feature sets yielded the highest pairwise performance with a MAPE value of 6.28. In conclusion, the final meta-model that utilized the outputs of LSTM ensembles, which included the base feature sets as well as feature sets influenced by similarity and causality, yielded the highest performance when compared to all other models, achieving a MAPE of 3.49.

The process of designing this meta-model, as well as the results of this study, contribute greatly towards the introduction and development of more robust and complex combinatorial models that address the current challenges of forecasting model design and are more resilient towards the available dataset structure. This novel forecasting approach presents ideas that mitigate important hindrances in the performance of LSTM models and investigate the potential benefits of influential community factors, assisting in the implementation of performant models when the available data and the prediction horizon are far from ideal. Our project hopes to encourage further work in this field since it was observed that the consideration of many different feature sets can achieve better aggregated results. It is important to note that related work in this field shows that the standalone concepts of similarity and causality can be effective in the prediction of energy data in various horizons [270,271], but to the best of our knowledge, there are not many available experiments that consider the combination of the two on either short-term or long-term predictions given a group of electricity clients regardless of data structure. Therefore, this work attempted to fill this research gap by providing useful insights given the scenario described in the case study. Future work on this class of meta-models could explore many different aspects, which were not available in the current dataset; for instance, it would be interesting to study the inclusion of more detailed features, such as occupancy and appliance information, in order to reinforce the results of similarity and causality tests. It would also be interesting to explore the performance of the model in a more ideal setting, where the available dataset contains consumption data from a much wider pool of clients, without missing values, in order to inspect how the model behaves with big data in a more ideal configuration. Finally, from the perspective of training performance and execution time, future work could parallelize this model and execute it on multiple graphics processors in order to inspect the improvements of the multithreaded implementation.

Chapter 7 Implementation of an Error Compensation Approach Towards Prediction Improvement

7.1 Motivation

Complementing the concepts presented in the introduction of the meta-modeling technique, a posteriori processing could be beneficial for short-term forecasting tasks in the energy sector without the need to shift the focus of the main forecasting module. The implementation of additional processing and estimation layers could be applied to the output module towards the direct adjustment of estimated target time series values. This approach could be impactful in the refinement of error and the stabilization of predictions when the target variables are strongly affected by the seasonal patterns of the influential features and exhibit a degree of volatility since the error component responsible for suboptimal performance could be isolated and examined at the later stages of the forecasting process. The a posteriori examination and subsequent estimation of error could lead to the discovery of patterns that could manipulate the degree of randomness from predicted residuals, resulting in more consistent error values. Additionally, these models could derive approximation functions that generate smoother error samples, resulting in residual error time series that are more resilient to error spikes and outliers. Therefore, load and price time series could be predicted more accurately and consistently within the studied forecasting horizons. It is evident that while this method has the potential to substantially improve error metrics and resolve the performance hinderances that could occur from the emergence of large errors, recent research projects, reviews and benchmarks focus on standalone and hybrid structures that do not utilize this processing step. This omission is critical as it highlights several research gaps with regards to forecasting design and hyperparameter optimization.

Forecasting methods designed based on a priori processing principles face inevitable model behaviors stemming from the relationship between output performance and dataset quality. Sophisticated model structures could drastically underperform when the input dataset does not match the assumptions of the forecasting technique or when abnormal behaviors are exhibited in certain data regions, requiring more transformations that could impact interpretability and convergence time negatively. Furthermore, the resulting models could perform differently based on the choice of different learning parameters. These parameters need to be recalibrated when new input data is handled. It is clear that as the standalone or hybrid estimator becomes more complex, the input data volume increases and the studied forecasting horizon becomes shorter, recalibration is restricted and parameter tuning becomes more difficult. As a result, highly granular predictions may not be derived within the expected time intervals and the optimal parameter set may not be successfully determined. The implementation of a posteriori processing techniques such as error estimation operate at the end of the forecasting process and could be input or model structure agnostic. These processes could derive deterministic strategies for error minimization where the benefit yielded is dependent solely on the shape of the error component. Consequently, when time and resource constraints are considered, the derivation of satisfactorily accurate predictions could be derived through less computationally expensive error refinement processes instead of the expansive search for additional data transformations and model parameters. Alternatively, these methods could be utilized as a feedback mechanism in order to monitor model performance through the examination of error shape, signaling for more parameter adjustments. Since the effectiveness of these methods depends on the structure of the error component, this examination could highlight certain aspects that need to be altered during recalibration.

Moreover, the insufficient examination of those methods could be detrimental to the evolution of novel hyperparameter parameter optimization strategies as the set of hyperparameters would only be partially explored. A posteriori processing methods estimating the error component often introduce a separate forecasting pipeline combined with heuristics and statistical tests that consolidate the strategies involved. Therefore, the decision parameters and the criteria that validate those methods could be included in the search for optimal solutions for the entirety of the forecasting structure, forming a more robust expanded set of hyperparameters. This perspective could be challenging as more hyperparameters could increase the overall complexity and computation time of the forecasting structure. However, through the integration of this additional forecasting layer and the potential introduction of new hyperparameters, the boundaries and tradeoffs of model complexity could be examined further since the path towards a balanced structure featuring a performant forecasting module and an adequately efficient error compensation module at the output could be highlighted.

Considering the previous observations and research gaps, in this chapter a novel hybrid neural network structure featuring an error compensation autoregression module is developed for a posteriori processing towards the reinforcement of error stability and improvement of forecasting accuracy. The case study presented in this work utilized Nord Pool power market data for the formulation of a day-ahead electricity price forecasting task. The proposed method contributes towards the development of more flexible hybrid neural network models and the potential integration of the error estimation module in future benchmarks, given a small and interpretable set of hyperparameters. The following sections correspond to the introduction, methodology, results and discussion of this hybrid forecasting approach.

7.2 Introduction to Error Enhanced Day-Ahead Electricity Price Forecasting

Modern energy markets follow increasingly complex processes in order to perform efficient electricity trading that balances supply and demand while reacting to the dynamics derived from the unique characteristics and challenges of each energy system. One of the main challenges that urge the development of more sophisticated techniques for the coordinated production and supply of electricity is price volatility [272]. The price of electricity can fluctuate due to several factors and the sudden peaks and valleys in the price curves could lead to suboptimal energy market agent behavior, hindering the ability of those entities to execute economic transactions in the electricity market to the best of their envisaged capacity. Some of the most notable factors that could cause price fluctuations to include seasonal trends [273], weather conditions [274], penetration of renewable energy sources [275], challenges involving economic growth and changes in fuel cost [276], supply availability [277] and neighboring market dynamics [278]. It can be easily observed that load and generation dependencies on the time of day or year as well as seasonal trends coupled with extreme hot or cold temperatures and extreme conditions, such as hurricanes could have a noticeable impact on the electricity price. Furthermore, the infrastructural development of growing economies often leads to increased energy demand and electricity costs. Additionally, electricity price fluctuations could depend on the availability of fossil fuels for sufficient generation. Price spikes could occur if more expensive forms of electricity generation are used due to fossil fuel shortages. Since renewable energy sources constitute enticing alternatives for electricity generation, the stability of each energy source is related to the stability of electricity prices and the price fluctuation patterns vary for each region. Lastly, the impact of neighboring markets on price volatility could be attributed to the increased player participation and decentralization that increases the complexity of price formation.

Real-time energy markets could be negatively affected by price volatility since market participants could be unable to react proactively when price fluctuations and energy transactions occur throughout the operating day. However, price volatility can be tempered with the development of day-ahead energy markets that allow buyers and sellers to determine and secure energy prices before the operating day [279]. Therefore, short-term forecasting models that predict day-ahead prices are valuable for the successful monitoring of price trends and coordination of supply and demand. Price data and influential features are typically collected in the form of time series, following an hourly sampling rate. Statistical methods and machine learning models contribute greatly towards the development of accurate and robust day-ahead electricity price forecasting models that are capable of processing time series data efficiently and handling the complexity of those energy markets [280,281]. Forecasting models derived from statistical methods often utilize linear regression [282] in order to model the target variable as the linear combination of independent features. Additionally, autoregressive models [283–285] expanded on this concept by highlighting the importance of autocorrelation between values of the same variable from previous time steps. Machine learning models for day-ahead forecasting often rely on the development of neural networks that operate as function approximators and aim to detect the linear and nonlinear relationships between the input and output features. The primary neural network types utilized for this forecasting task include the multi-layer perceptron (MLP) [286] and the feed-forward deep neural network (DNN) [287], long short-term memory networks (LSTM) [288] and convolutional neural networks (CNN) [289]. The appeal of methods involving the MLP and DNN [290] can be justified due to the ease of use and the simplicity of structure since MLPs include fully connected layers of neurons that form a computation path from the input to the output, resulting in a network that is acyclic in nature. On the other hand, LSTMs are recurrent neural networks (RNN) [291] that follow a block structure consisting of gates that interact with the previous and next state of the network. Long short-term memory networks are more complex when compared to fully connected feed-forward network types, but their structure could handle temporal dependencies between time series lags of unknown length more efficiently. Lastly, CNN networks [292] use one-dimensional convolution to learn patterns within specific time windows and can inspect the data from a broader perspective through data shuffling. Neural networks constitute impactful short-term forecasting tools in the energy sector and a plethora of different standalone and combinatorial structures are studied for fast and accurate predictions [185].

Models derived from the aggregation of previously mentioned networks form the category of ensemble learning [293] and have substantial forecasting performance benefits. The combination of different types of models belonging either in the statistical method or the machine learning class with the integration of modules that contribute towards data decomposition, feature selection, clustering, or heuristic optimization, form the class of hybrid forecasting methods [294–296] that often succeed in the analysis of more complex dynamics and patterns. Benchmarks in the field of day-ahead price forecasting mainly utilize autoregressive and deep neural network models since these structures offer state-of-the-art performance and simplicity of implementation. The evaluation of new approaches and the process of model selection through those benchmarks rely primarily on hyperparameter optimization, feature selection and regularization techniques [297].

Recent research projects and reviews highlighted interesting short-term electricity price forecasting approaches that utilize elements from statistical and machine learning methods. Alamaniotis et al. [226] proposed a multiple regression model based on relevance vector machines for day-ahead electricity price forecasting, contributing towards the development of optimal bidding strategies in electricity markets. Moreover, Alamaniotis et al. [227] developed a hybrid forecasting model featuring relevance vector machines in a linear regression ensemble method for efficient short-term price forecasting. Zhang et al. [298] presented a forecasting method that aggregates the combined predictions from CNN and RNN structures in a gradient boosting regressor yielding improved performance. Additionally, this study highlighted the importance of elastic net regularization for the stability and reliability of this combinatorial method. Alamaniotis et al. [299] developed a combinatorial approach that couples load and price forecasting and modifies forecasted load demand through the implementation of smart scheduling algorithms. Chinnathambi et al. [300] developed a multi-stage day-ahead forecasting model based on the autoregressive integrated moving average (ARIMA) statistical approach and the consequent residual error forecast that improves the performance of the initial predictions for different time periods. This research project provides some useful insights on the utilization of post-processing factors, such as the error for the improvement of statistical methods. Chang et al. [301] proposed a forecasting model that utilizes wavelet transform and an LSTM network featuring the stochastic gradient optimizer Adam, demonstrating that a well-optimized recurrent neural network could capture and process the nonlinear patterns in this task efficiently. Su et al. [302] utilized the least squares regression boosting algorithm to predict natural gas spot prices, outperforming existing approaches, such as linear regression. Atef and Eltawil [303] conducted a comparison between support vector regression (SVR) and LSTM electricity price forecasting models, concluding that while both methods could be suitable for this predictive task, the deep learning approach outperforms the regression model in terms of error metrics. Bissing et al. [304] investigated the different combinations of regression, namely the ARIMA and Holt-Winters models, for day-ahead forecasting and provided some interesting results regarding the performance benefits of hybrid implementations. Xu and Baldick [305] compared different neural network architectures and some state-of-the-art statistical methods, concluding that neural network models could perform better for price forecasting while yielding lower mean absolute error. Zhang et al. [306] studied the performance of deep recurrent neural networks for electricity price forecasts in a deregulated market, providing useful insights on the suitability of this neural network type as a multivariate time series model. Lago et al. [307] presented a review of state-of-the-art price forecasting models covering statistical, machine learning and hybrid approaches. Furthermore, this research work provided a useful open-access benchmark including a regression and a deep neural network model that utilize hyperparameter optimization for future model comparisons. Tao et al. [253] proposed a bias compensation LSTM network utilizing the LightBGM algorithm for feature selection. This work contributed significantly towards the development of hybrid short-term forecasting models since the introduction of residual error analysis for recurrent neural networks is a novel approach that could refine time series predictions. Vega-Márquez et al. [308] approached the electricity price forecasting task from a univariate time series perspective and tested well-known deep learning and statistical methods through hyperparameter optimization, distinguishing LSTM, CNN and regression tree methods as the most performant. Jiang et al. [309] utilized a decomposition-selection-ensemble forecasting system that adapts to different data characteristics and focuses on accurate and stable price predictions. Li et al. [310] presented a price forecasting model based on variational mode decomposition and sparse Bayesian learning of time series, showing that aggregate predictions derived from components featuring simple characteristics could outperform state-of-the-art models. Pourdaryaei et al. [311] investigated the impact of different optimization methods for day-ahead price forecasting. This research work focuses mostly on the pre-processing and learning steps, while the impact of post-processing optimization techniques remains unexplored.

After a thorough overview of the literature, it is important to note that while a plethora of forecasting models exist and deep neural networks have been some of the most frequently used models, the effect of error compensation for the state-of-the-art feed-forward DNN is not sufficiently covered. We can observe that benchmarks and relevant studies utilize hyperparameter optimization as well as feature selection to tune the models and achieve lower error metrics, but fewer studies have applied post-processing techniques in order to refine and improve the predictions. Therefore, while there are recent studies that utilize error residuals for this short-term forecasting task, the application of this technique on the simple yet highly performant DNN is not thoroughly explored. As a result, the potential utilization of an error estimation module for benchmarks utilizing the DNN model as an additional tuning tool remains an open question. In this study, we identified these research gaps and developed a hybrid error compensation deep neural network model, the ERC–DNN, which utilizes a feed-forward deep neural network for day-ahead electricity price predictions, as well as an autoregression module, which operates on the hourly residual error sequences and performs a step-by-step error estimation to refine the predicted prices. The main goals of this research project are: (i) to showcase the improvement of price predictions in terms of error metrics; (ii) to investigate the stability of hourly predicted sequences after the error refinement; and (iii) to provide insights into the suitability of error estimation modules in modern benchmarks for future integration, when the appropriate parameters are defined. This hybrid approach was evaluated on the dataset of the Nord Pool market following the guidelines of the benchmark presented in [307], and through different training scenarios that highlight the positive impact of error refinement. Moreover, the resulting error metrics of this approach are compared to a baseline DNN structure developed using well-known configuration and training practices in order to achieve a similar score to the DNN benchmark with a static set of hyperparameters that does not alter the tests and produces consistent results during recalibration. Additionally, the error metrics of ERC–DNN are compared to the benchmark scores despite the differences in training epochs and hyperparameter optimization in order to highlight the overall effect of the error estimation module.

Section 7.3 presents the main methods utilized in the implementation of the proposed forecasting approach with references to the core components of the network, as well as information regarding the dataset and the configuration of the experiments. Furthermore, this section defines the error metrics used to evaluate the performance of ERC–DNN. Section 7.4 discusses the results of the experiments and compares performance metrics to the baseline and benchmark models. Finally, in Section 7.5, the advantages, as well as the challenges of this hybrid model, are outlined. Additionally, comments regarding the impact of this model as a standalone project, the potential expansion of the proposed architecture, and the integration of this model to more complex forecasting structures and open-access benchmarks in the future are included, in the hope that they contribute to the intelligence gathered in this area of research.

7.3 Materials and Methods

7.3.1 Feedforward Deep Neural Network

The feedforward deep neural network is an acyclic artificial neural network [312] that follows a simple layer structure and extends the MLP architecture for the purposes of function approximation. The base unit of the feedforward DNN is the neuron which is a node designed to receive a specified number of inputs, perform computations and pass the output to connected nodes found deeper in the network. The value of the output at each node is determined by activation functions, such as the rectified linear unit and hyperbolic tangent [313]. The neurons of the DNN are organized into layers and the connections of those layers denote the computation path from the input to the output. The simplest and most frequently used DNN structure contains the input layer, where input features are passed to the first set of neurons, several hidden layers that perform additional computations and tune the learnable parameters of the network, and the output layer where one or more output values are generated at each node. For the purposes of this study, we consider the role of the feedforward DNN for the supervised learning task of regression [314] since we focus on the prediction of the electricity price for the next day. Based on this task, the goal of the DNN is to learn the mapping function that describes the complex relationship between the input variables and the output variables. As a general example, we consider the fully connected DNN presented in Figure 7.1. The DNN features an input layer i containing k inputs, a variable number f of hidden layers h, where each one contains a variable number of neurons z and, finally, an output layer o containing j neurons for the predictions of j outputs.

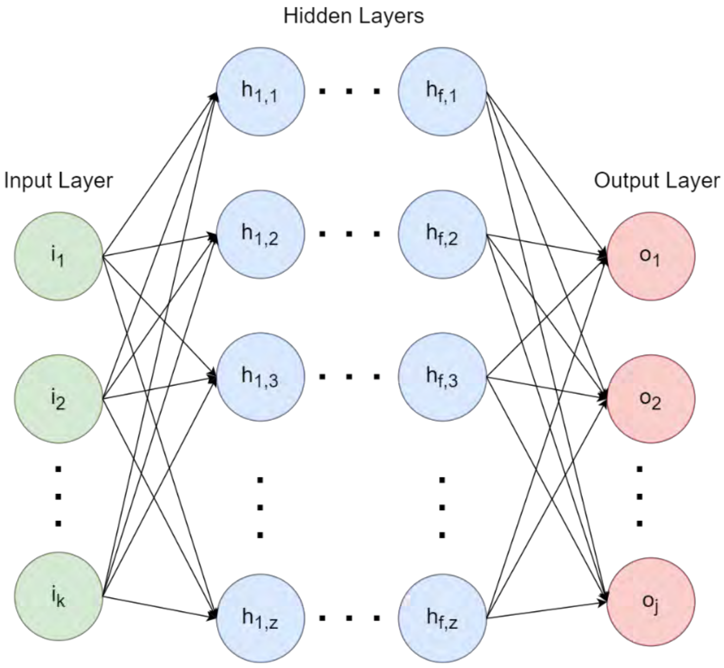


Figure 7.1: General structure of the fully connected feedforward deep neural network.

The main learnable parameters of the DNN are the weights and biases [315]. Those parameters are initially randomized and iteratively refined through the training process since the network will be able to predict the output after several passes of the training dataset, called epochs. Weights quantify the influential strength that a change in the input could have on the output and biases denote the difference between the generated output and the desired output, essentially quantifying the extent to which the network assumes that the output should have specific values. The training process of the DNN mainly follows the back-propagation algorithm [316] where the generated output values are compared to the desired output and the value of error, which is calculated by a plethora of pre-specified loss functions [317], is fed back to the network, in order to adjust the weights. Since the goal of this training process is to minimize the error function and consequently discover the best weights, optimization methods, such as gradient descent need to be specified for the training process.

The DNN architecture shows an impressive performance in time series forecasting tasks and it is widely used in the energy sector as a standalone network or as a member of hybrid and ensemble learning methods. However, the default configuration of this structure may not always be sufficient for the generation of accurate predictions due to several training scenarios that need to be avoided, such as the existence of local minima [318] of the error function that could hinder the convergence of the network and the occurrence of overfitting or underfitting that are connected to the relative complexity of the model and the dataset structure. Most deep learning models achieve optimal performance either by following a set of best practices or by exhaustively searching for the best training configuration through hyperparameter optimization [319]. Some of the most important hyperparameters include the number of neurons and layers, the choice of activation function, the choice of optimizer and the associated learning rate [320], the number of training epochs, regularization [321] and the application of early stopping [322]. The search space of those hyperparameters could be large and the total training time needed for the derivation of the best set of hyperparameters could be restrictive for models aimed at short-term and real time forecasts. Therefore, while we often see meticulous and time consuming hyperparameter optimization approaches being suitable for benchmarks, many deep learning approaches rely on the results of experiments with different combinations of best practices complemented by feature selection techniques, in order to derive their baseline models and conduct comparisons. The interpretation of those results, given a specified set of parameters, requires considerable effort towards the practical evaluation of a network and the overall demystification of the black-box structure that provides added value to research work.

7.3.2 Autoregressive Forecasting Model and Model Selection

Autoregressive models constitute a class of simple time series models used to forecast future values of the target variable based on previous observations of the same variable, called lags [323]. The target variable is linearly dependent on the lags and this relationship occurs due to some degree of correlation between lags of adjacent time steps. The number of lags utilized in the construction of an autoregressive model determines the order of the model and it is usually derived from the inspection of partial autocorrelations. The maximum lag at time step $t-n$ beyond which all other partial autocorrelations are close to zero is often used as an indicator of the order, and the model is expected to perform adequately when including lags up to that time step. The definition of the autoregressive model is made complete by the estimation of the coefficients φ_i that are multiplied by each lag, the constant term c as well as the error term ε_t. The estimation of those parameters is usually achieved with the use of the ordinary least squares method [324]. In order to present a general example, we consider the autoregressive model of order p for the prediction of the value y_t on the next time step of the sequence formed by the variable y with time lags ranging from y_{t-1} to y_{t-p}. The formula that defines this autoregressive model given the previously mentioned parameters is the following:

$$y_t = c + \sum_{i=1}^{p} \varphi_i * y_{t-i} + \varepsilon_t \tag{7.1}$$

Since autoregressive models are widely used forecasting tools with several applications in the energy sector, a few core elements need to be explored for optimal performance and the fairness of the model selection process. First, the stationarity of the data needs to be investigated since statistical models often perform better when no trend or seasonality is present. Different implementations of the autoregressive model take into consideration constant and time-dependent trends but the potential inaccurate detection of the trends and their effects on the time series forecast could sometimes lead to larger error terms. In this situation, the augmented Dickey–Fuller test [325] is utilized to determine the stationarity of a time series. According to this method, the null hypothesis assumes that a unit root exists in a time series sample and the alternate hypothesis rejects the previous assumption and considers that the time series is stationary. The p-value of the statistic results in the rejection of the null hypothesis when it is lower than 0.05. Alternatively, the comparison is between the values of the statistic and the critical values of the Dickey–Fuller t-distribution, where the value of the statistic must be more negative than the critical values to confirm stationarity. The stationarity criterion imposes restrictions to the autoregressive model that could often be seen as necessary countermeasures towards the overall reduction of uncertainty.

Second, the selection of the best autoregressive model plays a crucial role towards the minimization of forecasting error and several information criteria could be considered for the statistical evaluation of fitness to the data, such as the Akaike information criterion (AIC) [326], the Bayesian information criterion (BIC) [327] and the Hannan–Quinn information criterion (HQIC) [328]. The Akaike information criterion provides an estimation of information loss given the number of estimated model parameters k and the maximum value $\hat{L}$ of the likelihood function for the model with the following formula:

$$AIC = 2k - 2\ln(\hat{L}) \tag{7.2}$$

Furthermore, the Bayesian information criterion follows a similar formula with a slightly altered first term that features the sample size n of the observed data:

$$BIC = k\ln(n) - 2\ln(\hat{L}) \tag{7.3}$$

Lastly, the Hannan–Quinn information criterion utilizes the previously mentioned parameters in order to derive a more consistent fitness evaluation metric when compared to the AIC and follows the formula:

$$HQIC = 2k\ln(\ln(n)) - 2\ln(\hat{L}) \tag{7.4}$$

The selection of models with the lowest values of information criteria and the search for lags that have high autocorrelation values could result in a more accurate estimation of the target variable.

7.3.3 Proposed Model Structure

This research project focused on the design and implementation of a hybrid day-ahead electricity price forecasting model based on the well-known feedforward deep neural network architecture, with an additional error compensation module that estimates the prediction error and contributes towards the refinement of the final prediction. At the first step, the dataset of the model is constructed, and market data is processed in order to derive the input features, consisting of electricity price lags and exogenous variables relevant to the price time series, as well as the output features of the targeted electricity price sequences for the next day. The dataset is split into training and validation sets, undergoes normalization and is fed to the input layer of the feedforward deep neural network. At the second step, the deep neural network is trained for m epochs featuring an early-stopping mechanism that monitors the decrease of the loss function for the avoidance of overfitting with a specified patience interval, proportional to the number of epochs. Consequently, after m epochs or after the loss function stops decreasing in that patience interval, 24 sequences are generated at the output layer, each one denoting the electricity price prediction for the i_{th} hour of the next day.

At the third step, the sequences are inverted back to their original values and the residual forecasting error for each hourly sequence is calculated from the training set. The definition of the residual training error at every hour h for the price p of the day of interest d given the known values of the training dataset and the predicted output is defined by the formula:

$$p_{d,h}^{residual} = p_{d,h}^{expected} - p_{d,h}^{predicted} \tag{7.5}$$

Following this step, the residual error sequences are fed to an autoregressive model for their step-by-step estimation, resulting in the derivation of coefficients that are used to predict the error value of the next hour based on historical error data. The final price prediction is derived from the addition of the estimated error and the price forecast of the feedforward DNN. The structure of this model is presented in Figure 7.2 and this forecasting approach is used in our case study featuring several experiments on different training scenarios for the interpretation and analysis of the error compensation process. We refer to this model as ERC–DNN in the remainder of this paper.

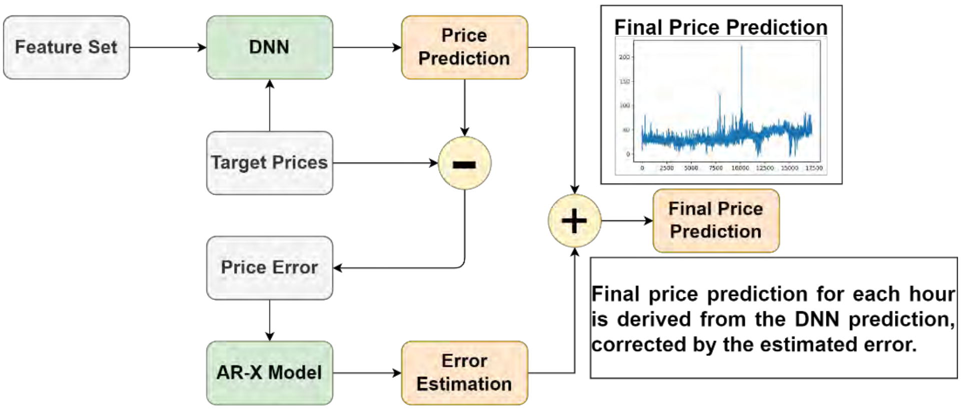


Figure 7.2: The structure of the ERC–DNN model featuring a feedforward deep neural network and an autoregressive model for error compensation.

7.3.4 Case Study and Experiments

In this section, we present a case study consisting of several experiments used to test the forecasting performance of the proposed ERC–DNN model and investigate the impact of error compensation in the stability of error profiles for each hour in the day-ahead electricity price prediction task. The dataset used for our experiments contains hourly observations of day-ahead electricity prices, as well as the exogenous sequences that represent the day-ahead forecast of load and the day-ahead forecast of wind generation for the Nord Pool energy market during the time period between 01.01.2013 and 24.12.2018. The dataset is freely available in [329] and was used by the open access benchmark of [307] to evaluate the performance of the standard feedforward deep neural network. The data is organized according to the feature formation proposed by the benchmark. The input features include historical day-ahead prices from the previous three days as well as the prices from one week ago labeled as $p_{d-1,h}$, $p_{d-2,h}$, $p_{d-3,h}$ and $p_{d-7,h}$, respectively, where d denotes the day of interest and h denotes the hour ranging from 1 to 24. Additionally, the day-ahead forecasts of the two exogenous variables are included for the day of prediction, made available on the previous day and labeled as $x^1_{d,h}$ and $x^2_{d,h}$, essentially defining a set of 48 features. Furthermore, historical values of each exogenous variable for the previous day and one week ago, labeled as $x^1_{d-1,h}$, $x^1_{d-7,h}$, $x^2_{d-1,h}$ and $x^2_{d-7,h}$. Lastly, a feature representing the day of the week as a binary vector with 7 elements is included, resulting in a total of 241 input features. The output features consist of the 24 h of day-ahead electricity prices. The dataset is split into a training set of the first 3 years, including the hourly observations from 2013 through 2016 and a validation set of the last 2 years, including years 2017 and 2018 similar to the benchmark model. According to the review and benchmark of [307], the recommended minimum testing period for the evaluation of electricity price forecasting models includes one year of observations since the common practice of including a total of four weeks, one for each season, could be unsuitable due to inadequate representation of the average model performance, the potential exclusion of extreme events that could have an impact on dataset values and the possibility of selecting only the weeks where the model shows improved performance. Therefore, following these recommendations and acknowledging the two-year period used in the benchmark, we believe that the selection of testing period in this review is a suitable evaluation practice and utilize it for the evaluation of our model. Moreover, we acknowledge that the training period varies between price forecasting models and select the maximum available historical data in the remainder of this dataset for our case study in order to have a sufficient number of observations for the convergence of the deep neural network.

Following the guidelines of the open access benchmark, we first constructed a baseline feedforward deep neural network of 4 layers for this multivariate time series forecasting task. The base DNN implements a set of best practices and consists of a fixed set of hyperparameters in order to exclude the performance benefits of hyperparameter optimization and isolate the effects of error compensation in our comparison. The exclusion of hyperparameter tuning at the preprocessing and training steps highlights the role of error estimation as an additional computational layer that reinforces the interpretability of the performance improvement through a smaller and simpler set of parameters. It is evident that the best set of hyperparameters for a forecasting model designed to perform well on a specific machine learning task is dependent on several factors including the dataset, the forecasting horizon, system or application constraints and the intended architecture. The search space for those optimal parameters is large and the resulting optimal set is often chosen based on the improvement of error metrics without having a direct and easily interpretable association to the architecture of the model. On the other hand, error estimation presents the simple concept of error refinement through the discovery of the coefficients that define the polynomial which best fits to the residual error sequences, providing a prediction of the error value that could correct the final prediction of the network by bringing the initial forecast to a value closer to the target. Therefore, error estimation operates independently from the computational structure of the deep neural network and the search goal shifts towards the selection of parameters that could prevent the values of error from exhibiting large variations and irregular patterns instead of proposing a set of parameters that attempt to configure a black-box approach.

The baseline model achieves comparable performance to the open-access benchmark in terms of error metrics as we will analyze in the following sections. The DNN structure contains an input layer of 241 neurons, two fully connected hidden layers with 100 and 52 neurons, respectively, and an output layer with 24 neurons for the prediction of the 24 hourly sequences of the day-ahead prices. The activation function is the rectified linear unit (ReLU) [330] and the optimizer is based on stochastic gradient descent [331] with a learning rate of 0.0005 for the avoidance of local minima. The dataset is normalized using min-max normalization and the neural network features an early-stopping mechanism with a patience interval that is equal to 10% of the total number of epochs in order to ensure the stability of predictions and the avoidance of overfitting. Figure 7.3 presents the structure of the baseline DNN, which is used to derive the day-ahead price predictions as a core component of the ERC–DNN model.

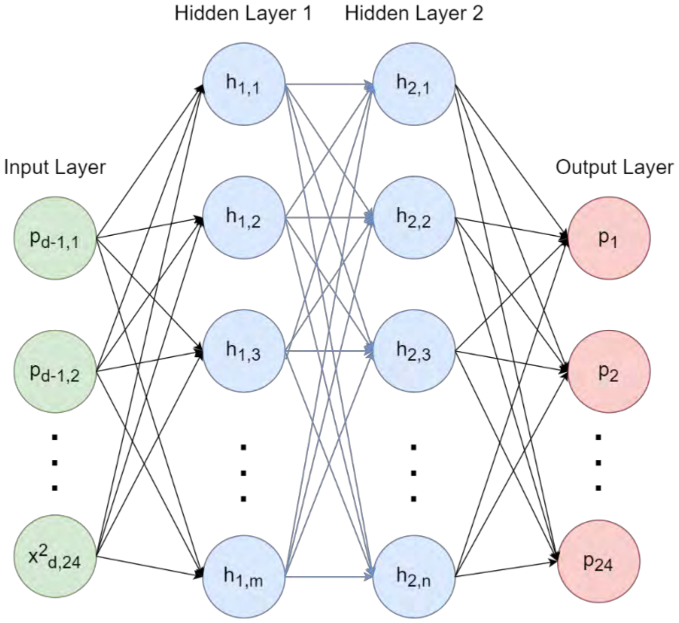


Figure 7.3**:** Baseline deep neural network structure integrated in the ERC–DNN model with $m = 100$ and $n = 52$.

The DNN is trained and the sequences for price prediction are generated at the output. The experiments presented in this work consider three training scenarios, with 10, 100 and 1000 epochs, respectively, for the investigation of error compensation in a scenario where the values of error are large and the network is not near convergence, a moderate scenario where the error has improved but there is still room for further training and a training scenario where the error of the network could marginally improve after a large number of epochs. In all three experiments, the residual error sequences for each hour are calculated and their stationarity is verified by the augmented Dickey–Fuller test. Additionally, the inspection of the partial autocorrelation function for each error sequence reveals that after the first 24 lags the partial autocorrelations decay to values near zero. The results of the stationarity test as well as the observation of the partial autocorrelation function encourage the integration of an autoregressive model for the estimation of each error sequence. Therefore, the residual error sequences are passed to an AR model utilizing a window of 24 lags for the prediction of the next value of error in each sequence. After the fitting of the model to the data, the autoregression coefficients are computed and the estimated hourly error sequences are added to the electricity price forecasts for the refinement of the final prediction. Furthermore, the information criteria of AIC, BIC and

HQIC were examined for the suitability of the 24-lag autoregressive model and the potential refinement of the model selection process when a threshold for feature autocorrelation is set at 0.2, 0.3 and 0.4. This additional experiment could contribute towards the appropriate selection of hyperparameters that could be included in future benchmarks adopting this technique for post-prediction processing of the model. Since hyperparameter optimization for this type of forecasting task already considers a sizable set of hyperparameters, the choice between the window length and the more complex threshold inspection based on information criteria could often be an important decision that could determine the size of the search space and the overall computational burden for the recalibration of a model or benchmark, given that short-term and real-time forecasting models need to recalibrate relatively fast. Figure 7.4 presents the diagram for the autoregressive model of the ERC–DNN used in the experiments.

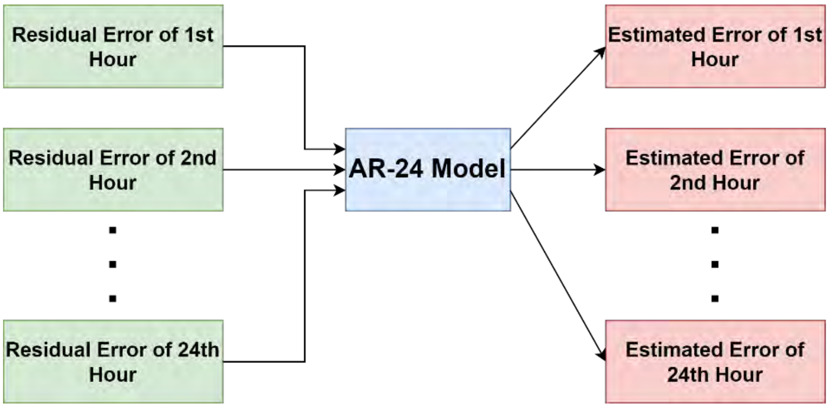

Residual Error of 1st Hour
Residual Error of 2nd Hour
Residual Error of 24th Hour
AR-24 Model
Estimated Error of 1st Hour
Estimated Error of 2nd Hour
Estimated Error of 24th Hour

Figure 7.4: Diagram of the autoregressive error estimation model.

The ERC–DNN model and the experiments analyzed in this research project were developed in Python 3.8.8, using pandas 1.2.3, numpy 1.19.2 and scikit-learn 0.24.1 for data analysis, tensorflow 2.3.0 and keras 2.4.3 for the implementation of the deep neural network model, statsmodels 0.12.2 for the implementation and evaluation of the autoregressive error estimation model and matplotlib 3.3.4 for the visualization of results. The project was executed on a desktop computer with an AMD Ryzen 1700X processor, 8 gigabytes of RAM, and a NVIDIA 1080Ti graphics processor. The code of this day-ahead electricity price forecasting model is publicly available on GitHub [332].

7.3.5 Performance Metrics

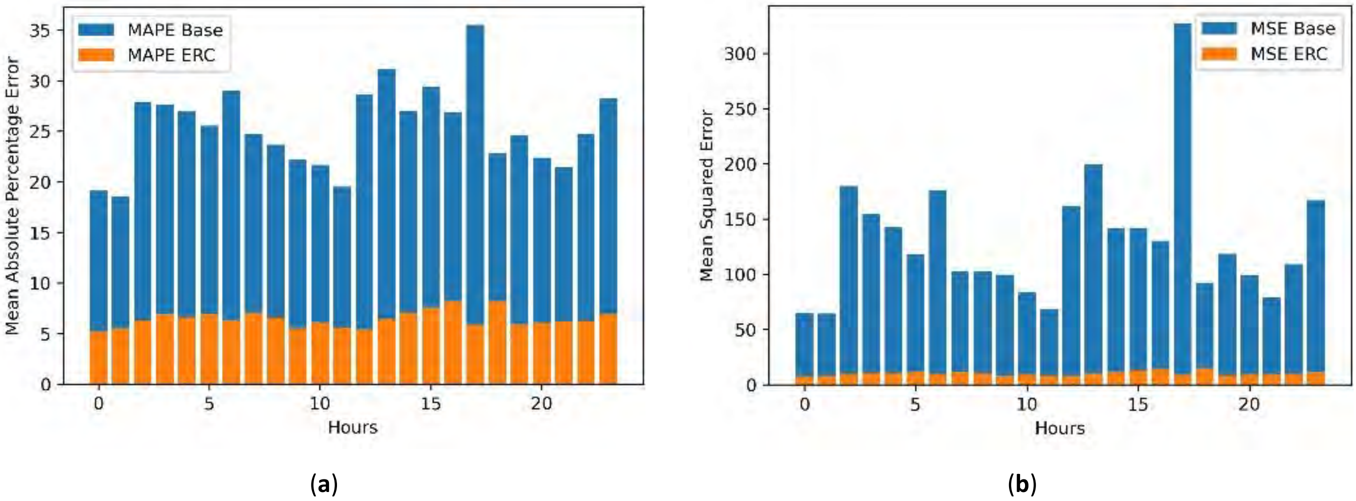
In this section, we outline the performance metrics utilized in our experiments for the comparison of forecasting error and the examination of the error refinement on the stability of error metrics for each hourly sequence of this day-ahead forecasting task. For the purposes of this study, four error metrics were used to cover different characteristics of the performance evaluation process. Mean absolute error was used as a loss function for the training of the deep neural network, the configuration of the early-stopping mechanism, as well as the evaluation of the ERC–DNN approach since it is an easily interpretable error metric. Mean absolute percentage error was utilized for the generalized measurement of relative error. Furthermore, the metrics of MSE and RMSE are included in the performance evaluation of the experiments since they provide quadratic loss functions that measure the forecasting uncertainty while focusing on the impact of large errors. The values of MSE could express the sum of the variance and square value of bias, further contributing to the performance analysis of a model. Additionally, the values of RMSE increase with the variance of the frequency distribution of error magnitudes, resulting in larger values when large error values are present [87-90,333].

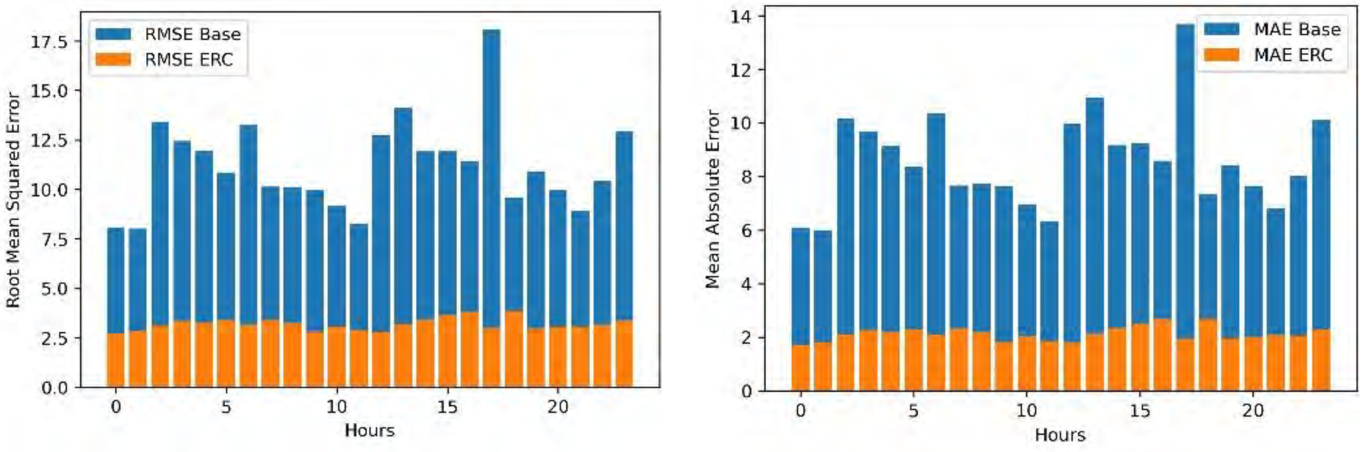
7.4 Results

In this section, we present the results of the experiments with the inclusion of figures featuring a comparison of error metrics between the ERC–DNN and the baseline DNN for each training scenario. This comparison provides an overview of the stability and performance refinement that occurred in each hourly price sequence after the autoregressive error compensation module is added to the DNN architecture. Additionally, the overall performance of the model for each scenario is presented based on aggregated error metrics, in order to examine the generalized improvement in prediction accuracy stemming from the error estimation process. Furthermore, the exploration of information criteria for the selection of a refined autoregressive model is investigated and the value of implementing a threshold method instead of the window of lagged error observations for error estimation is discussed. Since the performance metrics did not fluctuate greatly after consecutive executions, the results presented in this section constitute averages from 10 executions for each experiment. It is worth noting that the baseline DNN structure presented in this work performs similarly to the DNN model of the open access benchmark

[307] since it achieves a MAE of 1.987, a MAPE of 6.895 and an RMSE score of 3.877 after 4000 training epochs, while the DNN benchmark configuration with the lowest error metrics achieved a MAE of 1.797, a MAPE of 5.738 and an RMSE of 3.474 after hyperparameter optimization. Therefore, the resulting ERC–DNN model is utilizing a highly performant neural network component for the experiments.

First, we consider the training scenario of 10 epochs. The main purpose of this experiment is to present the effect of error compensation on the DNN forecast when the error has larger values that fluctuate greatly from sequence to sequence. In the simple univariate case, we could assume that this scenario refers to a network that has not reached convergence and could be unstable or not properly trained, while in the multivariate case we could observe that each output sequence differs greatly from the desired values and error magnitudes vary for each hour. Error compensation has the greatest impact on this scenario, as the accurate error estimation leads to a larger prediction refinement. In the subplots of Figure 7.5, we can observe that after the implementation of error compensation, large errors are no longer present, and this greatly improves the MSE and RMSE scores of the model. Moreover, the error profile for each hourly sequence is stabilized, resulting in an average model performance that is close to the model performance for each hourly predicted sequence.

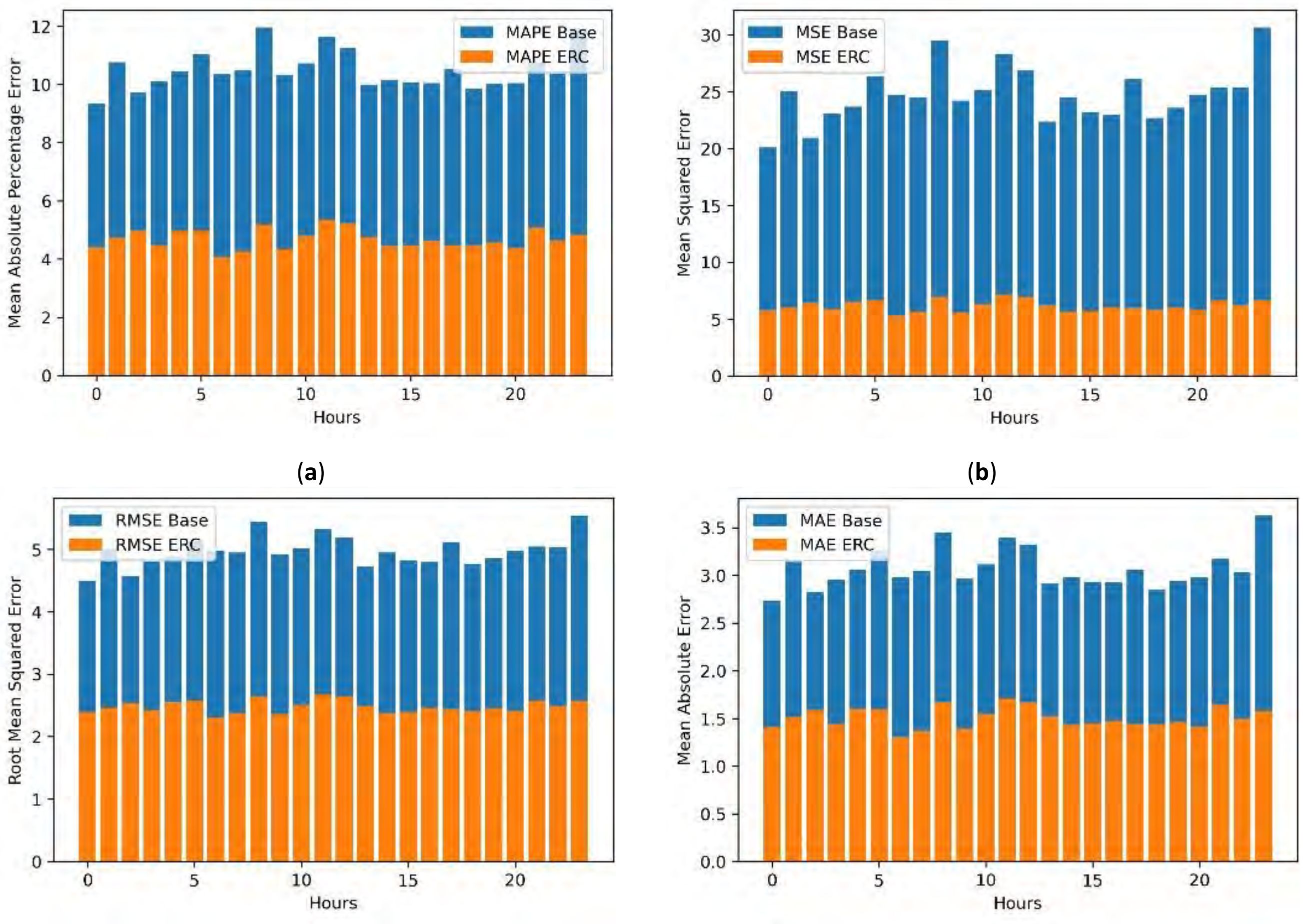




(**c**) (**d**)

Figure 7.5**:** Hourly error metric comparison between the baseline DNN and the ERC–DNN model for 10 training epochs including: (**a**) MAPE; (**b**) MSE; (**c**) RMSE; (**d**) MAE.

The second experiment considers the training scenario of 100 epochs. In this task, the neural network reaches a more acceptable forecasting performance with each hourly sequence having similar error metrics. As can be observed from the subplots of Figure 7.6, there are slight error variations between the hourly sequences showing that the network is still unable to predict every hour of the day-ahead prediction equally well. The effect of error compensation in the ERC–DNN improves the forecasting performance and the error metrics are lower than those presented in the open-access benchmark. Since neural network models on sufficiently large datasets do not typically converge after 100 epochs and the values of error are not distinctly high, the slight error variations observed in the baseline evaluation are passed down to the ERC–DNN. Therefore, when compared to the 10-epoch scenario, the performance of the model improved in a similar way but the stability improvement of error among hourly sequences was not as drastic.



(**a**) (**b**)

(**c**) (**d**)

Figure 7.6**:** Hourly error metric comparison between the baseline DNN and the ERC–DNN model for 100 training epochs including: (**a**) MAPE; (**b**) MSE; (**c**) RMSE; (**d**) MAE.

The third scenario considers 1000 training epochs and refers to models that are near finalization, where the model converges to predicted values close to the target output and the error metrics remain relatively low. Through this experiment, we can observe that the error metrics could follow more consistent patterns, in this case denoting that the first hourly sequences of the day-ahead forecasting task are predicted more accurately when compared to the last few hours. This phenomenon could be a cause of concern when the model is deployed for real-world applications since the model could generate substantially divergent values for the last few hours of each day. The error compensation improves the performance of this model and flattens the previously described effect, resulting in more consistently accurate predictions. However, it is worth noting that as the neural network is close to reaching convergence, the error values are considerably lower, and the overall refinement of predictions is smaller for larger numbers of epochs. The subplots of Figure 7.7 visualize this scenario.

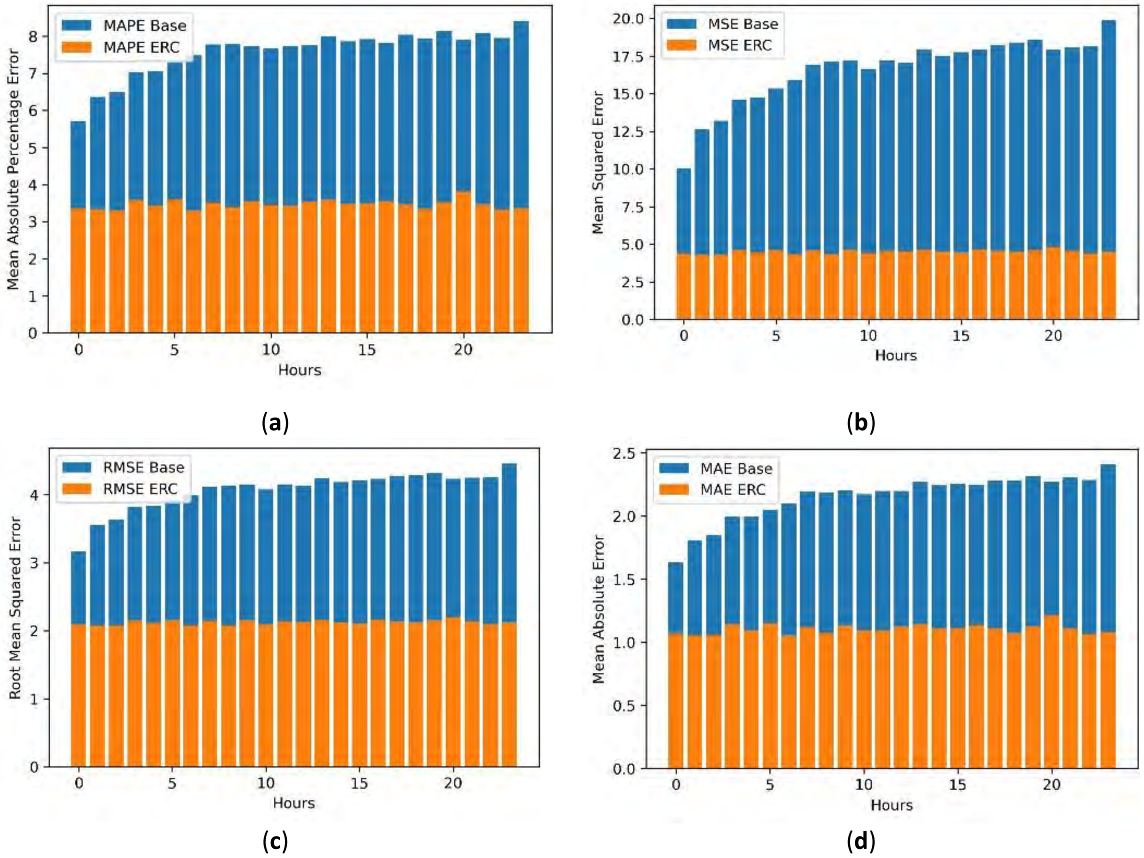


Figure 7.7**:** Hourly error metric comparison between the baseline DNN and the ERC–DNN model for 1000 training epochs including: (**a**) MAPE; (**b**) MSE; (**c**) RMSE; (**d**) MAE.

Overall, we can observe that across all four performance metrics, the integration of the error compensation module refined the predictions and resulted in improved performance in every training scenario, denoting that better and substantially more stable error metrics can be derived even in situations where the neural network is not close to convergence. Table 7.1 presents the overall error metric comparison that cohesively depicts the impact of this post-processing error estimation model.

Table 7.1: Error metrics for the performance evaluation of the baseline DNN and the proposed ERC–DNN.

Model	Scenario	MAPE	MSE	RMSE	MAE
Base DNN	10 Epochs	25.375	130.332	11.194	8.581
ERC–DNN	10 Epochs	6.456	10.367	3.206	2.137
Base-DNN	100 Epochs	10.492	24.761	4.970	3.068
ERC–DNN	100 Epochs	4.688	6.165	2.481	1.507
Base-DNN	1000 Epochs	7.583	16.625	4.067	2.156
ERC–DNN	1000 Epochs	3.464	4.510	2.123	1.105

Hyperparameter optimization considers a large space of training parameters in search of a combination that produces optimal error metrics after training. These parameters are specified before the training process starts and affect the error of the model during the training iterations. After the inspection of the results presented in this work, the argument for the inclusion of parameters that regulate error estimation and affect the error after the initial training is complete, such as the window of lagged observations for the definition of an autoregressive model, or the choice of error estimation method could be valid as future benchmarks could consider the full spectrum of error optimization, in an attempt at setting the new standard for model comparisons, where prediction refinement becomes one of the core final steps. However, expanding the search space and introducing additional hyperparameters is not always a viable option, especially when we consider the potential lack of computing power or the time restrictions imposed by the short recalibration period of real-time models. In this study, the consideration of an autoregressive model utilizing a window of 24 lagged observations for error estimation was a reasonable and computationally inexpensive choice, since the total execution time of the experiments was not dramatically increased. Additionally, the execution of the experiments considered the parameters that could encourage the usage of an autoregressive model for this task, such as the augmented Dickey–Fuller test of stationarity, the computation of partial autocorrelations, and the computation of information criteria for the error estimator.

While the search for the optimal window size based on partial autocorrelations could be regarded as an important step in model selection, and as a potential hyperparameter in more complex optimization problems, the investigation of different model selection criteria could introduce additional hyperparameters is equally necessary. This work explored the information criteria threshold selection method as an alternative to the simpler window selection. The information criteria threshold selection method iteratively fits the autoregressive model using lagged observations that surpass a specified autocorrelation function threshold (ACF). The three information criteria scores of AIC, BIC and HQIC are computed and the model that achieves the lowest score for each hourly error sequence is selected. After examining the scores extracted from this alternative model selection approach in Tables 7.2–7.4, we observed that in the scenario of 10 epochs, where the error compensation model achieves the greatest prediction refinement, not all error sequences led to improved information criteria when lagged observations over a certain autocorrelation threshold were selected since the values depend on the error sequences generated by the DNN. This also holds true for the 100 and 1000 epoch scenarios. Furthermore, the improvement of the information criteria is negligible when compared to the 24-lagged window method. Consequently, in the scenario where all hourly error sequences were able to benefit from the threshold method, the increase in forecasting performance would not be impactful enough to justify the computational burden of iteratively searching for the model that satisfies that criteria. Hence, the simplicity of the window method for autoregressive error estimation would be the preferred method for ERC–DNN and the window size would be an appropriate hyperparameter to tune that model.

Table 7.2: Comparison between the 24-lag window method and the threshold method based on AIC scores for the 10-epoch scenario of ERC–DNN. Cells colored in green denote an improvement in information criteria score while cells colored in blue denote worse overall scores when compared to the window method.

Criterio	**24 Lag Window**	**ACF≥0.2**	**ACF≥0.3**	**ACF≥0.4**
AIC H0	1.9790	1.9292	1.9457	1.9593
AIC H1	2.3797	2.3523	2.3672	2.4724

AIC H2	2.2449	2.2150	2.2319	2.2567
AIC H3	1.9656	1.9224	1.9224	1.9353
AIC H4	1.8745	1.8592	1.8865	1.9663
AIC H5	1.9847	1.9182	1.9391	2.1244
AIC H6	1.8880	1.8673	1.8815	1.9277
AIC H7	2.1403	2.1229	2.1346	2.2355
AIC H8	2.1850	2.1650	2.2633	2.2739
AIC H9	1.7241	1.7407	1.8363	1.8498
AIC H10	1.9550	1.9456	1.9843	2.1282
AIC H11	1.9167	1.8956	1.9077	1.9281
AIC H12	2.1428	2.0839	2.0982	2.1227
AIC H13	1.9478	1.9120	1.9276	1.9573
AIC H14	2.1011	2.0691	2.0691	2.1047
AIC H15	2.0097	1.9600	1.9786	1.9980
AIC H16	1.7016	1.6398	1.6466	1.6662
AIC H17	2.0564	2.0659	2.0754	2.1979
AIC H18	2.1026	2.0382	2.0585	2.0754
AIC H19	2.1297	2.0518	2.0838	2.0931
AIC H20	1.8115	1.7889	1.8569	1.8683
AIC H21	2.3006	2.2009	2.2022	2.2343
AIC H22	1.9503	1.8856	1.8926	1.9041
AIC H23	1.9740	1.9508	1.9687	1.9879

Table 7.3: Comparison between the 24-lag window method and the threshold method based on BIC scores for the 10-epoch scenario of ERC–DNN. Cells colored in green denote an improvement in information criteria score while cells colored in blue denote worse overall scores when compared to the window method.

Criterio	**24 Lag Window**	**ACF≥0.2**	**ACF≥0.3**	**ACF≥0.4**
BIC H0	2.0017	1.9432	1.9736	2.0901
BIC H1	1.9672	1.9173	1.9173	1.9173

BIC H2	1.9315	1.8794	1.8915	1.9250
BIC H3	2.1973	2.1854	2.3153	2.3311
BIC H4	2.0103	1.9635	1.9782	2.0938
BIC H5	1.9218	1.8858	1.8889	1.9136
BIC H6	1.8862	1.8436	1.8553	1.8759
BIC H7	2.2287	2.1715	2.1715	2.1715
BIC H8	2.0533	1.9991	2.0230	2.2113
BIC H9	1.7851	1.7451	1.7534	1.7636
BIC H10	1.9882	1.9916	2.0489	2.0610
BIC H11	1.9267	1.8791	1.9633	2.0198
BIC H12	1.9583	1.8886	1.9166	2.0390
BIC H13	2.2644	2.2398	2.2596	2.2722
BIC H14	1.9082	1.8561	1.8827	1.9004
BIC H15	2.0200	1.9953	2.0252	2.0324
BIC H16	2.2123	2.1749	2.1958	2.2805
BIC H17	1.9061	1.8738	1.9013	1.9500
BIC H18	2.2660	2.2364	2.3039	2.3164
BIC H19	2.2894	2.2566	2.2606	2.2723
BIC H20	2.0338	2.0195	2.0519	2.1707
BIC H21	1.9222	1.8812	1.8894	1.8964
BIC H22	2.1217	2.1079	2.1177	2.1850
BIC H23	2.2551	2.1922	2.2201	2.2400

Table 7.4: Comparison between the 24-lag window method and the threshold method based on HQIC scores for the 10-epoch scenario of ERC–DNN. Cells colored in green denote an improvement in information criteria score while cells colored in blue denote worse overall scores when compared to the window method.

Criterio	**24 Lag Window**	**ACF≥0.2**	**ACF≥0.3**	**ACF≥0.4**
HQIC H0	1.7015	1.6794	1.7168	1.7683

HQIC H1	1.7430	1.7622	1.9189	1.9189
HQIC H2	1.9680	1.8916	1.8972	1.9285
HQIC H3	2.1361	2.1003	2.1188	2.1306
HQIC H4	2.0807	2.0603	2.0747	2.1937
HQIC H5	2.1427	2.0985	2.1003	2.1074
HQIC H6	1.9908	1.9275	1.9384	2.0965
HQIC H7	2.2277	2.1472	2.1683	2.3292
HQIC H8	1.9853	1.9692	2.0512	2.0531
HQIC H9	1.7594	1.7209	1.7608	1.7660
HQIC H10	1.9160	1.9157	2.0202	2.0238
HQIC H11	1.7992	1.7590	1.7652	1.7720
HQIC H12	1.7352	1.6870	1.6892	1.6954
HQIC H13	2.0105	1.9698	1.9835	2.0081
HQIC H14	2.1561	2.1161	2.2623	2.2693
HQIC H15	2.2703	2.2803	2.3123	2.3182
HQIC H16	2.4097	2.3869	2.4141	2.5297
HQIC H17	1.9114	1.8745	1.8745	1.8745
HQIC H18	2.3749	2.4596	2.4653	2.4653
HQIC H19	1.8727	1.8393	1.8588	1.8701
HQIC H20	1.9275	1.8702	1.8853	2.0014
HQIC H21	1.9794	1.9832	2.0800	2.0936
HQIC H22	2.0047	1.9994	2.1103	2.1193
HQIC H23	2.1320	2.0913	2.0997	2.1247

7.5 Discussion

This work presented an error compensation deep neural network for the task of day-ahead electricity price forecasting. The proposed model used an autoregressive module to estimate hourly residual error sequences and refine and improve the predictions of the neural network model. This approach was tested in three different training scenarios, where the values of the error were high, moderate, and low in order to cover several potential network behaviors, ranging from fairly unstable to nearly convergent. The ERC–DNN yielded impressive results, with improved error metrics in every training scenario when compared to the baseline model. In detail, the error compensation method stabilized the performance of the poorly trained network in the first scenario, decreasing the value of MAE from 8.581 to 2.137. Additionally, significant performance improvements were observed in the moderate and the longer training scenarios with the values of MAE decreasing from 3.068 to 1.507 in the 100-epoch experiment and from 2.156 to 1.105 in the 1000-epoch experiment. This forecasting approach resulted in improved error metrics when compared to the benchmark results presented in [307].

The improvement of forecasting performance is not the only benefit provided by this approach, since the error compensation method manages to create more consistent predictions, resulting in multivariate models that can predict each hourly sequence at a similar level of accuracy. The inclusion of an autoregressive module resulted in a clear and interpretable approach to error improvement since it operates on the output of neural networks. Therefore, error estimation and refinement through this approach could be easily associated with the analysis of hourly residual error sequences instead of searching for the optimal combination of structural parameters that configure complex deep neural networks in a black-box approach. The design, implementation and testing of this method provides some useful insights towards the development of more robust and stable hybrid models, as well as the integration of error compensation as an additional optimization option for benchmarks during post-processing. However, one potential disadvantage of this method is the dependence on the error sequences and their characteristics. In this project, we implemented several methods, such as stationarity and autocorrelation analysis to ensure that the autoregressive module would behave appropriately. In scenarios where those methods would yield inconsistent results, this approach may not result in substantial error improvements. As a result, we believe that the analysis of error sequences is a crucial part that precedes the integration of that data in post-processing techniques and should not be omitted. Since most hybrid models and benchmarks utilize hyperparameter optimization to search for the optimal combination of parameters that minimize the error metrics, the integration of error compensation could introduce a wide set of additional parameters that would increase the overall complexity of the models and potentially render that refinement more computationally expensive. While the simple choice of the window size in an autoregressive error estimation model seems to be an appropriate hyperparameter for the configuration of this method, the consideration of more complex estimation methods could result in refinement techniques that greatly hinder the execution time of those models.

The contribution of this work is not limited to the research and development of electricity price forecasting models since there are several ways this approach could benefit market participants and the grid. Firstly, this approach could reduce the price uncertainty of generators while assisting them indirectly in the maximization of profit. Since generators often need to select the highest price after inspecting offers from different markets in order to sell the production [334], this method could lead to more informed decisions due to the increased stability and forecasting performance. Secondly, trading companies could develop more robust short-term contracts due to the availability of more accurate price estimates. Lastly, the grid could benefit from more stable and accurate price predictions since the effect of price volatility could lead to more blackouts and the urgent usage of reserves.

This project attempted to cover several research gaps through the investigation of the error compensation effect on the well-known DNN structure used in open-access benchmarks and several forecasting applications. While recent studies shared a similar direction in the implementation of error compensation on the LSTM structure [253] as well as more traditional statistical methods for different forecasting tasks, this study considered the feedforward deep neural network as the building block for the development of performant forecasting models that include error estimation. The examination of the results in conjunction with recent research findings derived from statistical and machine learning models reinforces the concept that error estimation is a beneficial post-processing technique for deep learning models in the energy sector. There are several additional aspects regarding this method that could be explored in future work. First, a wide comparison of error estimation models ranging from simple statistical approaches to the increasingly complex neural network models could contribute towards the optimal model selection of the error refinement module post-training. Second, the ERC–DNN model could be tested on many electricity markets that display different price characteristics, such as different levels of price fluctuations in an attempt to study the effects of the unique price curve behavior on the training error. Additionally, the inspection of distinctly different error sequences could result in useful insights into the behavior of the model and the adaptability to different market dynamics. Lastly, the benefits of hyperparameter optimization could be studied in combination with error compensation, in an attempt to quantify the overall performance improvement and the computational tradeoff for short-term and real-time applications.

Chapter 8 Conclusions and Future Work

8.1 Summary of Contributions

In this dissertation, the forecasting structure for regression tasks in the energy sector was examined with emphasis on short-term load and price predictions. The main forecasting modules and procedures involved in the forecasting pipeline were highlighted in order to denote the significance of each module and outline the flow of information from the initial data collection to the final output estimation. Through this examination, several challenges and research gaps directly connected to the interpretability, scalability, flexibility and accuracy of the most prominent forecasting methodologies in this research area were identified. In response to those challenges, extensive model comparisons and novel design strategies were developed towards the improvement of the main forecasting modules. The proposed methodologies were tested in use cases where the challenges, performance hinderances and structural intricacies of those components could be easily detected and associated to specific forecasting scenarios.

The study of the preprocessing module highlighted the need for robust feature selection and efficient management of uncertainty. Robust feature selection could lead to dimensionality reduction as well as improved model performance and generalization capability and due to the identification of the most important features. Additionally, since several influential features in the prediction of load and price could include some degree of uncertainty, mechanisms that are capable of generating compact sets of rules could enhance the overall interpretability of the model, assisting in the discovery of optimal model parameters. In this scope, we examined the role of fuzzy inference in forecasting methods for the generation of rules that attempt to explain uncertain features and concluded that while robust feature selection and uncertainty management could be addressed separately, there is a connection between these challenges. Since the studied environments in the energy sector are increasingly complex, the rules utilized for the definition of relationships between variables could cause scalability and interpretability issues when all input variables are considered, resulting in poor estimation performance and rendering some forecasting tasks infeasible. Therefore, our contribution focused on the development of a rule generation strategy that improved upon the prominent linearized tree structure and derived a small and accurate set of rules through the integration of a robust hybrid feature selection method. This approach identified the most important features in the dataset, denoting the impact of efficient feature selection towards dimensionality reduction and utilized them for rule generation, denoting the accurate extraction of relationships that influence the variable of load.

The study of the forecasting framework highlighted several challenges and research gaps with regards to the type of modeling approach utilized for variable estimation. In standalone modeling, most prominent and state-of-the-art models utilized in regression tasks are readily available through several application programming interfaces. However, there is uncertainty surrounding the selection of similarly performant estimators since the learning behavior and error evaluation of those models is not fully explored for all regression tasks through experimentation. Following this observation, it is evident that while prominent standalone neural network structures could be used interchangeably for some forecasting tasks, yielding satisfactory accuracy, there are some edge cases where the usage of some classes of neural networks would not be suitable in terms of convergence time or performance. These edge cases are not properly examined through comprehensive comparisons, leading to confusion and poor decision-making throughout the research process. Consequently, our contribution considers the examination of neural network structures for minutely sampled active power predictions in order to address the edge case of high resolution very short-term predictions where the brevity of the learning process and the substantial adjustment of weights could impact forecasting performance. This research work provided a comprehensive comparison that denoted the superiority of MLP over LSTM and CNN baseline architectures in terms of accuracy and convergence time when the sampling is highly granular for point forecasts. Through this project, the comparison of prominent neural network models reinforced clarity and provided insight towards the behavior of those structures, serving as the building block for more complex architectures in this very short-term scenario.

In combinatorial modeling, the utilization of multiple estimators typically yields improved accuracy and leads to more flexible approaches that could adapt better to the input data. Furthermore, efficient combinatorial modeling could be impactful in forecasting tasks where the varying patterns and distribution shifts introduce the challenges of data and concept drift, offering improved resilience as the parameters of the estimator members are adjusted to optimally fit subsets of data. However, it is evident that the combination of estimators is not always deterministic, resulting in arbitrary decision making, poor reproducibility and interpretability. Therefore, our contribution towards the development and implementation of a novel estimator selection strategy based on structural time series characteristics provided a robust solution towards the generation of estimator sets that best explain the training data and yield improved performance. This project considered the design of stacking and voting estimators since ensemble learning approaches are some of the most prominent methods in combinatorial modeling and provide performance benefits that could be easily monitored. Peak and non-peak indices were the main structural characteristics considered for estimator selection and our methodology denoted that the ensemble approaches generated from the error metric examination of those characteristics achieved the expected performance boost.

In meta-modeling design, the scope shift of the main forecasting structure for the derivation of alternative time series representations and the inclusion of additional forecasting layers that estimate the target variable add significant value to the generalization capabilities of forecasting models and offer increased resilience towards the initial dataset structure through the extraction of knowledge. We observed that meta-modeling principles could be applied to short-term forecasting tasks in the energy sector since the data collected and analyzed for the purposes of real-world applications could have a suboptimal structure, rendering the task of pattern identification increasingly difficult. Additionally, these datasets could contain hidden relationships between time series features that stem from the impact of community influential factors as the data collection process often considers time series of different types of consumers, buildings and energy markets. Our contribution focused on the extraction of knowledge from the examination of consumer similarity and causality for the estimation of alternative load time series representations through LSTM ensembles that were combined to predict the target variable through an MLP generalizer. This approach denoted that the combination of those components vastly improved the error metrics when compared to the base model. This performance improvement highlighted that the extraction of similar and causal load time series representations boosted the accuracy of the base model as the initial dataset that isolated the features for individual consumption exhibited poor quality.

The study of the output module highlighted the intricacies of estimated time series evaluation and interpretation since standalone, combinatorial and meta-modeling estimators could derive suboptimal and unstable predictions due to challenges related to dataset quality and model tuning. The risk of poor performance in a priori processing could be mitigated through the design of complementary a posteriori methods applied to the output module towards the improvement of model metrics and the stabilization of error profiles. It is clear that a posteriori processing methods are not sufficiently presented in recent short-term forecasting research projects in the energy sector and often omitted from relative reviews. Following these observations, our contribution focused on the design of a hybrid estimator that features a deep neural network structure for a priori processing and an autoregressive error compensation module for a posteriori processing. The proposed architecture was applied on the forecasting task of day-ahead electricity price prediction and was compared to a benchmark deep neural network model that shared the same a priori processing structure. The results denoted that the error compensation module improved error metrics and let to more consistent predictions when different training scenarios were considered. Consequently, this a priori strategy highlighted the benefits of residual error estimation, deeming it essential for error refinement. Through the experiments presented in this work, some important observations were made towards hyperparameter tuning for both the a priori and a posteriori processing paradigm. Moreover, this study denoted the performance challenges that may arise due to the expansion of the parameter space with the inclusion of the error compensation module. These challenges are directly connected to the complexity of the forecasting structure and the time needed for recalibration and tuning. The consideration of short-term forecasting horizons imposes time constraints that may render the deployment of more complex estimation structures infeasible. Therefore, the development of balanced hybrid architectures that respect the benefits of both processing paradigms while featuring a parameter space that does not introduce performance bottlenecks at any stage should be the goal for most estimation approaches in the future.

Evidently, the study of recent research projects as well as the experiments conducted for the design and development of novel forecasting strategies indicated that the prominence of data processing methodologies and forecasting structures is directly associated with trade-offs relevant to the structure of the energy time series and the studied research questions. The quality and quantity of the available data denotes the preprocessing methods that need to be included in the forecasting pipeline in order to achieve higher compatibility between the input and the initial model assumptions. Forecasting models that utilize an insufficient number of preprocessing techniques often result in poor training. On the other side of the spectrum, data overprocessing may result in input datasets that no longer capture the unique characteristics and irregularities of the studied time series, resulting in poor generalization. Furthermore, the trade-off between execution time and accuracy influences the selection of processing techniques and estimators since the problem framing process in energy research and the goals of energy applications set specific requirements. Therefore, projects utilizing input at a higher sampling rate and requiring faster recalibration when the available computational power is limited, could benefit from simpler forecasting architectures such as linear regressors, tree-based estimators and their hybrid variants. Alternatively, research work and applications focusing solely on accuracy, could utilize more complex hybrid neural network structures that include meta-modeling techniques and attention mechanisms. Lastly, the trade-off between model complexity and transparency needs to be considered in the development of forecasting pipelines. Transparent architectures that feature enhanced interpretability and explainability typically share a simpler structure and enable the thorough understanding of forecasting mechanisms as well as clearer interpretation of results without requiring an extensive technical background. Consequently, business and consumer-level applications could benefit from less complex and more transparent models since the expected behavior of the model could be easily understood through the processing of sample data. Research efforts tend to focus on structure-agnostic transparency, in an attempt at generating more complex models utilizing interpretable estimation processes deterministically.

8.2 Future Work

The examination of the short-term forecasting pipeline in the energy sector and the contributions presented in this dissertation enable several interesting research directions that could improve the performance of the modules involved in the forecasting structure and introduce more robust design strategies.

In the examination of the preprocessing module our contribution focused on the selection of the most impactful features and the detailed interpretation of influential factors through the generation of rules. Since this study addressed the a priori analysis of features for the derivation of optimal input sets, extensive experiments could be conducted towards the interpretability and explainability of those features from the forecasting models after the training process. The quantification of feature importance and the examination of metrics that specify the degree of attention dedicated to selected sequence segments provide insight towards the thorough understanding of the learning process. Some robust forecasting structures such as the temporal fusion transformer already provide information about feature interpretability through attention graphs and feature importance scores. However, this level of comprehensive data analysis could be extended to other models and side-by-side comparisons could denote the different decisions that led to the convergence of each structure.

In the examination of standalone modeling, our study contributed towards the performance analysis of edge cases that were not sufficiently explored before in terms of accuracy, training behavior and training time, reinforcing the decision-making process for the informed selection of estimators in similar forecasting tasks. Since the process of estimator selection constitutes a wide research topic, future research could focus on the detailed comparison of estimators in terms several complementary aspects such as scalability and the compact analysis of data requirements for optimal training in a plethora of forecasting tasks utilizing energy data. Future research efforts could contribute towards the detailed taxonomy of forecasting methods with regards to those aspects in the form of reviews and benchmarks that consider several edge cases such as minutely forecasting.

Furthermore, our contribution towards combinatorial modeling focused on the development of a deterministic estimator selection strategy for ensemble regressors. This approach considered the cross-examination of structural characteristics such as peak and non-peak data points. This strategy could be extended in future projects in order to examine different time series characteristics connected to the shape of the target sequence and the respective properties of trend and seasonal components. This estimator selection approach could be tested on estimator sets belonging to different statistical and machine learning subcategories. Additionally, the inspection of structural time series characteristics could benefit the design of more complex forecasting structures such as deeper multi-stage ensemble estimators as well as cooperative and sequential hybrid models since clarity needs to be reinforced in the development of those architectures.

The study of meta-modeling and the development of a forecasting approach that utilizes community influential factors for the derivation of alternative target time series representations was tested considering full consumer anonymity and included only the essential features needed for power consumption predictions in order to simulate real-world scenarios where the quality and availability of consumer data are far from ideal. However, the exploration of more descriptive client features could enable the development of more versatile meta-modeling approaches that study impact of community and derive more time series components that could enhance the generalization capabilities of the surrogate model. It is worth mentioning that since the research area of meta-modeling approaches is vast, several novel methodologies, operating on a different context at the base forecasting structure, could be introduced in future projects. One increasingly interesting research direction could consider the estimation of time series that focus on the explanation of trend and seasonality components at the base estimator and the reconstruction of the estimated target at the meta-modeling output.

Lastly, for the improvement of the output module, more robust feedback mechanisms for the interpretation and subsequent minimization of error could be developed. Our contribution introduced a simple autoregressive structure in order to maintain relatively low computational cost and include a small set of additional hyperparameters that are directly connected to the error series but not dependent on the main forecasting structure. Future studies could utilize more interpretable neural network structures for error estimation and introduce hyperparameters that could express the connection between the main forecasting structure and the error estimation module. Furthermore, a thorough time complexity analysis could outline the cost of combining error estimation mechanisms with several prominent forecasting structures in order to suggest performant and scalable architectures that result in sufficient error stability for each forecasting horizon. In this scope the application and impact of error estimation modules in long term energy forecasting tasks could be worth examining since error values tend to be higher and less consistent as the prediction horizon increases.

Publications

1. Kontogiannis, D.; Bargiotas, D.; Daskalopulu, A. Minutely Active Power Forecasting Models Using Neural Networks. *Sustainability* **2020**, *12*, 3177. https://doi.org/10.3390/su12083177
2. Kontogiannis, D.; Bargiotas, D.; Daskalopulu, A. Fuzzy Control System for Smart Energy Management in Residential Buildings Based on Environmental Data. *Energies* **2021**, *14*, 752. https://doi.org/10.3390/en14030752
3. Kontogiannis, D.; Bargiotas, D.; Daskalopulu, A.; Tsoukalas, L.H. A Meta-Modeling Power Consumption Forecasting Approach Combining Client Similarity and Causality. *Energies* **2021**, *14*, 6088. https://doi.org/10.3390/en14196088
4. Kontogiannis, D.; Bargiotas, D.; Daskalopulu, A.; Arvanitidis, A.I.; Tsoukalas, L.H. Error Compensation Enhanced Day-Ahead Electricity Price Forecasting. *Energies* **2022**, *15*, 1466. https://doi.org/10.3390/en15041466
5. Kontogiannis, D.; Bargiotas, D.; Daskalopulu, A.; Arvanitidis, A.I.; Tsoukalas, L.H. Structural Ensemble Regression for Cluster-Based Aggregate Electricity Demand Forecasting. *Electricity* **2022**, *3*, 480-504. https://doi.org/10.3390/electricity3040025
6. Kontogiannis, D.; Bargiotas, D.; Daskalopulu, A.; Tsoukalas, L.H. Explainability Analysis of Weather Variables in Short-Term Load Forecasting. Submitted to IISA 2023 (Under Review)

Contributions that indirectly influenced this dissertation

1. Arvanitidis, A.I.; Bargiotas, D.; Daskalopulu, A.; Kontogiannis, D.; Panapakidis, I.P.; Tsoukalas, L.H. Clustering Informed MLP Models for Fast and Accurate Short-Term Load Forecasting. *Energies* **2022**, *15*, 1295. https://doi.org/10.3390/en15041295
2. Arvanitidis, A.I.; Bargiotas, D.; Kontogiannis, D.; Fevgas, A.; Alamaniotis, M. Optimized Data-Driven Models for Short-Term Electricity Price Forecasting Based on Signal Decomposition and Clustering Techniques. *Energies* **2022**, *15*, 7929. https://doi.org/10.3390/en15217929
3. Arvanitidis, A. I.; Kontogiannis, D.; Vontzos, G.; Laitsos, V.; Bargiotas, D. Stochastic Heuristic Optimization of Machine Learning Estimators for Short-Term Wind Power Forecasting. *2022 57th International Universities Power Engineering Conference (UPEC)* **2022**. https://doi.org/10.1109/upec55022.2022.9917957
4. Arvanitidis, A.I.; Kontogiannis, D.; Vontzos, G.; Laitsos, V.; Bargiotas, D.; Alamaniotis, M. Performance Analysis of Single and Multi-Step Short-Term Load Forecasts Using Multi-Layer Perceptron. ENERGY 2023: The Thirteenth International Conference on Smart Grids, Green Communications and IT Energy-aware Technologies. Available online: https://www.thinkmind.org/articles/energy_2023_1_110_30063.pdf

References

[1] Henderson, M. I.; Novosel, D.; Crow, M. L. Electric Power Grid Modernization Trends, Challenges, and Opportunities. *IEEE* **2017**. Available online: https://www.ieee.org/content/dam/ieee-org/ieee/web/org/about/corporate/ieee-industry-advisory-board/electric-power-grid-modernization.pdf (accessed on 8 December 2022)

[2] Meeus, L. *The evolution of electricity markets in Europe*; Edward Elgar: Cheltenham, **2020**. Available online: https://cadmus.eui.eu/bitstream/handle/1814/69266/9781789905465%20ebook.pdf?sequence=4&isAllowed=y (accessed on 8 December 2022)

[3] Salehimehr, S.; Taheri, B.; Sedighizadeh, M. Short-term load forecasting in smart grids using artificial intelligence methods: A survey. *The Journal of Engineering* 2022, ***2022***, 1133–1142. https://doi.org/10.1049/tje2.12183

[4] Mandal, P.; Senjyu, T.; Urasaki, N.; Funabashi, T.; Srivastava, A. K. Short-term price forecasting for competitive electricity market. *2006 38th North American Power Symposium* **2006**. https://doi.org/10.1109/NAPS.2006.360135

[5] Elias, R. J.; Montgomery, D. C.; Kulahci, M. An overview of short-term statistical forecasting methods. *International Journal of Management Science and Engineering Management* **2006**, *1*, 17–36. https://doi.org/10.1080/17509653.2006.10670994

[6] Zor, K.; Timur, O.; Teke, A. A state-of-the-art review of artificial intelligence techniques for short-term electric load forecasting. *2017 6th International Youth Conference on Energy (IYCE)* **2017**. https://doi.org/10.1109/IYCE.2017.8003734

[7] Runge, J.; Zmeureanu, R.; Le Cam, M. Hybrid short-term forecasting of the electric demand of supply fans using machine learning. *Journal of Building Engineering* **2020**, *29*, 101144. https://doi.org/10.1016/j.jobe.2019.101144

[8] Zhou, Y.; Liang, Y.; Pan, Y.; Yuan, X.; Xie, Y.; Jia, W. A Deep-Learning-Based Meta-Modeling Workflow for Thermal Load Forecasting in Buildings: Method and a Case Study. *Buildings* **2022**, *12*, 177. https://doi.org/10.3390/buildings12020177

[9] Ribeiro, A.M.N.C.; do Carmo, P.R.X.; Endo, P.T.; Rosati, P.; Lynn, T. Short- and Very Short-Term Firm-Level Load Forecasting for Warehouses: A Comparison of Machine Learning and Deep Learning Models. *Energies* **2022**, *15*, 750. https://doi.org/10.3390/en15030750

[10] Guerard, J. B. Regression analysis and forecasting models. *Introduction to Financial Forecasting in Investment Analysis* **2012**, 19–45. 10.1007/978-1-4614-5239-3_2

[11] Tsekouras, G. J.; Salis, A. D.; Tsaroucha, M. A.; Karanasiou, I. S. Load Time-series classification based on Pattern Recognition Methods. *Pattern Recognition Techniques, Technology and Applications* **2008**. 10.5772/6250

[12] De Greve, Z.; Lecron, F.; Vallee, F.; Mor, G.; Perez, D.; Danov, S.; Cipriano, J. Comparing time-series clustering approaches for individual electrical load patterns. *CIRED – Open Access Proceedings Journal* **2017**, *2017*, 2165–2168. 10.1049/oap-cired.2017.1222

[13] Characteristics of Time Series. Available online: https://www.ibm.com/docs/en/spss-modeler/saas?topic=data-characteristics-time-series (accessed on 8 December 2022).

[14] Das, S. 10 frequently encountered issues in data preprocessing. Available online: https://www.analyticsvidhya.com/blog/2022/08/10-frequently-encountered-issues-in-data-preprocessing/ (accessed on 8 December 2022).

[15] Shah, T. About train, validation and test sets in machine learning. Available online:https://towardsdatascience.com/train-validation-and-test-sets-72cb40cba9e7 (accessed on 8 December 2022).

[16] Wang, H.; Alattas, K. A.; Mohammadzadeh, A.; Sabzalian, M. H.; Aly, A. A.; Mosavi, A. Comprehensive review of load forecasting with emphasis on intelligent computing approaches. *Energy Reports* **2022**, *8*, 13189–13198. https://doi.org/10.1016/j.egyr.2022.10.016

[17] Moon, J.; Jung, S.; Rew, J.; Rho, S.; Hwang, E. Combination of short-term load forecasting models based on a stacking ensemble approach. *Energy and Buildings* **2020**, *216*, 109921. 10.1016/j.enbuild.2020.109921

[18] Matijaš, M.; Suykens, J. A. K.; Krajcar, S. Load forecasting using a multivariate meta-learning system. *Expert Systems with Applications* **2013**, *40*, 4427–4437. 10.1016/j.eswa.2013.01.047

[19] Javed, U.; Ijaz, K.; Jawad, M.; Ansari, E.A.; Shabbir, N.; Kütt, L.; Husev, O. Exploratory Data Analysis Based Short-Term Electrical Load Forecasting: A Comprehensive Analysis. *Energies* **2021**, *14*, 5510. https://doi.org/10.3390/en14175510

[20] IEA, Energy end-use data collection methodologies and the emerging role of digital technologies, *IEA* **2020**, Paris. Available online: https://www.iea.org/reports/energy-end-use-data-collection-methodologies-and-the-emerging-role-of-digital-technologies (accessed on 8 December 2022).

[21] The use of administrative sources in energy data collection in Ghana. Available online: https://studylib.net/doc/16698354/the-use-of-administrative-sources-in-energy-data-collecti... (accessed on 8 December 2022).

[22] Manual for Statistics on energy consumption in households, Publications Office of the European Union, Luxembourg, **2013**. Doi:10.2785/45686

[23] Parker, S. A.; Hunt, W. D.; McMordie Stoughton, K.; Boyd, B. K.; Fowler, K. M.; Koehler, T. M.; Sandusky, W. F.; Sullivan, G. P.; Pugh, R. Metering best practices: A guide to achieving utility resource efficiency, release 3.0. **2015**. https://doi.org/10.2172/1178500

[24] Swan, L. G.; Ugursal, V. I. Modeling of end-use energy consumption in the residential sector: A review of Modeling Techniques. *Renewable and Sustainable Energy Reviews* **2009**, *13*, 1819–1835. https://doi.org/10.1016/j.rser.2008.09.033

[25] D'Agostino, D.; Parker, D.; Epifani, I.; Crawley, D.; Lawrie, L. Datasets on Energy Simulations of Standard and Optimized Buildings under Current and Future Weather Conditions across Europe. *Data* **2022**, *7*, 66. https://doi.org/10.3390/data7050066

[26] Neshat, N.; Amin-Naseri, M. R.; Danesh, F. Energy models: Methods and characteristics. *Journal of Energy in Southern Africa* **2014**, *25*, 101–111. 10.17159/2413-3051/2014/v25i4a2243

[27] Dong, Y.; Peng, C.-Y. J. Principled missing data methods for researchers. *SpringerPlus* **2013**, *2*. 10.1186/2193-1801-2-222

[28] Wang, X.; Wang, C. Time Series Data Cleaning with Regular and Irregular Time Intervals. *arXiv* **2020**. https://doi.org/10.48550/arXiv.2004.08284

[29] Saad, M.; Chaudhary, M.; Karray, F.; Gaudet, V. Machine learning based approaches for imputation in time series data and their impact on forecasting. *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* **2020**. 10.1109/smc42975.2020.9283191

[30] Chen, X. Temporal matrix factorization for multivariate time series forecasting. Available online: https://medium.com/@xinyu.chen/temporal-matrix-factorization-for-multivariate-time-series-forecasting-b1c59faf05ea (accessed on 8 December 2022).

[31] Geng, R.; Cao, J.; Zhao, Q.; Wang, Y. Time series data imputation using expectation-maximization with principal component analysis. *IoT and Big Data Technologies for Health Care* **2022**, 352–357. 10.1007/978-3-030-94182-6_26

[32] Serrano Ardila, V.M.; Maciel, J.N.; Ledesma, J.J.G.; Ando Junior, O.H. Fuzzy Time Series Methods Applied to (In)Direct Short-Term Photovoltaic Power Forecasting. *Energies* **2022**, *15*, 845. https://doi.org/10.3390/en15030845

[33] Brownlee, J. Basic Feature Engineering with time series data in Python. Available online: https://machinelearningmastery.com/basic-feature-engineering-time-series-data-python/ (accessed on 8 December 2022).

[34] Cai, S.; Gallina, B.; Nyström, D.; Seceleanu, C. Data Aggregation Processes: A survey, a taxonomy, and design guidelines. *Computing* **2018**, *101*, 1397–1429. 10.1007/s00607-018-0679-5

[35] Data Science in 5 minutes: What is one hot encoding? Available online: https://www.educative.io/blog/one-hot-encoding (accessed on 8 December 2022).

[36] Encoding cyclical features for Deep Learning. Available online: https://www.avanwyk.com/encoding-cyclical-features-for-deep-learning/ (accessed on 8 December 2022).

[37] Lewinson, E. Three approaches to feature engineering for Time Series. Available online: https://towardsdatascience.com/three-approaches-to-feature-engineering-for-time-series-2123069567be (accessed on 8 December 2022).

[38] Introduction to dimensionality reduction. Available online: https://www.geeksforgeeks.org/dimensionality-reduction/ (accessed on 8 December 2022).

[39] Train test validation split: How to & best practices [2022]. Available online: https://www.v7labs.com/blog/train-validation-test-set (accessed on 8 December 2022).

[40] Garza, A. Segmentation analysis with K-means clustering. Available online: https://medium.com/analytics-vidhya/segmentation-analysis-with-kmeans-clustering-93d05565a9f8 (accessed on 8 December 2022).

[41] Brownlee, J. How to use power transforms for time series forecast data with python. Available online: https://machinelearningmastery.com/power-transform-time-series-forecast-data-python/ (accessed on 8 December 2022).

[42] Differencing (of time series). Available online: https://www.statistics.com/glossary/differencing-of-time-series/ (accessed on 8 December 2022).

[43] Normalization vs standardization. Available online: https://www.geeksforgeeks.org/normalization-vs-standardization/ (accessed on 8 December 2022).

[44] Asesh, A. Normalization and bias in time series data. *Digital Interaction and Machine Intelligence* **2022**, 88–97. 10.1007/978-3-031-11432-8_8

[45] Brownlee, J. How to decompose time series data into trend and seasonality. Available online: https://machinelearningmastery.com/decompose-time-series-data-trend-seasonality/ (accessed on 8 December 2022).

[46] Fang, Q. S.; Zhong, Y. X.; Xie, C. L.; Zhang, H. Y.; Li, S. S. Research on PCA-LSTM-based short-term load forecasting method. *IOP Conference Series: Earth and Environmental Science* **2020**, *495*, 012015. 10.1088/1755-1315/495/1/012015

[47] Li, H.; Yang, L.; Guo, C. Improved piecewise vector quantized approximation based on normalized time subsequences. *Measurement* **2013**, *46*, 3429–3439. 10.1016/j.measurement.2013.05.012

[48] Wang, L.; Tian, T.; Xu, H.; Tong, H. Short-term power load forecasting model based on T-Sne Dimension Reduction Visualization analysis, VMD and LSSVM improved with Chaotic Sparrow search algorithm optimization. *Journal of Electrical Engineering & Technology* **2022**, *17*, 2675–2691. 10.1007/s42835-022-01101-7

[49] McInnes, L.; Healy, J.; Saul, N.; Großberger, L. UMAP: Uniform manifold approximation and projection. *Journal of Open Source Software* **2018**, *3*, 861. 10.21105/joss.00861

[50] Ashfaq, T.; Javaid, N. Short-term electricity load and price forecasting using enhanced KNN. *2019 International Conference on Frontiers of Information Technology (FIT)* **2019**. 10.1109/fit47737.2019.00057

[51] How to calculate feature importance leveraging python. Available online: https://aicorespot.io/%E2%80%8Bhow-to-calculate-feature/ (accessed on 8 December 2022).

[52] Li, S. A quick introduction on Granger causality testing for time series analysis. Available online: https://towardsdatascience.com/a-quick-introduction-on-granger-causality-testing-for-time-series-analysis-7113dc9420d2 (accessed on 8 December 2022).

[53] Serrà, J.; Arcos, J. L. An empirical evaluation of similarity measures for time series classification. *Knowledge-Based Systems* **2014**, *67*, 305–314. 10.1016/j.knosys.2014.04.035

[54] Nettleton, D. Selection of variables and factor derivation. *Commercial Data Mining* **2014**, 79–104. 10.1016/b978-0-12-416602-8.00006-6

[55] Ye, J.; Xiao, C.; Esteves, R. M.; Rong, C. Time series similarity evaluation based on Spearman's correlation coefficients and distance measures. *Cloud Computing and Big Data* **2015**, 319–331. 10.1007/978-3-319-28430-9_24

[56] Lendave, V. What are autocorrelation and partial autocorrelation in time series data? Available online: https://analyticsindiamag.com/what-are-autocorrelation-and-partial-autocorrelation-in-time-series-data/ (accessed on 8 December 2022).

[57] Vandeput, N. Simple exponential smoothing. Available online: https://towardsdatascience.com/simple-exponential-smoothing-749fc5631bed (accessed on 8 December 2022).

[58] Holt-winters exponential smoothing. Available online: https://timeseriesreasoning.com/contents/holt-winters-exponential-smoothing/ (accessed on 8 December 2022).

[59] Billah, B.; Hyndman, R. J.; Koehler, A. B. Empirical information criteria for time series Forecasting model selection. *Journal of Statistical Computation and Simulation* **2005**, *75*, 831–840. 10.1080/00949650410001687208

[60] Linear regression. Available online: https://www.mastersindatascience.org/learning/machine-learning-algorithms/linear-regression/ (accessed on 8 December 2022).

[61] Oleszak, M. Regularization Tutorial: Ridge, Lasso & Elastic Net Regression. Available online: https://www.datacamp.com/tutorial/tutorial-ridge-lasso-elastic-net (accessed on 8 December 2022).

[62] Clyde, M.; Çetinkaya-Rundel, M.; Rundel, C.; Banks, D.; Chai, C.; Huang, L. An introduction to bayesian thinking. Available online: https://statswithr.github.io/book/introduction-to-bayesian-regression.html (accessed on 8 December 2022).

[63] Forecasting: Principles and practice (2nd ed). Available online: https://otexts.com/fpp2/arima.html (accessed on 8 December 2022).

[64] DataTechNotes Regression example with SGDRegressor in python. Available online: https://www.datatechnotes.com/2020/09/regression-example-with-sgdregressor-in-python.html (accessed on 8 December 2022).

[65] 1.4. Support Vector Machines. Available online: https://scikit-learn.org/stable/modules/svm.html#svm-regression (accessed on 8 December 2022).

[66] Decision Tree Regression Available online: https://www.saedsayad.com/decision_tree_reg.htm (accessed on 8 December 2022).

[67] Chen, T.; Guestrin, C. XGBoost. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* **2016**. 10.1145/2939672.2939785

[68] Madhiarasan, M.; Louzazni, M. Different Forecasting Horizons Based Performance Analysis of Electricity Load Forecasting Using Multilayer Perceptron Neural Network. *Forecasting* **2021**, *3*, 804-838. https://doi.org/10.3390/forecast3040049

[69] Lang, C.; Steinborn, F.; Steffens, O.; Lang, E. W. Applying a 1D-CNN network to electricity load forecasting. *Contributions to Statistics* **2020**, 205–218. 10.1007/978-3-030-56219-9_14

[70] Zhang, L.; Yang, L.; Gu, C.; Li, D. LSTM-based short-term electrical load forecasting and anomaly correction. *E3S Web of Conferences* **2020**, *182*, 01004. 10.1051/e3sconf/202018201004

[71] Ke, K.; Hongbin, S.; Chengkang, Z.; Brown, C. Short-term electrical load forecasting method based on stacked auto-encoding and gru neural network. *Evolutionary Intelligence* **2019**, *12*, 385–394. 10.1007/s12065-018-00196-0

[72] Banga, A.; Sharma, S. C.; Ahuja, R. Stacking machine learning models to forecast hourly and daily electricity consumption of household using internet of things. *Journal of Scientific & Industrial Research* **2021**, *80*. 10.56042/jsir.v80i10.42241

[73] Sarkar, D.; AO, T.; Gunturi, S. K. Bootstrap aggregating approach to short-term load forecasting using meteorological parameters for demand side management in the north-eastern region of India. **2021**. 10.21203/rs.3.rs-610295/v1

[74] Wu, D.; Wang, B.; Precup, D.; Boulet, B. Boosting based multiple kernel learning and transfer regression for electricity load forecasting. *Machine Learning and Knowledge Discovery in Databases* **2017**, 39–51. 10.1007/978-3-319-71273-4_4

[75] Tseng, F.-M.; Tzeng, G.-H. A fuzzy seasonal Arima model for forecasting. *Fuzzy Sets and Systems* **2002**, *126*, 367–376. 10.1016/s0165-0114(01)00047-1

[76] Fullér, R. Introduction to neuro-fuzzy systems. **2000**. 10.1007/978-3-7908-1852-9

[77] Jallal, M. A.; González-Vidal, A.; Skarmeta, A. F.; Chabaa, S.; Zeroual, A. A hybrid neuro-fuzzy inference system-based algorithm for time series forecasting applied to energy consumption prediction. *Applied Energy* **2020**, *268*, 114977. 10.1016/j.apenergy.2020.114977

[78] Zhang, H.; Li, S.; Chen, Y.; Dai, J.; Yi, Y. A novel encoder-decoder model for multivariate time series forecasting. *Computational Intelligence and Neuroscience* **2022**, *2022*, 1–17. 10.1155/2022/5596676

[79] Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Advances in neural information processing systems* **2017**, *30*. https://doi.org/10.48550/arXiv.1706.03762

[80] Victor, A. What is the attention mechanism in deep learning? Available online: https://insights.daffodilsw.com/blog/what-is-the-attention-mechanism-in-deep-learning (accessed on 8 December 2022).

[81] Hartmann, T.; Moawad, A.; Schockaert, C.; Fouquet, F.; Le Traon, Y. Meta-modelling meta-learning. *2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems (MODELS)* **2019**. 10.1109/models.2019.00014

[82] Hyperparameter tuning using grid search and random search in python. Available online: https://www.kdnuggets.com/2022/10/hyperparameter-tuning-grid-search-random-search-python.html (accessed on 8 December 2022).

[83] Gülcü, A.; Kuş, Z. Multi-objective simulated annealing for hyper-parameter optimization in Convolutional Neural Networks. *PeerJ Computer Science* **2021**, *7*. 10.7717/peerj-cs.338

[84] Koehrsen, W. A conceptual explanation of Bayesian hyperparameter optimization for Machine Learning. Available online: https://towardsdatascience.com/a-conceptual-explanation-of-bayesian-model-based-hyperparameter-optimization-for-machine-learning-b8172278050f (accessed on 8 December 2022).

[85] Optimization and hyper-parameter tuning with genetic algorithm. Available online: https://algoritmaonline.com/optimization-with-genetic-algorithm/ (accessed on 8 December 2022).

[86] Bakhteev, O. Y.; Strijov, V. V. Comprehensive analysis of gradient-based hyperparameter optimization algorithms. *Annals of Operations Research* **2019**, *289*, 51–65. 10.1007/s10479-019-03286-z

[87] Fürnkranz, J.; Chan, P.; Craw, S.; Sammut, C.; Uther, W.; Ratnaparkhi, A.; Jin, X.; Han, J.; Yang, Y.; Morik, K.; et al. Mean Absolute Error. In *Encyclopedia of Machine Learning*; Springer: Boston, MA, USA, 2011; p. 652. https://doi.org/10.1007/978-0-387-30164-8_525.

[88] de Myttenaere, A.; Golden, B.; Le Grand, B.; Rossi, F. Mean Absolute Percentage Error for regression models. *Neurocomputing* **2016**, *192*, 38–48. https://doi.org/10.1016/j.neucom.2015.12.114.

[89] Chaudhary, M. Python|Mean Squared Error—GeeksforGeeks, GeeksforGeeks. 2019. Available online: https://www.geeksforgeeks.org/python-mean-squared-error/ (accessed on 14 January 2022).

[90] Hyndman, R.; Koehler, A.B. Another look at measures of forecast accuracy. *Int. J. Forecast.* **2006**, *22*, 679–688. https://doi.org/10.1016/j.ijforecast.2006.03.001.

[91] Dimid, D. N. Overfitting and underfitting principles. Available online: https://towardsdatascience.com/overfitting-and-underfitting-principles-ea8964d9c45c (accessed on 8 December 2022).

[92] DeepAI Early stopping. Available online: https://deepai.org/machine-learning-glossary-and-terms/early-stopping-machine-learning (accessed on 8 December 2022).

[93] Kumar, A. Hold-out method for training machine learning models. Available online: https://vitalflux.com/hold-out-method-for-training-machine-learning-model/ (accessed on 8 December 2022).

[94] Time series cross-validation. Available online: https://tscv.readthedocs.io/en/latest/ (accessed on 8 December 2022).

[95] Baeldung. Learning curves in machine learning. Available online: https://www.baeldung.com/cs/learning-curve-ml (accessed on 8 December 2022).

[96] Tannor, P. Data Drift vs. concept drift. Available online: https://deepchecks.com/data-drift-vs-concept-drift-what-are-the-main-differences/ (accessed on 8 December 2022).

[97] Saikia, P. Data Drift Detection: Importance of Data Drift Detection. Available online: https://www.analyticsvidhya.com/blog/2021/10/mlops-and-the-importance-of-data-drift-detection/ (accessed on 8 December 2022).

[98] Mihai, C.; Ilea, D.; Mircea, P. Use of load profile curves for the energy market. In Proceedings of the 2016 International Conference on Development and Application Systems (DAS), Suceava, Romania, 19–21 May 2016, doi:10.1109/daas.2016.

[99] Muratori, M.; Rizzoni, G. Residential Demand Response: Dynamic Energy Management and Time-Varying Electricity Pricing, *IEEE Trans. Power Syst.* **2016**, *31*, 1108–1117, doi:10.1109/tpwrs.2015.2414880.

[100] Verleysen, M.; François, D. The Curse of Dimensionality in Data Mining and Time Series Prediction. In *Computational Intelligence and Bioinspired Systems;Springer Berlin Heidelberg, Berlin, Germany,* 2005; pp. 758–770, doi:10.1007/11494669_93.

[101] Bracke, D. Limitations of Traditional Statistical Forecasting Techniques—Solventure LIFe. Solventure LIFe, 2020. Available online:https://life.solventuregroup.com/2020/05/27/limitations-of-forecasting-techniques/ (accessed on 16 November 2020).

[102] Gil, M.Á.; Hryniewicz, O. Statistics with Imprecise Data. In *Computational Complexity*; Springer New York, New York, USA, 2012; pp. 3052–3063, doi:10.1007/978-1-4614-1800-9_190.

[103] Fuzzy Logic in Artificial Intelligence: Architecture, Applications, Advantages & Disadvantages. upGrad blog, 2020. Available online:https://www.upgrad.com/blog/fuzzy-login-in-artificial-intelligence/ (accessed on 16 November 2020).

[104] Zeng, Y.; Wang, L.; Xu, X. An integrated model to select an erp system for chinese small- and medium-sized enterprise under uncertainty. *Technol. Econ. Dev. Econ.* **2015**, *23*, 38–58, doi:10.3846/20294913.2015.1072748.

[105] Mani, P.; Joo, Y. Fuzzy logic-based integral sliding mode control of multi-area power systems integrated with wind farms. *Inf. Sci.* **2021**, *545*, 153–169, doi:10.1016/j.ins.2020.07.076.

[106] Mori, H. Fuzzy neural network applications to power systems. In Proceedings of the 2000 IEEE Power Engineering Society Winter Meeting. Conference Proceedings (Cat. No.00CH37077), Singapore, 23–27 January 2000, doi:10.1109/pesw.2000.850130.

[107] Azadeh, A.; Saberi, M.; Ghaderi, S.; Gitiforouz, A.; Ebrahimipour, V. Improved estimation of electricity demand function by integration of fuzzy system and data mining approach. *Energy Convers. Manag.* **2008**, *49*, 2165–2177, doi:10.1016/j.enconman.2008.02.021.

[108] Lau, H.; Cheng, E.; Lee, C.; Ho, G. A fuzzy logic approach to forecast energy consumption change in a manufacturing system. *Expert Syst. Appl.* **2008**, *34*, 1813–1824, doi:10.1016/j.eswa.2007.02.015.

[109] Suganthi, L.; Iniyan, S.; Samuel, A. Applications of fuzzy logic in renewable energy systems—A review. *Renew. Sustain. Energy Rev.* **2015**, *48*, 585–607, doi:10.1016/j.rser.2015.04.037.

[110] Emagbetere, E.; ThankGod, E.; Iyabo, S. Fuzzy Based System for Power Consumption Prediction. *Int. J. Eng. Sci. Comput.* **2017**, *7*, 4353–4361.

[111] Javaid, S.; Javaid, N.; Iqbal, S.; Mughal, M. Controlling energy consumption with the world-wide adaptive thermostat using fuzzy inference system in smart grid. In Proceedings of the 2017 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, South Korea, 18–20 October 2017, doi:10.1109/ictc.2017.8190944.

[112] Zhang, R.; Ashuri, B.; Deng, Y. A novel method for forecasting time series based on fuzzy logic and visibility graph. *Adv. Data Anal. Cl.* **2017**, *11*, 759–783, doi:10.1007/s11634-017-0300-3.

[113] Bissey, S.; Jacques, S.; le Bunetel, J. The Fuzzy Logic Method to Efficiently Optimize Electricity Consumption in Individual Housing. *Energies* **2017**, *10*, 1701, doi:10.3390/en10111701.

[114] Krishna, P.; Gupta, S.; Shankaranarayanan, P.; Sidharth, S.; Sirphi, M. Fuzzy Logic Based Smart Home Energy Management System. In Proceedings of the 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Bangalore, India, 10–12 July 2018, doi:10.1109/icccnt.2018.8493744.

[115] Nebot, À.; Mugica, F. Energy Performance Forecasting of Residential Buildings Using Fuzzy Approaches. *Appl. Sci.* **2020**, *10*, 720, doi:10.3390/app10020720.

[116] Hüllermeier, E. Fuzzy Logic in Machine Learning. Department of Computer Science Paderborn University, 2017. Available online: https://eventos.citius.usc.es/evia2017/presentations/EVIA2017%20-%20Wednesday%20-%2003%20-%20Eyke%20Hullermeier%20-%20Fuzzy%20Logic%20in%20Machine%20Learning.pdf (accessed on 16 November 2020).

[117] Izquierdo, S.; Izquierdo, L. Mamdani Fuzzy Systems for Modelling and Simulation: A Critical Assessment. *J. Artif. Soc. Soc. Simul.* **2018**, *21*, doi:10.18564/jasss.3660.

[118] Novák, V.; Perfilieva, I. The Principles of Fuzzy Logic: Its Mathematical and Computational Aspects. In *Lectures on Soft Computing and Fuzzy Logic*; *Physica-Verlag HD, Heidelberg, Germany,* 2001; pp. 189–237, doi:10.1007/978-3-7908-1818-5_12.

[119] Ahmed, H.Y. Re: Is There a Limit on the Number of Output Membership Functions (mf) for Fuzzy Logic Controller? Can We Use 100 mf? 2015. Available online: https://www.researchgate.net/post/Is-there-a-limit-on-the-number-of-output-membership-functions-mf-for-Fuzzy-Logic-Controller-Can-we-use-100-

mf/557c5d135cd9e354bc8b45da/citation/download (accessed on 16 November 2020).

[120] "eMathTeacher: Mamdani's Fuzzy Inference Method—Fuzzy Operators. Dma.fi.upm.es, **2020**. Available online: http://www.dma.fi.upm.es/recursos/aplicaciones/logica_borrosa/web/fuzzy_inferencia/fuzzyop_en.htm (accessed on 16 November 2020).

[121] Hall, L.; Lande, P. Generation of Fuzzy Rules from Decision Trees. *J. Adv. Comput. Intell. Intell. Inf.* **1998**, *2*, 128–133, doi:10.20965/jaciii.1998.p0128.

[122] Sitompul, O.; Nababan, E.; Alim, Z. Adaptive distributed grid-partition in generating fuzzy rules. In Proceedings of the 2017 11th International Conference on Information & Communication Technology and System (ICTS), Surabaya, Indonesia, 31 October **2017**, doi:10.1109/icts.2017.8265656.

[123] Zhang, H.; Zhang, B.; Wang, F. Automatic Fuzzy Rules Generation Using Fuzzy Genetic Algorithm. In Proceedings of the 2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery, Tianjin, China, 14–16 August **2009**, doi:10.1109/fskd.2009.420.

[124] Shi, Y.; Mizumoto, M.; Shi, P. Fuzzy if-then rule generation based on neural network and clustering algorithm techniques. In Proceedings of the 2002 IEEE Region 10 Conference on Computers, Communications, Control and Power Engineering. TENCOM '02. Proceedings, Beijing, China, 28–31 October **2002**, doi:10.1109/tencon.2002.1181358.

[125] Leekwijck, W.; Kerre, E. Defuzzification: Criteria and Classification. *Fuzzy Sets Syst.* **1999**, *108*, 159–178, doi:10.1016/s0165-0114(97)00337-0.

[126] What is One Hot Encoding and How to Do It. Medium, **2020**. Available online: https://medium.com/@michaeldelsole/what-is-one-hot-encoding-and-how-to-do-it-f0ae272f1179 (accessed on 13 December 2020).

[127] "dimkonto/Fuzzy-Energy-System", GitHub, **2020**. Available online: https://github.com/dimkonto/Fuzzy-Energy-System (accessed on 18 November 2020).

[128] Candanedo, L.; Feldheim, V.; Deramaix, D. Data driven prediction models of energy use of appliances in a low-energy house. *Energy Build.* **2017**, *140*, 81–97, doi:10.1016/j.enbuild.2017.01.083.

[129] Feist, W*. Passive House Planning Package 2007*; Passivhaus Institut: Darmstadt, Germany, **2007**.

[130] Ahmad, W.; Ayub, N.; Ali, T.; Irfan, M.; Awais, M.; Shiraz, M.; Glowacz, A. Towards Short Term Electricity Load Forecasting Using Improved Support Vector Machine and Extreme Learning Machine. *Energies* **2020**, *13*, 2907, doi:10.3390/en13112907.

[131] H. Hamilton, Machine Learning/Inductive Inference/Decision Trees/Decision Tree Rules & Pruning. Www2.cs.uregina.ca, 2020. Available online:http://www2.cs.uregina.ca/~dbd/cs831/notes/ml/dtrees/4_dtrees3.html (accessed on 17 November 2020).

[132] Guillaume, S. Designing fuzzy inference systems from data: An interpretability-oriented review. *IEEE Trans. Fuzzy Syst.* **2001**, *9*, 426–443, doi:10.1109/91.928739.

[133] What is Fuzzy Logic? Advantage and Disadvantage. Hackr.io, **2021**. Available online: https://hackr.io/blog/what-is-fuzzy-logic (accessed on 16 January 2021).

[134] Magoutas, B.; Apostolou, D.; Mentzas, G. Situation-aware Demand Response in the smart grid. In Proceedings of the 2011 16th International Conference on Intelligent System Applications to Power Systems, Hersonissos, Greece, 25–28 September **2011**; pp. 1–6. doi:10.1109/isap.2011.6082176.

[135] Rantou, K. *Missing Data in Time Series and Imputation Methods*, MSc, University of the Aegean: Lesbos, Greece, **2017**.

[136] Waser, M. *Nonliniear Dependencies in and between Time Serie*, MSc, Vienna University of Technology: Vienna, Austria, **2010**.

[137] Deep Learning for Time Series and why DEEP LEARNING? Medium. **2020**. Available online: https://towardsdatascience.com/deep-learning-for-time-series-and-why-deep-learning-a6120b147d60 (accessed on 26 March 2020).

[138] Hua, Y.; Zhao, Z.; Li, R.; Chen, X.; Liu, Z.; Zhang, H. Deep Learning with Long Short-Term Memory for Time Series Prediction. *IEEE Commun. Mag.* **2019**, *57*, 114–119, doi:10.1109/mcom.2019.1800155.

[139] Koprinska, I.; Wu, D.; Wang, Z. Convolutional Neural Networks for Energy Time Series Forecasting. In Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–8.

[140] Shiblee, M.; Kalra, P.K.; Chandra, B. Time Series Prediction with Multilayer Perceptron (MLP): A New Generalized Error Based Approach. In *Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2009; Volume 5507, pp. 37–44.

[141] Alamaniotis, M.; Tsoukalas, L.H. Anticipation of minutes-ahead household active power consumption using Gaussian processes. In Proceedings of the 2015 6th International Conference on Information, Intelligence, Systems and Applications (IISA), Corfu, Greece, 6–8 July 2015; pp. 1–6.

[142] Singh, S.; Hussain, S.; Bazaz, M.A. Short term load forecasting using artificial neural network. In Proceedings of the 2017 Fourth International Conference on Image Information Processing (ICIIP), Waknaghat, India, 21–23 December2017; pp. 1–5.

[143] Kuo, P.-H.; Huang, C.-J. A High Precision Artificial Neural Networks Model for Short-Term Energy Load Forecasting. *Energies* **2018**, *11*, 213, doi:10.3390/en11010213.

[144] Hossen, T.; Nair, A.S.; Chinnathambi, R.A.; Ranganathan, P. Residential Load Forecasting Using Deep Neural Networks (DNN). In Proceedings of the 2018 North American Power Symposium (NAPS), Fargo, ND, USA, 9–11 September 2018; pp. 1–5, doi:10.1109/naps.2018.8600549.

[145] Zhang, D.; Han, X.; Deng, C.; Taiyuan University of Technology; China Electric Power Research Institute Review on the research and practice of deep learning and reinforcement learning in smart grids. *CSEE J. Power Energy Syst.* **2018**, *4*, 362–370, doi:10.17775/cseejpes.2018.00520.

[146] Kampelis, N.; Tsekeri, E.; Kolokotsa, D.; Kalaitzakis, K.; Isidori, D.; Cristalli, C. Development of Demand Response Energy Management Optimization at Building and District Levels Using Genetic Algorithm and Artificial Neural Network Modelling Power Predictions. *Energies* **2018**, *11*, 3012, doi:10.3390/en11113012.

[147] Koponen, P.; Hänninen, S.; Mutanen, A.; Koskela, J.; Rautiainen, A.; Järventausta, P.; Niska, H.; Kolehmainen, M.; Koivisto, H. Improved modelling of electric loads for enabling demand response by applying physical and data-driven models: Project Response. In Proceedings of the 2018 IEEE International Energy Conference (ENERGYCON), Limassol, Cyprus, 3–7 June 2018; pp. 1–6.

[148] Ahmad, A.; Javaid, N.; Mateen, A.; Awais, M.; Khan, Z.A. Short-Term Load Forecasting in Smart Grids: An Intelligent Modular Approach. *Energies* **2019**, *12*, 164, doi:10.3390/en12010164.

[149] Walther, J.; Spanier, D.; Panten, N.; Abele, E. Very short-term load forecasting on factory level–A machine learning approach. *Procedia CIRP* **2019**, *80*, 705–710, doi:10.1016/j.procir.2019.01.060.

[150] Zhu, J.; Yang, Z.; Mourshed, M.; Li, K.; Zhou, Y.; Chang, Y.; Wei, Y.; Feng, S. Electric Vehicle Charging Load Forecasting: A Comparative Study of Deep Learning Approaches. *Energies* **2019**, *12*, 2692, doi:10.3390/en12142692.

[151] Deep Learning for Time Series Forecasting: The Electric Load Case, GroundAI. 2020. Available online: https://www.groundai.com/project/deep-learning-for-time-series-forecasting-the-electric-load-case/1 (accessed on 26 March 2020).

[152] Time Series Analysis, Visualization & Forecasting with LSTM, Medium. 2020. Available online: https://towardsdatascience.com/time-series-analysis-visualization-forecasting-with-lstm-77a905180eba (accessed on 14 February 2020).

[153] Neural Networks Over Classical Models in Time Series, Medium. 2020. Available online: https://towardsdatascience.com/neural-networks-over-classical-models-in-time-series-5110a714e535 (accessed on 14 February 2020).

[154] LSTM for Time Series Prediction, Medium. 2020. Available online: https://towardsdatascience.com/lstm-for-time-series-prediction-de8aeb26f2ca (accessed on 14 February 2020).

[155] UCI Machine Learning Repository: Individual Household Electric Power Consumption Data Set, Archive.ics.uci.edu. 2020. Available online: https://archive.ics.uci.edu/ml/datasets/individual+household+electric+power+consumption (accessed on 14 February 2020).

[156] Perceptron Learning Algorithm: A Graphical Explanation Of Why It Works, Medium. 2020. Available online: https://towardsdatascience.com/perceptron-learning-algorithm-d5db0deab975 (accessed on 26 March 2020).

[157] Multilayer Perceptron—DeepLearning 0.1 Documentation, Deeplearning.net. 2020. Available online: http://deeplearning.net/tutorial/mlp.html (accessed on 26 March 2020).

[158] CS231n Convolutional Neural Networks for Visual Recognition, Cs231n.github.io. 2020. Available online: http://cs231n.github.io/convolutional-networks/ (accessed on 26 March 2020).

[159] A. LeNail, "NN-SVG: Publication-Ready Neural Network Architecture Schematics", Journal of Open Source Software, vol. 4, no. 33, p. 747, 2019. Available: 10.21105/joss.00747 (accessed on 14 April 2020).

[160] Recurrent Neural Network—An Overview | ScienceDirect Topics, Sciencedirect.com. 2020. Available online: https://www.sciencedirect.com/topics/engineering/recurrent-neural-network (accessed on 26 March 2020).

[161] Understanding LSTM Networks—Colah's Blog, Colah.github.io. 2020. Available online: https://colah.github.io/posts/2015-08-Understanding-LSTMs/ (accessed on 26 March 2020).

[162] How to Select the Right Evaluation Metric for Machine Learning Models: Part 1 Regression Metrics, Medium. 2020. Available online: https://medium.com/@george.drakos62/how-to-select-the-right-evaluation-metric-for-machine-learning-models-part-1-regrression-metrics-3606e25beae0 (accessed on 26 March 2020).

[163] Dimkonto/Minutely-Power-Forecasting, GitHub. 2020. Available online: https://github.com/dimkonto/Minutely-Power-Forecasting (accessed on 15 February 2020).

[164] Ebeid, E.; Heick, R.; Jacobsen, R. Deducing Energy Consumer Behavior from Smart Meter Data. *Futur. Internet* **2017**, *9*, 29, doi:10.3390/fi9030029.

[165] Brownlee, J. How to Develop Multi-Step LSTM Time Series Forecasting Models for Power Usage, Machine Learning Mastery. 2020. Available online: https://machinelearningmastery.com/how-to-develop-lstm-models-for-multi-step-time-series-forecasting-of-household-power-consumption/ (accessed on 16 February 2020).

[166] Kingma, D.; Ba, J. Adam: A Method for Stochastic Optimization, In Proceedings of the 3rd International Conference for Learning Representations, San Diego, CA, USA, 7–9 May 2015. Available online: https://arxiv.org/abs/1412.6980v8 (accessed on 9 April 2020).

[167] An Intro to Hyper-parameter Optimization Using Grid Search and Random Search, Medium. 2020. Available online: https://medium.com/@cjl2fv/an-intro-to-hyper-parameter-optimization-using-grid-search-and-random-search-d73b9834ca0a (accessed on 24 February 2020).

[168] Impram, S.; Varbak Nese, S.; Oral, B. Challenges of renewable energy penetration on power system flexibility: A survey. *Energy Strategy Rev.* **2020**, *31*, 100539. https://doi.org/10.1016/j.esr.2020.100539.

[169] Tsampasis, E.; Bargiotas, D.; Elias, C.; Sarakis, L. Communication challenges in Smart Grid. *MATEC Web Conf.* **2016**, *41*, 01004. https://doi.org/10.1051/matecconf/20164101004.

[170] Barai, G.; Krishnan, S.; Venkatesh, B. Smart metering and functionalities of smart meters in smart grid—A review. In Proceedings of the 2015 IEEE Electrical Power and Energy Conference (EPEC), London, ON, Canada, 26–28 October 2015. https://doi.org/10.1109/EPEC.2015.7379940.

[171] Muñoz, A.; Sánchez-Úbeda, E.; Cruz, A.; Marín, J. Short-term Forecasting in Power Systems: A Guided Tour. In *Handbook of Power Systems II*; Springer: Berlin, Germany, 2010; pp. 129–160. https://doi.org/10.1007/978-3-642-12686-4_5.

[172] Manojpraphakar, T.; A, S. Energy Demand Prediction Using Linear Regression. In *Proceedings of International Conference on Artificial Intelligence, Smart Grid and Smart City Applications*; Springer: Cham, Switzerland, 2020; pp. 407–417. https://doi.org/10.1007/978-3-030-24051-6_40.

[173] Cetinkaya, M.; Acarman, T. Next-Day Electricity Demand Forecasting Using Regression. In Proceedings of the 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), JCT College of Engineering and Technology, India, 25–27 March 2021. https://doi.org/10.1109/ICAIS50930.2021.9395926.

[174] Tang, N.; Mao, S.; Wang, Y.; Nelms, R. Solar Power Generation Forecasting With a LASSO-Based Approach. *IEEE Internet Things J.* **2018**, *5*, 1090–1099. https://doi.org/10.1109/JIOT.2018.2812155.

[175] Sivanantham, G.; Gopalakrishnan, S. Stochastic Gradient Descent Optimization Model for Demand Response in a Connected Microgrid. *KSII Trans. Internet Inf. Syst.* **2022**, *16*, 97–115. https://doi.org/10.3837/tiis.2022.01.006.

[176] Levenbach, H. Time series forecasting using robust regression. *J. Forecast.* **1982**, *1*, 241–255. https://doi.org/10.1002/for.3980010304.

[177] Dang, X.; Peng, H.; Wang, X.; Zhang, H. Theil-Sen Estimators in a Multiple Linear Regression Model. Olemiss Edu 2008. Available online: http://home.olemiss.edu/~xdang/papers/MTSE.pdf (accessed on 1 July 2022).

[178] Wang, Y.; Sun, S.; Chen, X.; Zeng, X.; Kong, Y.; Chen, J.; Guo, Y.; Wang, T. Short-term load forecasting of industrial customers based on SVMD and XGBoost. *Int. J. Electr. Power Amp; Energy Syst.* **2021**, *129*, 106830.

[179] Gelper, S.; Croux, C. Least Angle Regression for Time Series Forecasting with Many Predictors. 2008. Available online: https://www.researchgate.net/publication/255575037_Least_angle_regression_for_time_series_forecasting_with_many_predicpred (accessed on 1 July 2022).

[180] Wahid, F.; Kim, D. A Prediction Approach for Demand Analysis of Energy Consumption Using K-Nearest Neighbor in Residential Buildings. *Int. J. Smart Home* **2016**, *10*, 97–108. https://doi.org/10.14257/ijsh.2016.10.2.10.

[181] Guo, Q.; Feng, Y.; Sun, X.; Zhang, L. Power Demand Forecasting and Application based on SVR. *Procedia Comput. Sci.* **2017**, *122*, 269–275. https://doi.org/10.1016/j.procs.2017.11.369.

[182] Leite Coelho da Silva, F.; da Costa, K.; Canas Rodrigues, P.; Salas, R.; López-Gonzales, J.L. Statistical and Artificial Neural Networks Models for Electricity Consumption Forecasting in the Brazilian Industrial Sector. *Energies* **2022**, *15*, 588. https://doi.org/10.3390/en15020588.

[183] Arvanitidis, A.I.; Bargiotas, D.; Daskalopulu, A.; Kontogiannis, D.; Panapakidis, I.P.; Tsoukalas, L.H. Clustering Informed MLP Models for Fast and Accurate Short-Term Load Forecasting. *Energies* **2022**, *15*, 1295. https://doi.org/10.3390/en15041295.

[184] Arvanitidis, A.I.; Bargiotas, D.; Daskalopulu, A.; Laitsos, V.M.; Tsoukalas, L.H. Enhanced Short-Term Load Forecasting Using Artificial Neural Networks. *Energies* **2021**, *14*, 7788. https://doi.org/10.3390/en14227788.

[185] Kontogiannis, D.; Bargiotas, D.; Daskalopulu, A. Minutely Active Power Forecasting Models Using Neural Networks. *Sustainability* **2020**, *12*, 3177. https://doi.org/10.3390/su12083177.

[186] Wang, L.; Mao, S.; Wilamowski, B.; Nelms, R. Ensemble Learning for Load Forecasting. *IEEE Trans. Green Commun. Netw.* **2020**, *4*, 616–628. https://doi.org/10.1109/TGCN.2020.2987304.

[187] Cho, J.; Yoon, Y.; Son, Y.; Kim, H.; Ryu, H.; Jang, G. A Study on Load Forecasting of Distribution Line Based on Ensemble Learning for Mid- to Long-Term Distribution Planning. *Energies* **2022**, *15*, 2987. https://doi.org/10.3390/en15092987.

[188] Motlagh, O.; Berry, A.; O'Neil, L. Clustering of residential electricity customers using load time series. *Appl. Energy* **2019**, *237*, 11–24. https://doi.org/10.1016/j.apenergy.2018.12.063.

[189] Bellahsen, A.; Dagdougui, H. Aggregated short-term load forecasting for heterogeneous buildings using machine learning with peak estimation. *Energy Build.* **2021**, *237*, 110742. https://doi.org/10.1016/j.enbuild.2021.110742.

[190] Ackerman, S.; Farchi, E.; Raz, O.; Zalmanovici, M.; Dube, P. Detection of data drift and outliers affecting machine learning model performance over time**.** *arXiv preprint* **2021**, arXiv:2012.09258. https://doi.org/10.48550/arXiv.2012.09258.

[191] Lu, J.; Liu, A.; Dong, F.; Gu, F.; Gama, J.; Zhang, G. Learning under Concept Drift: A Review. *IEEE Trans. Knowl. Data Eng.* **2018**, *31*, 2346–2363. https://doi.org/10.1109/TKDE.2018.2876857.

[192] Ceperic, E.; Ceperic, V.; Baric, A. A Strategy for Short-Term Load Forecasting by Support Vector Regression Machines. *IEEE Trans. Power Syst.* **2013**, *28*, 4356–4364. https://doi.org/10.1109/TPWRS.2013.2269803.

[193] Wijaya, T.; Vasirani, M.; Humeau, S.; Aberer, K. Cluster-based aggregate forecasting for residential electricity demand using smart meter data. In Proceedings of the 2015 IEEE International Conference on Big Data (Big Data), Santa Clara, CA, USA, 29 October–1 November 2015. https://doi.org/10.1109/BigData.2015.7363836.

[194] Karthika, S.; Margaret, V.; Balaraman, K. Hybrid short term load forecasting using ARIMA-SVM. In Proceedings of the 2017 Innovations in Power and Advanced Computing Technologies (i-PACT), Vellore, India, 21–22 April 2017. https://doi.org/10.1109/IPACT.2017.8245060.

[195] Laurinec, P.; Lucká, M. Usefulness of Unsupervised Ensemble Learning Methods for Time Series Forecasting of Aggregated or Clustered Load. In *New Frontiers in Mining Complex Patterns*; Springer: Cham, Switzerland, 2018; pp. 122–137. https://doi.org/10.1007/978-3-319-78680-3_9.

[196] Fu, X.; Zeng, X.; Feng, P.; Cai, X. Clustering-based short-term load forecasting for residential electricity under the increasing-block pricing tariffs in China. *Energy* **2018**, *165*, 76–89. https://doi.org/10.1016/j.energy.2018.09.156.

[197] Li, Y.; Che, J.; Yang, Y. Subsampled support vector regression ensemble for short term electric load forecasting. *Energy* **2018**, *164*, 160–170. https://doi.org/10.1016/j.energy.2018.08.169.

[198] Bian, H.; Zhong, Y.; Sun, J.; Shi, F. Study on power consumption load forecast based on K-means clustering and FCM–BP model. *Energy Rep.* **2020**, *6*, 693–700. https://doi.org/10.1016/j.egyr.2020.11.148.

[199] Sarajcev, P.; Jakus, D.; Vasilj, J. Ensemble learning with time-series clustering for aggregated short-term load forecasting. In Proceedings of the 2020 IEEE 20th

Mediterranean Electrotechnical Conference (MELECON), Palermo, Italy, 16–18 June 2020. https://doi.org/10.1109/MELECON48756.2020.9140676.

[200] Cini, A.; Lukovic, S.; Alippi, C. Cluster-based Aggregate Load Forecasting with Deep Neural Networks. In Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 19–24 July 2020. https://doi.org/10.1109/IJCNN48605.2020.9207503.

[201] Kontogiannis, D.; Bargiotas, D.; Daskalopulu, A.; Tsoukalas, L.H. A Meta-Modeling Power Consumption Forecasting Approach Combining Client Similarity and Causality. *Energies* **2021**, *14*, 6088. https://doi.org/10.3390/en14196088.

[202] Stratigakos, A.; Bachoumis, A.; Vita, V.; Zafiropoulos, E. Short-Term Net Load Forecasting with Singular Spectrum Analysis and LSTM Neural Networks. *Energies* **2021**, *14*, 4107. https://doi.org/10.3390/en14144107.

[203] Zafeiropoulou, M.; Mentis, I.; Sijakovic, N.; Terzic, A.; Fotis, G.; Maris, T.I.; Vita, V.; Zoulias, E.; Ristic, V.; Ekonomou, L. Forecasting Transmission and Distribution System Flexibility Needs for Severe Weather Condition Resilience and Outage Management. *Appl. Sci.* **2022**, *12*, 7334. https://doi.org/10.3390/app12147334.

[204] Phyo, P.-P.; Byun, Y.-C.; Park, N. Short-Term Energy Forecasting Using Machine-Learning-Based Ensemble Voting Regression. *Symmetry* **2022**, *14*, 160. https://doi.org/10.3390/sym14010160.

[205] Ensemble Learning Techniques Available online: https://towardsdatascience.com/ensemble-learning-techniques-6346db0c6ef8 (accessed on 1 July 2022).

[206] Wolpert, D. Stacked generalization. *Neural Netw.* **1992**, *5*, 241–259. https://doi.org/10.1016/S0893-6080(05)80023-1.

[207] Tugay, R.; Gündüz Öğüdücü, Ş. Demand Prediction using Machine Learning Methods and Stacked Generalization. Proceedings of the 6th International Conference on Data Science, Technology and Applications, Madrid Spain, 24–26 July 2017. https://doi.org/10.5220/0006431602160222.

[208] 1.11. Ensemble Methods. Available online: https://scikit-learn.org/stable/modules/ensemble.html#voting-regressor (accessed on 1 July 2022).

[209] An, K.; Meng, J. Voting-Averaged Combination Method for Regressor Ensemble. In *Lecture Notes in Computer Science*; Springer: Berlin, Germany, 2010; pp. 540–546. https://doi.org/10.1007/978-3-642-14922-1_67.

[210] Humaira, H.; Rasyidah, R. Determining The Appropiate Cluster Number Using Elbow Method for K-Means Algorithm. In Proceedings of the 2nd Workshop on Multidisciplinary and Applications (WMA) 2018, Padang, Indonesia, 24–25 January 2018. https://doi.org/10.4108/eai.24-1-2018.2292388.

[211] Shahapure, K.; Nicholas, C. Cluster Quality Analysis Using Silhouette Score. In Proceedings of the 2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA), Sydney, Australia, 6–9 October 2020. https://doi.org/10.1109/DSAA49011.2020.00096.

[212] Yuan, C.; Yang, H. Research on K-Value Selection Method of K-Means Clustering Algorithm. *J* **2019**, *2*, 226–235. https://doi.org/10.3390/j2020016.

[213] Shutaywi, M.; Kachouie, N.N. Silhouette Analysis for Performance Evaluation in Machine Learning with Applications to Clustering. *Entropy* **2021**, *23*, 759. https://doi.org/10.3390/e23060759.

[214] Time Series Clustering—Tslearn 0.5.2 Documentation. Available online: https://tslearn.readthedocs.io/en/stable/user_guide/clustering.html (accessed on 1 July 2022).

[215] Senin, P. Dynamic Time Warping Algorithm Review. 2008. Available online: https://www.researchgate.net/publication/228785661_Dynamic_Time_Warping_Algorithm_Review (accessed on 1 July 2022).

[216] scipy.signal.find_peaks—SciPy v1.8.1 Manual. Available on: https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.find_peaks.html (accessed on 1 July 2022).

[217] UCI Machine Learning Repository: Electricity Load Diagrams 2011–2014 Data Set. Available online: https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014 (accessed on 1 July 2022).

[218] 3.2. Tuning the Hyper-Parameters of an Estimator. Available online: https://scikit-learn.org/stable/modules/grid_search.html#randomized-parameter-search (accessed on 1 July 2022).

[219] Scikit-Learn: Machine Learning in Python—Scikit-Learn 1.1.1 Documentation. Available online: https://scikit-learn.org/stable/ (accessed on 1 July 2022).

[220] Python Package Introduction—xgboost 1.6.1 documentation. Available online: https://xgboost.readthedocs.io/en/stable/python/python_intro.html (accessed on 1 July 2022).

[221] GitHub—Dimkonto/Structural-Ensemble-Regression-Models-for-CBAF: Structural Ensemble Regression for Cluster-Based Aggregate Electricity Load Forecasting. Available online: https://github.com/dimkonto/Structural-Ensemble-Regression-Models-for-CBAF (accessed on 11 July 2022).

[222] Wang, Z.; Bovik, A. Mean squared error: Love it or leave it? A new look at Signal Fidelity Measures. In *IEEE Signal Processing Magazine;* IEEE: New York, USA, 2009; *26*, pp. 98-117. https://doi.org/10.1109/MSP.2008.930649

[223] Hodson, T. Root mean square error (RMSE) or mean absolute error (MAE): When to use them or not. *Geosci. Model Dev. Discuss.* **2022**, *15*, 5481-5487. https://doi.org/10.5194/gmd-2022-64.

[224] Von Krannichfeldt, L.; Wang, Y.; Hug, G. Online Ensemble Learning for Load Forecasting. *IEEE Trans. Power Syst.* **2021**, *36*, 545–548. https://doi.org/10.1109/TPWRS.2020.3036230.

[225] Sengar, S.; Liu, X. Ensemble approach for short term load forecasting in wind energy system using hybrid algorithm. *J. Ambient. Intell. Humaniz. Comput.* **2020**, *11*, 5297–5314. https://doi.org/10.1007/s12652-020-01866-7.

[226] Alamaniotis, M.; Ikonomopoulos, A.; Bargiotas, D.; Tsoukalas, L.; Alamaniotis, A. Day-ahead Electricity Price Forecasting using Optimized Multiple-Regression of Relevance Vector Machines. In Proceedings of the 8th Mediterranean Conference on Power Generation, Transmission, Distribution and Energy Conversion (MEDPOWER 2012), Cagliari, Italy, 1–3 October 2012, doi:10.1049/cp.2012.2032.

[227] Alamaniotis, M.; Bargiotas, D.; Bourbakis, N.; Tsoukalas, L. Genetic Optimal Regression of Relevance Vector Machines for Electricity Pricing Signal Forecasting in Smart Grids. *IEEE Trans. Smart Grid* **2015**, *6*, 2997–3005, doi:10.1109/tsg.2015.2421900.

[228] Alamaniotis, M.; Bargiotas, D.; Tsoukalas, L.H. Towards smart energy systems: application of kernel machine regression for medium term electricity load forecasting. *SpringerPlus* **2016**, *5*, 1–15, https://doi.org/10.1186/s40064-016-1665-z.

[229] Hwang, J.; Suh, D.; Otto, M.-O. Forecasting Electricity Consumption in Commercial Buildings Using a Machine Learning Approach. *Energies* **2020**, *13*, 5885, doi:10.3390/en13225885.

[230] Mir, A.; Alghassab, M.; Ullah, K.; Khan, Z.; Lu, Y.; Imran, M. A Review of Electricity Demand Forecasting in Low and Middle Income Countries: The Demand Determinants and Horizons. *Sustainability* **2020**, *12*, 5931, doi:10.3390/su12155931.

[231] Kontogiannis, D.; Bargiotas, D.; Daskalopulu, A. Fuzzy Control System for Smart Energy Management in Residential Buildings Based on Environmental Data. *Energies* **2021**, *14*, 752, doi:10.3390/en14030752.

[232] Dalal, M.; Li, A.; Taori, R. Autoregressive Models: What Are They Good For? *arXiv* **2019**, arXiv:1910.07737. Available online: https://arxiv.org/abs/1910.07737 (accessed on 9 July 2021).

[233] Johnston, F.R.; Boyland, E.J.; Meadows, M.; Shale, E. Some properties of a simple moving average when applied to forecasting a time series. *J. Oper. Res. Soc.* **1999**, *50*, 1267–1271, doi:10.1057/palgrave.jors.2600823.

[234] Advanced Time Series Analysis with ARMA and ARIMA, Medium. 2021. Available online: https://towardsdatascience.com/advanced-time-series-analysis-with-arma-and-arima-a7d9b589ed6d (accessed on 9 July 2021).

[235] Luetkepohl, H. Forecasting with VARMA Models. In *Handbook of Economic Forecasting*, 1st ed.; Elliott, G., Granger, C., Timmerman, A., Eds.; Elsevier: Amsterdam, The Netherlands, 2006; Chapter 6, Volume 1, pp. 287–325.

[236] Ostertagová, E.; Ostertag, O. Forecasting using simple exponential smoothing method. *Acta Electrotech. et Inform.* **2012**, *12*, 62–66, doi:10.2478/v10198-012-0034-2.

[237] An Overview of Time Series Forecasting Models Part 1: Classical Time Series Forecasting Models, Medium. 2021. Available online: https://shaileydash.medium.com/an-overview-of-time-series-forecasting-models-part-1-classical-time-series-forecasting-models-2d877de76e0f (accessed on 9 July 2021).

[238] Mitrea, C.A.; Lee, C.K.M.; Wu, Z. A Comparison between Neural Networks and Traditional Forecasting Methods: A Case Study. *Int. J. Eng. Bus. Manag.* **2009**, *1*, 11, doi:10.5772/6777.

[239] Khodabakhsh, A.; Ari, I.; Bakır, M.; Alagoz, S.M. Forecasting Multivariate Time-Series Data Using LSTM and Mini-Batches. In Proceedings of the 7th International Conference on Contemporary Issues in Data Science, Zanjan, Iran, 6–8 March 2019; Spinger: Cham, Switzerland, 2019; pp. 121–129, doi:10.1007/978-3-030-37309-2_10.

[240] López de Prado, M. Overfitting: Causes and Solutions (Seminar Slides). 2020. Available online: http://dx.doi.org/10.2139/ssrn.3544431 (accessed on 9 July 2021).

[241] Jayalakshmi, N.; Shankar, R.; Subramaniam, U.; Baranilingesan, I.; Karthick, A.; Stalin, B.; Rahim, R.; Ghosh, A. Novel Multi-Time Scale Deep Learning Algorithm for Solar Irradiance Forecasting. *Energies* **2021**, *14*, 2404, doi:10.3390/en14092404.

[242] Dai, W.; Yoshigoe, K.; Parsley, W. Improving Data Quality Through Deep Learning and Statistical Models. *Adv. Intell. Syst. Comput.* **2017**, 515–522, doi:10.1007/978-3-319-54978-1_66.

[243] Choi, J.Y.; Lee, B. Combining LSTM Network Ensemble via Adaptive Weighting for Improved Time Series Forecasting. *Math. Probl. Eng.* **2018**, *2018*, 1–8, doi:10.1155/2018/2470171.

[244] Tian, C.; Ma, J.; Zhang, C.; Zhan, P. A Deep Neural Network Model for Short-Term Load Forecast Based on Long Short-Term Memory Network and Convolutional Neural Network. *Energies* **2018**, *11*, 3493, doi:10.3390/en11123493.

[245] Mujeeb, S.; Javaid, N.; Ilahi, M.; Wadud, Z.; Ishmanov, F.; Afzal, M.K. Deep Long Short-Term Memory: A New Price and Load Forecasting Scheme for Big Data in Smart Cities. *Sustainability* **2019**, *11*, 987, doi:10.3390/su11040987.

[246] Markovič, R.; Gosak, M.; Grubelnik, V.; Marhl, M.; Virtič, P. Data-driven classification of residential energy consumption patterns by means of functional connectivity networks. *Appl. Energy* **2019**, *242*, 506–515, doi:10.1016/j.apenergy.2019.03.

[247] Jin, X.-B.; Zheng, W.-Z.; Kong, J.-L.; Wang, X.-Y.; Bai, Y.-T.; Su, T.-L.; Lin, S. Deep-Learning Forecasting Method for Electric Power Load via Attention-Based Encoder-Decoder with Bayesian Optimization. *Energies* **2021**, *14*, 1596, doi:10.3390/en14061596.

[248] Tian, Y.; Sehovac, L.; Grolinger, K. Similarity-Based Chained Transfer Learning for Energy Forecasting with Big Data. *IEEE Access* **2019**, *7*, 139895–139908, doi:10.1109/access.2019.2943752.

[249] Chen, T.-L.; Cheng, C.-H.; Liu, J.-W. A Causal Time-Series Model Based on Multilayer Perceptron Regression for Forecasting Taiwan Stock Index. *Int. J. Inf. Technol. Decis. Mak.* **2019**, *18*, 1967–1987, doi:10.1142/s0219622019500421.

[250] Boersma, K. Using Influencing Factors and Multilayer Perceptrons for Energy Demand Prediction. 2019. Available online: http://essay.utwente.nl/78789/ (accessed on 10 July 2021).

[251] Emamian, M.; Milimonfared, J.; Aghaei, M.; Hosseini, R. Solar Power Forecasting with Lstm Network Ensemble, Researchgate. 2019. Available online: https://www.researchgate.net/publication/337494650 (accessed on 10 July 2021).

[252] Guo, L.; Wang, L.; Chen, H. Electrical Load Forecasting Based on LSTM Neural Networks. *BDECE* **2019**, 107–111, doi:10.2991/acsr.k.191223.024.

[253] Tao, C.; Lu, J.; Lang, J.; Peng, X.; Cheng, K.; Duan, S. Short-Term Forecasting of Photovoltaic Power Generation Based on Feature Selection and Bias Compensation–LSTM Network. *Energies* **2021**, *14*, 3086, doi:10.3390/en14113086.

[254] An Introduction to Gradient Descent and Backpropagation, Medium. 2021. Available online: https://towardsdatascience.com/an-introduction-to-gradient-descent-and-backpropagation-81648bdb19b2 (accessed on 10 July 2021).

[255] Kamal, I.M.; Bae, H.; Sunghyun, S.; Yun, H. DERN: Deep Ensemble Learning Model for Short- and Long-Term Prediction of Baltic Dry Index. *Appl. Sci.* **2020**, *10*, 1504, doi:10.3390/app10041504.

[256] Ramchoun, H.; Idrissi, M.A.J.; Ghanou, Y.; Ettaouil, M. Multilayer Perceptron. In Proceedings of the 2nd international Conference on Big Data, Cloud and Applications, New York, NY, USA, 29–30 March 2017, doi:10.1145/3090354.3090427.

[257] Rohrhofer, F.; Saha, S.; Cataldo, S.; Geiger, B.; Linden, W.; Boeri, L. Importance of feature engineering and database selection in a machine learning model: A case study on carbon crystal structures. *arxiv* **2021**, Available online: https://arxiv.org/abs/2102.00191 (accessed on 10 July 2021).

[258] Cassisi, C.; Montalto, P.; Aliotta, M.; Cannata, A.; Pulvirenti, A.C.A.A. *Adv. Data Min. Knowl. Discov. Appl.* **2012**, doi:10.5772/49941.

[259] Cuturi, M.; Blondel, M. Soft-DTW: A Differentiable Loss Function for Time-Series. *arxiv* **2017**. Available online: https://arxiv.org/abs/1703.01541 (accessed on 10 July 2021).

[260] Time Series Similarity Using Dynamic Time Warping -Explained, Medium. 2021. Available online: https://medium.com/walmartglobaltech/time-series-similarity-using-dynamic-time-warping-explained-9d09119e48ec (accessed on 10 July 2021).

[261] Inferring Causality in Time Series Data, Medium. 2021. Available online: https://towardsdatascience.com/inferring-causality-in-time-series-data-b8b75fe52c46 (accessed on 10 July 2021).

[262] Amornbunchornvej, C.; Zheleva, E.; Berger-Wolf, T. Variable-Lag Granger Causality for Time Series Analysis. In Proceedings of the 2019 IEEE International Conference on

Data Science and Advanced Analytics (DSAA), Washington, DC, USA, 5–8 October 2019, doi:10.1109/dsaa.2019.00016.

[263] Brownlee, J. Time Series Forecasting as Supervised Learning, Machine Learning Mastery. 2021. Available online: https://machinelearningmastery.com/time-series-forecasting-supervised-learning/ (accessed on 10 July 2021).

[264] Introduction to Early Stopping: An Effective Tool to Regularize Neural Nets, Medium. 2021. Available online: https://towardsdatascience.com/early-stopping-a-cool-strategy-to-regularize-neural-networks-bfdeca6d722e (accessed on 10 July 2021).

[265] Liashchynskyi, P.; Liashchynskyi, P. Grid Search, Random Search, Genetic Algorithm: A Big Comparison for NAS. *arxiv* **2019**. Available online: https://arxiv.org/abs/1912.06059 (accessed on 10 July 2021).

[266] Brownlee, J. Stacking Ensemble Machine Learning with Python, Machine Learning Mastery. *arxiv* **2021**. Available online: https://machinelearningmastery.com/stacking-ensemble-machine-learning-with-python/ (accessed on 10 July 2021).

[267] Parraga-Alava, J.; Moncayo-Nacaza, J.D.; Revelo-Fuelagán, J.; Rosero-Montalvo, P.D.; Anaya-Isaza, A.; Peluffo-Ordóñez, D.H. A data set for electric power consumption forecasting based on socio-demographic features: Data from an area of southern Colombia. *Data Brief* **2020**, *29*, 105246, doi:10.1016/j.dib.2020.105246.

[268] Shapiro, S.S.; Wilk, M.B. An Analysis of Variance Test for Normality (Complete Samples). *Biometrika* **1965**, *52*, 591, doi:10.2307/2333709.

[269] dimkonto/Client-Power-Consumption-Forecasting, GitHub. 2021. Available online: https://github.com/dimkonto/Client-Power-Consumption-Forecasting (accessed on 13 July 2021).

[270] Dudek, G. Pattern similarity-based methods for short-term load forecasting – Part 1: Principles. *Appl. Soft Comput.* **2015**, *37*, 277–287, doi:10.1016/j.asoc.2015.08.040.

[271] Cordova, J.; Sriram, L.M.K.; Kocatepe, A.; Zhou, Y.; Ozguven, E.E.; Arghandeh, R. Combined Electricity and Traffic Short-Term Load Forecasting Using Bundled

Causality Engine. *IEEE Trans. Intell. Transp. Syst.* **2018**, *20*, 3448–3458, doi:10.1109/tits.2018.2876871.

[272] What Is Price Volatility, Eia.gov. 2021. Available online: https://www.eia.gov/naturalgas/weekly/archivenew_ngwu/2003/10_23/Volatility%2010-22-03.htm (accessed on 11 January 2022).

[273] Fanelli, V.; Schmeck, M. On The Seasonality In The Implied Volatility Of Electricity Options. *Quant. Financ.* **2019**, *19*, 1321–1337. https://doi.org/10.1080/14697688.2019.1582792.

[274] NRG Editorial Voices. The Relationship between Weather and Energy Prices, NRG Energy. 2021. Available online: https://www.nrg.com/insights/innovation/the-relationship-between-weather-and-energy-prices.html (accessed on 11 January 2022).

[275] Pereira da Silva, P.; Horta, P. The Effect Of Variable Renewable Energy Sources On Electricity Price Volatility: The Case Of The Iberian Market. *Int. J. Sustain. Energy* **2019**, *38*, 794–813. https://doi.org/10.1080/14786451.2019.1602126.

[276] Prices and Affordability—World Energy Outlook 2021—Analysis—IEA, IEA. 2021. Available online: https://www.iea.org/reports/world-energy-outlook-2021/prices-and-affordability (accessed on 11 January 2022).

[277] Van Cleef, H. Electricity Monitor—Power Prices Are Skyrocketing! ABN AMRO Group Economics. 2021. Available online: https://www.abnamro.com/research/en/our-research/electricity-monitor-power-prices-are-skyrocketing1 (accessed on 11 January 2022).

[278] Bjarghov, S.; Loschenbrand, M.; Ibn Saif, A.; Alonso Pedrero, R.; Pfeiffer, C.; Khadem, S.; Rabelhofer, M.; Revheim, F.; Farahmand, H. Developments And Challenges In Local Electricity Markets: A Comprehensive Review. *IEEE Access* **2021**, *9*, 58910–58943. https://doi.org/10.1109/access.2021.3071830.

[279] Day-Ahead and Real-Time Energy Markets, Iso-ne.com. 2021. Available online: https://www.iso-ne.com/markets-operations/markets/da-rt-energy-markets/ (accessed on 11 January 2022).

[280] Vivas, E.; Allende-Cid, H.; Salas, R. A Systematic Review of Statistical and Machine Learning Methods for Electrical Power Forecasting with Reported MAPE Score. *Entropy* **2020**, *22*, 1412. https://doi.org/10.3390/e22121412.

[281] Cincotti, S.; Gallo, G.; Ponta, L.; Raberto, M. Modeling and forecasting of electricity spot-prices: Computational intelligence vs classical econometrics. *AI Commun.* **2014**, *27*, 301–314. https://doi.org/10.3233/AIC-140599.

[282] Ferreira, Â.; Ramos, J.; Fernandes, P. A Linear Regression Pattern For Electricity Price Forecasting In The Iberian Electricity Market. *Rev. Fac. De Ing. Univ. De Antioq.* **2019**, 93,117–127. https://doi.org/10.17533/udea.redin.20190522.

[283] Zhang, J.; Han, J.; Wang, R.; Hou, G. Day-Ahead Electricity Price Forecasting Based On Rolling Time Series And Least Square-Support Vector Machine Model. In Proceedings of the 2011 Chinese Control and Decision Conference (CCDC), Mianyang, China, 23–25 May 2011. https://doi.org/10.1109/ccdc.2011.5968342.

[284] Bikcora, C.; Verheijen, L.; Weiland, S. Density forecasting of daily electricity demand with ARMA-GARCH, CAViaR, and CARE econometric models. *Sustain. Energy Grids Netw.* **2018**, *13*, 148–156. https://doi.org/10.1016/j.segan.2018.01.001.

[285] Marín, J.; Villada, F. Regionalized discount rate to evaluate renewable energy projects in Colombia. *Int. J. Energy Econ. Policy* **2020**, *10*, 332–336. https://doi.org/10.32479/ijeep.8924.

[286] Cerjan, M.; Matijaš, M.; Delimar, M. Dynamic Hybrid Model for Short-Term Electricity Price Forecasting. *Energies* **2014**, *7*, 3304–3318. https://doi.org/10.3390/en7053304.

[287] Marcjasz, G.; Lago, J.; Weron, R. Neural Networks In Day-Ahead Electricity Price Forecasting: Single Vs. Multiple Outputs. *arXiv* **2020**. arXiv:2008.08006. Available online: https://arxiv.org/abs/2008.08006 (accessed on 11 January 2022).

[288] Jiang, L.; Hu, G. Day-Ahead Price Forecasting For Electricity Market Using Long-Short Term Memory Recurrent Neural Network. In Proceedings of the 2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV), Singapore, 18–21 November 2018. https://doi.org/10.1109/icarcv.2018.8581235.

[289] Khan, Z.; Fareed, S.; Anwar, M.; Naeem, A.; Gul, H.; Arif, A.; Javaid, N. Short Term Electricity Price Forecasting Through Convolutional Neural Network (CNN). In *Advances in Intelligent Systems and Computing*; Springer, Switzerland, 2020; pp. 1181–1188. https://doi.org/10.1007/978-3-030-44038-1_108.

[290] Honkela, A. Multilayer Perceptrons. 2001. Available online: https://users.ics.aalto.fi/ahonkela/dippa/node41.html (accessed on 14 January 2022).

[291] Srivastava, P. Long Short Term Memory|Architecture of LSTM, Analytics Vidhya. 2017. Available online: https://www.analyticsvidhya.com/blog/2017/12/fundamentals-of-deep-learning-introduction-to-lstm/ (accessed on 14 January 2022).

[292] Smeda, K. Understand the Architecture of CNN, Medium. 2021. Available online: https://towardsdatascience.com/understand-the-architecture-of-cnn-90a25e244c7 (accessed on 14 January 2022).

[293] Brownlee, J. A Gentle Introduction to Ensemble Learning Algorithms, Machine Learning Mastery. 2021. Available online: https://machinelearningmastery.com/tour-of-ensemble-learning-algorithms/ (accessed on 14 January 2022).

[294] Yajnik, A. The Emergence of Hybrid Models in Time Series Forecasting, Medium. 2021. Available online: https://medium.com/@ayushyajnik2/the-emergence-of-hybrid-models-in-time-series-forecasting-31e0341bb538 (accessed on 14 January 2022).

[295] de Marcos, R.; Bello, A.; Reneses, J. Electricity price forecasting in the short term hybridising fundamental and econometric modelling. *Electr. Power Syst. Res.* **2019**, *167*, 240–251. https://doi.org/10.1016/j.epsr.2018.10.034.

[296] Bhagat, M.; Alamaniotis, M.; Fevgas, A. Extreme Interval Electricity Price Forecasting of Wholesale Markets Integrating ELM and Fuzzy Inference. In Proceedings of the 2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA), Patras, Greece, 15–17 July 2019. https://doi.org/10.1109/IISA.2019.8900703.

[297] Marcjasz, G. Forecasting Electricity Prices Using Deep Neural Networks: A Robust Hyper-Parameter Selection Scheme. *Energies* **2020**, *13*, 4605. https://doi.org/10.3390/en13184605.

[298] Zhang, W.; Cheema, F.; Srinivasan, D. Forecasting of Electricity Prices Using Deep Learning Networks. In Proceedings of the 2018 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC), Kota Kinabalu, Malaysia, 7–10 October 2018. https://doi.org/10.1109/appeec.2018.8566313.

[299] Alamaniotis, M.; Gatsis, N.; Tsoukalas, L. Virtual Budget: Integration of electricity load and price anticipation for load morphing in price-directed energy utilization. *Electr. Power Syst. Res.* **2018**, *158*, 284–296. https://doi.org/10.1016/j.epsr.2018.01.006.

[300] Angamuthu Chinnathambi, R.; Mukherjee, A.; Campion, M.; Salehfar, H.; Hansen, T.M.; Lin, J.; Ranganathan, P. A Multi-Stage Price Forecasting Model for Day-Ahead Electricity Markets. *Forecasting* **2019**, *1*, 26–46. https://doi.org/10.3390/forecast1010003.

[301] Chang, Z.; Zhang, Y.; Chen, W. Electricity price prediction based on hybrid model of adam optimized LSTM neural network and wavelet transform. *Energy* **2019**, *187*, 115804. https://doi.org/10.1016/j.energy.2019.07.134.

[302] Su, M.; Zhang, Z.; Zhu, Y.; Zha, D. Data-Driven Natural Gas Spot Price Forecasting with Least Squares Regression Boosting Algorithm. *Energies* **2019**, *12*, 1094. https://doi.org/10.3390/en12061094.

[303] Atef, S.; Eltawil, A. A Comparative Study Using Deep Learning and Support Vector Regression for Electricity Price Forecasting in Smart Grids. In Proceedings of the 2019 IEEE 6th International Conference on Industrial Engineering and Applications (ICIEA), Tokyo, Japan, 12–15 April 2019. https://doi.org/10.1109/iea.2019.8715213.

[304] Bissing, D.; Klein, M.; Chinnathambi, R.; Selvaraj, D.; Ranganathan, P. A Hybrid Regression Model for Day-Ahead Energy Price Forecasting. *IEEE Access* **2019**, *7*, 36833–36842. https://doi.org/10.1109/access.2019.2904432.

[305] Xu, J.; Baldick, R. Day-Ahead Price Forecasting in ERCOT Market Using Neural Network Approaches. In Proceedings of the Tenth ACM International Conference on Future

Energy Systems, Phoenix, AZ, USA, 25–28 June 2019. https://doi.org/10.1145/3307772.3331024.

[306] Zhang, C.; Li, R.; Shi, H.; Li, F. Deep learning for day-ahead electricity price forecasting. *IET Smart Grid* **2020**, *3*, 462–469. https://doi.org/10.1049/iet-stg.2019.0258.

[307] Lago, J.; Marcjasz, G.; De Schutter, B.; Weron, R. Forecasting day-ahead electricity prices: A review of state-of-the-art algorithms, best practices and an open-access benchmark. *Appl. Energy* **2021**, *293*, 116983. https://doi.org/10.1016/j.apenergy.2021.116983.

[308] Vega-Márquez, B.; Rubio-Escudero, C.; Nepomuceno-Chamorro, I.A.; Arcos-Vargas, Á. Use of Deep Learning Architectures for Day-Ahead Electricity Price Forecasting over Different Time Periods in the Spanish Electricity Market. *Appl. Sci.* **2021**, *11*, 6097. https://doi.org/10.3390/app11136097.

[309] Jiang, P.; Liu, Z.; Wang, J.; Zhang, L. Decomposition-selection-ensemble forecasting system for energy futures price forecasting based on multi-objective version of chaos game optimization algorithm. *Resour. Policy* **2021**, *73*, 102234. https://doi.org/10.1016/j.resourpol.2021.102234.

[310] Li, T.; Qian, Z.; Deng, W.; Zhang, D.; Lu, H.; Wang, S. Forecasting crude oil prices based on variational mode decomposition and random sparse Bayesian learning. *Appl. Soft Comput.* **2021**, *113*, 108032. https://doi.org/10.1016/j.asoc.2021.108032.

[311] Pourdaryaei, A.; Mohammadi, M.; Karimi, M.; Mokhlis, H.; Illias, H.A.; Kaboli, S.H.A.; Ahmad, S. Recent Development in Electricity Price Forecasting Based on Computational Intelligence Techniques in Deregulated Power Market. *Energies* **2021**, *14*, 6104. https://doi.org/10.3390/en14196104.

[312] Subarna, D. Structure of Neural Network|Artificial Intelligence, Engineering Notes India. 2021. Available online: https://www.engineeringenotes.com/artificial-intelligence-2/neural-network-artificial-intelligence-2/structure-of-neural-network-artificial-intelligence/35410 (accessed on 14 January 2022).

[313] Ronaghan, S. Deep Learning: Overview of Neurons and Activation Functions, Medium. 2018. Available online: https://srnghn.medium.com/deep-learning-

overview-of-neurons-and-activation-functions-1d98286cf1e4 (accessed on 14 January 2022).

[314] Kurama, V. Regression in Machine Learning: What It Is and Examples of Different Models, Built In, 2019. Available online: https://builtin.com/data-science/regression-machine-learning (accessed on 14 January 2022).

[315] Weight (Artificial Neural Network), DeepAI. 2021. Available online: https://deepai.org/machine-learning-glossary-and-terms/weight-artificial-neural-network (accessed on 14 January 2022).

[316] Kostadinov, S. Understanding Backpropagation Algorithm, Medium. 2021. Available online: https://towardsdatascience.com/understanding-backpropagation-algorithm-7bb3aa2f95fd (accessed on 14 January 2022).

[317] Brownlee, J. Loss and Loss Functions for Training Deep Learning Neural Networks, Machine Learning Mastery. 2019. Available online: https://machinelearningmastery.com/loss-and-loss-functions-for-training-deep-learning-neural-networks/ (accessed on 14 January 2022).

[318] Keim, R. Understanding Local Minima in Neural-Network Training, All About Circuits. 2020. Available online: https://www.allaboutcircuits.com/technical-articles/understanding-local-minima-in-neural-network-training/ (accessed on 14 January 2022).

[319] Ippolito, P. Hyperparameters Optimization, Medium. 2019. Available online: https://towardsdatascience.com/hyperparameters-optimization-526348bb8e2d (accessed on 14 January 2022).

[320] Mack, D. How to pick the best learning rate for your machine learning project, freeCodeCamp.org. 2018. Available online: https://www.freecodecamp.org/news/how-to-pick-the-best-learning-rate-for-your-machine-learning-project-9c28865039a8/ (accessed on 14 January 2022).

[321] Goyal, C. Regularization|Regularization Techniques in Machine Learning, Analytics Vidhya. 2021. Available online: https://www.analyticsvidhya.com/blog/2021/05/complete-guide-to-regularization-techniques-in-machine-learning/ (accessed on 14 January 2022).

[322] Vijay, U. Early Stopping to avoid overfitting in neural network- Keras, Medium. 2019. Available online: https://medium.com/zero-equals-false/early-stopping-to-avoid-overfitting-in-neural-network-keras-b68c96ed05d9 (accessed on 14 January 2022).

[323] Radečić, D. Time Series From Scratch — AutoRegression Theory and Implementation, Medium. 2021. Available online: https://towardsdatascience.com/time-series-from-scratch-autoregression-theory-and-implementation-f8c614f738f2 (accessed on 14 January 2022).

[324] Jantana, P.; Sudasna-na-Ayudthya, P. Least Squares and Discounted Least Squares in Autoregressive Process. Available online: https://www.thaiscience.info/journals/Article/SUIJ/10559421.pdf (accessed on 14 January 2022).

[325] Verma, Y. Complete Guide To Dickey-Fuller Test In Time-Series Analysis, Analytics India Magazine. 2021. Available online: https://analyticsindiamag.com/complete-guide-to-dickey-fuller-test-in-time-series-analysis/ (accessed on 14 January 2022).

[326] Bevans, R. An introduction to the Akaike information criterion, Scribbr, 2020. Available online: https://www.scribbr.com/statistics/akaike-information-criterion/ (accessed on 14 January 2022).

[327] Analyttica Datalab. What is Bayesian Information Criterion (BIC)? Medium. 2019. Available online: https://medium.com/@analyttica/what-is-bayesian-information-criterion-bic-b3396a894be6 (accessed on 14 January 2022).

[328] van der Pas, S.; Gruenwald, P. Almost the Best of Three Worlds: Risk, Consistency and Optional Stopping for the Switch Criterion in Nested Model Selection. *Stat. Sin.* **2018**, *28*, 229–253. https://doi.org/10.5705/ss.202016.0011.

[329] *Datasets for Day-Ahead Electricity Prices (Version 3)*; 2020. https://doi.org/10.5072/zenodo.715409.

[330] Bharath, K. Understanding ReLU: The Most Popular Activation Function in 5 Minutes! Medium. 2020. Available online: https://towardsdatascience.com/understanding-relu-the-most-popular-activation-function-in-5-minutes-459e3a2124f (accessed on 14 January 2022).

[331] Stojiljković, M. Stochastic Gradient Descent Algorithm With Python and NumPy—Real Python. Realpython.com. 2021. Available online: https://realpython.com/gradient-descent-algorithm-python/ (accessed on 14 January 2022).

[332] GitHub—Dimkonto/ERC-DNN: Error Compensation On Deep Neural Network Model for Day-Ahead Electricity Price Forecasting, GitHub. 2022. Available online: https://github.com/dimkonto/ERC-DNN (accessed on 14 January 2022).

[333] Hale, J. Which Evaluation Metric Should You Use in Machine Learning Regression Problems? Medium. 2021. Available online: https://towardsdatascience.com/which-evaluation-metric-should-you-use-in-machine-learning-regression-problems-20cdaef258e (accessed on 14 January 2022).

[334] Maciejowska, K.; Nitka, W.; Weron, T. Day-Ahead vs. Intraday—Forecasting the Price Spread to Maximize Economic Benefits. *Energies* **2019**, *12*, 631. https://doi.org/10.3390/en12040631.