ΤΜΗΜΑ ΛΟΓΙΣΤΙΚΗΣ ΚΑΙ ΧΡΗΜΑΤΟΟΙΚΟΝΟΜΙΚΗΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

Artificial Intelligence and Metaheuristic Optimization Algorithms for Financial Risk Assessment and Investment Decision-Making

Αλγόριθμοι Τεχνητής Νοημοσύνης και μεταευρετικής βελτιστοποίησης για την εκτίμηση χρηματοοικονομικού κινδύνου και τη λήψη επενδυτικών αποφάσεων

ΓΚΟΝΗΣ ΒΑΣΙΛΕΙΟΣ

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ : ΤΣΑΚΑΛΟΣ ΙΩΑΝΝΗΣ

Διδακτορική Διατριβή που υποβάλλεται στο καθηγητικό σώμα:

1. ΤΣΑΚΑΛΟΣ ΙΩΑΝΝΗΣ
2. ΦΑΣΣΑΣ ΑΘΑΝΑΣΙΟΣ
3. ΑΝΔΡΙΚΟΠΟΥΛΟΣ ΑΝΔΡΕΑΣ

για την εκπλήρωση των υποχρεώσεων απόκτησης του διδακτορικού τίτλου του Προγράμματος Διδακτορικών Σπουδών του Τμήματος Λογιστικής και Χρηματοοικονομικής του Πανεπιστημίου Θεσσαλίας

ΛΑΡΙΣΑ , 2025

ΕΠΤΑΜΕΛΗΣ ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ

Τσάκαλος Ιωάννης
Αναπληρωτής Καθηγητής
Τμήμα Λογιστικής και Χρηματοοικονομικής
Πανεπιστήμιο Θεσσαλίας

Φάσσας Αθανάσιος
Καθηγητής
Τμήμα Λογιστικής και Χρηματοοικονομικής
Πανεπιστήμιο Θεσσαλίας

Ανδρικόπουλος Ανδρέας
Αναπληρωτής καθηγητής
Τμήμα Ναυτιλιακών Σπουδών
Πανεπιστήμιο Πειραιά

Καμπούρης Ηλίας
Αναπληρωτής Καθηγητής
College of Business
Abu Dhabi University

Κυριαζής Νικόλαος
Επίκουρος Καθηγητής
Τμήμα Λογιστικής και Χρηματοοικονομικής
Πανεπιστήμιο Θεσσαλίας

Κώτσιος Στυλιανός
Καθηγητής
Τμήμα Οικονομικών επιστημών
Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών

Τσούτσα Παρασκευή
Επίκουρη Καθηγήτρια
Τμήμα Λογιστικής και Χρηματοοικονομικής
Πανεπιστήμιο Θεσσαλίας

Η έγκριση Διδακτορικής Διατριβής από το Τμήμα Λογιστικής και Χρηματοοικονομικής του Πανεπιστημίου Θεσσαλίας δεν υποδηλώνει την αποδοχή των γνωμών του συγγραφέα (Ν. 5343/1932, άρθρο 202, παρ.2)

Disclaimer of academic ethics

Με ατομική μου ευθύνη και έχοντας επίγνωση της δυνατότητας επιβολής κυρώσεων από τη Συνέλευση του Τμήματος, δηλώνω υπεύθυνα ότι για τη συγγραφή της Διδακτορικής μου Διατριβής δεν χρησιμοποίησα ολόκληρο ή μέρος έργου άλλου δημιουργού ή τις ιδέες και αντιλήψεις άλλου δημιουργού χωρίς να γίνεται αναφορά στην πηγή προέλευσης (βιβλίο, άρθρο από εφημερίδα ή περιοδικό, ιστοσελίδα κ.λπ.)

ΓΚΟΝΗΣ ΒΑΣΙΛΕΙΟΣ ____________________ ΛΑΡΙΣΑ, 2025

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Dr. Ioannis Tsakalos, for his valuable guidance and encouragement. His support was indispensable to this research. I also wish to thank the advisory committee for their constructive feedback and support across the duration of this work, as well as the people who shared ideas and had helpful discussions during this journey. Finally, I dedicate this thesis to my supervisor Dr. Ioannis Tsakalos and my family, especially my mother, Christothea Kosti, in recognition of the support they have provided me over the years.

Contents

Περίληψη

Ο κλάδος της χρηματοοικονομικής αντιμετωπίζει διαχρονικά σύνθετες προκλήσεις. Η παρούσα διατριβή, με τίτλο «Αλγόριθμοι Τεχνητής Νοημοσύνης και μεταευρετικής βελτιστοποίησης για την εκτίμηση χρηματοοικονομικού κινδύνου και τη λήψη επενδυτικών αποφάσεων», μελετά και αξιολογεί σύγχρονους αλγορίθμους Τεχνητής Νοημοσύνης και μεταευρετικής βελτιστοποίησης καθώς και προτείνει υβριδικά μοντέλα για την αντιμετώπιση κρίσιμων προβλημάτων, όπως η αξιολόγηση πιστωτικού κινδύνου σε χρηματοπιστωτικά ιδρύματα, η βελτιστοποίηση χαρτοφυλακίου και η πρόβλεψη τιμών κλεισίματος μετοχών. Αρχικά, παρουσιάζονται οι στόχοι και οι ερευνητικές προκλήσεις που επιχειρεί να απαντήσει η διατριβή. Έπειτα ακολουθεί βιβλιογραφική ανασκόπηση, η οποία αναδεικνύει το αυξανόμενο ερευνητικό ενδιαφέρον τα τελευταία χρόνια για αυτού του τύπου τους αλγορίθμους, τόσο στο ευρύτερο πεδίο της Χρηματοοικονομικής όσο και ειδικότερα στο κομμάτι της βελτιστοποίησης χαρτοφυλακίου.

Στη συνέχεια παρουσιάζονται διεξοδικά τέσσερις εφαρμογές στον τομέα του χρηματοοικονομικού κινδύνου και της λήψης επενδυτικών αποφάσεων. Πρώτα, μελετάται το πρόβλημα της φυγής πελατών από τα χρηματοπιστωτικά ιδρύματα, το οποίο αποτελεί απαιτητική πρόκληση όχι μόνο για τον τραπεζικό τομέα αλλά και για μια πληθώρα επιχειρήσεων σε διάφορους κλάδους. Εξετάζεται η απόδοση μοντέλων βαθιάς μάθησης σε συνάρτηση με την επιλογή συνάρτησης ενεργοποίησης και αλγορίθμου βελτιστοποίησης. Ερευνώνται διαφορετικές συναρτήσεις ενεργοποίησης και μέθοδοι βελτιστοποίησης με σκοπό την αξιολόγηση της απόδοσής τους, ενώ προτείνεται η χρήση της συνάρτησης APTx, η οποία φαίνεται ικανή να προσφέρει καλύτερα αποτελέσματα· η παρούσα εργασία είναι η πρώτη που προτείνει τη χρήση της σε πρακτική εφαρμογή. Επιπλέον, δεδομένου ότι τέτοιου τύπου προβλήματα εμφανίζουν έντονη ανισορροπία κλάσεων, αξιολογούνται διάφορες μέθοδοι αναδειγματοληψίας.

Στη δεύτερη εφαρμογή εξετάζεται μια ακόμη εφαρμογή του πιστωτικού κινδύνου, και πιο συγκεκριμένα η πρόβλεψη αθέτησης πληρωμής της πιστωτικής κάρτας. Για τον σκοπό αυτό προτείνεται ένα βελτιστοποιημένο μοντέλο μηχανικής μάθησης LightGBM, μέσω του μεταευρετικού αλγορίθμου Pareto-like Sequential Sampling (PSS). Η απόδοσή του PSS συγκρίνεται με έναν δημοφιλή αλγόριθμο Μπεϋζιανής βελτιστοποίησης, τον Tree-Structured Parzen Estimator (TPE). Επιπλέον, προτείνεται η χρήση των αναδιαταγμένων ακολουθιών Halton (Scrambled Halton Sequences, SHS). Όπως αναφέρουν οι συγγραφείς του PSS (Shaqfa & Beyer, 2021), ο συγκεκριμένος μεταευρετικός αλγόριθμος, που προσομοιάζει τον νόμο του Pareto, είναι συμβατός με οποιαδήποτε μέθοδο δειγματοληψίας. Τα αποτελέσματα έδειξαν ότι η προτεινόμενη χρήση του SHS προσφέρει καλύτερα αποτελέσματα και καλύτερο σφάλμα γενίκευσης σε σχέση με τις μεθόδους Monte Carlo και LHS. Επιπλέον, στη συγκεκριμένη εφαρμογή αξιοποιούνται οι ενσωματωμένες λειτουργίες του LightGBM για την αντιμετώπιση της ανισορροπίας κλάσεων και της πολυδιαστατικότητας. Τα αποτελέσματα δείχνουν ότι το προτεινόμενο μοντέλο προσφέρει καλύτερη επίδοση σε σύγκριση με πολλές παρόμοιες

εργασίες της βιβλιογραφίας, χωρίς την ανάγκη σύνθετων και ιδιαίτερα περίπλοκων αρχιτεκτονικών.

Έπειτα, διερευνάται η χρήση μεταευρετικών αλγορίθμων για τη βελτιστοποίηση χαρτοφυλακίου. Προτείνεται ειδικότερα ο Artificial Gorillas Troops Optimizer (AGTO). Συγκρίνονται δεκαέξι διαφορετικοί μεταευρετικοί αλγόριθμοι, προκειμένου να αποτιμηθεί η αποτελεσματικότητά τους στη βελτιστοποίηση χαρτοφυλακίου—πολλοί εκ των οποίων έχουν περιορισμένη έως ανύπαρκτη εφαρμογή στη σχετική βιβλιογραφία, όπως ο Βελτιστοποιητής Σμήνους Τόνου (Tuna Swarm Optimizer), ο Βελτιστοποιητής Τεχνητών Κουνελιών (Artificial Rabbits Optimizer) και ο Αλγόριθμος Θαλάσσιων Θηρευτών (Marine Predators Algorithm, MPA). Εξετάζεται επίσης η επίδοσή τους ως προς διαφορετικά μεγέθη πληθυσμού (*population size*) και αριθμό εποχών εκπαίδευσης (*epochs*). Βελτιστοποιώντας ένα χαρτοφυλάκιο που αποτελείται από τις μετοχές του δείκτη Dow Jones Industrial Average (DJIA), τα αποτελέσματα έδειξαν ότι ο AGTO μπορεί να προσφέρει τα καλύτερα αποτελέσματα στη βελτιστοποίηση χαρτοφυλακίου.

Τέλος, προτείνεται το υβριδικό μοντέλο GJO-GRU|LSTM, δηλαδή ένα βελτιστοποιημένο υβριδικό βαθύ νευρωνικό δίκτυο μέσω του αλγορίθμου των Χρυσών Τσακαλιών (Golden Jackal Optimizer, GJO), με σκοπό την πρόβλεψη της τιμής κλεισίματος μετοχών. Ενώ πολλές εργασίες στη βιβλιογραφία επιχειρούν τη βελτιστοποίηση των υπερπαραμέτρων δικτύων LSTM ή GRU για πρόβλεψη χρονοσειρών, στη συγκεκριμένη εφαρμογή επιχειρείται η βελτιστοποίηση τόσο των παραμέτρων όσο και της ίδιας της αρχιτεκτονικής: αναλόγως της περίστασης επιλέγεται δυναμικά ο τύπος δικτύου (GRU ή LSTM) και οι αντίστοιχες παράμετροί του. Η απόδοση του προτεινόμενου μοντέλου συγκρίνεται με δεκαέξι μοντέλα βαθιάς μάθησης, καθώς και με τα GA-GRU|LSTM και PSO-GRU|LSTM, τα οποία υποδηλώνουν βελτιστοποιημένα μοντέλα μέσω Γενετικών Αλγορίθμων (GA) και Βελτιστοποίησης Σμήνους Σωματιδίων (PSO). Τα αποτελέσματα έδειξαν ότι το GJO-GRU|LSTM παρέχει ακριβέστερες προβλέψεις στη μεγάλη πλειονότητα των περιπτώσεων.

Η διατριβή ολοκληρώνεται με τα κύρια ευρήματα και τη σύνοψη της συνεισφοράς. Στόχος είναι ο εμπλουτισμός της εφαρμοσμένης έρευνας σε αλγορίθμους Τεχνητής Νοημοσύνης και μεταευρετικής βελτιστοποίησης μέσω μοντέλων ικανών να υποστηρίζουν τεκμηριωμένες αποφάσεις σε καίρια ζητήματα της χρηματοοικονομικής. Τα τελευταία χρόνια, η εφαρμογή αυτών των αλγόριθμων έχει αυξηθεί, καθώς μπορούν να αντιμετωπίσουν προκλήσεις που τα κλασικά μοντέλα δυσκολεύονται να χειριστούν. Εκτιμούμε ότι η αξιολόγηση των αλγορίθμων που διερευνήθηκαν και οι προτεινόμενες προσεγγίσεις συμβάλλουν ουσιαστικά στο πεδίο, εμβαθύνοντας στην επίδρασή τους στη χρηματοοικονομική επιστήμη και εμπλουτίζοντας την υπάρχουσα βιβλιογραφία τόσο στον κλάδο της χρηματοοικονομικής όσο και στο ευρύτερο πεδίο της Τεχνητής Νοημοσύνης και της μεταευρετικής βελτιστοποίησης.

Λέξεις-Κλειδιά : Τεχνητή Νοημοσύνη · Μηχανική Μάθηση · Μεταευρετική Βελτιστοποίηση · Εκτίμηση χρηματοοικονομικού κινδύνου· Λήψη Επενδυτικών Αποφάσεων

Summary

The field of finance has long faced complex, difficult challenges. The present thesis, titled "*Artificial Intelligence and Metaheuristic Optimization Algorithms for Financial Risk Estimation and Investment Decision-Making*" studies and evaluates modern artificial intelligence and metaheuristic optimization algorithms and proposes hybrid models to address critical problems such as credit risk assessment in financial institutions, portfolio optimization, and stock closing-price prediction. It first presents the objectives and the research challenges the thesis seeks to address. This is followed by a literature review that highlights the growing research interest in these types of algorithms in recent years, both in the broader domain of Financial Science and, more specifically, in the area of portfolio optimization.

Next, four applications are presented in detail in the fields of financial risk and investment decision-making. First, the problem of customer churn in financial institutions is examined, which is a demanding challenge not only in banking but across many industries. The performance of deep learning models is evaluated with respect to the choice of activation function and optimization algorithm. Different activation functions and optimization methods are investigated with the aim of assessing their performance, while the use of the APTx activation function is proposed. The respective activation function appears capable of delivering better results, and this work is the first to recommend its use in a practical application. In addition, given that such problems typically exhibit pronounced class imbalance, various resampling methods are evaluated.

The second application explores another application of credit risk, the case of credit card default, an important challenge for financial institutions. For this purpose, an optimized LightGBM via the metaheuristic Pareto-like Sequential Sampling (PSS) algorithm is proposed. The performance of the proposed methodology compared against a popular Bayesian optimization method, the Tree-Structured Parzen Estimator (TPE). Furthermore, the use of Scrambled Halton Sequences (SHS) is proposed. As noted by the authors of PSS , this algorithm is compatible with any sampling method. The results showed that the proposed use of SHS provides more accurate results and better generalization performance compared to Monte Carlo and Latin Hypercube Sampling (LHS) methods. In this application, LightGBM's built-in functionalities are also used to address class imbalance and high dimensionality. The results show that the proposed model achieves better performance compared to similar studies in the literature, without the need to use complex architectures.

Subsequently, the use of metaheuristic algorithms for portfolio optimization is investigated. In particular, the Artificial Gorillas Troops Optimizer (AGTO) is proposed. Sixteen different metaheuristic algorithms are compared to assess their effectiveness in portfolio optimization—many of which have limited or no prior application in the literature—such as the Tuna Swarm Optimizer, the Artificial Rabbits Optimizer, and the Marine Predators Algorithm (MPA). Their performance is also examined with respect to different population sizes and numbers of training epochs. By optimizing a portfolio

consisting of the constituents of the Dow Jones Industrial Average (DJIA), the results showed that AGTO can provide the best performance in portfolio optimization.

Finally, the hybrid model GJO-GRU|LSTM is proposed, namely an optimized hybrid deep neural network using the Golden Jackal Optimizer (GJO), for predicting stock closing prices. While many studies in the literature focus on tuning the hyperparameters of LSTM or GRU networks for time-series forecasting, the present application attempts to optimize both the parameters and the architecture itself: depending on the case, the network type (GRU or LSTM) and its corresponding parameters are selected dynamically. The performance of the proposed model is compared against sixteen deep-learning models, as well as GA-GRU|LSTM and PSO-GRU|LSTM, which denote optimized models based on Genetic Algorithms (GA) and Particle Swarm Optimization (PSO). The results indicated that GJO-GRU|LSTM delivers more accurate forecasts in the vast majority of cases.

The thesis concludes with the main findings and a summary of its contributions. Its aim is to enrich applied research on Artificial Intelligence and metaheuristic optimization through models capable of supporting well-founded decisions on critical issues in finance. In recent years, these algorithms have been widely adopted across various fields, due to their ability to overcome the limitations of traditional models. We consider that the evaluation of the algorithms investigated, and the proposed approaches make a significant contribution to the field, deepening our understanding of their impact on financial science and enriching the existing literature both within finance and in the broader domains of artificial intelligence and metaheuristic optimization.

Keywords: Artificial intelligence · Machine Learning · Metaheuristic Optimization · Financial Risk Assessment · Investment Decision-Making

1. Motivation, Aims, and Research Questions

Artificial Intelligence (AI) has become an indispensable part of daily life, with worldwide interest and usage rapidly increasing over the years. The applications of AI models have increased among corporations, banks, investors, and practitioners due to their ability to address various difficult tasks that influence decision-making. The exponential growth of data from social media and mobile devices (Borgi *et al.*, 2017), in addition to conventional corporate data, has resulted in an exponential increase in accessible information. McCulloch & Pitts (1943) introduced the core idea of artificial neural networks. The inspiration of these models, as perceived, comes from biological neural networks of the human brain. Although from a theoretical point of view it seems a quite interesting approach, no training method was available until the late 1950s, with the invention of the perceptron network (Rosenblatt, 1958) and the associated learning rule. Furthermore, two years later Widrow & Hoff (1960) introduced the adaptive linear neuron network (ADALINE) and the LMS algorithm as a learning rule. Unfortunately, both networks suffered from the same coalescent limitations, which received extensive publicity by Minsky & Papert (1969). This led to the idea that neural network research had reached a stalemate. It was the backpropagation algorithm (Rumelhart *et al.*, 1986; Werbos, 1990) for training multilayer perceptrons, which has been implemented by various researchers, that was the response to neural networks' criticisms. The reignition of interest in AI algorithms also has been exposed the last decade after some seminal works (Hinton & Salakhutdinov, 2006; Hinton *et al.*, 2006; Krizhevsky *et al.*, 2012). In finance, these models increasingly tackle difficulties in risk management, fraud detection, operational workflows, and investing activities such as portfolio optimization and stock price forecasts;nonetheless, their efficacy depends on meticulous hyperparameter selection.

Metaheuristic algorithms are effective for addressing difficult optimization tasks, since they are able to produce near-optimal, or even optimal solutions. These types of algorithms mimic several natural phenomena, such as evolutionary processes, swarm intelligence, physical laws, and human behaviour (Abdollahzadeh et al., 2024). Four primary reasons contribute to the increasing development of these algorithms (Mirjalili et al., 2014). Metaheuristic algorithms are based on simple concepts, allowing researchers to introduce new or modifying existing ones. Furthermore, metaheuristics can be utilized across many tasks without altering its framework. The researcher must delineate the problem concerning inputs and assess the outputs. Moreover, the majority of metaheuristic algorithms employ derivation-free processes, which are advantageous for cases where derivative information is unknown, or their computation is too complex. Finally, their stochastic characteristics mitigate stagnation in local optima, an important challenge when addressing real-world scenarios.Numerous metaheuristic algorithms exist; in recent years, more than five hundred have been introduced (Rajwar et al., 2023).

In this thesis we try to answer several questions regarding the effectiveness of AI models and metaheuristic optimization algorithms, to be more specific we try to answer the following **Research Questions**:

> **RQ1**: How have artificial intelligence (AI) and metaheuristic optimization algorithms transformed the financial sector, and which challenges can they solve?
>
> **RQ2**: How effective are these methods at addressing financial risk challenges such as customer churn and credit risk?
>
> **RQ3**: How do they support well-informed decisions in areas such as portfolio optimization and stock closing-price prediction?
>
> **RQ4**: To what extent are machine learning (ML) and AI models affected by hyperparameter selection, and how does this impact forecasting quality?
>
> **RQ5**: How does hybridizing these algorithms enhance the forecasting performance of classical AI and ML models?

In this thesis, we are motivated to answer these **RQs** following the increasing interest of research community over the recent years. Several works and thesiss over the years have tried to evaluate the performance of AI models and metaheuristic algorithms. Although its superiority of these algorithms and the several prestigious works that have been published, there still investigation that is needed. Wolpert & McCready (1997) established the No Free Launch (NFL) theorem for optimization, where the authors state that no superior algorithm exists for all problems. Moreover, Wolpert (2002) formulated the NFL theorem for the broader field of machine learning, recognizing the absence of a superior algorithm for all tasks.

This lack of superior algorithms has inspired the research community to develop over the years ne w algorithms and further evaluate the existing ones depending on the task. This fact emerges consistently the research community to consistently evaluate the performance of these types of models in several domains. Regarding neural networks, several researchers in the existent literature perform experimental and comparative analysis for a variety of neural network architectures across several sectors and tasks (Kandhro *et al.*, 2020; Li *et al.*, 2021; Dalli, 2022).In addition regarding metaheuristic algorithms, aside from the algorithm-specific parameters defined by its core architecture, two critical factors influencing the performance of population-based algorithms, are the population size and the number of iterations (epochs). Several studies have evaluated the performance individual algorithms under different configurations (Shi & Eberhart, 1999; Piotrowski, 2017; Piotrowski et al., 2020), whereas others compare multiple ones. Dhal et al., (2019) examined the impact of population size on the performance of three swarm intelligence algorithms. Their study evaluated algorithm behaviour across various function types and dimensions, stating that the proper selection population size has impact on the results.Moreover the authors claim that, the optimal population size depends on the specific algorithm, problem type, and dimensionality.

Agushaka et al. (2023) examined the performance of eleven metaheuristic algorithms for several population sizes and number of epochs on multiple benchmark datasets.The authors conclude that, although not optimal algorithm exists in all cases, keeping enough population diversity and using a large number of epochs is important to achieve to optimal results. Si-Ma et al. (2025) analyzed the performance of 123 swarm intelligence algorithms and the results showed, that to achieve good performance for simple problems are usually needed 30 to 80 epochs, and a higher number of epochs is needed for more complex problems.

Over the recent years, the integration of metaheuristic algorithms with AI and machine learning models has shown promising performance since there are cases that these algorithms can assist classical machine learning models in addressing various challenges. One common challenge is handling high feature dimensionality, and many methods have been proposed over the years. (Chandrashekar & Sahin, 2014; Cai et al., 2018). Metaheuristic algorithms can be used for that purpose (Agrawal et al.,2021; Dokeroglu et al.,2022; Barrera-García et al., 2023; Nssibi et al., 2023). Another issue that a practitioner may face during the employment of a machine learning model is the proper selection of its hyperparameters, which can affect prediction quality (Bischl et al., 2023; Morales-Hernández et al., 2023). Over the last years, metaheuristic algorithms have been widely employed in this regard (Akay et al., 2022; Kaveh & Mesgari, 2023; Morales-Hernández et al., 2023).

Although prior work has examined AI and metaheuristic optimization algorithms in finance, we believe that two gaps remain. First, there is no up-to-date comparative evaluation of recent algorithms across core tasks, such as risk assessment and investment decision-making, including portfolio optimization and stock price forecasting. Second, while hybrid approaches have been proposed, a systematic study that integrates and benchmarks the latest models is still lacking. In general, most studies examine classical, well-established machine-learning models alongside with well-known metaheuristic algorithms, while largely neglecting recent deep-learning models and newer metaheuristics and their performance especially in the financial sector.

In the following sections, at first, we provide bibliometric analyses to better understand the impact of these algorithms on the scientific community and their applications. In the subsequent sections, we present the current literature review to date, the methodology we followed, and the results of the applications. Finally, we conclude with our findings and answer the **RQs** set in this thesis.

2. Literature Review

In the following section, we examine current trends in AI and metaheuristic optimization in finance. We begin with two bibliometric analyses to map the literature. At first, we provide a general overview of applications of these algorithms, and then we present a focused bibliometric analysis of metaheuristic approaches to portfolio optimization. Next, the following subsections present targeted literature reviews.At first on financial risk assessment, then on investment decision-making. Regarding financial risk assessment, we review churn prediction using AI models and evaluate the performance metaheuristic optimized machine learning models for credit-card default prediction. For investment decision-making, we discuss portfolio optimization with metaheuristic algorithms, followed by stock-price prediction using classical AI models and hybrid metaheuristic–AI approaches. Each subsection concludes with the specific gaps identified in that area, and the final subsection synthesizes these into the broader research gaps that motivate our further examination of these algorithms in the financial sector.

2.1 Artificial Intelligence and metaheuristic Optimization in Finance: A bibliometric Review

2.1.1 Introduction

The global financial sector has faced considerable opportunities and problems over time, mainly due to the complex procedures that result from the Big Data age (Fang *et al.*, 2016; Sun *et al.*, 2019; Sun *et al.*, 2020; Goldstein *et al.*, 2021). Individuals, businesses, and financial institutions have access to a plethora of information, which enables them to optimize their operations and decision-making, even avoid potential hazards, thus can establish long-term sustainability and increase profitability. This data enables the avoidance of numerous real-world problems. Institutions are able to use this data to manage severas challenges such as financial risk (Cerchiello & Giudici, 2016; Kang, 2019; Du *et al.*, 2021), bankruptcy prediction (Jo & Shin, 2016), detect fraudulent transactions(Hipgrave, 2013; Vaughan, 2020),and improve the quality of service(Raguseo & Vitari, 2018; Giebe *et al.*, 2019).

For consumers, the rise of big data has offered substantial chances for growing personal finances, improve investing decisions, and minimize financial risks. (Farboodi & Veldkamp, 2023). Furthermore, individuals can improve their investments by employing technologies that use data to provide insights on market pattern. Numerous factors are responsible for the explosion of data in the financial sector over the last decade. The integration of technologies such as online banking, digital wallets, and mobile applications has transformed the financial sector, allowing millions of individuals to access banking services instantly (Durai & Stella, 2019), benefiting both banks and consumers (Acharya *et al.*, 2008; Hernández-Murillo *et al.*, 2010; Szopiński, 2016). This transition has resulted in an increase of data, as each transaction is recorded, which allows financial firms to gather more data than before.

Over the last decade, social media have experienced significant global popularity. Organizations and individuals, significantly influence market dynamics and investment strategies through social media. Companies can benefit from social media in numerous ways, such as using it for advertising, acquiring new clients, improving operations, and maintaining strong relationships with existing customers (Kumar & Devi, 2014; Dootson et al., 2016; Elena, 2016; Cao & Weerawardena, 2023; Bruce et al., 2023; Masciandaro et al., 2024). Social media users are able to share opinions and real-time data regarding financial markets. The users' insights and preferences, provide important information that may be used to evaluate market sentiment and forecast stock fluctuations using sentiment analysis methodologies, which can impact the global financial markets (Chen et al., 2023; Rodríguez-Ibánez et al., 2023; Ren et al., 2024).

Furthermore, due to the environmental degradation which has been observed in recent years businesses have adjusted their operational strategies to prioritize sustainable finance(Cuhna *et al.*, 2021; Edmans & Kacperczyk, 2022; Tao *et al.*, 2022). Governments and regulatory authorities are adopting laws that encourage sustainable finance. The Environmental Social Governance (ESG) score has had a beneficial impact on society (Chen *et al.*, 2023). Investors incorporate ESG criteria into their investment decisions (Eccles *et al.*, 2017), potentially generating financial advantages while also benefiting society and the environment, as companies improve their ESG scores (Lisin *et al.*, 2022; Wong *et al.*, 2022). In recent decades, AI algorithms have been widely used across many fields to tackle numerous problems. These algorithms are able to provide better solutions in cases when traditional methods fall short. The curse of dimensionality (Bellman, 1957) may limit the performance of some statistical methods. In contrast to classical statistical methods, which often rely on strict assumptions, AI models make fewer strict assumptions about data characteristics (Mansini et al., 2023). Moreover, the explosion of data availability over the last years can pose difficulties for traditional models, since they are not designed to handle enormous datasets (Al Jarrah et al., 2015; Cerqueira et al., 2019; Kontopoulou et al., 2023). In such scenarios, which exist in real-world applications, AI models can be beneficial. Moreover AI models,can achieve high performance in complex problems such as image recognition, natural language processing, and speech recognition (Jiao & Zhao, 2019, Wang et al., 2021, Iqbal et al., 2023). In real-life datasets, unstructured data of various types may exist, including numerical or non-numerical, as well as categorical and non-categorical. This has prompted professionals to use these models (Kaya et al., 2019; Bonsón et al., 2021; Anantrasirichai & Bull, 2022; Fares et al., 2022; Doumpos et al., 2023).

This study seeks to address key questions in light of the rapid advancement of metaheuristic algorithms in recent years, examining their impact on the worldwide landscape of AI applications in banking and determining whether any patterns are evident, while also considering how the big-data era has reshaped the financial sector, we seek to identify current developments and suggest avenues for future research.

Numerous well-regarded studies have reviewed either AI models or metaheuristic algorithms across various sectors, including finance (Ahmed et al., 2022; Bahoo et al., 2023). Recent studies have investigated the integration of these two approaches. However, as far as we know, there are few studies that focus especially on the combined use of machine learning and metaheuristic algorithms. Szénási & Légrádi (2024) performed a systematic review on the hybridization of these methods, pointing out their growing use in numerous areas. Paz et al. (2025) present a thorough literature analysis on credit risk assessment, demonstrating that hybrid approaches have become widely adopted. The following subsections provide a bibliometric analysis of the hybridization of these algorithms specifically in the financial sector. We provide a comprehensive analysis of research from recent decades, highlighting the growing adoption of this hybridization of these methodologies

2.1.2 Dataset and methodology

The Scopus dataset has been utilized for data collecting, as it is among the most widely used research-literature databases, covering a broad array of peer-reviewed journals across multiple publishers. The respective database has been employed in several bibliometric analyses in the existing literature (Goodell *et al.*,2021; Ahmed *et al.*,2022; Kriouich & Sarir,2024; Utama *et al.*, 2025; Singh & Dwivedi,2025)

Regarding the keywords associated with AI models, we followed an expanded version of Ahmed *et al.* (2022) query. For metaheuristic algorithms, we included the keyword "metaheuristic(s)" in the query, along with a variety of algorithms provided in the MEALPY Python's package (Van Thieu & Mirjalili, 2023), and for finance-related topics , we selected a wide range of keywords. The complete query is presented in Appendix A. Since, the application of AI and metaheuristic algorithms in financial topic, is studied by scholars from various sectors except of the Subject Areas of *Business, Management and Accounting* and *Economics, Econometrics and Finance*, we also include work from the Subject Areas of *Computer Science, Decision Sciences, and Mathematics*.

The query was conducted using the Title, Abstract, and Keywords fields, including both English research articles and review papers, including works published up to the year 2024. In total 1,729 documents were retrieved and following the screening process, 446 articles were discarded, resulting in a final sample of 1,283 studies. Due to the involvement of several subject areas in our query, a more comprehensive review was required, since cases existed where the documents included unrelated topics such as "*blood banks*" or "*cutting stock problem*." Moreover, for the analysis, we included only studies that validated their methods on benchmark or real-world datasets, with at least one dataset involving financial data. We excluded works that discussed ML challenges (e.g., class imbalance or feature selection) but referenced finance only in the abstract without actually testing on financial datasets.

Regarding the bibliometric analysis, we employed various widely used tools in the literature to obtain significant insights from our dataset. The Bibliometrix package (Aria & Cuccurullo, 2017) used as a primary tool for our analysis, since this package can provide valuable information regarding the documents. Moreover, the VOSviewer tool(Van Eck *et al.*,2013) was employed for visualization purposes as well as Python's Matplotlib library (Hunter, 2007) was used in particular scenarios. For the visualization via VOSviewer, the default *Association Strength* (Van Eck & Waltman,2007 ; Van Eck & Waltman,2010) normalization used as the similarity measure between nodes, while clusters were detected using the VOS clustering technique optimized via the Smart Local Moving (SLM) algorithm (Waltman & Van Eck,2013). Although each table is intended to show the top 25, we expand the list when ties occur to keep the reporting fair and complete

2.1.3 General overview of the dataset

The dataset (Table 1) spans a 30-year period from 1994 to 2024, which includes 1,283 documents from 426 sources. The results show an annual growth rate of 16.16%, which reveals increasing research activity in the field. On average, documents in this dataset are 6.96 years old and receive 33.99 citations per document, indicating substantial scholarly impact. The dataset includes a total of 48,148 references, 5,200 Keywords Plus (ID), and 3,029 Author's Keywords (DE). In total 3,085 authors have contributed, with 166 documents being single-authored and 136 individuals responsible for those works. Collaborative research is evident, with an average of 3 co-authors per document and international co-authorship accounting for 20.5% of the contributions. Figure 1 shows an increasing trend in the number of published documents from. Activity remained quite low and stable until about 2008; productivity increased from 2010 onward and accelerated markedly after 2018. Publication peaks are noted in 2022 (169 documents) and 2024 (179 documents).

Table 1. Dataset's main information

Description	Results
Timespan	1994:2024
Sources	421
Documents	1283
Annual Growth Rate %	16.16
Document Average Age	6.96
Average citations per doc	33.99
References	48148
Keywords Plus (ID)	5200
Author's Keywords (DE)	3029
Authors	3085
Authors of single-authored docs	135
Single-authored docs	166
Co-Authors per Doc	3
International co-authorships %	20.5

Note. *Keywords plus (ID)* are referring to keywords, which have been generated based on title and cited reference.

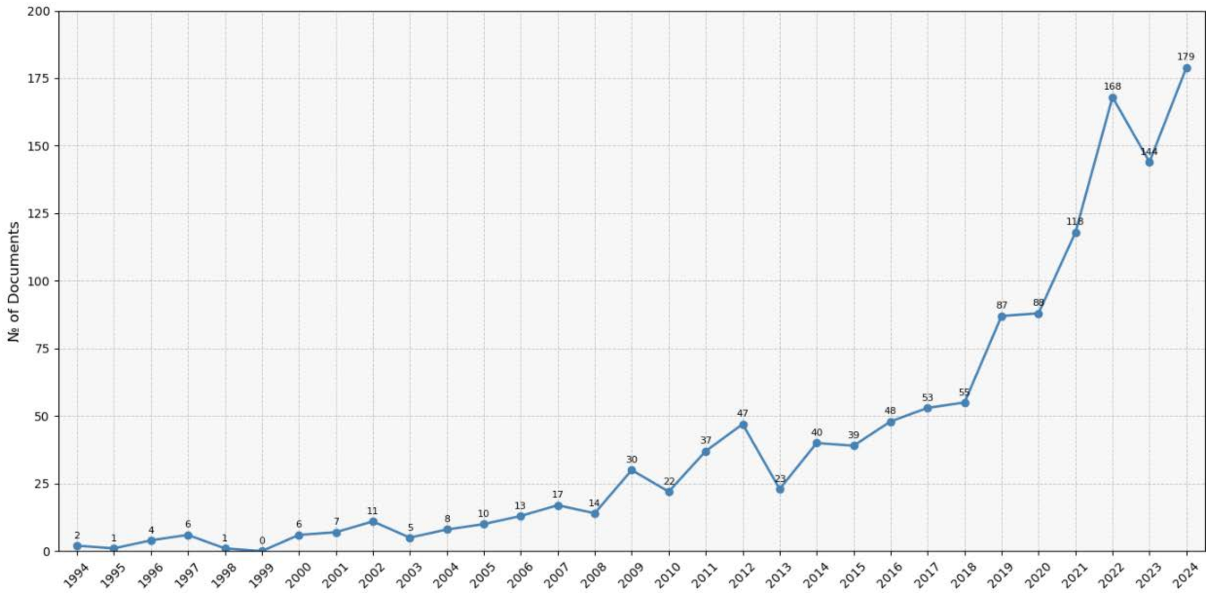


Figure 1. Annual Scientific Production

2.1.4 Sources and Authors analysis

Table 2 shows the most active journals contributing to the field, with Expert Systems with Applications (ESWA) leading significantly at in total of 102 publications. This is followed by Applied Soft Computing (65 articles) and IEEE Access (45), both prominent outlets for intelligent systems and computational techniques. We are aligned with the work of Goodell *et al.* (2021), which highlighted the relatively lower activity of AI research in finance journals, since we observe a similar pattern. Our analysis indicate that journals focused on computational methods and applications tend to be more active in publishing metaheuristic and AI-related research compared to financial journals. Figure 2 illustrates the bibliographic coupling among journals, where ESWA emerges as the most connected node, reinforcing its important role in integrating research across multiple computational techniques. Another interesting finding is the number of works that have been published at Physica A: Statistical Mechanics and Its Applications, where various applications are being discussed regarding finance. Given the growing trend in the field of Econophysics where the respective journal is leading (Rodrigues *et al.*, 2024), this is not surprising. The articles from the respective journal are listed in Appendix A.

Table 2. Most relevant sources

Sources	Articles
Expert Systems with Applications	102
Applied Soft Computing	65
IEEE Access	45
Soft Computing	29
Neural Computing and Applications	27
Computational Economics	25
International Journal of Advanced Computer Science and Applications	23
Computational Intelligence and Neuroscience	21
Information Sciences	21
Neurocomputing	20
Knowledge-Based Systems	16
Physica A: Statistical Mechanics and Its Applications	15
Journal Of Intelligent and Fuzzy Systems	14
Mathematics	12
Mathematical Problems in Engineering	11
Applied Intelligence	10
Journal Of Forecasting	9
Kybernetes	9
Wireless Communications and Mobile Computing	9
Applied Mathematics and Nonlinear Sciences	8
Applied Sciences (Switzerland)	8
European Journal of Operational Research	8
Expert Systems	8
IEEE Transactions on Evolutionary Computation	8
International Journal of Computational Intelligence Systems	8
Journal Of Organizational and End User Computing	8
Journal Of Theoretical and Applied Information Technology	8
PEERJ Computer Science	8

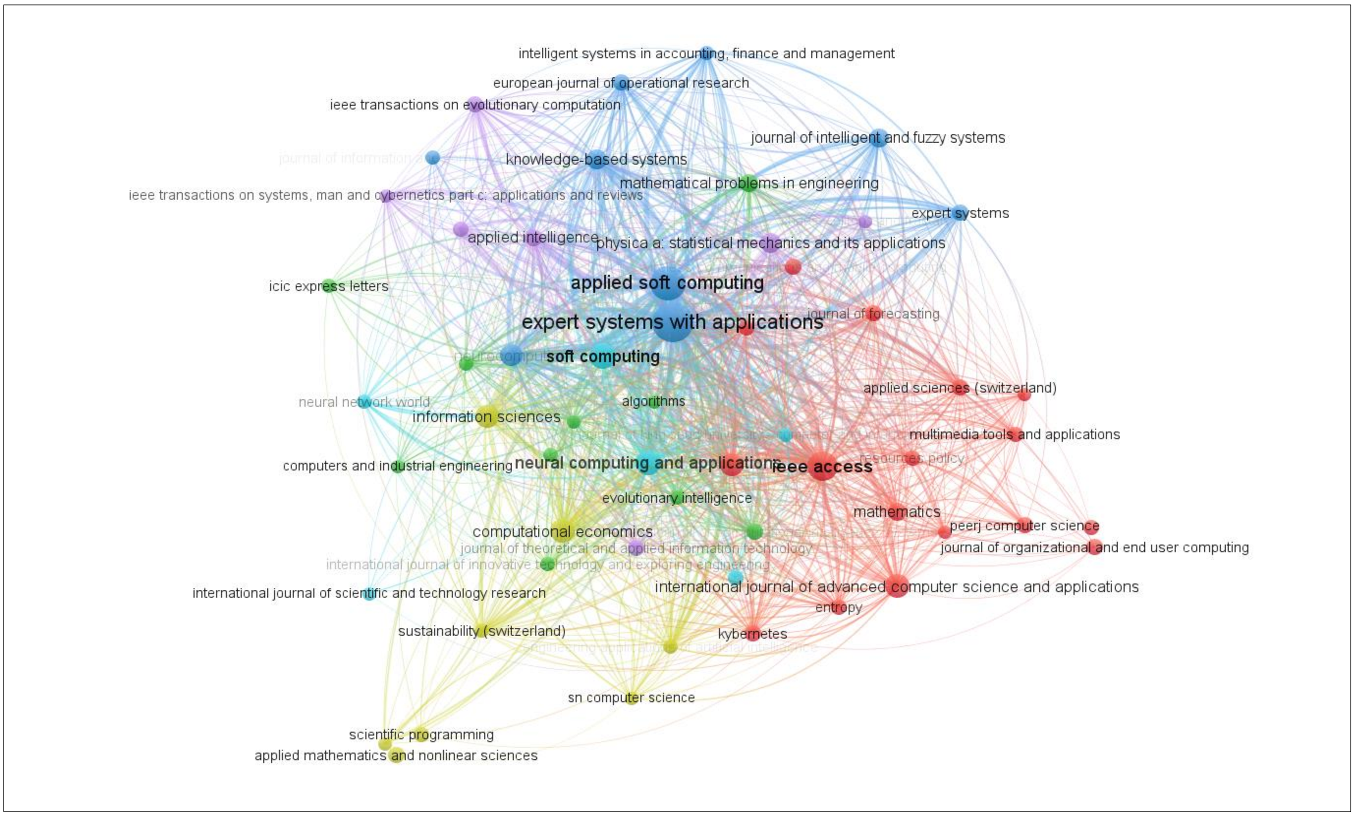


Figure 2. Sources Bibliographic Coupling. **Note.** The sources' bibliographic coupling network map illustrates the relationships among various academic journals based on shared references

In terms of authors production, Erol Egrioglu leads with 18 publications, followed by Pradipta Kishore Dash and Sarat Chandra Nayak with 16 each. Notably, Salim Lahmiri stands out with the highest fractionalized score (8.83), suggesting substantial contributions to fewer co-authored paper (Table 3). Figure 3 adds depth to this analysis by showcasing the publication trends of the top authors over time. It clearly highlights how key contributors like Egrioglu, Dash, and Nayak have consistently published, with a noticeable surge in output, especially after 2018. The figure also highlights emerging contributors whose activity has surged in recent years, indicating evolving research momentum within the field. The size and density of the bubbles emphasize both consistency and impact, with several authors showing sharp increases in productivity in the last five years. In Figure 4. the Lotka's Law (1926) describes the distribution of authorship. The law is formulated as following x^n y=c where x refers to number of articles and y to the propotion of authors (Aytac *et al.*,2025). Figure 4 shows a highly uneven distribution – most authors contribute just one or two papers, while only a few are exceptionally prolific. This distribution is aligned with Lotka's inverse square law, indicating a core group of highly active researchers amidst a larger pool of occasional contributors. However, considering the recent surge in interest in metaheuristic algorithms, particularly in conjunction with AI applications the concentration of publications in the last decade further reinforces the applicability of Lotka's law in this context. Finally, Figure 5 displays the bibliographic coupling among the authors, where five clusters are observed.

Table 3 Most Relevant Authors

Authors	Articles	Articles Frac.
Egrioglu, Erol	18	6.65
Dash, P.K.	16	6.00
Nayak, Sarat Chandra	16	7.67
Bas, Eren	15	5.07
Bisoi, Ranjeeta	10	3.50
Lahmiri, Salim	10	8.83
Ravi, V.	10	3.75
Chen, Mu-Yen	9	4.70
Dash, Rajashree	9	4.42
Deng, Shangkun	9	1.65
Kim, Kyoung-Jae	8	5.00
Wang, Jujie	8	2.78
Chen, Huiling	7	1.09
Han, Ingoo	7	3.00
Karathanasopoulos, Andreas	7	2.32
Quek, Chai	7	2.04
Zhang, Wenyu	7	1.77
Chiu, Deng-Yiv	6	2.83
Enke, David	6	2.58
Hu, Yi-Chung	6	4.25
Melin, Patricia	6	1.95
Sermpinis, Georgios	6	1.53
Shin, Kyung-Shik	6	2.83
Wang, Jianzhou	6	1.32
Yolcu, Ufuk	6	1.78
Zhu, Yingke	6	1.06

Note. *Articles Frac.*: Articles Fractionalized

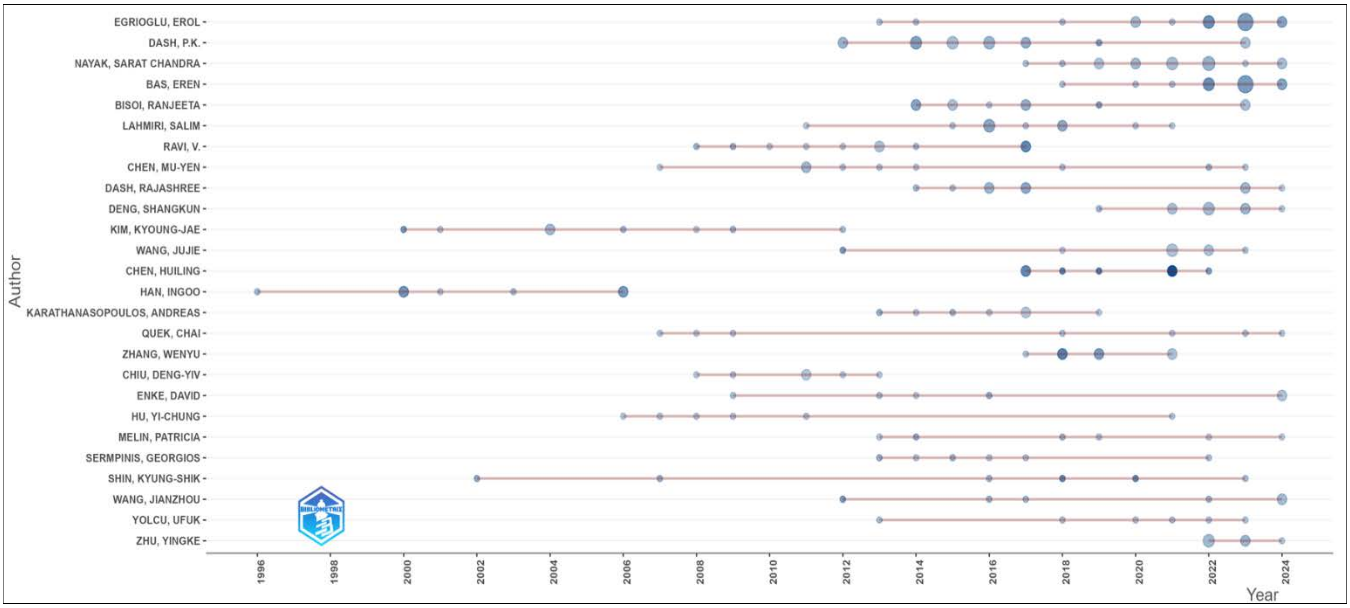


Figure 3. Authors' Production over Time

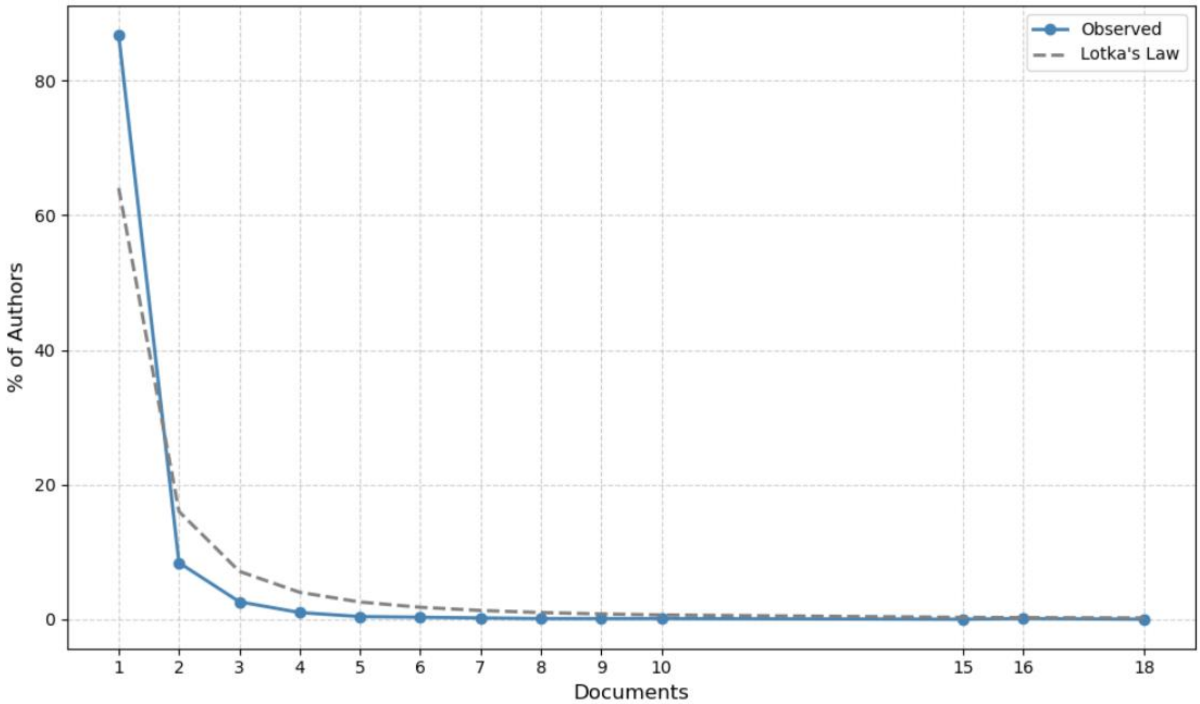


Figure 4. Lotka’s Law

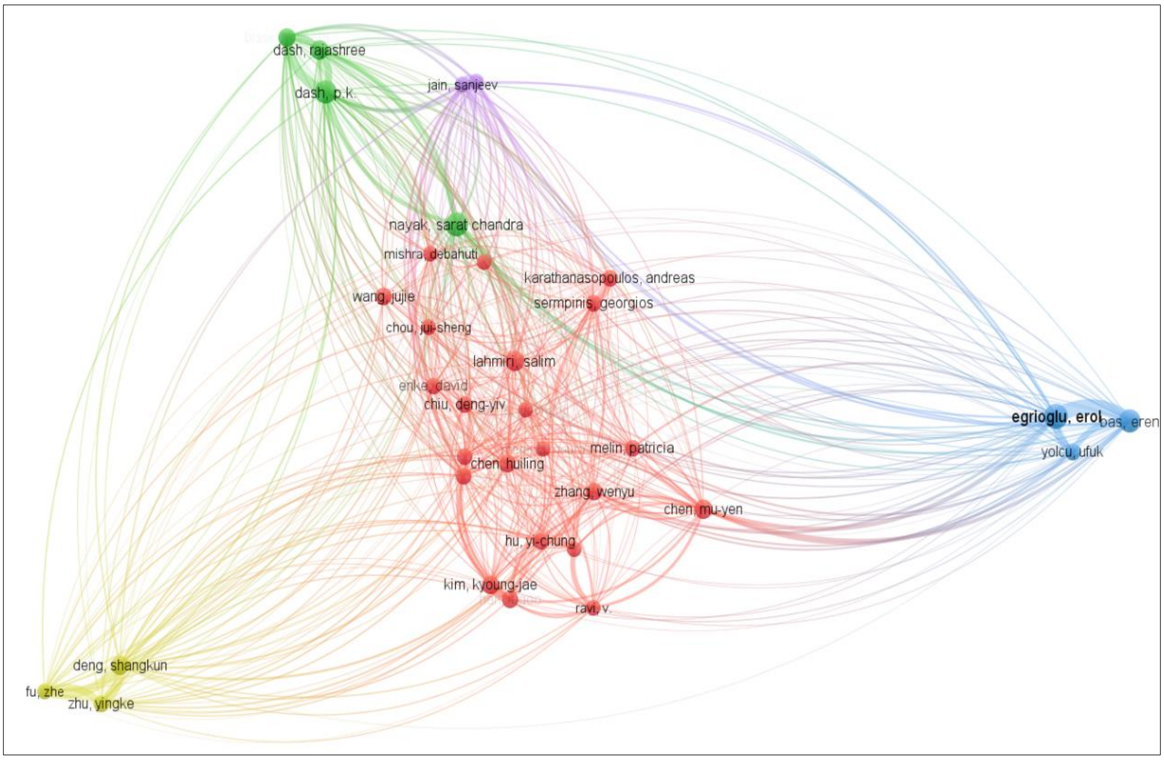


Figure 5. Authors Bibliographic Coupling. **Note:** The authors' bibliographic coupling network map illustrates the relationships among individual researchers based on shared references in their publications.

2.1.5 Affiliation and countries analysis

The analysis of institutional affiliations (Table 4) shows a strong performance by Asian affiliations, especially from China, which contributes the largest number of institutions and article counts—led by China Three Gorges University (39), Zhejiang University of Finance and Economics (37), and several others with high productivity. Iran and Turkey also show high research activity, with Islamic Azad University (49 articles) and Giresun University (43) leading the overall list. In addition, institutions from Taiwan, Singapore, and South Korea also yield consistent performance, indicating a broad regional engagement with AI and metaheuristics in finance. European and American representation is limited, with only a few entries such as Singidunum University (Serbia) and Tijuana Institute of Technology (Mexico).

Table 5 and Figure 6 provide the outcomes regarding the corresponding authors countries. China is leading in this field, accounting for 33.7% of all publications (447 papers), with the vast majority being created through single-country publications (SCP), indicating a primarily internal research ecosystem. India and Iran follow with 151 and 63 publications, respectively, albeit with somewhat lower rates of international collaboration. Notably, Malaysia (56.5%), Italy (55.6%), and the United Arab Emirates

(71.4%) have high levels of multi-country collaboration (MCP), indicating considerable international participation despite lower overall productivity. In contrast, countries such as Indonesia, Greece, and Croatia exhibit no international collaboration, indicating a more regional research approach. Overall, while Asia leads in volume, the most collaborative efforts are coming from smaller or middle-tier contributing countries, emphasizing the differences in research strategy throughout regions.

Table 6 presents the total citations (TC) and average article citations by country. China has the most total citations (15,081), although its average citations per article (33.7) are very moderate in comparison to other Western nations. Significantly, although producing fewer papers, Norway (131.5), the Netherlands (135.0), and Croatia (109.8) show exceptionally high average citations, suggesting that their publications are extremely influential or extensively referenced benchmark research. In contrast, nations like Malaysia (15.9) and Saudi Arabia (9.7) exhibit lower citation averages, suggesting that although research is being conducted, its global impact or influence may be limited. The United States (62.4) and the United Kingdom (49.7) exhibit elevated average citation counts, underscoring their prominence as substantial contributors. Figure 7 depicts the bibliographic coupling network among nations, demonstrating the intensity of co-citation relationships derived from common references.

Table 4. Most Relevant Affiliations

Affiliation	Country	Articles
Islamic Azad University	Iran	49
Giresun University	Turkey	43
China Three Gorges University	China	39
Zhejiang University of Finance and Economics	China	37
Wenzhou University	China	24
Nanjing University of Information Science and Technology	China	22
Singidunum University	Serbia	20
Amirkabir University of Technology	Iran	19
Henan Polytechnic University	China	19
Nanyang Technological University	Singapore	19
National Central University	Taiwan	19
Wuhan University of Technology	China	19
National Kaohsiung University of Science and Technology	Taiwan	16
South China University of Technology	China	16
Universiti Sains Malaysia	Malaysia	16
Xidian University	China	16
Dongbei University of Finance and Economics	China	15
National Taiwan University of Science and Technology	Taiwan	15

Notreported	N/A	15
University Of Tehran	Iran	15
Yuan Ze University	Taiwan	15
Lanzhou University	China	14
National University of Singapore	Singapore	13
Tijuana Institute of Technology	Mexico	13
Yonsei University	South Korea	13

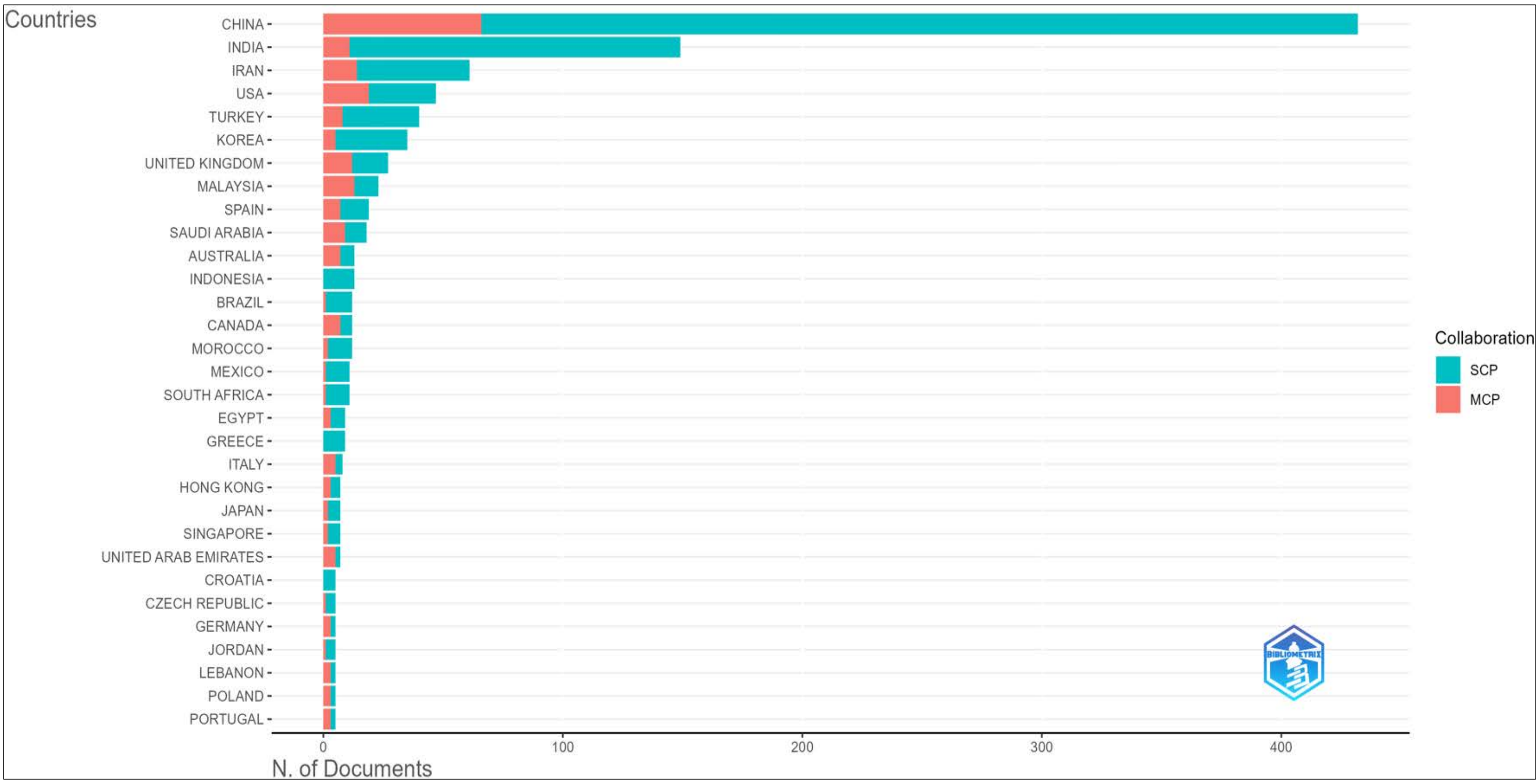


Figure 6. Corresponding Author's Countries**. Note:** *SCP*: Single country publications; *MCP*: Multiple country publications

Table 5. Corresponding Author's Countries

Country	Docs	Docs%	SCP	MCP	MCP %
China	432	33.7	366	66	15.3
India	149	11.6	138	11	7.4
Iran	61	4.8	47	14	23
Usa	47	3.7	28	19	40.4
Turkey	40	3.1	32	8	20
Korea	35	2.7	30	5	14.3
United Kingdom	27	2.1	15	12	44.4
Malaysia	23	1.8	10	13	56.5
Spain	19	1.5	12	7	36.8
Saudi Arabia	18	1.4	9	9	50
Australia	13	1	6	7	53.8
Indonesia	13	1	13	0	0
Brazil	12	0.9	11	1	8.3
Canada	12	0.9	5	7	58.3
Morocco	12	0.9	10	2	16.7
Mexico	11	0.9	10	1	9.1
South Africa	11	0.9	10	1	9.1
Egypt	9	0.7	6	3	33.3
Greece	9	0.7	9	0	0
Italy	8	0.6	3	5	62.5
Hong Kong	7	0.5	4	3	42.9
Japan	7	0.5	5	2	28.6
Singapore	7	0.5	5	2	28.6
United Arab Emirates	7	0.5	2	5	71.4
Croatia	5	0.4	5	0	0
Czech Republic	5	0.4	4	1	20
Germany	5	0.4	2	3	60
Jordan	5	0.4	4	1	20
Lebanon	5	0.4	2	3	60
Poland	5	0.4	2	3	60
Portugal	5	0.4	2	3	60

Note. *SCP*: Single country publications; *MCP*: Multiple country publications

Table 6. Citations by country

Country	TC	Av.TC
China	14726	34.1
India	3753	25.2
Usa	3009	64
Korea	2880	82.3
Iran	1728	28.3
United Kingdom	1342	49.7
Turkey	1269	31.7
Australia	1077	82.8
Spain	901	47.4
Croatia	549	109.8
Canada	498	41.5
Italy	447	55.9
Poland	405	81
Germany	381	76.2
Greece	371	41.2
Malaysia	365	15.9
Portugal	355	71
Singapore	297	42.4
Morocco	290	24.2
Hong Kong	287	41
Japan	284	40.6
Mexico	275	25
South Africa	272	24.7
Brazil	257	21.4
Saudi Arabia	174	9.7

Note. *TC*: Total citations,
Av.TC: Average Total Citations

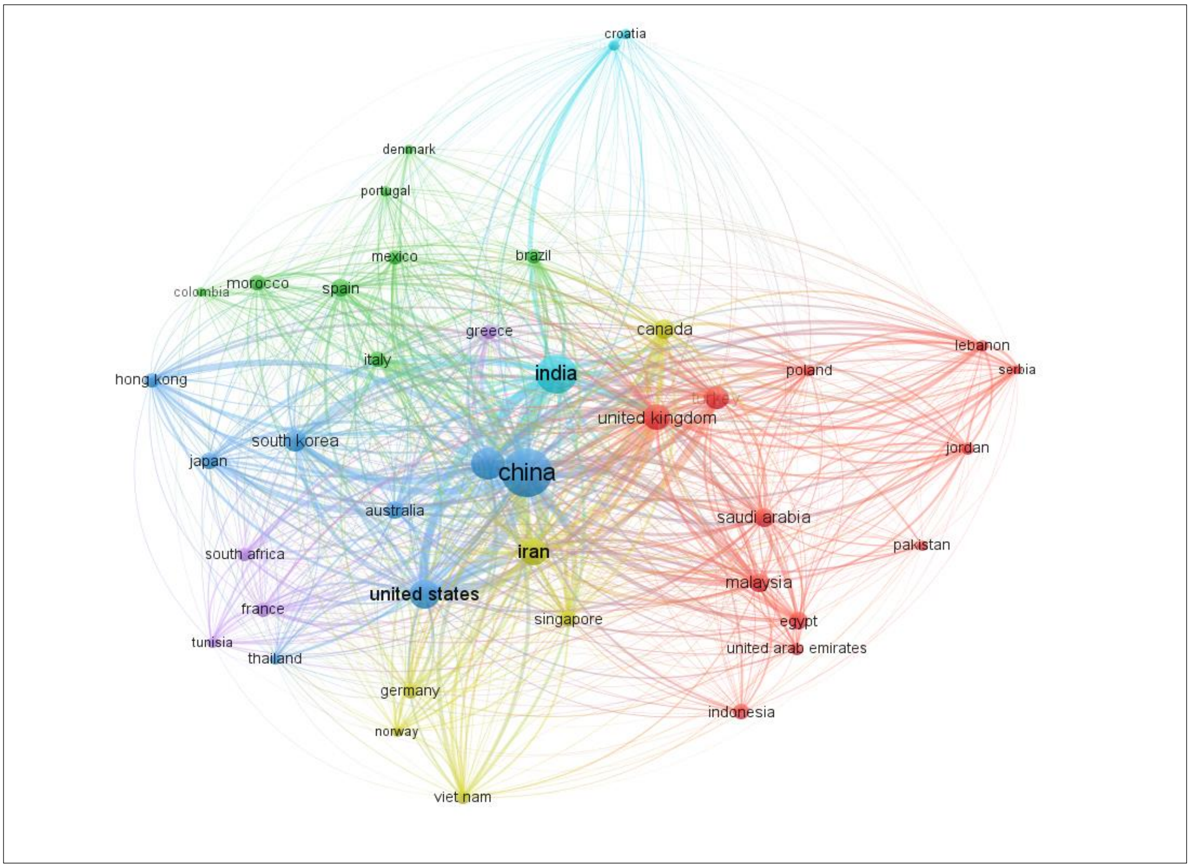


Figure 7. Countries' Bibliographic Coupling. **Note.** Countries' bibliographic coupling network map illustrates the relationships between countries based on shared references in academic publications

2.1.6 Documents and Keywords Analysis

Table 7 lists the most frequently cited documents worldwide. The top cited document is the work of Pan (2012), in which the author proposed the Fruit Fly Optimization Algorithm (FOA), inspired by the foraging behaviour of fruit flies, to optimize a General Regression Neural Network (GRNN) for the task of financial distress prediction. The proposed FOAGRNN outperformed both GRNN and Multiple Regression (MR) models in terms of AUC and Gini scores.

Huang et al. (2007) proposed an optimized Support Vector Machine (SVM) via Genetic Algorithms (GA) to address the challenges of credit rating. The GA was used not only to tune the hyperparameters of an SVM model but also to reduce feature complexity. The work of Kurani et al. (2023) ranks first, in both Total Citations per Year (TCpY) and Normalized Total Citations (NTC). The authors present an analysis of SVM and ANN applications for the task of accurate stock forecasting. In addition, the authors highlight the growing importance of metaheuristic algorithms for improving predictive performance.

Table 7. Most Global Cited Documents

Paper	DOI	TC	TCpY	NTC
Pan W-T, 2012, Knowl Based Syst	10.1016/j.knosys.2011.07.001	1562	111.57	20.95
Huang C-L, 2007, Expert Sys Appl	10.1016/j.eswa.2006.07.007	655	34.47	5.75
Kim K-J, 2000, Expert Sys Appl	10.1016/S0957-4174(00)00027-0	572	22.00	3.80
Espejo Pg, 2010, Ieee Trans Syst Man Cybern Pt C Appl Rev	10.1109/TSMCC.2009.2033566	491	30.69	6.43
Kazem A, 2013, Appl Soft Comput J	10.1016/j.asoc.2012.09.024	409	31.46	8.12
Rather Am, 2015, Expert Sys Appl	10.1016/j.eswa.2014.12.003	388	35.27	7.33
Shen W, 2011, Knowl Based Syst	10.1016/j.knosys.2010.11.001	381	25.40	9.93
Oreski S, 2014, Expert Sys Appl	10.1016/j.eswa.2013.09.004	357	29.75	7.40
Kurani A, 2023, Ann Data Sci	10.1007/s40745-021-00344-x	352	117.33	25.31
Wu C-H, 2007, Expert Sys Appl	10.1016/j.eswa.2005.12.008	318	16.74	2.79
Altan A, 2019, Chaos Solitons Fractals	10.1016/j.chaos.2019.07.011	316	45.14	7.93
Min S-H, 2006, Expert Sys Appl	10.1016/j.eswa.2005.09.070	315	15.75	4.82
Tsai C-F, 2010, Decis Support Syst	10.1016/j.dss.2010.08.028	307	19.19	4.02
Shin K-S, 2002, Expert Sys Appl	10.1016/S0957-4174(02)00051-9	302	12.58	2.74
Ghoddusi H, 2019, Energy Econ	10.1016/j.eneco.2019.05.006	299	42.71	7.50
Wang J-J, 2012, Omega	10.1016/j.omega.2011.07.008	279	19.93	3.74
Ong C-S, 2005, Expert Sys Appl	10.1016/j.eswa.2005.01.003	276	13.14	3.36
Göçken M, 2016, Expert Sys Appl	10.1016/j.eswa.2015.09.029	270	27.00	6.17
Hassan Mdr, 2007, Expert Sys Appl	10.1016/j.eswa.2006.04.007	269	14.16	2.36
Kuo Rj, 2001, Fuzzy Sets Syst	10.1016/S0165-0114(98)00399-6	251	10.04	3.91
Jadhav S, 2018, Appl Soft Comput J	10.1016/j.asoc.2018.04.033	250	31.25	4.63
Luo J, 2018, Appl Math Model	10.1016/j.apm.2018.07.044	248	31.00	4.60
Armano G, 2005, Inf Sci	10.1016/j.ins.2003.03.023	242	11.52	2.94
Chung H, 2018, Sustainability	10.3390/su10103765	234	29.25	4.34
Wang Y-F, 2002, Expert Sys Appl	10.1016/S0957-4174(01)00047-1	233	9.71	2.11

Note. The table displays the corresponding author of the top-cited documents, along with the journal and its respective DOI**.** *TC*: Total Citations**;** *TCpY* **:** Total Citations per Year**;** *NTC***:** normalized Total Citations

Next, we conducted a Keywords analysis to identify current patterns and provide useful insights. Table 8 provides the most common author keywords. We find that Genetic Algorithms is the most popular term. Furthermore, the PSO and Differential evolution algorithms tend to be more common, which is not surprising given that they are among the earliest and most established algorithms. Next, we see multiple terms related to machine learning models, as well as keywords that describe various financial sector challenges such as stock market, bankruptcy prediction, credit risk, financial distress, and fraud detection. Figure 8 depicts the co-occurrence of authors' terms, demonstrating the range of metaheuristic and machine learning techniques, as well as financial activities. Figure 9 displays the trend subjects, where the early years (2005-2012) were dominated by core methods including logistic regression, decision trees, and genetic algorithms, indicating a preference for classic statistical and optimization approaches. As time passes, more advanced machine learning techniques such as neural networks, deep learning, and (Long Short-Term Memory) LSTM model gain prominence, particularly after 2015, showing a shift toward more sophisticated predictive modelling. Simultaneously, nature-inspired algorithms such as particle swarm optimization, the firefly algorithm, and the more recent sparrow search algorithm are becoming increasingly relevant. In addition, based on the keywords analysis, we observed a high interest in feature selection, which is a common challenge in the real-world datasets, where the number of features may add complexity during predictive modelling.

Table 8. Most Frequent Author Keywords

Keyword	Occurrences
Genetic Algorithms	339
Neural Networks	217
Particle Swarm Optimization	183
Stock Market	168
Forecasting	116
Machine Learning	106
Support Vector Machine	87
Deep Learning	75
Feature Selection	64
Artificial Intelligence	59
Bankruptcy Prediction	57
LSTM	55
Portfolio Optimization	52
Data Mining	44
Classification	43
Credit Scoring	42
Optimization	41
Metaheuristics	37
Support Vector Regression	35
Differential Evolution	27
Prediction	24
Credit Risk	21
Ensemble Learning	21
Evolutionary Computation	20
Financial Distress	20
Fraud Detection	20
Random Forest	20

Note: Keyword synonyms have been merged.

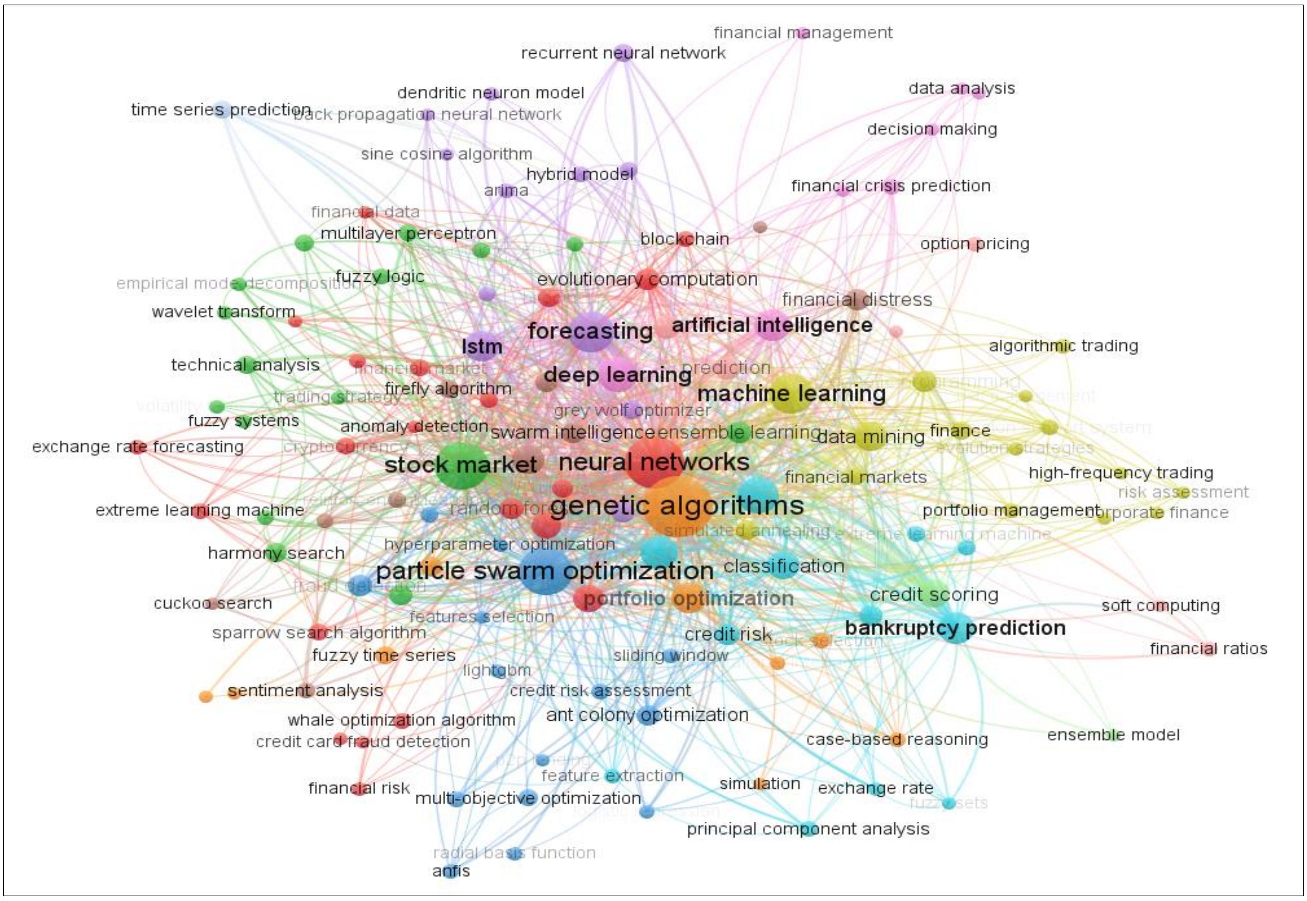


Figure 8. Author's keywords co-occurrence network. **Note.** Keyword synonyms have been merged.

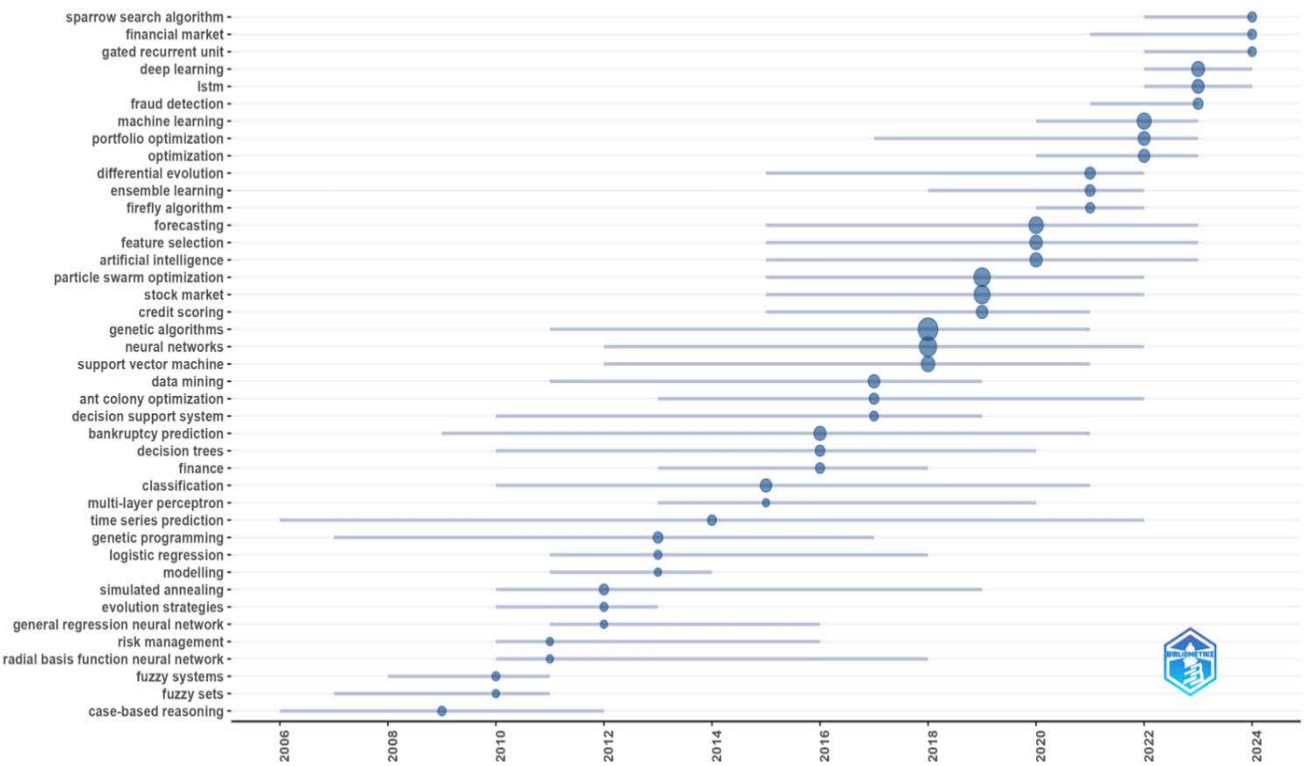


Figure 9. Trend Topics. **Note:** Keyword synonyms have been merged.

2.1.7 General Overview of the results

In general, the results demonstrate a growing use of AI and metaheuristic algorithms to address financial problems over the years. Researchers increasingly apply metaheuristic methods not only for classical optimization tasks but also to enhance the performance of AI models. New metaheuristic algorithms are being developed with specific applications in machine learning, as shown in the case of the widely cited work of Pan (2012), who proposed the FA to optimize machine learning models. However, as our findings suggest, this hybridization remains a relatively new area, with only a limited number of authors consistently contributing to the field. While Asian countries and institutions are highly productive, interest in the field is expanding worldwide.

In the era of big data, the financial sector faces new and difficult challenges, pushing both researchers and practitioners to explore more advanced AI techniques and metaheuristic algorithms. Our analysis shows that numerous challenges of the financial sector, including credit risk and market forecasting, have received increasing attention. Moreover, as financial technologies evolve, challenges such as bankruptcy prediction, financial distress, and fraud detection have become central topics of discussion. In light of these developments, financial institutions, banks, and investors should take hybrid models into account, as the existing literature demonstrates their effectiveness in addressing these critical issues.

Regarding the research question posed in this study, we observed a clear upward trend in the use of metaheuristic algorithms to enhance the performance of AI models in the financial sector, particularly over the last decade. This observation aligns with prior reviews (Szénási & Légrádi, 2024; Paz et al., 2025), which report similar findings. Journals focused on computer science lead in publishing works on this topic, as earlier studies also found (Goodell et al., 2021). Additionally, we observe strong interest in the task of feature selection, which frequently addressed with metaheuristic algorithms. There are numerous works that discuss the application of metaheuristic algorithms to this regard (Agrawal et al., 2021; Dokeroglu et al., 2022; Barrera-García et al., 2023; Nssibi et al., 2023). In conclusion, although AI models show impressive performance, the integration of optimization algorithms, and especially metaheuristics, can improve the performance of AI models. Their ability to efficiently identify near-optimal solutions has increasingly captured the attention of the academic community.

The findings indicate a growing interest among researchers in this area over the years. While numerous researchers have contributed to this growing field, only a relatively small group appears to be consistently active in related research, probably because this hybrid approach is relatively new. In terms of institutional and geographical production, the analysis suggests that Asian affiliations and countries have shown active contribution. However, several countries contribute to this field, highlighting the worldwide interest in this topic. Although this analysis is focused on the financial sector, we observe that many leading documents are published in computer science journals. The analysis of keywords and documents shows a strong focus on various financial topics, including stock market prediction, credit risk, bankruptcy forecasting, and fraud detection. In recent years, researchers have shown a growing preference for more advanced AI models and modern metaheuristic algorithms to tackle these challenges.

Overall, integrating metaheuristic methods to improve AI performance appears promising, especially for tasks such as hyperparameter tuning and feature selection. This study is not without limitations. Although an extensive search query was used, some relevant keywords may have been unintentionally omitted. Furthermore, the analysis was based only on documents indexed in the Scopus database. While Scopus is a respected and widely used resource, including other databases and expanding the keyword query would improve overall coverage.

2.2 Metaheuristic algorithms for portfolio optimization: A bibliometric Review

2.2.1 Introduction

In the following subsection we provide a bibliometric analysis regarding the application of metaheuristics algorithms for the task of portfolio optimization which has been extensively examined in the existing literature. Several works (Doering *et al.*,2019; Erwin & Engelberchart,2023) provide valuable insights into the utilization of metaheuristic algorithms to this regard. In addition, Metaxiotis & Liagkouras (2012) provide a literature review of multiobjective evolutionary algorithms in this regard. Since Pritchard (1969) established bibliometric analysis, numerous studies have examined the evolution of diverse research fields, providing useful information for professionals and scholars. To our knowledge, despite several studies investigating the application of metaheuristic algorithms in portfolio optimization, there is no recent study that systematically highlights the newest advancements. This subsection seeks to address that gap by providing a bibliometric overview of recent advancements in portfolio optimization through metaheuristic algorithms. we seek to identify key authors, sources, organizations, and areas within the academic field, as well as evolving methodological approaches, providing an overview of the field's recent advancements and trends.

2.2.2 Methodology

Following the Subsection 2.1, the Scopus database was chosen for this study, as it is among the most widely used research-literature databases, covering a broad array of peer-reviewed journals across multiple publishers. Regarding the keywords related to metaheuristic algorithms, we followed to the research of Erwin & Engelbrecht (2023), whereas the portfolio optimization terminology was derived from frequently utilized keywords in the existing literature. The complete query is presented in Table 9.

Most bibliometric studies of portfolio optimization mainly restrict their Subject Areas to Business, Management, Accounting, Economics, Econometrics, and Finance. Considering that portfolio optimization is studied by scholars in various fields, we seek to also include the domains of Computer Science, Engineering, Mathematics, and Decision Sciences. Additionally, since journals may be categorized across multiple classifications, we removed records that were co-classified in categories beyond our scope, even if they were also included in one of the previously specified categories.The dataset was restricted to English-language reviews and articles in the final publication phase, published until August 8, 2025. A total of 740 papers were acquired. To maintain the quality and reliability of our analysis, we excluded publications from journals scored below Q1 or Q2 in the 2024 Scimago Journal Rank (SJR), yielding a final sample of 541 papers. Regarding the tools and methodology, we use the Bibliometrix package and VOSviewer software following the Subsection 2.1

In alignment with the majority of current literature, our aim is to provide a thorough bibliometric analysis covering the most significant sources, affiliations, nations, and authors. To prevent any possible mistake arising from surname similarities, the author's full name has been utilized for the analysis. Ultimately, we perform a keyword analysis, merging relevant keywords as necessary, to identify the latest trends in metaheuristic portfolio optimization.

Table 9. Final query used for data collection.

TITLE-ABS-KEY((" metaheuristic*" OR "Artificial bee colony " OR "Ant colony optimization" OR "Bat algorithm" OR "Beetle antennae search" OR "Bacterial foraging algorithm" OR "Chemical reaction optimization" OR " Cat swarm optimization" OR "Evolutionary algorithm" OR "Firefly algorithm" OR "Fireworks algorithm" OR "Genetic algorithm*" OR "Grasshopper optimization" OR "Grey wolf optimization" OR "Krill herd" OR "Moth flame optimization" OR "Particle swarm optimization" OR "Simulated annealing" OR "Tabu search" OR "Whale Optimization Algorithm") AND ("portfolio optimization" OR "asset allocation" OR "Portfolio selection" OR "portfolio diversif*" OR "investment portfolio " OR "portfolio construction " OR "constrained portfolio*" OR "unconstrained portfolio*" OR "Black–Litterman" OR "Markowitz model" OR "cryptocurrency portfolio" OR "stock portfolio" OR "Sharpe ratio optimization" OR "Sortino Ratio optimization"))

Note: The wildcard symbol (*) used in the query to capture all variations of a word, such as singular, plural, and different suffixes.

2.2.3 Dataset Main information

Table 10 summarizes the dataset, including 541 documents from 157 sources, and shows an annual growth rate of 11.71%. A high level of collaboration is obvious, with 1205 authors contributing to this topic and just 31 single-authored works, yielding an average of 3.13 co-authors per document, while 22.74% of publications have international co-authors. The total number of 2,984 keywords shows a diverse array of topics explored, while the 3,405 references demonstrate a strong reliance on existing literature.In addition, each document averages 34.79 citations. Figure 10 shows the progression of annual scientific output over time. The first work was published back in 1994 (Konrad, 1994), and until the 21st century, it remained the only work in this domain; nevertheless, since 2009, a significant increase in publications has been noted. In recent years, there has been a steady rise in the volume of publications.

As of 2025, the figure remained comparatively elevated at the time the research took place. Figure 11 illustrates the distribution of journal subject categories within the dataset. The cumulative amount of category assignments is 1,109, as several journals are classified into multiple categories. Computer Science constitutes the largest category (402; 36.2%), succeeded by Mathematics (257; 23.2%) and Engineering (187; 16.9%).

Table 10. Summary statistics of the final dataset

Description	Results
Timespan	1994:2025
Sources	157
Documents	541
Annual Growth Rate %	11.71
Average citations per doc	34.79
References	3405
Keywords	2984
Authors	1205
Authors of single-authored docs	31
Co-Authors per Doc	3.13
International co-authorships %	22.74

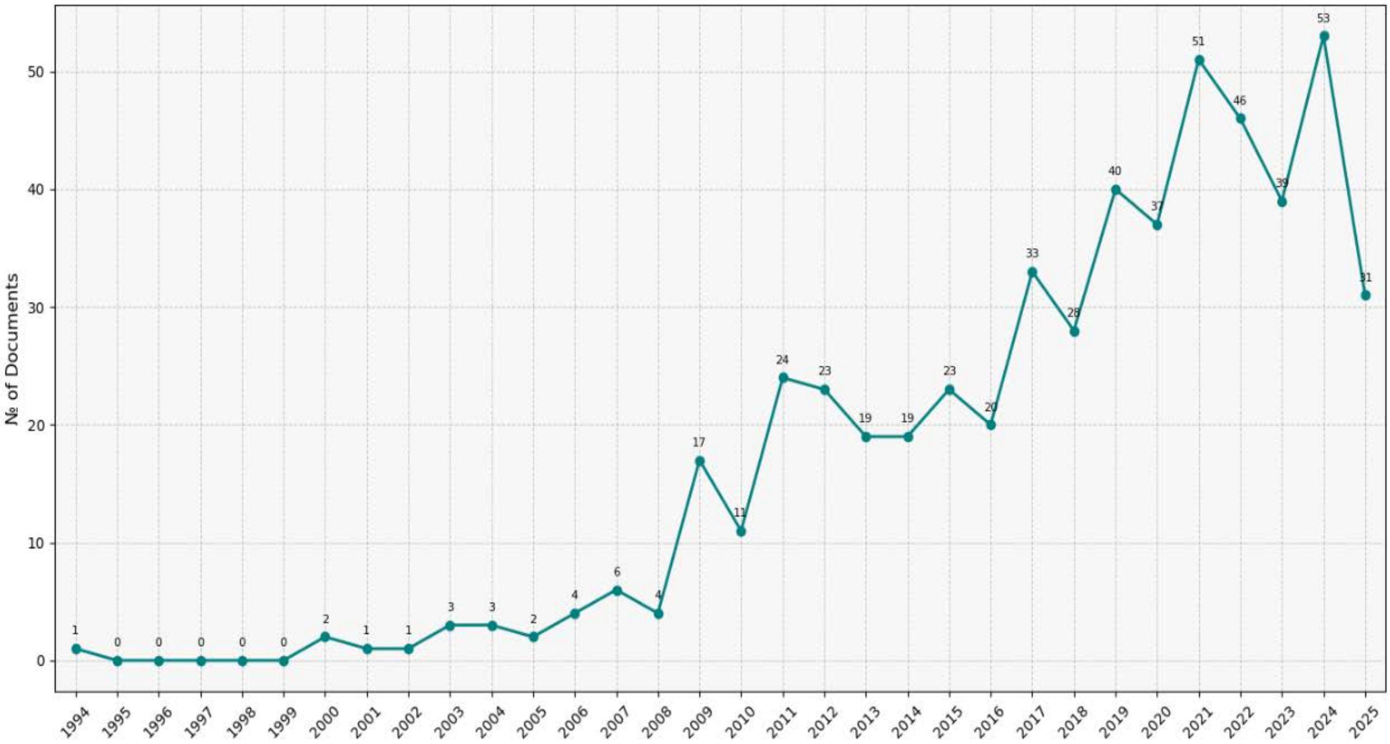


Figure 10 Documents Published per Year

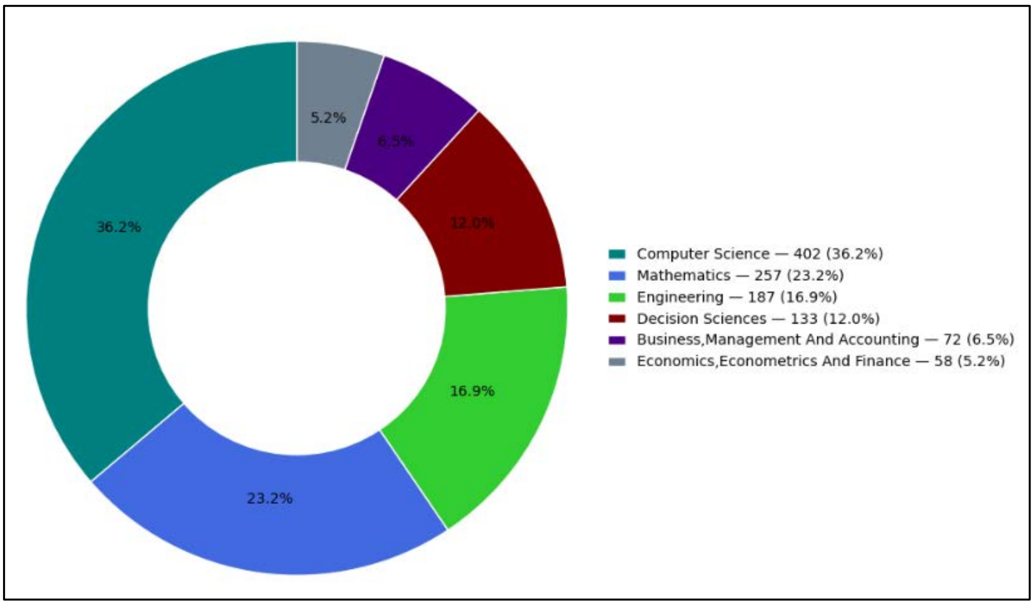


Figure 11 Distribution of journals by subject category

2.2.4 Most relevant sources

Next, we conduct an analysis of the most relevant journals. Numerous respected journals in Computing, Engineering, and Operations Research have published multiple studies in the topic of metaheuristic portfolio optimization (Table 11). Expert Systems with Applications (ESWA) ranks first, followed by Applied Soft Computing in second place, and the European Journal of Operational Research in third. Figure 12 illustrates the annual output of the journals, indicating that ESWA has retained its status as the leading journal in this topic for over a decade, whereas all journals have experienced a rise in publications in recent years.

Table 11. Top 15 productive journals in the field

Sources	Articles
Expert Systems with Applications	62
Applied Soft Computing	37
European Journal of Operational Research	21
Soft Computing	19
Swarm and Evolutionary Computation	18
Annals of Operations Research	13
Computers and Operations Research	12
IEEE Transactions on Fuzzy Systems	12
Information Sciences	12
Computational Economics	11
Mathematics	11
Mathematical Problems in Engineering	10
Computers and Industrial Engineering	10
Algorithms	9
Applied Intelligence	8

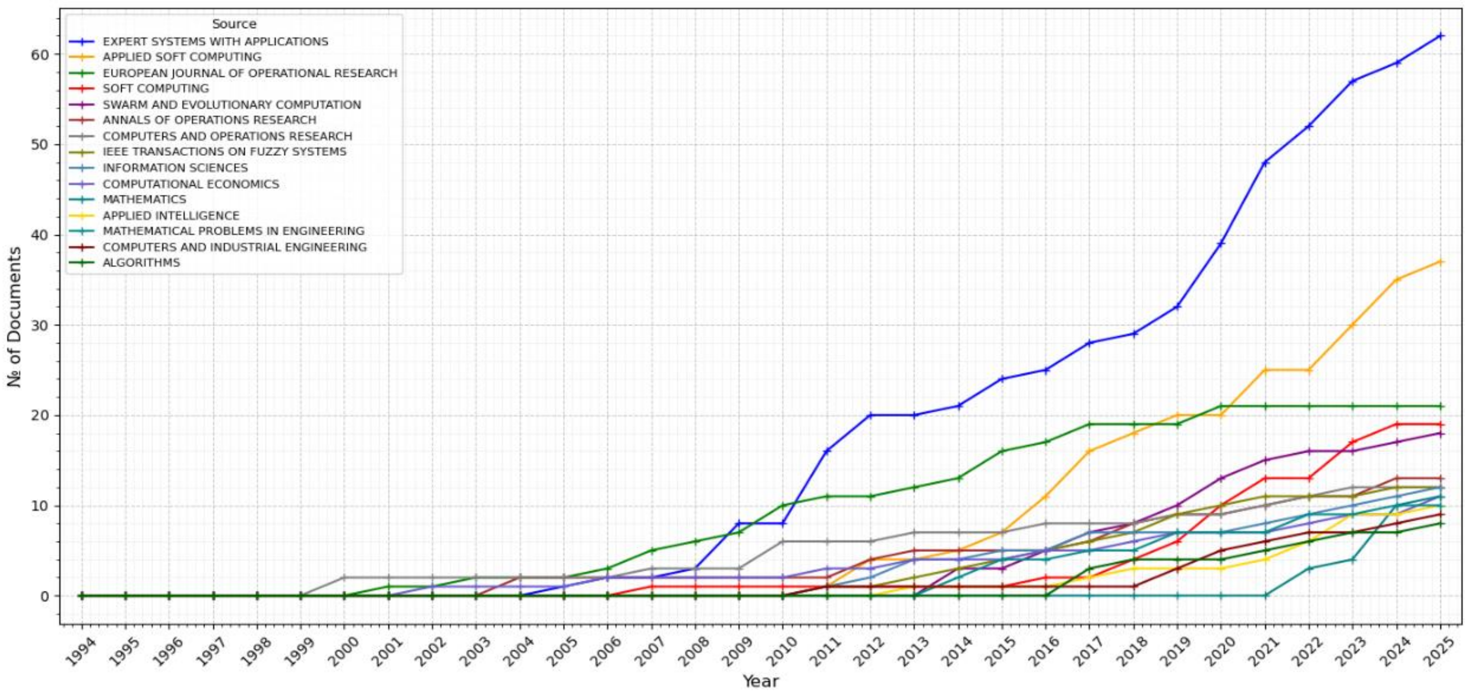


Figure 12 Yearly publication volume for the top 15 journals

2.2.5 Authorship analysis

Next, we examine the scholarly influence of authors throughout time. Table 12 indicates that Yongjun Liu and Mehlawat Mukesh Kumar are the most active authors, each publishing 16 works. In terms of Total Citations (TC), Mehlawat Mukesh Kumar leads with 650 TC, followed closely by Li Xhiang with 648 TC. Concerning article fractionalization, we see an average of 3.46 fractionalized articles per author. This suggests that, when accounting for co-authorship, the average author's productivity corresponds to almost three and a half single-authored articles, highlighting significant collaboration within the discipline. The significant discrepancies between raw and fractionalized counts in Table 12 indicate that publications are primarily generated by multi-author teams rather than individual authors.

Figure 13 illustrates the output per author across the years. Participation among notable authors considerably increased after 2013, reaching its zenith between 2018 and 2021. Output peaked in 2021, particularly for Mukesh Kumar Mehlawat. Xiang Li became an early participant in 2009. Many researchers exhibit persistent, multi-year contributions rather than occasional instances, indicating a consistent core of knowledge. Activity continues from 2023 to 2025, albeit at a diminished annual rate, indicative of a maturing field transitioning into a more stable pattern. Figure 14 illustrates the co-authorship network. Building on previous research (Van Nunen *et al.*, 2018; Kaushik & Nadeem, 2024), we incorporate authors with a minimum of two publications on the subject; the resultant network consists of 80 clusters.

Table 12
Top 15 most productive authors in the field

Authors	Articles	Articles Fractionalized	Total Citations
Liu, Yongjun	16	7.33	474
Mehlawat, Mukesh Kumar	16	5.28	650
Zhang, Weiguo	15	6.08	480
Gupta, Pankaj	13	4.92	387
Chen, Wei	9	2.62	467
Liagkouras, Konstantinos	9	4.83	303
Mittal, Garima	9	2.92	228
Thakur, Manoj	9	3.33	235
Fernandez, Eduardo R.	8	1.85	229
Li, Xiang	8	2.4	648
Metaxiotis, Kostas S.	8	3.83	262
Cruz–Reyes, Laura R.	7	1.35	121
Gómez-Santillán, Claudia Guadalupe	7	1.35	121
Katsikis, Vasilios N.	7	1.85	237
Bermúdez, José D.	6	1.98	376

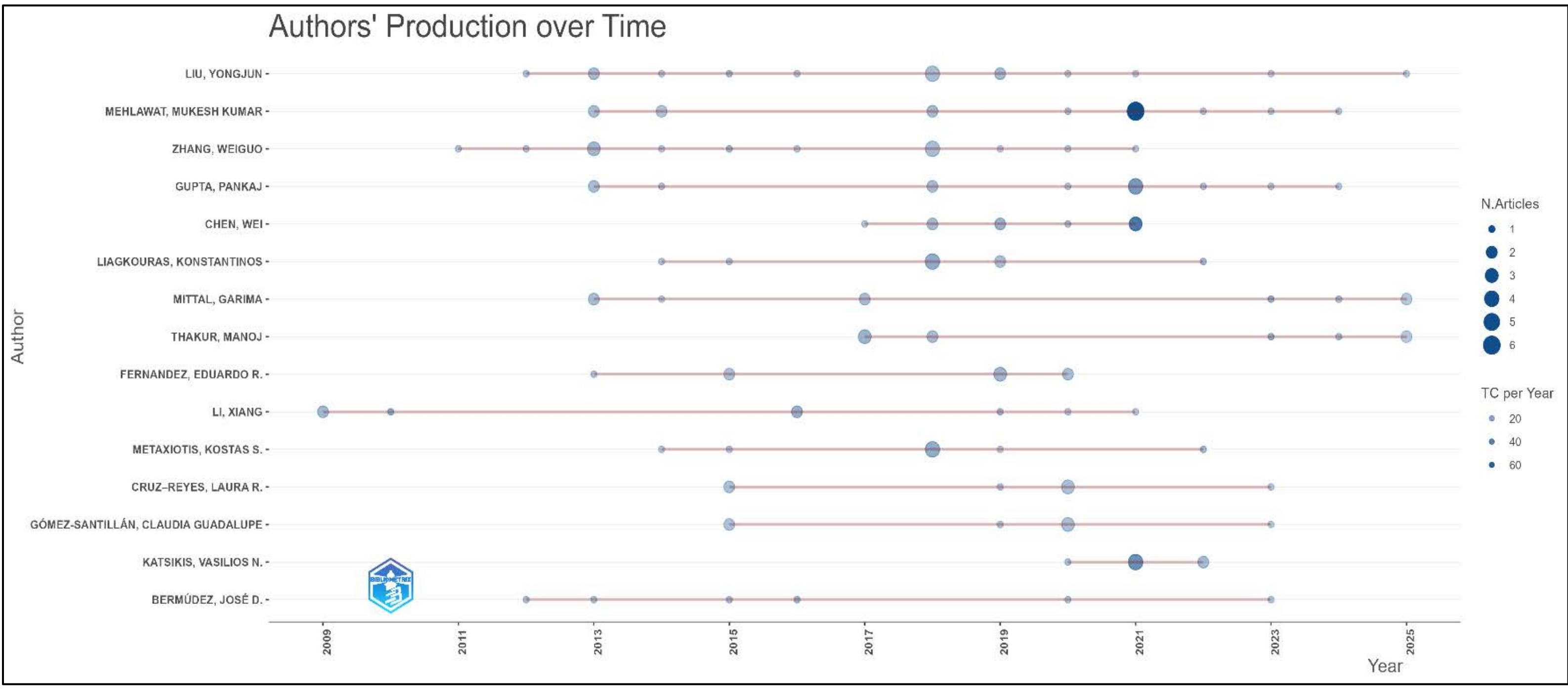


Figure 13. Top 15 authors' production over time

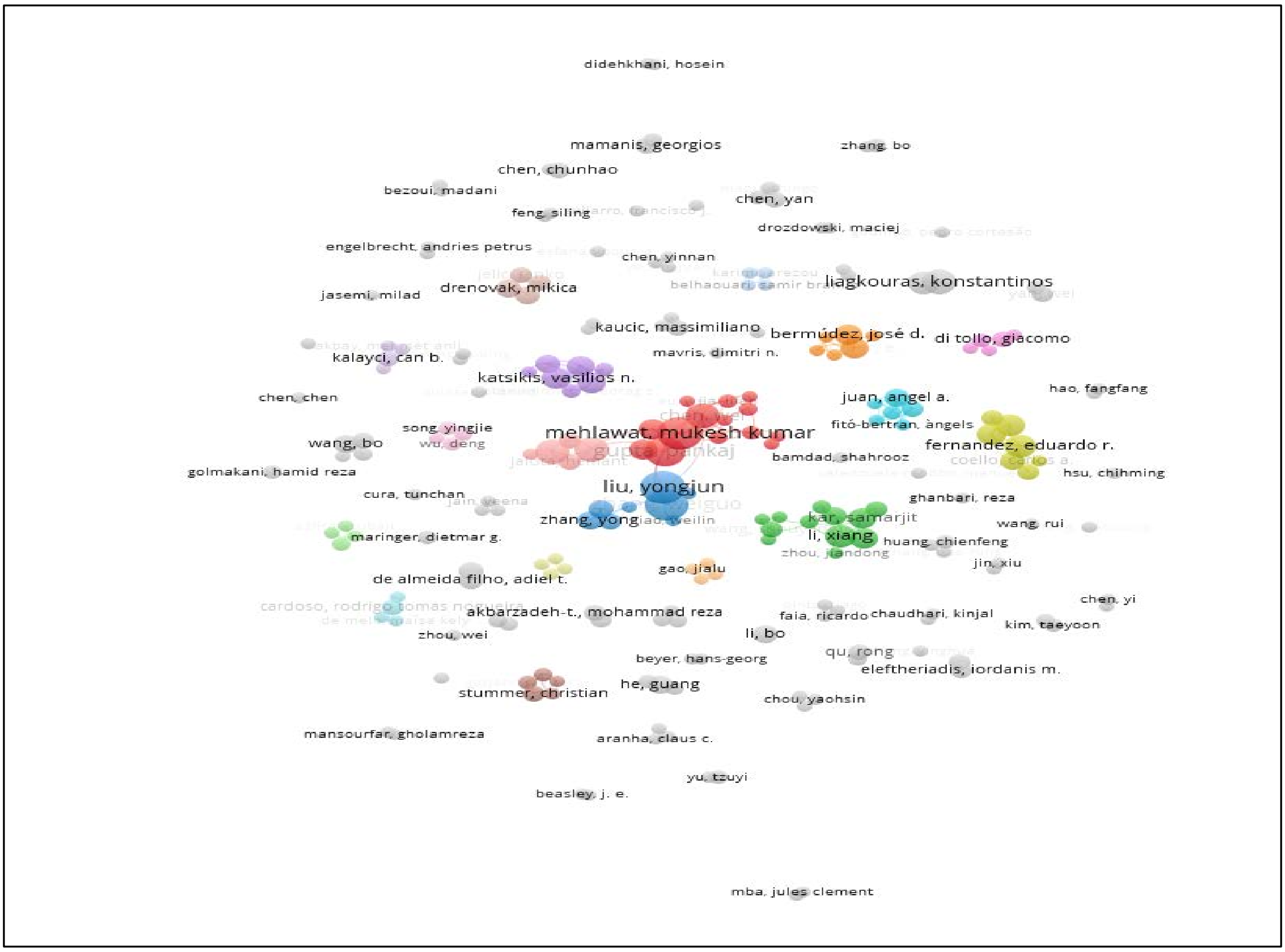


Figure 14. Co-authorship network in metaheuristic portfolio optimization

2.2.6 Affiliations and Countries Analysis

Table 13 and Figure 15 show the affiliation analysis, demonstrating that South China University of Technology and Islamic Azad University have produced the highest volume of publications over the years. Among the top 15 affiliations, Asia dominates in output, producing 274 of 351 papers (~78%), indicating significant capacity and strong collaboration among Asian universities. Europe ranks a distant second position with 48 articles (~14%), predominantly produced by Greece, Spain, and Austria, indicating a modest but consistent European presence. Table 14 and Figure 16 shows the country productions over times, based on the authors' affiliated nations (full counting), indicating that Asian countries, notably China, India, and Iran, are the predominant producers. Figure 17 shows the co-authorship among countries that have generated at least two publications. Eleven clusters have been found, signifying a worldwide collaboration in metaheuristic portfolio optimization.

Table 13
Top 15 most productive affiliations

Affiliation	Country	Articles
South China University of Technology	China	49
Islamic Azad University	Iran	39
University Of Delhi	India	39
Capital University of Economics and Business	China	32
National University of Defense Technology China	China	27
Amirkabir University of Technology	Iran	25
Guangdong University of Technology	China	20
University Of Piraeus	Greece	18
Universitat De València	Spain	16
Indian Institute of Technology Mandi	India	15
Universidad Autónoma De Sinaloa	Mexico	15
Tamkang University	Taiwan	14
University Of New South Wales At Australian Defence Force Academy	Australia	14
Universität Wien	Austria	14
Waseda University	Japan	14

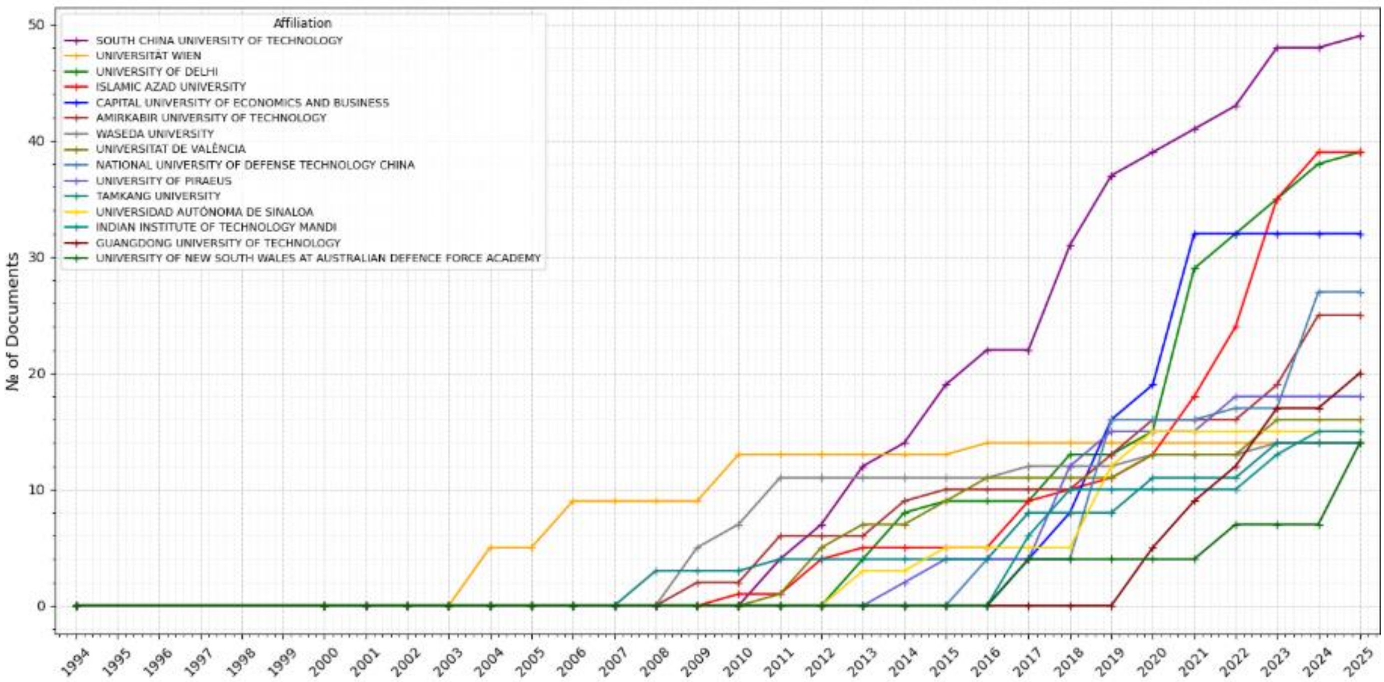


Figure 15. Top 15 affiliations' production over time

Table 14. Top 15 most productive countries

Country	Frequency
China	610
India	151
Iran	147
Spain	72
United Kingdom	71
Mexico	70
Greece	53
Italy	44
Brazil	41
USA	36
Austria	30
Australia	29
Japan	28
Portugal	25
Germany	24

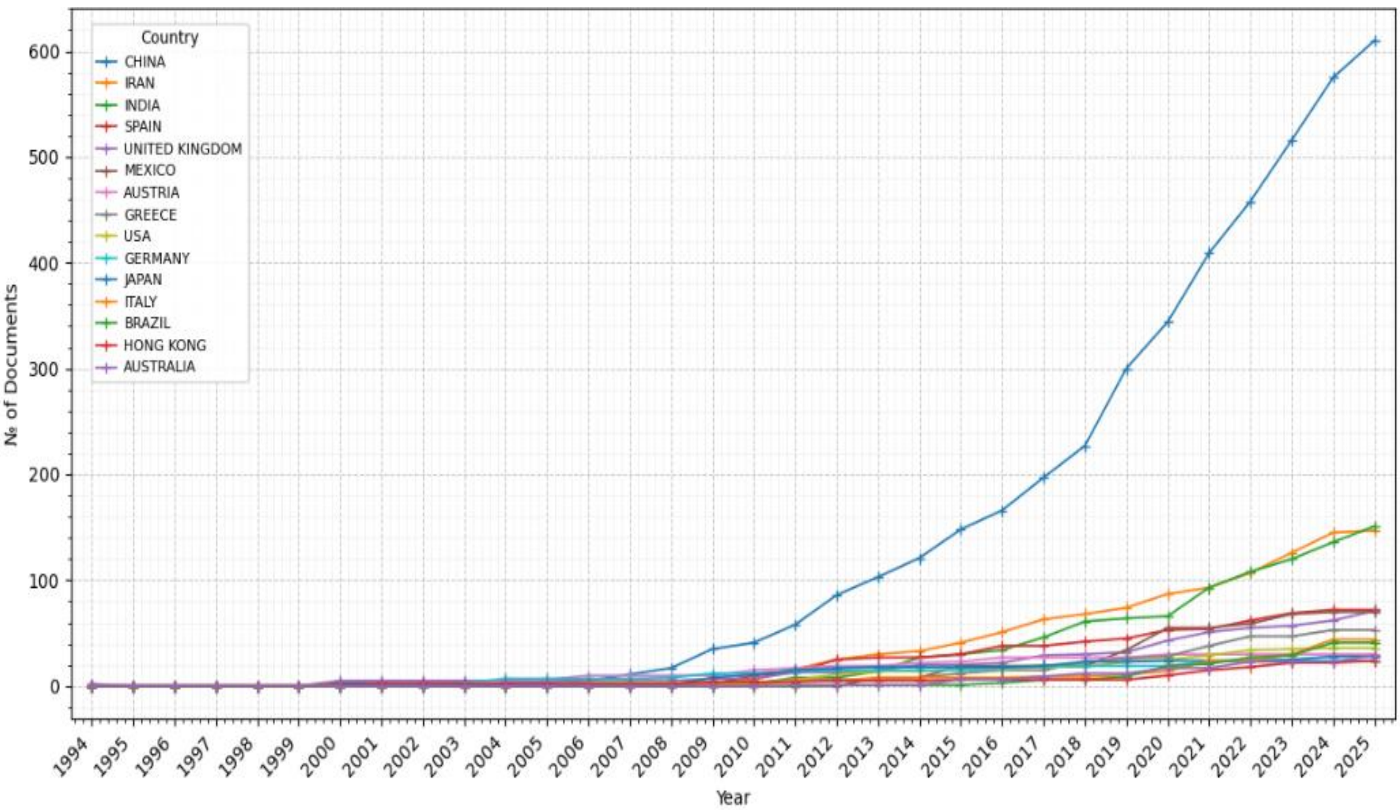


Figure 16. Top 15 countries' production per year

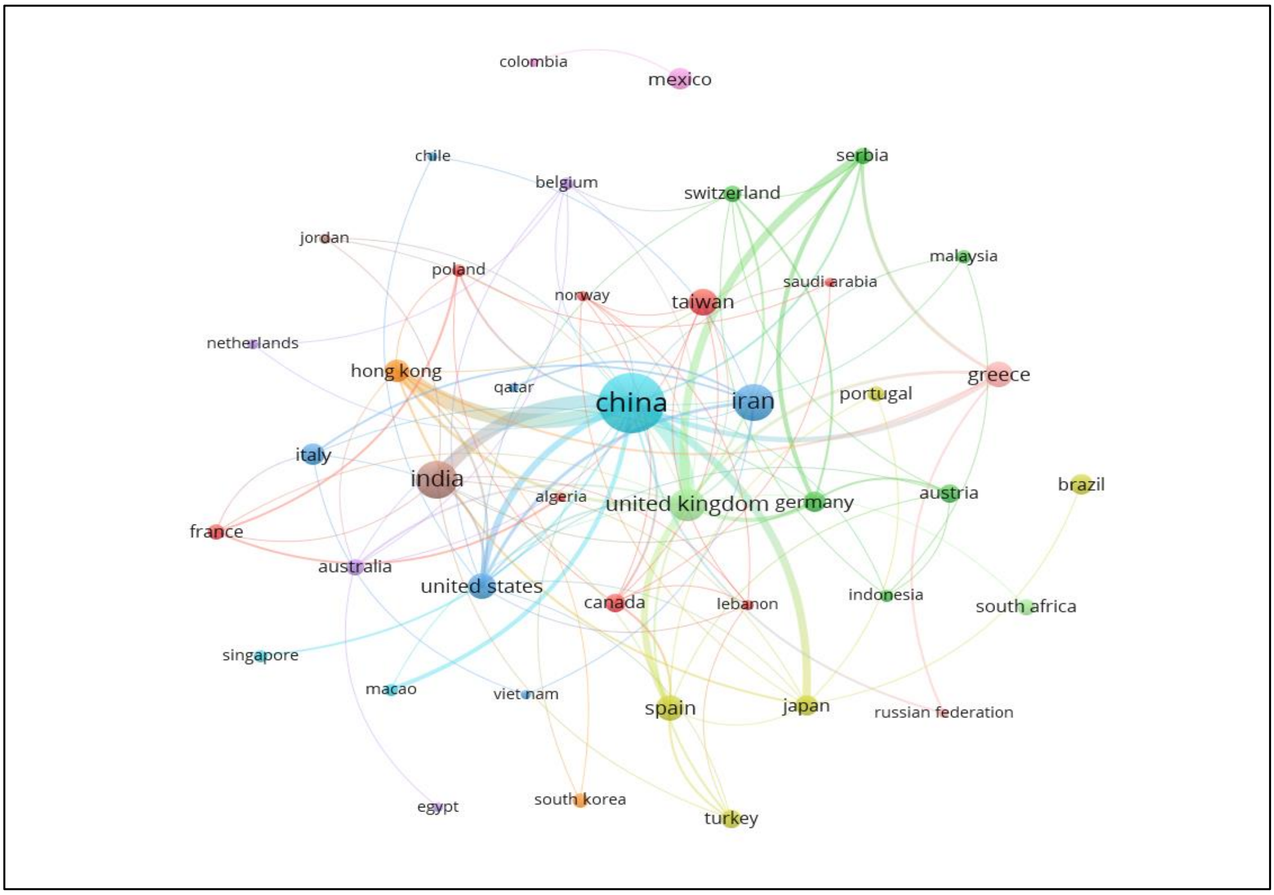


Figure 17. International co-authorship network among countries

2.2.7 Most-cited documents

Table 15 provides the documents that are most commonly referenced. The work by Chang *et al.* (2000), one of the first to explore the use of metaheuristics in cardinality-constrained portfolio optimization, is the most often cited work in respect of both total and average citations. The work of Doerner *et al.* (2024), which presented the Pareto Ant Colony Optimization metaheuristic for solving the portfolio optimization problem, is placed second in terms of TC. Ponsich *et al.* (2012), provide a taxonomy of multi-objective evolutionary algorithms applicable to finance and economics, especially in portfolio optimization, is ranked second in annual citation count. In terms of annual TC and normalized TC, the research of Chen *et al.* (2021) holds the highest position. The authors investigate portfolio construction using a hybrid model that combines an optimized eXtreme Gradient Boosting (XGBoost) model,fine-tuned by an Improved Firefly Algorithm (IFA) for stock price forecasting, and a mean-variance (MV) optimization model for portfolio selection. Their proposed methodology exhibits superior effectiveness compared to earlier benchmark models.

The journals with the highest citation rates largely belong to the domains of fields of computer science, mathematics, and operational research, as confirmed by Table 3. Among the 15 most cited documents, the study of Chen *et al.* (2021) is the most recently published. In recent years, metaheuristic algorithms have garnered significant popularity for their integration with AI models, as they proficiently tackle various challenges, including hyperparameter optimization and feature selection (Bacanin *et al.*, 2023; Tayebi & El Kafhali *et al.*, 2025; Dokeroglu *et al.*, 2022; Sadeghian *et al.*, 2025).

Table 15. Top 15 Most-Cited Articles

Paper	DOI	TC	TC/Year	NormTC
Chang *et al.*,2000, Comput. Oper. Res.	10.1016/S0305-0548(99)00074-X	709	27.27	1.62
Doerner *et al.*,2004, Ann. Oper. Res.	10.1023/B:ANOR.0000039513.99038.c6	367	16.68	2.58
Ponsich,2013, IEEE Trans. Evol. Comput.	10.1109/TEVC.2012.2196800	301	23.15	4.90
Xiang *et al.*,2013, R. J.	10.32614/rj-2013-002	283	21.77	4.61
Zhu *et al.*,2011, Expert Syst. Appl.	10.1016/j.eswa.2011.02.075	282	18.80	3.91
Crama & Schyns,2003, Eur. J. Oper. Res.	10.1016/S0377-2217(02)00784-1	260	11.30	1.91
Fernández & Gómez,2007, Comput. Oper. Res.	10.1016/j.cor.2005.06.017	257	13.53	2.84
Li *et al.*,2010, Eur. J. Oper. Res.	10.1016/j.ejor.2009.05.003	250	15.63	3.09
Chang *et al.*,2009, Expert Syst. Appl.	10.1016/j.eswa.2009.02.062	238	14.00	3.12
Cura,2009, Nonlinear Anal. Real World Appl.	10.1016/j.nonrwa.2008.04.023	218	12.82	2.86
Chen *et al.*,2021, Appl. Soft Comput.	10.1016/j.asoc.2020.106943	213	42.60	7.32
Huang,2012, Appl. Soft Comput.	10.1016/j.asoc.2011.10.009	205	14.64	5.31
Huang,2008, Comput. Appl Math.	10.1016/j.cam.2007.06.009	198	11.00	1.79

Lin & Liu,2008, Eur. J. Oper. Res.	10.1016/j.ejor.2006.12.024	198	11.00	1.79
Ardia *et al.*,2011, R J.	10.32614/rj-2011-005	182	12.13	2.52

Note. TC: Total Citations, **NormTC:** Normalized Total Citations

2.2.8 Keywords Analysis

The following subsection presents a keyword analysis to offer a wider picture of the latest trends in metaheuristic portfolio optimization. Figure 18 and Table 16 show the 15 most frequently utilized authors' keywords. The most commonly detected keyword is "portfolio optimization," followed by "genetic algorithms," "multi-objective optimization," and "particle swarm optimization." The trend map (Figure 19) illustrates the progression of the domain. Prior years emphasized conventional financial theories, such as the Markowitz model and the efficient frontier, with classical heuristic methods including tabu search, local search, and simulated annealing. Since the mid-2010s, attention has transitioned to metaheuristic algorithms, such as genetic algorithms, particle swarm optimization, and evolutionary algorithms.

GA has been extensively examined over the years for several optimization tasks, including portfolio optimization. Nevertheless, Particle Swarm Optimization (PSO) has gained increasing popularity compared to GA, especially in recent times, due to the development of numerous approaches and modifications. (Kaushik D. & Nadeem, 2025; Bulani *et al.*, 2025). Furthermore, there is an increasing interest in multi-objective optimization. The primary emphasis transitioned to finance-related topics from 2019 to 2022, reflecting an increasing interest in electronic trading, while cardinality limitations continued to be pertinent. In recent years (2022–2024), methodological approaches have advanced to incorporate machine learning, index tracking, sensitivity analysis, and uncertainty theory.

The evolution of metaheuristic portfolio optimization shifts from conventional techniques to modern metaheuristic algorithms, especially those derived from swarm intelligence, and more recently to hybrid, data-driven approaches that integrate machine learning and deep learning models, emphasizing scalability, uncertainty, and robust decision support. Figure 20 shows the co-occurrence network of keywords, highlighting those that appear a minimum of five times, resulting in a total of seven visible clusters. This picture illustrates the increasing use of several metaheuristic algorithms, indicating that academics frequently analyze and assess them for portfolio optimization. This is anticipated, as existing algorithms like PSO have demonstrated commendable performance, and the community persists in evaluating their performance. Considering that the majority of metaheuristic algorithms were introduced in the past decade (Rajwar *et al.*, 2023), we assert that there remains an opportunity for thorough assessment of recent algorithms in this domain.

Table 16. Top 15 most frequent keywords.

Terms	Frequency
portfolio optimization	689
genetic algorithm	306
multi-objective optimization	220
particle swarm optimization	162
financial data processing	158
investments	157
evolutionary algorithms	130
optimization	106
electronic trading	105
financial markets	100
heuristics	85
cardinality constraints	80
metaheuristics	69
commerce	64
decision making	58

Note: Keyword synonyms have been merged

Figure 18 Word cloud of the 15 most frequent keywords. **Note**: Keyword synonyms have been merged

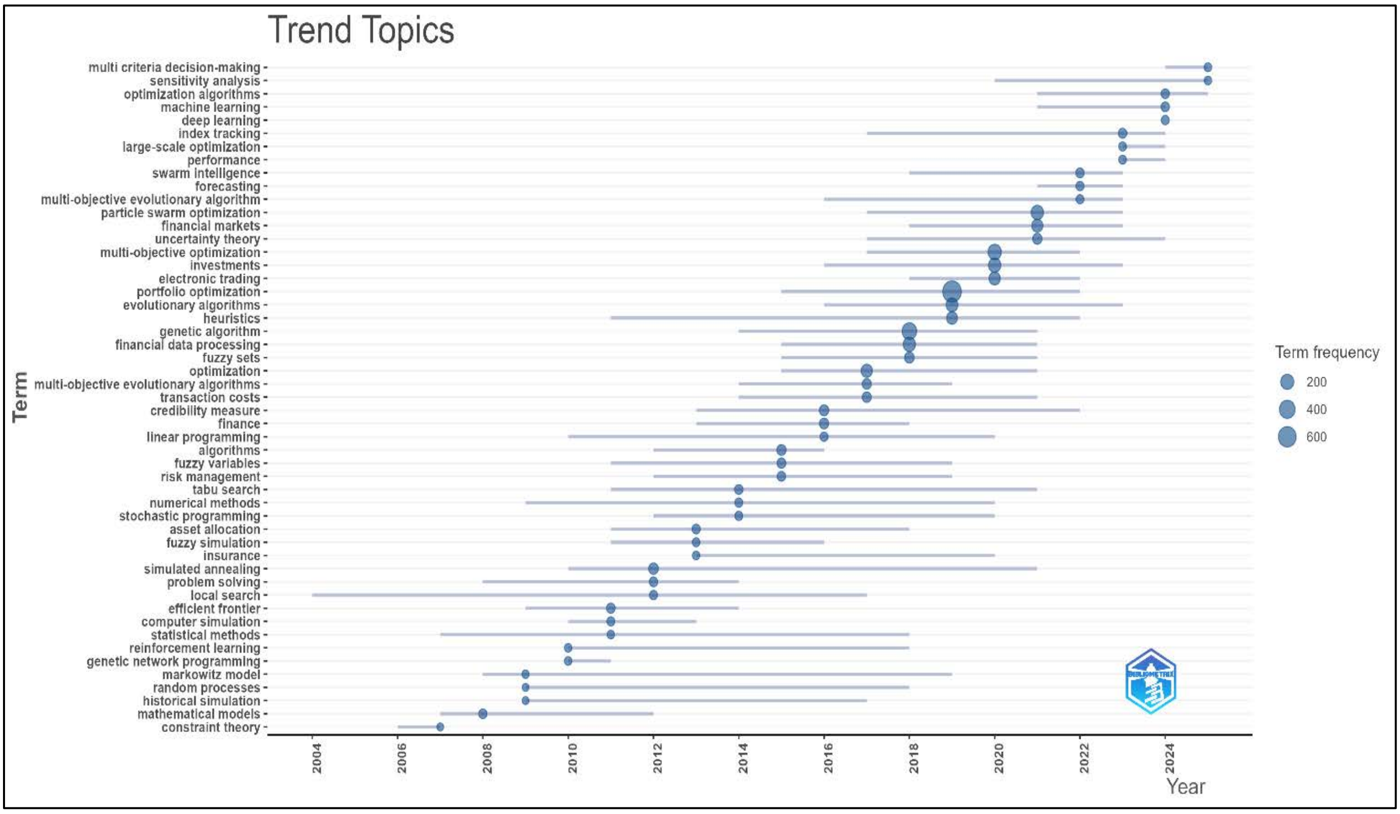


Figure 19 Thematic evolution from keyword analysis. **Note:** Keyword synonyms have been merged

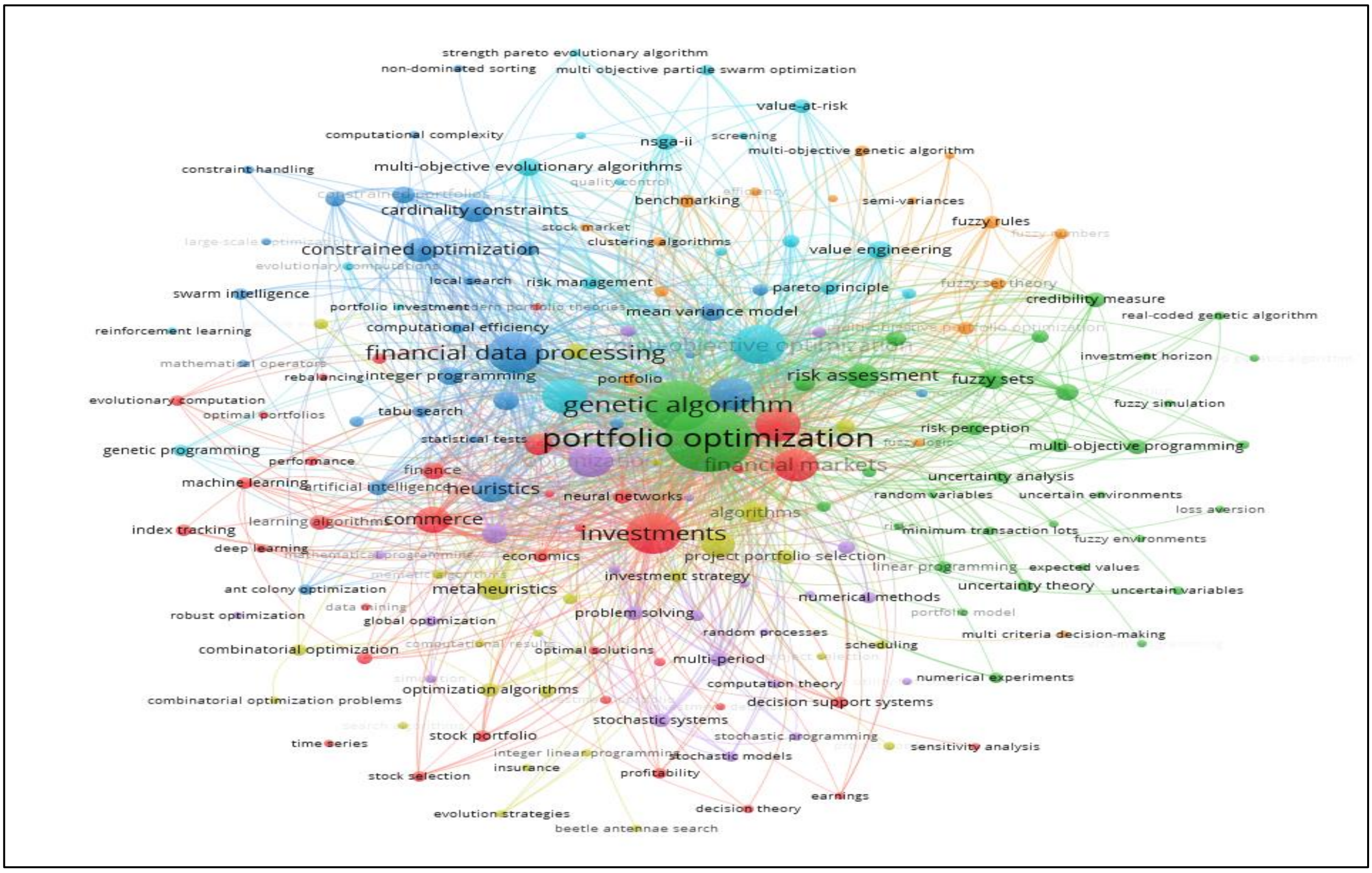


Figure 20. Co-occurrence of keywords in metaheuristic portfolio optimization**. Note:** Keyword synonyms have been merged

2.2.9 Discussion

This subsection presents a bibliometric analysis of metaheuristic portfolio optimization. The results reveal a steady rise throughout time, with contributions from several authors, affiliations, and nations. The global application of metaheuristic algorithms for portfolio optimization has emerged as a significant research domain, with contributions from numerous countries. Asian nations dominate in publishing output, and affiliations located in Asia exhibit the highest productivity. We also note significant cross-national collaboration among authors, highlighting considerable interest within the research community. Moreover, the most productive journals are concentrated iin computational disciplines—computer science and mathematics—and in operations research. Keyword analysis indicates a transition towards modern swarm-intelligence methodologies, moving away from traditional finance techniques. Despite this shift over the years, authors continue to favor more established metaheuristic algorithms.

Furthermore, analysis of trends indicates that in recent years, AI algorithms, including machine and deep learning models, have been combined with metaheuristic algorithms for portfolio optimization, underscoring the potential for future developments. Nonetheless, we contend that, despite the proliferation of publications regarding these algorithms in the past decade, only a limited subset has undergone consistent evaluation; hence, a gap persists in the application and comparative analysis of numerous metaheuristic algorithms. Furthermore, despite the growing interest in electronic trading in recent years, research assessing these algorithms for portfolio optimization for modern financial instruments, including cryptocurrencies, remains limited.This work does not come without limitations. Although we analyzed various published metaheuristics, a broader search query and the incorporation of non-English articles would likely uncover further insights. Applying SJR-based criteria at the final screening stage improved the credibility of the dataset but may have excluded some valuable studies. Furthermore, at the time of this analysis, publication numbers are increasing; however, the data for the current year are incomplete.

Our findings suggest an increasing significance of metaheuristic algorithms in portfolio optimization and an increasing global interest among countries, institutions, and researchers. Although the literature contains shortcomings—particularly in assessing newer algorithms—the field seems to be advancing well. Given to their versatility and computational efficacy, these algorithms can address complex portfolio challenges, rendering them a robust choice for investors and practitioners seeking efficient and informed decisions.

2.3 Churn Prediction

He et al. (2014) introduced a radial basis function–support vector machine (RBF–SVM) model for predicting customer churn by employing random sampling techniques within the majority class, leading to a more balanced training sample and improved training performance. The researchers examined several combinations of sampling strategies employing a dataset from a Chinese bank, which, following feature engineering, had 46,406 valid data points. Their findings indicated that the RBF–SVM model employing an under-sampling technique yielded the most favorable performance. Vafeiadis et al. (2015) performed a comparison of several machine learning models, including Decision Trees (DT), SVM, K-nearest neighbors (KNN), Naïve Bayes (NB), and neural network (NN) models, using data from a Greek telecommunications company. The SVM and NN models performed better, with the NN model slightly outperforming the SVM.De Caigny et al. (2018) introduced a hybrid approach, the Logit Leaf Model (LLM) which combines the advantages of DT and logistic regression (LR) to overcome the drawbacks of standalone use. The authors performed experiments on 14 churn datasets, revealing that the LLM surpasses both DT and LR in terms of accuracy.

Höppner et al. (2020) point out an important limitation in classical machine learning models. These models often conflict with the core goal of profit maximization because they ignore misclassification costs and the benefits of accurate predictions. For that purpose the authors proposed the expected maximum profit measure (EMPC) for customer churn to identify the optimal churn model. The authors presented ProfTree, a novel classifier that integrates the EMPC metric. ProfTree employs an evolutionary algorithm to create multiple DTs based on profit optimization. ProfTree showed better performance in comparison to traditional tree-based approaches across multiple telecommunications datasets. Geiler et al. (2022) examined the advantages and disadvantages of machine learning models for forecasting customer churn. The authors highlighted the importance of feature selection, data preprocessing, and effective handling of class imbalance. In addition, the authors discuss the use of the proper evaluation criteria. Numerous works have examined the importance of feature engineering and data preprocessing for developing effective models (Mitrović et al., 2018; Sana et al., 2022).

Kimura (2022) examines the performance of several machine learning models, including also three different resampling methods. The findings showed that ensemble models perform better than classical machine learning models. The results indicated that resampling approaches improve the F1-score for all models. However, in terms of AUC score, the resampling methods did not improve the performance of the model. Regarding the effectiveness of ensemble methods, similar results have been observed in previous studies (Imani & Arabnia, 2023). Durkaya et al. (2023) introduced the hybrid PCA–GWO–SVM model. At first the proposed approach starts with Principal Component Analysis (PCA) to identify the most important characteristics of the dataset. Then the Grey Wolf Optimizer (GWO) metaheuristic algorithm is employed to optimize the hyperparameters of an SVM model. The results showed that the proposed hybrid model surpassed other machine learning models in terms of accuracy, recall, and F1 score.Simsek & Tas (2024) introduced the ROS-Voting (RF-LGBM) model, which employs the Random Oversampling (ROS) approach with a voting ensemble method, which consists of a random forest (RF) and a Light Gradient-Boosting Machine (LightGBM) model. The findings show that the suggested approach performed better in comparison to other applied machine learning models. Haddadi et al. (2024) state that Synthetic Minority Oversampling Technique (SMOTE) is often a conventional way for resampling techniques, and the utilization of SMOTE–Tomek Links and SMOTE–ENN in customer churn prediction is limited.

In the field of customer churn prediction, several informative works provide a comparison and evaluate the performance of classical and ensemble machine learning models, as well as neural network models, with or without resampling techniques (Burez & Vanden Poel, 2009; Vafeiadis et al., 2015; Zhu et al., 2017; Geiler et al., 2022; Imani & Arabnia, 2023; Haddadi et al., 2024). Additionally, there is limited research in this area evaluating NN performance and the impact of hyperparameters (Dalli, 2022; Domingos et al., 2021). To our knowledge, customer-churn modelling—especially in the banking sector—has not systematically investigated the combined effects of NN components (including recently proposed variants) and resampling methodologies. Most prior studies that employ NN for related tasks rely on well-established activation functions and optimizers. However, newer components that may improve forecasting accuracy—such as the APTx activation—remain largely unevaluated. We address this gap by concurrently varying network components and comparing overall performance across multiple resampling methods.

2.4 Credit card Default

Hand & Henley (1997) have provided an extensive review of several classification approaches for credit scoring and discuss the strengths and limitations of each approach. Although the authors provide some informative guidelines, they conclude that no single approach is universally superior. The performance of each method highly depends on the dataset and the problem which tries to address. Srinivasan & Kim (1987) evaluated the performance of parametric and non-parametric methods for credit granting. The findings showed that non-parametric methods, such as DT, can produce superior classification results. Malhotra & Malhotra (2000) performed a comparative analysis of statistical models and several NN models across twelve different datasets of credit unions. The findings showed that the NN models surpassed LR model. The double-layered backpropagation model with adaptive learning and the Learning Vector Quantization (LVQ) model achieved the best classification results. Furthermore, Galindo & Tamayo (2000) reviewed various statistical and machine learning methods for credit risk evaluation. The results indicated that Classification and Regression Trees (CART) outperform both KNN and NN models. Furthermore, all machine learning models outperform the conventional probit model.

Abdou (2009) evaluates the effectiveness of Genetic Programming (GP) in comparison to Probit Analysis (PA) and the weight of evidence (WOE) measure, within Egyptian public sector banks. The results showed that GP attained superior average correct classification rates and reduced estimated misclassification costs. Nonetheless, WOE is beneficial in situations with high misclassification costs. The research suggests that GP is a viable tool in credit scoring, while recognizing the contextual applicability of WOE. Furthermore, Yang & Zhang (2018) performed an investigation of five different machine learning models, revealing through three evaluation criteria that LightGBM outperforms LR, Neural Networks (NN), SVM, and XGBoost. Gan et al. (2023) further demonstrated the robust performance of LightGBM model.

Yeh & Lien (2009) focus on analyzing the forecasting accuracy of default probabilities through six distinct data mining methodologies. They underscore the importance of risk management by emphasizing the significance of precisely forecasting the real possibility of default rather than relying on a binary classification of client credibility. To handle the issue of the uncertain real default probability, they proposed the 'Sorting Smoothing Method'. The results show that the NN model performed better in comparison to KNN, LR, Discriminant Analysis (DA), NB, and Classification Trees. The performance of several Machine learning models has been evaluated on the respective dataset over the years(Teng & Lee, 2019; Sariannidis et al., 2020)

Sun & Vasarhelyi (2018), utilizing a dataset of over 700,000 credit card holders from a major Brazilian bank, compared the performance of Deep Neural Networks (DNN) against various machine learning models. The results indicate that DNNs models provide better forecasting performance in predicting credit card defaults. He et al. (2018) introduced an ensemble model to improve credit scoring performance by adjusting different class-imbalance ratios. The proposed model is an extension of the BalanceCascade methodology, which is a supervised undersampling method, that generates balanced subsets specific to the imbalance ratio of the training data. The proposed methodology uses a three-stage ensemble that employs optimized RF and XGBoost as basic classifiers, using the Particle Swarm Optimization (PSO) algorithm for optimization purposes. The experiments on different datasets using various measures, confirmed its accuracy over other classical models. Jadhav et al. (2018) examines the problem of high-dimensional datasets, which can reduce model accuracy. The authors introduced the Information Gain Directed Feature Selection (IGDFS) approach, which ranks features by information gain, and then employs a Genetic Approach wrapper (GAW) with KNN, NB, and SVM classifiers. Their method improves computational efficiency by dropping the less significant features of the dataset, and the experiments showed that can surpass the baseline models.

Alam et al. (2020) evaluated the performance of various resampling methods and machine learning models across many benchmark datasets. The findings suggested that a balanced dataset can enhance classifier performance, especially when the of k-means SMOTE was applied. The findings also showed that the Gradient Boosted Decision Tree (GBDT) outperforms other machine learning models. Şen et al. (2020) introduced a framework to assess loan applications for small businesses using a binary score. The authors state that small enterprises constitute 60% of the economy, so, even small improvements in terms of prediction are important. For that purpose, the authors proposed a hybrid methodology that comprises of an optimized SVM via GA algorithm. Using multiple benchmark datasets, the proposed model showed promising performance.

The work of Gao et al. (2021) show that transaction flow features are important, and these features can improve the forecasting performance of an XGBoost model. Moreover, the authors employed an LSTM model, allowing transaction-flow data to be properly processed. The hybrid XGBoost-LSTM performed the best in comparison to other machine learning models. Sharifi et al. (2021) presented an approach that combines NN with an enhanced Owl Search algorithm (IOSA) and a C5 DT model to optimize credit risk prediction. Their methodology starts with the DT operating on an IOSA to find the optimal weights for the NN model. Next, the NN model acquires the weights together with the respected credit data. The proposed methodology evaluated utilizing data from an Iranian bank, where it demonstrated promising performance. Zhang et al. (2021a) introduced a multi-stage model to improve accuracy in credit rating. The authors employed a bagging-enhanced local outlier factor (BLOF) method to handle outliers adaptively, a dimension-reduced feature transformation technique, and a stacking-based ensemble model by using multiple candidate optimizers and self-adapting parameters to improve the classifier's performance.

Liu et al. (2022) proposed an advanced multi-layered GBDT, the multi-grained and multi-layered GBDT (mg-mGBDT) model. The authors state that classical tree-based models, although are effective in credit scoring, lack sufficient representational depth to reveal the underlying structure of loan data. The mg-mGBDT integrates a multi-grained scanning approach that improves feature representation using a multi-layered GBDT, hence augmenting classification capability. Evaluated across six credit datasets, mg-mGBDT demonstrated great performance, either surpassing or matching the accuracy of classical approaches such as XGBoost. Zhou et al. (2023) suggested a two-stage credit scoring model for small businesses in China, using loan data from 3,045 firms . Their methodology includes the SMOTE method to address class imbalance. Next, a RF model constructs predictive credit features that serve as inputs for a LR model. Their approach evaluates both financial and non-financial factors, emphasizing on macroeconomic conditions. The performance of the model's was evaluated using several machine learning methods and datasets, showing its effectiveness.

Singh et al. (2024) presented the Self-Organizing Map Clustering with Metaheuristic Voting Ensembles model (SCMVE). The particular methodology combines clustering and classification algorithms. At first, the self-organizing map (SOM) is utilized to cluster data according to borrower characteristics, followed by the application of an optimized SVM-KNN through the Sailfish Optimizer for classifying each cluster. The authors compared the performance of the SCMVE model on three different datasets and the proposed approach showed better performance. Xu & Qu (2024) examined the performance of four different metaheuristic algorithms for optimizing a NN model to improve predictive performance in credit risk assessment. The four algorithms include the Whale Optimization Algorithm (WOA), Harmony Search (HS), Multi-Verse Optimization (MVO), and Vortex Search (VS).The MVO is a metaheuristic algorithm inspired by physics and the assumption that multiple universes exist. The MVO-NN achieved the best performance in both training and test scores compared to the other three metaheuristic algorithms. The findings are aligned with the previous work by Zhang & Qiu (2020), who also examined metaheuristic-optimized NN models.

Wang et al. (2025) discuss the challenges of interpreting AI models, although their exceptional performance. The authors employed a two-stage methodology that includes SHapley Additive exPlanations (SHAP) for feature importance analysis, followed by the application of a GA algorithm to produce sparse, effective counterfactuals that alter model outcomes. The researchers tested the efficiency of the suggested method on three different, datasets using four different classifiers.The results showed that discovering enhanced performance in terms of both effectiveness and sparsity relative to conventional and prototype-based counterfactual methods.Zhang et al. (2025) introduced the IWSL model, which builds inherently interpretable features by deriving representative eigenvectors for defaulted, non-defaulted, and midpoint sample distributions in feature space. Next, the Cuckoo Search (CS) algorithm fine-tunes attribute weights by optimizing generalization performance on the validation set. The results—based on three public consumer credit datasets—show that IWSL outperforms 12 benchmark models.

Prior studies on credit-card default prediction span classical statistical methods and a wide range of machine-learning models, with the latter generally outperforming the former. The literature also shows extensive use of hyperparameter optimization and hybrid pipelines to tackle class imbalance and high dimensionality. Given the strong capabilities of LightGBM and the efficiency of the PSS optimizer, we ask whether careful hyperparameter tuning alone can deliver competitive performance—potentially reducing the need for increasingly complex and time-consuming modelling pipelines.

2.5 Portfolio Optimization

Dueck & Winkler (1992) presented the Threshold Accepting (TA) algorithm, which can optimize portfolios subject to nearly arbitrary constraints and virtually any utility function. The authors claim that, by employing technical analysis, an investor might enhance their portfolio within a realistic model. They conclude that risk is not a standard unit of measurement; hence, portfolios that perform well within one paradigm may underperform in another. Testing TA across numerous scenarios for portfolios comprising bonds from the Federal Republic of Germany, the results revealed the absence of a universal risk measure and underscored the necessity for diverse optimizers for different risk functions. We could claim that they unofficially formulated an NFL theorem for portfolio optimization.

One year later, Arnone et al. (1993) reapproached the concept of portfolio optimization utilizing GA (Holland, 1992), stating that stochastic methods could be advantageous when deterministic optimization algorithms underperform and that GA is capable of exploring extensive search spaces. Mozafari (2011) attempted to address a realistic optimization problem with floor, ceiling, and cardinality constraints. The author introduced the hybrid IPSO-SA model, which integrates an improved Particle Swarm Optimization (PSO) (Kennedy & Eberhart, 1995) with a modified Simulated Annealing (SA) (Kirkpatrick et al., 1983) algorithm. IPSO-SA surpassed GA, Tabu Search (TS) (Glover, 1989), SA, and PSO in performance across five benchmark data sets. Mishra et al. (2014) introduced the Non-dominated Sorting Multi-Objective Particle Swarm Optimization (NS MOPSO) to tackle a realistic portfolio asset selection with budget floor and ceiling constraints and cardinality as a multi-objective optimization problem. The researchers evaluated the model's performance along with four other metaheuristic algorithms over six distinct market indices, and the results showed that the proposed method appears highly promising.

Bacanin & Tuba (2014) proposed a modified Firefly Algorithm (FA) (Łukasik & Żak, 2019) for a cardinality constrained mean portfolio optimization problem with entropy constraints. The authors combined the Firefly algorithm (FA) with the Artificial Bee Colony (ABC) optimization algorithm (Karaboga, 2005) to improve FA's exploration performance. Their approach provided better results in contrast to other traditional metaheuristic algorithms. Similarly, for ABC, a modification was proposed by Chen (2015), who presented the Modified Artificial Bee Colony (MABC). Ruiz-Torrubiano & Suárez (2015) developed a memetic technique that combines GA and quadratic programming to address the issue of portfolio selection with cardinality constraints and piecewise linear transaction costs. Their approach works by handling the continuous and combinatorial parts of optimization independently. The modification of the Random Assorting Recombination (RAR) crossover operator (Radcliffe, 1993) to modify traditional attributes in the genetic representation that indicate the direction of the transactions in each asset during rebalancing is one of the key contributions of their study.

Kalayci et al. (2020) introduced a hybrid metaheuristic algorithm, which combines the continuous Ant Colony Optimization (ACO) (Dorigo et al., 2006) with GA and ABC algorithms, for addressing the cardinality-constrained portfolio optimization problem. The proposed approach includes the elitism mechanism from GA, the modification rate from ABC, and the Gaussian formulation from ACO. The authors evaluate their approach on seven publicly available benchmarks and show that it is efficient and competitive with state-of-the-art methods.

Chen et al. (2021) compared the performance of PSO and Particle Swarm with Center of Mass Optimization (PSOCM) (Jamous et al., 2015) against the Sequential Least Squares Programming (SLSQP) method. The authors optimized both the Sharpe and Sortino ratios. In some cases, PSO and PSOCM outperformed SLSQP in terms of speed and achieved better performance, especially for the Sortino ratio. Jin et al. (2019) examined a multi-period, tri-objective portfolio optimization problem where asset returns are treated as uncertain variables. Using uncertainty theory, the authors proposed a portfolio model that accounts for loss-averse utility, liquidity risk, and diversification. The authors used a Self-Adaptive Particle Swarm Optimization (SAPSO) algorithm to address this problem, including a self-adaptive stochastic ranking mechanism to balance exploration and exploitation. Mishra et al. (2021) proposed the LASSO–TLBO–SVR, a hybrid model combining LASSO, the Teaching Learning-Based Optimization algorithm (TLBO), and the Support Vector Regression (SVR) model for optimal portfolio selection. At first, the LASSO method is used for feature selection, followed by an optimized SVR via TLBO, ultimately choosing the best stocks. To optimize the stock's weight, the authors employed the Generalized Reduced Gradient (GRG) Nonlinear technique, GA, and the Equal Weight technique. Of the three methodologies, the GA provided better results. The beetle antennae search algorithm (BAS) is inspired by the longhorn beetles (Jiang & Li, 2017) and mimics the functioning of antennae and the random walking of beetles in the nature. BAS has been widely studied in the literature for the concept of portfolio optimization. The Non-linear Activated Beetle Antennae Search (NABAS) introduced by Khan et al. (2015) addresses the non-convex tax-aware portfolio optimization challenge. Experiments using the NASDAQ stock data showed its better performance in comparison to classical algorithms like BAS, PSO, and GA. Over the years, numerous BAS modifications have been proposed, especially for the task of portfolio optimization. (Medvedeva et al., 2021; Mourtas & Katsikis, 2021).

Metaheuristic algorithms are extensively employed in finance and portfolio optimization (Di Tollo & Roli, 2008; Grishina et al., 2017; Saiz et al., 2022; Erwin & Engelbrecht, 2023), with numerous additional significant studies available. Nevertheless, a substantial body of literature is available to provide a comprehensive and insightful evaluation of the existing research. Doering et al. (2019) investigated the status of metaheuristic algorithms in portfolio optimization and risk management, noting a rise in publications in this domain in recent years. A significant conclusion is the dominance of population-based algorithms and the tendency towards the development of hybrid algorithms. The authors agree with the NFL theorem, since no single algorithm is universally successful. Over the years, numerous metaheuristic algorithms have been applied to the portfolio optimization problem. However, to the best of our knowledge, there is a gap in the literature that provides an extensive comparison of these algorithms epsecially for the task of portfolio optimization.

2.6 Stock Price Forecasting

The extant literature contains several studies with the goal of developing models for stock price prediction. This section initially addresses the applications of traditional Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks. Subsequently, we present an overview of existing works that explores the metaheuristic-optimized models for this task.

2.6.1 LSTM and GRU for stock price forecasting

Various NN models have been examined for precise stock price prediction. In recent years, LSTM and GRU have showed better performance over traditional machine learning models for this task. Di Persio & Honchar (2017) evaluated several RNN architectures, in predicting the movements of Google stock price across different time horizons. The results showed that LSTM can handles long sequences effectively and shows better forecasting accuracy. The performance of LSTM networks has been extensively evaluated in the literature, and numerous studies report that that LSTM newtworks are able to outperform other NN and machine learning models (Fischer and Krauss, 2018; Nabipour et al., 2020).On the contrary, the work of Dutta et al. (2020) shows that that GRU with recurrent dropout had better performance relative to the others RNN architectures, for predicting the price of Bitcoin, including both exogenous and endogenous features. The authors demonstrated that basic trading techniques integrated with a GRU model can generate financial profits.The ability of GRU to surpass LSTM was further shown by Hamayel and Woda (2021). Nonetheless, existing literature presents instances that highlight the application of dimensionality reduction techniques such as LASSO and PCA, where LSTM and GRU show comparable performance in stock price prediction (Gao et al., 2021).

Numerous researchers integrate sentiment analysis with LSTM networks, demonstrating promising performance. Liu et al. (2021) integrates the LSTM model with data collected via online social networks. The authors utilized data from EastyMoney and created daily social networks. In addition to standard variables, the authors used the network variable for each stock as input for LSTM networks. The findings showed that the proposed approach improved the forecasting accuracy for the closing prices of SSE 50 constituent stocks. Wu et al. (2021) proposed the S_I_LSTM, which combines numerous sources of data and investor sentiment analysis to improve forecasting accuracy. At first the authors handle non-traditional data extracting and processing information, including market data, technical indicators, and resources such as posts and news articles. Then, a Convolutional Neural Network (CNN) is utilized to generate the investor index . Finally, they employed the the relevant index plus historical data and technical indications to forecast stock prices using LSTM. The suggested model, evaluated in the China Shanghai A-share market, and the results showed that S_I_LSTM achieved better results compared to classical methods.

Fang et al. (2024) proposed the VIX-Lasso-GRU approach for forecasting China's intraday stock index futures. This approach improves the performance of a GRU network by combining VIX to minimize prediction errors, and Lasso method to speed up the training procedure. The VIX-Lasso-GRU achieved better performance in terms of speed and accuracy in comparison to classical GRU and LSTM networks. Liu et al. (2024) proposed the LSTM-GRU-SA-AM model for predicting stock prices of new energy vehicle (NEV) businesses combining data from multiple sources. The model combines LSTM, GRU, Self-Attention (SA), and an Attention Mechanism (AM) to improve accuracy. The authors evaluated their approach based on data of four Chinese NEV equites and compare it with several baseline models. The findings show that multi-source data can improve accuracy, and their proposed model outperforms all alternatives. Jiang (2021) presents an extensive analysis of the applications of deep learning models in stock market forecasting, classifying several NN topologies, data sources, and assessment criteria, while also discussing issues regarding the implementation and repeatability.

2.6.2 Metaheuristic optimized models

Göçken et al. (2016) evaluate the importance of technical indicators in predicting the Turkish stock market with optimized NN models. The methodology combines HS and GA metaheuristic algorithms to select the most important indicators and optimize the number of hidden units. The models are evaluated by loss functions, investment returns, and graphical analysis, revealing that the HS-NN surpasses the other models in terms of prediction. Chung & Shin (2018) suggested a hybrid methodology that integrates LSTM networks with GA to enhance predictive accuracy. The model is assessed using daily data from the Korea Stock Price Index (KOSPI), and the findings demonstrate that the GA-LSTM model substantially surpasses baseline models. Shahvaroughi Farahani & Razavi Hajiagha (2021) suggested a metaheuristic-optimized NN model strategy utilizing the Social Spider Algorithm (SSA) and the Bat Algorithm (BA). Furthermore, the authors utilized twenty technical indicators and subsequently implemented GA for feature selection. Additionally, they evaluated the efficacy of metaheuristic-optimized NN models against conventional ARMA and ARIMA models, employing a dataset comprising the five most significant worldwide indices. The findings indicated that metaheuristic-optimized NNs yield superior predicting quality.

Ji et al. (2021) introduced the IPSO-LSTM, an enhanced LSTM model with the Improved Particle Swarm Optimization (IPSO) algorithm. The suggested IPSO incorporates an adaptive mutation factor to prevent premature convergence, and the authors introduced a non-linear method to enhance the inertia weight. The suggested approach was assessed using the Australian stock market index. The findings indicated that IPSO-LSTM exhibited superior accuracy compared to SVR, LSTM, and optimized LSTM via PSO. Ribeiro et al. (2021) introduced the HAR-PSO-ESN model, which integrates the design of the heterogeneous autoregressive (HAR) model, the predictive capabilities of the echo state network (ESN), and the usage of PSO for hyperparameter optimization. The authors evaluated the proposed method for predicting daily realized volatilities of three Nasdaq equities, revealing that HAR-PSO-ESN achieves better accuracy in the majority of cases, in contrast to benchmark models.

Chou et al. (2022) introduced a candlestick forecasting model that employs a metaheuristic-optimized Multi-output Least Squares Support Vector Regression (MLSSVR) model. For optimization purposes, the forensic-based investigation (FBI), symbiotic organisms search (SOS), and TLBO were applied. The results indicate that FBI-MLSSVR surpasses the other models regarding forecasting accuracy.

Kumar et al. (2022a) developed an optimized LSTM combined with an adaptive PSO hybrid model for forecasting the prices of three key stock indices. The authors contrasted the suggested model with GA-LSTM, Elman Neural Network (ENN), and regular LSTM, revealing the model's superiority. Kumar et al. (2022b) present a hybrid feed-forward neural network that integrates discrete particle swarm optimization (DPSO), particle swarm optimization (PSO), and the Levenberg–Marquardt (LM) algorithm for forecasting stock prices. DPSO identifies appropriate features and hidden layer neurons, whereas PSO adjusts weights and biases with precision. The model was evaluated on principal indices and juxtaposed with a genetic algorithm-based methodology, demonstrating its promising efficacy.

Gülmez (2023) suggested an enhanced LSTM model with the Artificial Rabbit Optimizer (ARO) for predicting stocks within the Dow Jones Industrial Average (DJIA) index. The proposed method surpassed ANN, conventional LSTM models, and a GA-optimized LSTM (GA-LSTM). Mu et al. (2023) introduced the MS-SSA-LSTM model, which combines multi-source data, sentiment analysis, the Sparrow Search Algorithm (SSA), and an LSTM network. model at first, creates a sentiment index from data from the EastyMoney forum. The SSA was used for hyperparameter optimization for refining the hyperparameters of LSTM. The proposed method surpassed several classical and optimized models in terms of performance when the authors evaluated their efficacy across all A-share equities in China's market.

Yu et al. (2023) emphasize the significance of the realized volatility (RV) of the stock price index. The authors proposed the GVMD-Q-DBN-LSTM-GRU approach for this purpose. The developed hybrid approach integrates an enhanced Variation Model Decomposition (VMD) utilizing the GWO algorithm to obtain the Intrinsic Mode Functions (IMFs). The same IMF is anticipated simultaneously with Deep Belief Network (DBN), LSTM, and GRU. Finally, the Q-learning method is employed to determine the ideal weights for the three previously described models. The authors employed the RV for three distinct indicators. The suggested model has superior performance compared to other NN models. Xiaopeng (2024) introduced an optimized GRU via the Aquila optimizer (AO) algorithm, the AO-GRU model, for predicting stock prices. The AO-GRU performed better than the classical GRU model, an optimized GRU via Ant Lion Optimizer (ALO-GRU), and GWO-GRU.

The extant literature points out the ability of machine-learning models in managing financial time series, in comparison to classical statistical methods. Several studies have employed metaheuristic algorithms for hyperparameter optimization of a specific model. Rather than optimizing one specific model, this study optimizes a deep learning architecture that can incorporate LSTM, GRU, or both. Since prior studies report differing performance between LSTM and GRU, we believe that optimizing the network's architecture can improve predictive accuracy. Furthermore, although the Golden Jackal Optimizer (GJO) (Chopra & Ansari, 2022) has shown promising performance, its application in financial time series is still limited.

2.7 Discussion

The findings of the previous subsections show that the application of AI and metaheuristic optimization algorithms has gained significant popularity over the years in the field of finance, as these algorithms can address key challenges in the financial sector. Regarding churn prediction, we observed several informative works that employ various machine-learning models. However, most works use classical machine-learning models and rarely discuss the performance of deep-learning models. In addition, although several activation functions and optimizers have been published in the extant literature, we observe that a comparative analysis that discusses the most recent components is missing.

As for credit-card default, we further observe that several models including both classical machine learning and metaheuristic-optimized models have been proposed. However, in the majority of cases the approaches involve complex models for predicting credit-card default. Given LightGBM's ability to handle imbalanced and high-dimensional data, we observe that its performance has not been studied for this purpose. Moreover, the application of the PSS algorithm has been studied only to a limited extent. Although the extant literature shows that hybrid approaches for tackling dimensionality and class imbalance are beneficial, we question whether there is a simpler approach in which hyperparameter optimization of a single model achieves better performance, potentially reducing the need for elaborate, time-heavy modelling pipelines.

Regarding investment decision-making and portfolio optimization, we observed that several works employ metaheuristic optimization algorithms and that these approaches can yield good results. However, given the large number of algorithms published over recent years (Rajwar et al., 2023), we observe that relatively few have been thoroughly evaluated consistently, and they are generally benchmarked against only a few classical metaheuristic algorithms. Although there are informative general-purpose works that consistently compare various algorithms on benchmark functions (Agushaka et al., 2023; Si-Ma et al., 2025), there is a lack of studies evaluating multiple algorithms across different numbers of epochs and population sizes specifically for portfolio optimization. In addition, several algorithms have limited application in the extant literature, especially in portfolio optimization.

Finally, the literature on stock-price prediction reveals substantial progress over the years, evaluating the performance of classical methods, AI models, and metaheuristic-optimized models. However, as the findings of Subsection 2.6.1 show, there are cases in which LSTMs perform better and others in which GRUs outperform LSTMs. This suggests that optimizing the core architecture of a deep-learning model for each time series is important. In addition, although the relatively new Golden Jackal Optimizer has shown promise, it has not been evaluated for stock-price prediction. To conclude, despite the considerable progress in the existing literature, further examination is needed, especially of the most recent advancements in AI and metaheuristic optimization algorithms for the financial sector.

3. Methodology

In the following subsections, we present the methodology for each application examined in this thesis. As noted earlier, our goal is to evaluate the performance of AI and metaheuristic-optimized methods for key challenges in the financial sector. We begin by discussing the application of neural networks and imbalanced learning, highlighting the main components of our analysis aimed at improving churn-prediction accuracy. Next, we propose an optimized LightGBM model using the PSS algorithm for credit-card default prediction. Given LightGBM's capabilities, which has been also explored in the extant literature, we believe that proper hyperparameter optimization can yield superior results. We then describe the methodology for investment decision-making applications. For portfolio optimization, we present the metaheuristic algorithms evaluated in this study and provide an extended discussion of Artificial Gorilla Troops (the proposed algorithm), along with the formal optimization problem. Finally, we propose an optimized deep-learning architecture tuned via the GJO algorithm, noting that both LSTM and GRU can achieve state-of-the-art performance in different scenarios. Motivated by GJO's promising results in prior work, we consider GJO-GRU and GJO-LSTM models and compare them to appropriate baselines. Across all four applications, our aim is to evaluate—or introduce—algorithms that have not been thoroughly examined in the literature for each topic, in order to identify approaches that offer strong predictive performance.

3.1 Imbalanced Learning and Deep Neural Networks for Churn Prediction

3.1.1 Class imbalance and resampling methods

A balanced dataset is important when developing an accurate machine learning model. However, real-world datasets are usually imbalanced. When the distribution of data is significantly skewed towards a specific class, models can be biased. Thus, these models often forecast more accurate the majority class due to its higher frequency, leading to poor predictive performance for the minority class. Sampling methods have been designed to tackle these limitations. Oversampling methods add artificially generated data in the minority class and undersampling approaches remove some data from the majority class. Although oversampling improves representation of minority class in the training set and avoids information loss, some studies state that this approach may lead to longer training duration and overfitting issues (Paula et al., 2015; Fernández et al., 2018). In addition, synthetic examples might not accurately represent the minority class (Tarawneh et al., 2022). On the other hands, undersampling techniques may remove information that's important for the model (Sun et al., 2015; Vuttipittayamongkol & Elyan, 2020).Thus, numerous researchers have proposed the integration of oversampling and undersampling methods to handle imbalanced datasets (Estabrooks et al., 2004).

Except class imbalance, real-world datasets may include both noise and overlapping samples(López et al., 2013; Fernández et al., 2018). Noise refers to erroneous data points that might affect the model's performance, while overlapping indicates that samples from both classes lie close together in feature space, making it hard to draw a clear decision boundary and leading to more misclassifications. In overlapping regions, the generation of synthetic data can worsen the overlap, making the decision boundary less clear for the model. Moreover, if the minority class includes noisy samples, oversampling methods increase the noise which also may leads to overfitting. In an alternative case where the majority class includes noisy data, undersampling methods may remove useful, and not noisy data points. Moreover, in cases with data overlap, removing too many samples from the majority-class may give the false impression that some regions are dominated by the minority class

It is important to note these methods should be applied only on the training set (Burez & Van den Poel, 2009; Ghorbani & Ghousi, 2020; Kimura, 2022; Imani & Arabnia, 2023; Haddadi et al., 2024) as they affect the class distribution of the data, potentially affecting the model's performance evaluation. Employing resampled data in the test set gives an overly positive evaluation of model performance and produces inaccurate results.

SMOTE (Chawla et al., 2002) is a notable technique that is extensively utilized and has influenced other oversampling methods or serves as a key component in hybrid resampling strategies. Unlike random oversampling, which just replicates existing data points, SMOTE creates unique samples from the minority class by utilizing the (often Euclidean) distance between each data point. These synthetic samples increase the number of samples from the minority class, which makes the dataset more balanced.

SMOTE-Tomek Links is a hybrid approach, which combines SMOTE with Tomek Links (Tomek, 1976) undersampling method. Tomek Links trims the dataset by removing samples from the majority class, that share similar characteristics with data from the minority class. These pairs which are called Tomek Links, represent misclassified samples between the two classes, which results in added noise. SMOTE-Tomek Links aims to create a more clear decision boundary by eliminating noisy samples and balancing the remaining dataset. At first, SMOTE oversamples the minority class, after which the Tomek Links algorithm is employed to eliminate overly close or overlapping synthetic samples

SMOTE-ENN (Batista et al., 2004), a variant of the traditional SMOTE approach, integrates SMOTE with ENN (Wilson, 1972), a modification of the KNN algorithm. The basic idea of KNN is that the closest neighbor of the data, in terms of distance, belongs to a corresponding class. The ENN methodology aims at finding and removing samples from both the majority and minority classes that are at risk of misclassification, reducing noise and eliminating data overlap within the dataset. The procedure starts with SMOTE, followed by the application of ENN for the objective of undersampling, similar to SMOTE-Tomek Links.

3.1.2 Activation Functions

The selection of an appropriate activation function is important for the performance of a deep learning model. Without including an activation function and only using the weights and biases, each neuron would only conduct a linear transformation of the inputs. While linear transformations simplify NN, they are ineffective to identify complex patterns inside data. The most famous activation function in the concept of NN models, is the Rectified Linear Unit (ReLU), introduced by Nair and Hinton in 2010. ReLU is computationally efficient as it necessitates only a threshold operation. The piecewise linear design allows faster convergence during the training process and reduces the likelihood of vanishing gradients. However, ReLU has limitations since it faces the "dying ReLU" phenomenon (Lu et al., 2019), where certain neurons become inactive and fail to contribute during the learning process due to their zero output for negative inputs.

Leaky ReLU (LReLU) (Maas et al., 2013) is a modification of the ReLU that adds a minor, non-zero gradient for the negative inputs. LReLU ensures the activity of all neurons and eliminates the "dying ReLU" problem. Exponential Linear Units (ELU) (Clevert et al., 2015) add an exponential term for the negative inputs, which promotes gradient smoothing and improves the learning process. The ELU activation function reduces the "dying ReLU" problem and centers the mean of the activations closer to zero, which improves the training process by minimizing bias variation.

Swish (Ramachandran et al., 2017) is a self-gated activation function which can outperform ReLU in several scenarios. The sigmoid function used for gating provides a smooth transition between linear and nonlinear behaviors. Swish is a good choice for gradient-based optimization techniques because of its smooth and differentiable characteristics. The value of β plays and important role for the function's execution. β is most often set to 1, promoting simplicity and computational efficiency. In this case, Swish is equal to the sigmoid-weighted linear unit (SILU) (Elfwing et al., 2017).

Misra (2019) introduced the Mish activation function, which is similar to Swish. Mish is a smooth, non-monotonic, and continuously differentiable, which is able to handle issues like vanishing gradients and inactivity of neurons during training. Mish improves gradient flow, increasing learning efficiency in deep neural networks, especially in deeper architectures where vanishing gradients can slow the learning process. Mish showed better performance in comparison to other activation functions, like ReLU and Swish, in several deep learning benchmark task. Moreover, its self-regularizing behavior decreases overfitting, improving overall performance. APTx activation function (Kumar, 2022) behaves similarly to Mish while being more computationally efficient. APTx requires fewer mathematical operations during forward and backward propagation. This fact, enables APTx to train NNs faster and provide fast, reliable inference on modest compute resources,making it a great fit for NN running on Internet of Things (IoT) edge devices.

This study compares APTx with ReLU, LReLU, ELU, Swish, and Mish. Figure 21 shows the line graphs of the activation functions. Due to the same behavior of the curves, we present a zoomed-in view in Figure 22 for clarity. Figure 23 illustrates the corresponding derivatives curves. We provide each function along with its respective derivative below.

- $ReLU(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases}$

- $ReLU'(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases}$

- $LReLU(x) = \begin{cases} \alpha x, & x < 0 \\ x, & x \geq 0 \end{cases}$

- $\text{L}ReLU'(x) = \begin{cases} \alpha, & x < 0 \\ 1, & x \geq 0 \end{cases}$

- $ELU(x) = \begin{cases} \alpha(e^{x} - 1), & x \leq 0 \\ x, & x > 0 \end{cases}$

- $\text{ELU}'(x) = \begin{cases} \alpha(e^{x}), & x \leq 0 \\ 1, & x \geq 0 \end{cases}$

- $Swish(x) = x * sigmoid(\beta\chi)$

- $Swish'(x) = \frac{e^{x}(x+e^{x}+1)}{(e^{x}+1)^{2}}$ $\beta = 1$

- $Mish(x) = x * tanh(softplus(x))$

- $Mish'(x) = \frac{e^{x}(4(x+1)+4e^{2x}+e^{3x}+e^{x}(4x+6))}{(2e^{x}+e^{2x}+2)^{2}}$

- $APTx(x) = (a + \tanh(\beta\chi)) * \gamma\chi$

- $APTx'(x) = \frac{e^{2x}(1 + e^{2x} + 2x)}{(1 + e^{2x})^{2}}$ α=1,β=1,γ=1/2

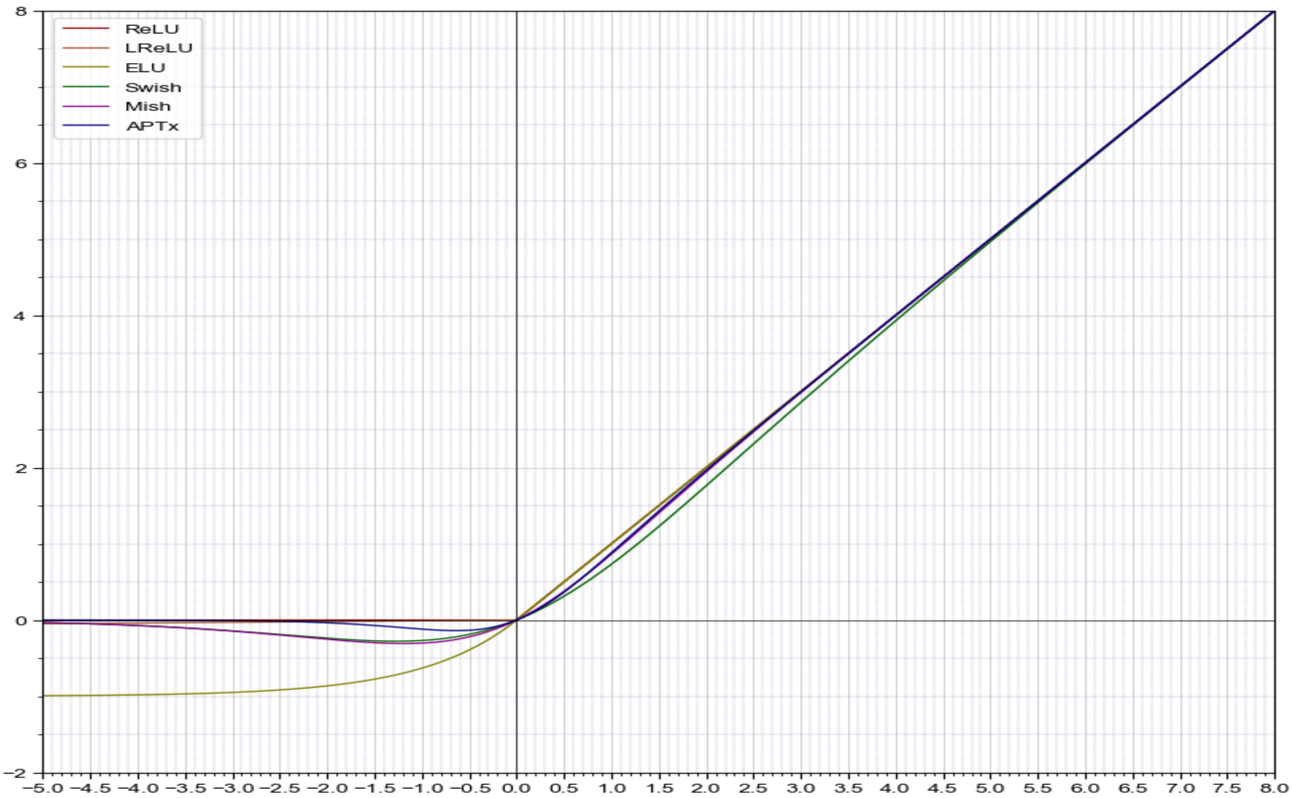


Figure 21. Graphical representation of the activation functions

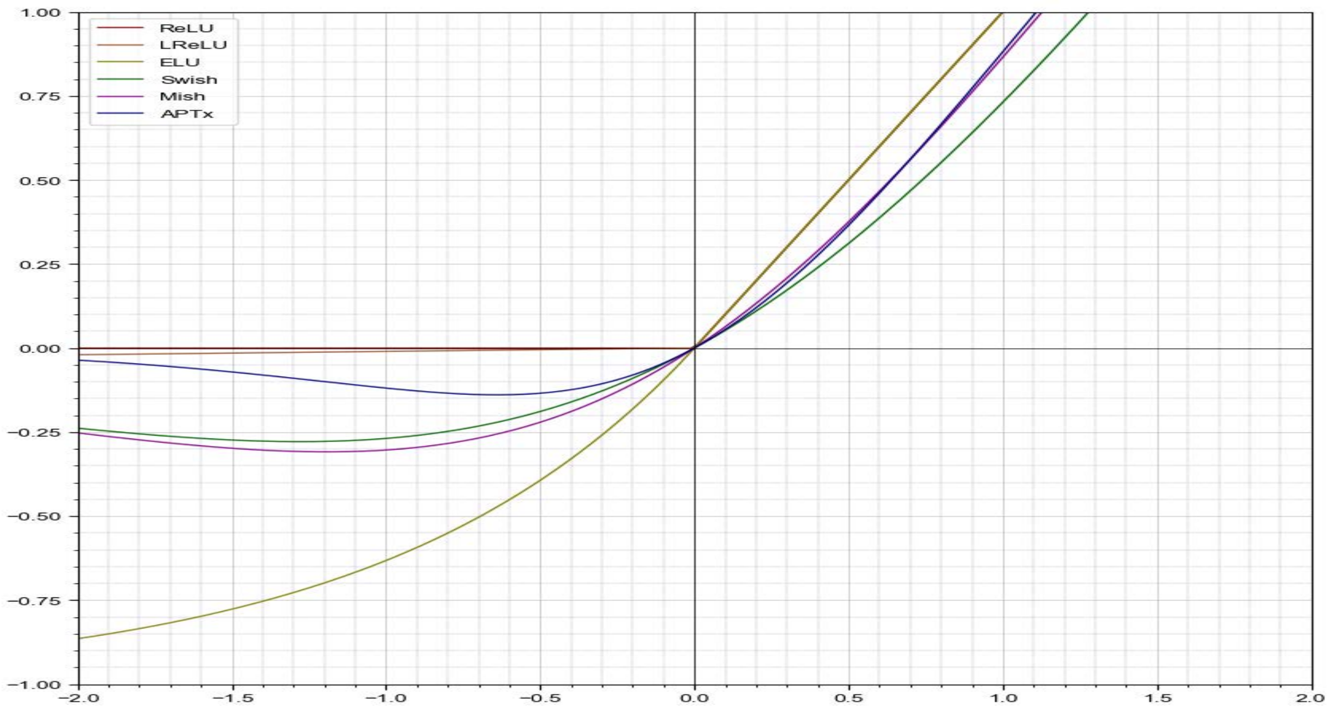


Figure 22. Graphical representation of the activation functions with a focus on the axis origin

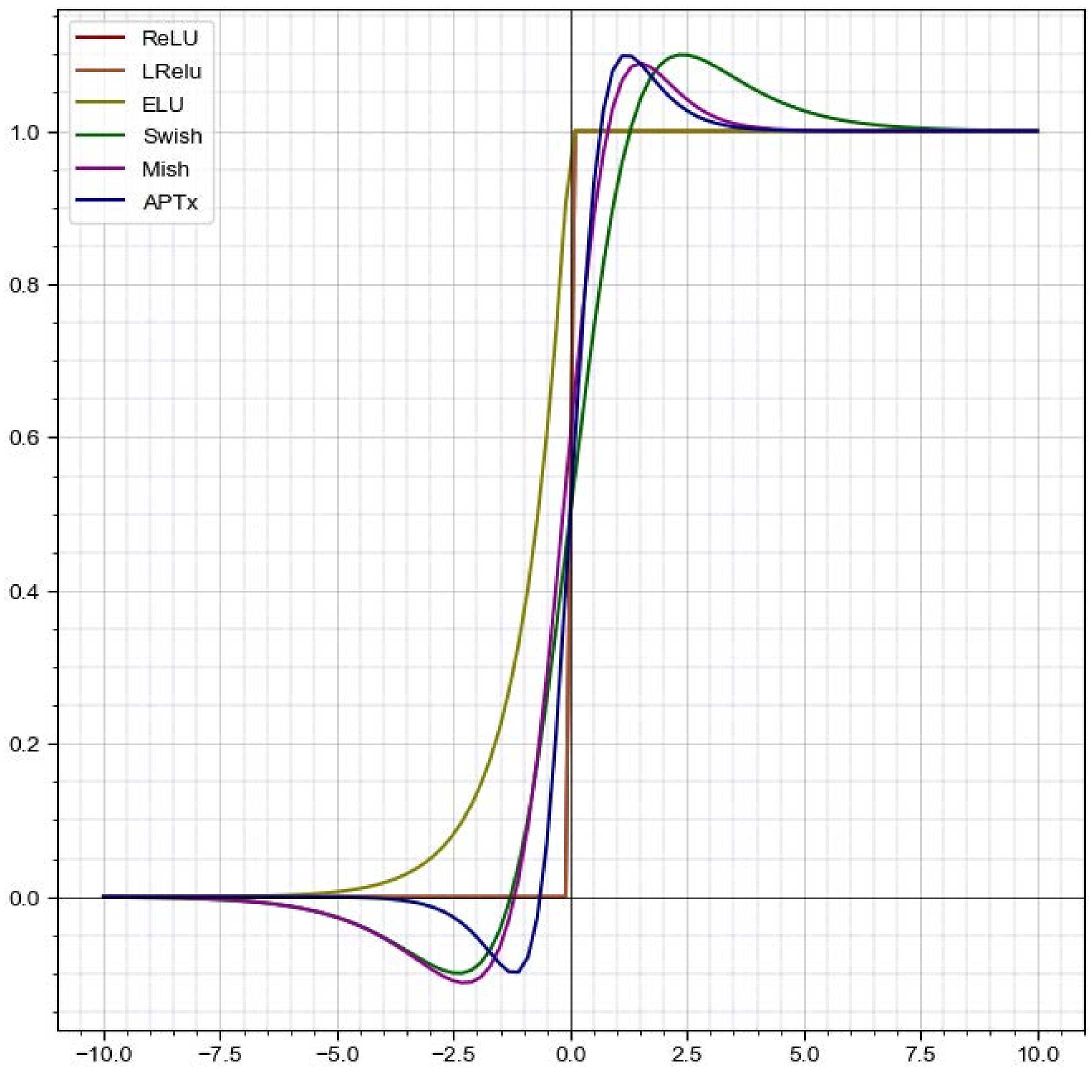


Figure 23. Graphical illustration of the activation functions' derivatives

3.1.3 Neural Network Optimization

The selection of appropriate optimization algorithm is another important component that affects the performance of a NN model. The selection of optimizer used during training is vital for determining the speed of the training process and the overall predictive effectiveness of the model. Stochastic Gradient Descent (SGD), a well-known optimizer, updates the model's weights by approximating the true gradient with a mini-batch, hence enabling faster convergence compared to the classical gradient descent approach. Various modifications of SGD, such as Nesterov accelerated gradient (Nesterov, 1983), use approaches to improve convergence speed and escape local minima. Adaptive optimization algorithm, such as the Adaptive Gradient Algorithm (Adagrad) (Duchi et al., 2011), Root Mean Square Propagation (RMSProp) (Tieleman & Hinton, 2012), and Adaptive Moment Estimation (Adam) (Kingma & Ba, 2014), update the learning rate for each weight based on the previous gradients. Adagrad adjusts the learning rate to be increased for irregular parameters and reduced for frequent ones.

During the training phase, the learning rates decrease as squared gradients accumulate, leading the algorithm unable to collect further information. RMSProp is aimed to overcome the declining learning rates problem of Adagrad. RMSProp preserves a moving average of the squared gradients, so avoiding a fast decrease in the learning rate. It uses an exponential moving average of squared gradients to shrink the learning rate, making weight updates more balanced. RMSProp makes optimization smoother and helps the model converge faster

Adam applies different moving averages for gradients and squared gradients for each weight, utilizing advantages of both methods. This information is then utilized to bias-correct the estimations and adaptively modify the training learning rate for each weight. Adam's strength with handling sparse gradients and noisy data is beneficial, making it highly suitable for numerous applications. Nonetheless, despite its popularity, Adam has been criticized for instability during convergence, especially if the learning rate is poorly chosen or the model shows high variance.

This limitation motivated the development of RAdam (Liu et al., 2019), that addresses these issues. RAdam improves the Adam optimizer by adding a rectification mechanism that controls the variation of the adaptive learning rate by predicting the duration of the "warm-up" phase, where in this phase the adaptive learning rate shows high variance. RAdam improves training stability by updating the learning rate based on the observed variance, making this approach less sensitive to the initial learning rate and more stable during convergence. In the extant literature, RAdam has showed improved convergence and generalization performance compared to Adam, especially when learning rates are poorly tuned or the model suffers from high variance. Various scholars (Cui et al., 2020; Sun et al., 2020; Mazumder et al., 2022; Ko et al.,2024) have investigated the performance of RAdam across several tasks, showing Radam's promising performance.

3.2 Optimized LightGBM via Pareto-Like Sequential Sampling algorithm for Credit Card Default Prediction

3.2.1 Pareto-Like Sequential Sampling

Shaqfa & Beyer (2021) introduced the Pareto-like Sequential Sampling (PSS) metaheuristic algorithm, which emulates the *80/20* rule, commonly referred to as the Pareto principle (Pareto, 1897). The optimization technique in PSS utilizes a dual sampling approach. The algorithm primarily concentrates on a certain area of the search domain, known as the current prominent region. The location is selected based on its previous track record of positive results, aiming to carry out a more targeted inquiry that could perhaps identify superior solutions that align well with the already established beneficial ones. Additionally, to provide a comprehensive examination, the algorithm simultaneously generates samples from the broader search domain. By utilizing a well-rounded approach, potential solutions outside of the main area of concentration are not disregarded, since the ability to explore is maintained while the most productive areas are utilized, allowing an efficient trade-off between exploration and exploitation.

This approach is based on the assumption that the current optimal solution vector x_{best} is located within a prominent sub-domain Ω'.Therefore, PSS selects the majority, approximately α × 100% of the features from the nearby vicinity. This approach is similar to PSO's g_{best} parameter. The size of the prominent domain is contingent upon the algorithm's remaining iterations, and the analogy of (1 − α). The maximum size of the prominent zone is determined by a value of (1 − α) % of Ω, with its center located at x_{best}. The acceptance probability α is used to sample the solution components from the prominent neighborhood. Approaching the maximum number of iterations diminishes the opportunity for modifications, consequently leading to a reduction in the size of the prominent zone. The PSS algorithm is utilized by increasing the density of sampling for the design variables of solutions inside a narrowed search area, namely the current prominent region.

Mathematical Formulation of PSS

Let's consider the optimization Problem $f(x)$ is defined on search domain $\Omega \in \mathbb{R}^{\mathbf{n}}$ and for all subdomains $\Omega' \subset \Omega$ and the corresponding boundaries Γ and Γ' then, the optimization problem is defined by Equation 1:

$$\textbf{Optimize } f(x), x \in \Omega, \Omega \in \mathbb{R}^{\mathbf{n}}, \textit{ where} \qquad (1)$$

$$\mathbf{x} = \{\mathbf{x_1} \ldots\ldots, \mathbf{x_n}\}, \boldsymbol{\Gamma}^- \leq \boldsymbol{x} \leq \boldsymbol{\Gamma}^+$$

Where $[\boldsymbol{\Gamma}^-, \boldsymbol{\Gamma}^+]$ corresponds to lower and upper bounds for Ω .For Ω' the lower and upper bounds are defined as $[{}^{\boldsymbol{i}}\boldsymbol{\Gamma}'^-, {}^{\boldsymbol{i}}\boldsymbol{\Gamma}'^+]$.Moreover let $\boldsymbol{x_{opt}}$ reclines in $\Omega'$and minimizes $f(.)$, where $f(\boldsymbol{x_{opt}}) \leq f(x) \forall\, \boldsymbol{x} \in \Omega \textit{ is a global solution}$. Furthermore, i corresponds to current iteration, k is the index of the solution vector. In the population matrix and j indicates the j^{th} design variable feature x_j of any solution vector x

The initial population sampling for a continuous landscape composes by Equation 2:

$$k^x = \boldsymbol{\Gamma}^- + {}_k\boldsymbol{u} \odot (\boldsymbol{\Gamma}^+ - \boldsymbol{\Gamma}^-) \qquad (2)$$

In the above equation the ${}_k\boldsymbol{u}$ is a vector of random coefficients sampled using one of the classical DOE methods

The matrix $\boldsymbol{u}^i$ (Equation 3) is sampled at each iteration with chosen DOE method.

$$[u^i] = \begin{bmatrix} {}_1u_1^i & {}_1u_2^i & \cdots & {}_1u_n^i \\ {}_2u_1^i & {}_2u_2^i & \cdots & {}_2u_n^i \\ \vdots & \vdots & \ddots & \vdots \\ {}_\beta u_;^i & {}_\beta u_2^i & \cdots & {}_\beta u_n^i \end{bmatrix} \quad (3)$$

Once the algorithm has assessed the initial population and identified the optimal solution x_{best}^i , it proceeds to compute the domain-tightening coefficients. These coefficients are utilized to revise the existing upper and lower limits for each design feature, as illustrated by Equations. (4,5):

$$^{i}\boldsymbol{\Gamma}'^{+} = x_{best}^i + \eta^i, \ ^{i}\boldsymbol{\Gamma}'^{+} \leq \boldsymbol{\Gamma}^{+} \quad (4)$$

$$^{i}\boldsymbol{\Gamma}'^{-} = x_{best}^i + \eta^i, \ ^{i}\boldsymbol{\Gamma}'^{+} \leq \boldsymbol{\Gamma}^{+} \quad (5)$$

$$\eta^{\boldsymbol{i}} = \frac{(\mathbf{1}-\boldsymbol{\alpha})\left(\mathbf{1}-\frac{\boldsymbol{i}}{\gamma}\right)}{\mathbf{2}}(\boldsymbol{\Gamma}^{+} - \boldsymbol{\Gamma}^{-}) \quad (6)$$

The bandwidth $\eta^{\boldsymbol{i}}$ is updated every time x_{best}^i is updated. The PSS applies time-dependent bandwidth tightening using the term $\left(\mathbf{1}-\frac{\boldsymbol{i}}{\gamma}\right)$ which controls the greediness of the algorithm. The (1-a) term is used to maintain the $\boldsymbol{a}/(\mathbf{1}-\boldsymbol{a})$ ratio which resembles the Pareto “80/20” analogy. In order to regulate the quantity of features sampled per solution from the salient domain Ω, the acceptance probability α has been used. When the algorithm opts to utilize the prominent domain Ω', the feature is generated using Equations. (4-6) in conjunction with Equation. (7).

$$_{k}x_j^i = {}_j^i\boldsymbol{\Gamma}'^{-} + {}_k u_j^i ({}_j^i\boldsymbol{\Gamma}'^{+} - {}_j^i\boldsymbol{\Gamma}'^{-}) \quad (7)$$

In the alternative scenario, where the entire domain Ω is selected then:

$$_{k}x_j^i = {}_j\boldsymbol{\Gamma}^{-} + {}_k u_j^i ({}_j\boldsymbol{\Gamma}'^{+} - {}_j\boldsymbol{\Gamma}'^{-}) \quad (8)$$

The procedure keeps occurring, until the maximum number of iterations (γ), is reached. The entire pseudocode is presented in Appendix A

3.2.2 Halton & Scrambled Halton Sequence

PSS is compatible with any sampling method, such as Monte Carlo (MC) and Latin Hypercube Sampling (LHS), as indicated by the authors. In their work, they especially focus on assessing the algorithm's performance using the MC, and the outcomes showed promising results. The HS (Halton, 1960), an extension of van der Corput sequences, is a quasi-random, low-discrepancy number generator that is predominantly applied to multidimensional numerical integration problems. In applications where representative coverage of the parameter space is critical, such as complex integrations and simulations, HS can be advantageous. Although the HS offers several benefits, it faces specific difficulties when implemented in situations that involve extremely high dimensions. When this occurs, the sequence of variables may demonstrate correlation properties, which may introduce biases or diminish the effectiveness of the sampling procedure. (Bhat, 2003; Hess & Polak, 2003**).** Several researchers have proposed various algorithms to overcome the limitations of the standard Halton sequence **(**Hellekalek, 1984; Kocis & Whiten, 1997; Owen, 1995;2006;2017**)** by incorporating scrambling techniques to address the HS correlation properties. In this work for the implementation of SHS, we employ the algorithm proposed by Owen (2017).

3.2.3 LightGBM

Gradient Boosting Decision Trees (GBDT) are characterized by their sequential construction of DT, where each tree is trained to correct the errors of its predecessors (Friedman, 2001). This approach allows GBDT to capture complex nonlinear relationships within the data. However, when the quantity of samples is massive or the dimension of the features is high, GBDT cannot achieve adequate results and also faces computational complexity. This occurs since traditional GBDT's implementations require analyzing all data instances for each feature in order to calculate the information gain of each potential
split point. As a result, this computational challenge is related to the number of instances and features of data. LightGBM is a type of GBDT that employs two novel methodologies: Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB).

GOSS improves efficiency by directing computational resources towards more demanding segments of the data. The core idea behind this approach is that if an instance has a small gradient, this also indicates a small training error. Hence, GOSS prioritizes instances with larger gradients. However, discarding all these instances will affect data distribution, which probably also affects the forecasting ability of the trained model. GOSS not only retains instances with large gradients, but also randomly samples instances with small gradients. This approach allows LightGBM to focus primarily on the under-trained instances without significantly altering the data distribution. Multidimensional data is usually sparse; however, several features are mutually exclusive. Given this, EFB bundles these exclusive features into a single feature, which can reduce data dimensionality while retaining a significant quantity of information. LightGBM is able to handle massive datasets, ensuring optimal performance and memory efficiency. The authors conducted experimental analysis, and the results showed that LightGBM can outperform other ensemble models in a variety of scenarios.

3.2.4 Class Imbalance and High-Dimensional Data

As discussed in earlier subsections, class imbalance and high dimensionality pose persistent challenges for researchers. Various methods are typically used to address these issues and to build reliable, robust models. LightGBM mitigates them through built-in techniques. For categorical features, it applies Fisher's (1958) optimal-partitioning approach—ordering categories and searching for the best split—which often outperforms one-hot encoding. For class imbalance, LightGBM offers two options: *is_unbalance* (automatic reweighting) and *scale_pos_weight* (manual-class weighting). In this study, we use the *is_unbalance* option. Several prior studies have employed LightGBM's built-in parameters to tackle at least one of these issues (Jin *et al.*, 2020; Hancock & Khoshgoftaar,2021; Sun *et al.*,2023; Yilmaz *et al.*,2023; Daoud & Ben-Hur,2025; Yun & Cho,2025; Liu *et al.*,2020; Chen *et al.*, 2024)

3.3 Artificial Gorillas Troop For Portfolio Optimization

In this section, we begin with a brief introduction to the benchmark algorithms considered in this application. We then offer a detailed overview of AGTO and conclude with the formulation of the problem addressed.

3.3.1 Swarm Intelligence Algorithms

Swarm intelligence algorithms are inspired by the social structure and behaviour of insects and animals in nature. Eberhart and Kennedy introduced PSO, a well-known algorithm, drawing inspiration from the swarm behaviour and dynamic movements of birds and fish. Other established and popular swarm-based algorithms include the ABC, which models the cooperative behaviour of bees as they search for food. The Grasshopper Optimizer Algorithm (GOA) (Saremi et al., 2017) describes the swarming behaviour of grasshoppers while searching for food and avoiding predators. The Social Spider Algorithm (SSA) (James & Li, 2015) is inspired by the social behaviour of spiders, especially their approach for collaboration in the group. Spiders show complex behaviours such as communication through webs, collaborative hunting, and resource sharing. The Antlion Optimization algorithm (ALO) (Mirjalili, 2015) mimics the hunting strategy of antlions, which involves hunting and trapping ants in five different steps.

The GWO algorithm is inspired by the social structure and hunting behaviour of grey wolves, where the group works together to capture prey. The Coyote Optimization Algorithm (COA) (Pierezan & Coelho, 2018) is based on the social structure of coyotes in nature (Canis latrans). COA Instead of just focusing on hunting behaviour like GWO, it simulates how coyotes share information and adapt to their environment. On the contrary, the Artificial Rabbit Optimizer (ARO) (Wang et al., 2022) mimics rabbits' survival strategies in nature, simulating the detour foraging strategy and random hiding. The Aquila Optimizer (Abualigah et al., 2021) replicates the natural behaviour of aquilas by switching among four tactics to capture their prey. The Cuckoo Search (CS) (Yang & Deb, 2009), draws inspiration from the fact that some cuckoo species exhibit brood parasitism. The Harris Hawks Optimization (HHO) (Heidari et al., 2019) algorithm mimics Harris Hawks' grouped attacking behaviour, which employs the "surprise pounce" strategy to surprise their prey during hunting.

The humpback whales use the bubble-net strategy during hunting, which has inspired the Whale

Optimization Algorithm (WOA) (Mirjalili & Lewis, 2016). In the same vein, the hunting strategies of tunas, which include spiral and parabolic foraging, have inspired the Tuna Swarm Optimization Algorithm (TSO) (Xie et al., 2021). The Sailfish Optimization Algorithm (SOA) (Shadravan et al., 2019) is inspired by sailfish, the fastest fish in the ocean, and mimics its hunting behaviour. Another popular algorithm is Marine Predator Algorithms (MPA) (Faramarzi et al., 2020), a metaheuristic optimization algorithm inspired by the hunting strategies of marine predators, especially those at the top of the oceanic food chain, such as sharks, orcas, and other large predators.

3.3.2 Artificial Gorilla Troops Optimizer

This section offers an overview of AGTO, as it is the proposed metaheuristic algorithm for portfolio optimization. Gorillas are highly intelligent animals, which live in complex communities and show deep emotions and impressive cognitive skills. Gorillas live in groups with a dominant silverback male who leads the group, settles fights, protects the group, and preserves discipline. Blackbacks, which are younger male gorillas, help guard the troop even more and can take over as leader if the silverback dies. Female gorillas form strong bonds with silverbacks for protection and reproduction. Males often fight for dominance, but all-male groups can be steady and peaceful.

AGTO mimics these dynamics by structuring its search process around exploration and exploitation phases. During exploration, three operators enhance the algorithm’s ability to search diverse areas of the optimization space. In the exploitation phase, two mechanisms intensify the search around local optima. The algorithm simulates gorilla troop behaviour using three types of solutions: X (current position), GX (candidate positions), and the silverback (best solution), with only one silverback per population representing the optimal solution. If a GX outperforms its corresponding X, it replaces it; otherwise, it's stored in memory. Inspired by how gorillas seek better food sources and group positions, weaker solutions are moved away from while better ones are approached, allowing AGTO to effectively balance exploration and exploitation and adaptively navigate complex search spaces.

Exploration phase

Gorillas live in groups under the leadership of a silverback, whose commands are obeyed. However, gorillas may leave their group and migrate to different locations, known or unknown. In the AGTO algorithm, gorillas are considered candidate solutions, with the best candidate solution at each optimization operation stage representing the silverback.

Three mechanisms are used in the exploration phase: *migration to an unknown location*, *migration toward a known location*, and *moving toward other gorillas*. These mechanisms are selected based on a general procedure involving a parameter named $p \in [0,1]$ which must set before optimization procedure and a random number $rand \sim U(0,1)$. If rand < p the *migration to an unknown location* is selected. If r ≥ 0.5, the mechanism of *movement toward other gorillas* is selected, and if r < 0.5, migration to a known location is selected. Each mechanism provides specific benefits to the AGTO algorithm's performance. The first mechanism enables the algorithm to extensively explore the problem space, the second improves AGTO's exploration performance, and the third strengthens AGTO's ability to escape local optimal points (Equation 9).

$$GX(t+1) \begin{cases} (UB - LB) \times r_1 + LB & rand < p \\ (r_2 - C) \times X_r(t) + L \times H & rand \geq 0.5 \\ X(i) - L \times \left(L \times X(t) - GX_r(t)\right) + r_3 \times (X(t) - GX_r(t))), & rand < 0.5 \end{cases} \quad (9)$$

Where:

$$r_1, r_2, r_3 \sim U(0,1)$$

$$LB : Lower\ bound$$

$$UB : Upper\ bound$$

In Equation (9) $GX(t+1)$ denotes the gorilla candidate position in the next iteration and. GX_r is one randomly selected gorilla from the group. C (Equation 10) takes into account the current iteration (*It*), the overall value of iterations (*MaxIt*) to execute the process of optimization, and variable *F* (Equation 11).

$$C = F \times \left(1 - \frac{It}{MaxIt}\right) \quad (10)$$

$$F = cos(2 \times r_4) + 1, \quad r_4 \in [0,1] \quad (11)$$

As the authors have visually shown in Equation (10), early phases of the optimization process gives values with abrupt changes across a wide time frame; however, this interval of change gets smaller in the last stages.

Next, the L parameter (Equation 12) is calculated, which simulates the silverback's leadership. Gorillas lack adequate experience in the early phases of group leadership which may lead them to take poor decisions. However, silverbacks attain enough experience over the "iterations", which results in a strong and stable leadership.

$$L = C \times l, \quad l \sim U(-1,1) \qquad (12)$$

Finally, H and Z, which is a random variable, are calculated using Equation (13) and Equation (14) respectively

$$H = Z \times X(t) \qquad \mathbf{(13)}$$

$$Z = [-C, C] \qquad \mathbf{(14)}$$

Exploitation phase

The AGTO algorithm's exploitation phase involves two mechanisms. The selection of the two phases is determined by the values of C, and W parameter which should be set before the optimization process. The silverback gorilla leads the group, makes decisions, directs the gorillas toward food sources, and is responsible for their safety and well-being. All gorillas in the group follow the silverback's decisions. If $C \geq W$ then the *Follow the Silverback* (Equation 15) mechanism is selected:

$$GX(t+1) = L \times M + (X(t) - X_{\text{silverback}}) + X(t) \quad \mathbf{(15)}$$

$$M = \left(|\frac{1}{N} \sum_{i=1}^{N} GX_i(t)|^g \right)^{\frac{1}{g}} \qquad \mathbf{(16)}$$

$$g = 2^L \qquad \mathbf{(17)}$$

However, the silverback may weaken, get aged which will ultimately leading to death, other blackback male gorillas may become the group leader. Moreover, even if silverback is healthy, the blackbacks may exhibit signs of rebelization which leads to engage with the silverback and seek to the domination of the group. In such scenarios, the *competition for adult females* (Equation 18) mechanism is selected. The pseudocode of the algorithm, as outlined by Abdollahzadeh *et al.* (2021), is provided in the Appendix A.

$$GX(i) = X_{\text{silverback}} - (X_{\text{silverback}} \times Q - X(t) \times Q) \times A \ \mathbf{(18)}$$

$$Q = 2 \times r_5 - 1 \qquad \mathbf{(19)}$$

$$A = \beta \times E \qquad \mathbf{(20)}$$

$$E = \begin{cases} N_1, rand \geq 0.5 \\ N_2, rand \geq 0.5 \end{cases} \quad \textbf{(21)}$$

3.3.3 Constraint satisfaction and portfolio selection

According to the investor's preferences and strategy, a number of portfolio selection constraints can be considered. The budget constraint is the primary one, ensuring that the total weights of the portfolio's assets add to one. Risky portfolios can be categorized as restricted or unrestricted based on the presence or absence of certain constraints. Restricted portfolios do not allow short selling, keeping all weights non-negative, while unrestricted portfolios allow short selling. Sector constraints an play a key role, especially when the investor wants to construct a diversified portfolio. In this case, it is advisable to use weights as safeguards to enhance diversification and mitigate sector risk.

Moreover, cardinality constraints are important, since investors often prefer simpler portfolios by restricting the number of assets and reducing transaction costs. Such constraints transform the portfolio optimization problem into a mixed integer programming problem, which complicates the computational process of identifying the optimal solution. In alignment with portfolio diversity and simplification, an investor may consider incorporating floor and ceiling limits. These types of constraints could be considered as bounds during the optimization process. Maintaining a weight below 5% for several assets may be prohibitively expensive and impossible, while individual assets exceeding 50% would lead to excessive concentration and higher risk.Introduced by William Sharpe in 1966, the Sharpe ratio (Equation 22) remains a prevalent measure among investors to evaluate the performance of assets in relation to their risk. It computes the excess return of a portfolio over the risk-free rate per unit of risk, as determined by the standard deviation of the portfolio.

$$Sharpe\ Ratio = \frac{R_p - r_f}{\sigma_p} \quad (\mathbf{22})$$

where R_p is the expected portfolio return, r_f is the risk-free rate, and σ_p is the standard deviation of the portfolio's returns.

An investor aims to maximize their investment for a given level of risk according to the Sharpe ratio (Equation 23). In this work, we consider the case of restricted portfolio with budget constraint (Equation 24), which ensures that the total sum of the portfolio's asset weights equals one. Moreover, the portfolio is restricted from short selling (Dupačová & Kopa ,2014; Pagnoncelli et al.,2017; Jin et al.,2019; Mishra et al.,2021) thus, negative weights are not allowed. (Equation 25).

$$\text{Maximize}: \frac{R_p - r_f}{\sigma_p}, (\mathbf{23})$$

Subject to:

$$\sum_{i=1}^{N} w_i = 1 , \mathbf{(24)}$$

$$0 \leq w_i \leq \ 1, \mathbf{(25)}$$

where N is the total number of assets, and w_i corresponds to the i_{th} asset in the portfolio. One of the necessary restrictions for the specific portfolio optimization problem is that the sum of the weights must be equal to one. To achieve this, we use the normalization approach (Equation. 26) of Pai & Michel (2014), where each respective weight W_i in normalized as follows:

$$W_i^{(norm)} = \frac{(W_i - W_{min})}{\sum_{i=1}^{N}(W_i - W_{min})} \quad \mathbf{(26)}$$

3.4 Hybrid Deep Learning architectures via Golden Jackal Optimization algorithm stock price forecasting

This section describes the components of the proposed approach. At first, we the LSTM and GRU networks adopting the mathematical notation as established by Olah (2015). Finally, we provide an extensively description of GJO, as it is the proposed algorithm in this work.

3.4.1 LSTM

RNN is an established deep learning architecture that, unlike classical NN, integrates feedback loops to process sequential data processing, with a hidden state that preserves prior information. However, RNNs have challenges in managing long-term dependencies. This type of NN experiences the "vanishing gradient problem," wherein gradients may exhibit either exploding or diminishing behavior during backpropagation across time, complicating the network's ability to successfully maintain long-range memory (Hochreiter, 1992). Hochreiter & Schmidhuber (1997) introduced LSTM networks, which address the limitations of traditional RNNs. An LSTM network of comprises memory blocks interconnected between layers, with each block featuring gates that regulate information flow. A unit comprises cell states and three gate types, namely forget, input, and output. For each gate, the output calculation includes the sigmoid transformation of the current timestep x_t, the output of the previous timestep(h_{t-1}), together with the corresponding weight matrices and bias vectors.

The forget gate (f_t) (Gers *et al.*,2000) (Equation 27) decides the information to be eliminated from the cell state (c_t) (Equation 28) from the prior time step(c_{t-1}). Next, the current c_t is updated. To that end, the candidate cell state, candidate values ($\tilde{c}_t$) are calculated per Equation 29. The quantity of new information stored in the c_t is defined from the element-wise multiplication of $\tilde{c}_t$ with the output of input gate (i_t) (Equation 30).As indicated in Equation 29 the current c_t consists of data from the prior timestep and the new information. Finally, the current hidden state (h_t) (Equation 31) is computed as the element-wise multiplication of the output gate o_t (Equation 32) and the transformed c_t through the hyperbolic tangent activation function. The h_t performs as both the output for the present time step and the input for the subsequent time step (h_{t-1}), allowing the network to carry forward relevant information through the sequence. As with the classical NNs, weights biases are updated during training to minimize the respective loss function.

$$f_t = \sigma\left(W_f \cdot \left[h_{t-1,}x_t\right] + b_f\right) \qquad (27)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \qquad (28)$$

$$\tilde{c}_t = \tanh(W_c \cdot \left[h_{t-1,}x_t\right] + b_c) \qquad (29)$$

$$i_t = \sigma\left(W_i \cdot \left[h_{t-1,}x_t\right] + b_i\right) \qquad (30)$$

$$h_t = o_t \odot \tanh(c_t) \qquad (31)$$

$$o_t = \sigma\left(W_o \cdot \left[h_{t-1,}x_t\right] + b_o\right) \qquad (32)$$

Where:

$W_{\{f,i,o,c\}}$: weight matrices for the forget, input, output gates and the cell state

$b_{\{f,i,o,c\}}$: corresponding bias vectors

3.4.2 GRU

The innovative approach of LSTM networks has motivated many academics. Cho *et al.* (2014) presented GRU networks, which are an adaptation of LSTM.In contrast to LSTM, GRU incorporates two gates and possesses fewer parameters. Unlike LSTM, GRUs do not utilize cell states. The update gate (z_t)(Equation 33) controls how much information from the previous timestep (h_{t-1}) should be integrated with the new information (x_t). The reset gate (r_t)(Equation 34) decides how much of the previous hidden state (h_{t-1}) should be discarded. Next, the candidate hidden state/current memory content ($\tilde{h}_t$)(Equation 35) is calculated by integrating the r_t with x_t and h_{t-1}. Finally, the current hidden state (h_t) is being computed from both h_{t-1} and $\tilde{h}_t$, with the z_t controlling this combination via element-wise multiplication, as shown in Equation 36.

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \tag{33}$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \tag{34}$$

$$\tilde{h}_t = tanh(W \cdot [r_t \odot h_{t-1}, x_t]\,) \tag{35}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \tag{36}$$

3.4.3 Golden Jackal Optimizer

Golden jackals are medium-sized predator, and due to their size and elongated legs are able to cover vast distances in search of prey. Golden jackals are monogamous animals and live in mated pairs. Their family groups consist of one or two "*helpers*" who live with the parents for approximately a year and attend to the next child. They provide protection and nourishment to the young, while helpers employ predator barks and rumbling growls to signal them to find shelter. Golden jackals use several types of howls for communication, aiding in social bonding and hunting. Foraging family groups consistently inhabit territory covering many square kilometers, with areas delineated to discourage preyers. Cooperative hunts allow jackals to hunt larger prey within distances. The primary stages of golden jackal pair hunting consist of searching and moving towards the prey, enclosing and aggravating the prey until it stops moving, then pouncing towards it.

Formulation of the search domain

At first, the initial solution follows a uniform distribution across the search space (Equation 37).

$$Y_0 = Y_{min} + rand(Y_{max} - Y_{min}) \tag{37}$$

Where *rand* represents a random vector, and Y_{min} and Y_{max} corresponds to lower and upper bounds. This initialization generates the *Prey* matrix (Equation 38) where the top two solutions correspond to the male and female jackal, respectively. The F_{OA} matrix (Equation 39), estimates the fitness value for each prey.

$$Prey = \begin{bmatrix} Y_{1,1} & Y_{1,2} & \cdots & Y_{1,d} \\ Y_{2,1} & Y_{2,2} & \cdots & Y_{2,d} \\ \vdots & \vdots & \vdots & \vdots \\ Y_{n,1} & Y_{n,2} & \cdots & Y_{n,d} \end{bmatrix} (38)$$

$$F_{OA} = \begin{bmatrix} f(Y_{1,1}; Y_{1,2}; \cdots; Y_{1,d}) \\ f(Y_{2,1}; Y_{2,2}; \cdots; Y_{2,d}) \\ \vdots \\ f(Y_{n,1}; Y_{n,2}; \cdots; Y_{n,d}) \end{bmatrix} (39)$$

Exploration Stage (Searching the prey)

The golden jackals have the ability to identify and track their prey. However, if they determine that the prey cannot be caught easily, they may opt to wait and seek other prey. During each epoch (t), the leader of hunting is the male jackal (Equation 40), and the female follows (Equation 41)

$$Y_1(t) = Y_M(t) - E \cdot |Y_M(t) - rl \cdot Prey(t)| \quad (40)$$

$$Y_2(t) = Y_{FM}(t) - E \cdot |Y_{FM}(t) - rl \cdot Prey(t)| \quad (41)$$

Prey(t) is the position vector of the prey, where the $\mathrm{Y_M(t)}$ and $\mathrm{Y_{FM}(t)}$ corresponds top two solutions. The updated jackals' positions, with respect to the prey, are defined as $\mathrm{Y_1(t)}$ and $\mathrm{Y_2(t)}$.Prey's evading energy (E) (Equation 42) is equal to the multiplication of diminishing energy of the prey ($\mathrm{E_0}$) (Equation 43) and the initial energy state ($\mathrm{E_1}$) (Equation 44). c_1 is a conastant equal to 1.5, where T is the total number of epochs. $\mathrm{E_1}$ decreases linearly from 1.5 to 0 over the iterations.

$$E = E_1 \cdot E_0 \ (42)$$

$$E_0 = 2 \cdot r - 1 \ (43)$$

$$E_1 = c_1 \cdot (1 - 1(t/T)) \ (44)$$

The variable "rl" (Equation 45) is a vector of random numbers drawn from a Lévy distribution, where LF denotes the Lévy-flight function (Equation 46). Finally, Equation 47 defines the updated positions of the jackal pair.

$$rl = 0.05 \cdot LF(y) \ (45)$$

$$LF(y) = 0.01 \cdot (\mu \cdot \sigma) / \left(\left| v^{\left(\frac{1}{\beta}\right)} \right| \right); \sigma = \left(\frac{\Gamma(1+\beta) \cdot \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \cdot \beta \cdot \left(2^{\frac{\beta-1}{2}}\right)} \right)^{\frac{1}{\beta}} \ (46)$$

$Where \ \ \mu, v \in (0,1)$

$\beta = 1.5$

$$Y(t+1) = \frac{Y_1(t) + Y_2(t)}{2} \ (47)$$

Exploitation stage

As the prey is chased, its energy to evade decreases, enabling the jackals to surround it. Finally, the jackals attack their prey. (Equations 48 & 49)

$$Y_1(t) = Y_M(t) - E \cdot |rl \cdot Y_M(t) - Prey(t)| \quad (48)$$

$$Y_2(t) = Y_{FM}(t) - E \cdot |rl \cdot Y_{FM}(t) - Prey(t)| \quad (49)$$

The primary purpose of "*rl*" parameter, is to encourage randomness during the stage of exploitation, hence promoting exploration and avoidance of local optima. This parameter aids in overcoming the inertia of local optima, particularly in the final trials. This factor may be regarded as a consequence of barriers to accessing the prey. Environmental obstacles typically restrict jackals' efficient and rapid approach to their prey.

Balancing exploration and exploitation

As previously stated in the exploration phase description, the jackals may let the prey to escape if they judge that it cannot be easily captured. This process enables the GJO algorithm to transition from exploration to exploitation. During the hunt, it is anticipated that the prey may exhibit fatigue, as indicated by Equation 42. While E_0 decreases from zero to minus one, e prey experiences exhaustion , whereas when E_0 increases from zero to one, it indicates a rise in prey's strength. Given, these conditions when $|E| > 1$ the jackals will stop the pursuit of the targeted prey. Conversely, when $|E| < 1$ the jackals will attack through the prey conducting the exploitation phase. The GJO's pseudocode is presented in Appendix A.

4. Experimental Results

In the following subsections, we discuss the results of the applications proposed in this thesis. For financial risk assessment, both tasks are classification problems, and we evaluate two benchmark datasets that have been extensively studied in the literature. Churn prediction is an important problem not only for financial institutions but also across many sectors, as retaining valuable customers is crucial. Predicting credit card default is another key challenge for financial institutions, thus the early identification can improve risk management and reduce losses.For investment decision-making, we first examine portfolio optimization using metaheuristic algorithms, maximizing the Sharpe ratio which is a metric widely used by researchers and practitioners. Finally, we present the performance of the proposed approach for forecasting stock closing prices, reporting a range of metrics relevant to time-series prediction. Together, these studies not only re-examine the performance of AI and metaheuristic algorithms but also offer practical guidelines for researchers and investors in the financial domain.

4.1 Churn Prediction

4.1.1 Evaluation Metrics

Measuring the ability of the model to provide accurate predictions is vital for drawing accurate conclusions. Various measures can be employed to assess the robustness of the model, depending upon the particular task. In classification tasks like predicting customer churn, several metrices can be calculated from confusion matrix (Table 17). While measuring the accuracy of classifications is often used, this metric may yield arbitrary conclusions, particularly in the context of an imbalanced dataset. Recall is the best suitable statistic for assessing the accuracy of predicted positive classes. Precision is an important metric, which is defined as the ratio of true positives to the overall number of positives. The F1-score offers a more comprehensive viewpoint between precision and recall. This is defined as the ratio of the harmonic mean to the algebraic mean of precision and recall (Table 17). To correctly predict customer churn classes, both classes must be regarded with the same weight. The previously m entioned depend only on churn predictions.

Hence, further metrics should be evaluated. The True Positive Rate (TPR) or sensitivity criterion, is equivalent to recall, providing a clearer view of how many actual positives are predicted. Furthermore, the specificity criterion derives equivalent information about non-churners. As the F1 score summarizes precision and recall in a single metric, the Receiver Operating Characteristic (ROC) curve for provides aprovides an overall view of sensitivity versus specificity. Computing the Area Under the ROC Curve (AUC) provides a more comprehensive view of the model's performance. This metric, is less sensitive imbalanced datasets and following prior works (Burez & Van den Poel, 2009; De Caigny et al., 2018) it can be considered a more reliable metric for predicting customer churn.

Table 17.
Confusion matrix

	Predicted Negative ($\hat{y} \in 0$)	Predicted Positive ($\hat{y}$ $\in 1$)
Actual Negative ($y \in 0$)	True Negative (**TN**)	False Positive (**FP**)
Actual Positive ($y \in 1$)	False Negative (**FN**)	True Positive (**TP**)

$$\boldsymbol{Accuracy = \frac{TP + TN}{TP + TN + FP + FN}}$$

$$\boldsymbol{Recall = \frac{TP}{TP + FN}}$$

$$\boldsymbol{Precision = \frac{TP}{TP + FP}}$$

$$\boldsymbol{F1 - score = \frac{Precision \times Recall}{\frac{Precision + Recall}{2}} = \frac{2TP}{2TP + FP + FN}}$$

$$\textbf{Sensitivity} = \boldsymbol{\frac{TP}{TP + FN}}$$

$$\textbf{Specificity} = \boldsymbol{\frac{TN}{TN + FP}}$$

$$\boldsymbol{ROC = \frac{TPR}{FPR} = \frac{Sensitivity}{1 - Specificity}}$$

$$\boldsymbol{AUC = \int_0^1 \frac{TP}{TP + FN}\, d\frac{FP}{TN + FP}}$$

4.1.2 Dataset Description

The dataset, sourced from Kaggle, featured 10,000 bank customer records free of missing data and had 14 characteristics. The characteristics "Row Number," "Customer ID," and "Surname" were excluded due to their irrelevance. Therefore, our dataset includes 11 features (Table 18), with 'Churn' designated as the key variable in our research. The dataset exhibited a customer leave rate of 20.37%, contrasted with a retention rate of 79.63%, suggesting a significant imbalance between the two classes (Figure 24). The dataset includes several economic and demographic characteristics as detailed in Table 18. The numerical representation of these variables is located in Tables 19-24. Given that the dataset has no missing values, no additional actions are required in this regard. The dataset includes some outliers, particularly for the Age variable (Figure 25). Next the presence of correlation among the features was examined (Figure 26). The existence of correlation in the dataset can cause algorithmic challenges that affect training success and result in overfitting. Figure 6 demonstrates a minimal correlation among the variables, as shown by the highest correlation coefficient not surpassing 0.3 (Akande *et al.*, 2015).

Table 18
Features Description

Feature	Description
Age	Customer's age
Balance	Customer's account balance
Churn	Customer has exited the bank
Credit Score	Score reflecting credit usage
Estimated Salary	Estimated salary of the customer
Gender	Customer's gender
Geography	Customer's country/region
HasCrCard	Customer holds a credit card
IsActiveMember	Customer is an active member
Num Of Products	Number of products the customer uses
Tenure	Length of time the account has been held

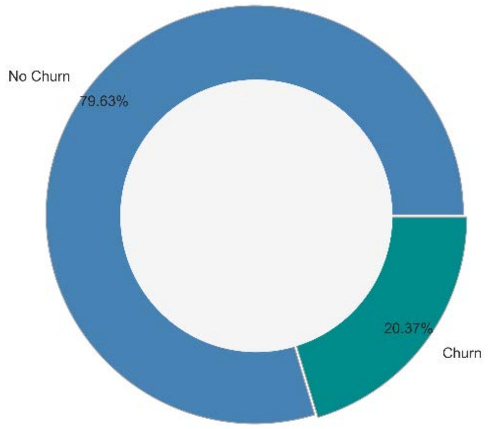


Figure 24. Churn Distribution

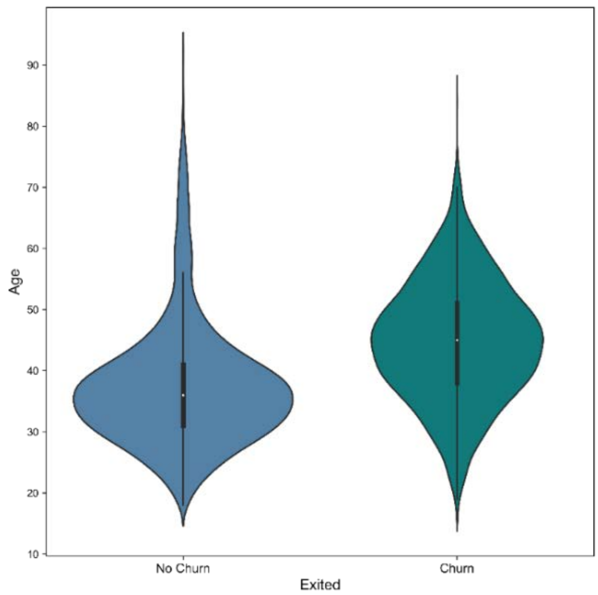


Figure 25. Violin Plot for Age feature

Table 19. Customer's Gender

Label	Female	Male
Non-churner	3404	4559
Churn	1139	898

Table 20. Customer's Country

Label	France	Germany	Spain
Non-churner	4204	1695	2064
Churn	810	814	413

Table 21. Customer's Activity

Label	Inactive	Active
Non-churner	3547	4416
Churn	1302	735

Table 22. Customer Has a Credit Card

Label	No	Yes
Non-churner	2332	5631
Churn	613	1424

Table 23. Products per Customer

Label	1	2	3	4
Non-churner	3675	4242	46	0
Churn	1409	348	220	60

Table 24. Description of continuous variables

	Age	Balance	CreditScore	EstimatedSalary	Tenure
Min	18	0	350	11.58	0
Max	92	250,898	850	199,993	10
Mean	38.9	76,486	650.5	100,090	5
Std Dev	10.5	62,397	96.7	57,511	2.9
Median	37	97,199	652	100,194	5

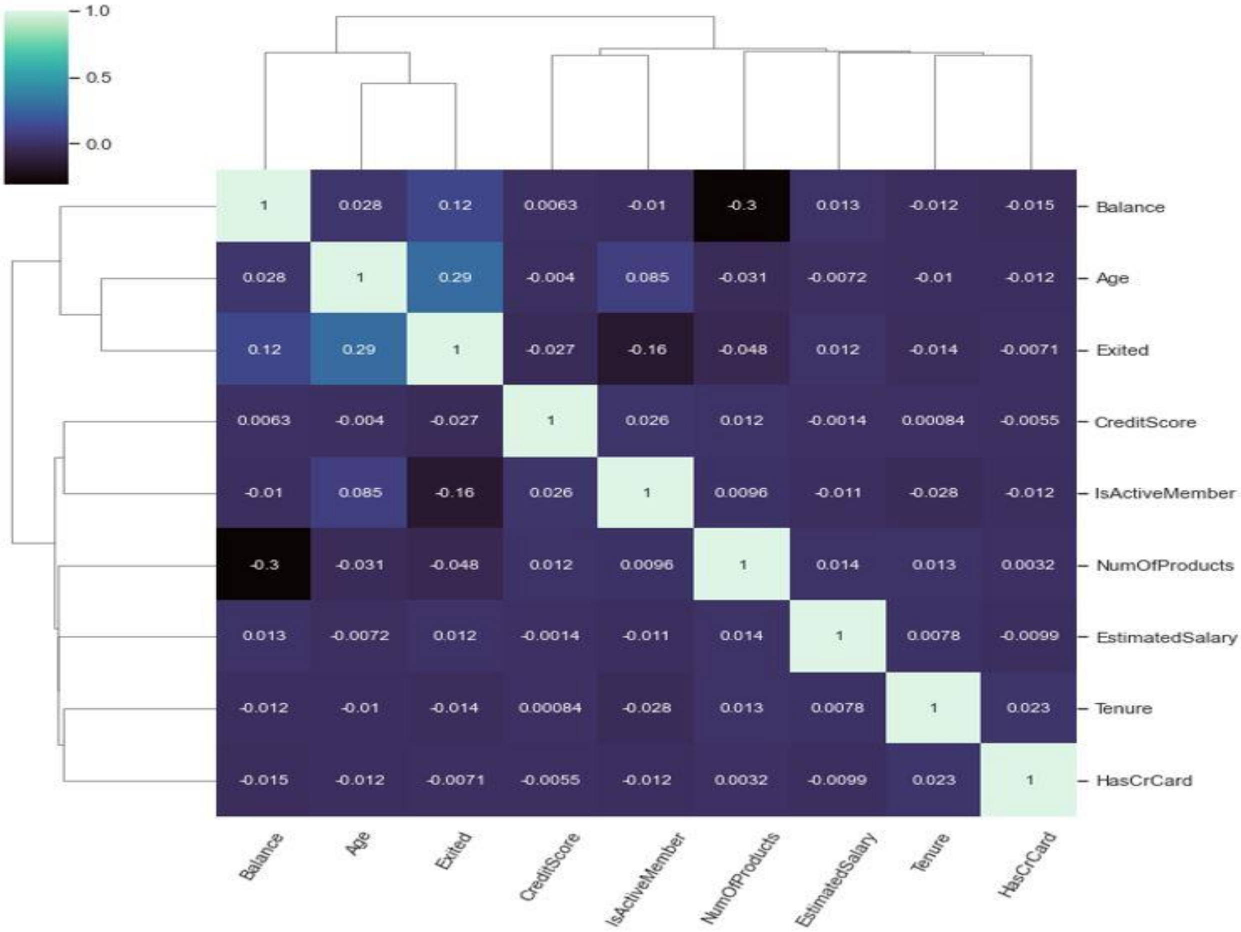


Figure 26. Clustered heatmap for correlation analysis

4.1.3 Data Engineering and Methodology

For the trials the Python programming language was utilized. For the implementation of the activation functions and optimizers Keras library was employed, a framework developed on top of TensorFlow (Abadi *et al.*, 2016). The APTx activation function was manually defined using Keras based on the formula supplied by Kumar (2022), and the RAdam optimizer was implemented via the keras_Radam. We also employed the Imbalanced-learn package (Lemaître *et al.*, 2017) for the implementation of resampling methods. In our dataset, the categorical features "Gender" and "Country" were preprocessed by One-Hot encoding, yielding a total of thirteen input variables, while for feature scaling the RobustScaler was employed. Typically, standard scaling is utilized in the majority of studies analyzing customer attrition. However, in this work we use RobustScaler, as it is more suitable for this dataset due to the presence of outliers.

We used stratified 10-fold cross-validation to evaluate performance. Each model consists of two hidden layers, as per Domingos *et al.* (2021), of seven units, using the specified activation function, with a sigmoid output layer, since churn prediction is a binary classification task. Each model was compiled using binary cross-entropy. The number of units in each hidden layer can affect performance. We followed a common practice of setting hidden-layer size to the average of the input and output layer sizes. Adam, Adagrad, RAdam, RMSProp, optimizers were examined, with their default learning rate. Each model was trained for 100 epochs with a batch size of 16, and performance was assessed using Accuracy and AUC. As mentioned in Subsection 4.1.1, AUC score is the most appropriate metric for this task; however, we also seek to investigate the trade-off between these two metrics. For experiments except the case of original dataset, we evaluated the performance of five resampling methods. These included ENN and Tomek Links for undersampling, SMOTE for oversampling, and the hybrid methods SMOTE–ENN and SMOTE–Tomek. As stated in Subsection 3.1.1, we implement the designated resampling techniques exclusively on the training set to guarantee unbiased and reliable results. Tables 25-30 display the findings, which will be further analyzed in the subsections that follow.

Table 25. Performance Results without resampling method

	Adagrad		Adam		RAdam		RMSProp	
	Accuracy	**AUC score**	**Accuracy**	**AUC score**	**Accuracy**	**AUC score**	**Accuracy**	**AUC score**
APTx	81.03	57.8	86.18	72.16	86.51	**73.41**	86.18	72.54
ELU	80.53	57.52	86.1	72.13	86.28	72.81	85.93	71.06
LReLU	80.38	56.99	86.05	71.62	86.19	71.82	86.24	72.03
Mish	80.64	57.43	86.42	72.73	86.48	72.82	86.21	72.53
ReLU	80.93	57.05	86.24	71.7	86.04	71.8	85.73	71.06
Swish	80.3	56.65	86.34	72.61	86.26	72.5	86.32	72.78

Table 26. Performance Results (ENN)

	Adagrad		Adam		RAdam		RMSProp	
	Accuracy	**AUC score**	**Accuracy**	**AUC score**	**Accuracy**	**AUC score**	**Accuracy**	**AUC score**
APTx	79.55	68.06	83.24	77.69	83.42	**77.86**	83.58	77.65
ELU	79.97	68.54	83.27	77.07	83.01	77.66	83.45	77.44
LReLU	78.56	65.72	83.43	77.78	83.51	77.72	83.59	77.37
Mish	80.25	68.26	83.26	77.05	83.08	77.23	83.72	77.19
ReLU	78.92	66.1	83.01	77	83.29	77.73	84.06	77.26
Swish	79.87	66.56	83.11	77.18	83.29	77.73	83.81	77.69

Table 27. Performance Results (Tomek Links)

	Adagrad		Adam		RAdam		RMSProp	
	Accuracy	**AUC score**	**Accuracy**	**AUC score**	**Accuracy**	**AUC score**	**Accuracy**	**AUC score**
APTx	81.25	55.23	85.87	73.52	86.08	73.82	86.26	**74.28**
ELU	81.71	60.75	86.06	73.95	85.98	73.71	85.87	73.57
LReLU	81.14	59.74	85.8	73.49	86.17	73.82	86.17	74.24
Mish	82.46	59.79	85.87	73.5	85.96	73.48	85.97	74.26
ReLU	79.63	50	85.89	73.6	86.04	73.92	86.03	73.65
Swish	82.11	58.69	85.84	72.61	86.1	73.59	86.1	74.12

Table 28. Performance Results (SMOTE)

	Adagrad		Adam		RAdam		RMSProp	
	Accuracy	**AUC score**	**Accuracy**	**AUC score**	**Accuracy**	**AUC score**	**Accuracy**	**AUC score**
APTx	74.43	74.77	81.07	78.32	81.08	78.17	79.33	77.63
ELU	71.69	71.99	80.7	78.13	80.02	78.08	80.18	77.49
LReLU	71.71	72.46	79.74	77.65	78.45	77.15	80.38	**78.40**
Mish	73.7	74.04	80.12	77.74	80.66	77.6	79.51	77.2
ReLU	74.46	73.86	80.08	77.35	79.95	77.6	80.16	77.63
Swish	73.06	73.49	80.65	77.69	81	78.35	80.39	77.84

Table 29. Performance Results (SMOTE–ENN)

	Adagrad		Adam		RAdam		RMSProp	
	Accuracy	AUC score	Accuracy	AUC score	Accuracy	AUC score	Accuracy	AUC score
APTx	66.45	71.04	76.69	**78.32**	77.03	77.8	75.86	76.93
ELU	66.27	70.80	73.68	76.69	75.83	77.41	76.55	77.55
LReLU	65.61	70.19	75	77.41	74.73	76.97	76.06	77.04
Mish	66.51	70.95	76.16	77.23	77.3	78.02	75.93	77.32
ReLU	65.02	70.22	75.06	76.56	74.47	76.21	75.77	76.75
Swish	66.4	70.86	76.09	76.89	75.78	77.26	76.26	77.35

Table 30. Performance Results (SMOTE–Tomek Links)

	Adagrad		Adam		RAdam		RMSProp	
	Accuracy	AUC score	Accuracy	AUC score	Accuracy	AUC score	Accuracy	AUC score
APTx	72.99	73.6	80.85	78.13	80.34	**78.5**	79.46	77.47
ELU	71.7	71.97	81.05	77.67	81.81	77.54	80.39	78.35
LReLU	70.73	71.45	80.1	77.71	80.3	78.02	79.37	77.42
Mish	71.93	73.1	80.05	77.23	80.27	77.97	79.89	77.6
ReLU	72.51	72.46	80.71	78.39	81.93	78.1	78.51	77.01
Swish	71.34	72.54	80.74	77.99	81.06	77.75	79.44	77.68

4.1.4 Results

In the first experiment, we evaluated the models' performance on the original dataset (Table 25). As we can observe the conclusions derived only from accuracy may be inaccurate. The existance of class imbalance resulted in a significant trade-off between accuracy and the AUC score. Across the different optimizers, the ReLU Across the different optimizers was the only activation function to achieve an AUC score below 72%. When the RMSProp was employed optimizer, LReLU achieved the best performance (72.03%). ELU performed better when combined with Radam optimized. Swish performed similar to the three activation functions mentioned above. Still, it performed worse than the ELU-Radam configuration. When Radam was employed, both APTx and Mish showed better performance. Notably, APTx combined with RAdam produced an AUC score over 73%. A key finding of this experiment is Adagrad's weak performance.

Following that, we utilized the ENN and Tomek Links undersampling methods (Tables 26-27). Upon the application of ENN, we noted better performance across all optimizers–activation functions combinations in contrast to the first experiment. The Radam-ReLU performed the best with an AUC score of 77.73%. Moreover, LReLU exhibited improved performance when combined with Adam (77.78%.), whereas the ELU attained a peak performance of 77.66% when coupled with RAdam. In addition, when combined with RAdam, Swish reached a peak AUC of 77.73%, matching that of ReLU–

RAdam. While Mish achieved its highest AUC when paired with RAdam, the best overall result came from APTx–RAdam, which reached 77.86%. In general, APTx showed similar performance across all optimizers, with the exception of Adagrad. Adagrad consistently yielded the lowest AUC values across all combinations of activation functions. Tomek Links underperformed compared with ENN. When combined with Radam, ReLU achieved the greatest AUC score of 73.92%. When combined with Adam, ELU achieved the greatest AUC score of 73.95%. The remaining four activation functions attained their highest AUCs with the RMSProp optimizer, all exceeding 74%. APTx-RMSProp achieved the greatest AUC score in this context (74.28%). Adagrad consistently shown the poorest performance among all optimizers. When combined with ELU, it exhibited superior performance relative to other activation functions (60.75%).

Next, we employed SMOTE (Table 28), resulting in better AUC scores compared to ENN in most instances, where the LReLU-RMSProp configuration yielded the best results (78.50%) in this scenario. Swish attained the second greatest performance in conjunction with RAdam (78.35%). ELU, Mish, and APTx shown optimal performance when combined with Adam. Adagrad demonstrated superior performance in this experiment compared to prior circumstances. The utilization of APTx as an activation function yields a 74.77% enhancement. Nevertheless, it performed poorly in comparison to other optimizers.

In our concluding experiments, we employed the two mixed resampling methods: SMOTE-ENN and SMOTE-Tomek Links (Tables 29-30). The utilization of SMOTE-ENN resulted in ReLU achieving the best performance with RMSProp.Swish and ELU and Swish performed the best when paired with RMSProp. When utilized alongside Adam, LReLU attained a maximum AUC score of 77.41%. APTX-Adam exhibited the highest AUC score in this context (78.32%). Mish-Radam comes second in terms of performance, with an AUC score of 78.02%. Once again, Adagrad had the poorest performance among the optimizers in this experiment. Nonetheless, when APTx was employed, Adagrad performed the best (71.04%).

Finally, the SMOTE-Tomek Links was employed. All activation functions surpassed or nearly approached the 78% threshold of AUC score. The combination of ReLU and Adam achieved a maximum AUC score of 78.39%. Among all scenarios, this was the top score for both ReLU and Adam. The combination of LReLU and Radam achieved an AUC score of 78.02%. ELU performed the best (78.35%) when paired with RMSprop. Mish and Swish achieved an AUC score of approximately 78%. The APTX-RAdam combination attained the greatest AUC score in this scenario, having a peak performance of 78.50%. Adagrad consistently demonstrates the lowest performance among the optimizers. However, when paired with APTx, it obtained the greatest AUC score among all activation functions.

4.1.5 Discussion

The results indicate that RAdam provides better performance than Adam in certain scenarios. Although, the application of Radam is relatively limited in the existing literature, the results are aligned with prior works (Cui *et al.*, 2020; Sun *et al.*, 2020; Mazumder *et al.*, 2022; Ko *et al.*, 2024). Moreover, Sun *et al.* (2020) conducted a comparative analysis on the performance of the Mish and ReLU,where they observed that Mish performed better than ReLU when utilized with the RAdam optimizer, consistent with our first scenario, where we did not employed a resampling technique. When using a sampling method, it is impossible to obtain a clear result. RMSProp-APTx and RMSProp-LRelu, when employing Tomek Links and SMOTE, achieved the highest AUC values. Across all optimizers, Adagrad performed the worst. Several works exist, point out Adagrad's inferior performance, particularly in cases of imbalanced datasets (Sarigül & Avci, 2018; Dalli, 2022; Rex *et al.*, 2022; Zhou *et al.*, 2022; Liu *et al.*, 2023; Reyad *et al.*, 2023; Woźniak *et al.*, 2023). RAdam performed in four out of six situations.

The APTx activation function showed better results in all experiments, with the scenario when SMOTE was employed as the exception. In half of the cases, APTx–RAdam achieved the highest AUC. When SMOTE-Tomek Links was employed, an improved stability was observed in terms of performance across all activation functions, each attaining or nearly reaching an AUC score of 78% with Adam, RAdam, RMSProp as the optimizers. SMOTE as a resampling method showed good performance. Regarding the two undersampling methods examined in this work, Edited Nearest Neighbors (ENN) achieved better results compared to Tomek Links. Overfitting was noticed during the training procedure for both ENN and SMOTE-ENN. In general, ENN can more aggressively undersample the majority class compared to other undersampling strategies. This work used the same model architecture for all combinations without integrating additional steps to tackle overfitting. However, APTx demonstrated a reduced sensitivity to overfitting, as it achieved the highest AUC score in both cases.

Moreover, we concur with the conclusions of Imani & Arabnia (2023), as it is evident that SMOTE-Tomek Links can provide better results in comparison to SMOTE, and we also observed that SMOTE-ENN exhibited worst performance relative to the other two resampling techniques. In contrast, Kimura (2022) work’s show that XGBoost model, achieved the maximum performance regarding AUC score without applying any resampling techniques. Both works assessed the accuracy of several classical and ensemble machine learning models. Our research demonstrates that employing resampling techniques is crucial for improving the predictive accuracy of a DNN model. No single dominant resampling method is evident; its efficacy is dependent on the chosen model, the dataset's attributes, and the data engineering processes, as supported by various studies in the literature (Burez & Van den Poel, 2009; Imani & Arabnia, 2023; Haddadi *et al.*, 2024). Our findings reveal that the proper choice of a DNN’s components can influence predictive performance, in the context of imbalanced learning. Nonetheless, the APTx demonstrated superior stability and achieved the highest AUC score in all cases. Except when SMOTE was employed. Furthermore, the results demonstrated that the RAdam optimizer outperformed Adam. Significantly, even with the use of SMOTE, the integration of RAdam with Swish produced an AUC score nearly equivalent to that of RMSprop-LeakyReLU. In conclusion, the results show that APTx can be regarded as a viable activation function for NN models, since it can improve forecasting quality.

4.2 Credit Card Default Prediction

4.2.1 Data Description

The examined dataset was obtained from the UCI Repository and consists of 30,000 client records from Taiwan (Yeh, 2009). Approximately 78% of these observations belong to non-defaulters, whereas the remainder refers to defaulters. In addition to the considerable imbalance ratio between these two classes, the dataset includes other variables, such as credit amount, demographics (gender, education, marital status, age), payment history, bill statements, and past payments. Among the 24 features (Table 31), the "ID" column is omitted due to the absence of predictive value Due to the inherent insensitivity of tree-based models to varying scale, feature scaling is generally not essential. Our dataset demonstrates significant imbalance between the two classes and contains a substantial number of features. To handle high-dimensional data, we rely on LightGBM's native categorical processing rather than one-hot encoding. To handle the class imbalance, we set the LightGBM parameter is_unbalance=True. We optimize the LightGBM model using AUC score, which is a more robust metric with imbalance datasets.

Table 31. Dataset Description

Feature	Description
ID	Client's ID
Limit_Bal	Credit amount provided
Sex	Client's Gender
Education	Client's educational level
Marriage	Marital Status
Age	Client's age
Pay_1-Pay_5	Repayment status
Bill_Amt1-Bill_Amt6	Amount of bill statement
Pay_Amt1-Pay_Amt6	Prior payment amount
Default Payment_Next Month	Default will occur next month

4.2.2 Experimental Setup

Proper tuning of LightGBM's hyperparameters is crucial to improving the model's performance. In this approach, we propose 10 essential hyperparameters of LightGBM for the optimization task, as per the research of Hou *et al.* (2025). Table 32 presents the default settings for these hyperparameters along with the corresponding optimization range.The TPE sampler, an established and respected Bayesian optimization method, will serve as the benchmark model for the optimization framework. Concerning PSS, we employ four resampling methodologies: MC, LHS, HS, and SHS. Additionally, inspired by the authors of PSS work, we utilize two separate acceptance probabilities: 0.85 and 0.95. The population size and the number of epochs were established at 30 and 100, respectively. For TPE-LightGBM, we evaluated five different numbers of trials (50, 100, 200, 500, 3000). The number of trials represents the optimization budget, indicating the quantity of hyperparameter configurations evaluated by TPE during hyperparameter optimization. While thousands of trials are rare in literature, we incorporate 3000 trials to align with the search effort of PSS-LightGBM, which assesses around 100 epochs (100×30 agents ≈1000 candidate solutions), facilitating a fair comparison.

TPE-LightGBM functions as our reference model, given that TPE is a recognized optimization approach in the literature; comparing PSS to it offers further validation of its robustness. To determine the resilience of the optimized models, we do a 5-Fold Stratified Cross Validation. Each model was executed 30 times utilizing the identical set of random seeds to guarantee an equitable comparison. Our principal metric is the mean AUC score and the generalization performance (AUC train–test disparity), and we additionally give the highest test AUC attained during the 30 iterations. Tables 33 & 34 provide the related results, whereas Table 35 reports the time performance of the models. The subsequent subsection will address the comparative performance of each model. We performed experiments utilizing the Python programming language. The TPE sampler was employed via the Optuna program (Akiba, 2019), whilst the metaheuristic algorithms were executed utilizing the MEALPY package (Van Thieu & Mirjalili, 2023). Furthermore, the Scipy package (Virtanen et al.,2020) assisted in integrating the HS and SHS within the PSS context. In the next section, we present the results, emphasizing the best outcomes in bold for both the AUC score and generalization error.

Table 32. Selected Hyperparameters for the optimization task

Hyperparameters	Default values	Candidate values
Num_leaves	31	[4,64]
max_depth	-1	[3,12]
learning_rate	0.1	[0.001-1]
n_estimators	100	[10-4000]
Min_data_in_leaf	20	[2-200]
min_sum_hessian_in_leaf	0.001	[0.001-1]
feature_fraction	1.0	[0.01-1]
bagging_fraction	1.0	[0.01-1]
lambda_l1	0.0	[0-1]
lambda_l2	0.0	[0-1]

Note: The default value of -1 for max_depth parameter indicates no limit.

Table 33. Classification results for Benchmark Models

Model	NT	Mean AUC	Train-Test AUC	Max AUC	Recall	Precision	F1score
Standard-LightGBM	*N/A*	0.7781 ± 0.0077	0.1049	0.7971	0.6240	0.4676	0.5346
TPE-LightGBM	*50*	0.7801 ± 0.0076	0.0342	0.7944	0.6330	0.4651	0.5361
TPE-LightGBM	*100*	0.7812 ± 0.0076	0.0333	0.7962	0.6353	0.4633	0.5358
TPE-LightGBM	*200*	0.7818 ± 0.0077	0.0345	0.7968	0.6379	0.4622	0.5360
TPE -LightGBM	*500*	0.7816 ± 0.0079	0.0331	0.7994	0.6353	0.4637	0.5360
TPE -LightGBM	*3000*	0.7817 ± 0.0076	0.0319	0.7989	0.6363	0.4638	0.5365

Note. *NT*: *Number of Trials. The table reports mean metric values over 30 runs and the maximum AUC score*

Table 34. Classification results for PSS-LightGBM

Model	AR	AUC	Train-Test AUC	Max AUC	Recall	Precision	F1score
PSS- MC -LightGBM	0.85	0.7813 ± 0.0079	0.0351	0.7985	0.63534	0.46255	0.53525
PSS- MC -LightGBM	0.95	0.7814 ± 0.0074	0.0336	0.7991	0.63640	0.46374	0.53645
PSS- LHS -LightGBM	0.85	0.7816 ± 0.0075	0.0332	0.7990	0.63605	0.46315	0.53590
PSS- LHS -LightGBM	0.95	0.7813 ± 0.0079	0.0373	0.8011	0.63763	0.46321	0.53651
PSS-HS-LightGBM	0.85	0.7818 ± 0.0076	0.0340	0.7991	0.63567	0.46264	0.53545
PSS-HS-LightGBM	0.95	0.7809 ± 0.0079	0.0368	0.7994	0.63640	0.46422	0.53675
PSS-SHS-LightGBM	0.85	**0.7819 ± 0.0072**	**0.0298**	0.7993	0.63703	0.46253	0.53584
PSS-SHS-LightGBM	0.95	0.7812 ± 0.0076	0.0343	0.7989	0.63471	0.46264	0.53508

Note. *AR: Acceptance Rate. The table reports mean metric values over 30 runs and the maximum AUC score*

Table 35. Mean Runtime by Model

Model	Mean runtime (s)
Standard-LightGBM	0.09
TPE-LightGBM (NT=50)	8.96
TPE-LightGBM (NT=100)	13.84
TPE-LightGBM (NT=200)	22.53
TPE -LightGBM (NT=500)	64.82
TPE -LightGBM (NT=3000)	653.64
PSS- MC -LightGBM (AR=0.85)	383.36
PSS- MC -LightGBM (AR=0.95)	235.51
PSS- LHS -LightGBM (AR=0.85)	362.48
PSS- LHS -LightGBM (AR=0.95)	223.83
PSS-HS-LightGBM (AR=0.85)	345.51
PSS-HS-LightGBM (AR=0.95)	226.35
PSS-SHS-LightGBM (AR=0.85)	383.36
PSS-SHS-LightGBM (AR=0.95)	235.51

4.2.3 Results and discussion

The results indicate that hyperparameter optimization is crucial for the performance of LightGBM. In our first experiment, LightGBM, utilizing the default hyperparameters, achieved an average AUC score of 77.81%. It is clear from the 10.49% difference between the training and test AUC scores that our model is significantly overfitting and lacks the ability to generalize to unseen data. Next, we conducted hyperparameter optimization via the TPE approach. The results demonstrated that TPE-LightGBM produced superior outcomes, as assessed by both the mean AUC score and generalization error. Increasing the quantity of TPE trials enhances the mean AUC beyond the conventional LightGBM baseline (0.7781 ± 0.0077) to around 0.781–0.782 for all trial quantities, achieving the optimal mean at 200 trials (0.7818 ± 0.0077). TPE consistently diminishes the train–test AUC disparity from 0.1049 to approximately 0.032–0.035 (with a minimum of 0.0319 at 3000 trials), indicating enhanced generalization. The expense escalates significantly: the average execution time increases from 8.96 seconds (50 trials) to 653.64 seconds (3000 trials) with minimal enhancement in AUC; the maximum AUC fluctuates somewhat (0.794–0.799). Among all TPE-LightGBM executions, 200 trials produced the optimal average performance; yet, regarding generalization and maximum AUC, TPE with 3,000 trials exhibited superior results.

Subsequently, we assess the efficacy of PSS-LightGBM. In general, we note comparable performance among all resampling techniques. When the Monte Carlo method was utilized as a resampling technique, its performance was marginally superior with an acceptance rate (AR) of 0.95 compared to 0.85. When LHS was utilized as a resampling technique with an AR of 0.95, it attained the greatest AUC score among all tests (0.8011). Nonetheless, when the AR was adjusted to 0.85, it attained enhanced stability in performance and reduced generalization error. In the case of PSS-HS-LightGBM, an AR of 0.85 resulted in enhanced stability (0.7818 ± 0.0076) and an improved AUC gap (0.0340). The optimal outcome was attained using PSS–SHS-LightGBM (AR=0.85) with a value of 0.7819 ± 0.0072, slightly surpassing the best TPE mean (0.7818 ± 0.0077 at 200 trials). Nonetheless, when the AR was established at 0.95, PSS-SHS-LightGMB exhibited inferior performance. Overall, SHS with AR=0.85 provides the optimal accuracy-generalization balance among all configurations in the tests, exhibiting a generalization error of 0.0298 and attaining the third-highest maximum AUC score of 0.7993. In terms of time complexity, we noted that an AR of 0.85 requires more time than an AR of 0.95. This is anticipated, as a lower AR results in increased exploration by the PSS, hence extending the duration of the optimization process.

In summary, PSS-SHS-LightGBM demonstrates superior predictive accuracy and reduced generalization error compared to other sampling methods, regular LightGBM, and TPE-LightGBM. We noted that an AR of 0.85 yielded more generalizable models compared to an AR of 0.95, except when MC was utilized. The authors of PSS addressed the challenge of constructing a reinforced concrete beam as articulated by Shaqfa & Orban (2019) utilizing an AR of 0.85, due to the complexity of the optimization issue and the necessity for greater diversification. We noted our alignment with the authors, since setting the AR to 0.85 resulted in superior generalization performance in the majority of PSS-LightGBM models.

Concerning time complexity, we observed that TPE delivers solid solutions in a reduced duration. In contrast, utilizing 3000 trials, analogous to 100 epochs and 30 agents for a metaheuristic algorithm, resulted in an average time of 653.64 seconds for TPE, significantly exceeding all configurations of PSS-LightGBM and PSS-SHS-LightGBM with an AR of 0.85 (365.14 seconds). This study acknowledges the efficacy of the TPE algorithm, which is well-established in the existing literature. We noted that TPE yielded favourable outcomes in various instances. The primary objective of this study is to identify a configuration that produces reliable and flexible machine learning models while avoiding complex architectures and to investigate the potential of the PSS metaheuristic algorithm. While the performance of TPE-LightGBM remains noteworthy, PSS-SHS-LightGBM appears to be a more dependable method for forecasting credit card default. Using LightGBM's built-in features with proper hyperparameter tuning produces a reliable credit-card default model.

Table 36

Performance comparison of the proposed model with approaches from other studies

Authors	Model	AUC score	Rank
He *et al.* (2018)	BalanceCascade,PSO-optimized ensemble method	0.78773	4
Jadhav *et al.* (2018)	IGDFS+NB	0.69861	12
Hsu *et al.*,2019	RNN-RF	0.7820	5
Zhang & Qui (2020)	CSO-ANN	0.7360	10
Zhang *et al.* (2021a)	Multi-stage ensemble model	0.80913	1
Zhang *et al.*, (2021b)	Multi-stage ensemble model	0.79373	3
Shen *et al.*,2021	Improved SMOTE+Combined LSTM-Adaboost	0.7475	9
Liu *et al.*, (2022)	mg-mGBDT	0.7490	8
Zou & Gao (2022)	AugBoost-ELM	0.7516	7
Xu & Qu (2024)	MVO-MLP	0.7329	11
Zhang *et al.*,2025	IWSL	0.753	6
Proposed approach	PSS-SHS-LightGBM	0.7993	2

Table 36 shows a comparison to the available research that has analyzed the respective dataset. Although the lack of considerable data engineering and hybrid techniques, our model performed well in terms of test AUC. When AR epochs were set to 0.85, the PSS-SHS-LightGBM model had the most stable performance (0.7819 ± 0.0072), the best generalization performance (2.98%), and the highest AUC score (79.93%). It is worth noting that we do not claim effective that effective data processing is unnecessary; using dimensionality reduction techniques or even resampling approaches could improve our model's performance. However, the proposed approach enables financial institutions to reduce possible risks even more. Furthermore, a risk manager may decide not to use sophisticated and time-consuming data engineering techniques (Jin *et al.*, 2020), particularly for less complicated datasets with lower dimensionality and class imbalance. Given this, practitioners can avoid using sophisticated structures while yet achieving a high level of predictability for credit card default, thus contributing significantly to credit risk stabilization.

4.3 Metaheuristic Portfolio Optimization

4.3.1 Dataset and evaluated algorithms

The experimental evaluation was carried out using historical DJIA stock data from Yahoo Finance dated January 1, 2023, to August 23, 2024 (Table 37). For each stock, we maximize the Sharpe ratio with a risk-free rate of 5%. The metaheuristic algorithms used in this investigation are outlined in Subsection 3.3.1 and summarized in Table 38. The metaheuristic algorithms were chosen to run for three epochs (30, 50, 100) and four population sizes (10, 30, 50, and 100). All algorithms were implemented via MEALPY package. Furthermore, as with previous studies (Mirjalili *et al.*, 2014, Mirjalili *et al.*, 2015, Mirjalili *et al.*, 2016, Abdollahzadeh *et al.*,2021), this study records and preserves the highest, average, and standard deviation of each algorithm during 30 iterations. Reporting average and standard deviation can provide information about the stability of any algorithm's performance (Mirjalili, 2015). However, given the stochastic nature of metaheuristic algorithms, statistical tests should be used to gain a better understanding of their performance (Derrac *et al.*, 2011). To verify comparable statistical significance, use the Wilcoxon rank sum test (Derrac *et al.*, 2011; Mirjalili & Lewis, 2013; Mirjalili, 2015). The Wilcoxon rank-sum test is a non-parametric statistical approach for determining whether there is a statistically significant difference between two groups.

Table 37. Tickers of DIJA

Stock	Ticker	Stock	Ticker
3M	MMM	IBM	IBM
American Express	AXP	Intel	INTC
Amgen	AMGN	Johnson & Johnson	JNJ
Amazon	AMZN	JPMorgan Chase	JPM
Apple	AAPL	McDonald's	MCD
Boeing	BA	Merck	MRK
Caterpillar	CAT	Microsoft	MSFT
Chevron	CVX	Nike	NKE
Cisco	CSCO	Procter & Gamble	PG
Coca-Cola	KO	Salesforce	CRM
Disney	DIS	Travelers	TRV
Dow	DOW	United Health Group	UNH
Goldman Sachs	GS	Verizon	VZ
Home Depot	HD	Visa	V
Honeywell	HON	Walmart	WMT

Table 38. Evaluated Metaheuristic Algorithms

Algorithm
Artificial Bee Colony (ABC)
Artificial Gorilla Troop (AGTO)
Ant Lion Optimizer (ALO)
Aquila Optimizer (AO)
Artificial Rabbit Optimizer (ARO)
Bat Algorithm (BA)
Coyote Optimization Algorithm (COA)
Cuckoo Search Algorithm (CSA)
Grasshopper Optimization Algorithm (GOA)
Grey Wolf Optimizer (GWO)
Harris Hawk Optimizer (HHO)
Marines Predator Algorithm (MPA)
Particle Swarm Optimizer (PSO)
Sailfish Optimizer (SFO)
Social Spider Algorithm (SSA)
Optimization Algorithm (WOA)

Table 39. Optimization results utilizing 30 epochs

	Maximum				Average				Standard Deviation			
	10	30	50	100	10	30	50	100	10	30	50	100
ABC	2.029	1.929	1.955	1.987	1.815	1.854	1.894	1.914	0.068	0.046	0.036	0.039
AGTO	***2.216***	**2.266**	**2.270**	**2.273**	**1.892**	**2.173**	**2.228**	**2.246**	0.239	0.107	0.050	0.027
ALO	2.023	2.099	2.192	2.258	1.612	1.877	1.976	2.062	0.208	0.133	0.130	0.115
AO	1.887	1.951	2.055	2.014	1.142	1.353	1.365	1.450	0.345	0.273	0.285	0.208
ARO	2.087	2.213	2.213	2.238	1.758	1.956	2.029	2.131	0.215	0.122	0.124	0.062
BA	1.036	1.115	1.115	1.115	0.754	0.860	0.901	0.943	0.112	0.110	0.095	0.084
COA	1.835	1.722	2.008	1.897	1.368	1.550	1.676	1.705	0.206	0.129	0.142	0.112
CSA	1.076	1.097	1.228	1.192	0.911	0.981	1.017	1.069	0.073	0.060	0.069	0.058
GOA	0.961	1.188	1.387	1.722	0.722	0.969	1.139	1.452	0.122	0.103	0.125	0.118
GWO	1.901	2.118	2.196	2.226	1.779	2.059	2.130	2.200	0.074	0.035	0.035	0.019
HHO	2.109	2.166	2.166	2.260	1.588	1.846	1.902	2.007	0.221	0.205	0.177	0.148
MPA	1.702	2.165	2.238	2.240	1.267	1.716	1.943	2.037	0.157	0.180	0.156	0.102
PSO	1.219	1.437	1.419	1.476	1.096	1.202	1.254	1.307	0.068	0.086	0.062	0.060
SFO	1.501	1.581	1.590	1.762	1.149	1.278	1.364	1.446	0.162	0.126	0.115	0.148
SSA	1.531	1.647	1.695	1.901	1.286	1.466	1.550	1.636	0.122	0.094	0.079	0.085
TSO	2.164	2.245	**2.270**	**2.273**	1.802	2.062	2.104	2.168	0.227	0.148	0.126	0.107
WOA	2.101	2.231	2.231	2.261	1.640	1.893	1.968	2.057	0.258	0.228	0.172	0.158

Note: The table presents the performance of each metaheuristic algorithm based on experiments conducted over 30 epochs for each population size.

Table 40. Optimization results utilizing 50 epochs

	Maximum				Average				Standard Deviation			
	10	30	50	100	10	30	50	100	10	30	50	100
ABC	2.093	2.100	2.136	2.163	**2.020**	2.053	2.071	2.084	0.044	0.024	0.025	0.025
AGTO	**2.241**	**2.272**	**2.272**	**2.273**	1.963	2.205	**2.243**	2.258	0.264	0.069	0.036	0.024
ALO	2.148	2.216	2.248	2.248	1.759	1.981	2.060	2.119	0.215	0.137	0.114	0.099
AO	1.883	2.117	2.143	1.760	1.151	1.359	1.400	1.453	0.320	0.319	0.240	0.176
ARO	2.182	2.257	2.266	2.262	1.967	2.153	2.215	2.242	0.185	0.118	0.065	0.021
BA	1.036	1.115	1.115	1.115	0.759	0.860	0.902	0.943	0.109	0.110	0.094	0.084
COA	1.863	1.952	2.084	2.063	1.530	1.718	1.785	1.828	0.187	0.127	0.132	0.112
CSA	1.090	1.215	1.313	1.203	0.937	1.013	1.047	1.088	0.075	0.059	0.092	0.057
GOA	0.970	1.235	1.450	1.824	0.734	1.013	1.209	1.551	0.122	0.106	0.136	0.134
GWO	2.102	2.243	2.255	2.266	2.013	**2.217**	**2.243**	**2.261**	0.059	0.013	0.009	0.003
HHO	2.109	2.180	2.232	2.261	1.700	1.895	2.008	2.090	0.209	0.179	0.168	0.113
MPA	1.715	2.246	2.245	2.272	1.393	1.945	2.141	2.235	0.133	0.170	0.119	0.032
PSO	1.383	1.398	1.454	1.507	1.160	1.246	1.301	1.353	0.083	0.078	0.070	0.093
SFO	1.501	1.581	1.590	1.786	1.155	1.282	1.374	1.455	0.158	0.124	0.115	0.153
SSA	1.539	1.779	1.791	1.875	1.382	1.571	1.632	1.752	0.082	0.090	0.083	0.073
TSO	2.226	2.250	2.260	2.261	1.926	2.108	2.121	2.184	0.197	0.142	0.117	0.084
WOA	2.101	2.228	2.243	**2.273**	1.650	1.897	1.970	2.077	0.245	0.185	0.183	0.151

Note: The table presents the performance of each metaheuristic algorithm based on experiments conducted over 50 epochs for each population size.

Table 41. Optimization results utilizing 100 epochs

	Maximum				Average				Standard Deviation			
	10	30	50	100	10	30	50	100	10	30	50	100
ABC	2.243	2.243	2.246	2.245	2.216	2.226	2.232	2.235	0.012	0.009	0.008	0.004
AGTO	2.260	**2.273**	**2.273**	**2.273**	2.005	2.232	2.252	2.264	0.274	0.045	0.025	0.012
ALO	2.114	2.167	2.231	2.232	1.833	2.038	2.077	2.123	0.136	0.082	0.103	0.087
AO	1.950	2.164	2.164	2.164	1.274	1.473	1.529	1.556	0.298	0.256	0.266	0.223
ARO	**2.269**	2.271	**2.273**	**2.273**	2.089	2.203	2.238	2.265	0.146	0.099	0.076	0.021
BA	1.036	1.115	1.115	1.115	0.759	0.860	0.903	0.943	0.109	0.110	0.094	0.084
COA	2.020	2.140	2.131	2.132	1.765	1.933	1.969	2.005	0.179	0.099	0.093	0.082
CSA	1.090	1.215	1.313	1.263	0.976	1.045	1.095	1.124	0.049	0.053	0.067	0.060
GOA	0.995	1.331	1.585	1.993	0.761	1.098	1.327	1.693	0.124	0.123	0.142	0.147
GWO	2.254	2.269	2.269	2.271	**2.227**	**2.265**	**2.267**	2.270	0.017	0.002	0.001	0.001
HHO	2.100	2.228	2.257	2.261	1.764	1.928	2.011	2.079	0.225	0.205	0.156	0.122
MPA	2.052	2.271	**2.273**	**2.273**	1.697	2.217	2.253	**2.271**	0.193	0.093	0.074	0.006
PSO	1.311	1.435	1.494	1.559	1.189	1.292	1.321	1.381	0.078	0.066	0.067	0.080
SFO	1.501	1.581	1.610	1.839	1.161	1.299	1.400	1.473	0.155	0.109	0.128	0.156
SSA	1.679	1.872	1.924	2.054	1.541	1.709	1.770	1.869	0.086	0.094	0.100	0.086
TSO	2.246	2.259	2.272	**2.273**	2.029	2.112	2.192	2.243	0.140	0.126	0.077	0.043
WOA	2.176	2.261	2.261	2.231	1.742	1.918	1.987	2.070	0.215	0.193	0.191	0.130

Note: The table presents the performance of each metaheuristic algorithm based on experiments conducted over 100 epochs for each population size.

4.3.2 Results

The highest Sharpe ratio recorded in the experiments was 2.273. Initially, we will examine the highest, second highest, and lowest performing algorithms based on their maximum and average scores. Ultimately, we will present a comparative analysis of the algorithms' efficacy. In our initial experiment (Table 39), we employed a total of 30 epochs. As expected, choosing a population size of 10 would likely lead to weaker results due to the limited number of agents involved. AGTO attained the highest Sharpe ratio of 2.216. TSO attained the second-highest performance with a Sharpe ratio of 2.164, and GOA recorded the lowest performance at 0.961. AGTO outperforms all other metaheuristic algorithms in average performance, with ABC attaining the second-best results. Furthermore, GOA exhibited the lowest average performance with a Sharpe ratio of 0.722.

Subsequently, we employed a population size of 30. The overwhelming majority of the algorithms exhibited superior performance, as expected. AGTO and TSO attained the highest and second-highest Sharpe ratios of 2.266 and 2.245, respectively. In this context, CSA exhibited the poorest performance. All algorithms exhibited superior average performance compared to the prior experiment. In the following instances, with population sizes of 50 and 100, AGTO and TSO attained the highest Sharpe ratio scores of 2.270 and 2.273, respectively. For a population size of 50, MPA ranks second with a

Sharpe ratio of 2.238, while for a population size of 100, WOA achieves the second-best performance with a Sharpe ratio of 2.261. Nonetheless, AGTO had greater average performance in both circumstances. In both instances, BA had the lowest performance, both in average and peak metrics.

Subsequently, we employed 50 epochs with the same population sizes (Table 40). Assigning a population size of 10 yielded a Sharpe ratio of 2.241 for AGTO. AGTO outperformed the 30 epochs with the identical population size. Unlike the prior studies, ARO ranks second. In terms of average performance, ABC and GWO received the highest scores, recording 2.020 and 2.013, respectively. GOA attained the lowest Sharpe ratio for both maximum and average performance. AGTO once more attained the greatest Sharpe ratio (2.272) utilizing populations of 30 and 50. In the preceding scenario, ARO attained the second highest Sharpe Ratio scores of 2.257 and 2.266, respectively. In these cases, BA exhibited the poorest performance, recording a Sharpe ratio of 1.115. Regarding average performance, GWO exhibited superior results, while AGTO ranked second. In both cases, BA exhibited the worse performance on average. AGTO and WOA attained the greatest Sharpe ratio of 2.273 with a population size of 100. The BA algorithm produced the poorest results. Consistent with the preceding two situations, GWO and AGTO attained the highest average performances.

In our concluding experiment, we employed a total of 100 epochs (Table 41). With the population size set to 10, we noted that ARO achieved the maximum performance with a Sharpe ratio of 2.269, while AGTO ranked second with a Sharpe ratio of 2.260. In this instance, GOA exhibited the poorest performance. Regarding average performance, GWO and ABC exhibited superior results. When the population size was fixed at 30, AGTO attained the highest Sharpe ratio of 2.273, succeeded by ARO and MPA at 2.271. In terms of average performance, GWO and AGTO exhibited superior results. Subsequently, we employed a population size of 50. AGTO, ARO, and MPA attained a Sharpe ratio of 2.273, closely followed by TSO at 2.272, while BA recorded the lowest Sharpe ratio of 1.115. GWO and AGTO had superior performance on average. With a population size of 100, four metaheuristic algorithms attained the highest Sharpe ratio: AGTO, ARO, MPA, and TSO, while GWO secured the second-best result. Regarding average performance, MPA achieved the highest Sharpe ratio of 2.271, while GWO followed closely with a Sharpe ratio of 2.270. In the context of 100 epochs, BA had the poorest performance across all instances regarding both the maximum and average Sharpe ratio.

4.3.3 Statistical analysis

To strengthen the validity of our findings, we performed the Wilcoxon sum-rank test, with the corresponding results displayed in Table 42. In 192 comparisons, AGTO had greater performance in 175 instances relative to the other algorithms. Utilizing 30 epochs revealed statistically significant differences in nearly all instances between the performance of AGTO and the other algorithms. The sole exception arose with a population size of 10, where no statistically significant difference was detected between AGTO and TSO.

With 50 epochs and a population size of 10, we found that no definitive conclusions could be made about the superiority of AGTO compared to ABC, ARO, GWO, and TSO. Moreover, no conclusive results were noted for the statistical efficacy of AGTO in comparison to GWO for population sizes of 50 and 100.In the context of 100 epochs, a definitive conclusion about the comparative performance of AGTO, ARO, GWO, and MPA in terms of statistical significance could not be established. As indicated in Tables 39-41, AGTO attained the greatest Sharpe ratio in most situations, highlighting its strong overall performance. Nonetheless, as will be examined in the subsequent part, GWO, TSO, ARO, MPA, and ABC showed solid performance in specific scenarios, underscoring their competitiveness.

Table 42. Wilcoxon Rank-Sum Test Results per Epoch and Population Size

	30				**50**				**100**			
	10	**30**	**50**	**100**	**10**	**30**	**50**	**100**	**10**	**30**	**50**	**100**
	p-value				**p-value**				**p-value**			
ABC	1.2E-02	3.3E-10	2.9E-11	2.9E-11	8.4E-01	5.8E-10	2.9E-11	2.9E-11	5.3E-06	6.8E-03	9.2E-06	3.3E-10
ALO	7.0E-06	1.5E-09	2.3E-10	5.3E-10	1.5E-04	1.3E-08	1.6E-09	2.5E-10	1.1E-04	5.8E-11	8.6E-11	3.2E-11
AO	1.6E-09	3.5E-11	2.9E-11	2.9E-11	7.0E-10	3.9E-11	3.5E-11	2.9E-11	2.3E-09	5.8E-11	2.9E-11	2.9E-11
ARO	5.7E-03	4.5E-08	1.1E-09	3.6E-10	5.7E-01	4.8E-02	8.8E-04	2.4E-05	3.7E-01	3.1E-01	8.4E-01	1.7E-01
BA	3.2E-11	2.9E-11	2.9E-11	2.9E-11	3.2E-11	2.9E-11	2.9E-11	2.9E-11	3.2E-11	2.9E-11	2.9E-11	2.9E-11
COA	2.8E-09	2.9E-11	2.9E-11	2.9E-11	2.5E-08	2.9E-11	2.9E-11	2.9E-11	1.1E-05	3.2E-11	2.9E-11	2.9E-11
CSA	3.5E-11	2.9E-11	2.9E-11	2.9E-11	3.9E-11	2.9E-11	2.9E-11	2.9E-11	6.4E-11	2.9E-11	2.9E-11	2.9E-11
GOA	2.9E-11	2.9E-11	2.9E-11	2.9E-11	2.9E-11	2.9E-11	2.9E-11	2.9E-11	2.9E-11	2.9E-11	2.9E-11	2.9E-11
GWO	4.3E-04	1.5E-06	3.5E-08	5.3E-10	6.7E-01	4.4E-02	8.6E-02	2.7E-01	1.7E-06	5.8E-02	1.8E-01	1.8E-01
HHO	3.7E-06	7.4E-09	7.0E-11	6.4E-10	1.3E-05	4.0E-10	7.0E-10	1.7E-10	3.7E-05	4.8E-10	4.4E-10	7.8E-11
MPA	7.7E-10	3.0E-10	2.3E-10	1.0E-10	1.6E-09	1.6E-08	1.1E-07	1.7E-04	1.9E-06	2.4E-01	2.3E-01	5.9E-01
PSO	4.0E-10	2.9E-11	2.9E-11	2.9E-11	5.3E-10	2.9E-11	2.9E-11	2.9E-11	4.8E-10	2.9E-11	2.9E-11	2.9E-11
SFO	4.0E-10	2.9E-11	2.9E-11	2.9E-11	4.4E-10	2.9E-11	2.9E-11	2.9E-11	4.8E-10	2.9E-11	2.9E-11	2.9E-11
SSO	9.3E-10	2.9E-11	2.9E-11	2.9E-11	2.1E-09	2.9E-11	2.9E-11	2.9E-11	1.4E-08	2.9E-11	2.9E-11	2.9E-11
TSO	9.8E-02	2.5E-04	4.0E-06	3.3E-04	1.8E-01	7.5E-04	6.3E-07	4.5E-08	4.8E-01	2.4E-06	2.7E-05	4.1E-04
WOA	8.9E-05	1.5E-07	2.1E-09	1.6E-08	7.5E-06	1.2E-09	5.8E-10	8.1E-09	2.1E-05	2.3E-09	1.6E-09	2.9E-11

Note: The table presents the p-values of Wilcoxon Rank-Sum Test of each metaheuristic versus AGTO for each combination of epoch and population sizes.

4.3.4 Comparative analysis of algorithms' performance

The findings show that five of the evaluated algorithms failed to achieve a Sharpe ratio above 2 in any scenario. — BA, CSA, GOA, PSO, SFO. Furthermore, SSA successfully exceeded a Sharpe ratio above 2, solely in the most computationally demanding case of 100 epochs and population size. The six previously mentioned algorithms showed weaker average performance. Notably, while AO reached a high maximum Sharpe ratio, its average performance was relatively weak. ABC and ALO demonstrated consistent performance throughout many epochs and population sizes. Overall, ABC outperformed ALO alone in the context of 100 epochs across all population size combinations and in the instance of 30 epochs with 10 population size.

In the other instances, ALO demonstrated superiority. The HO demonstrated stability in outcomes across all combinations of epochs and population size. Nonetheless, performance exhibited an average deficiency in stability. Conversely, WOA seems to surpass HHO in most cases regarding the attainment of the maximum Sharpe ratio. However, regarding average performance, HHO somewhat surpasses WOA. Concerning GWO, it demonstrated reliable performance. Nonetheless, it never attained the optimal Sharpe ratio. In contrast, regarding average performance, it exhibited consistent results and surpassed the other algorithms in the majority of instances. The other algorithms attained a maximum Sharpe ratio of 2.273. Upon selecting 100 epochs, MPA attained the optimal Sharpe ratio in two instances. However, with a population size of 10, MPA exhibited inferior performance relative to AGTO, ARO, and TSO. Setting the epoch number and population size at 100 yielded the optimal average performance across all methods.

In our examination of ARO and TSO performance, it is evident that TSO surpassed ARO in achieving the highest Sharpe ratio while employing 30 epochs. This tendency was also found with 50 epochs and a population size of 10. With the exception of cases involving 100 epochs and 100 population sizes, where a tie was noted, ARO surpassed TSO in all other instances. Regarding average performance, TSO exceeded only ARO in the scenario of 30 epochs. Notably, TSO equaled AGTO in scenarios involving 30 epochs and population sizes of 50 and 100, with both algorithms achieving the optimal Sharpe ratio. Ultimately, it is evident that AGTO demonstrated the most consistent performance. In all instances, except for 100 epochs and a population size of 10, ARO exceeded it. AGTO not only attained superior performance but also consistently produced a Sharpe ratio over 2,260, except for 30 epochs and 10 population sizes. In this case, the performance exhibited potential. AGTO delivered the most consistent average performance across 30 epochs. In the remaining scenarios, it still performed well; yet it was marginally inferior to other methods.

Although AGTO was initially suggested as the method for portfolio optimization, it would be unfair to overlook the noteworthy performance of alternative metaheuristic algorithms. Such an example is TSO, which, as far as we know, is the first one that has been used for the concept of portfolio optimization. It is evident that certain cases were matched with AGTO. Nevertheless, the mean performance was inferior to that of AGTO. Furthermore, both ARO and MPA had remarkable performances. As far as we know, the application of these algorithms in portfolio optimization is limited. Furthermore, the findings indicated that the widely utilized GWO performs well and offers greater stability on average. However, none of the examples succeeded in achieving the optimal Sharpe ratio.

In conclusion, AGTO exhibited optimal performance in all instances, apart from one. Generally, one of the most formidable challenges that a practitioner or researcher may encounter is to address the trade-off between the number of epochs and population sizes. AGTO has reduced sensitivity to the selection of these components, as indicated by the attainment of the greatest Sharpe ratio in five out of twelve instances and a Sharpe ratio of no less than 2.270 in eight instances. Literature typically refrains from employing a sample size of 10. We have acquired significant insights on the performance of the assessed models. To that end, the results demonstrate that indeed the AGTO can be considered a viable algorithm for solving the case of optimal portfolio selection, since a researcher or practitioner can obtain optimal results across various combinations of epochs and population sizes.

4.4 Stock Price Forecasting

In this section, we present the evaluation criteria which are being used in order to evaluate a time series model. Next, we present the dataset as well as the experimental setup. Finally, we present the results, where we discuss the performance of the evaluated models.

4.4.1 Evaluation Criteria

The Mean Squared Error (MSE) (Equation 50) is defined as the average of the squared differences between the actual and predicted values. MSE penalizes large variations between actual and predicted values by squaring the differences, hence emphasizing large deviations. Furthermore, MSE is A differentiable function, which makes it beneficial for gradient-based optimization techniques, such as NN models. While the values of MSE provide useful insights into our model's performance, RMSE enhances the model’s explainability. The RMSE represents the difference between actual stock prices and predicted values. Hence, RMSE makes the model’s performance easier to interpret. The Mean Absolute Error (MAE) (Equation 51) is a widely utilized statistic that measures the mean of the absolute discrepancies between predicted and actual values. Unlike MSE, it shows reduced sensitivity to outliers. Mean Absolute Percentage Error (MAPE) (Equation 52) is a another widely accepted evaluation metric for regression tasks. MAPE estimates the model's accuracy as a percentage, which is especially useful when evaluating the model's performance over time series with varying price ranges. However, MAPE might be misleading when the actual values are near zero, such as stocks with very low closing prices or stock returns. Nonetheless, in our experiments, we have not included such stocks; hence, it can still be regarded as a reliable metric.

$$\text{MSE}=\frac{1}{N}(\sum_i^N(y_i-\hat{y}_i)^2 \quad (50)$$

$$\text{MAE}=\frac{1}{N}(\sum_i^N |y_i-\hat{y}_i|) \quad (51)$$

$$\text{MAPE}=\frac{100}{N}\sum_i^N \frac{y_i-\hat{y}_i}{y_i} \quad (52)$$

4.4.2 Experimental Setup

The dataset comprises the twenty S&P 500 stocks with the biggest market capitalization (Table 42). The data was obtained from Yahoo Finance, and the associated descriptive statistics are presented in Table 43. In accordance with Gülmez (2023), we employed a five-year dataset, covering the period from 1st July 2019 to 1st July 2024, and designated a lookback value of 20 as input. Additionally, we evaluate the performance of GJO against PSO and GA, as these algorithms are well-established in the literature and have been extensively employed for several optimization tasks, including also hyperparameter selection.

The MEALPY package (Van Thieu & Mirjalili, 2023) supports the implementation of these particular algorithms. We selected a modest quantity of epochs and a population size, both established at 30. Generally, the literature typically uses higher values for both parameters throughout most research. Our primary objective is to assess the efficacy of our suggested method in a more competitive and confined environment, ensuring its effectiveness despite low computational resources or restricted timeframe. We used min-max scaling for data transformation, which is a widely accepted approach for the task of stock price prediction. We employed the TensorFlow 2.0 (Abadi *et al.*, 2016) framework for the development of LSTM and GRU networks, and utilized Scikit-learn (Pedregosa *et al.*, 2011) for data preprocessing. Furthermore, we utilize early halting to alleviate the overfitting phenomenon.

The optimized models can choose up to two layers, which may be GRU or LSTM. Moreover, we incorporate Dropout layers to tackle the risk of overfitting. Ultimately, another crucial aspect of our process is the optimization of batch size, which can significantly influence the model's performance. Most prior studies fail to account for the optimization of this component. For optimizer selection, we consider the RMSProp,Adam, and the recently introduced RAdam, which has shown promising outcomes (Gkonis & Tsakalos, 2024). Table 44 delineates the components designated for optimization, and the candidate values. Furthermore, we incorporate various benchmark models with fixed initializations. Table 45 describes their architectures. Table 46 displays the outcomes optimized by metaheuristics, whereas Tables 47–49 present the benchmark results; we will examine these findings in the subsequent subsections. The optimization was conducted on the MSE; however, we choose to describe the results based on the RMSE, as it is more interpretable for financial time series, as noted in subsection 4.1. Therefore, presenting both MSE and RMSE tables may be redundant and would not contribute any more information. Figures (27-29) illustrate the timeseries of metaheuristic-optimized models, demonstrating that in the majority of instances, the projected closing prices align closely with the actual values. We have excluded the figures of the benchmark models to prevent the presentation of almost four hundred line graphs, which may overwhelm the viewer and lead to confusion. Additionally, apart from RMSE, we also provide the MAE and MAPE measures.

Table 42.

The twenty stocks evaluated in this study and their respective tickers

Stock	Ticker
Alphabet (Class A)	GOOGL
Amazon	AMZN
Apple	AAPL
Berkshire Hathaway (Class B)	BRK.B
Broadcom.	AVGO
Costco Wholesale	COST
Eli Lilly & Co.	LLY
Exxon Mobil	XOM
Home Depot	HD
JPMorgan Chase & Co.	JPM
Mastercard	MA
Meta Platforms (Class A)	META
Microsoft	MSFT
Netflix	NFLX
Nvidia	NVDA
Procter & Gamble	PG
Tesla	TSLA
United health Group	UNH
Visa	V
Walmart	WMT

Table 43. Summary Statistics

Asset	Mean	Std. Dev.	Min	Max
Alphabet (Class A)	107.67	31.10	52.58	185.16
Amazon	136.24	31.35	81.82	197.85
Apple	134.90	42.39	46.59	216.42
Berkshire Hathaway (Class B)	284.30	64.34	162.13	420.52
Broadcom	55.86	31.56	14.61	181.71
Costco Wholesale	442.84	141.24	240.89	867.40
Eli Lilly & Co.	308.66	196.76	99.13	907.77
Exxon Mobil	70.38	27.73	24.81	118.21
Home Depot	275.88	51.37	135.78	388.06
JPMorgan Chase & Co.	128.57	28.78	69.26	203.66
Mastercard	345.93	53.70	197.67	485.88
Meta (Class A)	264.15	94.67	88.64	526.32
Microsoft	260.10	79.73	125.86	452.04
Netflix	425.65	127.97	166.37	691.69
Nvidia	25.90	24.26	3.70	135.57
Procter & Gamble	130.25	17.11	86.34	165.56
Tesla	187.36	95.93	14.09	409.97
United health Group	400.81	102.97	181.84	542.60
Visa	211.91	29.69	131.00	288.22
Walmart	45.46	6.94	32.18	68.55

Table 44. Hyperparameter Search Space

Architecture	Candidate Values
GRU | LSTM	[1-50]
Dropout rate	[0.2-0.7]
GRU | LSTM (*optional*)	[1-50]
Dropout rate (*optional*)	[0.2-0.7]
Batch Size	[4,50]
Optimizer	{Adam, RAdam RMSProp}

Table 45. Benchmark Model Description

Model	Configuration	Hyperparameters
GRU	GRU	10
	Dropout rate	0.2
	Optimizer	{Adam, RAdam RMSProp}
LSTM	LSTM	10
	Dropout rate	0.2
	Optimizer	{Adam, RAdam RMSProp}
GRU2D	GRU	10
	Dropout rate	0.2
	GRU	10
	Dropout rate	0.2
	Optimizer	{Adam, RAdam RMSProp}
LSTM2D	LSTM	10
	Dropout rate	0.2
	LSTM	10
	Dropout	0.2
	Optimizer	{Adam, RAdam RMSProp}
GRU-LSTM	GRU	10
	Dropout rate	0.2
	LSTM	10
	Dropout rate	0.2
	Optimizer	{Adam, RAdam RMSProp}
LSTM-GRU	LSTM	10
	Dropout rate	0.2
	GRU	10
	Dropout rate	0.2
	Optimizer	{Adam, RAdam RMSProp}

Note. All models were trained with a batch size of 16.

Table 46. Performance results metaheuristic optimized models per asset

	GJO-GRU | LSTM			**PSO-GRU | LSTM**			**GA-GRU | LSTM**		
Stock	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE
AAPL	**2.65**	1.94	1.06	2.70	1.97	1.07	3.84	2.90	1.58
AMZN	**3.04**	2.32	1.49	3.24	2.49	1.59	3.13	2.40	1.53
AVGO	**3.75**	2.53	2.13	3.90	2.60	2.20	4.45	2.94	2.48
BRK-B	**3.43**	2.73	0.72	3.95	3.10	0.81	3.69	2.93	0.77
COST	**14.30**	11.63	1.69	16.64	13.44	1.96	16.38	13.46	1.99
GOOGL	3.29	2.55	1.74	**2.66**	1.90	1.32	3.39	2.62	1.81
HD	**4.70**	3.64	1.12	4.75	3.74	1.15	4.72	3.66	1.12
JPM	3.09	2.29	1.38	3.30	2.62	1.49	**2.31**	1.75	1.02
LLY	23.26	19.09	2.79	21.16	16.27	2.40	**14.87**	11.63	1.84
MA	**6.71**	5.57	1.28	11.05	9.26	2.13	6.77	5.50	1.29
META	**9.46**	6.21	1.61	10.89	7.60	1.95	10.71	7.44	1.90
MSFT	**5.12**	4.07	1.08	6.81	5.70	1.48	5.97	4.89	1.29
NFLX	12.43	8.72	1.75	**11.68**	8.16	1.61	12.10	8.44	1.70
NVDA	4.40	2.84	3.48	3.51	2.31	3.20	**3.21**	2.24	3.28
PG	**1.31**	0.98	0.65	1.47	1.10	0.73	1.46	1.13	0.75
TSLA	**6.98**	5.07	2.40	6.99	5.14	2.45	7.54	5.65	2.68
UNH	8.50	6.17	1.26	**7.71**	5.46	1.11	8.73	6.47	1.31
V	2.87	2.36	0.91	**2.31**	1.87	0.73	4.84	4.13	1.59
WMT	**0.80**	0.56	0.99	0.87	0.64	1.10	0.85	0.62	1.09
XOM	**1.48**	1.17	1.12	1.93	1.52	1.45	1.69	1.34	1.28

Note. MAPE values are reported as percentages

Table 47. Benchmark model results (RMSE)

Stock	GRU	GRU2D	GRU-LSTM	LSTM	LSTM2D	LSTM-GRU
RMSProp						
AAPL	5.57	13.87	7.91	4.52	10.98	7.42
AMZN	3.83	5.55	9.47	3.91	8.22	4.12
AVGO	21.29	23.88	17.23	18.97	33.67	14.41
BRK-B	9.68	17.90	15.27	11.74	15.16	12.20
COST	33.42	105.74	73.25	56.83	71.48	43.62
GOOGL	5.71	9.57	6.94	4.50	8.79	6.40
HD	6.68	12.13	19.77	6.21	10.04	8.96
JPM	11.50	6.53	8.65	3.27	7.56	4.58
LLY	108.18	158.70	195.28	90.87	180.38	207.50
MA	7.06	16.43	28.96	8.22	24.22	16.13
META	19.09	22.52	33.07	42.21	35.81	16.48
MSFT	22.39	31.69	30.21	31.43	27.85	20.72
NFLX	19.60	21.04	20.73	19.10	22.50	24.63
NVDA	25.55	25.66	21.26	22.10	25.07	24.72
PG	2.22	2.48	4.58	2.86	2.65	2.22
TSLA	10.12	11.95	10.63	9.79	15.16	10.33
UNH	13.26	18.70	17.51	11.88	17.72	14.28
V	9.98	9.80	10.57	7.82	4.56	4.50
WMT	2.06	6.47	3.91	1.73	2.72	2.52
XOM	3.16	3.48	2.43	2.84	4.57	2.48
Adam						
AAPL	6.45	8.19	9.78	5.69	8.47	9.72
AMZN	4.24	6.47	4.95	4.48	7.24	7.11
AVGO	11.08	29.75	15.01	25.58	29.36	17.52
BRK-B	13.06	12.77	24.07	17.14	17.06	18.04
COST	110.77	65.29	83.51	86.06	51.48	118.20
GOOGL	7.28	11.09	7.17	5.19	7.31	7.31
HD	18.71	11.22	8.99	11.18	12.04	11.22
JPM	9.48	9.89	6.55	11.76	8.28	7.12
LLY	198.04	252.88	124.19	154.30	237.30	158.05
MA	17.27	17.34	23.76	20.82	22.65	14.39
META	14.04	35.36	29.51	28.39	31.25	40.38
NFLX	26.84	52.03	20.27	19.87	21.59	24.32
MSFT	23.54	31.87	35.69	34.79	25.24	21.38
NVDA	22.09	29.68	24.59	25.75	32.33	25.09
PG	1.99	3.82	2.98	2.86	3.33	3.45
TSLA	8.82	15.20	8.94	10.10	11.99	13.32
UNH	20.99	15.62	15.24	13.03	14.32	16.15
V	6.38	7.43	9.70	6.45	4.91	12.94
WMT	6.42	3.26	4.55	2.82	5.66	4.55
XOM	2.97	3.33	2.42	2.94	3.29	4.40
Radam						
AAPL	6.84	10.48	14.19	6.33	10.50	7.24

AMZN	5.37	8.21	4.00	6.08	9.27	7.93
AVGO	36.88	24.94	22.52	26.70	25.07	19.45
BRK-B	14.28	19.01	21.41	12.45	17.98	8.60
COST	143.90	141.37	36.58	45.56	118.48	36.93
GOOGL	14.20	9.93	8.04	4.74	8.12	6.92
HD	10.34	8.85	10.26	12.72	22.45	9.27
JPM	5.63	25.60	6.43	2.84	9.96	5.06
LLY	89.02	205.46	125.04	172.19	181.77	118.15
MA	20.13	27.83	10.16	16.06	15.37	15.40
META	41.29	19.16	18.29	14.30	61.02	17.14
MSFT	13.51	30.31	22.84	18.29	25.21	14.97
NFLX	22.73	17.53	20.72	24.58	21.04	17.68
NVDA	27.35	30.77	16.17	14.09	25.59	25.34
PG	4.51	2.41	1.76	3.14	2.67	4.86
TSLA	11.81	11.52	10.19	10.02	12.24	10.16
UNH	15.96	14.51	15.55	14.34	17.25	10.97
V	6.97	8.39	12.63	6.87	12.62	5.71
WMT	4.24	2.74	1.98	2.01	4.16	1.01
XOM	3.62	3.91	2.89	2.79	3.82	2.48

Table 48. Benchmark model results (MAE)

Stock	GRU	GRU2D	GRU-LSTM	LSTM	LSTM2D	LSTM-GRU
RMSProp						
AAPL	4.27	11.26	6.19	3.43	8.66	5.82
AMZN	3.01	4.51	8.50	3.05	6.99	3.18
AVGO	16.74	18.34	12.79	14.13	27.99	10.45
BRK-B	8.15	15.04	12.68	9.62	12.96	10.09
COST	26.31	85.12	56.45	42.98	55.95	33.97
GOOGL	4.57	7.49	5.54	3.54	7.02	5.18
HD	5.35	10.21	16.89	4.96	8.21	7.30
JPM	9.21	5.25	6.81	2.61	6.02	3.84
LLY	89.23	135.24	172.89	78.56	156.87	183.13
MA	5.73	13.98	24.79	6.63	20.57	13.70
META	14.86	18.13	25.05	31.18	27.70	11.69
MSFT	18.51	27.19	25.35	25.81	23.00	16.83
NFLX	15.39	16.46	16.43	15.08	17.55	20.38
NVDA	18.39	17.11	14.40	14.41	17.64	16.03
PG	1.70	2.03	3.80	2.36	2.19	1.76
TSLA	7.66	9.35	8.34	7.57	11.90	7.91
UNH	10.57	15.66	14.84	9.13	15.00	11.70
V	8.65	8.42	9.18	6.75	3.65	3.56
WMT	1.64	5.13	3.03	1.24	1.95	1.91
XOM	2.54	2.79	1.97	2.34	3.59	2.01
Adam						

AAPL	5.10	6.37	7.62	4.47	6.59	7.53
AMZN	3.26	5.31	4.02	3.59	6.06	5.91
AVGO	7.55	23.89	10.94	20.81	24.77	12.70
BRK-B	11.06	10.91	20.48	14.18	13.92	14.95
COST	84.53	52.18	65.19	65.35	41.31	92.96
GOOGL	5.69	8.17	5.51	4.21	5.89	5.92
HD	15.73	9.33	7.40	9.29	10.22	9.11
JPM	7.54	8.08	5.46	9.15	6.88	6.03
LLY	174.14	228.78	104.66	128.83	211.10	134.40
MA	14.73	15.10	19.92	17.86	19.65	12.11
META	9.89	26.96	22.81	21.98	23.95	31.28
NFLX	21.90	42.42	15.65	15.45	17.23	18.95
MSFT	20.03	26.74	30.08	28.83	21.00	17.65
NVDA	14.58	21.60	16.36	17.20	23.38	17.27
PG	1.56	3.19	2.48	2.39	2.81	2.89
TSLA	6.66	12.00	6.79	7.77	9.39	10.60
UNH	17.77	12.90	12.65	10.33	11.63	13.54
V	5.48	6.54	8.52	5.61	4.04	11.29
WMT	4.86	2.43	3.60	2.26	4.49	3.65
XOM	2.37	2.65	1.97	2.44	2.66	3.58
Radam						
AAPL	5.47	8.16	12.40	4.95	8.18	5.68
AMZN	4.40	6.88	3.13	5.04	7.86	6.71
AVGO	30.15	19.74	17.88	19.74	19.81	14.90
BRK-B	11.83	15.93	18.71	10.47	15.30	6.96
COST	117.44	116.83	27.89	35.46	96.04	26.78
GOOGL	12.32	7.57	6.53	3.68	6.51	5.57
HD	8.31	7.20	8.56	10.63	19.05	7.73
JPM	4.80	21.42	5.23	2.13	7.93	4.22
LLY	73.24	178.10	104.79	147.48	157.97	96.41
MA	17.12	24.62	8.58	13.46	13.36	13.24
META	31.72	14.36	14.05	10.54	46.29	12.86
MSFT	10.91	25.38	19.02	15.11	21.15	12.19
NFLX	18.34	13.15	16.12	20.32	15.91	13.40
NVDA	18.93	22.46	10.51	8.47	17.61	17.39
PG	3.73	1.94	1.37	2.57	2.20	4.05
TSLA	9.27	9.09	7.78	7.66	9.67	7.66
UNH	13.08	12.05	13.04	11.75	13.50	8.55
V	5.89	7.21	11.06	6.10	11.12	4.85
WMT	3.16	2.08	1.39	1.52	3.05	0.72
XOM	2.89	3.13	2.35	2.23	3.04	2.04

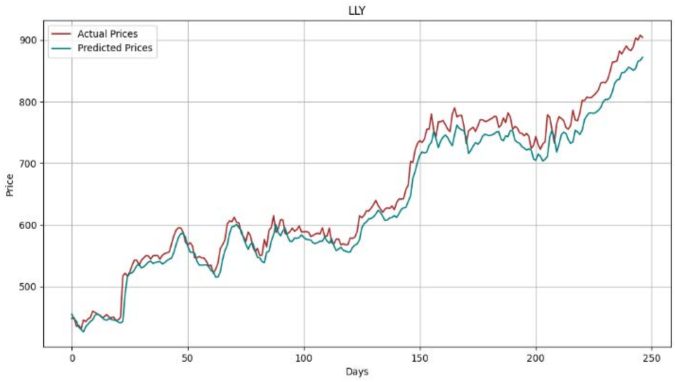
Table 49. Benchmark model results (MAPE)

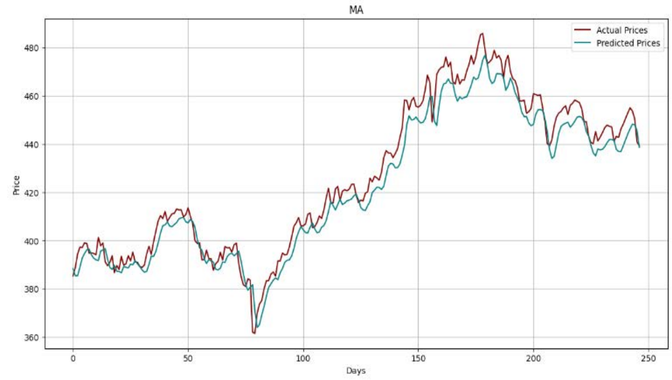
Stock	GRU	GRU2D	GRU-LSTM	LSTM	LSTM2D	LSTM-GRU
RMSProp						
AAPL	2.31	5.93	3.27	1.87	4.55	3.08
AMZN	1.90	2.76	5.33	1.93	4.27	2.04
AVGO	13.49	14.54	10.04	11.07	22.96	8.23
BRK-B	2.11	3.84	3.23	2.48	3.33	2.59
COST	3.69	11.91	7.80	5.92	7.77	4.75
GOOGL	3.04	4.90	3.66	2.39	4.62	3.46
HD	1.64	3.11	5.01	1.52	2.52	2.24
JPM	5.10	2.96	3.77	1.51	3.35	2.22
LLY	12.50	19.11	24.83	11.23	22.36	26.27
MA	1.34	3.18	5.60	1.55	4.65	3.12
META	3.57	4.60	5.73	6.93	6.35	3.01
MSFT	4.64	6.85	6.35	6.44	5.77	4.26
NFLX	2.96	3.18	3.22	3.07	3.67	3.98
NVDA	22.35	19.60	16.72	16.28	21.08	18.02
PG	1.14	1.33	2.46	1.55	1.44	1.17
TSLA	3.59	4.40	3.88	3.55	5.66	3.73
UNH	2.15	3.13	2.96	1.86	3.01	2.36
V	3.28	3.19	3.47	2.56	1.40	1.40
WMT	2.78	8.67	5.10	2.09	3.27	3.22
XOM	2.37	2.61	1.89	2.19	3.32	1.93
Adam						
AAPL	2.71	3.37	4.00	2.39	3.48	3.96
AMZN	2.10	3.31	2.51	2.27	3.72	3.65
AVGO	5.95	19.30	8.61	16.95	20.53	9.92
BRK-B	2.85	2.82	5.22	3.61	3.54	3.81
COST	11.62	7.28	9.03	8.98	5.80	12.89
GOOGL	3.76	5.26	3.61	2.83	3.90	3.94
HD	4.68	2.86	2.26	2.80	3.08	2.81
JPM	4.21	4.53	3.09	5.05	3.90	3.43
LLY	24.95	33.21	14.75	18.05	30.39	18.99
MA	3.35	3.45	4.50	4.05	4.47	2.78
META	2.63	6.17	5.24	5.06	5.50	7.14
NFLX	4.30	7.66	3.26	3.03	3.47	3.99
MSFT	5.05	6.71	7.53	7.19	5.29	4.46
NVDA	16.68	26.49	18.74	19.74	28.43	20.35
PG	1.04	2.08	1.62	1.57	1.83	1.89
TSLA	3.14	5.65	3.20	3.72	4.39	4.94
UNH	3.54	2.62	2.55	2.10	2.36	2.73
V	2.14	2.50	3.22	2.13	1.57	4.26
WMT	8.13	4.07	6.07	3.84	7.58	6.18
XOM	2.24	2.50	1.90	2.29	2.49	3.33
Radam						
AAPL	2.92	4.28	6.59	2.62	4.30	3.01

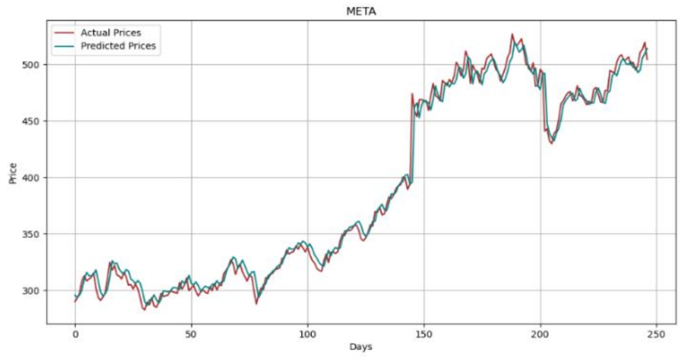
AMZN	2.71	4.19	1.99	3.11	4.75	4.09
AVGO	24.54	15.87	14.41	15.40	15.92	11.89
BRK-B	3.02	4.06	4.80	2.69	3.92	1.82
COST	16.50	16.48	3.90	4.96	13.46	3.70
GOOGL	8.18	4.92	4.29	2.52	4.29	3.71
HD	2.57	2.21	2.57	3.18	5.63	2.34
JPM	2.76	11.95	2.94	1.29	4.40	2.42
LLY	10.25	25.33	14.71	20.88	22.50	13.42
MA	3.88	5.59	1.97	3.05	3.06	3.02
META	7.18	3.75	3.43	2.72	10.32	3.22
MSFT	2.79	6.37	4.78	3.81	5.32	3.13
NFLX	3.68	2.73	3.37	4.01	3.32	2.74
NVDA	22.39	27.51	12.00	9.48	20.65	20.36
PG	2.42	1.28	0.91	1.70	1.45	2.62
TSLA	4.31	4.22	3.65	3.66	4.47	3.61
UNH	2.68	2.42	2.62	2.38	2.78	1.73
V	2.25	2.72	4.17	2.34	4.20	1.85
WMT	5.28	3.50	2.36	2.57	5.09	1.24
XOM	2.71	2.91	2.20	2.12	2.84	1.95

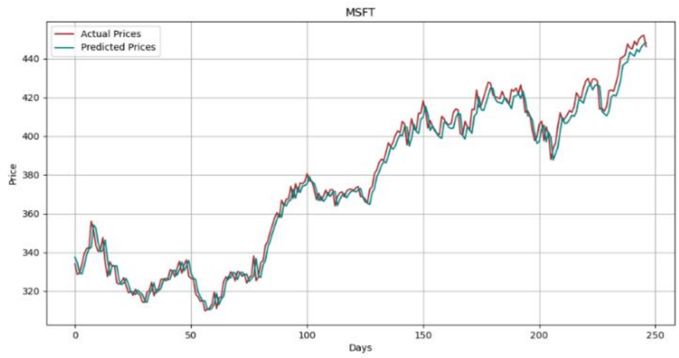
Note. MAPE values are reported as percentages

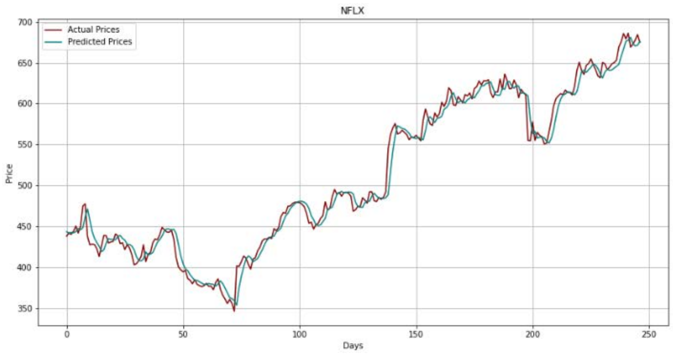
AAPL
AMZN
AVGO
BRK-B
COST
GOOGL
HD
JPM
Actual Prices
Predicted Prices
Days
Price

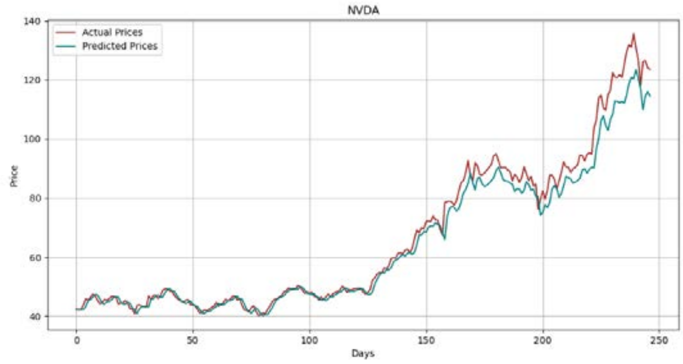

LLY
Actual Prices
Predicted Prices
Price
Days

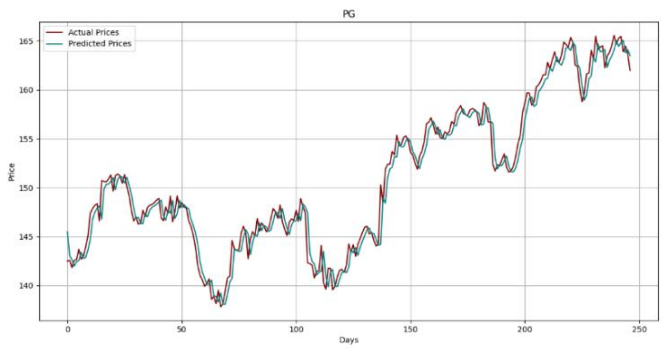

MA
Actual Prices
Predicted Prices
Price
Days

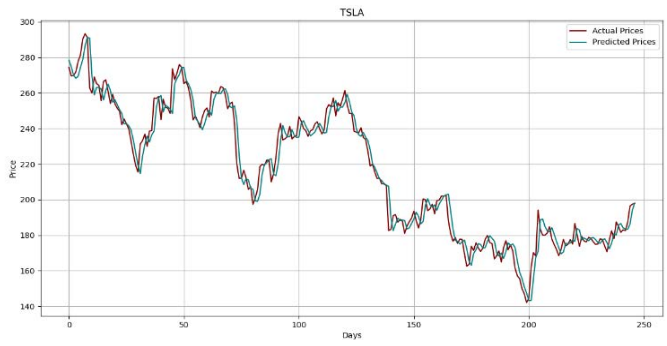

META
Actual Prices
Predicted Prices
Price
Days


MSFT
Actual Prices
Predicted Prices
Price
Days


NFLX
Actual Prices
Predicted Prices
Price
Days


NVDA
Actual Prices
Predicted Prices
Price
Days


PG
Actual Prices
Predicted Prices
Price
Days


TSLA
Actual Prices
Predicted Prices
Price
Days

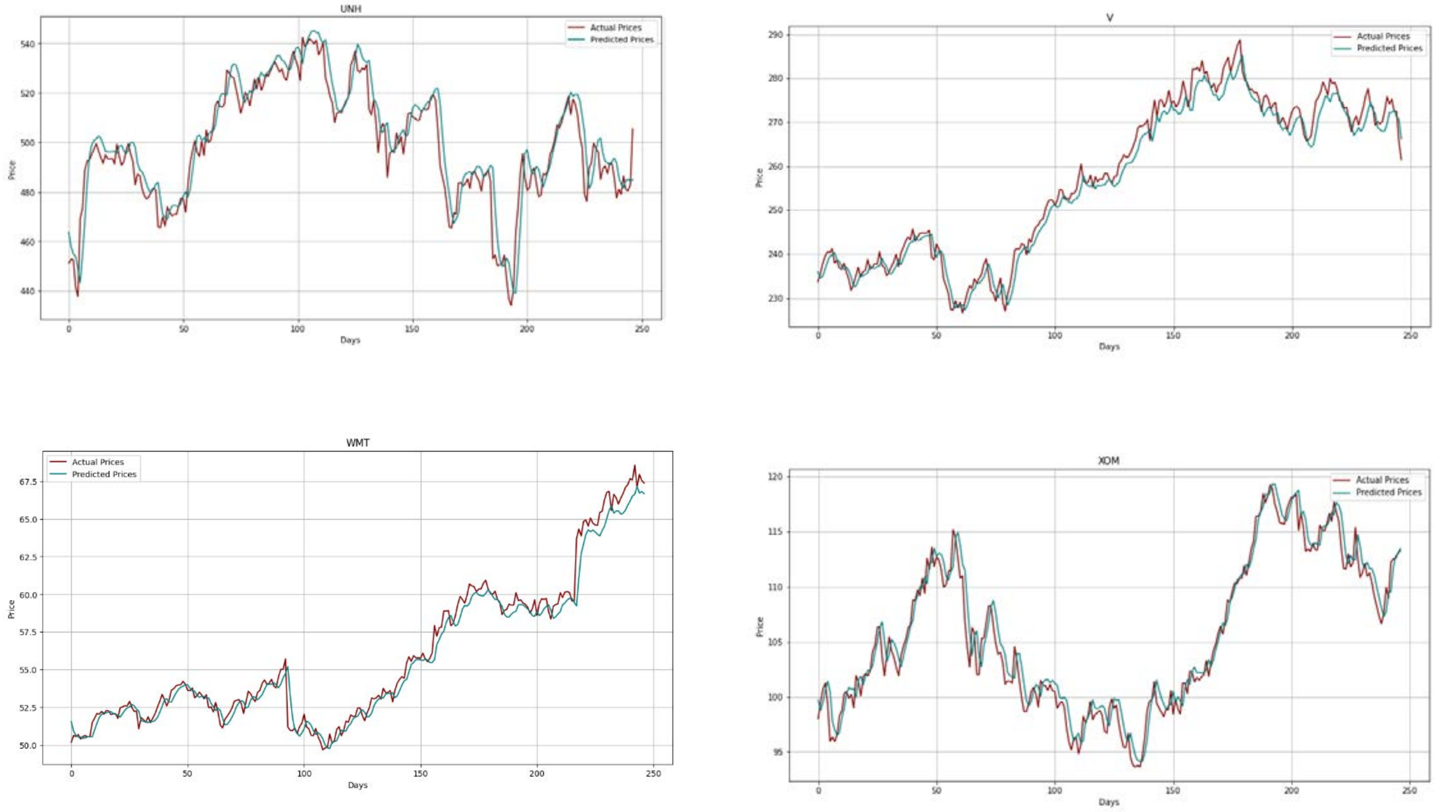
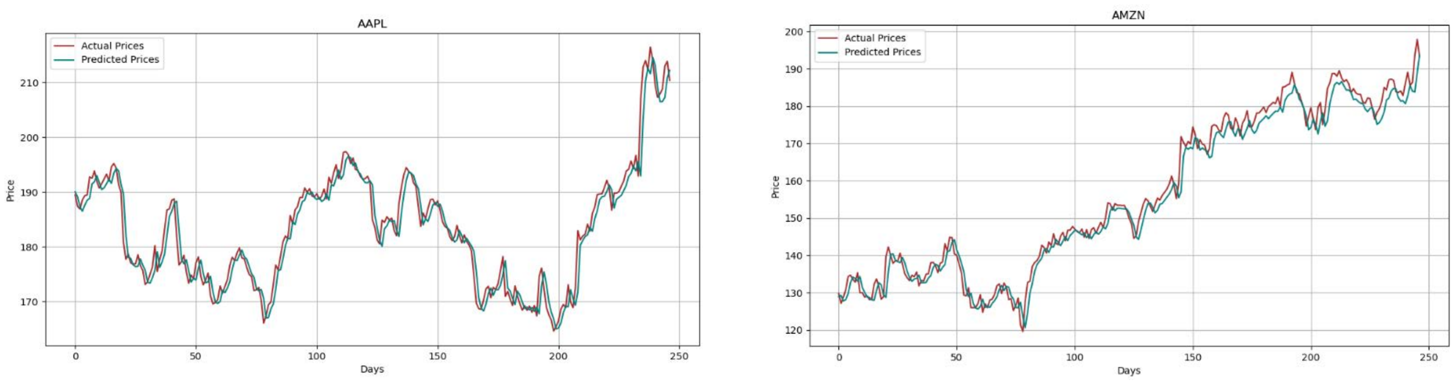
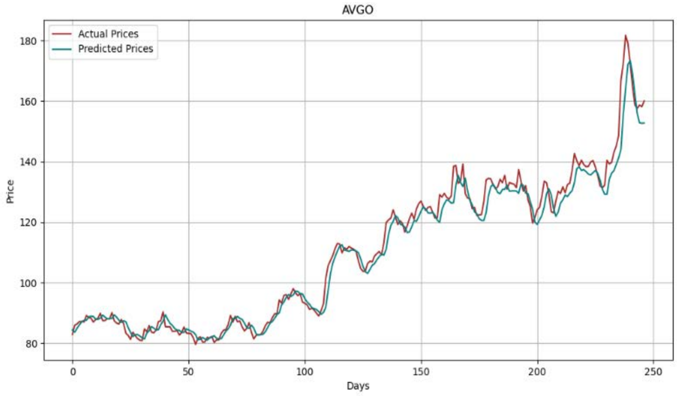
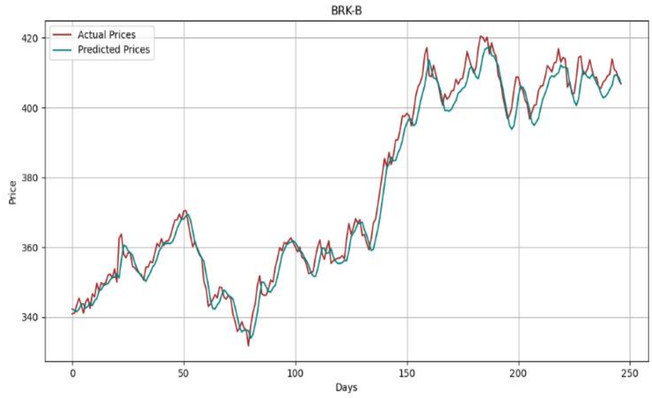
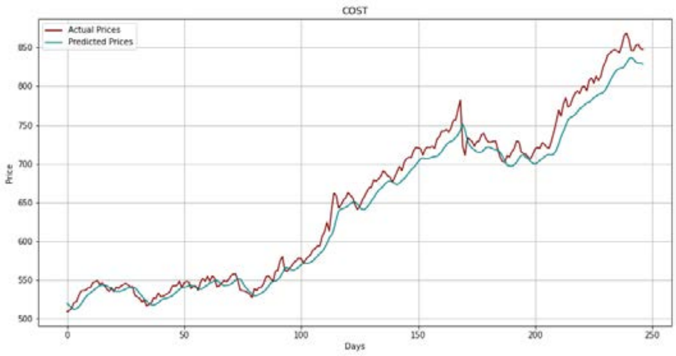


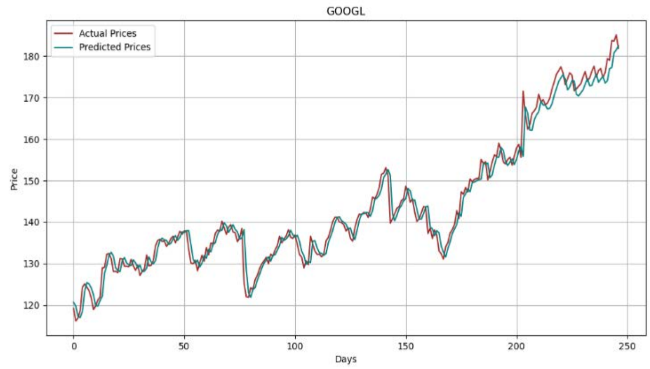
Figure 27. Visualization of GJO-optimized architectures

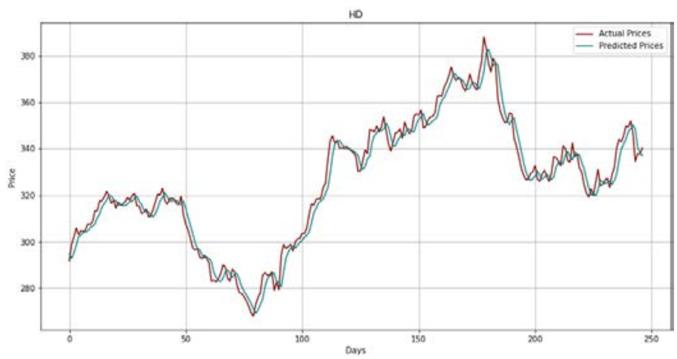


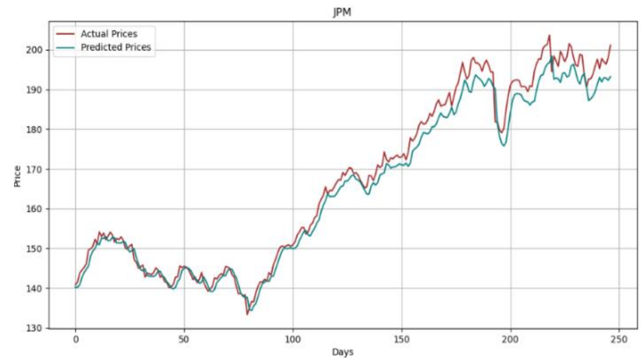

AVGO
Actual Prices
Predicted Prices
Price
Days

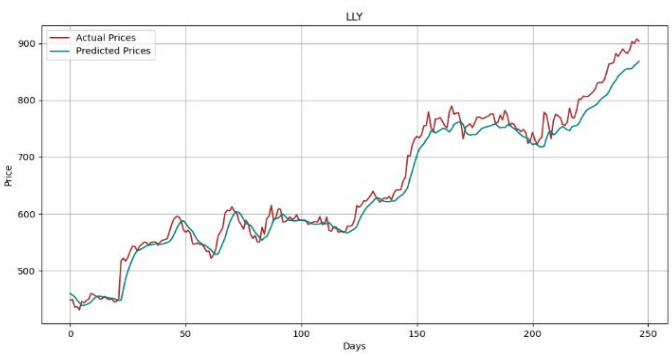

BRK-B
Actual Prices
Predicted Prices
Price
Days

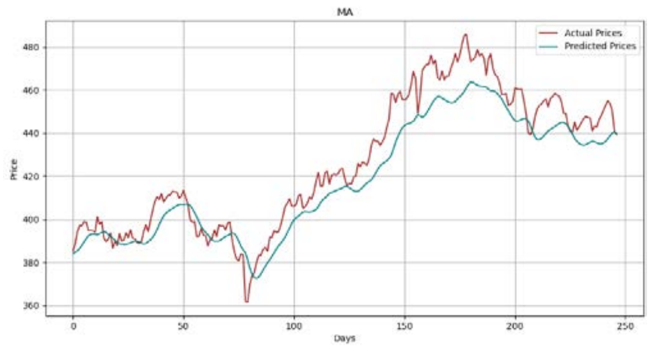

COST
Actual Prices
Predicted Prices
Price
Days

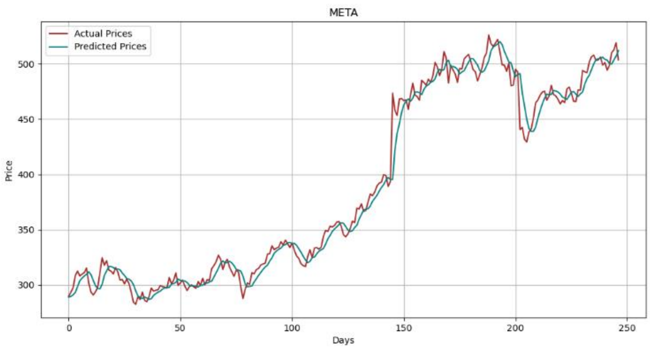

GOOGL
Actual Prices
Predicted Prices
Price
Days

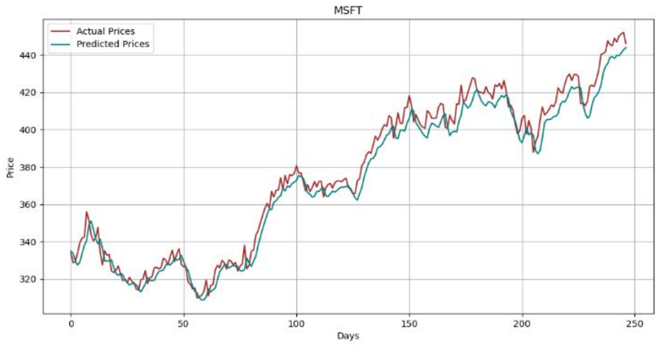

HD
Actual Prices
Predicted Prices
Price
Days

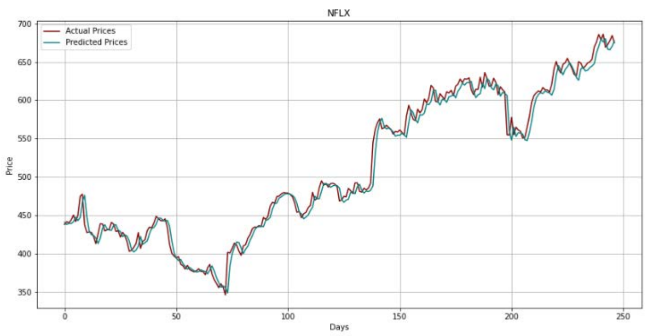

JPM
Actual Prices
Predicted Prices
Price
Days

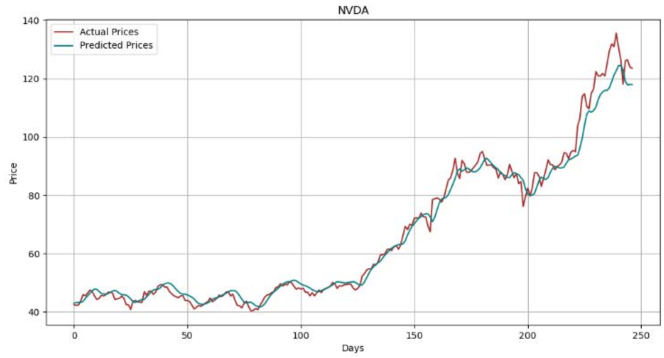

LLY
Actual Prices
Predicted Prices
Price
Days

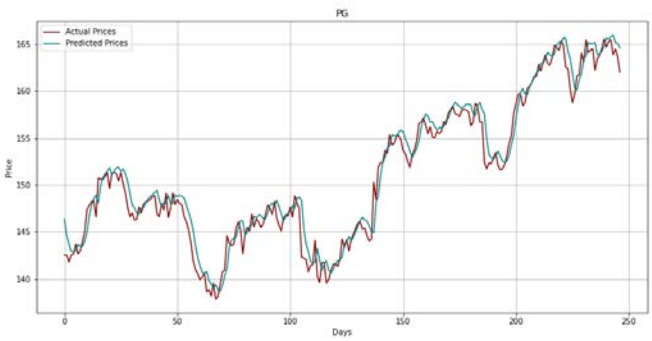

MA
Actual Prices
Predicted Prices
Price
Days

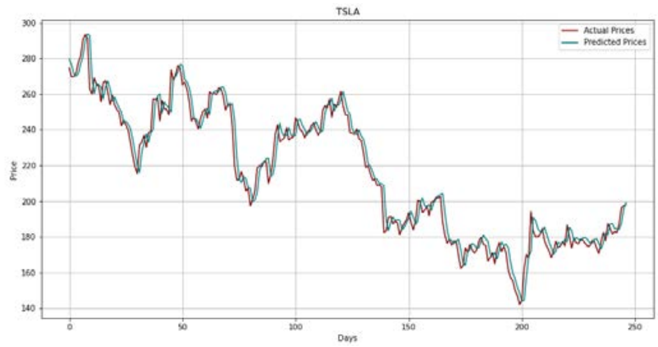

META
Actual Prices
Predicted Prices
Price
Days

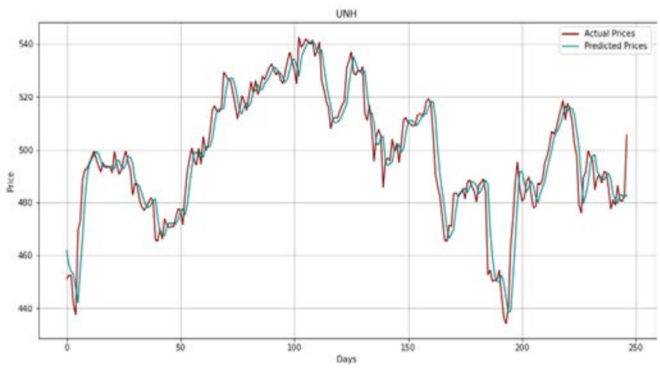

MSFT
Actual Prices
Predicted Prices
Price
Days

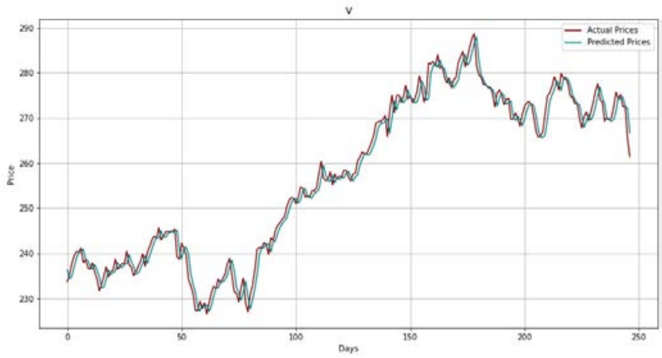

NFLX
Actual Prices
Predicted Prices
Price
Days


NVDA
Actual Prices
Predicted Prices
Price
Days


PG
Actual Prices
Predicted Prices
Price
Days


TSLA
Actual Prices
Predicted Prices
Price
Days


UNH
Actual Prices
Predicted Prices
Price
Days


V
Actual Prices
Predicted Prices
Price
Days

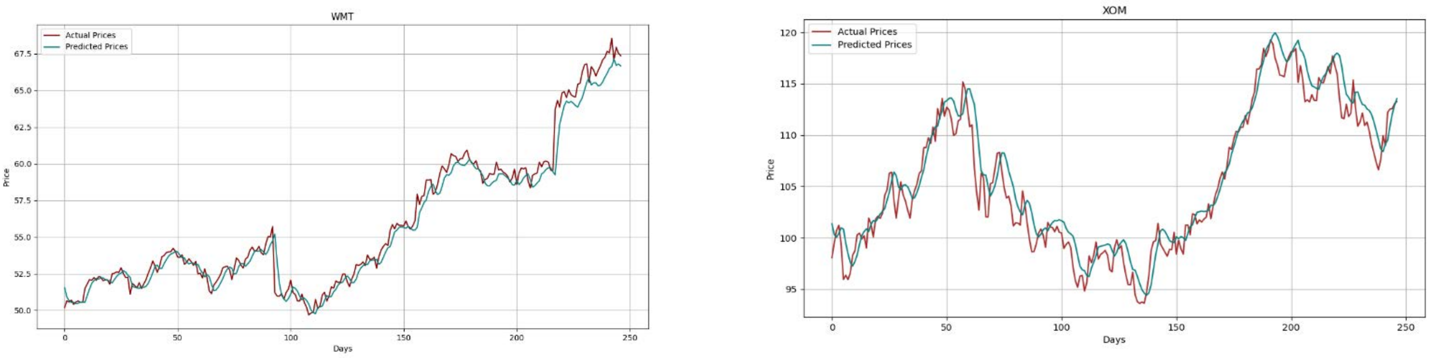
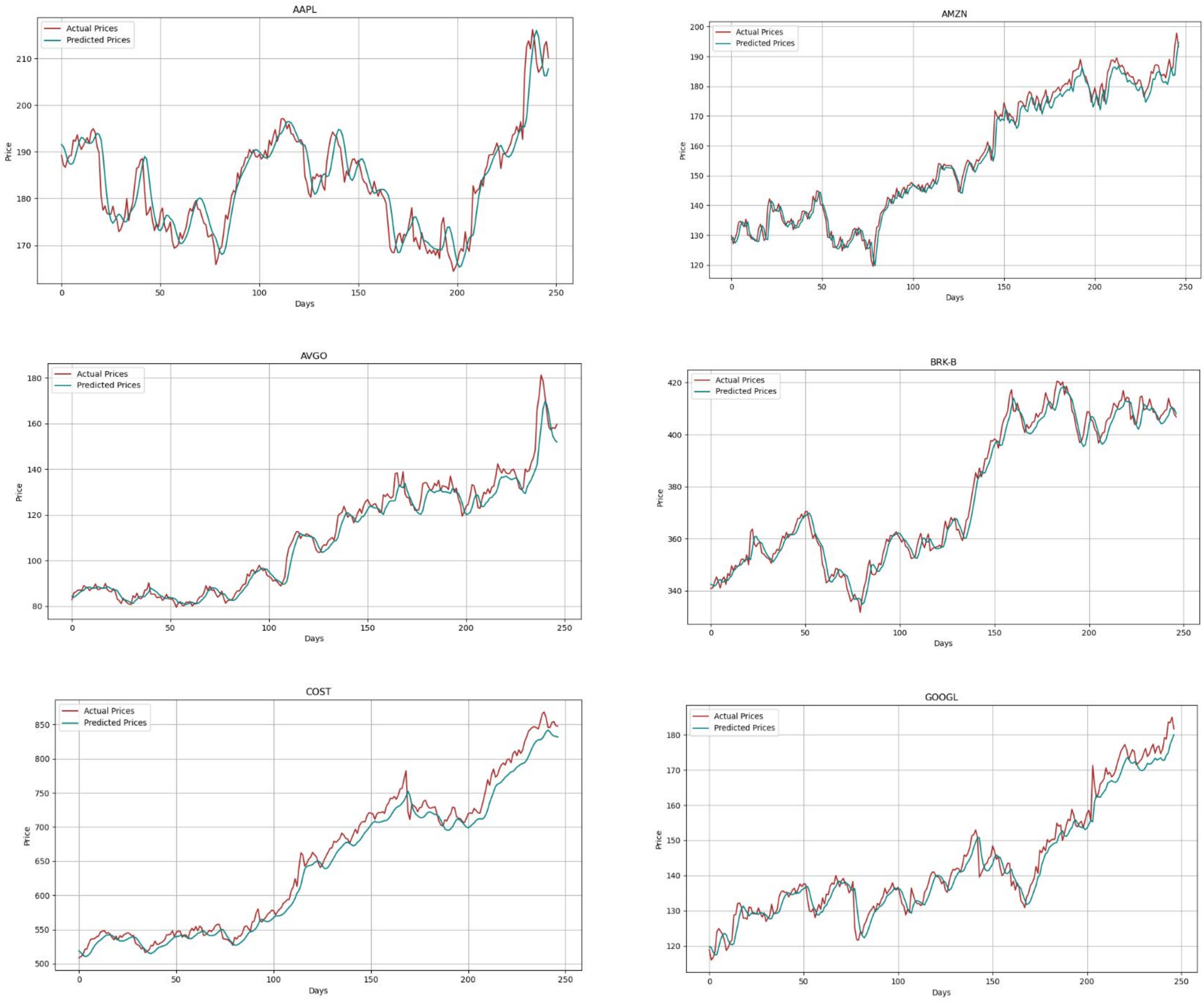
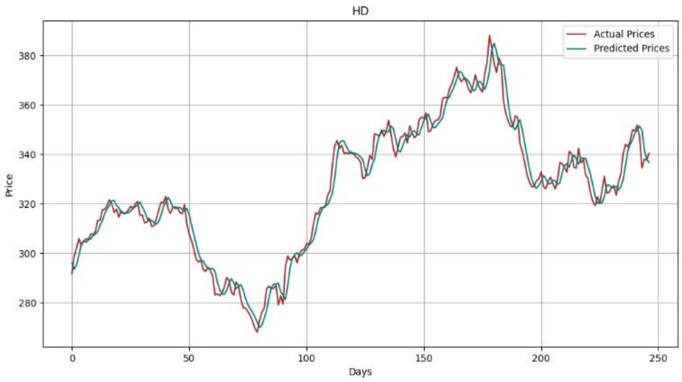
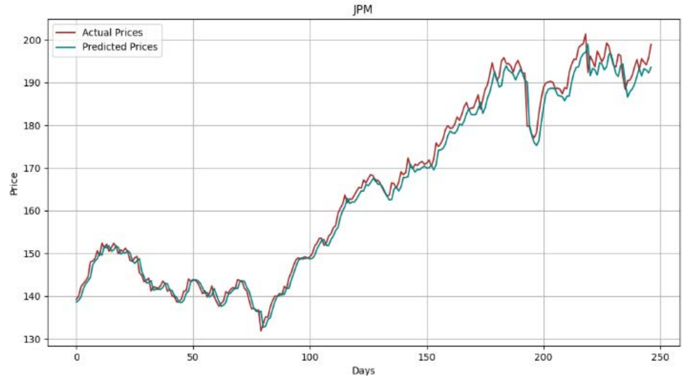
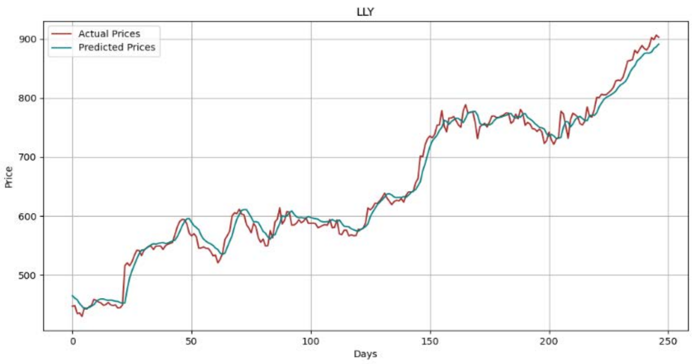


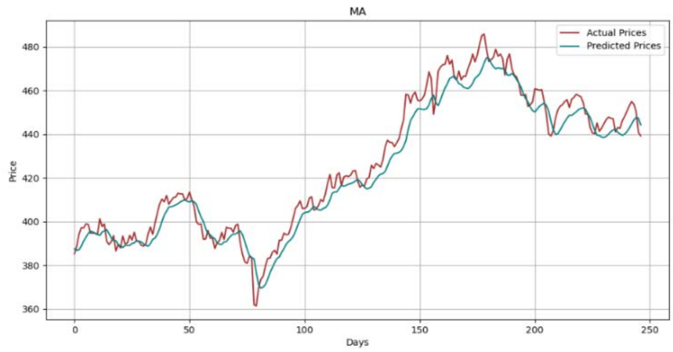
FIGURE 28. Visualization of PSO-optimized architectures

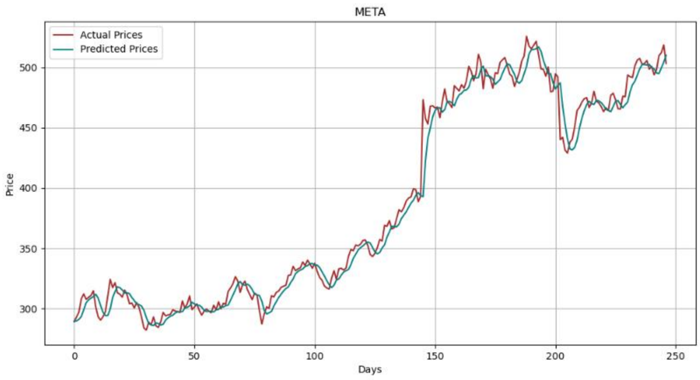


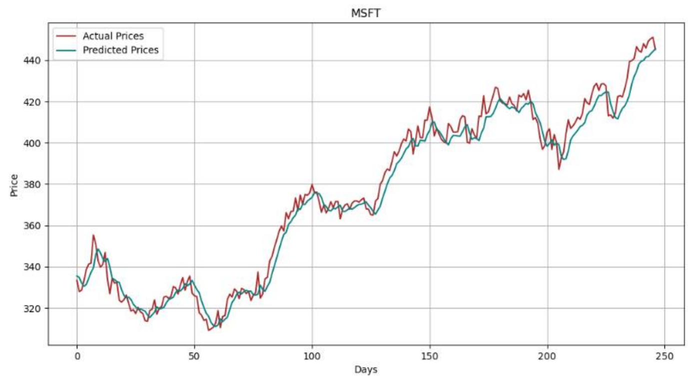

HD
Actual Prices
Predicted Prices
Price
Days
380
360
340
320
300
280
0
50
100
150
200
250

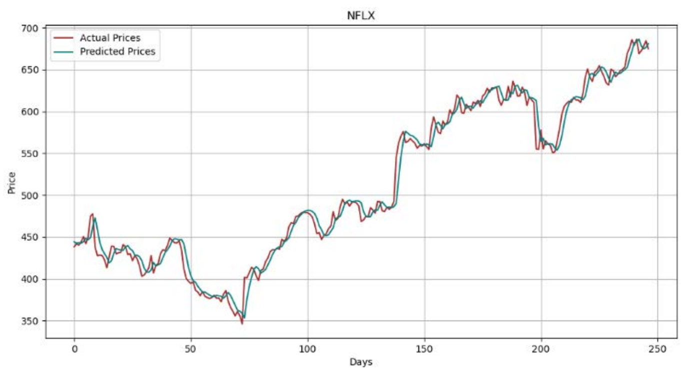

JPM
Actual Prices
Predicted Prices
Price
Days
200
190
180
170
160
150
140
130
0
50
100
150
200
250

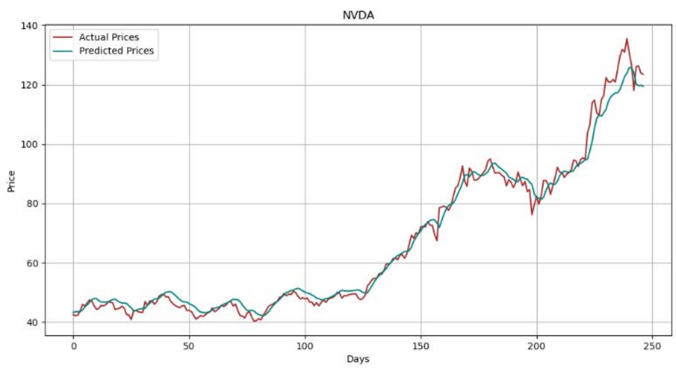

LLY
Actual Prices
Predicted Prices
Price
Days
900
800
700
600
500
0
50
100
150
200
250


MA
Actual Prices
Predicted Prices
Price
Days
480
460
440
420
400
380
360
0
50
100
150
200
250


META
Actual Prices
Predicted Prices
Price
Days
500
450
400
350
300
0
50
100
150
200
250


MSFT
Actual Prices
Predicted Prices
Price
Days
440
420
400
380
360
340
320
0
50
100
150
200
250


NFLX
Actual Prices
Predicted Prices
Price
Days
700
650
600
550
500
450
400
350
0
50
100
150
200
250


NVDA
Actual Prices
Predicted Prices
Price
Days
140
120
100
80
60
40
0
50
100
150
200
250

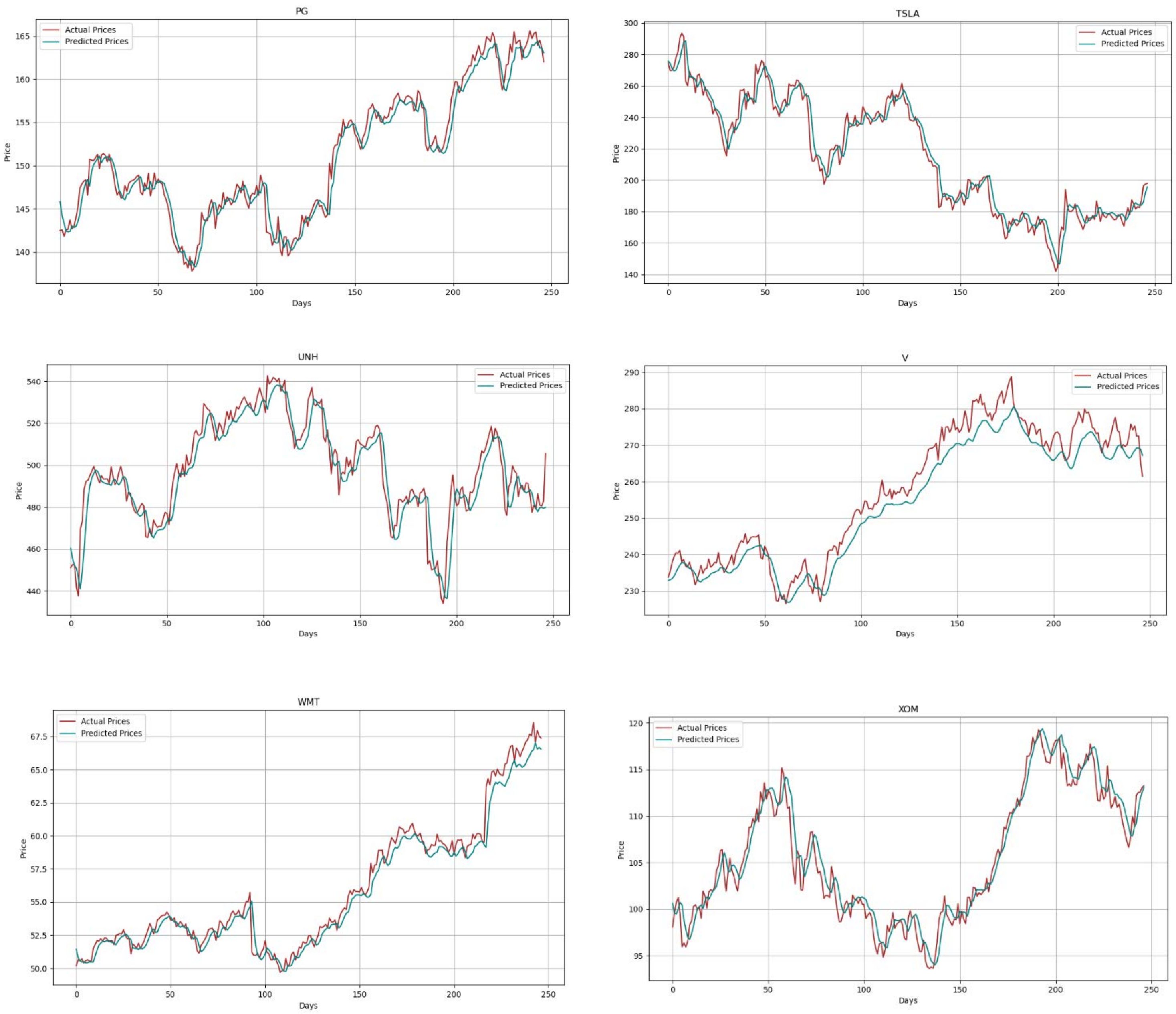


FIGURE 29. Visualization of GA-optimized architectures

4.4.3 Results

The findings in Table 46 shows that metaheuristic-optimized models improve forecasting quality. Regarding the RMSE score, the GJO-GRU | LSTM outperforms the other two metaheuristic-optimized models in the majority of instances. Out of the twenty cases, the proposed approach delivered better results in thirteen scenarios. At the same time, PSO-GRU | LSTM achieved the best performance in four scenarios, whilst GA-GRU | LSTM ranked first in three cases. More specifically, GA-GRU | LSTM showed higher accuracy in forecasting the prices of Eli Lilly & Co and JPMorgan Chase & Co., significantly surpassing the performance of the other two models. About Nvidia's stock price, GA-GRU | LSTM showed the best performance, while GJO-GRU | LSTM underperformed. In most instances, GA-GRU | LSTM exhibited poor performance, especially in the case of Visa, where it where it produced the weakest results. The PSO-GRU | LSTM model showed strong performance for the stocks of Alphabet, Netflix, United health Group, and Visa. Regarding the price prediction of Mastercard's stock, the PSO-GRU | LSTM exhibited the least effective performance relative to the other two models. In the thirteen instances where GJO-GRU | LSTM was the optimal model, it achieved better accuracy, especially for the stock prices of Costco Wholesale, Meta platforms, and Exxon Mobil. Regarding the Home Depot's stock price prediction, while GJO-GRU | LSTM achieved the best prediction accuracy, all metaheuristic-optimized approaches performed well.

In addition, in certain instances, GJO-GRU | LSTM and PSO-GRU | LSTM showed comparable performance, as observed with AAPL and TSLA, whereas GA-GRU | LSTM showed similar performance to GJO-GRU | LSTM in predicting Mastercard's stock price, with PSO-GRU | LSTM exhibiting lower accuracy. Despite the limited number of epochs and the small population size, the proposed approach achieved promising results. We recognize that augmenting these two elements could further improve forecasting accuracy. This is especially apparent in instances such as Costco Wholesale, where the highest prediction among benchmark models was attained by GRU-RMSProp, yielding an RMSE of 33.42, which is markedly inferior to metaheuristic-optimized models, particularly the suggested GJO-GRU |LSTM.

Concerning the benchmark models, we note that they do not adhere to a standard pattern.Someone may claim that we might obtain good performance utilizing the default architecture. However, we aim to prove a point. There is no optimal architecture for forecasting stock prices. Diverse scholars attempt to implement optimal structures for this purpose. Considering the performance of the benchmark models, it is evident tit is impossible to reach a clear decision. Among the benchmark models, we achieved the highest predictive performance for Eli Lilly & Co. and Microsoft using GRU-RAdam. In contrast, this specific design had poor results in predicting the stock prices of GOOGL and AVGO. Concurrently, with respect to Apple's and Amazon's stocks we note a change in the efficacy of the two most effective models. In one situation, the LSTM-RMSProp achieves the highest rank, succeeded by the GRU-RMSProp, whereas the second scenario inverts these results. It is likely accurate to assert that no stacked architecture can yield satisfactory results, as is often the case. Nevertheless, in instances involving hybrid stacked LSTM-GRU or GRU-LSTM architectures, notable performance was observed, shown by Berkshire Hathaway, where LSTM-GRU-RAdam produced the greatest prediction accuracy. Despite establishing a uniform number of units and batch size in our benchmark models, we can ascertain that no architecture is superior, as each time series possesses unique properties. Concerning optimizers, RAdam was utilized across various architectures, yielding optimal performance, including LSTM-RAdam in the case of predicting the price of JPMorgan Chase & Co., which outperformed both metaheuristic-optimized deep learning models; however, it also served as the optimizer in instances exhibiting worst performance in half of the cases.

A practitioner or a researcher may question the need for metaheuristic algorithms and their significance in this context, along with the relevance of this research and similar research. Analyzing the hyperparameter ranges in Table 44, we note that the candidate components yield nearly five million distinct neural network topologies. The analysis of the data, which includes twenty equities, results in more than one hundred million distinct deep learning models. Nonetheless, as previously stated, the results of the benchmark model reveal the lack of a dominant architecture. The use of optimization algorithms, especially metaheuristics, is crucial in creating improved deep learning models, hence eliminating the need for manual hyperparameter selection. Enhancing the architecture, including components such as optimizers, layers, number of units, and batch size, is vital for achieving a robust deep learning model. The proposed GJO-GRU | LSTM appears to be a promising method for this objective, as it tends to yield superior performance in most instances.

4.5 Summary of Experimental Results

In this section, we present the results of the applications proposed in this thesis. Across four practical studies, our goal was to develop approaches that combine AI and metaheuristic optimization algorithms to address financial risk and investment decision-making. The results show that the proposed methodologies achieve strong accuracy on the evaluated tasks. In the first application, we observed that the choice of activation function and optimizer in neural networks, as well as the use of resampling methods, can play a vital role in model performance. In the second application, optimizing the hyperparameters of the LightGBM model via PSS, using the SHS sampling method, improved credit card default prediction. For portfolio optimization, AGTO achieved better results in most cases than the other metaheuristic algorithms, which required more epochs to reach comparable performance. Finally, the proposed GJO–GRU|LSTM model provided better results in most cases for predicting stock closing prices. Table 45 summarizes, in tabular form, the best-performing approach for each application. Overall, the examined approaches delivered the strongest performance among the methods considered.

Table 45. Summary of findings per application

Application	Approach	Evaluation Metric	Best Performance
Churn Prediction	Deep Learning Models	AUC Score	APTx-Radam-SMOTE Tomek Links
Credit Card Default Prediction	Optimized LightGBM model	Train-Test AUC	PSS-SHS-LightGBM
Portfolio Optimization	Swarm-Intelligence metaheuristic algorithms	Sharpe Ratio	AGTO
Stock Price prediction	Metaheuristic optimized architectures	RMSE	GJO-GRU|LSTM

5. Conclusions

In this thesis, we examined the performance of artificial intelligence and metaheuristic algorithms in addressing challenges in the financial sector, including risk assessment and investment decision-making. The final conclusions show that these algorithms are able to provide better predictive performance. In addition, the results demonstrate that hybridizing these two approaches enhances predictive quality in both financial risk and investment decision-making. Regarding financial risk, we observed that the components of a deep learning model can play a vital role in predictive power. In this research, the use of the APTx activation function is proposed, which has not been widely evaluated in the literature. Beyond the extensive comparison with other well-known activation functions, we further examine the performance of deep learning models with respect to another important component: the network's optimization algorithm. Finally, we provide a thorough evaluation of common class-imbalance methods to address a typical challenge in customer-churn datasets—namely, imbalanced classes where the classifier tends to ignore the minority class. The results show that the APTx activation function provides more stable results across the majority of configurations, whereas the application of RAdam appears relatively limited. In addition, the experiments indicate that mixed sampling methods that combine oversampling and undersampling may yield better results. These findings contribute not only to the case of customer attrition—an important challenge for banks and financial institutions—but also to the broader areas of AI algorithms and imbalanced learning.

Another topic examined in this work is the application of metaheuristic-optimized machine learning models in credit risk, especially credit-card default prediction. This thesis proposes the novel PSS-SHS-LightGBM model for efficient credit-card default prediction. The model consists of LightGBM optimized via the PSS algorithm with the incorporation of SHS. This work is the first to propose using SHS within PSS. Moreover, instead of extensive data engineering and model stacking, we adopt an approach that uses LightGBM's internal features to handle class imbalance and feature dimensionality. Benchmarked against standard LightGBM and LightGBM optimized via the TPE algorithm, the results show that PSS-SHS-LightGBM provides more generalizable performance, allowing institutions and risk managers to avoid overly complex models while achieving high predictability. These findings offer valuable insights not only for the financial sector but also for the field of metaheuristic-optimized models.

The challenge of portfolio optimization is also studied, where several metaheuristic algorithms are evaluated for optimizing a portfolio comprising the stocks of the DJIA. In total, fifteen metaheuristic algorithms are assessed across multiple epochs and population sizes. The results show that the proposed AGTO delivers more stable outcomes in most cases compared with other metaheuristics. In addition, algorithms with limited application in this field—such as TSO and ARO—are evaluated within the portfolio-optimization context. This work not only sheds light on practical portfolio construction, benefiting practitioners and investors, but also provides important methodological insights. Finally, the hybrid GJO-GRU|LSTM model is proposed. The model optimizes not only the hyperparameters of a deep learning model but also its architecture, incorporating both GRU and LSTM networks and their key components. It is evaluated on the top twenty S&P 500 stocks by market capitalization, surpassing GA-GRU|LSTM, PSO-GRU|LSTM, and sixteen benchmark deep learning models in forecasting accuracy.

The bibliometric analysis presented in Chapter 2 shows increasing interest in the application of AI and metaheuristic optimization algorithms. In addition, the Chapter 2 literature review—focusing on recent publications—indicates that these algorithms can be considered an emerging approach in this field; thus, these findings directly address **RQ1**. Regarding **RQ2** and **RQ3**, the findings suggest that these types of algorithms provide more reliable results than classical models. They can handle and analyze massive datasets efficiently and produce accurate results. Next, we address **RQ4**. The applications examined in this work show that, despite the strengths of AI and ML models, a major limitation remains: proper hyperparameter optimization plays a vital role in their performance. Hence, it is important for practitioners and researchers to tackle this issue via optimization algorithms. This point speaks directly to **RQ5**, where we observed that applying metaheuristic algorithms to hyperparameter optimization in AI and ML models benefits overall performance. Our findings for **RQ2–RQ5** are also consistent with the literature review presented in Chapter 2.

Although this thesis addresses the research questions, several directions for future work remain. In the financial risk assessment studies, we relied on two widely used benchmark datasets to allow fair comparison with prior work and to maintain methodological consistency. Testing the proposed models on real-world datasets would help confirm their robustness. For the task of portfolio optimization, future work could explore the performance of a broader range of metaheuristic algorithms on more challenging optimization problem. Regarding stock-price prediction, applying the proposed model to a larger set of stocks and benchmarking against additional metaheuristic-tuned models would strengthen the evidence for robustness. In addition to the specific applications examined, future work could study the performance of these algorithms for fraud detection, bankruptcy prediction, and cryptocurrency markets.

In summary, the applications of AI and metaheuristic optimization algorithms can address various challenges in the financial sector. Despite the superiority of AI and machine-learning models, several issues must be addressed; key among these is optimal hyperparameter selection. As shown in this thesis, metaheuristic algorithms are a viable tool in this regard, as they can help AI models improve predictability without manually testing numerous configurations. As the bibliometric analysis shows, these algorithms have gained increasing popularity across a variety of financial tasks. Beyond pure optimization tasks such as portfolio optimization, they are increasingly integrated with AI models to enhance forecasting quality. Although AI models are strong predictors, they are highly sensitive to hyperparameter choices, and applications of metaheuristic algorithms in finance remain relatively limited. The results of this work demonstrate that AI and metaheuristic optimization algorithms can be valuable tools for the financial sector. These algorithms can enhance financial risk management and investment decision-making by producing more accurate predictions, enabling researchers, investors, and financial institutions to make better-informed decisions.

Appendix A

Table A1. Final Query of the analysis

("metaheuristic*" OR "Evolutionary Programming" OR "Evolution Strategies" OR "Memetic Algorithm" OR "Genetic Algorithm" OR "Differential Evolution" OR "Success-History Adaptation Differential Evolution" OR "Flower Pollination Algorithm" OR "Coral Reefs Optimization" OR "Particle Swarm Optimization" OR "Bacterial Foraging Optimization Algorithm" OR "Bees Algorithm" OR "Cat Swarm Optimization" OR "Artificial Bee Colony" OR "Ant Colony Optimization" OR "Cuckoo Search" OR "Firefly Algorithm" OR "Fireworks Algorithm" OR "Bat Algorithm" OR "Fruit Fly Optimization Algorithm" OR "Social Spider Optimization" OR "Grey Wolf Optimizer" OR "Social Spider Algorithm" OR "Ant Lion Optimizer" OR "Moth Flame Optimization" OR "Elephant Herding Optimization" OR "Jaya Algorithm" OR "Whale Optimization Algorithm" OR "Dragonfly Optimization" OR "Bird Swarm Algorithm" OR "Spotted Hyena Optimizer" OR "Salp Swarm Optimization" OR "Swarm Robotics Search And Rescue" OR "Grasshopper Optimisation Algorithm" OR "Coyote Optimization Algorithm" OR "Moth Search Algorithm" OR "Sea Lion Optimization" OR "Naked Mole-Rat Algorithm" OR "Pathfinder Algorithm" OR "Sailfish Optimizer" OR "Harris Hawks Optimization" OR "Manta Ray Foraging Optimization" OR "Bald Eagle Search" OR "Sparrow Search Algorithm" OR "Hunger Games Search" OR "Aquila Optimizer" OR "Hybrid Grey Wolf-Whale Optimization Algorithm" OR "Marine Predators Algorithm" OR "Honey Badger Algorithm" OR "Sand Cat Swarm Optimization" OR "Tuna Swarm Optimization" OR "African Vultures Optimization Algorithm" OR "Artificial Gorilla Troops Optimization" OR "Artificial Rabbits Optimization" OR "Egret Swarm Optimization Algorithm" OR "Fox Optimizer" OR "Golden Jackal Optimization" OR "Giant Trevally Optimization" OR "Mountain Gazelle Optimizer" OR "Sea-Horse Optimization" OR "Simulated Annealing" OR "Wind Driven Optimization" OR "Multi-Verse Optimizer" OR "Tug of War Optimization" OR "Electromagnetic Field Optimization" OR "Nuclear Reaction Optimization" OR "Henry Gas Solubility Optimization" OR "Atom Search Optimization" OR "Equilibrium Optimizer" OR "Archimedes Optimization Algorithm" OR "Chernobyl Disaster Optimization" OR "Energy Valley Optimization" OR "Fick's Law Algorithm" OR "Physical Phenomenon of RIME-ice" OR "Culture Algorithm" OR "Imperialist Competitive Algorithm" OR "Teaching Learning-Based Optimization" OR "Brain Storm Optimization" OR "Queuing Search Algorithm" OR "Search And Rescue Optimization" OR "Life Choice-Based Optimization" OR "Social Ski-Driver Optimization" OR "Gaining Sharing Knowledge-Based Algorithm" OR "Coronavirus Herd Immunity Optimization" OR "Forensic-Based Investigation Optimization" OR "Battle Royale Optimization" OR "Student Psychology Based Optimization" OR "Heap-Based Optimization" OR "Human Conception Optimization" OR "Dwarf Mongoose Optimization Algorithm" OR "War Strategy Optimization" OR "Invasive Weed

Optimization" OR "Biogeography-Based Optimization" OR "Virus Colony Search" OR "Satin Bowerbird Optimizer" OR "Earthworm Optimisation Algorithm" OR "Wildebeest Herd Optimization" OR "Slime Mould Algorithm" OR "Barnacles Mating Optimizer" OR "Tunicate Swarm Algorithm" OR "Symbiotic Organisms Search" OR "Seagull Optimization Algorithm" OR "Brown-Bear Optimization Algorithm" OR "Tree Physiology Optimization" OR "Germinal Center Optimization" OR "Water Cycle Algorithm" OR "Artificial Ecosystem-Based Optimization" OR "Hill Climbing" OR "Cross-Entropy Method" OR "Tabu Search" OR "Sine Cosine Algorithm" OR "Gradient-Based Optimizer" OR "Arithmetic Optimization Algorithm" OR "Chaos Game Optimization" OR "Pareto-like Sequential Sampling" OR "Weighted Mean of Vectors" OR "Runge Kutta Optimizer" OR "Circle Search Algorithm" OR "Success History Intelligent Optimization" OR "Harmony Search")

AND

("artificial intelligence" OR "machine learning" OR "neural networks" OR "supervised learning" OR "unsupervised learning" OR "ensemble learning" OR "outlier detection" OR "multilayer perceptrons" OR "MLP" OR "deep learning" OR "deep neural networks" OR "clustering" OR "support vector machines" OR "SVM" OR "support vector regression" OR "decision trees" OR "artificial neural networks" OR "gradient boosting" OR "XGBoost" OR "LightGBM" OR "CatBoost" OR "AdaBoost" OR "supervised learning" OR "unsupervised learning" OR "random forests" OR "natural language processing" OR "NLP" OR "large language models" OR "LLM" OR "sentiment analysis" OR "reinforcement learning" OR "text processing" OR "Naive Bayes" OR "fuzzy systems" OR "stochastic systems" OR "AI" OR "Long Short-Term Memory" OR "LSTM" OR "Gated Recurrent Unit" OR "GRU" OR "Boltzmann Machines" OR "Restricted Boltzmann Machines" OR "RBM" OR "Deep Belief Networks" OR "DBN" OR "Autoencoders" OR "Self-Organizing Maps" OR "SOM" OR "Generative Adversarial Networks" OR "GAN" OR "Convolutional Neural Networks" OR "CNN" OR "Graph Neural Networks" OR "GNN" OR "Transformers" OR "Bidirectional Encoder Representations from Transformers" OR "BERT" OR "Principal Component Analysis" OR "PCA" OR "Linear Discriminant Analysis" OR "LDA" OR "t-Distributed Stochastic Neighbor Embedding" OR "t-SNE" OR "Uniform Manifold Approximation and Projection" OR "UMAP" OR "Hierarchical Clustering" OR "K-Means Clustering" OR "K-Means" OR "Density-Based Spatial Clustering of Applications with Noise" OR "DBSCAN" OR "Isolation Forest" OR "iForest")

AND

("finance" OR "financial management" OR "financial forecast*" OR "asset allocation" OR "portfolio optimization" OR "capital allocation" OR "financial technology" OR "fintech" OR "stock*" OR "stock market prediction" OR "stock price prediction" OR "stock price forecast*" OR "investment analysis" OR "algorithmic trading" OR "high-frequency trading" OR "asset pricing" OR "bond pricing" OR "hedging strategies" OR "futures trading" OR "derivatives pricing" OR "credit risk*" OR "credit risk assessment" OR "loan prediction" OR "market trend prediction" OR "banks" OR "bankruptcy" OR "financial risk*" OR "systemic risk*" OR "market risk*" OR "credit default risk" OR "liquidity risk" OR "operational risk*" OR "currency risk*" OR "European option*" OR "American option*" OR "Asian option*" OR "call option" OR "put option" OR "option strateg*" OR "option pricing" OR "bank lending" OR "debt management" OR "liquidity management" OR "exchange rate forecast*" OR "volatility model*" OR "commodity price forecast*" OR "interest rate forecast*" OR "macroeconomic forecast*" OR "interest rate" OR "corporate valuation" OR "corporate finance" OR "mergers and acquisitions" OR "mergers & acquisitions" OR "Environmental, Social, and Governance" OR "ESG" OR “ESG risk*” OR “ESG stress test” OR "sustainable finance" OR "climate finance" OR "green investment analysis" OR "quantitative finance" OR "financial econometrics" OR “Bitcoin” OR "cryptocurrenc*" OR "cryptocurrency price prediction" OR "cryptocurrency price forecast*" OR "token valuation" OR "NFT valuation" OR "credit scoring" OR "fraud detection" OR "financial contagion" OR "stress test*" OR "stock dividend*" OR "cash dividend*" OR "dividend yield" OR "dividend policy" OR "tax planning" OR "microfinance" OR "asset-liability management" OR "fund performance evaluation" OR "financial model*" OR "financial market*" OR "capital market*" OR "investment portfolio" OR "financial distress" OR "default probability" OR "interest rate risk*" OR "equity risk*" OR "commodity risk*" OR "volatility risk*")

Table A2. Publications of Physica A: Statistical Mechanics and its Applications

Authors	Year	DOI
Yelibi L., Gebbie T.	2020	10.1016/j.physa.2019.124049
Jiang M., Liu J., Zhang L., Liu C.	2020	10.1016/j.physa.2019.122272
Han T., Peng Q., Zhu Z., Shen Y., Huang H., Abid N.N.	2020	10.1016/j.physa.2020.124161
Shen F., Zhao X., Li Z., Li K., Meng Z.	2019	10.1016/j.physa.2019.121073
Fu S., Li Y., Sun S., Li H.	2019	10.1016/j.physa.2019.01.026
Das S.R., Kuhoo, Mishra D., Rout M.	2019	10.1016/j.physa.2018.09.021
Dash R.	2017	10.1016/j.physa.2017.05.044
Gong X., Zhuang X.	2017	10.1016/j.physa.2016.09.005
Lahmiri S.	2016	10.1016/j.physa.2015.09.061
Gong X., Zhuang X.	2016	10.1016/j.physa.2016.02.064
Yang G., Chen Y., Huang J.P.	2016	10.1016/j.physa.2015.09.071
Xiao W., Zhang W., Zhang X., Chen X.	2014	10.1016/j.physa.2013.09.033
Manahov V., Hudson R.	2013	10.1016/j.physa.2013.05.029
Fiebig H.R., Musgrove D.P.	2015	10.1016/j.physa.2015.01.055
Caetano M.A.L., Yoneyama T.	2011	10.1016/j.physa.2011.07.039

Algorithm 1. Pareto-like sequential sampling (PSS)

Algorithm 1*:* Pareto-like sequential sampling (PSS)

Result: Minimize the objective function $f(\boldsymbol{x})$
read parameter-settings.
pre-allocate and initialize memory containers.
begin
  $initialize\ a\ population\ sample\ \in\ \boldsymbol{\Gamma}$;
  $evaluate\ the\ initial\ population$;
  $\boldsymbol{for}\ i = 1\ to\ \gamma\ do$
    $//\ best\ solution\ in\ iteration\ i\ is\ x^{i}_{best}$
    find the current best x^{i}_{best}
    generate $[u^{i}]_{\beta\times n}\sim U(0,1)$;
    $\boldsymbol{for}\ k = 1\ \boldsymbol{to}\ \beta\ \boldsymbol{do}$
      if $f(x^{i}_{best}) < f(x^{i-1}_{best})$ **then**
        $update\ \boldsymbol{\eta}^{i}\ and\ {}^{i}\boldsymbol{\Gamma}'$;
      end
      foreach $\boldsymbol{x_j} \in \boldsymbol{x}\ \boldsymbol{do}$
        if $rand\sim U(0,1);\leq\alpha\ then$
          //sample from the prominent domain Ω'
          choose random ${}_{k}x^{i}_{j} \in \boldsymbol{\Gamma^{i}_{j}}$;
        else
          //sample from the overall domain Ω
          choose random ${}_{k}x^{i}_{j} \in \boldsymbol{\Gamma_{j}}$;
        end
      end
      $[population]_{\boldsymbol{\beta\times n}} <-{}_{k}x$;
      $evaluate\ f({}_{k}x)$;
    end
  end
end
return x_{best}

Algorithm 2. Artificial Gorillas Troops Optimizer (AGTO)

Algorithm 2. Pseudocode of AGTO

% GTO setting

Inputs: The population size N and maximum number of iterations T and parameters β and p

Outputs: The location of Gorilla and its fitness value

% initialization

Initialize the random population Xi $(i = 1, 2, \dots, N)$

Calculate the fitness values of Gorilla

% Main Loop

while (stopping condition is not met) do

Update the C using Equation (2)

Update the L using Equation (4)

% Exploration phase

for (each Gorilla (Xi)) do

Update the location Gorilla using Equation (1)

end for

% Create group

Calculate the fitness values of Gorilla

if GX is better than X, replace them

Set Xsilverback as the location of silverback (best location)

% Exploitation phase

for (each Gorilla (Xi)) do

if ($|\mathbf{C}| \geq \mathbf{1}$) **then**

Update the location Gorilla using Equation (7)

Else

Update the location Gorilla using Equation (10)

End if

end for

% Create group

Calculate the fitness values of Gorilla

if New Solutions are better than previous solutions, replace them

Set Xsilverback as the location of silverback (best location)

end while

Return XBestGorilla, bestFitness

Algorithm 3.Golden Jackal Optimizer (GJO)

Algorithm 3 Pseudocode of GJO

$\boldsymbol{Inputs}$: $The\ population\ size\ N\ and\ maximum\ number\ of\ iterations\ T$

$\boldsymbol{Outputs}$: $\boldsymbol{T}he\ location\ of\ prey\ and\ its\ fitness\ value$
$Initialize\ the\ random\ prey\ population\ Y_i (i = 1,2, \dots, N)$
$\boldsymbol{While}\ (t < T)$
$\quad Calculate\ the\ fitness\ values\ of\ preys$
$\quad Y_1 = best\ prey\ (Male\ Jackal\ Position)$
$\quad Y_2 = second\ best\ prey\ (Female\ Jackal\ Position)$

$\boldsymbol{for}\ (each\ prey)$
$\quad Update\ the\ evading\ energy\ \text{“}E\text{”}\ using\ Eq.(16), Eq.(17) and\ Eq.(18)$
$\quad Update\ \text{“}rl\text{”}\ using\ Eq.(9) and\ Eq.(10)$
$\quad if (|E| \geq 1)(Exploration\ phase)$
$\quad Update\ the\ prey\ position\ using\ Eq.(14), Eq.(15) and\ Eq.(21)$
$\quad if\ (|E| < 1)(Exploitation\ phase)$
$\quad Update\ the\ prey\ position\ using\ Eq.(22), Eq.(23)\ and\ Eq.(21)$
$\boldsymbol{end\ for}$
$\quad t = t + 1$

$\boldsymbol{end\ while}$
$return\ Y_1$

References

Abadi, M., A. Agarwal, P. Barham, *et al.* 2016. TensorFlow: LargeScale Machine Learning on Heterogeneous Distributed Systems. arXiv Preprint arXiv:1603.04467.

Abdou, H. A. (2009). Genetic programming for credit scoring: The case of Egyptian public sector banks. *Expert Systems with Applications, 36*(9), 11402–11417. https://doi.org/10.1016/j.eswa.2009.01.076

Abdollahzadeh, B., Soleimanian Gharehchopogh, F., & Mirjalili, S. (2021). Artificial gorilla troops optimizer: a new nature-inspired metaheuristic algorithm for global optimization problems. *International Journal of Intelligent Systems, 36*(10), 5887-5958. https://doi.org/10.1002/int.22535

Abdollahzadeh, B., Khodadadi, N., Barshandeh, S., Trojovský, P., Gharehchopogh, F. S., El-kenawy, E. S. M., ... & Mirjalili, S. (2024). Puma optimizer (PO): A novel metaheuristic optimization algorithm and its application in machine learning. *Cluster Computing*, 27**,** 5235–5283. https://doi.org/10.1007/s10586-023-04221-5

Acharya, R. N., Kagan, A., & Rao Lingam, S. (2008). Online banking applications and community bank performance. *International Journal of Bank Marketing*, *26*(6), 418-439. https://doi.org/10.1108/02652320810902442

Agrawal, P., Abutarboush, H. F., Ganesh, T., & Mohamed, A. W. (2021). Metaheuristic algorithms on feature selection: A survey of one decade of research (2009-2019). *Ieee Access*, *9*, 26766-26791.https://doi.org/10.1109/ACCESS.2021.3056407

Agushaka, J. O., Ezugwu, A. E., Abualigah, L., Alharbi, S. K., & Khalifa, H. A. E. W. (2023). Efficient initialization methods for population-based metaheuristic algorithms: A comparative study. *Archives of Computational Methods in Engineering*, 30(3), 1727-1787. https://doi.org/10.1007/s11831-022-09850-4

Ahmed, S., Alshater, M. M., El Ammari, A., & Hammami, H. (2022). Artificial intelligence and machine learning in finance: A bibliometric review. *Research in International Business and Finance*, *61*, 101646.https://doi.org/10.1016/j.ribaf.2022.101646

Akay, B., Karaboga, D., & Akay, R. (2022). A comprehensive survey on optimizing deep learning models by metaheuristics. *Artificial Intelligence Review*, *55*(2), 829-894.https://doi.org/10.1007/s10462-021-09992-0

Akande, K. O., Owolabi, T. O., & Olatunji, S. O. (2015). Investigating the effect of correlation-based feature selection on the performance of neural network in reservoir characterization. *Journal of Natural Gas Science and Engineering*, 27, 98-108. https://doi.org/10.1016/j.jngse.2015.08.042

Alam, T. M., Shaukat, K., Hameed, I. A., Luo, S., Sarwar, M. U., Shabbir, S., & Khushi, M. (2020). An investigation of credit card default prediction in the imbalanced datasets. *IEEE Access, 8*, 201173–201198. https://doi.org/10.1109/ACCESS.2020.3033784

Al-Jarrah, O. Y., Yoo, P. D., Muhaidat, S., Karagiannidis, G. K., & Taha, K. (2015). Efficient machine learning for big data: A review. Big Data Research, 2(3), 87-93. https://doi.org/10.1016/j.bdr.2015.04.001

Arnone, S., Loraschi, A., & Tettamanzi, A. (1993). A genetic approach to portfolio selection. Neural Network World, 3(6), 597-604.

Anantrasirichai, N., & Bull, D. (2022). Artificial intelligence in the creative industries: a review. *Artificial intelligence review*, *55*(1), 589-656. https://doi.org/10.1007/s10462-021-10039-7

Aria, M., & Cuccurullo, C. (2017). bibliometrix: An R-tool for comprehensive science mapping analysis. *Journal of informetrics*, *11*(4), 959-975.https://doi.org/10.1016/j.joi.2017.08.007

Aytac, S., Tran, C. Y., & Frye, N. (2025). Lotka's Law of Scientific Productivity Across the Research Disciplines: A Review. *Science & Technology Libraries*, 1-13.https://doi.org/10.1080/0194262X.2025.2503941

Bacanin, N., & Tuba, M. (2014). Firefly Algorithm for Cardinality Constrained Mean-Variance Portfolio Optimization Problem with Entropy Diversity Constraint. *The Scientific World Journal*, 2014(1), 721521. https://doi.org/10.1155/2014/721521

Bacanin, N., Stoean, C., Zivkovic, M., Rakic, M., Strulak-Wójcikiewicz, R., & Stoean, R. (2023). On the benefits of using metaheuristics in the hyperparameter tuning of deep learning models for energy load forecasting. *Energies*, *16*(3), 1434.

Bahoo, S., Cucculelli, M., & Qamar, D. (2023). Artificial intelligence and corporate innovation: A review and research agenda. *Technological Forecasting and Social Change*, *188*, 122264.https://doi.org/10.1016/j.techfore.2022.122264

Barrera-García, J., Cisternas-Caneo, F., Crawford, B., Gómez Sánchez, M., & Soto, R. (2023). Feature selection problem and metaheuristics: A systematic literature review about its formulation, evaluation and applications. *Biomimetics*, *9*(1), 9.https://doi.org/10.3390/biomimetics9010009

Batista, G. E., Bazzan, A. L., & Monard, M. C. (2003). Balancing Training Data for Automated Annotation of Keywords: a Case Study. *WOB, 3, 10-8.*

Batista, G. E., Prati, R. C., & Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD explorations newsletter*, *6*(1), 20-29.

Bellman R (1957) Dynamic programming. Princeton University Press, Princeton, NJ

Bhat, C. R. (2003). Simulation estimation of mixed discrete choice models using randomized and scrambled Halton sequences. *Transportation Research Part B: Methodological, 37*(9), 837–855.

Bischl, B., Binder, M., Lang, M., Pielok, T., Richter, J., Coors, S., ... & Lindauer, M. (2023). Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *13*(2), e1484. https://doi.org/10.1002/widm.1484

Bonsón, E., Lavorato, D., Lamboglia, R., & Mancini, D. (2021). Artificial intelligence activities and ethical approaches in leading listed companies in the European Union. *International Journal of Accounting Information Systems*, *43*, 100535.https://doi.org/10.1016/j.accinf.2021.100535

Borgi, T., Zoghlami, N., & Abed, M. (2017, January). Big data for transport and logistics: A review. In *2017 International Conference on Advanced Systems and Electric Technologies (IC_ASET)* (pp. 44-49). IEEE.

Burez, J., & Van den Poel, D. (2009). Handling class imbalance in customer churn prediction. *Expert Systems with Applications*, *36*(3), 4626-4636.

Bulani, V., Bezbradica, M., & Crane, M. (2025). Improving Portfolio Management Using Clustering and Particle Swarm Optimisation. *Mathematics*, *13*(10), 1623.

Bruce, E., Keelson, S., Amoah, J., & Bankuoru Egala, S. (2023). Social media integration: An opportunity for SMEs sustainability. *Cogent Business & Management*, *10*(1), 2173859.https://doi.org/10.1080/23311975.2023.2173859

Cai, J., Luo, J., Wang, S., & Yang, S. (2018). Feature selection in machine learning: A new perspective. *Neurocomputing*, *300*, 70-79.https://doi.org/10.1016/j.neucom.2017.11.077

Cao, G., & Weerawardena, J. (2023). Strategic use of social media in marketing and financial performance: The B2B SME context. *Industrial Marketing Management*, *111*, 41-54.https://doi.org/10.1016/j.indmarman.2023.03.007

Cerchiello, P., & Giudici, P. (2016). Big data analysis for financial risk management. *Journal of Big Data*, *3*(1), 18. https://doi.org/10.1186/s40537-016-0053-4

Cerqueira, V., Torgo, L., & Soares, C. (2019). Machine learning vs statistical methods for time series forecasting: Size matters. *arXiv preprint arXiv:1909.13316*.

Chang, T. J., Meade, N., Beasley, J. E., & Sharaiha, Y. M. (2000). Heuristics for cardinality constrained portfolio optimisation. *Computers & Operations Research*, *27*(13), 1271-1302.

Chandrashekar, G., & Sahin, F. (2014). A survey on feature selection methods. *Computers & electrical engineering*, *40*(1), 16-28.https://doi.org/10.1016/j.compeleceng.2013.11.024

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research, 16.* https://doi.org/10.1613/jair.953

Chen, W. (2015). Artificial bee colony algorithm for constrained possibilistic portfolio optimization problem. *Physica A: Statistical Mechanics and its Applications*, 429, 125-139.https://doi.org/10.1016/j.physa.2015.02.060

Chen, W., Zhang, H., Mehlawat, M. K., & Jia, L. (2021). Mean–variance portfolio optimization using machine learning-based stock price prediction. *Applied soft computing*, *100*, 106943.

Chen, W., Rabhi, F., Liao, W., & Al-Qudah, I. (2023). Leveraging state-of-the-art topic modeling for news impact analysis on financial markets: a comparative study. *Electronics*, *12*(12), 2605. https://doi.org/10.3390/electronics12122605

Chou, J. S., Nguyen, N. M., & Chang, C. P. (2022). Intelligent candlestick forecast system for financial time-series analysis using metaheuristics-optimized multi-output machine learning. *Applied Soft Computing*, *130*, 109642. https://doi.org/10.1016/j.asoc.2022.109642

Chopra, N., & Ansari, M. M. (2022). Golden jackal optimization: A novel nature-inspired optimizer for engineering applications. *Expert Systems with Applications*, 198, 116924. https://doi.org/10.1016/j.eswa.2022.116924

Clevert, D. A., Unterthiner, T., & Hochreiter, S. (2015). Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*.

Cui, K., Zhan, Z., & Pan, C. (2020). Applying radam method to improve treatment of convolutional neural network on banknote identification. *In 2020 International Conference on Computer Engineering and Application (ICCEA)*, 468-476. IEEE.

Cunha, F. A. F. D. S., Meira, E., & Orsato, R. J. (2021). Sustainable finance and investment: Review and research agenda. *Business Strategy and the Environment*, *30*(8), 3821-3838.https://doi.org/10.1002/bse.2842

Dalli, A. (2022). Impact of hyperparameters on deep learning model for customer churn prediction in telecommunication sector. *Mathematical Problems in Engineering*, *2022*, 1-11. https://doi.org/10.1155/2022/4720539

Daoud, A., & Ben-Hur, A. (2025). The role of chromatin state in intron retention: A case study in leveraging large scale deep learning models. *PLoS Computational Biology, 21*(1), e1012755. https://doi.org/10.1371/journal.pcbi.1012755

Dhal, K. G., Das, A., Sahoo, S., Das, R., & Das, S. (2021). Measuring the curse of population size over swarm intelligence based algorithms. *Evolving Systems*, *12*(3), 779-826.

De Caigny, A., Coussement, K., & De Bock, K. W. (2018). A new hybrid classification algorithm for customer churn prediction based on logistic regression and decision trees. *European Journal of Operational Research*, 269(2), 760-772. https://doi.org/10.1016/j.ejor.2018.02.009

Derrac, J., García, S., Molina, D., & Herrera, F. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, *1*(1), 3-18.

Doering, J., Kizys, R., Juan, A. A., Fito, A., & Polat, O. (2019). Metaheuristics for rich portfolio optimisation and risk management: Current state and future trends. *Operations Research Perspectives*, *6*, 100121.

Doerner, K., Gutjahr, W. J., Hartl, R. F., Strauss, C., & Stummer, C. (2004). Pareto ant colony optimization: A metaheuristic approach to multiobjective portfolio selection. *Annals of operations research*, *131*(1), 79-99.

Dokeroglu, T., Deniz, A., & Kiziloz, H. E. (2022). A comprehensive survey on recent metaheuristics for feature selection. *Neurocomputing*, *494*, 269-296.https://doi.org/10.1016/j.neucom.2022.04.083

Domingos, E., Ojeme, B., & Daramola, O.J. (2021). Experimental analysis of hyperparameters for deep learning-based churn prediction in the banking sector. *Computation, 9(3)*, 34. https://doi.org/10.3390/computation9030034

Dootson, P., Beatson, A., & Drennan, J. (2016). Financial institutions using social media–do consumers perceive value?. *International Journal of Bank Marketing*, *34*(1), 9-36. https://doi.org/10.1108/IJBM-06-2014-0079

Doumpos, M., Zopounidis, C., Gounopoulos, D., Platanakis, E., & Zhang, W. (2023). Operational research and artificial intelligence methods in banking. *European Journal of Operational Research*, *306*(1), 1-16.https://doi.org/10.1016/j.ejor.2022.04.027

Dorigo, M., Birattari, M., & Stutzle, T. (2007). Ant colony optimization. *IEEE computational intelligence magazine*, 1(4), 28-39. https://doi.org/10.1109/MCI.2006.329691

Di Persio, L., & Honchar, O. (2017). Recurrent neural networks approach to the financial forecast of Google assets. *International journal of Mathematics and Computers in simulation*, *11*, 7-13.

Di Tollo, G., & Roli, A. (2008). Metaheuristics for the portfolio selection problem. International Journal of Operations Research, 5(1), 13-35.

Du, G., Liu, Z., & Lu, H. (2021). Application of innovative risk early warning mode under big data technology in Internet credit financial risk assessment. *Journal of Computational and Applied Mathematics*, *386*, 113260.https://doi.org/10.1016/j.cam.2020.113260

Duchi, J., Hazan, E. & Singer, Y. (2011). Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research*, *12*, 2121-2159.

Dueck, G., & Winker, P. (1992). New concepts and algorithms for portfolio choice. Applied stochastic models and data analysis, 8(3), 159-178. https://doi.org/10.1002/asm.3150080306

Durai, T., & Stella, G. (2019). Digital finance and its impact on financial inclusion. *Journal of Emerging Technologies and Innovative Research*, *6*(1), 122-127.https://doi.org/10.1111/jacf.12267

Durkaya Kurtcan, B., &Ozcan, T. (2023). Predicting customer churn using grey wolf optimization-based support vector machine with principal component analysis. *Journal of Forecasting,42*(6), 1329–1340.https://doi.org/10.1002/for.2960

Dupačová, J., & Kopa, M. (2014). Robustness of optimal portfolios under risk and stochastic dominance constraints. European Journal of Operational Research, 234(2), 434-441. https://doi.org/10.1016/j.ejor.2013.06.018

Dutta, A., Kumar, S., & Basu, M. (2020). A gated recurrent unit approach to bitcoin price prediction. *Journal of risk and financial management*, *13*(2), 23. https://doi.org/10.3390/jrfm13020023

Eccles, R. G., Kastrapeli, M. D., & Potter, S. J. (2017). How to integrate ESG into investment decision-making: Results of a global survey of institutional investors. *Journal of Applied Corporate Finance*, *29*(4), 125-133.

Edmans A, Kacperczyk M (2022). Sustainable finance. Review of Finance 26(*6*),1309–1313 https://doi.org/10.1093/rof/rfac069

Elena, C. A. (2016). Social media–a strategy in developing customer relationship management. *Procedia Economics and Finance*, *39*, 785-790.https://doi.org/10.1016/S2212-5671(16)30266-0

Elfwing, S., Uchibe, E., & Doya, K. (2017). Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Networks*, *107*, 3-11. https://doi.org/10.1016/j.neunet.2017.12.012

Erwin, K., & Engelbrecht, A. (2023). Meta-heuristics for portfolio optimization. *Soft Computing*, *27*(24), 19045-19073.

Estabrooks, A., Jo, T., & Japkowicz, N. (2004). A multiple resampling method for learning from imbalanced data sets. *Computational intelligence*, *20*(1), 18-36.

Fang, B., & Zhang, P. (2016). Big data in finance. In *Big data concepts, theories, and applications* (pp. 391-412). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-27763-9_11

Fang, W., Zhang, S., & Xu, C. (2024). Improving prediction efficiency of Chinese stock index futures intraday price by VIX-Lasso-GRU Model. *Expert Systems with Applications*, *238*, 121968. https://doi.org/10.1016/j.eswa.2023.121968

Farboodi, M., & Veldkamp, L. (2023). Data and markets. *Annual Review of Economics*, *15*(1), 23-40.https://doi.org/10.1146/annurev-economics-082322-023244

Fares, O. H., Butt, I., & Lee, S. H. M. (2022). Utilization of artificial intelligence in the banking sector: a systematic literature review. *Journal of Financial Services Marketing*, 1.

Fernández, A., García, S., Galar, M., Prati, R. C., Krawczyk, B., & Herrera, F. (2018). *Learning from imbalanced data sets*,*10*,2018. Cham: Springer. 10.1007/978-3-319-98074-4

Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European journal of operational research*, *270*(2), 654-669. https://doi.org/10.1016/j.ejor.2017.11.054

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 1189–1232. https://doi.org/10.1214/aos/1013203451

Galindo, J., & Tamayo, P. (2000). Credit risk assessment using statistical and machine learning: Basic methodology and risk modelling applications. *Computational Economics, 15*, 107–143. https://doi.org/10.1023/A:1008699112516

Gao, Y., Wang, R., & Zhou, E. (2021). Stock prediction based on optimized LSTM and GRU models. *Scientific Programming*, *2021*(1), 4055281. https://doi.org/10.1155/2021/4055281

Gao, J., Sun, W., & Sui, X. (2021). Research on default prediction for credit card users based on XGBoost–LSTM model. *Discrete Dynamics in Nature and Society*, 1–13. https://doi.org/10.1155/2021/5080472

Gan, Z., Qiu, J., Li, F., & Qian, L. (2023). A LightGBM-based default prediction method for American Express. In *Proc. 2nd International Conference on Information Economy, Data Modeling and Cloud Computing (ICIDC 2023)*, Nanchang, China, June 2–4, 2023.

Geiler, L., Affeldt, S., & Nadif, M. (2022). A survey on machine learning methods for churn prediction. *International Journal of Data Science and Analytics, 14(3),* 217-242. https://doi.org/10.1007/s41060-022-00312-5

Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to forget: Continual prediction with LSTM. Neural computation, 12(10), 2451-2471. https://doi.org/10.1162/089976600300015015

Glover, F. (1989). Tabu search—part I. *ORSA Journal on computing*, *1*(3), 190-206.

Goldstein, I., Spatt, C. S., & Ye, M. (2021). Big data in finance. *The Review of Financial Studies*, *34*(7), 3213-3225.https://doi.org/10.1093/rfs/hhab038

Gómez, D., & Rojas, A. (2016). An empirical overview of the no free lunch theorem and its effect on real-world machine learning classification. *Neural computation*, *28*(1), 216-228.https://doi.org/10.1162/NECO_a_00793

Goodell, J. W., Kumar, S., Lim, W. M., & Pattnaik, D. (2021). Artificial intelligence and machine learning in finance: Identifying foundations, themes, and research clusters from bibliometric analysis. *Journal of Behavioral and Experimental Finance*, *32*, 100577.http://doi.org/10.21272/fmir.3(4).74-88.2019

Göçken, M., Özçalıcı, M., Boru, A., & Dosdoğru, A. T. (2016). Integrating metaheuristics and artificial neural networks for improved stock price prediction. *Expert Systems with Applications*, *44*, 320-331. https://doi.org/10.1016/j.eswa.2015.09.029

Ghorbani, R., & Ghousi, R. (2020). Comparing different resampling methods in predicting students' performance using machine learning techniques. *IEEE Access*, *8*, 67899-67911. https://doi.org/10.1109/ACCESS.2020.2986809

Giebe, C., Hammerström, L., & Zwerenz, D. (2019). Big data & analytics as a sustainable customer loyalty instrument in banking and finance.

Gkonis, V., & Tsakalos, I. (2025). Deep dive into churn prediction in the banking sector: The challenge of hyperparameter selection and imbalanced learning. *Journal of Forecasting, 44*(2), 281–296. https://doi.org/10.1002/for.3194

Grishina, N., Lucas, C. A., & Date, P. (2017). Prospect theory–based portfolio optimization: an empirical study and analysis using intelligent algorithms. *Quantitative Finance, 17*(3), 353-367. https://doi.org/10.1080/14697688.2016.1149611

Gülmez, B. (2023). Stock price prediction with optimized deep LSTM network with artificial rabbits optimization algorithm. *Expert Systems with Applications*, *227*, 120346. https://doi.org/10.1016/j.eswa.2023.120346

Haddadi, S. J., Farshidvard, A., dos Santos Silva, F., dos Reis, J. C., & da Silva Reis, M. (2024). Customer churn prediction in imbalanced datasets with resampling methods: A comparative study. *Expert Systems with Applications*, *246*, 123086. https://doi.org/10.1016/j.eswa.2023.123086

Hamayel, M. J., & Owda, A. Y. (2021). A novel cryptocurrency price prediction model using GRU, LSTM and bi-LSTM machine learning algorithms. *AI*, *2*(4), 477-496. https://doi.org/10.3390/ai2040030

Hancock, J. T., & Khoshgoftaar, T. M. (2021). Gradient boosted decision tree algorithms for Medicare fraud detection. *SN Computer Science, 2*(4), 268. https://doi.org/10.1007/s42979-021-00655-z

Hand, D. J., & Henley, W. E. (1997). Statistical classification methods in consumer credit scoring: A review. *Journal of the Royal Statistical Society: Series A, 160*(3), 523–541. https://doi.org/10.1111/j.1467-985X.1997.00078.x

Hanley, J. A., & McNeil, B. J. (1982). The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology, 143*(1), 29–36.

He, B., Shi, Y., Wan, Q., & Zhao, X. (2014). Prediction of customer attrition of commercial banks based on SVM model. *Procedia Computer Science*, *31*, 423–430. https://doi.org/10.1016/j.procs.2014.05.286

He, H., Zhang, W., & Zhang, S. (2018). A novel ensemble method for credit scoring: Adaption of different imbalance ratios. *Expert Systems with Applications, 98*, 105–117. https://doi.org/10.1016/j.eswa.2018.01.012

Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., & Chen, H. (2019). Harris hawks optimization: Algorithm and applications. Future generation computer systems, 97, 849-872. https://doi.org/10.1016/j.future.2019.02.028

Hellekalek, P. (1984). Regularities in the distribution of special sequences. *Journal of Number Theory, 18*(1), 41–55.

Hochreiter, S. (1991). Untersuchungen zu dynamischen neuronalen Netzen. Diploma, Technische Universität München, 91(1), 31.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735

Holland, J. H. (1992). Genetic algorithms. *Scientific american*, *267*(1), 66-73.

Höppner, S., Stripling, E., Baesens, B., vanden Broucke, S., & Verdonck, T. (2020). Profit driven decision trees for churn prediction. *European journal of operational research*, *284*(3), 920-933. https://doi.org/10.1016/j.ejor.2018.11.072

Hou, L., Bi, G., & Guo, Q. (2025). An improved sparrow search algorithm optimized LightGBM approach for credit risk prediction of SMEs in supply chain finance. *Journal of Computational and Applied Mathematics, 454*, 116197.

Hernández-Murillo, R., Llobet, G., & Fuentes, R. (2010). Strategic online banking adoption. *Journal of Banking & Finance*, *34*(7), 1650-1663.https://doi.org/10.1016/j.jbankfin.2010.03.011

Hess, S., & Polak, J. (2003). An alternative method to the scrambled Halton sequence for removing correlation between standard Halton sequences in high dimensions. *ERSA Conference Papers*.

Hipgrave, S. (2013). Smarter fraud investigations with big data analytics. *Network Security*, *2013*(12), 7-9.https://doi.org/10.1016/S1353-4858(13)70135-1

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *science*, *313*(5786), 504-507.

Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. *Neural computation*, *18*(7), 1527-1554.

Huang, C. L., Chen, M. C., & Wang, C. J. (2007). Credit scoring with a data mining approach based on support vector machines. *Expert systems with applications*, *33*(4), 847-856. https://doi.org/10.1016/j.eswa.2006.07.007

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in science & engineering*, *9*(03), 90-95.

Imani, M., & Arabnia, H. R. (2023). Hyperparameter Optimization and Combined Data Sampling Techniques in Machine Learning for Customer Churn Prediction: A Comparative Analysis. *Technologies*, *11*(6), 167. https://doi.org/10.3390/technologies11060167

Iqbal, S., N. Qureshi, A., Li, J., & Mahmood, T. (2023). On the analyses of medical images using traditional machine learning techniques and convolutional neural networks. *Archives of Computational Methods in Engineering*, *30*(5), 3173-3233.https://doi.org/10.1007/s11831-023-09899-9

Jadhav, S., He, H., & Jenkins, K. (2018). Information gain directed genetic algorithm wrapper feature selection for credit rating. *Applied Soft Computing, 69*, 541–553. https://doi.org/10.1016/j.asoc.2018.04.033

James, J. Q., & Li, V. O. (2015). A social spider algorithm for global optimization. Applied soft computing, 30, 614-627. https://doi.org/10.1016/j.asoc.2015.02.014

Ji, Y., Liew, A. W. C., & Yang, L. (2021). A novel improved particle swarm optimization with long-short term memory hybrid model for stock indices forecast. *IEEE Access*, *9*, 23660-23671. https://doi.org/10.1109/ACCESS.2021.3056713

Jiao, L., & Zhao, J. (2019). A survey on the new generation of deep learning in image processing. *Ieee Access*, *7*, 172231-172263.https://doi.org/10.1109/ACCESS.2019.2956508

Jiang, X., & Li, S. (2017). Beetle antennae search without parameter tuning (BAS-WPT) for multi-objective optimization. https://arxiv.org/abs/1710.10724

Jiang, W. (2021). Applications of deep learning in stock market prediction: recent progress. *Expert Systems with Applications*, *184*, 115537. https://doi.org/10.1016/j.eswa.2021.115537

Jin, X., Chen, N., & Yuan, Y. (2019). Multi-period and tri-objective uncertain portfolio selection model: A behavioral approach. *The North American journal of economics and finance, 47,* 492-504. https://doi.org/10.1016/j.najef.2018.06.005

Jin, J., Lu, Y., Qin, J., Cheng, Z., & Mao, Z. (2020). SwiftIDS: Real-time intrusion detection system based on LightGBM and parallel intrusion detection mechanism. *Computers & Security, 97*, 101984. https://doi.org/10.1016/j.cose.2020.101984

Jo, N. O., & Shin, K. S. (2016). Bankruptcy prediction modeling using qualitative information based on big data analytics. 지능정보연구, 22(2), 33-56. https://doi.org/10.13088/jiis.2016.22.2.033

Kalayci, C. B., Polat, O., & Akbay, M. A. (2020). An efficient hybrid metaheuristic algorithm for cardinality constrained portfolio optimization. *Swarm and Evolutionary Computation*, 54, 100662. https://doi.org/10.1016/j.swevo.2020.100662

Kang, Q. (2019). Financial risk assessment model based on big data. *International Journal of Modeling, Simulation, and Scientific Computing*, *10*(04), 1950021.https://doi.org/10.1142/S1793962319500211

Kaushik, D., & Nadeem, M. (2025). Portfolio optimization using batch enabled particle swarm optimization. In *Intelligent Computing and Communication Techniques* (pp. 89-94). CRC Press.

Kaveh, M., & Mesgari, M. S. (2023). Application of meta-heuristic algorithms for training neural networks and deep learning architectures: A comprehensive review. *Neural Processing Letters*, *55*(4), 4519-4622.https://doi.org/10.1007/s11063-022-11055-6

Kaya, O., Schildbach, J., AG, D. B., & Schneider, S. (2019). Artificial intelligence in banking. *Artificial intelligence* 1(1),1–9

Khan, A. T., Cao, X., Brajevic, I., Stanimirovic, P. S., Katsikis, V. N., & Li, S. (2022). Non-linear activated beetle antennae search: A novel technique for non-convex tax-aware portfolio optimization problem. *Expert Systems with Applications, 197,* 116631. https://doi.org/10.1016/j.eswa.2022.116631

Ko, J., Park, S., & Woo, H. G. (2024). Optimization of vision transformer-based detection of lung diseases from chest X-ray images. *BMC Medical Informatics and Decision Making*, *24*(1), 191

Kocis, L., & Whiten, W. J. (1997). Computational investigations of low-discrepancy sequences. *ACM Transactions on Mathematical Software, 23*(2), 266–294.

Kontopoulou, V. I., Panagopoulos, A. D., Kakkos, I., & Matsopoulos, G. K. (2023). A review of ARIMA vs. machine learning approaches for time series forecasting in data driven networks. *Future Internet*, *15*(8), 255.https://doi.org/10.3390/fi15080255

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980.*

Kirkpatrick, S., Gelatt Jr, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *science, 220*(4598), 671-680. https://doi.org/10.1126/science.220.4598.671

Kriouich, M., & Sarir, H. (2024, April). Artificial intelligence application in production scheduling problem systematic literature review: bibliometric analysis, research trend, and knowledge taxonomy. In *Operations Research Forum* (Vol. 5, No. 2, p. 29). Cham: Springer International Publishing.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, *25*.

Kumar, K. V., & Devi, V. R. (2014). Social media in financial services–a theoretical perspective. *Procedia Economics and Finance*, *11*, 306-313.https://doi.org/10.1016/S2212-5671(14)00198-1

Kumar, R. (2022). APTx: better activation function than MISH, SWISH, and ReLU's variants used in deep learning. *arXiv preprint arXiv:2209.06119.*

Kumar, G., Singh, U. P., & Jain, S. (2022a). An adaptive particle swarm optimization-based hybrid long short-term memory model for stock price time series forecasting. *Soft Computing*, *26*(22), 12115-12135. https://doi.org/10.1007/s00500-022-07451-8

Kumar, G., Singh, U. P., & Jain, S. (2022b). Swarm intelligence based hybrid neural network approach for stock price forecasting. *Computational Economics*, *60*(3), 991-1039. https://doi.org/10.1007/s10614-021-10176-9

Kurani, A., Doshi, P., Vakharia, A., & Shah, M. (2023). A comprehensive comparative study of artificial neural network (ANN) and support vector machines (SVM) on stock forecasting. *Annals of Data Science*, *10*(1), 183-208.https://doi.org/10.1007/s40745-021-00344-x

Lemaître, G., Nogueira, F., & Aridas, C. K. (2017). Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of machine learning research*, 18(17), 1-5.

Lotka, A. J. (1926). The frequency distribution of scientific productivity. *Journal of the Washington academy of sciences*, *16*(12), 317-323.

López, V., Fernández, A., García, S., Palade, V., & Herrera, F. (2013). An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Information Sciences., 250*, 113-141. https://doi.org/10.1016/j.ins.2013.07.007

Lisin, A., Kushnir, A., Koryakov, A. G., Fomenko, N., & Shchukina, T. (2022). Financial stability in companies with high ESG scores: evidence from North America using the Ohlson O-Score. *Sustainability*, *14*(1), 479. https://doi.org/10.3390/su14010479

Liu, L., Jiang, H., He, P., Chen, W., Liu, X., Gao, J., et.al. (2019). On the variance of the adaptive learning rate and beyond. *arXiv preprint arXiv:1908.03265*.

Liu, K., Zhou, J., & Dong, D. (2021). Improving stock price prediction using the long short-term memory model combined with online social networks. *Journal of Behavioral and Experimental Finance*, *30*, 100507. https://doi.org/10.1016/j.jbef.2021.100507

Liu, M., Yao, D., Liu, Z., Guo, J., & Chen, J. (2023). An improved Adam optimization algorithm combining adaptive coefficients and composite gradients based on randomized block coordinate descent. *Computational intelligence and neuroscience*, *2023*(1), 4765891. https://doi.org/10.1155/2023/4765891

Liu, J., & Wang, Z. (2024). Financial distress prediction using an improved particle swarm optimization wrapper feature selection method and tree boosting ensemble. *Journal of the Operational Research Society*, 1–24. https://doi.org/10.1080/01605682.2024.2385467

Liu, X., Wu, Y., Luo, M., & Chen, Z. (2024). Stock price prediction for new energy vehicle companies based on multi-source data and hybrid attention structure. *Expert Systems with Applications*, *255*, 124787. https://doi.org/10.1016/j.eswa.2024.124787

Lu, L., Shin, Y., Su, Y., & Karniadakis, G. E. (2019). Dying relu and initialization: Theory and numerical examples. arXiv preprint arXiv:1903.06733.

Łukasik, S., Żak, S. (2009). Firefly algorithm for continuous constrained optimization tasks. *Computational Collective Intelligence: Semantic Web, Social Networks and Multiagent Systems*, Wroclaw, Poland, 5-7 October. https://doi.org/10.1007/978-3-642-04441-0_8

Maas, A.L., Hannun, A.Y. and Ng, A.Y. (2013) Rectifier Nonlinearities Improve Neural Network Acoustic Models. *Proceedings of the 30th International Conference on Machine Learning*, *28*, 3.

Malhotra, R., & Malhotra, D. K. (2000). Identifying potential loan defaulters in the credit union environment: A comparative analysis of statistical and neural network models. *Journal of Information Technology Case and Application Research, 2*(2), 20–48. https://doi.org/10.1080/15228053.2000.10855952

Masini, R. P., Medeiros, M. C., & Mendes, E. F. (2023). Machine learning advances for time series forecasting. *Journal of economic surveys*, *37*(1), 76-111.

Masciandaro, D., Peia, O., & Romelli, D. (2024). Central bank communication and social media: From silence to Twitter. *Journal of Economic Surveys*, *38*(2), 365-388. https://doi.org/10.1111/joes.12550

Mazumder, A., Ghosh, S., Roy, S., Dhar, S., & Jana, N. D. (2022). Rectified adam optimizer-based cnn model for speaker identification. *In Advances in Intelligent Computing and Communication: Proceedings of ICAC 2021*, pp.155-162. Singapore: Springer Nature Singapore.

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, *5*(4), 115-133.

Medvedeva, M. A., Katsikis, V. N., Mourtas, S. D., & Simos, T. E. (2021). Randomized time-varying knapsack problems via binary beetle antennae search algorithm: Emphasis on applications in portfolio insurance. *Mathematical Methods in the Applied Sciences, 44*(2), 2002-2012. https://doi.org/10.1002/mma.6904

Metaxiotis, K., & Liagkouras, K. (2012). Multiobjective evolutionary algorithms for portfolio management: A comprehensive literature review. *Expert systems with applications*, *39*(14), 11685-11698.

Minsky, M., & Papert, S. (1969). Perceptrons," MIT Press, Cambridge, MA. *USA*.

Mishra, S. K., Panda, G., & Majhi, R. (2014). A comparative performance assessment of a set of multiobjective algorithms for constrained portfolio assets selection. *Swarm and Evolutionary Computation, 16,* 38-51. https://doi.org/10.1016/j.swevo.2014.01.001

Mishra, S., Padhy, S., Mishra, S. N., & Misra, S. N. (2021). A novel LASSO–TLBO–SVR hybrid model for an efficient portfolio construction. *The North American Journal of Economics and Finance, 55,* 101350. https://doi.org/10.1016/j.najef.2020.101350

Mitrović, S., Baesens, B., Lemahieu, W., & De Weerdt, J. (2018). On the operational efficiency of different feature types for telco Churn prediction. *European Journal of Operational Research*, *267*(3), 1141-1155. https://doi.org/10.1016/j.ejor.2017.12.015

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in engineering software*, 69, 46-61. https://dx.doi.org/10.1016/j.advengsoft.2013.12.007

Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. Advances in engineering software, 95, 51-67. https://dx.doi.org/10.1016/j.advengsoft.2016.01.008

Mirjalili, S., & Lewis, A. (2013). S-shaped versus V-shaped transfer functions for binary particle swarm optimization. *Swarm and Evolutionary Computation*, 9, 1-14. https://doi.org/10.1016/j.swevo.2012.09.002

Mirjalili, S. (2015). The ant lion optimizer. *Advances in engineering software*, 83, 80-98. https://dx.doi.org/10.1016/j.advengsoft.2015.01.010

Morales-Hernández, A., Van Nieuwenhuyse, I., & Rojas Gonzalez, S. (2023). A survey on multi-objective hyperparameter optimization algorithms for machine learning. *Artificial Intelligence Review*, *56*(8), 8043-8093.https://doi.org/10.1007/s10462-022-10359-2

Mourtas, S. D., & Katsikis, V. N. (2022). V-Shaped BAS: Applications on large portfolios selection problem. *Computational Economics, 60*(4), 1353-1373. https://dx.doi.org/10.1007/s10614-021-10184-9

Mozafari, M., Jolai, F., & Tafazzoli, S. (2011). A new IPSO-SA approach for cardinality constrained portfolio optimization. *International Journal of Industrial Engineering Computations*, 2(2), 249-262.https://dx.doi.org/10.5267/j.ijiec.2011.01.004

Mu, G., Gao, N., Wang, Y., & Dai, L. (2023). A stock price prediction model based on investor sentiment and optimized deep learning. *IEEE Access*, *11*, 51353-51367. https://doi.org/10.1109/ACCESS.2023.3278790

Nabipour, M., Nayyeri, P., Jabani, H., Mosavi, A., & Salwana, E. (2020). Deep learning for stock market prediction. *Entropy*, *22*(8), 840. https://doi.org/10.3390/e22080840

Narayanan, R., & Ganesh, N. (2024). A Comprehensive Review of Metaheuristics for Hyperparameter Optimization in Machine Learning. *Metaheuristics for Machine Learning: Algorithms and Applications*, 37-72. https://doi.org/10.1002/9781394233953.ch2

Nssibi, M., Manita, G., & Korbaa, O. (2023). Advances in nature-inspired metaheuristic optimization for feature selection problem: A comprehensive survey. *Computer Science Review*, *49*, 100559.https://doi.org/10.1016/j.cosrev.2023.100559

Owen, A. B. (1995). Randomly permuted (t, m, s)-nets and (t, s)-sequences. In *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing* (pp. 299–317). Springer.

Owen, A. B. (2006). Halton sequences avoid the origin. *SIAM Review, 48*(3), 487–503.

Owen, A. B. (2017). A randomized Halton algorithm in R. *arXiv preprint* arXiv:1706.02808.

Pagnoncelli, B. K., Cifuentes, A., & Denis, G. (2017). A two-step hybrid investment strategy for pension funds. *The North American Journal of Economics and Finance*, 42, 574-583.https://doi.org/10.1016/j.najef.2017.09.001

Pan, W. T. (2012). A new fruit fly optimization algorithm: taking the financial distress model as an example. *Knowledge-Based Systems*, *26*, 69-74.https://doi.org/10.1016/j.knosys.2011.07.001

Pareto, V. (1897). *Cours d'économie politique: professé à l'Université de Lausanne*. F. Rouge.

Paula, B., Torgo, L., & Ribeiro, R. (2015). A survey of predictive modelling under imbalanced distributions. *arXiv preprint arXiv*, 1505(01658).

Paz, Á., Crawford, B., Monfroy, E., Barrera-García, J., Peña Fritz, Á., Soto, R., ... & Yáñez, A. (2025). Machine Learning and Metaheuristics Approach for Individual Credit Risk Assessment: A Systematic Literature Review. *Biomimetics*, *10*(5), 326. https://doi.org/10.3390/biomimetics10050326

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *The Journal of machine Learning research*, 12, 2825-2830.

Pierezan, J., Coelho, L. D. S. (2018). Coyote optimization algorithm: a new metaheuristic for global optimization problems. 2018 IEEE Congress on Evolutionary Computation (CEC), Rio de Janeiro, Brazil, 8-13 July, pp. 8477769. https://dx.doi.org/10.1109/CEC.2018.8477769

Ponsich, A., Jaimes, A. L., & Coello, C. A. C. (2012). A survey on multiobjective evolutionary algorithms for the solution of the portfolio optimization problem and other finance and economics applications. *IEEE Transactions on evolutionary computation*, *17*(3), 321-344.

Pritchard, A. (1969). Statistical bibliography or bibliometrics. *Journal of documentation*, *25*, 348.

Radcliffe, N. J. (1993). Genetic set recombination. In: Whitley, D. (Ed.), Foundations of Genetic Algorithms 2, San Mateo, CA: Morgan Kaufmann, pp. 203-219. https://doi.org/10.1016/B978-0-08-094832-4.50019-2

Raguseo, E., & Vitari, C. (2018). Investments in big data analytics and firm performance: an empirical investigation of direct and mediating effects. *International Journal of Production Research*, *56*(15), 5206-5221.https://doi.org/10.1080/00207543.2018.1427900

Rajwar, K., Deep, K., & Das, S. (2023). An exhaustive review of the metaheuristic algorithms for search and optimization: taxonomy, applications, and open challenges. *Artificial Intelligence Review*, *56*(11), 13187-13257. https://doi.org/10.1007/s10462-023-10470-y

Ramachandran, P., Zoph, B., & Le, Q. V. (2017). Swish: A self-gated activation function. *arXiv preprint arXiv:1710.05941*.

Ren, J., Dong, H., Popovic, A., Sabnis, G., & Nickerson, J. (2024). Digital platforms in the news industry: how social media platforms impact traditional media news viewership. *European journal of information systems*, *33*(1), 1-18.https://doi.org/10.1080/0960085X.2022.2103046

Reyad, M., Sarhan, A., & Arafa, M. (2023). A modified Adam algorithm for deep neural network optimization. *Neural Computing and Applications*, *35*(12), 17095-17112. https://doi.org/10.1007/s00521-023-08568-z

Rex, C. E. S., Annrose, J., & Jose, J. J. (2022). Comparative analysis of deep convolution neural network models on small scale datasets. *Optik*, *271*, 170238. https://doi.org/10.1016/j.ijleo.2022.170238

Ribeiro, G. T., Santos, A. A. P., Mariani, V. C., & dos Santos Coelho, L. (2021). Novel hybrid model based on echo state neural network applied to the prediction of stock price return volatility. *Expert Systems with Applications*, *184*, 115490. https://doi.org/10.1016/j.eswa.2021.115490

Rodríguez-Ibánez, M., Casánez-Ventura, A., Castejón-Mateos, F., & Cuenca-Jiménez, P. M. (2023). A review on sentiment analysis from social media platforms. *Expert Systems with Applications*, *223*, 119862.https://doi.org/10.1016/j.eswa.2023.119862

Rodrigues, R. M. D. S. V., Santana, T. P., Dias, R. M. T. S., da Silva Filho, A. M., Galvão, R., Zebende, G. F., & Gonçalves, S. (2024). ECONOPHYSICS, STATE OF THE SCIENCE AND BIBLIOGRAPHIC PRODUCTION: A BIBLIOMETRIC ANALYSIS. *Revista de Gestão Social e Ambiental*, *18*(11), 1-25.https://doi.org/10.24857/rgsa.v18n11-234

Ruiz-Torrubiano, R., & Suárez, A. (2015). A memetic algorithm for cardinality-constrained portfolio optimization with transaction costs. Applied Soft Computing, 36, 125-142. *Applied Soft Computing, 36*, 125-142. https://dx.doi.org/10.1016/j.asoc.2015.06.053

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, *323*(6088), 533-536.

Sadeghian, Z., Akbari, E., Nematzadeh, H., & Motameni, H. (2025). A review of feature selection methods based on meta-heuristic algorithms. *Journal of Experimental & Theoretical Artificial Intelligence*, *37*(1), 1-51.

Sana, J. K., Abedin, M. Z., Rahman, M. S., & Rahman, M. S. (2022). A novel customer churn prediction model for the telecommunication industry using data transformation methods and feature selection. *PLOS ONE, 17(12).* https://doi.org/10.1371/journal.pone.0278095

Saiz, M., Lostumbo, M. A., Juan, A. A., & Lopez-Lopez, D. (2022). A clustering-based review on project portfolio optimization methods. *International Transactions in Operational Research, 29(1)*, 172-199. https://doi.org/10.1111/itor.12933

Saremi, S., Mirjalili, S., & Lewis, A. (2017). Grasshopper optimisation algorithm: theory and application. *Advances in engineering software*, *105*, 30-47.

Sarigül, M., & Avci, M. (2018). Performance comparison of different momentum techniques on deep reinforcement learning. *Journal of Information and Telecommunication*, *2*(2), 205-216. https://doi.org/10.1080/24751839.2018.1440453

Sariannidis, N., Papadakis, S., Garefalakis, A., Lemonakis, C., & Kyriaki-Argyro, T. (2020). Default avoidance on credit card portfolios using accounting, demographical and exploratory factors: Decision making based on machine learning techniques. *Annals of Operations Research, 294*(1), 715–739. https://doi.org/10.1007/s10479-019-03188-0

Shadravan, S., Naji, H. R., & Bardsiri, V. K. (2019). The Sailfish Optimizer: A novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems. Engineering Applications of Artificial Intelligence, 80, 20-34. https://dx.doi.org/10.1016/j.engappai.2019.01.001

Shahvaroughi Farahani, M., & Razavi Hajiagha, S. H. (2021). Forecasting stock price using integrated artificial neural network and metaheuristic algorithms compared to time series models. *Soft computing*, *25*(13), 8483-8513. https://doi.org/10.1007/s00500-021-05775-5

Sharifi, P., Jain, V., Arab Poshtkohi, M., Seyyedi, E., & Aghapour, V. (2021). Banks credit risk prediction with optimized ANN based on improved owl search algorithm. *Mathematical Problems in Engineering*, 8458501. https://doi.org/10.1155/2021/8458501

Shaqfa, M., & Orbán, Z. (2019). Modified parameter-setting-free harmony search (PSFHS) algorithm for optimizing the design of reinforced concrete beams. *Structural and Multidisciplinary Optimization*, *60*(3), 999-1019.

Shaqfa, M., & Beyer, K. (2021). Pareto-like sequential sampling heuristic for global optimisation. *Soft Computing*, *25*(14), 9077-9096.

Şen, Ç. Ç., Dönmez, Ç., & Yıldırım, U. M. (2020). A hybrid bi-level metaheuristic for credit scoring. *Information Systems Frontiers, 22*(5), 1009–1019. https://doi.org/10.1007/s10796-020-10037-0

Si-Ma, S., Liu, H. M., Zhan, H. X., Liu, Z. F., Guo, G., Yu, C., & Hu, P. C. (2025). Efficient maximum iterations for swarm intelligence algorithms: a comparative study. *Artificial Intelligence Review*, 58(3), 87. https://doi.org/10.1007/s10462-024-11104-7

Simsek, M., & Tas, I. C. (2024). A classification application for usinglearning methods in bank costumer's portfolio churn. *Journal of Forecasting*, *43*(2), 391–401.https://doi.org/10.1002/for.3038

Singh, D. P., Kothari, S., Aditya, S., Rajora, M., Agarwal, C., & Gautam, V. (2024). A hybrid metaheuristic optimized ensemble classifier with self-organizing map clustering for credit scoring. *Operational Research, 24*(4), 1–42. https://doi.org/10.1007/s12351-024-00864-3

Singh, R., & Dwivedi, G. (2025). Agri-food Supply Chain Management: A Review Using Bibliometric, Network, and Content Analyses. In *Operations Research Forum* (Vol. 6, No. 2, p. 62). Cham: Springer International Publishing.

Srinivasan, V., & Kim, Y. H. (1987). Credit granting: A comparative analysis of classification procedures. *Journal of Finance, 11*(1), 23–42. https://doi.org/10.1111/j.1540-6261.1987.tb04576.x

Sun, T., & Vasarhelyi, M. A. (2018). Predicting credit card delinquencies: An application of deep neural networks. *Intelligent Systems in Accounting, Finance and Management, 25*(4), 174–189. https://doi.org/10.1002/isaf.1437

Sun, Z., Song, Q., Zhu, X., Sun, H., Xu, B., & Zhou, Y. (2015). A novel ensemble method for classifying imbalanced data. *Pattern Recognition*, *48*(5), 1623-1637.https://doi.org/10.1016/j.patcog.2014.11.014

Sun, Y., Shi, Y., & Zhang, Z. (2019). Finance big data: management, analysis, and applications. *International Journal of Electronic Commerce*, *23*(1), 9-11. https://doi.org/10.1080/10864415.2018.1512270

Sun, H., Rabbani, M. R., Sial, M. S., Yu, S., Filipe, J. A., & Cherian, J. (2020). Identifying big data's opportunities, challenges, and implications in finance. *Mathematics*, *8*(10), 1738. https://doi.org/10.1016/j.eswa.2024.125192

Szénási, S., & Légrádi, G. (2024). Machine learning aided metaheuristics: A comprehensive review of hybrid local search methods. *Expert Systems with Applications*, *258*, 125192.

Szopiński, T. S. (2016). Factors affecting the adoption of online banking in Poland. *Journal of business research*, *69*(11), 4763-4768.https://doi.org/10.1016/j.jbusres.2016.04.027

Tao, H., Zhuang, S., Xue, R., Cao, W., Tian, J., & Shan, Y. (2022). Environmental finance: an interdisciplinary review. *Technological Forecasting and Social Change*, *179*, 121639. https://doi.org/10.1016/j.techfore.2022.121639

Tarawneh, A. S., Hassanat, A. B., Altarawneh, G. A., & Almuhaimeed, A. (2022). Stop oversampling for class imbalance learning: A review. *IEEE Access*, *10*, 47643-47660.

Tayebi, M., & El Kafhali, S. (2024). Performance analysis of metaheuristics based hyperparameters optimization for fraud transactions detection. *Evolutionary intelligence*, *17*(2), 921-939.

Tieleman, T., & Hinton, G. (2012). Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning, 4* (2), 26–30.

Tomek I. I. (1976). Two modifications of CNN. In *IEEE Transactions on Systems, Man, and Cybernetics*, *6*, 769-772.

Utama, D. M., Abdullah, F. F., Wangsa, I. D., & Jauhari, W. A. (2025, January). The Vendor–Buyer Inventory Model in Two Echelon Supply Chain: A Systematic Literature Review and Bibliometric Analysis. In *Operations Research Forum* (Vol. 6, No. 1, p. 11). Cham: Springer International Publishing.

Vafeiadis, T., Diamantaras, K. I., Sarigiannidis, G., & Chatzisavvas, K. C. (2015). A comparison of machine learning techniques for customer churn prediction. *Simulation Modelling Practice Theory, 55*, 1-9. https://doi.org/10.1016/j.simpat.2015.03.003

Van Eck, N. J., & Waltman, L. (2007). Bibliometric mapping of the computational intelligence field. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *15*(05), 625-645.

Van Eck NJ, & Waltman L. (2010). Software survey: VOSviewer, a computer program for bibliometric mapping. Scientometrics, 84 (2), 523-538. https://doi.org/10.1007/s11192-009-0146-3

Van Eck NJ, Waltman L (2013) VOSviewer manual. Leiden University, Leiden, 1(1),1–53

Van Nunen, K., Li, J., Reniers, G., & Ponnet, K. (2018). Bibliometric analysis of safety culture research. *Safety science*, *108*, 248-258.

Van Thieu, N., & Mirjalili, S. (2023). MEALPY: An open-source library for latest meta-heuristic algorithms in Python. *Journal of Systems Architecture*, *139*, 102871. https://doi.org/10.1016/j.sysarc.2023.102871

Vaughan, G. (2020). Efficient big data model selection with applications to fraud detection. *International Journal of Forecasting*, *36*(3), 1116-1127. https://doi.org/10.1016/j.ijforecast.2018.03.002

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., ... & Van Mulbregt, P. (2020). SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature methods*, *17*(3), 261-272.

Vuttipittayamongkol, P., & Elyan, E. (2020). Neighbourhood-based undersampling approach for handling imbalanced and overlapped data. *Information Sciences*, *509*, 47-70.

Wilson, D. L. (1972). Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man, and Cybernetics, 3*,408-421. https://doi.org/10.1109/TSMC.1972.4309137

Waltman, L., & Van Eck, N. J. (2013). A smart local moving algorithm for large-scale modularity-based community detection. *The European physical journal B*, *86*(11), 471.

Wang, P., Fan, E., & Wang, P. (2021). Comparative analysis of image classification algorithms based on traditional machine learning and deep learning. *Pattern recognition letters*, *141*, 61-67. https://doi.org/10.1016/j.patrec.2020.07.042

Wang, L., Yu, Z., Ma, J., Chen, X., & Wu, C. (2025). A two-stage interpretable model to explain classifier in credit risk prediction. *Journal of Forecasting*. https://doi.org/10.1002/for.3288

Werbos, P. J. (2002). Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, *78*(10), 1550-1560.

Wolpert, D. H., & Macready, W. G. (2002). No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, *1*(1), 67-82. https://doi.org/10.1109/4235.585893

Wolpert, D. H. (2002). The supervised learning no-free-lunch theorems. *Soft computing and industry: Recent applications*, 25-42. https://doi.org/10.1007/978-1-4471-0123-9_3

Wong, W. C., Ahmad, A. H., Mohamed-Arshad, S. B., Nordin, S., & Adzis, A. A. (2022). Environmental, social and governance performance: continuous improvement matters. *Malaysian Journal of Economic Studies*, *59*(1), 49-69.

Woźniak, M., Wieczorek, M., & Siłka, J. (2023). BiLSTM deep neural network model for imbalanced medical data of IoT systems. *Future Generation Computer Systems*, *141*, 489-499.

Wu, S., Liu, Y., Zou, Z., & Weng, T. H. (2022). S_I_LSTM: stock price prediction based on multiple data sources and sentiment analysis. *Connection Science*, *34*(1), 44-62. https://doi.org/10.1080/09540091.2021.1940101

Xiaopeng, Y. A. N. G. (2024). Prediction of Financial Markets Utilizing an Innovatively Optimized Hybrid Model: A Case Study of the Hang Seng Index. *International Journal of Advanced Computer Science & Applications*, *15*(4). https://doi.org/10.14569/IJACSA.2024.0150415

Xu, T., & Qu, M. (2024). Novel embedding model predicting the credit card's default using neural network optimized by harmony search algorithm and vortex search algorithm. *Heliyon, 10*(9). https://doi.org/10.1016/j.heliyon.2024.e30134

Yang, X. S., Deb, S., (2009). Cuckoo search via Lévy flights. 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC), Coimbatore, India, 9-11 December. https://doi.org/10.1109/NABIC.2009.5393690

Yang, S., & Zhang, H. (2018). Comparison of several data mining methods in credit card default prediction. *Intelligent Information Management, 10*(5), 115–122. https://doi.org/10.4236/iim.2018.105010

Yang, L., & Shami, A. (2020). On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, *415*, 295-316.https://doi.org/10.1016/j.neucom.2020.07.061

Yeh, I.-C. (2009). *Default of credit card clients* [Data set]. UCI Machine Learning Repository. https://doi.org/10.24432/C55S3H

Yeh, I.-C., & Lien, C.-H. (2009). The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert Systems with Applications, 36*(2), 2473–2480. https://doi.org/10.1016/j.eswa.2007.12.020

Yilmaz, E. A., Balcisoy, S., & Bozkaya, B. (2023). A link prediction-based recommendation system using transactional data. *Scientific Reports, 13*(1), 6905. https://doi.org/10.1038/s41598-023-34055-5

Yu, P., & Yan, X. (2020). Stock price prediction based on deep neural networks. *Neural Computing and Applications*, *32*(6), 1609-1628. https://doi.org/10.1007/s00521-019-04212-x

Yun, S. Y., & Cho, N. W. (2025). A machine learning-based detection for parameter tampering vulnerabilities in web applications using BERT embeddings. *Symmetry, 17*(7), 985. https://doi.org/10.3390/sym17070985

Zhang, R., & Qiu, Z. (2020). Optimizing hyper-parameters of neural networks with swarm intelligence: A novel framework for credit scoring. *PLoS One, 15*(6), e0234254. https://doi.org/10.1371/journal.pone.0234254

Zhang, W., Yang, D., Zhang, S., Ablanedo-Rosas, J. H., Wu, X., & Lou, Y. (2021a). A novel multi-stage ensemble model with enhanced outlier adaptation for credit scoring. *Expert Systems with Applications, 165*, 113872. https://doi.org/10.1016/j.eswa.2020.113872

Zhang, W., Yang, D., & Zhang, S. (2021b). A new hybrid ensemble model with voting-based outlier detection and balanced sampling for credit scoring. *Expert Systems with Applications, 174*, 114744. https://doi.org/10.1016/j.eswa.2021.114744

Zhou, J., Wu, Z., Wang, Q., & Yu, Z. (2022). Fault diagnosis method of smart meters based on DBN-CapsNet. *Electronics*, *11*(10), 1603.

Zhou, Y., Shen, L., & Ballester, L. (2023). A two-stage credit scoring model based on random forest: Evidence from Chinese small firms. *International Review of Financial Analysis, 89*, 102755. https://doi.org/10.1016/j.irfa.2023.102755

Zhang, R., Li, I., Ding, Z., & Zhu, T. (2025). IWSL model: A novel credit scoring model with interpretable features for consumer credit scenarios. *Journal of Forecasting*. https://doi.org/10.1002/for.70004