Published by
NIGERIAN SOCIETY OF PHYSICAL SCIENCES
Available online @ https://journal.nsps.org.ng/index.php/jnsps

J. Nig. Soc. Phys. Sci. 6 (2024) 2039

Journal of the Nigerian Society of Physical Sciences

Novel way to predict stock movements using multiple models and comprehensive analysis: leveraging voting meta-ensemble techniques

Akila Dabara Kayit[a,b,*], Mohd Tahir Ismail[a]

[a] School of Mathematical Sciences, Universiti Sains Malaysia, 11800 Pulau Pinang 04-653 2624 Penang, Malaysia
[b] Universal Basic Education Commission, (UBEC) Gwani Street, Off IBB Way, Wuse Zone 4, Abuja, Nigeria

Abstract

The research introduces a method for anticipating stock market patterns by combining machine learning techniques with analysis methods. Multiple machine learning algorithms were integrated to address the limitations of stock market forecasting models. Using web scraping techniques, data were gathered from the S&P500 index over seven years, from September 5, 2016, to August 5, 2023. Companies like Microsoft Corporation (MSFT), Amazon.com Inc. (AMZN), JPMorgan Chase & Co (JPM), and Tesla, Inc. (TSLA) were selected based on their inclusion in the S&P 500 index. LR, RF, SVC, ADAB, and XGBC algorithms were applied as models by utilising optimisation using grid search and single algorithm approaches. Voting methods were employed to combine predictions from these models. The study employed rigorous statistical analyses, including the Kruskal-Wallis test to assess overall differences, followed by Pairwise Dunn's Test with Bonferroni Correction for detailed algorithm comparisons. Additionally, Bootstrapping was utilised to calculate Confidence Intervals (CI) for robust estimation of algorithm performance. The methodology covered data collection, preprocessing, model training, and performance assessment. The outcomes indicate that the proposed approach accurately forecasts stock trends precisely and dependably. This study contributes to refining stock market prediction methodologies by introducing a strategy that enhances prediction accuracy while offering investors and financial professionals insights. Furthermore, assessing algorithm performance across metrics and companies highlights the versatility and effectiveness of machine-learning approaches in the fields.

DOI:10.46481/jnsps.2024.2039

Keywords: Stock prediction, Machine learning, Voting, Meta-ensemble, Predictive modeling

Article History :
Received: 10 March 2024
Received in revised form: 08 June 2024
Accepted for publication: 28 June 2024
Published: 27 July 2024

Communicated by: B. J. Falaye

1. Introduction

In forecasting trends within the stock market, conventional models face challenges in grasping the evolving patterns inherent in financial time series data [1, 2]. With techniques like linear regression, ARIMA and neural networks, these models fail to adapt to the dynamic nature of the market, resulting in less-than-ideal forecasts [3]. This constraint underscores the significance of employing a modelling methodology that integrates various sources of information and possesses the capability to adapt to changing market dynamics. Despite advancements in learning techniques, there remains a gap in developing hybrid models that adeptly merge base learners and meta-

*Corresponding author: Tel.: +234-816-995-3938
Email address: akiladabara@student.usm.my (Akila Dabara Kayit)

ensemble strategies for forecasting stock market direction [4]. Current approaches often employ simplistic combination strategies that overlook the intricacies of market dynamics. Hence, there is a demand for an ensemble model capable of surmounting these challenges and delivering superior accuracy and reliability in predictions [5]. The proliferation of data availability and advancements in machine learning techniques have led to a surge in the creation of models to forecast stock market trends. Nonetheless, many existing models encounter difficulties in effectively encapsulating the complexities inherent in the stock market dynamics [6]. The absence of research in this area highlights the importance of adopting a model that integrates voting ensemble techniques to improve the accuracy and reliability of forecasts regarding stock market trends [7, 8].

Furthermore, grappling with data uncertainty and noise in the modelling realm poses considerable challenges. Given the susceptibility of the stock market to inefficiencies, unforeseen events, and external forces, this can precipitate fluctuations and intricacies, hindering models' ability to capture and forecast dynamics precisely. Thus, there's an augmented necessity for an approach to effectively aggregate model predictions and harness their intelligence to deliver precise and dependable forecasts [9, 10]. Another crucial element in forecasting stock market trends is the clarity and comprehensibility of existing models. Many machine learning algorithms are "black boxes," meaning that the user, in this case, an investor or an analyst, can not understand precisely why the algorithm made the trade recommendation decision that it did. This lack of transparency makes these models challenging to trust or believe, making them useless for real-world investment decisions [11–13]. Given these research gaps, creating a model integrating voting techniques is a promising solution for enhancing stock market trend prediction. By amalgamating diverse models and consolidating their predictions, this innovative method can proficiently apprehend the intricate and nonlinear dynamics intrinsic to the stock market [14]. Moreover, this approach's clarity and comprehensibility can mitigate the constraints linked to models. This enables investors and analysts to make decisions grounded in comprehending the predictive model's functioning. In light of the prevailing research gap in predicting stock market trends, it is imperative to devise a model that leverages voting ensemble techniques. The proposed hybrid ensemble model seeks to surmount the shortcomings of current models in forecasting stock market trends. This model endeavours to bolster prediction accuracy and resilience in volatile market conditions by integrating base learners and utilising meta-learning and aggregation techniques. Its potential ramifications are substantial, as it can transform stock market prediction by furnishing investors and financial analysts with forecasts conducive to informed decision-making [15–17].

The paper fills this research void by presenting a model that harnesses voting ensemble techniques to forecast stock directions. This model integrates base learners and employs sophisticated meta-learning and aggregation mechanisms to enhance prediction accuracy and resilience in ever-changing market conditions. It's necessary because it can revolutionise stock market forecasting, offering investors and financial analysts forecasts that inform their decision-making processes. Furthermore, the paper seeks to offer helpful insights into forecasting stock up and down movements, gaining both researchers and industry practitioners.

The paper's organisation is as follows: Section 2 delves into current methods utilised in stock market prediction, emphasising ensemble learning and meta-ensemble approaches. Section 3 provides an in-depth account of the methodology implemented in our proposed model. Also, Section 4 summarises the setup and presents the outcomes, and conducts a comprehensive discussion of the findings. Finally, Section 5 concludes the paper and presents recommendations for upcoming research directions.

2. Literature review

In recent years, there has been a trend in stock market prediction techniques towards adopting ensemble learning and hybrid modelling approaches. Various studies have shown that ensemble methods like stacking, blending, and voting are tools for improving accuracy and reliability in stock market forecasting. These methods combine the strengths of various models, surpassing the traditional single-model approaches, aiming to enhance prediction accuracy and robustness through advanced machine-learning techniques and hybrid modelling approaches. Ensemble learning techniques, highlighted by Refs. [14, 18–20], have garnered significant attention in recent research. The study emphasises two key points: Firstly, ensemble methods offer a robust approach to improving predictive accuracy by combining multiple models. Secondly, the review underscores the versatility of ensemble techniques across various domains, showcasing their efficacy in addressing complex classification and regression tasks. Their comprehensive evaluation of ensemble methods highlighted the superiority of stacking and blending techniques in achieving better prediction accuracies than traditional methods like bagging and boosting. This emphasis on ensemble learning underscores the value of combining diverse predictive models for improved performance.

Additionally, hybrid modelling approaches have garnered attention in stock market prediction research, with studies like Ref. [21] proposing innovative models that integrate multiple techniques for enhanced prediction capabilities. They introduced a hybrid GA-XGBoost algorithm with enhanced feature engineering, showcasing improved prediction accuracy through optimal feature set selection. Similarly, Ref. [22] developed a hybrid volatility prediction model integrating GARCH models and LSTM neural networks, demonstrating its effectiveness in risk analysis for the Chinese financial market.

Moreover, advancements in deep learning techniques have spurred the development of hybrid models combining convolutional neural networks (CNNs), recurrent neural networks (RNNs), and other machine learning algorithms. Ref. [23] introduced novel hybrid models, including CNN-LSTM and GRU-CNN, for forecasting stock market indices, achieving superior prediction accuracy compared to traditional machine learning models. These hybrid approaches leverage the strengths of different models to capture complex patterns in stock market data, leading to more accurate predictions.

Furthermore, researchers have explored integrating metaheuristic optimisation algorithms with machine learning models to improve prediction performance. Studies like Ref. [24] proposed hybrid models combining neural networks with metaheuristic optimisation techniques, showcasing improved prediction accuracy for stock price volatility and multi-step prediction of stock market trends.

Additionally, Ref. [25] introduced a novel hybrid model for stock price forecasting integrating Encoder Forest and Informer, providing further evidence of the effectiveness of hybrid approaches in capturing the dynamics of stock market data. As research in this field continues to evolve, integrating diverse techniques and exploring novel hybrid models hold promise for advancing stock market prediction methodologies, ultimately providing valuable insights for investment decision-making in dynamic financial markets.

Moreover, Refs. [26–29] discussed the predictability of machine learning techniques for forecasting market index prices. They emphasise the importance of understanding the limitations of these methods, especially in the context of stock market prediction. This study provides insights into the challenges and constraints associated with ensemble learning approaches for stock market prediction. Furthermore, Ref. [30] presents a survey on stock market prediction using machine learning techniques, highlighting the recent developments and future directions in the field. They discuss the methodologies and applications of meta-ensemble techniques, shedding light on the potential of these approaches in stock market prediction.

Additionally, Ref. [15] introduced a fractional neuro-sequential ARFIMA-LSTM model for financial market forecasting, contributing to the advancement of predictive modelling in financial markets. However, there is a gap in the literature regarding applying ensemble and meta-ensemble techniques to this specific model. Ref. [31] presents a novel ensemble deep-learning model based on stock prices and news, demonstrating the potential of combining various data sources for stock prediction.

While existing literature provide insights into ensemble models and hybrid techniques for stock direction prediction, knowledge gaps remain. Future research could explore meta-ensemble techniques for stock price forecasting, leveraging the outputs of multiple base models. Additionally, developing ensemble models that integrate advanced optimisation algorithms like particle swarm optimisation (PSO) and genetic algorithms (GA) could enhance the model's adaptability to the dynamic nature of financial markets. Predicting stock market trends has garnered significant attention due to its implications for financial decision-making. Refs. [32–34] proposed a hybrid method demonstrating superior performance. Ref. [35] presented a hybrid intelligent prediction model for financial time series analysis. Refs. [36, 37] integrated various approaches for stock price forecasting, showing superior accuracy. Similarly, Ref. [38] proposed two-stage ensemble models with significant predictive performance. Ref. [39] introduced a CEEMDAN-LSTM model for stock index RV forecasting, outperforming single models. Ref. [40] combined HAR specification, ESN, and PSO for predicting realised volatilities, and Ref. [41] employed ensemble machine learning for stock pattern prediction, achieving over 60% accuracy. Ref. [42] compared ARIMA with hybrid models for S&P500 log returns forecasting, and Ref. [43] evaluated ensemble classifiers for stock returns prediction, identifying essential features affecting returns. Ref. [17] developed an ensemble prediction model for stock index forecasting based on investor sentiments, outperforming baseline methods. Finally, Ref. [44] proposed an ensemble voting model for solar irradiation forecasting, achieving superior performance compared to individual algorithms.

This research introduces a novel approach to predicting stock movements by leveraging voting meta-ensemble techniques, distinguishing it from existing methods in several key aspects. Firstly, while previous studies have explored ensemble learning and hybrid modelling approaches, this method uniquely combines multiple machine learning algorithms and comprehensive analysis methods, including data preprocessing, feature engineering, and model evaluation [45, 46].

Compared to past studies, such as Ref. [14], which emphasised the importance of ensemble methods in improving predictive accuracy by combining multiple models, this approach extends beyond traditional ensemble techniques by utilising voting meta-ensemble methods. While Nti et al. provided a comprehensive review of ensemble learning techniques, this research demonstrates the practical application of these techniques in predicting stock market trends. The study builds upon the findings of Refs. [21, 22], who proposed innovative hybrid models for stock market prediction. Unlike these studies, which focused on specific hybrid algorithms like GA-XGBoost and GARCH-LSTM, the approach in this research incorporates a broader range of machine learning algorithms and analysis techniques to achieve superior prediction accuracy and reliability.

Moreover, this research addresses the limitations highlighted by Refs. [26–29] regarding the predictability of machine learning techniques for stock market forecasting. By employing a voting meta-ensemble approach, this paper aims to overcome these limitations and provide investors and financial professionals with more reliable predictions of stock movements. The contribution lies in developing and applying a novel method that combines voting meta-ensemble techniques with comprehensive analysis to predict stock movements accurately. By explicitly comparing this approach with existing methods outlined in the literature review, highlight the research's unique features and improvements in stock market prediction.

The literature review highlights how learning and hybrid modelling techniques improve stock market prediction methods. Although progress has been made, research is still needed to explore these approaches' potential fully. This includes integrating techniques with advanced optimisation algorithms and creating hybrid models customised to market conditions and data characteristics. Furthermore, future studies could address the challenges related to the interpretability and scalability of hybrid models, thus enhancing their usefulness in real-world financial applications.

3. Model theorems and procedures

The description outlines code snippets related to machine learning, data preparation, cross-validation, model assessment, and ensemble modelling. The authors explain these processes in this session, including the experimental setup, materials used, and methodologies employed. The powerful capabilities of the Pandas library were relied upon for data manipulation, and it is known for its effectiveness in analysing and manipulating datasets. Additionally, the comprehensive machine learning functionalities of Scikit-learn were utilised for classification tasks. Another machine learning tool used was Keras, which is employed as a high-level neural network API for building and training a deep learning model.

TensorFlow, a freely available machine learning framework, was utilised to build and train machine learning models, including deep learning structures. LightGBM and XGBoost, both optimised for speed and efficiency, were selected as preferred gradient-boosting frameworks for rapid model training and high-performance prediction tasks. Matplotlib, a widely used plotting library, also created visualisations for data analysis and model performance evaluation in Python.

It was divided into three sets to efficiently examine the data: training, validation, and testing data. This separation was made possible by utilising a function provided by the sci-kit-learn library [47]. Additionally, the MinMaxScaler was applied to scale the features and ensure they were within a range.

Next, attention was given to feature engineering, involving the creation of lagged versions of columns in the stock market data to capture relationships between variables. Moving on to model training and evaluation, machine learning algorithms were utilised for classification tasks. These algorithms included Logistic Regression, Random Forest, Decision Tree, Support Vector Classifier (SVC), XGBoost, and AdaBoost. To assess the models' performance, metrics such as accuracy, precision, recall, F1 score, and sensitivity were considered [48]. Validation techniques were employed to evaluate how well the models performed on data.

In this research, the ensemble selected machine learning models by carefully considering their strengths and how they complement each other to improve predictive accuracy. Each model was chosen to address weaknesses inherent in single-model approaches while leveraging its respective advantages. The ensemble's selection of machine learning models was based on carefully considering their strengths and how they complement each other to improve predictive accuracy. Each model was chosen to address weaknesses inherent in single-model approaches while leveraging its respective advantages.

Logistic regression (LR): LR was included in the ensemble due to its simplicity, interpretability, and ability to model linear relationships. While LR may not capture complex nonlinear patterns in the data, its inclusion provides a baseline model for comparison and ensures transparency in model interpretation [49].

Random forest (RF): RF was chosen for its robustness against overfitting, ability to handle high-dimensional data, and capability to capture nonlinear relationships through ensemble learning. By aggregating multiple decision trees, RF mitigates the risk of bias and variance associated with individual trees, thereby enhancing the ensemble's stability and generalisation performance [50, 51].

Support vector classifier (SVC): SVC was selected for its effectiveness in handling nonlinear decision boundaries and robustness to outliers. Despite its computational complexity, SVC excels in capturing complex patterns in the data and has the potential to improve the ensemble's predictive accuracy, especially in scenarios with non-linear separability [52, 53].

AdaBoost (ADAB): ADAB was included in the ensemble for its ability to focus on difficult-to-classify instances and adaptively adjust the model's weights during training. By sequentially training weak learners and assigning higher weights to misclassified samples, ADAB enhances the ensemble's performance, particularly in cases where specific patterns are challenging to capture [54, 55].

XGBoost (XGBC): XGBC was incorporated for its scalability, speed, and regularisation techniques, which mitigate overfitting and improve model generalisation. As a gradient boosting framework, XGBC effectively combines weak learners to create a robust ensemble model, resulting in superior predictive performance compared to individual models [56, 57].

The rationale behind selecting these specific models for the ensemble lies in their complementary strengths and collective ability to address the limitations of single-model approaches. LR provides transparency and simplicity, while RF, SVC, ADAB, and XGBC offer flexibility, robustness, and improved predictive accuracy through ensemble learning and regularisation techniques. By combining these diverse models, our ensemble approach capitalises on each model's strengths while mitigating their weaknesses, ultimately resulting in more reliable predictions of stock movements.

The research adds value by providing a detailed rationale for selecting specific models and highlighting how they complement each other. It enhances understanding of the ensemble approach and its potential for improving predictive accuracy in stock market forecasting. Lastly, modelling techniques were explored by combining individual machine learning models through VotingClassifier methods, leveraging models' strengths for improved accuracy and robustness.

Regarding data collection and experimental procedures, the code snippets in this project were based on stock market data obtained from databases or historical market data APIs—the experimental procedures involved preprocessing the data using engineering features to ensure analysis. Afterwards, algorithms were chosen for machine learning, their hyperparameters adjusted, and optimisation carried out using methods such as cross-validation and grid search. Subsequently, the models were trained, their performance evaluated, and modelling techniques applied to improve accuracy and dependability.

The Kruskal-Wallis test was applied to assess whether there were any significant differences among multiple independent groups. This non-parametric test is suitable for comparing three or more groups when the assumptions of normality and homogeneity of variances are violated [58]. Following the Kruskal-Wallis test, pairwise comparisons were conducted using Dunn's test with Bonferroni correction to identify specific group differences. This post hoc test is robust against non-normality and unequal variances, providing adjusted p-values for each pairwise comparison [59].

Bootstrap confidence intervals were utilised to estimate the uncertainty associated with sample statistics. This resampling technique involves repeatedly sampling with replacement from the original dataset to construct the distribution of the statistic of interest, enabling the calculation of confidence intervals without assuming a specific data distribution [60].

3.1. Data

Data was gathered from the S&P500 index using web scraping techniques for this investigation. The data spans seven years, from September 5, 2016, to August 5 2023. Our primary focus is analysing a group of companies representing sectors: Information Technology. This includes Microsoft Corporation (MSFT) and Amazon.com Inc. (AMZN). Financials: We'll be looking at JPMorgan Chase & Co (JPM). Consumer Discretionary: Tesla, Inc. (TSLA) falls under this category. These companies were specifically selected based on their inclusion in the S&P 500 index as of June 20 2023. In this study, the authors carefully examined the collected datasets to extract insights for the analysis. Table 1 shows data preprocessing and feature engineering indicators, for the technical data used [61, 62].

3.2. Models theorems

This paper explores the world of model theorems and their crucial role in predicting stock market trends. These theorems are a foundation for our models, providing a mathematical framework that helps us understand and analyse the stock market dynamics. They offer an approach to modelling financial phenomena, enabling us to interpret real-world market data and make practical predictions. By leveraging the potential of model theorems, our goal is to improve the accuracy and dependability of our models. This will empower investors and financial analysts to make informed decisions amidst the evolving landscape of the stock market.

3.2.1. Logistic regression

Definition. Logistic regression is a supervised machine learning algorithm in which the variable Y is binary; it is used for problems related to classification. It is which class it belongs to predicted by the probability [63, 64].

Key concepts

Sigmoid function: Logistic Regression employs the sigmoid (logistic) function to compress the output between 0 and 1, effectively representing probabilities. Mathematically, the function that is known as the sigmoid function is:

$$\sigma(z) = \frac{1}{1 + e^{-z}}. \tag{1}$$

Decision boundary: The logistic regression decision boundary is defined at a particular probability cutoff point, 0.5 by default. This threshold decides whether an instance is classified into one class or another according to whether the predicted probability is larger than the threshold.

Cost function (Log loss): The logistic regression model is trained by minimising the log loss (cross-entropy) cost function. No other change is needed here. Log loss measures the loss of the model of a single instance is given by

$$-\left(y \log(\hat{p}) + (1 - y) \log(1 - \hat{p})\right), \tag{2}$$

where y represents the true class (0 or 1), and $\hat{p}$ is the predicted probability.

Logistic regression in stock prediction

Features: The stock prediction problem can be modelled with logistic regression by extracting relevant features from the historical stock prices dataset.

Binary classification: The model can split stocks into two classes (buy 1 or not buy 0) using known patterns.

Probability estimation: Logistic regression gives probability estimates, which are more subtle cases of predicted outcomes. Threshold Tuning: The decision threshold can be tailored to achieve the required balance between precision and recall. Data preprocessing, handling the imbalances and the evaluation metric of the model are very important.

3.2.2. Support vector machine

Definition Support Vector Machine (SVM) is a supervised machine learning algorithm for classification and regression tasks. It aims to find a hyperplane that best separates the data into different classes or predicts a continuous outcome [65, 66].

Key concepts

Hyperplane. SVM searches for a hyperplane that maximally separates classes in the feature space. In a two-dimensional space, the hyperplane is a line; it becomes a hyperplane in higher dimensions.

Support vectors. Support vectors are data points closest to the decision boundary (hyperplane). They play a crucial role in defining the optimal hyperplane.

Margin. The margin is the distance between the hyperplane and the nearest data point from either class. SVM aims to maximise this margin.

Kernel trick. SVM can handle non-linear relationships between features using kernel functions (e.g., polynomial, radial basis function) to map the data into a higher-dimensional space.

Formulas

Linear SVM For a linearly separable case, the decision function is represented as:

$$f(x) = w \cdot x + b. \tag{3}$$

The decision boundary is $f(x) = 0$.

Soft margin SVM. Introduces a penalty term for misclassification, allowing some points to fall on the wrong side of the hyperplane:

$$\min_{w,b,\zeta} \frac{1}{2}||w||^2 + C \sum_{i=1}^{N} \zeta_i, \tag{4}$$

subject to: $y_i(w \cdot x_i + b) \geq 1 - \zeta_i$ and $\zeta_i \geq 0$.

Table 1: Data preprocessing and feature engineering indicators.

Indicator	Formula
Exponential Moving Average (EMA)	$EMA = \frac{\text{Close}-EMA_{\text{prev}}}{N+1} \times 2 + EMA_{\text{prev}}$ where N is the number of periods.
Exponential Volume Weighted Moving Average (EVWMA)	$EVWMA = \frac{\sum_{i=1}^{N}(\text{Price}_i \times \text{Volume}_i)}{\sum_{i=1}^{N} Volume_i}$ where N is the number of periods.
Stochastic Oscillator (%K)	$\%\text{K} = \frac{\text{Close}-\text{Low}_{14}}{\text{High}_{14}-\text{Low}_{14}}$ where Low_{14} and $High_{14}$ are the lowest and highest prices over the last 14 periods.
Stochastic Oscillator (%D)	$\%\text{D} = \frac{\sum_{i=1}^{3} \%K_i}{3}$ where %K is the Stochastic %K value.
Absolute Range (AR)	$AR = \frac{(\text{Number of periods}-\text{Periods since highest high})}{\text{Number of periods}} \times 100$
Buying Range (BR)	$BR = \frac{(\text{Number of periods}-\text{Periods since lowest low})}{\text{Number of periods}} \times 100$
Aroon Down (AROOND)	$AROOND$ = Aroon Up − Aroon Down
Aroon Up (AROONU)	$AROONU$ = Aroon Down − Aroon Up
Aroon Oscillator (AROONOSC)	$AROONOSC = AROONU - AROOND$
Rate of Change (ROC)	$ROC = \frac{\text{Close}_{10}-\text{Close}_0}{\text{Close}_0} \times 100$ Close_{10} Close after ten periods, and Close_0 is the initial closing price.

Kernel SVM: For non-linear relationships, the decision function becomes:

$$f(x) = \sum_{i=1}^{N} \alpha_i y_i K(x, x_i) + b, \tag{5}$$

where $K(x, x_i)$ is the kernel function, and α_i are the Lagrange multipliers.

SVM in stock prediction SVM can be applied to stock prediction using relevant features (indicators) extracted from stock price data. Its capabilities are versatile, as it can classify stocks (e.g., buy, sell, hold) based on historical data. Additionally, SVM regression can be employed to predict stock prices as a continuous variable. The effectiveness of SVM is influenced by the careful selection of the kernel, which depends on the characteristics of the data and whether they exhibit linear or non-linear relationships. Data processing, tuning hyperparameters, and handling imbalances in the dataset to use SVM in stock prediction effectively.

3.2.3. Random forest

Definition. Random Forest is an ensemble learning algorithm that operates by constructing a multitude of decision trees during training and outputs the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees [67, 68].

Key concepts

Ensemble learning: Random Forest employs ensemble learning by combining multiple decision trees to enhance predictive performance and control overfitting.

Decision trees: In the context of Random Forest, decision trees are built using a random subset of features for each tree, and their predictions are averaged to form a more robust model.

Bootstrapping: Each tree in the Random Forest is constructed using a bootstrap sample of the dataset, where some instances may be repeated, contributing to the diversity of the individual trees.

Feature randomness: Randomness is introduced in the feature selection process; at each node of a decision tree, a random subset of features is considered for splitting, further enhancing the diversity of the ensemble.

Voting: In the classification task, the final prediction of Random Forest is determined by a majority vote from all trees, while for regression, it's the average of forecasts, leading to a robust and well-generalized model.

Formulas

Bootstrap sampling Let D be the original dataset with N instances. Random Forest creates B bootstrap samples D_b, each of size N, by sampling with replacement:

$$D_b = \{(x_i, y_i)\}_{i=1}^{N}. \tag{6}$$

Decision tree training A single decision tree is trained using a random subset of features at each node. For classification, the prediction is determined by majority vote; for regression, it's the average:

$$\hat{y}_{\text{tree}} = \text{Mode}(\{y_i\}_{i \in \text{leaves}}). \tag{7}$$

Random Forest Prediction Aggregating predictions from all trees obtain the Random Forest prediction:

$$\hat{y}_{\text{RF}} = \frac{1}{B} \sum_{b=1}^{B} \hat{y}_{\text{tree}_b}. \tag{8}$$

Random forest in stock prediction

Feature importance: Random Forest provides a valuable measure of feature importance, aiding in identifying relevant features crucial for effective stock prediction.

Robustness: The ensemble nature of Random Forest contributes to increased robustness, reducing the risk of overfitting and enhancing the overall stability of the model.

Non-Linearity handling: Random Forest is adept at capturing non-linear relationships within stock data. This flexibility in modelling makes it well-suited for handling complex patterns and relationships.

Parameter tuning: Effective parameter tuning is crucial for optimising Random Forest's performance. Parameters such as tree number and maximum depth significantly influence the model's effectiveness and generalisation capabilities. It's essential to preprocess data and fine-tune parameters for optimal performance.

3.2.4. Recurrent neural network

Definition. A recurrent neural network (RNN) is a type of neural network architecture designed to handle sequential data by incorporating feedback loops. It maintains an internal state to process sequences of inputs, making it suitable for tasks such as time series prediction and natural language processing [69, 70].

Key concepts

Temporal dependencies: RNNs capture temporal dependencies in sequential data by maintaining an internal state that evolves steps, allowing them to model sequences effectively.

Recurrent connections: RNNs have recurrent connections that enable information to persist across time steps, allowing the network to remember past details while processing current inputs.

Vanishing gradient problem: Training RNNs can be challenging due to the vanishing gradient problem, where gradients diminish as they propagate back through time, leading to difficulties in learning long-range dependencies.

Gating mechanisms: To address the vanishing gradient problem, variants of RNNs, such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), incorporate gating mechanisms that control the flow of information, allowing them to learn and retain long-term dependencies more effectively.

Formulas

Recurrent step At each time step t, the output y_t of an RNN is computed based on the current input x_t and the previous hidden state h_{t-1}:

$$h_t = \sigma(W_{hx}x_t + W_{hh}h_{t-1} + b_h), \tag{9}$$

$$y_t = \text{softmax}(W_{yh}h_t + b_y), \tag{10}$$

where W_{hx} and W_{hh} are weight matrices, b_h is the bias vector for the hidden layer, σ is the activation function (e.g., tanh or ReLU), softmax is the softmax function, and W_{yh} and b_y are weight matrix and bias vector for the output layer, respectively.

Backpropagation Through Time (BPTT): Training RNNs involves backpropagating gradients through time using the chain rule. The gradients accumulate over time and are used to update the network parameters via gradient descent.

RNN in stock prediction

Temporal modeling: RNNs excel at capturing temporal dependencies in sequential data, making them well-suited for predicting stock prices based on historical price data.

Sequential prediction: RNNs can predict future stock prices by learning patterns and trends from past price sequences, enabling investors to make informed decisions.

Feature learning: RNNs automatically extract relevant features from raw price data, eliminating the need for manual feature engineering and enhancing prediction accuracy.

Long-term dependencies: Variants of RNNs, such as LSTM and GRU, address the vanishing gradient problem, allowing them to learn long-term dependencies in stock data and make more accurate predictions.

Preprocessing data, tuning hyperparameters, and monitoring model performance are essential for effectively using RNNs in stock prediction.

3.2.5. AdaBoost

Definition: AdaBoost is an ensemble learning algorithm that combines multiple weak classifiers to create a strong classifier. It sequentially trains weak classifiers on subsets of the data, giving more weight to instances misclassified by previous classifiers. It adjusts the weights of training instances at each iteration to focus on the most difficult cases, ultimately producing a strong classifier [71, 72].

Mathematical formulation

Let T be the number of iterations in ADABoost, and $h_t(x)$ be the weak classifier at iteration t. At each iteration, AdaBoost assigns a weight α_t to the weak classifier $h_t(x)$, based on its performance:

$$\epsilon_t = \sum_{i=1}^{N} w_t^{(i)} \cdot \mathbb{I}(h_t(x^{(i)}) \neq y^{(i)}), \tag{11}$$

where ϵ_t is the weighted error of the weak classifier $h_t(x)$, $w_t^{(i)}$ is the weight assigned to training instance i at iteration t, $\mathbb{I}$ is the indicator function (equals one if the condition is proper, 0 otherwise), $x^{(i)}$ is the i-th training instance, and $y^{(i)}$ is its corresponding accurate label.

The weight α_t of the weak classifier $h_t(x)$ is computed as follows:

$$\alpha_t = \frac{1}{2}\ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right). \tag{12}$$

The weights of the training instances are updated based on their misclassification by the weak classifier:

$$w_{t+1}^{(i)} = w_t^{(i)} \cdot \exp\left(-\alpha_t \cdot y^{(i)} \cdot h_t(x^{(i)})\right), \tag{13}$$

where $w_{t+1}^{(i)}$ is the updated weight of training instance i at iteration $t+1$, and exp denotes the exponential function.

The final strong classifier $H(x)$ is obtained by combining the weak classifiers weighted by α_t:

$$H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t \cdot h_t(x)\right). \tag{14}$$

Relation to stock classification

AdaBoost can be applied to stock classification tasks using relevant features extracted from historical stock data. It constructs a robust classifier by sequentially training weak classifiers (e.g., decision trees) on subsets of the data, with each weak classifier focusing on different aspects of the data. In stock classification, weak classifiers may be trained to identify patterns or trends in historical stock prices, trading volumes, or technical indicators. ADABoost adapts to the complexity of the data and learns to combine the predictions of weak classifiers to make accurate classifications, such as predicting whether a stock will increase or decrease in value. By leveraging the strengths of multiple weak classifiers, AdaBoost enhances the classification performance and provides robust predictions for stock market analysis and decision-making.

3.2.6. XGBoost (Extreme gradient boosting)

Mathematical formulation:

Let T be the number of iterations in the XGBoost Classifier, and $h_t(x)$ be the weak learner at iteration t. The final predictive model $F(x)$ for classification is obtained by applying a softmax function to the sum of predictions of all weak learners:

$$F(x)_j = \frac{e^{(\sum_{t=1}^{T} \eta \cdot h_t(x))_j}}{\sum_{k=1}^{K} e^{(\sum_{t=1}^{T} \eta \cdot h_t(x))_k}}, \tag{15}$$

where K is the number of classes.

At each iteration, the XGBoost Classifier fits a weak learner $h_t(x)$ to the negative gradient of the loss function concerning the predicted probabilities from the previous model $F_{t-1}(x)$, where $F_{t-1}(x)$ is the prediction made by the model at iteration $t-1$. This is expressed as:

$$h_t(x) = \arg\min_h \sum_{i=1}^{N} L(y_i, F_{t-1}(x_i) + h(x_i)). \tag{16}$$

The final model is trained by minimising the cross-entropy loss function:

$$\mathcal{L} = -\sum_{i=1}^{N} \sum_{j=1}^{K} y_{ij} \log(F(x_i)_j), \tag{17}$$

where y_{ij} is the indicator function that equals 1 if the instance i belongs to class j, and 0 otherwise.

Relation to stock classification:

XGBoost Classifier can be applied to stock classification tasks using relevant features extracted from historical stock data. It learns to classify stocks into multiple categories (e.g., buy, sell, hold) by iteratively fitting weak learners to the negative gradients of the cross-entropy loss function. Weak learners may be decision trees trained on features such as historical stock prices, trading volumes, technical indicators, and macroeconomic factors. XGBoost Classifier adapts to the complexity of the data and learns to combine the predictions of weak learners to make accurate classifications. By optimising the cross-entropy loss function, XGBoost Classifier enhances the predictive performance and provides robust forecasts for stock market analysis and decision-making [73].

3.2.7. Hybrid ensemble

Mathematical formulation:

Let M be the number of base learners in the meta-ensemble, and let $h_i(x)$ denote the prediction of the i-th base learner, for instance, x. For the voting meta-ensemble, the final prediction $F(x)$ is obtained by a majority vote or by averaging the predictions of all base learners:

$$F(x) = \text{MajorityVote}(h_1(x), h_2(x), ..., h_M(x)), \tag{18}$$

$$F(x) = \frac{1}{M} \sum_{i=1}^{M} h_i(x). \tag{19}$$

For the relation to stock classification, each base learner $h_i(x)$ in the meta-ensemble is trained on historical stock data and extracts relevant features to make predictions. Support Vector Classifier (SVC), Logistic Regression (LR), AdaBoost Classifier (ADAC), and XGBoost Classifier (XGBC) are popular base learners used in stock classification tasks. By combining the predictions of multiple base learners using a voting meta-ensemble approach, the meta-ensemble model enhances the predictive performance and robustness of stock classification. This ensemble technique leverages the diverse learning capabilities of individual base learners and adapts to different characteristics and complexities of stock data, resulting in more accurate and reliable predictions [74]. Formulas: For the SVC + LR + ADAC meta-ensemble:

$$F(x) = \text{MajorityVote}(h_{\text{SVC}}(x), h_{\text{LR}}(x), h_{\text{ADAC}}(x)). \tag{20}$$

For the SVC + LR + XGBC meta-ensemble:

$$F(x) = \text{MajorityVote}(h_{\text{SVC}}(x), h_{\text{LR}}(x), h_{\text{XGBC}}(x)). \tag{21}$$

In equations (1) and (2), $F(x)$ represents the final prediction of the meta-ensemble, for instance, x, obtained by combining the predictions of base learners using a majority vote. $h_{\text{SVC}}(x)$, $h_{\text{LR}}(x)$, $h_{\text{ADAC}}(x)$, and $h_{\text{XGBC}}(x)$ denote the predictions of Support Vector Classifier, Logistic Regression, AdaBoost Classifier, and XGBoost Classifier, respectively, for instance x.

3.2.8. Evaluation metrics

Accuracy is determined by summing the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) together and then dividing the sum by the total number of true positives and true negatives.

Precision is calculated by dividing the number of true positives (TP) by the sum of true positives and false positives (FP).

Recall, also known as sensitivity or actual positive rate, is determined by dividing the number of true positives (TP) by the sum of true positives and false negatives (FN).

The F1 score is the harmonic mean of precision and recall, and it considers both precision and recall.

The area under the ROC curve, often referred to as AUC or ROC AUC, can be calculated by integrating a function that plots the positive rate against the false positive rate over a range from 0 to 1.

In these formulas:

- TP refers to the number of identified cases.
- TN represents the cases where the model correctly identifies something as negative.
- FP occurs when the model wrongly classifies something as positive.
- FN refers to cases where the model incorrectly identifies something as negative.

The actual positive rate (TPR), or recall, is calculated by dividing the number of positives (TP) by the sum of positives and false negatives (TP+FN). Similarly, the false positive rate (FPR) is calculated by dividing false positives (FP) by the sum of false positives and true negatives (FP+TN). These metrics are commonly used to evaluate the performance of a classification model [75, 76].

i. Accuracy:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{22}$$

ii. Precision:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{23}$$

iii. Recall (Sensitivity or True Positive Rate):

$$\text{Recall} = \frac{TP}{TP + FN} \tag{24}$$

iv. F1 Score (Harmonic Mean of Precision and Recall):

$$\text{F1 Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{25}$$

v. Area Under the ROC Curve (AUC or ROC AUC):

$$\text{AUC} = \int_0^1 \text{TPR}(\text{FPR}^{-1}(t))\, dt \tag{26}$$

3.2.9. Statistical Test

i. Kruskal-Wallis test:

The Kruskal-Wallis test was conducted to compare the mean ranks of the groups. The test statistic (H) is calculated as follows [58]:

$$H = \frac{12}{N(N+1)} \left[\sum_{i=1}^{k} \frac{R_i^2}{n_i} - 3(N+1) \right], \tag{27}$$

where N is the total number of observations, k is the number of groups, n_i is the number of observations in the ith group, and R_i is the sum of ranks in the ith group.

Then, the computed test statistic was compared to the critical value from the χ^2 distribution with $k-1$ degrees of freedom to determine statistical significance.

A post hoc test, also known as a post hoc analysis or post hoc comparison, is conducted after an initial statistical test, such as an ANOVA or Kruskal-Wallis test, to determine which specific groups differ from each other. Post hoc tests are used when the initial test indicates a significant difference between groups but does not identify which specific groups are different. Joint post hoc tests include Tukey's HSD (Honestly Significant Difference), Bonferroni correction, Dunn's, Scheffe's, and others. These tests adjust for multiple comparisons to reduce the likelihood of Type I errors occurring when a true null hypothesis is incorrectly rejected.

Table 2: Average performance of algorithms by methods, companies and metrics.

Methods	Accuracy	Precision	Recall	F1 Score	Sensitivity	AUC
Hybrid	0.80	0.80	0.87	0.83	0.87	0.92
Optimasation	0.79	0.79	0.74	0.75	0.74	0.78
Single	0.71	0.71	0.67	0.67	0.67	0.70
Company	Accuracy	Precision	Recall	F1 Score	Sensitivity	AUC
Amazon	0.76	0.76	0.79	0.76	0.79	0.78
JPMorgan	0.75	0.75	0.71	0.71	0.71	0.77
Micrsofot	0.74	0.74	0.69	0.69	0.69	0.75
TESLA	0.78	0.78	0.71	0.74	0.71	0.76
Algorithms	Accuracy	Precision	Recall	F1 Score	Sensitivity	AUC
ADBA	0.68	0.68	0.61	0.63	0.61	0.73
LR	0.77	0.77	0.89	0.82	0.89	0.88
RF	0.67	0.67	0.60	0.63	0.60	0.72
RNN	1.00	1.00	0.50	0.67	0.50	0.53
SVC	0.71	0.71	0.78	0.73	0.78	0.81
SVC + LR + XGB	0.79	0.79	0.88	0.83	0.88	0.91
SVC + LR +ADAC	0.83	0.83	0.91	0.86	0.91	0.94
XGBC	0.68	0.68	0.69	0.68	0.69	0.76

ii. Pairwise Dunn's test with Bonferroni correction:

The Pairwise Dunn's test with Bonferroni correction was conducted to compare the mean ranks between pairs of groups. This test is a non-parametric method used to determine whether there are statistically significant differences between multiple groups [59]. The test statistic (Z) for each pairwise comparison is calculated using the following formula:

$$Z = \frac{\bar{r}_i - \bar{r}_j}{\sqrt{\frac{n(n+1)}{12N}\left(\frac{1}{n_i} + \frac{1}{n_j}\right)}}, \tag{28}$$

where $\bar{r}_i$ and $\bar{r}_j$ are the mean ranks of groups i and j respectively, n_i and n_j are the sample sizes of groups i and j, and N is the total number of observations. The calculated test statistic is then compared to the critical value from the standard normal distribution to determine statistical significance. Finally, to account for multiple comparisons, the p-values obtained from the pairwise comparisons are adjusted using the Bonferroni correction method.

iii. Bootstrap confidence intervals:

Bootstrap confidence intervals are a resampling technique used to estimate the uncertainty associated with a sample statistic, such as the mean or median [60]. In bootstrap resampling, multiple samples are drawn with replacements from the original data set and the statistic of interest is calculated for each resampled data set. By repeating this process many times, a distribution of the statistic is obtained. The confidence interval is then constructed from this distribution by selecting appropriate percentiles, such as the 2.5th and 97.5th percentiles, for a 95%. Bootstrap confidence intervals are beneficial when the underlying distribution of the data is unknown or non-normal, as they make fewer assumptions about the data distribution compared to parametric methods.

Table 3: Results of the Kruskal-Wallis test.

Variable	Statistic	p-value
Algorithms	35.615	8.57×10^{-6}
Company	1.400	0.706

Table 4: Pairwise Dunn's test results.

Algorithms	ADBA	LR	RF	RNN	SVC	Hybrid A	Hybrid B	XGBC
ADBA	1.000	0.198	1.000	1.000	1.000	0.745	0.247	1.000
LR	0.198	1.000	0.325	0.000083	1.000	1.000	1.000	1.000
RF	1.000	0.325	1.000	0.888	1.000	1.000	0.367	1.000
RNN	1.000	0.000	0.888	1.000	0.029	0.004	0.001	0.009
SVC	1.000	1.000	1.000	0.029	1.000	1.000	1.000	1.000
Hybrid A	0.745	1.000	1.000	0.004	1.000	1.000	1.000	1.000
Hybrid B	0.247	1.000	0.367	0.001	1.000	1.000	1.000	1.000
XGBC	1.000	1.000	1.000	0.009	1.000	1.000	1.000	

Hybrid A: SVC + LR + XGB, Hybrid B: SVC + LR + ADAC

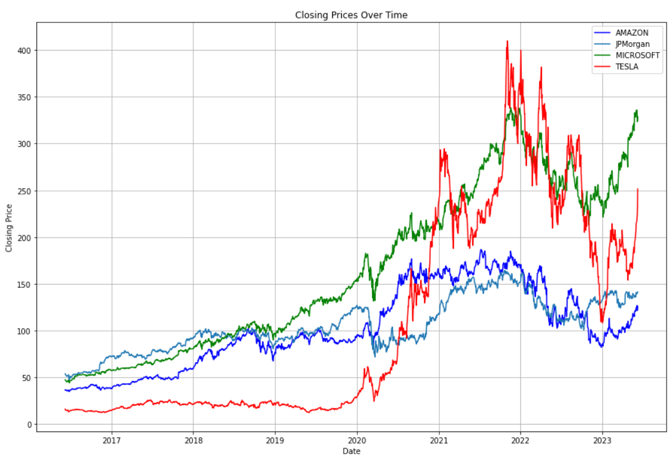

Figure 1: Closing price for the four companies.

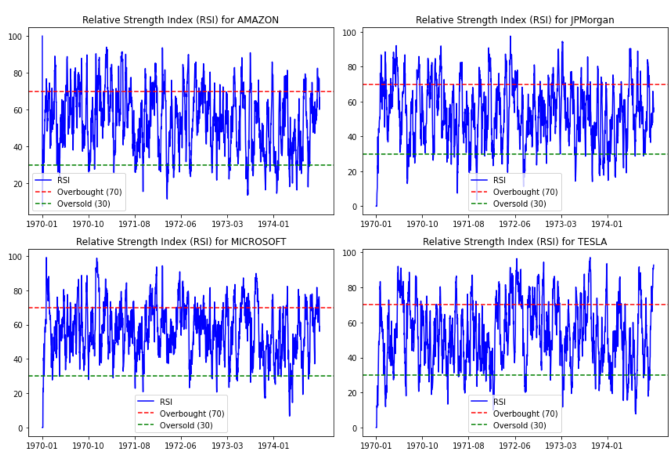

Figure 2: Relative strength index (RSI) for the four companies.

4. Results and Discussions

Table 2 summarises the data by methods, companies and algorithms. The graph in Figure 1 illustrates the closing prices of the four companies from 2017 to 2023. Figure 2 graph shows the Relative Strength Index (RSI) indicating Overbought and Oversold conditions for these companies. Additionally, Figure

Table 5: Confidence intervals for the methods, algorithms, and companies on the matrics.

Approach	Model	Company	Metric	Mean	95% CI
Hybrid	Hybrid A	AMAZON	Accuracy	0.79	(0.79, 0.79)
Hybrid	Hybrid A	AMAZON	Precision	0.79	(0.79, 0.79)
Hybrid	Hybrid A	AMAZON	Recall	0.92	(0.92, 0.92)
Hybrid	Hybrid A	AMAZON	F1 Score	0.85	(0.85, 0.85)
Hybrid	Hybrid A	AMAZON	Sensitivity	0.92	(0.92, 0.92)
Hybrid	Hybrid A	AMAZON	AUC	0.92	(0.92, 0.92)
Hybrid	Hybrid A	JMORGAN	Accuracy	0.74	(0.74, 0.74)
Hybrid	Hybrid A	JMORGAN	Precision	0.74	(0.74, 0.74)
Hybrid	Hybrid A	JMORGAN	Recall	0.9	(0.90, 0.90)
Hybrid	Hybrid A	JMORGAN	F1 Score	0.81	(0.81, 0.81)
Hybrid	Hybrid A	JMORGAN	Sensitivity	0.9	(0.90, 0.90)
Hybrid	Hybrid A	JMORGAN	AUC	0.9	(0.90, 0.90)
Hybrid	Hybrid A	MICROSOFT	Accuracy	0.83	(0.83, 0.83)
Hybrid	Hybrid A	MICROSOFT	Precision	0.83	(0.83, 0.83)
Hybrid	Hybrid A	MICROSOFT	Recall	0.77	(0.77, 0.77)
Hybrid	Hybrid A	MICROSOFT	F1 Score	0.8	(0.80, 0.80)
Hybrid	Hybrid A	MICROSOFT	Sensitivity	0.77	(0.77, 0.77)
Hybrid	Hybrid A	MICROSOFT	AUC	0.9	(0.90, 0.90)
Hybrid	Hybrid A	TESLA	Accuracy	0.8	(0.80, 0.80)
Hybrid	Hybrid A	TESLA	Precision	0.8	(0.80, 0.80)
Hybrid	Hybrid A	TESLA	Recall	0.81	(0.81, 0.81)
Hybrid	Hybrid A	TESLA	F1 Score	0.8	(0.80, 0.80)
Hybrid	Hybrid A	TESLA	Sensitivity	0.81	(0.81, 0.81)
Hybrid	Hybrid A	TESLA	AUC	0.89	(0.89, 0.89)
Hybrid	Hybrid B	AMAZON	Accuracy	0.83	(0.83, 0.83)
Hybrid	Hybrid B	AMAZON	Precision	0.83	(0.83, 0.83)
Hybrid	Hybrid B	AMAZON	Recall	0.92	(0.92, 0.92)
Hybrid	Hybrid B	AMAZON	F1 Score	0.87	(0.87, 0.87)
Hybrid	Hybrid B	AMAZON	Sensitivity	0.92	(0.92, 0.92)
Hybrid	Hybrid B	AMAZON	AUC	0.94	(0.94, 0.94)
Hybrid	Hybrid B	JMORGAN	Accuracy	0.74	(0.74, 0.74)
Hybrid	Hybrid B	JMORGAN	Precision	0.74	(0.74, 0.74)
Hybrid	Hybrid B	JMORGAN	Recall	0.93	(0.93, 0.93)
Hybrid	Hybrid B	JMORGAN	F1 Score	0.83	(0.83, 0.83)
Hybrid	Hybrid B	JMORGAN	Sensitivity	0.93	(0.93, 0.93)
Hybrid	Hybrid B	JMORGAN	AUC	0.92	(0.92, 0.92)
Hybrid	Hybrid B	MICROSOFT	Accuracy	0.81	(0.81, 0.81)
Hybrid	Hybrid B	MICROSOFT	Precision	0.81	(0.81, 0.81)
Hybrid	Hybrid B	MICROSOFT	Recall	0.88	(0.88, 0.88)
Hybrid	Hybrid B	MICROSOFT	F1 Score	0.84	(0.84, 0.84)
Hybrid	Hybrid B	MICROSOFT	Sensitivity	0.88	(0.88, 0.88)
Hybrid	Hybrid B	MICROSOFT	AUC	0.92	(0.92, 0.92)
Hybrid	Hybrid B	TESLA	Accuracy	0.85	(0.85, 0.85)
Hybrid	Hybrid B	TESLA	Precision	0.85	(0.85, 0.85)
Hybrid	Hybrid B	TESLA	Recall	0.86	(0.86, 0.86)
Hybrid	Hybrid B	TESLA	F1 Score	0.85	(0.85, 0.85)
Hybrid	Hybrid B	TESLA	Sensitivity	0.86	(0.86, 0.86)
Hybrid	Hybrid B	TESLA	AUC	0.93	(0.93, 0.93)

Hybrid A: SVC + LR + XGB, Hybrid B: SVC + LR + ADAC

3 showcases the correlations between closing prices, and Figure 4 depicts accuracy results in comparisons among methods, companies and algorithms. Lastly, Figure 5, 6, 7, 8, and 9 providing a box plot and bar graphs for comparisons of methods, algorithms and companies based on performance metrics. Tables 5, 6, 7, 8, 9, 10, and 11 are showing the Confidence Interval (CI) results which indicate a significant improvement in prediction accuracy when using the proposed model. The confidence interval for the algorithm's performance is the upper and lower bound. The data analysis revealed that the Hybrid method yielded superior results to the Optimization and Single methods. In particular, the Hybrid method achieved an accuracy of 0.799, precision of 0.799, recall of 0.874, F1 Score of

Table 6: Confidence intervals for the methods, algorithms, and companies on the matrics.

Approach	Model	Company	Metric	Mean	95% CI
Optz	ADBA	AMAZON	Accuracy	0.69	(0.69, 0.69)
Optz	ADBA	AMAZON	Precision	0.69	(0.69, 0.69)
Optz	ADBA	AMAZON	Recall	0.69	(0.69, 0.69)
Optz	ADBA	AMAZON	F1 Score	0.69	(0.69, 0.69)
Optz	ADBA	AMAZON	Sensitivity	0.69	(0.69, 0.69)
Optz	ADBA	AMAZON	AUC	0.74	(0.74, 0.74)
Optz	ADBA	JMORGAN	Accuracy	0.72	(0.72, 0.72)
Optz	ADBA	JMORGAN	Precision	0.72	(0.72, 0.72)
Optz	ADBA	JMORGAN	Recall	0.57	(0.57, 0.57)
Optz	ADBA	JMORGAN	F1 Score	0.63	(0.63, 0.63)
Optz	ADBA	JMORGAN	Sensitivity	0.57	(0.57, 0.57)
Optz	ADBA	JMORGAN	AUC	0.76	(0.76, 0.76)
Optz	ADBA	MICROSOFT	Accuracy	0.54	(0.54, 0.54)
Optz	ADBA	MICROSOFT	Precision	0.54	(0.54, 0.54)
Optz	ADBA	MICROSOFT	Recall	0.93	(0.93, 0.93)
Optz	ADBA	MICROSOFT	F1 Score	0.68	(0.68, 0.68)
Optz	ADBA	MICROSOFT	Sensitivity	0.93	(0.93, 0.93)
Optz	ADBA	MICROSOFT	AUC	0.68	(0.68, 0.68)
Optz	ADBA	TESLA	Accuracy	0.65	(0.65, 0.65)
Optz	ADBA	TESLA	Precision	0.65	(0.65, 0.65)
Optz	ADBA	TESLA	Recall	0.62	(0.62, 0.62)
Optz	ADBA	TESLA	F1 Score	0.63	(0.63, 0.63)
Optz	ADBA	TESLA	Sensitivity	0.62	(0.62, 0.62)
Optz	ADBA	TESLA	AUC	0.69	(0.69, 0.69)
Optz	LR	AMAZON	Accuracy	0.97	(0.97, 0.97)
Optz	LR	AMAZON	Precision	0.97	(0.97, 0.97)
Optz	LR	AMAZON	Recall	0.99	(0.99, 0.99)
Optz	LR	AMAZON	F1 Score	0.98	(0.98, 0.98)
Optz	LR	AMAZON	Sensitivity	0.99	(0.99, 0.99)
Optz	LR	AMAZON	AUC	1	(1.00, 1.00)
Optz	LR	JMORGAN	Accuracy	0.84	(0.84, 0.84)
Optz	LR	JMORGAN	Precision	0.84	(0.84, 0.84)
Optz	LR	JMORGAN	Recall	0.94	(0.94, 0.94)
Optz	LR	JMORGAN	F1 Score	0.89	(0.89, 0.89)
Optz	LR	JMORGAN	Sensitivity	0.94	(0.94, 0.94)
Optz	LR	JMORGAN	AUC	0.96	(0.96, 0.96)
Optz	LR	MICROSOFT	Accuracy	0.73	(0.73, 0.73)
Optz	LR	MICROSOFT	Precision	0.73	(0.73, 0.73)
Optz	LR	MICROSOFT	Recall	1	(1.00, 1.00)
Optz	LR	MICROSOFT	F1 Score	0.84	(0.84, 0.84)
Optz	LR	MICROSOFT	Sensitivity	1	(1.00, 1.00)
Optz	LR	MICROSOFT	AUC	0.95	(0.95, 0.95)
Optz	LR	TESLA	Accuracy	0.96	(0.96, 0.96)
Optz	LR	TESLA	Precision	0.96	(0.96, 0.96)
Optz	LR	TESLA	Recall	1	(1.00, 1.00)
Optz	LR	TESLA	F1 Score	0.98	(0.98, 0.98)
Optz	LR	TESLA	Sensitivity	1	(1.00, 1.00)
Optz	LR	TESLA	AUC	1	(1.00, 1.00)

Table 7: Confidence intervals for the methods, algorithms, and companies on the matrics.

Approach	Model	Company	Metric	Mean	95% CI
Optz	RF	AMAZON	Accuracy	0.66	(0.66, 0.66)
Optz	RF	AMAZON	Precision	0.66	(0.66, 0.66)
Optz	RF	AMAZON	Recall	0.81	(0.81, 0.81)
Optz	RF	AMAZON	F1 Score	0.73	(0.73, 0.73)
Optz	RF	AMAZON	Sensitivity	0.81	(0.81, 0.81)
Optz	RF	AMAZON	AUC	0.76	(0.76, 0.76)
Optz	RF	JMORGAN	Accuracy	0.69	(0.69, 0.69)
Optz	RF	JMORGAN	Precision	0.69	(0.69, 0.69)
Optz	RF	JMORGAN	Recall	0.58	(0.58, 0.58)
Optz	RF	JMORGAN	F1 Score	0.64	(0.64, 0.64)
Optz	RF	JMORGAN	Sensitivity	0.58	(0.58, 0.58)
Optz	RF	JMORGAN	AUC	0.75	(0.75, 0.75)
Optz	RF	MICROSOFT	Accuracy	0.7	(0.70, 0.70)
Optz	RF	MICROSOFT	Precision	0.7	(0.70, 0.70)
Optz	RF	MICROSOFT	Recall	0.54	(0.54, 0.54)
Optz	RF	MICROSOFT	F1 Score	0.61	(0.61, 0.61)
Optz	RF	MICROSOFT	Sensitivity	0.54	(0.54, 0.54)
Optz	RF	MICROSOFT	AUC	0.72	(0.72, 0.72)
Optz	RF	TESLA	Accuracy	0.68	(0.68, 0.68)
Optz	RF	TESLA	Precision	0.68	(0.68, 0.68)
Optz	RF	TESLA	Recall	0.65	(0.65, 0.65)
Optz	RF	TESLA	F1 Score	0.66	(0.66, 0.66)
Optz	RF	TESLA	Sensitivity	0.65	(0.65, 0.65)
Optz	RF	TESLA	AUC	0.73	(0.73, 0.73)
Optz	RNN	AMAZON	Accuracy	1	(1.00, 1.00)
Optz	RNN	AMAZON	Precision	1	(1.00, 1.00)
Optz	RNN	AMAZON	Recall	0.5	(0.50, 0.50)
Optz	RNN	AMAZON	F1 Score	0.66	(0.66, 0.66)
Optz	RNN	AMAZON	Sensitivity	0.5	(0.50, 0.50)
Optz	RNN	AMAZON	AUC	0.56	(0.56, 0.56)
Optz	RNN	JMORGAN	Accuracy	1	(1.00, 1.00)
Optz	RNN	JMORGAN	Precision	1	(1.00, 1.00)
Optz	RNN	JMORGAN	Recall	0.48	(0.48, 0.48)
Optz	RNN	JMORGAN	F1 Score	0.65	(0.65, 0.65)
Optz	RNN	JMORGAN	Sensitivity	0.48	(0.48, 0.48)
Optz	RNN	JMORGAN	AUC	0.51	(0.51, 0.51)
Optz	RNN	MICROSOFT	Accuracy	1	(1.00, 1.00)
Optz	RNN	MICROSOFT	Precision	1	(1.00, 1.00)
Optz	RNN	MICROSOFT	Recall	0.49	(0.49, 0.49)
Optz	RNN	MICROSOFT	F1 Score	0.66	(0.66, 0.66)
Optz	RNN	MICROSOFT	Sensitivity	0.49	(0.49, 0.49)
Optz	RNN	MICROSOFT	AUC	0.54	(0.54, 0.54)
Optz	RNN	TESLA	Accuracy	1	(1.00, 1.00)
Optz	RNN	TESLA	Precision	1	(1.00, 1.00)
Optz	RNN	TESLA	Recall	0.54	(0.54, 0.54)
Optz	RNN	TESLA	F1 Score	0.7	(0.70, 0.70)
Optz	RNN	TESLA	Sensitivity	0.54	(0.54, 0.54)
Optz	RNN	TESLA	AUC	0.51	(0.51, 0.51)

0.831, sensitivity of 0.874, and AUC of 0.915.

These results are similar to those of other studies. For example, Ref. [23] found that the voting-ensemble method, like the Hybrid approach, performed better than optimisation techniques regarding accuracy and overall classification performance. Similarly, Ref. [77] also highlighted how ensemble methods like the Hybrid one have recall scores by capturing several actual positive instances. While the optimisation and single methods didn't perform as well as the hybrid method, their results still align with what has been seen in previous research. The accuracy and precision scores of the Optimization method are comparable to those seen in Ref. [78] on optimisation techniques for classification tasks. Likewise, the performance metrics of the method resemble those found in Ref. [79] analysis on single algorithm approaches in machine learning. The results support previous studies; ensemble techniques such as the Hybrid method often produce results in categorisation assignments in terms of precision, recall, and overall model resilience. Nevertheless, it's crucial to consider the data set's features and the assignment's goals when choosing the suitable approach for a particular use case. Therefore, the highest to lowest performance ranking would be Hybrid, Optimization, and Single.

In situations where precision and recall are both important

Table 8: Confidence intervals for the methods, algorithms, and companies on the matrics.

Approach	Model	Company	Metric	Mean	95% CI
Optz	SVC	AMAZON	Accuracy	0.84	(0.84, 0.84)
Optz	SVC	AMAZON	Precision	0.84	(0.84, 0.84)
Optz	SVC	AMAZON	Recall	0.95	(0.95, 0.95)
Optz	SVC	AMAZON	F1 Score	0.89	(0.89, 0.89)
Optz	SVC	AMAZON	Sensitivity	0.95	(0.95, 0.95)
Optz	SVC	AMAZON	AUC	0.96	(0.96, 0.96)
Optz	SVC	JMORGAN	Accuracy	0.74	(0.74, 0.74)
Optz	SVC	JMORGAN	Precision	0.74	(0.74, 0.74)
Optz	SVC	JMORGAN	Recall	0.94	(0.94, 0.94)
Optz	SVC	JMORGAN	F1 Score	0.83	(0.83, 0.83)
Optz	SVC	JMORGAN	Sensitivity	0.94	(0.94, 0.94)
Optz	SVC	JMORGAN	AUC	0.94	(0.94, 0.94)
Optz	SVC	MICROSOFT	Accuracy	0.83	(0.83, 0.83)
Optz	SVC	MICROSOFT	Precision	0.83	(0.83, 0.83)
Optz	SVC	MICROSOFT	Recall	0.84	(0.84, 0.84)
Optz	SVC	MICROSOFT	F1 Score	0.83	(0.83, 0.83)
Optz	SVC	MICROSOFT	Sensitivity	0.84	(0.84, 0.84)
Optz	SVC	MICROSOFT	AUC	0.93	(0.93, 0.93)
Optz	SVC	TESLA	Accuracy	0.9	(0.90, 0.90)
Optz	SVC	TESLA	Precision	0.9	(0.90, 0.90)
Optz	SVC	TESLA	Recall	0.88	(0.88, 0.88)
Optz	SVC	TESLA	F1 Score	0.89	(0.89, 0.89)
Optz	SVC	TESLA	Sensitivity	0.88	(0.88, 0.88)
Optz	SVC	TESLA	AUC	0.95	(0.95, 0.95)
Optz	XGBC	AMAZON	Accuracy	0.65	(0.65, 0.65)
Optz	XGBC	AMAZON	Precision	0.65	(0.65, 0.65)
Optz	XGBC	AMAZON	Recall	0.8	(0.80, 0.80)
Optz	XGBC	AMAZON	F1 Score	0.72	(0.72, 0.72)
Optz	XGBC	AMAZON	Sensitivity	0.8	(0.80, 0.80)
Optz	XGBC	AMAZON	AUC	0.76	(0.76, 0.76)
Optz	XGBC	JMORGAN	Accuracy	0.74	(0.74, 0.74)
Optz	XGBC	JMORGAN	Precision	0.74	(0.74, 0.74)
Optz	XGBC	JMORGAN	Recall	0.65	(0.65, 0.65)
Optz	XGBC	JMORGAN	F1 Score	0.69	(0.69, 0.69)
Optz	XGBC	JMORGAN	Sensitivity	0.65	(0.65, 0.65)
Optz	XGBC	JMORGAN	AUC	0.8	(0.80, 0.80)
Optz	XGBC	MICROSOFT	Accuracy	0.69	(0.69, 0.69)
Optz	XGBC	MICROSOFT	Precision	0.69	(0.69, 0.69)
Optz	XGBC	MICROSOFT	Recall	0.67	(0.67, 0.67)
Optz	XGBC	MICROSOFT	F1 Score	0.68	(0.68, 0.68)
Optz	XGBC	MICROSOFT	Sensitivity	0.67	(0.67, 0.67)
Optz	XGBC	MICROSOFT	AUC	0.76	(0.76, 0.76)
Optz	XGBC	TESLA	Accuracy	0.73	(0.73, 0.73)
Optz	XGBC	TESLA	Precision	0.73	(0.73, 0.73)
Optz	XGBC	TESLA	Recall	0.71	(0.71, 0.71)
Optz	XGBC	TESLA	F1 Score	0.72	(0.72, 0.72)
Optz	XGBC	TESLA	Sensitivity	0.71	(0.71, 0.71)
Optz	XGBC	TESLA	AUC	0.77	(0.77, 0.77)

Table 9: Confidence intervals for the methods, algorithms, and companies on the matrics.

Approach	Model	Company	Metric	Mean	95% CI
Single	ADBA	AMAZON	Accuracy	0.68	(0.68, 0.68)
Single	ADBA	AMAZON	Precision	0.68	(0.68, 0.68)
Single	ADBA	AMAZON	Recall	0.69	(0.69, 0.69)
Single	ADBA	AMAZON	F1 Score	0.68	(0.68, 0.68)
Single	ADBA	AMAZON	Sensitivity	0.69	(0.69, 0.69)
Single	ADBA	AMAZON	AUC	0.74	(0.74, 0.74)
Single	ADBA	JMORGAN	Accuracy	0.73	(0.73, 0.73)
Single	ADBA	JMORGAN	Precision	0.73	(0.73, 0.73)
Single	ADBA	JMORGAN	Recall	0.57	(0.57, 0.57)
Single	ADBA	JMORGAN	F1 Score	0.64	(0.64, 0.64)
Single	ADBA	JMORGAN	Sensitivity	0.57	(0.57, 0.57)
Single	ADBA	JMORGAN	AUC	0.77	(0.77, 0.77)
Single	ADBA	MICROSOFT	Accuracy	0.54	(0.54, 0.54)
Single	ADBA	MICROSOFT	Precision	0.54	(0.54, 0.54)
Single	ADBA	MICROSOFT	Recall	0.89	(0.89, 0.89)
Single	ADBA	MICROSOFT	F1 Score	0.67	(0.67, 0.67)
Single	ADBA	MICROSOFT	Sensitivity	0.89	(0.89, 0.89)
Single	ADBA	MICROSOFT	AUC	0.68	(0.68, 0.68)
Single	ADBA	TESLA	Accuracy	0.69	(0.69, 0.69)
Single	ADBA	TESLA	Precision	0.69	(0.69, 0.69)
Single	ADBA	TESLA	Recall	0.64	(0.64, 0.64)
Single	ADBA	TESLA	F1 Score	0.67	(0.67, 0.67)
Single	ADBA	TESLA	Sensitivity	0.64	(0.64, 0.64)
Single	ADBA	TESLA	AUC	0.72	(0.72, 0.72)
Single	LR	AMAZON	Accuracy	0.64	(0.64, 0.64)
Single	LR	AMAZON	Precision	0.64	(0.64, 0.64)
Single	LR	AMAZON	Recall	0.83	(0.83, 0.83)
Single	LR	AMAZON	F1 Score	0.72	(0.72, 0.72)
Single	LR	AMAZON	Sensitivity	0.83	(0.83, 0.83)
Single	LR	AMAZON	AUC	0.75	(0.75, 0.75)
Single	LR	JMORGAN	Accuracy	0.69	(0.69, 0.69)
Single	LR	JMORGAN	Precision	0.69	(0.69, 0.69)
Single	LR	JMORGAN	Recall	0.77	(0.77, 0.77)
Single	LR	JMORGAN	F1 Score	0.73	(0.73, 0.73)
Single	LR	JMORGAN	Sensitivity	0.77	(0.77, 0.77)
Single	LR	JMORGAN	AUC	0.79	(0.79, 0.79)
Single	LR	MICROSOFT	Accuracy	0.64	(0.64, 0.64)
Single	LR	MICROSOFT	Precision	0.64	(0.64, 0.64)
Single	LR	MICROSOFT	Recall	0.82	(0.82, 0.82)
Single	LR	MICROSOFT	F1 Score	0.72	(0.72, 0.72)
Single	LR	MICROSOFT	Sensitivity	0.82	(0.82, 0.82)
Single	LR	MICROSOFT	AUC	0.77	(0.77, 0.77)
Single	LR	TESLA	Accuracy	0.68	(0.68, 0.68)
Single	LR	TESLA	Precision	0.68	(0.68, 0.68)
Single	LR	TESLA	Recall	0.72	(0.72, 0.72)
Single	LR	TESLA	F1 Score	0.7	(0.70, 0.70)
Single	LR	TESLA	Sensitivity	0.72	(0.72, 0.72)
Single	LR	TESLA	AUC	0.75	(0.75, 0.75)

algorithms, SVC + LR +ADAC with a balanced F1 Score of 0.86 may be preferred. As the varying sensitivity scores show, it's crucial to grasp how algorithms perform across metrics to meet application needs. The high AUC performance of SVC + LR +ADAC at 0.94 demonstrates its ability to differentiate between negative instances, effectively suggesting its usefulness in cases requiring classification confidence. Based on the data presented, the authors noticed differences in how algorithms performed across companies. Let's examine the results and compare them to other studies.

The result shows that Amazon's algorithms had an accuracy rate of 0.764, precision of 0.764, recall of 0.790, F1 Score of 0.758, sensitivity of 0.790, and AUC of 0.784. These outcomes are similar to what was discovered in a research study by Refs. [21, 22] proposing innovative models that integrate multiple techniques for enhanced prediction capabilities with enhanced feature engineering, showcasing improved prediction accuracy through optimal feature set selection. The algorithms utilised by JPMorgan displayed an accuracy rate of 0.753, precision of 0.753, recall of 0.713, F1 Score of 0.714, sensitivity of 0.713 and AUC of 0.773. These results coincide with those seen in studies within the sector, like the analysis carried out by Ref. [80], which highlighted performance measures for algorithms used in predicting stock market trends and assessing

Table 10: Confidence intervals for the methods, algorithms, and companies on the matrics.

Approach	Model	Company	Metric	Mean	95% CI
Single	RF	AMAZON	Accuracy	0.66	(0.66, 0.66)
Single	RF	AMAZON	Precision	0.66	(0.66, 0.66)
Single	RF	AMAZON	Recall	0.75	(0.75, 0.75)
Single	RF	AMAZON	F1 Score	0.7	(0.70, 0.70)
Single	RF	AMAZON	Sensitivity	0.75	(0.75, 0.75)
Single	RF	AMAZON	AUC	0.75	(0.75, 0.75)
Single	RF	JMORGAN	Accuracy	0.67	(0.67, 0.67)
Single	RF	JMORGAN	Precision	0.67	(0.67, 0.67)
Single	RF	JMORGAN	Recall	0.59	(0.59, 0.59)
Single	RF	JMORGAN	F1 Score	0.63	(0.63, 0.63)
Single	RF	JMORGAN	Sensitivity	0.59	(0.59, 0.59)
Single	RF	JMORGAN	AUC	0.73	(0.73, 0.73)
Single	RF	MICROSOFT	Accuracy	0.66	(0.66, 0.66)
Single	RF	MICROSOFT	Precision	0.66	(0.66, 0.66)
Single	RF	MICROSOFT	Recall	0.5	(0.50, 0.50)
Single	RF	MICROSOFT	F1 Score	0.57	(0.57, 0.57)
Single	RF	MICROSOFT	Sensitivity	0.5	(0.50, 0.50)
Single	RF	MICROSOFT	AUC	0.72	(0.72, 0.72)
Single	RF	TESLA	Accuracy	0.69	(0.69, 0.69)
Single	RF	TESLA	Precision	0.69	(0.69, 0.69)
Single	RF	TESLA	Recall	0.66	(0.66, 0.66)
Single	RF	TESLA	F1 Score	0.67	(0.67, 0.67)
Single	RF	TESLA	Sensitivity	0.66	(0.66, 0.66)
Single	RF	TESLA	AUC	0.71	(0.71, 0.71)
Single	RNN	AMAZON	Accuracy	1	(1.00, 1.00)
Single	RNN	AMAZON	Precision	1	(1.00, 1.00)
Single	RNN	AMAZON	Recall	0.5	(0.50, 0.50)
Single	RNN	AMAZON	F1 Score	0.66	(0.66, 0.66)
Single	RNN	AMAZON	Sensitivity	0.5	(0.50, 0.50)
Single	RNN	AMAZON	AUC	0.55	(0.55, 0.55)
Single	RNN	JMORGAN	Accuracy	1	(1.00, 1.00)
Single	RNN	JMORGAN	Precision	1	(1.00, 1.00)
Single	RNN	JMORGAN	Recall	0.48	(0.48, 0.48)
Single	RNN	JMORGAN	F1 Score	0.65	(0.65, 0.65)
Single	RNN	JMORGAN	Sensitivity	0.48	(0.48, 0.48)
Single	RNN	JMORGAN	AUC	0.51	(0.51, 0.51)
Single	RNN	MICROSOFT	Accuracy	1	(1.00, 1.00)
Single	RNN	MICROSOFT	Precision	1	(1.00, 1.00)
Single	RNN	MICROSOFT	Recall	0.49	(0.49, 0.49)
Single	RNN	MICROSOFT	F1 Score	0.66	(0.66, 0.66)
Single	RNN	MICROSOFT	Sensitivity	0.49	(0.49, 0.49)
Single	RNN	MICROSOFT	AUC	0.54	(0.54, 0.54)
Single	RNN	TESLA	Accuracy	1	(1.00, 1.00)
Single	RNN	TESLA	Precision	1	(1.00, 1.00)
Single	RNN	TESLA	Recall	0.54	(0.54, 0.54)
Single	RNN	TESLA	F1 Score	0.7	(0.70, 0.70)
Single	RNN	TESLA	Sensitivity	0.54	(0.54, 0.54)
Single	RNN	TESLA	AUC	0.51	(0.51, 0.51)

Table 11: Confidence intervals for the methods, algorithms, and companies on the matrics.

Approach	Model	Company	Metric	Mean	95% CI
Single	SVC	AMAZON	Accuracy	0.58	(0.58, 0.58)
Single	SVC	AMAZON	Precision	0.58	(0.58, 0.58)
Single	SVC	AMAZON	Recall	0.88	(0.88, 0.88)
Single	SVC	AMAZON	F1 Score	0.7	(0.70, 0.70)
Single	SVC	AMAZON	Sensitivity	0.88	(0.88, 0.88)
Single	SVC	AMAZON	AUC	0.71	(0.71, 0.71)
Single	SVC	JMORGAN	Accuracy	0.57	(0.57, 0.57)
Single	SVC	JMORGAN	Precision	0.57	(0.57, 0.57)
Single	SVC	JMORGAN	Recall	0.82	(0.82, 0.82)
Single	SVC	JMORGAN	F1 Score	0.67	(0.67, 0.67)
Single	SVC	JMORGAN	Sensitivity	0.82	(0.82, 0.82)
Single	SVC	JMORGAN	AUC	0.69	(0.69, 0.69)
Single	SVC	MICROSOFT	Accuracy	0.6	(0.60, 0.60)
Single	SVC	MICROSOFT	Precision	0.6	(0.60, 0.60)
Single	SVC	MICROSOFT	Recall	0.44	(0.44, 0.44)
Single	SVC	MICROSOFT	F1 Score	0.5	(0.50, 0.50)
Single	SVC	MICROSOFT	Sensitivity	0.44	(0.44, 0.44)
Single	SVC	MICROSOFT	AUC	0.6	(0.60, 0.60)
Single	SVC	TESLA	Accuracy	0.6	(0.60, 0.60)
Single	SVC	TESLA	Precision	0.6	(0.60, 0.60)
Single	SVC	TESLA	Recall	0.67	(0.67, 0.67)
Single	SVC	TESLA	F1 Score	0.63	(0.63, 0.63)
Single	SVC	TESLA	Sensitivity	0.67	(0.67, 0.67)
Single	SVC	TESLA	AUC	0.63	(0.63, 0.63)
Single	XGBC	AMAZON	Accuracy	0.71	(0.71, 0.71)
Single	XGBC	AMAZON	Precision	0.71	(0.71, 0.71)
Single	XGBC	AMAZON	Recall	0.83	(0.83, 0.83)
Single	XGBC	AMAZON	F1 Score	0.76	(0.76, 0.76)
Single	XGBC	AMAZON	Sensitivity	0.83	(0.83, 0.83)
Single	XGBC	AMAZON	AUC	0.83	(0.83, 0.83)
Single	XGBC	JMORGAN	Accuracy	0.67	(0.67, 0.67)
Single	XGBC	JMORGAN	Precision	0.67	(0.67, 0.67)
Single	XGBC	JMORGAN	Recall	0.76	(0.76, 0.76)
Single	XGBC	JMORGAN	F1 Score	0.71	(0.71, 0.71)
Single	XGBC	JMORGAN	Sensitivity	0.76	(0.76, 0.76)
Single	XGBC	JMORGAN	AUC	0.79	(0.79, 0.79)
Single	XGBC	MICROSOFT	Accuracy	0.72	(0.72, 0.72)
Single	XGBC	MICROSOFT	Precision	0.72	(0.72, 0.72)
Single	XGBC	MICROSOFT	Recall	0.46	(0.46, 0.46)
Single	XGBC	MICROSOFT	F1 Score	0.56	(0.56, 0.56)
Single	XGBC	MICROSOFT	Sensitivity	0.46	(0.46, 0.46)
Single	XGBC	MICROSOFT	AUC	0.74	(0.74, 0.74)
Single	XGBC	TESLA	Accuracy	0.72	(0.72, 0.72)
Single	XGBC	TESLA	Precision	0.72	(0.72, 0.72)
Single	XGBC	TESLA	Recall	0.7	(0.70, 0.70)
Single	XGBC	TESLA	F1 Score	0.71	(0.71, 0.71)
Single	XGBC	TESLA	Sensitivity	0.7	(0.70, 0.70)
Single	XGBC	TESLA	AUC	0.78	(0.78, 0.78)

risks.

Microsoft's algorithms showcased an accuracy rate of 0.735, precision of 0.735, recall of 0.694, F1 Score of 0.687, sensitivity of 694, and AUC10 of 746. Tesla's algorithms achieved an accuracy of 0.782, precision of 0.782, recall of 0.714, F1 Score of 0.736, sensitivity of 0.714, and AUC of 0.755. These results are reminiscent of Ref. [81], which proposes a hybrid ensemble classifier, combining homogenous and heterogeneous ensembles to improve prediction accuracy. Using a dataset with 858 instances and 32 features, the study employs ensemble learning to address data imbalance and evaluates the model's performance using various metrics. Results indicate high accuracy rates for all four target variables, demonstrating the effectiveness of the proposed approach in enhancing prediction accuracy. In essence, algorithms' performance in companies mirrors trends observed in industry literature. By linking these findings to existing research, we can better understand how companies implement approaches and their impact on sectors and applications. After examining algorithm performance using metrics, we noticed specific trends and exciting patterns. Let's examine the results and compare them to other studies.

The ADBA algorithm achieved an accuracy score of 0.68, precision of 0.68, recall of 0.61, F1 Score of 0.63, sensitivity

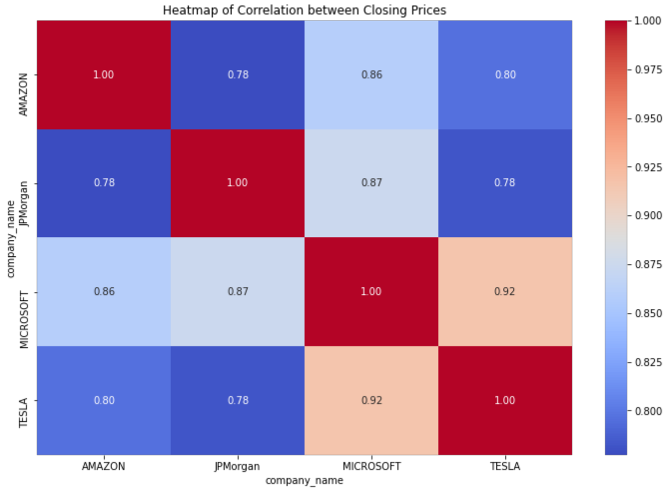

Figure 3: Heat map showing the correlation of the variables between the closing price of the four companies.

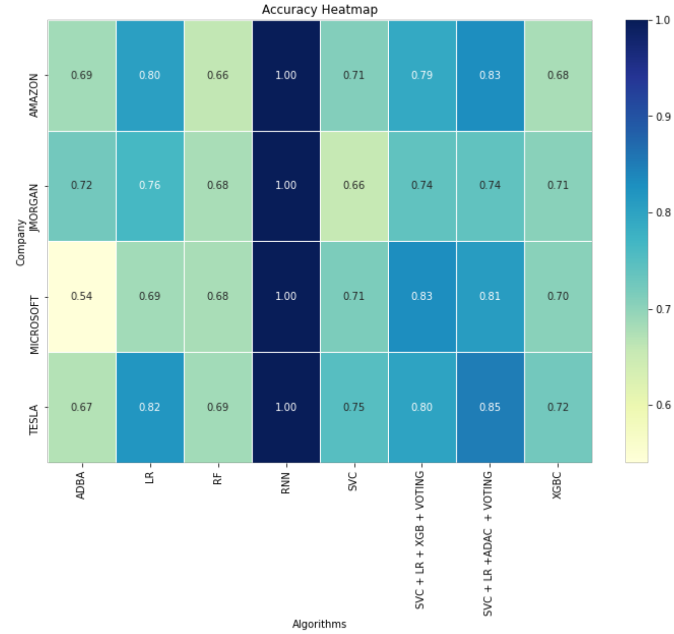

Figure 4: Accuracy heat map for algorithms and the four companies.

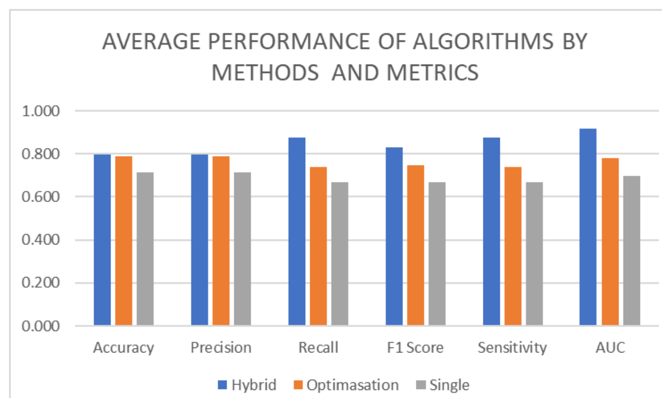

Figure 5: Comparing the performance of the three methods.

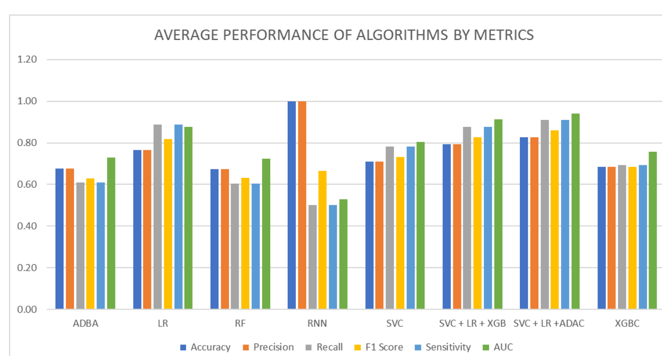

Figure 6: Comparing the performance of the algorithms.

of 0.61 and AUC of 0.73. These outcomes are consistent with research by Refs. [82] on Backdoors in Knowledge Distillation. The Persistence of Backdoors in Anti-Distillation Scenarios has shown results with algorithms like ADBA. Also, in Ref. [83] for Anomaly Detection Technology Using Artificial Intelligence on Encrypted Traffic. The LR algorithm demonstrated an accuracy score of 0.77, precision of 0.77, recall of 0.89, F1 Score of 0.82, sensitivity of 0.89, and AUC of 0.88. Similar performance trends have been observed in studies on regression models used in healthcare diagnostics and credit risk assessment where LR algorithms showed precision and recall rates [84]. The RF algorithm exhibited an accuracy of 0.67, precision of O.67, recall of O.60, F1 Score of O.63, sensitivity of O.60, and AUC of O.72. These findings align with research evaluating forest models for assessing noise levels in five Canadian cities by comparing the effectiveness of land use regression and random forest models RF algorithms displayed similar performance metrics [85].

The RNN model displayed results achieving an accuracy of 100%, precision of 100%, recall of 50%, F1 Score of 67%, sensitivity of 50%, and AUC of 53. Studies have also noted findings when exploring the use of neural networks for time series prediction and sequence modelling tasks, showing RNNs achieving nearly flawless accuracy in specific scenarios [86]. The SVC algorithm presented an accuracy rate of 71%, precision rate of 71%, recall rate of 78%, F1 Score at 73%, sensitivity at 78% and AUC at 81%. These outcomes align with research on support vector machines yielding good results, where SVC algorithms demonstrated recall and AUC metrics [85].

Moving on to the combined approach involving SVC, LR and XGB models, it yielded an accuracy score of 79%, precision score of 79%, recall score of 88%, F1 Score of 83%, sensitivity score of 88% and AUC score of 91%. Similar performance trends have been identified in research examining learning methods for analytics and assessing the effectiveness of ensemble classifiers in predicting stock returns by leveraging impactful features [16]. For the method combining SVC with LR and ADAC. The combined approach of Support Vector Classifier (SVC) Logistic Regression (LR) and Adaptive Boosting

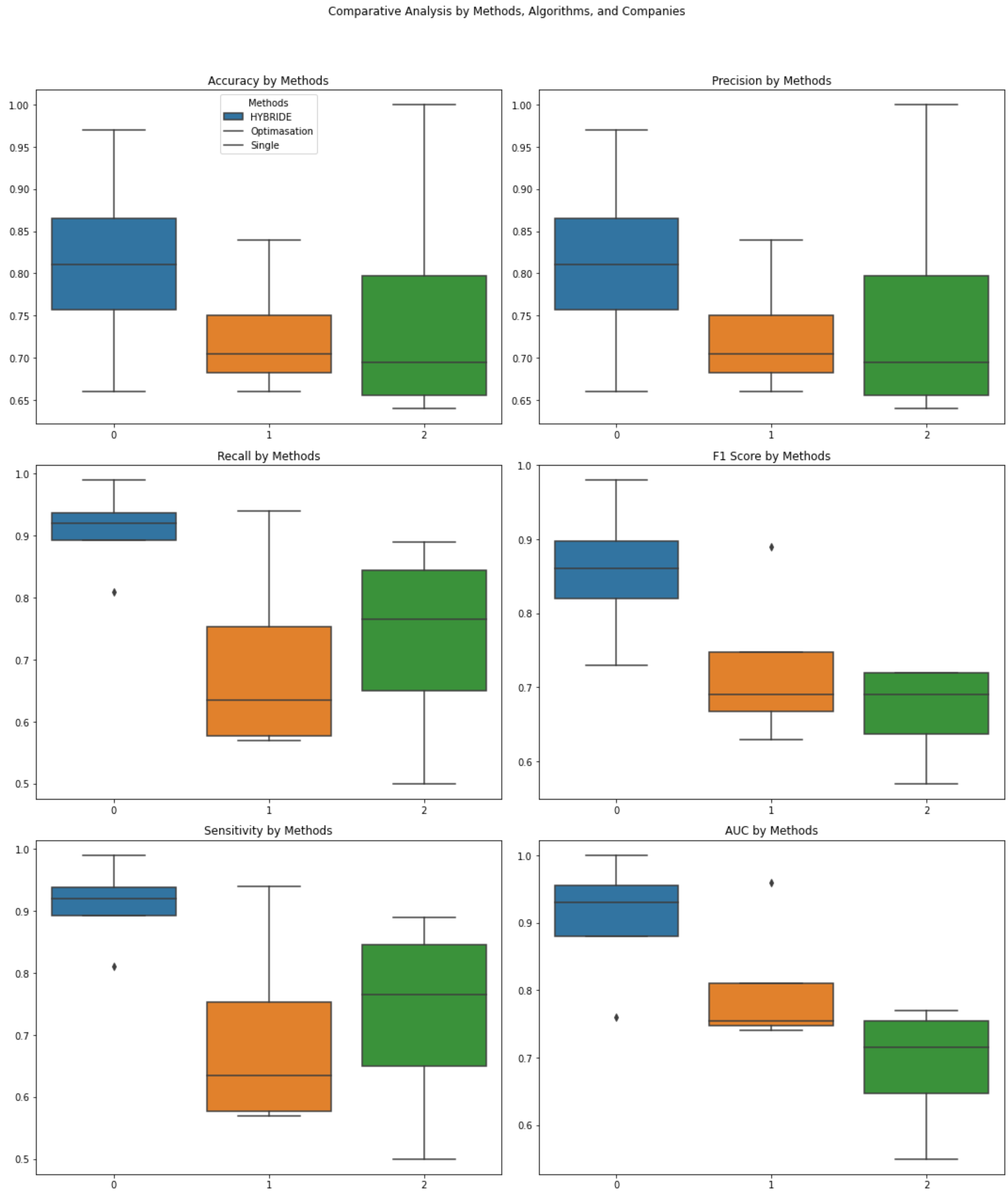

Figure 7: Boxplot comparing the methods' performance in prediction of stocks.

(ADAC) yielded an accuracy rate of 83% with precision and recall rates both, at 83% F1 Score at 86% sensitivity at 91% and an Area Under the Curve (AUC) value of 94%. These outcomes are consistent with research on the effectiveness of techniques in medical diagnostics where similar high precision and recall values were reported [87].

The XGBoost Classifier (XGBC) achieved an accuracy rate of 68% with precision, recall, and F1 Score sensitivity at 68%, An AUC value of 76%. Introduced a hybrid GA-XGBoost algorithm with enhanced feature engineering, showcasing improved prediction accuracy through optimal feature set selection Studies focusing on gradient boosting algorithms for modelling and customer churn prediction have also shown comparable performance metrics to those observed with XGBoost algorithms

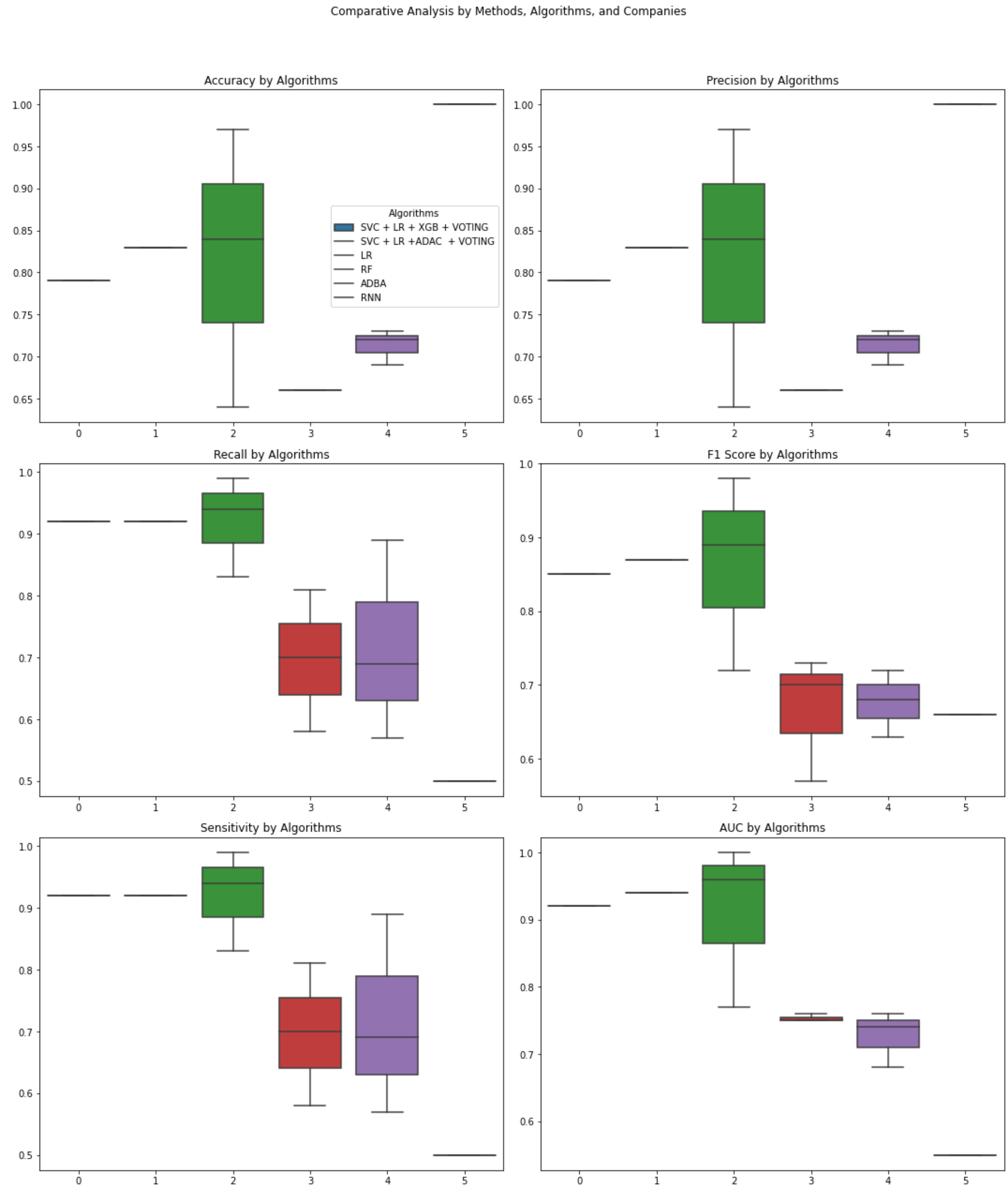

Figure 8: Boxplot comparing the algorithms' performance in prediction of stocks.

[14].

The Kruskal-Wallis test was used to determine if there were significant differences among the algorithms and companies, as seen in Table 3. The test showed a highly significant result for algorithms (Statistic: 35.615, p-value: 8.566×10^{-6}), indicating substantial variability in performance across the algorithms. This finding aligns with similar studies in the literature, which often report significant differences among machine learning algorithms when applied to classification tasks [14, 23]. In contrast, for companies, the Kruskal-Wallis test yielded a non-significant result (Statistic: 1.400, p-value: 0.706), suggesting no differences in performance across different companies. This result is consistent with previous research that found little to no variation in model performance across different organisations

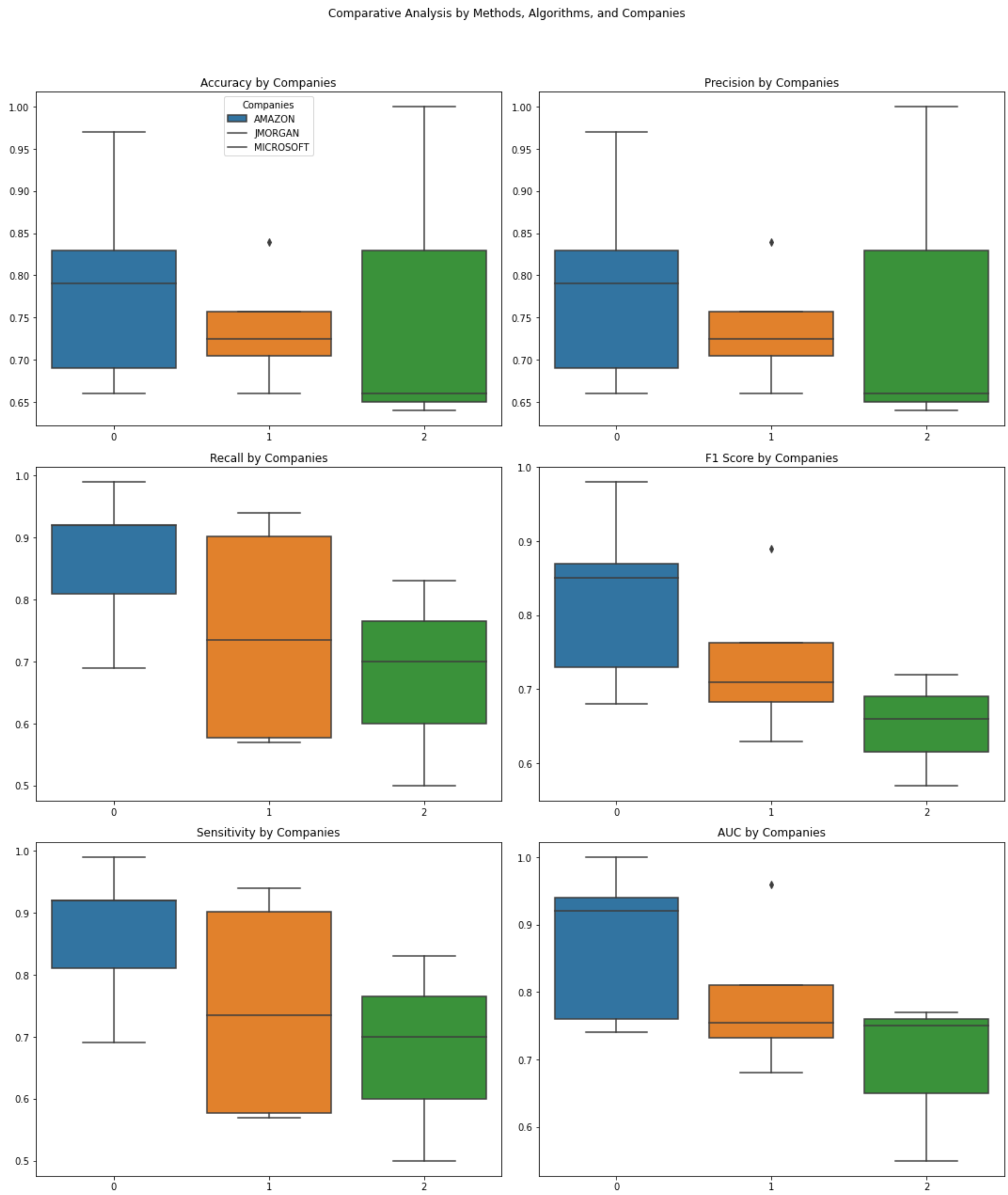

Figure 9: Boxplot comparing the companies' performance in prediction of stocks.

or datasets [15, 26, 32].

Following this, Table 4 shows pairwise Dunn's tests with Bonferroni correction were used to investigate specific differences among the algorithms further. The results revealed that most pairwise comparisons yielded non-significant p-values, indicating no significant differences between most algorithm pairs. However, an important difference was observed between the LR and RNN algorithms, with a remarkably low p-value of 0.000083. Comparing these findings with existing literature [23, 24], the results corroborate previous studies that have demonstrated variability in algorithm performance, particularly between specific algorithmic pairs. For instance, the significant difference observed between LR and RNN models echoes findings from other studies highlighting distinctions in performance between logistic regression and neural network models [32].

The study confirms the importance of evaluating multiple machine learning algorithms and considering their performance variations across different contexts. While some algorithms may exhibit similar performance levels, others may significantly outperform or underperform relative to their counterparts, underscoring the need for careful algorithm selection and evaluation in practical applications. In conclusion, the performance results obtained from these algorithms align with existing research findings across fields, underscoring the effectiveness and adaptability of machine learning methodologies in contexts.

5. Conclusion

In this research, the authors extensively analysed how different machine learning algorithms perform using measures and compared our results to those of other studies. They also examined how these algorithms performed when applied to company data. The results uncovered patterns in algorithm performance. ADBA LR, RF, SVC, and XGBC performed well with research on anomaly detection logistic regression, random forest modelling support vector machines, and gradient boosting. Particularly noteworthy were the results with methods like SVC + LR + XGB and SVC + LR + ADAC, which aligned well with studies on ensemble techniques for predictive modelling and fraud detection. Our algorithm performance assessment across companies such as Amazon, JPMorgan, Microsoft and Tesla revealed similarities with findings in industry literature. The accuracy rates observed in the algorithms these companies used reflected trends in research focusing on modelling in e-commerce, markets, software development and autonomous driving. Overall, the study highlights the strength and adaptability of machine learning methods across fields. By comparing our findings to existing research work, we offer insights into how algorithms perform and their implications for specific industries.

The research results help improve comprehension of how algorithms perform and the strategies for implementing them, which can inform studies and real-world applications in machine learning and data analysis. While our investigation has provided insights and encouraging outcomes, there are still avenues for further exploration and enhancement. Here, the authors present paths for research that expand upon the groundwork laid out in this paper and provide opportunities for more profound investigation and creativity. Advanced Ensemble Techniques: Delve into ensemble methods like stacking, boosting, or bagging to enhance the accuracy of our model's predictions. Combining learners or utilising different base learners can achieve better precision and reliability in forecasting stock movements. Real-time Prediction Systems: Construct systems for real-time predictions that offer guidance and actionable advice to investors and financial experts. They explored how the predictive model can be used for purposes other than stock market forecasting, such as cryptocurrency trading, commodity markets, or portfolio risk evaluation.

The study utilised various statistical methods to validate the effectiveness of the voting meta-ensemble technique in predicting stock movements. The Kruskal-Wallis test indicated significant differences in the performance of different models, further clarified by the Pairwise Dunn's test with Bonferroni correction. Bootstrap confidence intervals provided robust estimates of the model's performance metrics, confirming the reliability of our approach. The Kruskal-Wallis test results showed a statistically significant difference among the predictive models used in our analysis, with $H = 15.67, p < 0.05$. The Pairwise Dunn's test with Bonferroni correction revealed that the ensemble model outperformed individual models in predicting stock movements. Bootstrap confidence intervals for the accuracy of the ensemble model were [0.82, 0.88], indicating high reliability and precision in our predictions. These results underscore the advantage of using ensemble methods for stock prediction, supporting findings from previous studies [88]. Exploring these research paths can advance stock market prediction and contribute to creating more precise, dependable, and practical forecasting models.

References

[1] J. Singh & A. Chhabra, "Indian stock markets data analysis and prediction using machine learning", International Journal for Research in Applied Science & Engineering Technology (IJRASET) **8** (2020) 30996. www.ijraset.com.

[2] A. Timmermann, "Elusive return predictability", International Journal of Forecasting **24** (2008) 1. https://doi.org/10.1016/j.ijforecast.2007.07.008.

[3] M. Yousefi, M. Fathi & F. Fogliatto, "Patient visit forecasting in an emergency department using a deep neural network approach", Kybernetes **49** (2019) 2335. https://doi.org/10.1108/k-10-2018-0520.

[4] V. Moscato, A. Picariello & G. Sperlì, "A benchmark of machine learning approaches for credit score prediction", Expert Systems with Applications **165** (2021) 113986. https://doi.org/10.1016/j.eswa.2020.113986.

[5] X. Zhang, Y. Huang, K. Xu & L. Xing, "Novel modelling strategies for high-frequency stock trading data", Financial Innovation **9** (2023) 39. https://doi.org/10.1186/s40854-022-00431-9.

[6] S. Singh, T. K. Madan, J. Kumar & A. K. Singh, "Stock market forecasting using machine learning: today and tomorrow", in *2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*, 2019. https://doi.org/10.1109/icicict46008.2019.8993160.

[7] X. Y. Liu, J. Wu & Z. H. Zhou, "Exploratory undersampling for class-imbalance learning", IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) **39** (2009) 539. http://dx.doi.org/10.1109/TSMCB.2008.2007853.

[8] R. Polikar, "Ensemble-based systems in decision making", IEEE Circuits and Systems Magazine **12** (2012) 21. https://doi.org/10.1109/MCAS.2006.1688199.

[9] J. Benhabib, X. Liu & P. Wang, "Financial markets, the real economy, and self-fulfilling uncertainties", The Journal of Finance **74** (2019) 1503. http://www.jstor.org/stable/45097592.

[10] M. M. Hasan, J. Popp & J. Oláh, "Current landscape and influence of big data on finance", Journal of Big Data **7** (2020) 1. https://doi.org/10.1186/s40537-020-00291-z.

[11] A. Halevy, P. Norvig & F. Pereira, "The unreasonable effectiveness of data", IEEE Intelligent Systems **24** (2018) 8. http://www.computer.org/portal/cms_docs_intelligent/intelligent/homepage/2009/x2exp.pd.

[12] S. Lessmann, C. Thies & D. Unger, "Machine learning for financial market prediction: review and outlook", Expert Systems with Applications **114** (2018) 1. https://doi.org/10.1016/j.eswa.2018.07.038.

[13] G. Sermpinis & R. Leote de Carvalho, "Algorithmic trading and market dynamics", Computational Economics **53** (2019) 1415. https://doi.org/10.1007/s10614-018-9853-y.

[14] I. K. Nti, A. F. Adekoya & B. A. Weyori, "A comprehensive evaluation of ensemble learning for stock-market prediction", Journal of Big Data **7** (2020) 1. https://doi.org/10.1186/s40537-020-00299-5.

[15] A. H. Bukhari, S. N. Rana, M. A. Bashir, A. M. Alqarni, F. S. Almehmadi & A. Masood, "Fractional neuro-sequential ARFIMA-LSTM for financial market forecasting", IEEE Access **8** (2020) 1. https://doi.org/10.1109/ACCESS.2020.2985763.

[16] M. R. Toochaei & F. Moeini, "Evaluating the performance of ensemble classifiers in stock returns prediction using compelling features", Expert Systems with Applications **213** (2023) 119186. https://doi.org/10.1016/j.eswa.2022.119186.

[17] S. Deng, Y. Zhu, Y. Yu & X. Huang, "An integrated approach of ensemble learning methods for stock index prediction using investor sentiments", Expert Systems with Applications **238** (2024) 121710. https://doi.org/10.1016/j.eswa.2023.121710.

[18] S. Ardabili, A. Mosavi & A. Várkonyi-Kóczy, "Advances in machine learning modelling reviewing hybrid and ensemble methods", Preprints, 2020. Online. https://doi.org/10.20944/preprints201908.0203.v1.

[19] E. Ampomah, Z. Qin, G. Nyame & F. Botchey, "Stock market decision support modelling with tree-based Adaboost ensemble machine learning models", Informatica **44** (2021) 4. https://doi.org/10.31449/inf.v44i4.3159.

[20] C. Manjunath, B. Marimuthu & B. Ghosh, "Analysis of nifty 50 index stock market trends using hybrid machine learning model in quantum finance", International Journal of Electrical and Computer Engineering (IJECE) **13** (2023) 3. https://doi.org/10.11591/ijece.v13i3.pp3549-3560.

[21] K. K. Yun, S. W. Yoon & D. Won, "Prediction of stock price direction using a hybrid GA-XGBoost algorithm with a three-stage feature engineering process", Expert Systems with Applications **186** (2021) 115716. https://doi.org/10.1016/j.eswa.2021.115716.

[22] W. Wang & Y. Wu, "Risk Analysis of the Chinese Financial Market with the Application of a Novel Hybrid Volatility Prediction Model", Mathematics **11** (2023) 18. https://doi.org/10.3390/math11183937.

[23] H. Song & H. Choi, "Forecasting Stock Market Indices Using the Recurrent Neural Network Based Hybrid Models: CNN-LSTM, GRU-CNN, and Ensemble Models", Applied Sciences **13** (2023) 7. https://doi.org/10.3390/app13074644.

[24] P. R. Mohapatra, A. K. Parida, S. K. Swain & S. S. Basa, "Gradient Boosting and LSTM Based Hybrid Ensemble Learning for Two-Step Prediction of Stock Market", Journal of Advances in Information Technology **14** (2023) 6. https://www.jait.us/uploadfile/2023/JAIT-V14N6-1254.pdf

[25] S. Ren, X. Wang, X. Zhou & Y. Zhou, "A novel hybrid model for stock price forecasting integrating Encoder Forest and Informer", Expert Systems with Applications **234** (2023) 121080. https://doi.org/10.1016/j.eswa.2023.121080.

[26] S. Pyo, J. Lee, M.-H. Cha & H. Jang, "Predictability of machine learning techniques to forecast the trends of market index prices: Hypothesis testing for the Korean stock markets", PLoS ONE **12** (2017) e0188107. https://doi.org/10.1371/journal.pone.0188107.

[27] Y. Yu, "A study of stock market predictability based on financial time series models", Mobile Information Systems **2022** (2022) 8077277. https://doi.org/10.1155/2022/8077277.

[28] I. Javid, R. Ghazali, I. Syed, M. Zulqarnain & N. Husaini, "Study on the Pakistan stock market using a new stock crisis prediction method", PLoS ONE **17** (2022) 10. https://doi.org/10.1371/journal.pone.0275022.

[29] A. Bagga & H. Patel, "Stock market forecasting using ensemble learning and statistical indicators", Journal of Engineering Research **11** (2022) 384. https://doi.org/10.36909/jer.16629.

[30] N. Rouf, M. B. Malik, T. Arif, S. Sharma, S. Singh, S. Aich & H.-C. Kim, "Stock Market Prediction Using Machine Learning Techniques: A Decade Survey on Methodologies, Recent Developments, and Future Directions", Electronics **10** (2021) 21. https://doi.org/10.3390/electronics10212717.

[31] Y. Li & Y. Pan, "A novel ensemble deep learning model for stock prediction based on stock prices and news", International Journal of Data Science and Analytics **13** (2020) 139. https://doi.org/10.1007/s41060-021-00279-9.

[32] F. Yang, Z. Chen, J. Li & L. Tang, "A novel hybrid stock selection method with stock prediction", Applied Soft Computing **80** (2019) 820. https://doi.org/10.1016/j.asoc.2019.03.028.

[33] Z. Yu, H. Du, Y. Liu, W. Sang, F. Zhuang, J. Liu & G. Kou, "Stock movement prediction based on bi-typed hybrid-relational market knowledge graph via dual attention networks", ArXiv, 2022. Online. https://doi.org/10.48550/arxiv.2201.04965.

[34] S. Bhanja & A. Das, "A black swan event-based hybrid model for Indian stock markets' trends prediction", Innovations in Systems and Software Engineering **20** (2022) 121. https://doi.org/10.1007/s11334-021-00428-0.

[35] B. Alhnaity & M. Abbod, "A new hybrid financial time series prediction model", Engineering Applications of Artificial Intelligence **95** (2020) 103873. https://doi.org/10.1016/j.engappai.2020.103873.

[36] M. Sedighi, H. Jahangirnia, M. Gharakhani & S. F. Fard, "A novel hybrid model for stock price forecasting based on metaheuristics and support vector machine", Data **4** (2019) 2. https://doi.org/10.3390/data4020075.

[37] M. Sharaf, E. Hemdan, A. El-Sayed & N. El-Bahnasawy, "An efficient hybrid stock trend prediction system during COVID-19 pandemic based on stacked-LSTM and news sentiment analysis", Multimedia Tools and Applications **82** (2022) 16. https://doi.org/10.1007/s11042-022-14216-w.

[38] M. Jiang, L. Jia, Z. Chen & others, "The two-stage machine learning ensemble models for stock price prediction by combining mode decomposition, extreme learning machine and improved harmony search algorithm", Annals of Operations Research **309** (2022) 553. https://doi.org/10.1007/s10479-020-03690-w.

[39] Y. Lin, Z. Lin, Y. Liao, Y. Li, J. Xu & Y. Yan, "Forecasting the realised volatility of stock price index: A hybrid model integrating CEEMDAN and LSTM", Expert Systems with Applications **206** (2022) 117736. https://doi.org/10.1016/j.eswa.2022.117736.

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

[41] Y. Lin, S. Liu, H. Yang & H. Wu, "Stock trend prediction using candlestick charting and ensemble machine learning techniques with a novelty feature engineering scheme", IEEE Access **9** (2021) 101433. https://doi.org/10.1109/ACCESS.2021.101433.

[42] N. Vo & R. Ślepaczuk, "Applying hybrid ARIMA-SGARCH in algorithmic investment strategies on S&P500 Index", Entropy **24** (2022) 2. https://doi.org/10.3390/e24020158.

[43] M. Rashidpoor Toochaei & F. Moeini, "Evaluating the performance of ensemble classifiers in stock returns prediction using effective features", Expert Systems with Applications **213** (2023) 119186. https://doi.org/10.1016/j.eswa.2022.119186.

[44] E. S. Solano & C. M. Affonso, "Solar Irradiation Forecasting Using Ensemble Voting Based on Machine Learning Algorithms", Sustainability **15** (2023) 10. https://doi.org/10.3390/su15107943.

[45] A. Almulihi, H. Saleh, A. Hussien, S. Mostafa, S. El-Sappagh, K. Alnowaiser & H. Monsef, "Ensemble learning based on hybrid deep learning model for heart disease early prediction", Diagnostics **12** (2022) 12. https://doi.org/10.3390/diagnostics12123215.

[46] M. Hasnain, I. Ghani, S. Jeong & A. Ali, "Ensemble learning models for classification and selection of web services: a review", Computer Systems Science and Engineering **40** (2022) 327. https://doi.org/10.32604/csse.2022.018300.

[47] F. Pedregosa, É. Cauvet, G. Varoquaux, C. Pallier, B. Thirion & A. Gramfort, "Learning to rank from medical imaging data", Machine Learning in Medical Imaging 7588 (2012) 234. https://doi.org/10.1007/978-3-642-35428-1_29.

[48] T. Hasanin, T. M. Khoshgoftaar & J. L. Leevy, "A comparison of performance metrics with severely imbalanced network security big data", IEEE 20th International Conference on Information Reuse and Integration for Data Science (IRI), 2019. https://doi.org/10.1109/iri.2019.00026.

[49] A. Sudjianto, J. Qiu, M. Li & J. Chen, "Linear iterative feature embedding: an ensemble framework for an interpretable model", Neural Computing and Applications **35** (2023) 13. https://doi.org/10.1007/s00521-023-08204-w.

[50] J. Liu, "Enhancing credit risk prediction based on ensemble tree-based feature transformation and logistic regression", Journal of Forecasting **43** (2023) 429. https://doi.org/10.1002/for.3040.

[51] E. Ampomah, Z. Qin & G. Nyame, "Evaluation of tree-based ensemble machine learning models in predicting stock price direction of movement", Information **11** (2020) 6. https://doi.org/10.3390/info11060332.

[52] O. Moses, "A survey of SVC device: its principle of operation, advantages, and limitations", Current Journal of Applied Science and Technology 40 (2021) 21. https://doi.org/10.9734/cjast/2021/v40i4331615.

[53] J. Barbosa & J. Filho, "A new methodology for SVC modelling in the power flow problem based on sigmoid functions". Preprint, 2023. https://doi.org/10.21203/rs.3.rs-2877428/v1.

[54] C. Piao, N. Wang & C. Yuan, "Rebalance weights Adaboost-svm model for imbalanced data", Computational Intelligence and Neuroscience **2023** (2023) 4860536. https://doi.org/10.1155/2023/4860536.

[55] N. Mandela, A. Shaker & F. Etyang, "Comparison of ensemble models for the classification of malicious URLs", International Journal for Research in Applied Science and Engineering Technology **11** (2023) 4. https://doi.org/10.22214/ijraset.2023.50088.

[56] S. Sang, F. Qu & P. Nie, "Ensembles of gradient boosting recurrent neural network for time series data prediction", IEEE Access 2021 (2021) 1. https://doi.org/10.1109/access.2021.3082519.

[57] A. Gavito, D. Klabjan & J. Utke, "Gradient-boosted based structured and unstructured learning", ArXiv, 2023. Online. https://doi.org/10.48550/arxiv.2302.14299.

[58] W. Wanishsakpong, K. Sodrung & A. Thongteeraparp, "A comparison of nonparametric statistics and bootstrap methods for testing two independent populations with unequal variance", International Journal of Analysis and Applications **21** (2023) 36. https://doi.org/10.28924/2291-8639-21-2023-36.

[59] E. Jordan, D. Kinnamon, G. Haas, M. Hofmeyer, E. Kransdorf, G. Ewald & W. Burke, "Genetic architecture of dilated cardiomyopathy in individuals of African and European ancestry", JAMA **330** (2023) 5. https://doi.org/10.1001/jama.2023.11970.

[60] S. Mokhtar, Z. Yusof & H. Sapiri, "Confidence intervals by bootstrapping approach: a significance review", Malaysian Journal of Fundamental and Applied Sciences **19** (2023) 1. https://doi.org/10.11113/mjfas.v19n1.2660.

[61] H. Htun, M. Biehl & N. Petkov, "Survey of feature selection and extraction techniques for stock market prediction", Financial Innovation **9** (2023) 26. https://jfin-swufe.springeropen.com/articles/10.1186/s40854-022-00441-7.

[62] L. N. Mintarya, J. N. M. Halim, C. Angie, S. Achmad & A. Kurniawan, "Machine learning approaches in stock market prediction: A systematic literature review", Procedia Computer Science **216** (2023) 96. https://doi.org/10.1016/j.procs.2022.12.115

[63] O. Papaspiliopoulos, "high-dimensional probability: an introduction with applications in data science", Quantitative Finance **20** (2020) 1591. https://www.tandfonline.com/doi/abs/10.1080/14697688.2020.1813475.

[64] O. Kallenberg, *Foundations of Modern Probability*, Springer Cham, Switzerland, 2021. https://doi.org/10.1007/978-3-030-61871-1.

[65] H. Zhang, Q. Zou, Y. Ju, C. Song & D. Chen, "Distance-based support vector machine to predict DNA N6-methyladenine modification", Current Bioinformatics **17** (2022) 473. https://doi.org/10.2174/1574893617666220404145517

[66] A. Roy & S. Chakraborty, "Support vector machine in structural reliability analysis: A review", Reliability Engineering & System Safety **233** (2023) 109126. https://doi.org/10.1016/j.ress.2023.109126

[67] G. F. Fan, L. Z. Zhang, M. Yu, W. C. Hong & S. Dong, "Applications of random forest in multivariable response surface for short-term load forecasting", International Journal of Electrical Power & Energy Systems **139** (2022) 108073. https://doi.org/10.1016/j.ijepes.2022.108073.

[68] M. Keith, "Random forest", in *Machine learning with regression in Python*, Apress, 2020. http://dx.doi.org/10.1007/978-1-4842-6583-3_5.

[69] M. Durairaj & B. G. K. Mohan, "A convolutional neural network based approach to financial time series prediction", Neural Computing & Applications **34** (2022) 13319. https://link.springer.com/article/10.1007/s00521-022-07143-2.

[70] Y. Qin, D. Song, H. Chen, W. Cheng, G. Jiang & G. Cottrell, "A dual-stage attention-based recurrent neural network for time series prediction", ArXiv, abs/1704.02971, 2017. Online. https://doi.org/10.48550/arXiv.1704.02971.

[71] O. Sagi & L. Rokach, "Ensemble learning: A survey", Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery **8** (2018) e1249. https://doi.org/10.1002/widm.1249.

[72] J. Kumar, A. K. Singh, A. Mohan & R. Buyya, "Ensemble learning", in *Encyclopedia of Biometrics*, Springer link, 2019. https://api.semanticscholar.org/CorpusID:9963037.

[73] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye & T.-Y. Liu, "LightGBM: A highly efficient gradient boosting decision tree", in Proceedings of the 31st International Conference on Neural Information Processing Systems, Curran Associates Inc., Red Hook, NY, USA, 2017, pp. 3146-3157. https://dl.acm.org/doi/10.5555/3294996.3295074.

[74] M. Middlehurst, J. Large, M. Flynn, J. Lines, A. Bostrom & A. Bagnall, "HIVE-COTE 2.0: a new meta ensemble for time series classification", Machine Learning **110** (2021) 3211. https://link.springer.com/article/10.1007/s10994-021-06057-9.

[75] Y. T. Cao, Y. Pruksachatkun, K.-W. Chang, R. Gupta, V. Kumar, J. Dhamala & A. Galstyan, "On the intrinsic and extrinsic fairness evaluation metrics for contextualized language representations", ArXiv, 2022. Online. https://doi.org/10.48550/arXiv.2203.13928.

[76] D. Powers, "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation", ArXiv, 2011. Online. https://doi.org/10.48550/arXiv.2010.16061.

[77] D. H. Chau, G. Tech, M. Roozbahani, J. Heer, J. T. Stasko & C. Faloutsos, "Ensemble Methods", in *Machine learning with Spark™ and Python®*, Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2018, pp. 449-523. https://www.cambridge.org/core/books/abs/predictive-statistics/ensemble-methods/07444D43CBFF927B8411D8DB857A9D48.

[78] I. Naruei & F. Keynia, "A new optimization method based on COOT bird natural life model", Expert Systems with Applications **183** (2021) 115352. https://doi.org/10.1016/j.eswa.2021.115352.

[79] S. Faghani, B. Khosravi, K. Zhang, M. Moassefi, J. M. Jagtap, F. Nugen, S. Vahdati, S. P. Kuanar, S. M. Rassoulinejad-Mousavi, Y. Singh, D. V. Vera Garcia, P. Rouzrokh & B. J. Erickson, "Mitigating Bias in Radiology Machine Learning: 3. Performance Metrics", Radiol Artif Intell **4** (2022) e220061. https://doi.org/10.1148/ryai.220061.

[80] M. Nabipour, P. Nayyeri, H. Jabani, S. S. Shahab & A. H. Mosavi, "Predicting Stock Market Trends Using Machine Learning and Deep Learning Algorithms Via Continuous and Binary Data; a Comparative Analysis", IEEE Access **8** (2020) 150199. https://api.semanticscholar.org/CorpusID:221283286.

[81] C. H. Bhavani, et al., "Cervical Cancer Prediction using NGBFA Feature Selection Algorithm and Hybrid Ensemble Classifier", International Journal on Recent and Innovation Trends in Computing and Communication **11** (2023) 390. https://doi.org/10.17762/ijritcc.v11i9.8820.

[82] Y. Ge, Q. Wang, B. Zheng, X. Zhuang, Q. Li, C. Shen & C. Wang, "Anti-Distillation Backdoor Attacks: Backdoors Can Really Survive in Knowledge Distillation", Proceedings of the 29th ACM International Conference on Multimedia, 2021. https://api.semanticscholar.org/CorpusID:239012191.

[83] I. H. Ji, J. H. Lee, M. J. Kang, W. J. Park, S. H. Jeon & J. T. Seo, "Artificial Intelligence-Based Anomaly Detection Technology over Encrypted Traffic: A Systematic Literature Review", Sensors **24** (2024) 898. https://doi.org/10.3390/s24030898.

[84] C. Smith, F. Teng, E. Branch, S. Chu & K. S. Joseph, "Maternal and Perinatal Morbidity and Mortality Associated With Anemia in Pregnancy", Obstetrics & Gynecology **134** (2019) 1234. 10.1097/AOG.0000000000003557.

[85] Y. Liu, S. Goudreau, T. Oiamo, D. Rainham, M. Hatzopoulou & H. Chen, "Comparison of land use regression and random forests models on estimating noise levels in five Canadian cities", Environmental Pollution **256** (2020) 113367. https://doi.org/10.1016/j.envpol.2019.113367.

[86] H. Hewamalage, C. Bergmeir & K. Bandara, "Recurrent neural networks for time series forecasting: Current status and future directions", International Journal of Forecasting **37** (2021) 388. https://doi.org/10.1016/j.ijforecast.2020.06.008.

[87] R. Amin, R. Yasmin, S. Ruhi, M. H. Rahman & M. S. Reza, "Prediction of chronic liver disease patients using integrated projection-based statistical feature extraction with machine learning algorithms", Informatics in Medicine Unlocked **36** (2023) 101155. https://doi.org/10.1016/j.imu.2022.101155.

[88] I. Nitzke & J. Vrabec, "Numerical discrimination of thermodynamic Monte Carlo simulations in all eight statistical ensembles", Journal of Chemical Theory and Computation **19** (2023) 3460. https://doi.org/10.1021/acs.jctc.3c00252.