ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΡΟΣΑΡΜΟΣΤΙΚΗ ΕΠΕΞΕΡΓΑΣΙΑ ΔΕΔΟΜΕΝΩΝ ΕΚΠΑΙΔΕΥΣΗΣ ΓΙΑ ΜΟΝΤΕΛΑ ΕΚΜΑΘΗΣΗΣ

ΜΗΝΑΣ ΠΕΤΡΙΔΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

Τζιρίτας Νικόλαος
Αναπληρωτής Καθηγητής

Λαμία 2025

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΡΟΣΑΡΜΟΣΤΙΚΗ ΕΠΕΞΕΡΓΑΣΙΑ ΔΕΔΟΜΕΝΩΝ ΕΚΠΑΙΔΕΥΣΗΣ ΓΙΑ ΜΟΝΤΕΛΑ ΕΚΜΑΘΗΣΗΣ

ΜΗΝΑΣ ΠΕΤΡΙΔΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

Τζιρίτας Νικόλαος
Αναπληρωτής Καθηγητής

Λαμία 2025

SCHOOL OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE & TELECOMMUNICATIONS

ADAPTIVE MANAGEMENT OF TRAIN DATASETS IN MACHINE LEARNING

PETRIDIS MINAS

FINAL THESIS

ADVISOR

Nikolaos Tziritas
Associate Professor

Lamia 2025

«Με ατομική μου ευθύνη και γνωρίζοντας τις κυρώσεις (1), που προβλέπονται από της διατάξεις της παρ. 6 του άρθρου 22 του Ν. 1599/1986, δηλώνω ότι:

1. *Δεν παραθέτω κομμάτια βιβλίων ή άρθρων ή εργασιών άλλων αυτολεξεί* ***χωρίς να τα περικλείω σε εισαγωγικά*** *και χωρίς να αναφέρω το συγγραφέα, τη χρονολογία, τη σελίδα. Η αυτολεξεί παράθεση χωρίς εισαγωγικά χωρίς αναφορά στην πηγή, είναι λογοκλοπή. Πέραν της αυτολεξεί παράθεσης, λογοκλοπή θεωρείται και η παράφραση εδαφίων από έργα άλλων, συμπεριλαμβανομένων και έργων συμφοιτητών μου, καθώς και η παράθεση στοιχείων που άλλοι συνέλεξαν ή επεξεργάσθηκαν, χωρίς αναφορά στην πηγή. Αναφέρω πάντοτε με πληρότητα την πηγή κάτω από τον πίνακα ή σχέδιο, όπως στα παραθέματα.*

2. *Δέχομαι ότι η αυτολεξεί* ***παράθεση χωρίς εισαγωγικά****, ακόμα κι αν συνοδεύεται από αναφορά στην πηγή σε κάποιο άλλο σημείο του κειμένου ή στο τέλος του, είναι αντιγραφή. Η αναφορά στην πηγή στο τέλος π.χ. μιας παραγράφου ή μιας σελίδας, δεν δικαιολογεί συρραφή εδαφίων έργου άλλου συγγραφέα, έστω και παραφρασμένων, και παρουσίασή τους ως δική μου εργασία.*

3. *Δέχομαι ότι υπάρχει επίσης περιορισμός στο μέγεθος και στη συχνότητα των παραθεμάτων που μπορώ να εντάξω στην εργασία μου εντός εισαγωγικών. Κάθε μεγάλο παράθεμα (π.χ. σε πίνακα ή πλαίσιο, κλπ), προϋποθέτει ειδικές ρυθμίσεις, και όταν δημοσιεύεται προϋποθέτει την άδεια του συγγραφέα ή του εκδότη. Το ίδιο και οι πίνακες και τα σχέδια*

4. *Δέχομαι όλες τις συνέπειες σε περίπτωση λογοκλοπής ή αντιγραφής.*

Ημερομηνία: 05/03/2025

Ο – Η Δηλ.

(1) «Όποιος εν γνώσει του δηλώνει ψευδή γεγονότα ή αρνείται ή αποκρύπτει τα αληθινά με έγγραφη υπεύθυνη δήλωση
του άρθρου 8 παρ. 4 Ν. 1599/1986 τιμωρείται με φυλάκιση τουλάχιστον τριών μηνών. Εάν ο υπαίτιος αυτών των πράξεων
σκόπευε να προσπορίσει στον εαυτόν του ή σε άλλον περιουσιακό όφελος βλάπτοντας τρίτον ή σκόπευε να βλάψει άλλον, τιμωρείται με κάθειρξη μέχρι 10 ετών.»

ΠΕΡΙΛΗΨΗ

Η ραγδαία ανάπτυξη των τεχνολογιών IoT και των όλο και πιο έξυπνων έξυπνων συσκευών έχει οδηγήσει σε τεράστια αύξηση της καθημερινής παραγωγής δεδομένων. Η παραδοσιακή υπολογιστική νέφους δυσκολεύεται να χειριστεί λειτουργίες σε πραγματικό χρόνο λόγω συμφόρησης του δικτύου και υπερφόρτωσης της κυκλοφορίας, οδηγώντας σε σημαντική καθυστέρηση. Το Edge Computing (EC) έχει αναδειχθεί ως λύση, επιτρέποντας τη λήψη αποφάσεων σε πραγματικό χρόνο με την επεξεργασία δεδομένων πιο κοντά στις τελικές συσκευές. Ενώ η EC μειώνει αποτελεσματικά την καθυστέρηση, περιορίζεται από περιορισμένους υπολογιστικούς πόρους και αποθηκευτικούς χώρους, καθιστώντας την αποτελεσματική διαχείριση των δεδομένων κρίσιμη.

Για να το αντιμετωπίσουμε αυτό, αναπτύξαμε μια μέθοδο για τον εντοπισμό μετατοπίσεων συνόλου δεδομένων, αξιοποιώντας προ-εκπαιδευμένα μοντέλα για την εξαγωγή χαρακτηριστικών από τα εισερχόμενα δεδομένα. Η προσέγγιση αυτή ενισχύει την ικανότητα εντοπισμού σχετικών δεδομένων, εξασφαλίζοντας μεγαλύτερη ακρίβεια στην επεξεργασία σε πραγματικό χρόνο και βελτιστοποιώντας παράλληλα τη χρήση των πόρων σε περιβάλλοντα ακμών.

ABSTRACT

The rapid development of IoT technologies and increasingly intelligent smart devices has resulted in an immense surge in daily data generation. Traditional cloud computing struggles to handle real-time operations due to network congestion and traffic overload, leading to significant latency. Edge Computing (EC) has emerged as a solution, enabling real-time decision-making by processing data closer to end devices. While EC effectively reduces latency, it is constrained by limited computational resources and storage, making efficient data management critical.

To address this, we developed a method to identify dataset shifts by leveraging pre-trained models to extract features from incoming data. This approach enhances the ability to detect relevant data, ensuring higher accuracy in real-time processing while optimizing resource utilization in edge environments.

Table of Contents

List of Figures

List of Tables

CHAPTER 1 Introduction

A network of interconnected smart items or things, such as smart devices, smart surroundings, sensors, networks, and a variety of computer equipment, that create an omnipresent system that allows access from anywhere at any time is referred to as pervasive computing, or ubiquitous computing. Mark Weiser first put forth this idea in 1988 while he was the director of the Computer Science Lab at Xerox's Palo Alto Research Center (PARC). According to his vision, computers will eventually become less noticeable as machines [3].

With the rapid development of technology, Weiser's vison came closer to reality. Smart devices are equipped with various sensors, processors and connectivity features which makes them able to seamlessly host services and communicate with each other to facilitate people's everyday tasks. For instance, smart homes can adjust lighting for power saving, temperature and monitoring the perimeter of the house and alert for possible breach of security, while wearable health monitors provide real-time updates on vital signs, alerting users and healthcare providers to potential issues.

The Internet of Things (IoT) is made possible by the devices' smooth interaction and integration. The Internet of Things, or IoT, is a self-configuring wireless network of commonplace smart gadgets with the goal of connecting everything. The Internet of Things has paved the way for the era of pervasive networking and computing. IoT technology is used in the healthcare industry to store and analyze data for later use [4]. Similar to this, IoT applications in agriculture affect supply chains, quality control, and market accessibility, which in turn affects the entire system of food production and distribution [5].

However, the integration of IoT into ubiquitous computing raises challenges, including issues related to data privacy, security, and standardization. As more devices become interconnected, the risk of cyberattacks increases leading to attacks or even leakage of sensitive data [6]. In order to derive useful insights from the massive volume of data produced by IoT devices, effective storage, processing, and analysis solutions are also needed. To efficiently handle this vast size of data that IoT produce, Edge Computing (EC) and Artificial Intelligent (AI) are being adopted [7].

IoT devices transferring data to cloud, a data center where data that IoT produce is stored. Unlike the ability to keep a vast amount of data, cloud commonly can't response due to overloading or due to heavy traffic [8]. EC came to solve this problem by bringing data closer to the user. By lowering latency and bandwidth consumption, increasing connectivity, facilitating quicker computation and service delivery, guaranteeing scalability, streamlining storage, lowering network expenses, and boosting overall application performance, edge computing aims to boost network performance. [9].

Distributed analytics using collaborative sensing is one example of such an application [11]. An EC receives a query from a user and responds using the information at hand [10]. Unlike cloud, EC has significant much less storage, so data management is essential for ensuring seamless operation. The importance of data management attracts the interest of researchers so that more and more algorithms and methodologies are being created.

Due to this limitation, EC must be able recognize and relate incoming data to existing datasets effectively. Every dataset can be represented as a distribution, where the statistical properties such as mean, variance and underlying patterns characterize the data. Changes at these distributions refers as data drift, more specifically data drift occurs when the statistical properties of incoming data diverge from the original datasets. In machine learning, this phenomenon leads to accuracy reduction of the model. To solve this problem, we developed a method to identify datasets shifts by leveraging pre-trained models to extract features from incoming data. This approach enhances the ability to detect relevant data, ensuring higher accuracy in real-time processing while optimizing resource utilization in edge environments.

Abbreviations	Expansion
EC	Edge Computing
IoT	Internet of Things
MLP	Multi – Layer Perceptron
EMD	Earth Mover's Distance
PSI	Population Stability Index

CHAPTER 2 Literature Review

Edge Computing 2.1

Edge Computing definition 2.1.a

More smart gadgets are connecting to the Internet and producing massive volumes of data as a result of the Internet of Things' (IoT) quick development. Response times are increasing, bandwidth usage is increasing, and cloud computing security and privacy are becoming concerns. Edge computing emerged as the answer as a result of all of this. "Edge computing is the technology that allows computation at the network edge, enabling downstream data processing for the cloud and upstream data processing for IoT services [12]."

Edge computing processes all data generated by IoT devices and transmits only essential information to the cloud. Its primary goal is to optimize network performance by minimizing latency and bandwidth usage, improving connectivity, enabling faster service delivery and computation, ensuring scalability, optimizing storage, reducing network costs, and enhancing overall application performance [13].

Applications 2.1.b

Internet of Things (IoT)

Edge computing is an important part of IoT as it processes the huge data produced by IoT devices at the network edge. This minimizes latency for end users, facilitates real-time decision-making, and lowers bandwidth consumption.

Smart cities

Edge computing can be a huge benefit for smart cities. Data from various sensors can be processed in real time to control traffic in an optimal manner and enhance public safety. Further, environmental metrics such as air quality and noise pollution can be effectively monitored. By processing data in real time, cities can respond to emergencies in a timely manner, optimize resource utilization, and become operationally more efficient [13].

Autonomous vehicles

For autonomous cars, when making snap decisions is crucial, edge computing is crucial. Data uploading to the cloud is slowed down by autonomous cars gathering data and interacting with one another. As a result, the edge device itself must be capable of processing signals, audio, video, and other types of data. With the right services, edge computing can create real-time communication between cars and avert a potential collision.

Healthcare

Edge Computing can aid in healthcare by providing real time monitoring and collecting information from the patient remotely leading to faster diagnoses and rapid responses to emergencies. Protective health-monitoring systems track the health indica tors of the user and check for the occurrence of an emergency health situation. In the case of an emergency, an alert is produced and the emergency contact informed [35].

Pros and cons 2.1.c

Edge computing processes and stores data on the edge without using cloud uploads, providing numerous advantages over standard cloud computing.

I. Quick Data Processing and Real-Time Analysis: Edge computing allows for lightning-fast reaction times and processing of real-time data. Being close to the data source and storage enables the calculations to be carried out directly on the edge nodes, which reduces data transmission time.
II. Enhanced Security: Unlike cloud computing, which requires data to be uploaded for processing, edge computing operates locally and minimizes risks such as data leaks and loss. Since data is processed within its local scope, network transmission risks are eliminated. Additionally, if a cyberattack occurs, only local data is affected rather than the entire dataset.
III. Reduced Network Traffic: Data processing and analyze, executes local to the edge node. The need to load the data to cloud is not necessary, so the traffic of the network can be reduced. The processing and analysis of data is carried out at the edge of the network. This way, the need to send large data packages to the cloud is minimized. It in turn reduces the network load and optimizes the use of the channel.

Despite its advantages, the Edge Computing paradigm is faced with some challenges:

I. Limited Computational Resources: Although edge computing aims to transfer computation from the cloud to edge nodes, the nodes' processing power, memory, and storage are far lower than those of cloud servers. Performance may be impacted by this restriction, particularly in applications that require a lot of processing power.
II. Diverse Environments: Edge computing encompasses a variety of platforms, devices, and architectures, creating a diverse environment. It can be difficult to ensure compatibility and smooth integration across all devices.
III. Security Risks: While security is one of the benefits of edge computing, it also has its vulnerabilities. Since edge devices are closer to users, they have higher exposure to cyber threats. Moreover, because of their limited resources, conventional security protection techniques may not be completely deployable, making them more vulnerable to attacks.

Data drift 2.2

Definition 2.2.a

Data drift refers to changes in the statistical properties of input data over time. This occurs when a machine learning model in production encounters data that differs from the data it was trained on or from earlier production data.

Such shifts in input distribution can degrade the model's performance, as machine learning models are designed to perform well on data resembling their training set. When the input data changes significantly, the model may struggle to make accurate predictions or decisions, highlighting its limited ability to generalize beyond its training.

In essence, data drift represents a shift in input data that the model is unprepared to handle. Detecting and addressing data drift is essential to ensure the reliability and effectiveness of machine learning models in ever-changing environments.

Data drift detectors 2.3

The gradual change in data can result in poor performance of the model [34]. Drift detection involves monitoring the overall data distribution across an entire dataset to identify significant shifts compared to a previous period or the data used during model training. The primary objective is to determine whether the model can still be trusted to perform as expected and whether the operating environment remains consistent with the conditions under which the model was trained.

CHAPTER 3 Methodology

Problem 3.1

Machine learning, or ML, has gained popularity in computer science in recent years. Because machine learning (ML) can tackle issues like fraud protection, email spam filters, speech recognition, and many more, it has revolutionized computer science. Today's massive data availability and next-generation computing power have made machine learning more sophisticated and dependable. Such data must be processed and stored in traditional distributed systems in order to be used in machine learning techniques [8]. These apps worked well for jobs requiring little latency. The Internet of Things (IoT) is developing quickly, and gathering all the data to process and analyze on a single server could result in unneeded latency and increased bandwidth. Edge computing solved this problem by bringing the processing closer to where data is generated but with one huge disadvantage, limited storage. Edge nodes must be able to decide which data to keep and which to reject as not being relevant. In this paper we introduce a new method to detect data drift with pre trained models.

Dataset 3.2

We utilize the FSDKaggle2018 dataset for this experiment. 11,073 audio clips with 41 labels from the AudioSet Ontology make up the Freesound Dataset Kaggle 2018 (FSDKaggle2018) [22]. The Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge 2018 Task 2 made use of the dataset [23]. Every sample was collected from Freesound and divided into test and train sets. We will divide the train set, which consists of 9.5k samples that are appropriately distributed over 41 categories, into three equal subsets of 3.1k samples each for this experiment. We generate for each sample the Mel Spectrogram and then transform from the amplitude scale to the decibel scale and padded to meet the dimensions of the biggest sample.

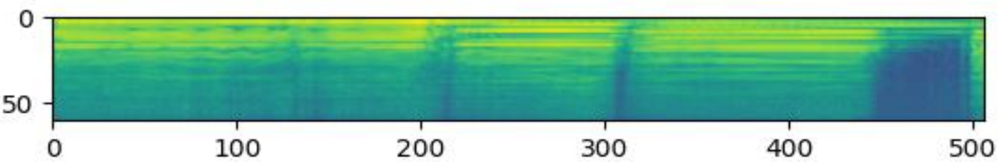

Figure 1: Audio sample

Models 3.3

For this experiment we will use 3 models which will be trained on the first dataset:

- Main model
- Multi – Layer Perceptron model
- Attention model

Main model 3.3.a

The model we use is a Convolutional Neural Network (CNN) for image classification with 8 ResBlocks based on ResNet architecture (Residual Network) [25]. ResBlocks are formed with 2 Convolution layers with ReLu activations. Every ResBlock takes the input and adds it to output after convolution layer 2, similar to ResNet architecture which takes the activations from n-1 convolution and adds it to the output of n+1 layer.

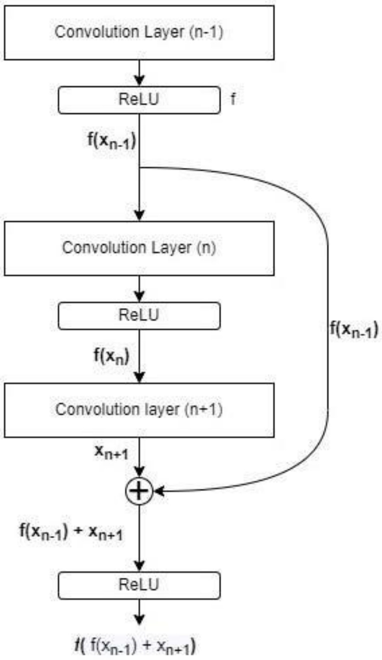

Figure 2: ResNet architecture. Source medium.com

Multi-Layer Perceptron 3.3.b

Each of the layers that make up MLP—an input layer, hidden layers, and an output layer—contains a collection of perceptual components called neurons [31]. Each layer is a fully connected layer, videlicet, every neuron in the layer is connected to every neuron in the previous and subsequent layer. In this experiment we use a MLP without a hidden layer.

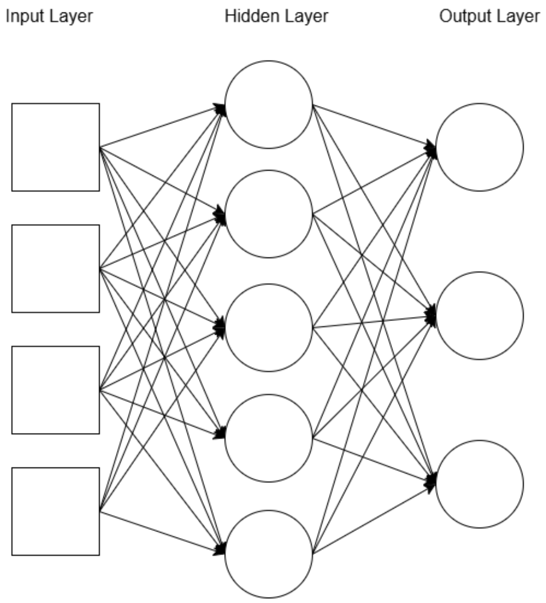

Figure 3: Multi-Layer Perceptron (MLP) architecture

Attention Mechanism 3.3.c

An attention function can be defined as a mapping that generates an output given a query and a collection of key-value pairs. In this instance, the output, values, keys, and query are all vectors. A compatibility function that determines how well the query matches the key determines the weight of each value, which is then used to produce the output as a weighted sum of the values [33].

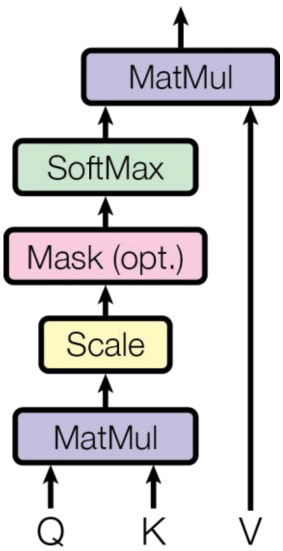

Figure 4: Scale Dot-Product Attention [12]

For this experiment we use Scaled Dot-Product Attention. For an input sequence of $vectors\ X = [x_1, x_2, \dots, x_n]$, where x_i is a vector of the i-th element, self-attention Y computes as:

$$Y = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

The weight matrices W_Q, W_K, and W_V, respectively, parameterize the learnt linear transformations of the input sequence X, Q (Query), K (Key), and V (Value). The dimension of the key vectors, d_k, is frequently selected to coincide with the dimensions of the value and query vectors. The softmax function that is applied along the matrix's rows is called SoftMax. The matrix of dot products between query and key vectors is denoted as QK^T. The value vectors are weighted using the resulting matrix.

Combined Models 3.3.d

The other two models are a combination of MLP output layer and Self-Attention mechanism.

- MLP + Attention: For this model we concatenate MLP with self-attention by feeding forward the output of MLP into Self-attention
- Attention + MLP: For this model we reverse the previous model

Both of these models are not trained, however the weights for each layer is extracted from the Multi – Layer Perceptron (MLP) and Self-Attention which are already trained.

Data drift detectors 3.4

For this experiment we used various data drift detectors to compare the accuracy of our approach. We choose 5 already know algorithms:

- Histogram Intersection
- Earth Mover's Distance (EMD)
- Energy Distance
- Population Stability Index (PSI)
- Hellinger Distance

Histogram Intersection

Histogram intersection is a method for comparing two histograms. In machine learning it can be used as a similarity metric for comparing features. Every object/feature is represented as a histogram, a graphical representation of the value distribution of the object/feature. Intersection is defined as:

$$\sum_{j=1}^{n} min(I_j, M_j)$$

Where I input histogram and M histogram of model, in our case histogram of the dataset our model is trained, each one containing n bins. The result of intersection is the number of features from the initial dataset that is similar to the input dataset.

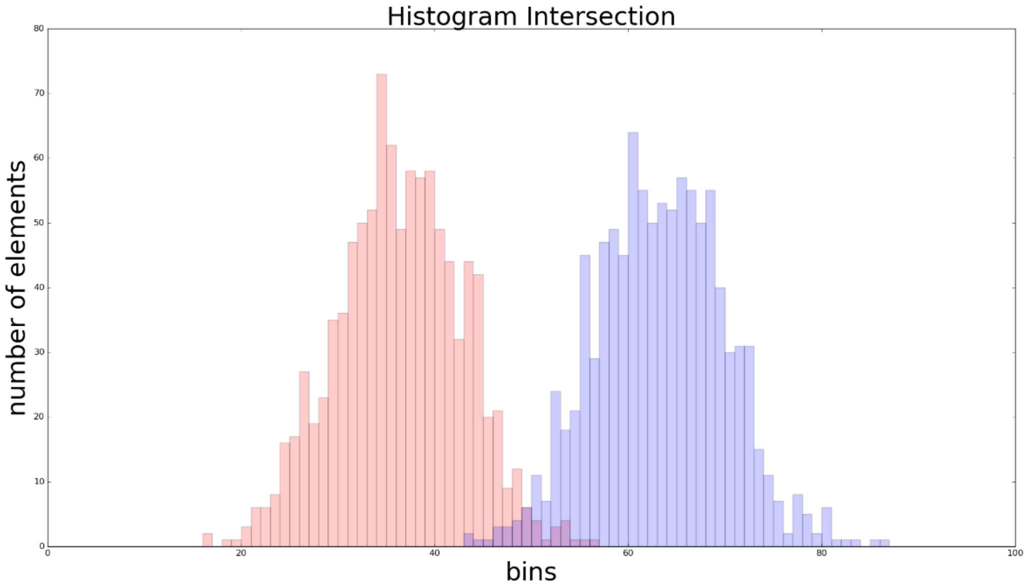

Figure 5: Histogram intersection Source: [github.io]

Earth Mover's Distance (EMD)

A measure of the lowest cost of converting one distribution into another is called Earth Mover's Distance (EMD). Informally, picture one distribution as a pile of dirt that is dispersed throughout an area, and the other as a group of holes in the area. The least amount of "work" needed to move the dirt to fill the holes is then measured by EMD. In essence, it determines how little work is required to shift the mass from the first distribution to the second [27].

Energy Distance

Energy distance calculates the distance between two probability distributions. It works on the concept of potential energy, which is only zero when the centers of gravity (or locations) of the two distributions are the same. The further apart the distributions are, the higher the potential energy, representing the widening gap between the two distributions [30].

Population Stability Index (PSI)

The Population Stability Index (PSI) determines if the training dataset's Y (response variable) distribution and an out-of-sample dataset's (test data) scoring variable distribution are identical. Generally speaking, determines if the distribution of Y differs from the dataset used to train the model. Based on PSI we can draw conclusions:

- PSI < 0.1: No major change
- PSI < 0.2: Moderate population change
- PSI ≥ 0.2: Significant population change

Hellinger Distance

The Hellinger Distance measures the similarity between two probability distributions. It is derived from the Hellinger Integral, which quantifies the difference between probability measures. Mathematically, the Hellinger Distance is defined as the square root of the squared Hellinger distance, expressed by the formula:

$$H(P,Q) = \frac{1}{\sqrt{2}}||\sqrt{P} - \sqrt{Q}||^2$$

where P and Q represent two probability distributions. The distance ranges from 0 to 1, with 0 indicating identical distributions and 1 representing complete dissimilarity.

CHAPTER 4 Experiment Analysis

In EC, data processing in edge nodes is resource constrained making efficient data management critical. Data drift detection is crucial for maintaining the accuracy, efficiency, and reliability of EC systems. By addressing drift proactively, edge nodes can adapt to real-world changes, optimize resource use, and reduce risks, ensuring long-term operational success in dynamic environments.

In this experiment, feature extraction techniques were applied to a dataset, followed by an evaluation of data drift detection methods. The primary objective was to assess the effectiveness of different techniques in identifying and quantifying drift between datasets (D1, D2, and D3).

As we are aforementioned, we spilt the data into 3 equal datasets, D1, D2, D3. Every model is trained with D1.

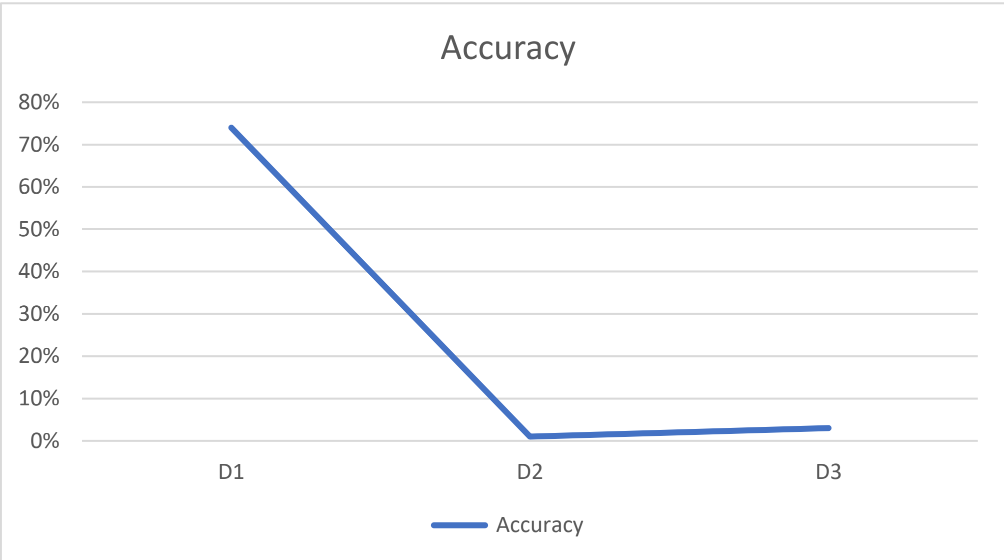

Figure 6: Accuracy of the main model with each dataset

As we can see from Figure 6 the model achieved accuracy 74%, 1% and 3% for D1, D2 and D3 respectively. The sudden accuracy drop is due to the changes of the data distribution that the model was trained causing the phenomenon of data drift. From the above we conclude that the distributions of datasets differ.

Data Drift detector	D1 – D2	D1 – D3
Histogram Intersection	0.264613679544014	0.264930335655477
Earth Mover’s Distance (EMD)	202.617437288001	202.273765427601
Energy Distance	5.48183135533463	5.4678152654039
Population Stability Index (PSI)	104.57885210844	104.707514881253
Hellinger Distance	0.322018377158285	0.322190498563739

Table 1: Results of drift detectors comparing generic data

From Table 1 we can see indeed that there is a data drift between the main dataset and the two others. Each detector gives results with high reliability, except for the Hellinger distance, where 0.3 converges more closely to the similarity of the datasets.

Model	Histogram Intersection	EMD	Energy distance	PSI	Hellinger Distance
MLP	0.526669921875	3654.56346828163	33.058307590546	357.949155401496	0.564374087615415
Attention	0.53740234375	133.902938802962	6.38888915278142	372.588336292568	0.574020231540249
MLP + Attention	0.527060546875	8722.52469531063	51.070447414525	360.372999946302	0.565087611454375
Attention + MLP	0.537685546875	3274.59285228009	32.4338204939169	374.655572081321	0.575230950301049

Table 2: Results between D1-D2 after inference each dataset

Model	Histogram Intersection	EMD	Energy distance	PSI	Hellinger Distance
MLP	0.52986328125	3692.63832247536	33.3510358543584	363.289536236302	0.567680645004591
Attention	0.5381640625	134.385678553739	6.37084599713929	369.972668935165	0.573285356249377
MLP + Attention	0.527744140625	8731.37590336972	50.9767718864323	361.266299385938	0.56567568392439
Attention + MLP	0.536005859375	3288.89921422952	32.407265990545	371.32712763968	0.573687002191916

Table 3: Results between D1-D3 after inference each dataset

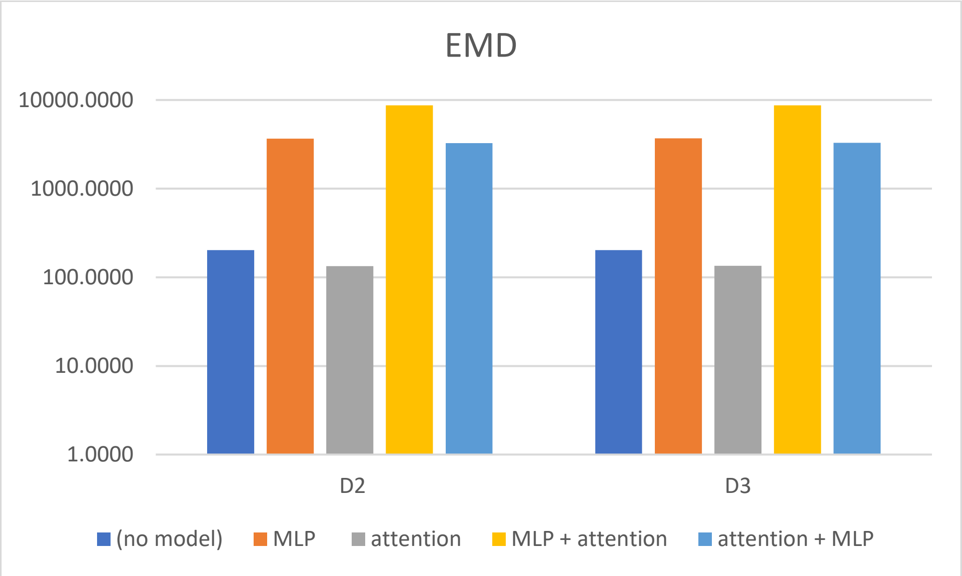

Figure 7: Results of EMD

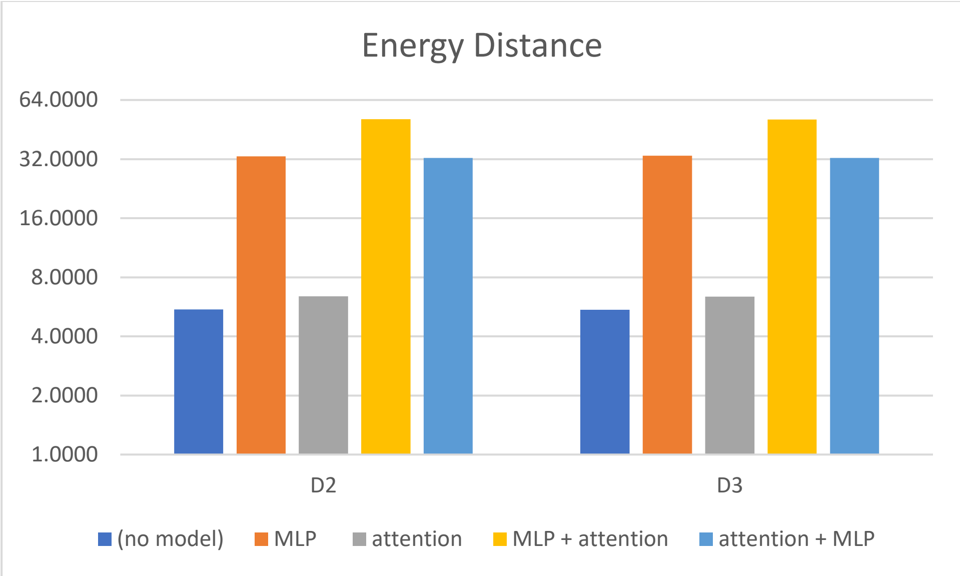

Figure 8:Results of Energy Distance

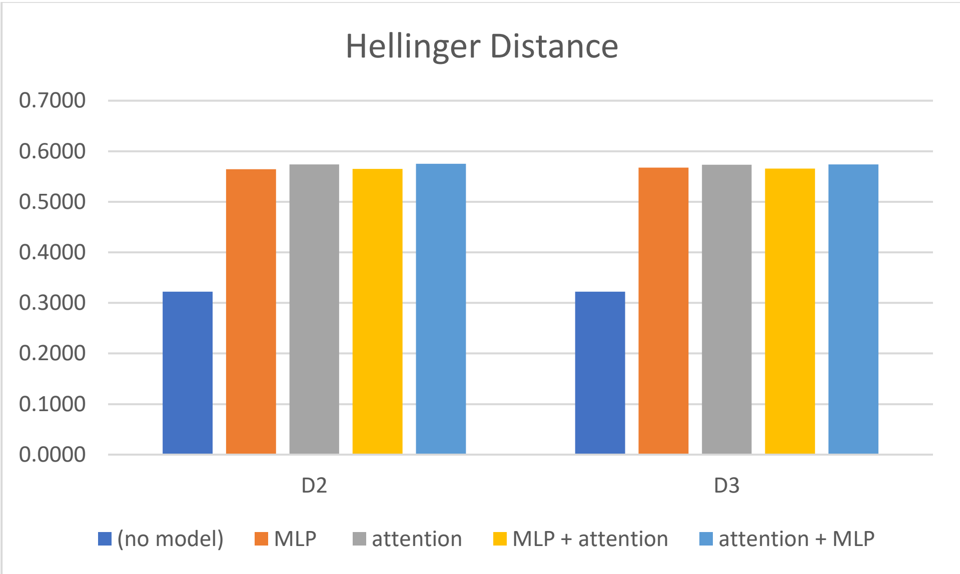

Figure 9: Results of Hillinger Distance

For our approach we inference the models with D1, D2, D3 to extract the features and then we compare them instead of the unprocessed datasets like Table 1. As we can see from Table 2 and Table 3, the confidence improved overall with EMD having an increase in MLP and MLP + attention models but a decrease in attention model (Figure 7). Energy distance with Attention model did improve but with lower rate (Figure 8). Hillinger Distance improved as the value came closer to 1 (Figure 9).

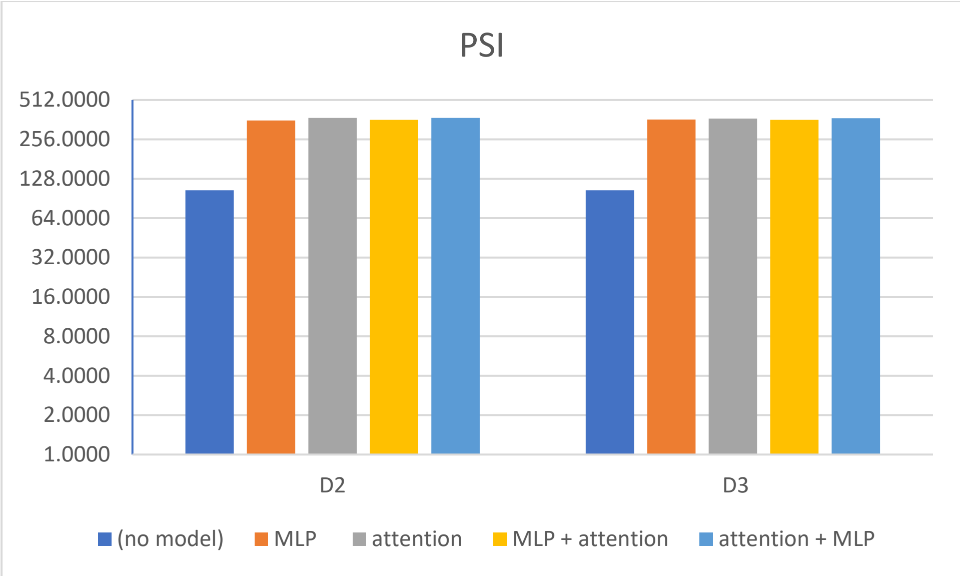

Figure 10: Results of PSI

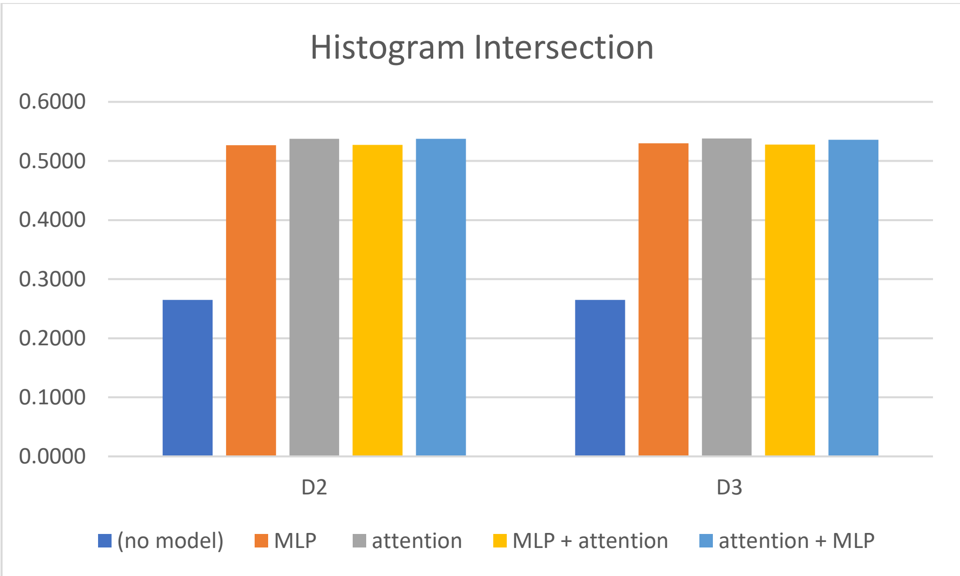

Figure 11: Results of Histogram Intersection

PSI made an improvement for all models resulting in similar results (Figure 10), unlike the Histogram Intersection which worsened for all models (Figure 11) as an increase in score means an increase in similarity.

CHAPTER 5 Conclusion

This thesis has explored the challenges of data drift in edge computing environments, where resource constraints and real-time data processing are critical. The research introduced a novel approach to detecting dataset shifts by leveraging pre-trained models to extract features from incoming data. This method improves the accuracy of data drift detection, ensuring better performance and optimized resource utilization in edge environments.

Through a series of experiments using the FSDKaggle2018 dataset, the effectiveness of various drift detection methods—such as Histogram Intersection, Earth Mover's Distance, Energy Distance, Population Stability Index, and Hellinger Distance was evaluated. Results showed that integrating feature extraction significantly enhances the reliability of these methods. For example, models such as MLP, Attention, and their combinations demonstrated overall improved accuracy in identifying and quantifying dataset shifts compared to traditional unprocessed datasets.

The findings underscore the importance of adapting machine learning models to evolving data distributions in real-world scenarios. By addressing data drift proactively, edge computing systems can maintain high levels of performance and reliability, even in dynamic environments with limited computational resources.

REFERENCES

1. L. Yanjun, W. Wei, L. Bo and H. Chengming, "Research of Pervasive Computing System Support Theory," 2010 First International Conference on Pervasive Computing, Signal Processing and Applications, Harbin, China, 2010, pp. 9-12, doi: 10.1109/PCSPA.2010.11.
2. Springer International Publishing Switzerland 2016 D.P.F. Mo¨ller, Guide to Computing Fundamentals in Cyber-Physical Systems, Computer Communications and Networks, DOI 10.1007/978-3-319-25178-3_5
3. Mark Weiser.The Computer for the 21st Century. ScientificAmerican,265(3):94-104, 1991.
4. J. -C. Liau and C. -Y. Ho, "Intelligence IoT(Internal of Things) Telemedicine Health Care Space System for the Elderly Living Alone," 2019 IEEE Eurasia Conference on Biomedical Engineering, Healthcare and Sustainability (ECBIOS), Okinawa, Japan, 2019, pp. 13-14, doi: 10.1109/ECBIOS.2019.8807821.
5. G. Singh and J. Singh, "Transformative Potential of IoT for Developing Smart Agriculture System: A Systematic Review," 2023 4th International Conference on Communication, Computing and Industry 6.0 (C216), Bangalore, India, 2023, pp. 1-6, doi: 10.1109/C2I659362.2023.10430789.
6. P. Fountas, M. Papathanasaki, K. Kolomvatsos and N. Tziritas, "Proactive Attack Detection at the Edge through an Ensemble Deep Learning Model," 2021 20th International Conference on Ubiquitous Computing and Communications (IUCC/CIT/DSCI/SmartCNS), London, United Kingdom, 2021, pp. 19-24, doi: 10.1109/IUCC-CIT-DSCI-SmartCNS55181.2021.00018.
7. T. Loukopoulos et al., "Data Stream Processing at Network Edges," 2018 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), Vancouver, BC, Canada, 2018, pp. 657-665, doi: 10.1109/IPDPSW.2018.00106.
8. Deepak, M. K. Upadhyay and M. Alam, "Edge Computing: Architecture, Application, Opportunities, and Challenges," 2023 3rd International Conference on Technological Advancements in Computational Sciences (ICTACS), Tashkent, Uzbekistan, 2023, pp. 695-702, doi: 10.1109/ICTACS59847.2023.10390171.
9. K. Kolomvatsos and C. Anagnostopoulos, "A Proactive Statistical Model Supporting Services and Tasks Management in Pervasive Applications," in IEEE Transactions on Network and Service Management, vol. 19, no. 3, pp. 3020-3031, Sept. 2022, doi: 10.1109/TNSM.2022.3161663.
10. S. Pasteris, S. Wang, C. Makaya, K. Chan and M. Herbster, "Data distribution and scheduling for distributed analytics tasks," 2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), San Francisco, CA, USA, 2017, pp. 1-6, doi: 10.1109/UIC-ATC.2017.8397417.
11. A. Mukherjee, H. S. Paul, S. Dey and A. Banerjee, "ANGELS for distributed analytics in IoT," 2014 IEEE World Forum on Internet of Things (WF-IoT), Seoul, Korea (South), 2014, pp. 565-570, doi: 10.1109/WF-IoT.2014.6803230.
12. W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, ''Edge computing: Vision and challenges,'' IEEE Internet Things J., vol. 3, no. 5, pp. 637–646, Oct. 2016.
13. Deepak, M. K. Upadhyay and M. Alam, "Edge Computing: Architecture, Application, Opportunities, and Challenges," 2023 3rd International Conference on Technological Advancements in Computational Sciences (ICTACS), Tashkent, Uzbekistan, 2023, pp. 695-702
14. K. Cao, Y. Liu, G. Meng and Q. Sun, "An Overview on Edge Computing Research," in IEEE Access, vol. 8, pp. 85714-85728, 2020

15. G. Kaur and R. S. Batth, "Edge Computing: Classification, Applications, and Challenges," 2021 2nd International Conference on Intelligent Engineering and Management (ICIEM), London, United Kingdom, 2021, pp. 254-259
16. I. D. Id, M. Abe and S. Hara, "Concept Drift Adaptation for Acoustic Scene Classifier Based on Gaussian Mixture Model," 2020 IEEE REGION 10 CONFERENCE (TENCON), Osaka, Japan, 2020, pp. 450-455,
17. Id, Ibnu Daqiqil, Masanobu Abe and Sunao Hara. "Evaluation of concept drift adaptation for acoustic scene classifier based on Kernel Density Drift Detection and Combine Merge Gaussian Mixture Model." *ArXiv* abs/2105.13220 (2021): n. pag.
18. Hoens, T.R., Polikar, R. & Chawla, N.V. Learning from streaming data with concept drift and imbalance: an overview. *Prog Artif Intell* **1**, 89–101 (2012). https://doi.org/10.1007/s13748-011-0008-0
19. F. Marozzo, A. Orsino, D. Talia and P. Trunfio, "Edge Computing Solutions for Distributed Machine Learning," 2022 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), Falerna, Italy, 2022, pp. 1-8,
20. G. Ditzler and R. Polikar, "Incremental Learning of Concept Drift from Streaming Imbalanced Data," in IEEE Transactions on Knowledge and Data Engineering, vol. 25, no. 10, pp. 2283-2301, Oct. 2013, doi: 10.1109/TKDE.2012.136.
21. Lu, J., Liu, A., Dong, F., Gu, F., Gama, J., Zhang, G.: Learning under Concept Drift: A Review. IEEE Transactions on Knowledge and Data Engineering 31(12), 2346–2363 (Dec 2019)
22. J. F. Gemmeke et al., "Audio Set: An ontology and human-labeled dataset for audio events," 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 2017, pp. 776-780, doi: 10.1109/ICASSP.2017.7952261
23. Eduardo Fonseca, Manoj Plakal, Frederic Font, Daniel P. W. Ellis, Xavier Favory, Jordi Pons, Xavier Serra. "General-purpose Tagging of Freesound Audio with AudioSet Labels: Task Description, Dataset, and Baseline". *Proceedings of the DCASE 2018 Workshop* (2018)
24. Frederic Font, Gerard Roma, and Xavier Serra. "Freesound technical demo." Proceedings of the 21st ACM international conference on Multimedia, 2013. https://freesound.org
25. K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90
26. Swain, M. J., and D. H. Ballard. "Color Indexing International Journal of Computer Vision 7." (1991): 11-32.
27. Rubner, Y., Tomasi, C. & Guibas, L.J. The Earth Mover's Distance as a Metric for Image Retrieval. *International Journal of Computer Vision* **40**, 99–121 (2000). https://doi.org/10.1023/A:1026543900054
28. Gábor J. Székely, Maria L. Rizzo, Energy statistics: A class of statistics based on distances, Journal of Statistical Planning and Inference, Volume 143, Issue 8, 2013,
29. Wu, D., Olson, D. Enterprise risk management: coping with model risk in a large bank. *J Oper Res Soc* **61**, 179–190 (2010). https://doi.org/10.1057/jors.2008.144
30. WIREs Comput Stat 2016, 8:27–38. doi: 10.1002/wics.1375
31. Kit Yan Chan, Bilal Abu-Salih, Raneem Qaddoura, Ala' M. Al-Zoubi, Vasile Palade, Duc-Son Pham, Javier Del Ser, Khan Muhammad, Deep neural networks in the cloud: Review, applications, challenges and research directions, Neurocomputing, Volume 545, 2023, 126327

32. Zhaoyang Niu, Guoqiang Zhong, Hui Yu, A review on the attention mechanism of deep learning, Neurocomputing, Volume 452, 2021,
33. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin, Attention Is All You Need, https://arxiv.org/abs/1706.03762
34. N. L. A. Ghani, I. A. Aziz and M. Mehat, "Concept Drift Detection on Unlabeled Data Streams: A Systematic Literature Review," 2020 IEEE Conference on Big Data and Analytics (ICBDA), Kota Kinabalu, Malaysia, 2020, pp. 61-65, doi: 10.1109/ICBDA50157.2020.9289802
35. A. O. Akmandor and N. K. Jha, "Smart Health Care: An Edge-Side Computing Perspective," in IEEE Consumer Electronics Magazine, vol. 7, no. 1, pp. 29-37, Jan. 2018, doi: 10.1109/MCE.2017.2746096.