**Predicting Stock Price Movements Using Machine Learning: A Comparative Analysis**

​

Vaishnave Jonnalagadda  
Graduate School of Arts and Sciences   
Fordham University New York, United States  
[vj1@fordham.edu](mailto:vj1@fordham.edu)

Amrutha Vinayaka  
Graduate School of Arts and Sciences   
Fordham University New York, United States  
[av3@fordham.edu](mailto:av3@fordham.edu)

***Abstract***

In the fast-paced landscape of entrepreneurship, the successful launch of a new business or product hinges on a profound understanding of market dynamics. Traditional market research methods, while crucial, often fall short in capturing nuanced details essential for effective market entry. This research addresses this challenge by advocating the use of descriptive data mining algorithms, particularly clustering, to comprehensively analyze market segments. Market segmentation, the process of categorizing a broader market based on specific criteria, serves as the foundation for this study. By deploying unsupervised machine learning techniques, the research aims to uncover hidden patterns within extensive datasets, facilitating a deeper comprehension of customer needs and behaviors. The primary goal is to empower businesses to make strategic decisions during the critical early stages of their ventures by identifying specific clusters where customer interest is likely to be most pronounced. Through the synthesis of market segmentation principles and advanced data mining techniques, this research provides entrepreneurs and business leaders with a strategic roadmap for navigating the complexities of market dynamics, ultimately enhancing the chances of success in the early phases of business development. While the study is ongoing, preliminary findings suggest that this innovative approach has the potential to reveal previously unidentified customer segments, offering businesses a competitive edge in tailoring their products and services to meet specific market demands.

Keywords Stock Market Prediction, Supervised Machine Learning

# Introduction

The prediction of stock price movements has been a longstanding challenge in financial markets. Traditional methods, such as fundamental and technical analysis, have limitations in capturing the complexities of market dynamics. With the advent of machine learning, there has been a surge in research aimed at developing predictive models that can leverage the vast amount of data available in financial markets.

Machine learning algorithms offer a promising approach to stock market prediction by extracting patterns and relationships from historical data to make informed forecasts. These algorithms can analyze large datasets containing various financial indicators and market variables, providing insights that traditional methods may overlook.

# background

The prediction of stock price movements has been a longstanding challenge in financial markets. Traditional methods, such as fundamental and technical analysis, have limitations in capturing the complexities of market dynamics. With the advent of machine learning, there has been a surge in research aimed at developing predictive models that can leverage the vast amount of data available in financial markets.

Machine learning algorithms offer a promising approach to stock market prediction by extracting patterns and relationships from historical data to make informed forecasts. These algorithms can analyze large datasets containing various financial indicators and market variables, providing insights that traditional methods may overlook.

# Methodology

In this study, we focus on predicting stock price movements for Intel Corp (INTC) using machine learning classification approaches. We explore two distinct trading strategies:

1. Strategy 1: Buy or sell based on whether the next trading day's closing price is higher or lower than the current day's closing price.
2. Strategy 2: Utilize the 50-day moving average vs. the 200-day moving average to identify bullish and bearish signals.

We follow a systematic process to implement these strategies, including:

* Data acquisition: Obtaining historical financial data for INTC.
* Preprocessing: Cleaning and formatting the data for analysis.
* Feature engineering: Selecting and creating relevant features for the models.
* Label generation: Defining the target variable (i.e., whether the price will increase or decrease).
* Model selection: Choosing machine learning classifiers for each strategy.
* Evaluation: Assessing the performance of the models on a test dataset.

We implement several machine learning classifiers for both strategies, including:

* K-Nearest Neighbors (KNN)
* Random Forest Classifier (RF)
* Gradient Boosting Classifier (GB)
* Support Vector Machines (SVMs)
* XGBoost Classifier
* Long Short Term Memory (LSTM)

1. K-Nearest Neighbors (KNN):

* KNN is a simple yet effective algorithm that classifies data points based on the majority class among their nearest neighbors.
* In stock market prediction, KNN can identify similar patterns in historical price movements and classify new instances accordingly.
* It's particularly useful when there are well-defined clusters or patterns in the data, making it suitable for identifying trends in stock prices.

1. Random Forest Classifier (RF):

* Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes as the prediction.
* RF is highly versatile and robust, capable of handling large datasets with high dimensionality.
* In stock market prediction, RF can capture complex relationships between various financial indicators and predict price movements with high accuracy.
* Its ability to handle noise and overfitting makes it well-suited for analyzing noisy financial data.

1. Gradient Boosting Classifier (GB):

* Gradient Boosting is another ensemble learning technique that builds decision trees sequentially, each correcting the errors of its predecessor.
* GB is known for its superior predictive performance and ability to handle heterogeneous data.
* In stock market prediction, GB can effectively capture subtle patterns and trends in the data, leading to accurate forecasts.
* Its iterative nature allows it to continuously improve model performance, making it ideal for predicting stock price movements over time.

1. Support Vector Machines (SVMs):

* SVMs are powerful supervised learning models used for classification and regression tasks.
* SVMs work by finding the hyperplane that best separates classes in a high-dimensional feature space.
* In stock market prediction, SVMs can identify nonlinear relationships between financial indicators and predict price movements with high precision.
* SVMs are particularly effective when dealing with small to medium-sized datasets and can handle both linear and nonlinear decision boundaries.

1. XGBoost Classifier:

* XGBoost is an optimized implementation of gradient boosting that offers superior performance and scalability.
* XGBoost builds upon the principles of gradient boosting and incorporates several regularization techniques to prevent overfitting.
* In stock market prediction, XGBoost can handle large datasets efficiently and produce highly accurate predictions.
* Its speed and efficiency make it suitable for real-time applications in high-frequency trading environments.

1. Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) architecture that excels in capturing sequential patterns and dependencies within time series data. When applied to stock market prediction, LSTM offers several advantages:

Sequential Modeling:

Stock market data is inherently sequential, with each data point dependent on previous observations. LSTM is designed to model such sequential dependencies, making it well-suited for analyzing historical price movements, trading volumes, and other financial indicators over time.

Long-Term Dependencies:

* Traditional neural networks, including standard RNNs, struggle with capturing long-term dependencies due to the vanishing gradient problem. LSTM addresses this issue by introducing a memory cell and gating mechanisms, allowing it to retain relevant information over extended time periods.
* In stock market prediction, LSTM's ability to capture long-term dependencies enables it to learn from past market trends and incorporate them into its forecasts, leading to more accurate predictions of future price movements.

Handling Time Lags:

* Stock prices may exhibit time lags, where the effects of certain events on market dynamics are not immediately reflected in prices. LSTM is capable of learning and accounting for such time lags by analyzing historical data over multiple time steps.
* By capturing the delayed responses of stock prices to external factors, LSTM can provide insights into the dynamics of market reactions and improve the timing of trading decisions.

Feature Representation Learning:

* LSTM has the ability to automatically learn complex feature representations from raw data, reducing the need for manual feature engineering. This is particularly advantageous in stock market prediction, where relevant features may be nonlinear and high-dimensional.
* By extracting meaningful representations of financial indicators, LSTM can uncover latent patterns in the data that contribute to more accurate predictions of stock price movements.

Flexibility in Model Architecture:

* LSTM allows for flexible model architectures, including stacked LSTM layers, bidirectional LSTMs, and attention mechanisms. These architectural variations enable analysts to tailor the model to the specific characteristics of the financial data being analyzed.
* By experimenting with different LSTM architectures, analysts can explore various ways to enhance model performance and uncover insights into the underlying dynamics of the stock market.

# Implementation:

Before delving into modeling and prediction, it's crucial to preprocess the raw data and gain insights through exploratory data analysis (EDA). In this project, we obtained historical stock data for Intel Corp (INTC) from the Yahoo Finance API, which serves as a valuable resource for financial data retrieval.

**Data Retrieval:**

* We utilized the Yahoo Finance API to fetch historical stock data for INTC, including daily opening, closing, high, and low prices, as well as trading volume.
* The data spanned a specified time period, typically several years, capturing a comprehensive view of INTC's historical performance.

**Data Preprocessing:**

* Upon retrieval, the raw data underwent preprocessing to ensure consistency and suitability for analysis.
* Preprocessing steps included handling missing values, removing duplicates, and ensuring data integrity.
* Additionally, the data may have been adjusted for factors such as stock splits and dividends to reflect accurate price movements.

**Exploratory Data Analysis (EDA)**

* EDA plays a pivotal role in understanding the underlying characteristics and patterns within the data.
* We conducted various analyses and visualizations to gain insights into INTC's stock performance over time.
* Key components of EDA may include:
* Summary Statistics: Calculating descriptive statistics such as mean, median, standard deviation, and percentiles to understand the central tendency and variability of stock prices and trading volume.
* Time Series Analysis: Visualizing trends, seasonality, and fluctuations in stock prices and volume over different time periods.
* Correlation Analysis: Examining the relationships between different variables, such as the correlation between stock prices and trading volume, to identify potential patterns or dependencies.
* Volatility Analysis: Assessing the volatility of stock prices using measures such as standard deviation or historical volatility, which can inform risk management strategies.
* Moving Averages: Calculating and visualizing moving averages of stock prices (e.g., 50-day and 200-day moving averages) to identify trends and potential buy or sell signals.

**Insights from EDA:**

* EDA provides valuable insights into the behavior and dynamics of INTC's stock price movements.
* It helps identify patterns, anomalies, and potential factors influencing stock performance, laying the foundation for subsequent modeling and prediction tasks.
* Through EDA, we can gain a deeper understanding of the underlying market dynamics, which informs the design and implementation of trading strategies and predictive models.

By performing thorough data processing and exploratory data analysis of Intel's stock data from the Yahoo Finance API, we gain essential insights that guide subsequent modeling efforts. These insights not only facilitate the development of accurate predictive models but also enhance our understanding of the broader financial landscape.

**Model implementation**

In this project, we implemented two distinct trading strategies for predicting stock price movements using historical financial data of Intel Corp (INTC). The strategies were as follows:

**Strategy 1**: Next-Day Price Prediction

* This strategy involves predicting whether the next trading day's closing price will be higher or lower than the current day's closing price.
* To implement this strategy, we followed a systematic process:
* Data Acquisition: We obtained historical financial data for INTC.
* Preprocessing: The data was cleaned and formatted for analysis, ensuring consistency and quality.
* Feature Engineering: Relevant features such as price changes, volume, and technical indicators were selected or created to capture important information for prediction.
* Label Generation: We defined the target variable, classifying instances as either "up" or "down" based on the next day's price movement.
* Model Selection: Several machine learning classifiers, including K-Nearest Neighbors (KNN), Random Forest Classifier (RF), Gradient Boosting Classifier (GB), Support Vector Machines (SVMs), XGBoost Classifier, and Long Short-Term Memory (LSTM), were implemented to predict price movements.
* Model Evaluation: The performance of each model was assessed using metrics such as accuracy, precision, recall, and F1-score on a test dataset.

2. **Strategy 2**: Moving Averages

* This strategy involves utilizing the 50-day moving average vs. the 200-day moving average to identify bullish and bearish signals.
* Similar to Strategy 1, we followed a systematic process to implement and evaluate this strategy, including data preprocessing, feature engineering, label generation, model selection, and evaluation.

**Fine Tuning:**

In our pursuit of optimizing predictive performance, we implemented fine-tuning techniques across all classifiers to refine their respective models. Fine-tuning involves systematically adjusting hyperparameters and model configurations to enhance predictive accuracy and robustness. For each classifier utilized in our analysis, including K-Nearest Neighbors (KNN), Random Forest, Gradient Boosting, Support Vector Machines (SVM), and XGBoost, we conducted a thorough exploration of hyperparameters such as learning rate, maximum depth, number of estimators, and regularization parameters. Through techniques such as grid search and random search, we systematically searched through the hyperparameter space to identify the optimal combination that maximizes model performance. Additionally, we employed cross-validation to validate the generalization ability of the models and mitigate overfitting. By iteratively fine-tuning the models and evaluating their performance on validation data, we were able to achieve improved predictive accuracy and reliability across all classifiers. This meticulous fine-tuning process underscores our commitment to developing robust and effective machine learning models for stock market prediction, ultimately enhancing the efficacy of algorithmic trading strategies.

**Model Performance Evaluation:**

* For both strategies, we evaluated the performance of each classifier and LSTM model based on their ability to predict price movements accurately.
* The classifiers were trained on historical data and tested on a separate dataset to assess their predictive performance.
* Evaluation metrics such as accuracy, precision, recall, and F1-score were calculated to measure the effectiveness of each model in predicting price movements.

**Role of Classifiers and LSTM:**

* Classifiers such as KNN, RF, GB, SVMs, and XGBoost offer diverse methodologies for modeling and predicting stock price movements.
* Each classifier has its strengths and weaknesses, contributing to the overall predictive performance of the models.
* LSTM, as a specialized architecture for sequential data, offers unique advantages in capturing temporal dependencies and patterns in stock market data.
* By incorporating LSTM alongside traditional classifiers, we leverage its ability to capture long-term dependencies and sequential patterns, enhancing the predictive capabilities of the models.
* Additionally, by experimenting with different model architectures, such as single-layer and multi-layer LSTM, we can explore improvements in predictive performance. Adding more layers to LSTM can enable the model to capture more complex relationships in the data, potentially leading to improved results.

Through the implementation and evaluation of various trading strategies and machine learning models, this project provides insights into predicting stock price movements using historical financial data. By leveraging a combination of classifiers and LSTM, we demonstrate the potential for improving predictive accuracy and robustness in algorithmic trading applications. Further research and experimentation, including the exploration of different model architectures and hyperparameter tuning, can contribute to enhancing the effectiveness of machine learning-based trading strategies in real-world financial markets.

# Results

|  |  |  |
| --- | --- | --- |
| ​ **Model** | **Strategy 1 (Next-Day Price Prediction)**​ | **Strategy 2 (Moving Averages)**​ |
| KNN​ | 0.55​ | 0.98​ |
| Random Forest​ | 0.51​ | 0.99​ |
| Gradient Boosting​ | 0.52​ | 0.99​ |
| SVM​ | 0.50​ | 0.97​ |
| XGBoost​ | 0.66​ | 0.92​ |

The results of our analysis reveal notable differences in the performance of the two trading strategies and various machine learning models. For Strategy 1, focusing on next-day price prediction, the Random Forest model emerges as the most accurate, with an accuracy score of 0.51. In contrast, Strategy 2, utilizing moving averages, demonstrates significantly higher accuracy across all models, with Random Forest, Gradient Boosting, and SVM achieving accuracy scores of 0.99. Interestingly, the K-Nearest Neighbors (KNN) model exhibits the highest accuracy for Strategy 2, with a remarkable score of 0.98. These results underscore the efficacy of utilizing moving averages as a trading signal, showcasing its potential to outperform traditional next-day price prediction approaches. Furthermore, the superiority of certain models, such as Random Forest and Gradient Boosting, highlights the importance of selecting appropriate algorithms tailored to specific trading strategies, ultimately influencing the effectiveness of algorithmic trading systems.

**Why is Strategy 2 more effective:**

Strategy 2 focuses on bigger picture of how stock prices move over time, rather than getting caught in day-day ups and downs - moving average method which smooths out the noicy fluctuations in stock prices. We don't see every tiny wiggle up and down, we rather see a smooth line - overall trend, this is the most popular algorithm amongst the traders. It's simple to understand and apply. and works well in different market situations  Instead of trying to predict exactly what will happen with stock prices every day, which can be really unpredictable, Strategy 2 gives a more steady and reliable way to make decisions about trading.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ALGORITHM​** | **RMSE​** | **MAE​** | **MSE​** | **Sharpe’s Ratio​** |
| K-Nearest Neighbors (KNN)​ | 0.2459​ | 0.0302​ | 0.0605​ | 0.0195​ |
| Random Forest Classifier (RF)​ | 0.2008​ | 0.0202​ | 0.0403​ | 0.0185​ |
| Gradient Boosting Classifier (GB)​ | 0.2008​ | 0.0202​ | 0.0403​ | 0.0662​ |
| Support Vector Machines (SVMs)​ | 0.3619​ | 0.0655​ | 0.1310​ | 0.2300​ |
| XGBoost Classifier​ | 0.1420​ | 0.0202​ | 0.0202​ | 0.0455​ |
| LSTM ​ | 0.56055​ | 0.01461​ | 0.1568​ | -0.0336 |

Based on the above results for various machine learning models, here's a summary of the performance:

1. K-Nearest Neighbors (KNN):

* Achieves moderate performance with RMSE of 0.2459 and relatively low MAE and MSE.
* However, the Sharpe's Ratio is quite low at 0.0195, indicating a low risk-adjusted return.

1. Random Forest Classifier (RF):

* Shows slightly better performance compared to KNN with lower RMSE and MSE.
* The Sharpe's Ratio is still low at 0.0185, indicating a suboptimal risk-adjusted return.

1. Gradient Boosting Classifier (GB):

* Similar performance to RF with slightly better Sharpe's Ratio at 0.0662.
* However, the overall performance remains moderate.

1. Support Vector Machines (SVMs):

* Demonstrates the highest RMSE among all models, indicating relatively poorer performance in terms of prediction accuracy.
* Despite the high RMSE, SVMs achieve the highest Sharpe's Ratio at 0.2300, suggesting a better risk-adjusted return compared to other models.

1. XGBoost Classifier:

* Shows the lowest RMSE among all models, indicating the best prediction accuracy.
* The Sharpe's Ratio is moderate at 0.0455, suggesting a decent risk-adjusted return.

1. LSTM\*\*:

* Yields the highest RMSE among all models, indicating relatively poor prediction accuracy.
* The Sharpe's Ratio is negative (-0.03363026), suggesting a negative risk-adjusted return, which may imply worse performance compared to a risk-free asset.

In summary, SVMs exhibit the best risk-adjusted return with the highest Sharpe's Ratio, while XGBoost Classifier shows the best prediction accuracy among the traditional machine learning models. However, the LSTM model performs poorly both in terms of prediction accuracy (highest RMSE) and risk-adjusted return (negative Sharpe's Ratio), indicating that it may not be suitable for this particular prediction task.

# Conclusion

SVMs, or Support Vector Machines, exhibit the best risk-adjusted return with the highest Sharpe's Ratio because they are effective at finding the optimal hyperplane that maximizes the margin between different classes in the feature space. This capability often leads to better generalization and robustness of the model, resulting in higher predictive accuracy on unseen data. Additionally, SVMs inherently handle outliers well and can capture complex relationships in the data, making them suitable for a wide range of applications.

On the other hand, the XGBoost Classifier shows the best prediction accuracy among traditional machine learning models because it is an ensemble learning method that combines the predictions from multiple decision trees. XGBoost is known for its efficiency, scalability, and effectiveness in handling both regression and classification tasks. By iteratively training weak learners and focusing on correcting errors, XGBoost tends to produce highly accurate predictions.

However, the LSTM model performs poorly in terms of prediction accuracy and risk-adjusted return for several reasons. LSTMs, or Long Short-Term Memory networks, are a type of recurrent neural network (RNN) architecture commonly used for sequence prediction tasks. While LSTMs are effective for capturing temporal dependencies and patterns in sequential data, they may not perform optimally for all types of data or prediction tasks. In this case, the LSTM model may not have been well-suited for the specific characteristics of the dataset or the nature of the prediction task, leading to suboptimal performance in both prediction accuracy (as indicated by the highest RMSE) and risk-adjusted return (as indicated by the negative Sharpe's Ratio). Additionally, LSTMs require careful tuning of hyperparameters and may suffer from issues such as vanishing gradients or overfitting, which could further contribute to their poor performance in this scenario.

# Analysis:

In this section, we present a comprehensive analysis of the dataset through a series of visualizations. These visual representations aim to provide a deeper understanding of key attributes such as Open, High, Low, Close, Adj Close, and Volume . Each visualization contributes to unraveling patterns and trends within the dataset, offering valuable insights for decision-making in the restaurant industry. Let's delve into each visualization to extract meaningful observations and implications.

**High and Low stock prices:**

***A graph showing the price of a stock market

Description automatically generated***

* This plot illustrates the high and low stock prices from 2015 to early 2024, depicted by the orange and blue lines, respectively.​
* The graph reveals significant fluctuations, with a notable peak in 2018 and a sharp decline thereafter. Post-2018, the stock exhibits increased volatility, showing multiple peaks and troughs, and a downward trend is observable from 2022 into early 2024.
* This visual analysis helps identify periods of high volatility and overall market trends, essential for understanding the stock's performance over these years.​

**Open and Close stock prices**

**A graph showing a line graph

Description automatically generated with medium confidence**

* This plot represents the opening and closing prices of Intel stock from 2015 to early 2024, depicted by orange and blue lines respectively.​
* It highlights the price volatility over several years, showing periods where the stock opened higher than it closed (or vice versa), indicating daily trading dynamics.
* Notably, the plot captures significant fluctuations, with both lines peaking around 2018 and then showing a downward trend post-2020 into 2023, illustrating a decrease in stock value in more recent years.​

​

**Volume Traded Over Time**

**A graph of blue lines

Description automatically generated**

* The plot illustrates fluctuations in trading volume, with several prominent spikes indicating days of unusually high trading activity. ​
* These spikes could correspond to specific market events or news releases impacting the stock. The overall pattern provides insights into trading interest and liquidity over time for the stock represented.​

**Correlation Matrix**

**A screenshot of a graph

Description automatically generated**

* Open price has a very strong positive correlation with high, low, close, and adjusted close prices. This means that if the opening price of a stock is high, then the high, low, and close prices are also likely to be high.
* Volume has a weak negative correlation with open, high, low, and close prices. This means that there is not a strong relationship between volume and price. However, there is a slight tendency for lower volume to be associated with higher prices.

**Close Price vs 30 Day Moving Average**

**A graph of a line graph

Description automatically generated with medium confidence**

* The closing price appears to be generally trending upwards over time. There are some periods where the price dips but overall, there is an upward trajectory.
* The 30-day moving average is a line that smooths out the fluctuations in the closing price. It shows the average closing price over the past 30 days.
* In the graph, the 30-day moving average is currently below the closing price. This suggests that the closing price has been increasing more quickly than the average over the past 30 days. This could be a sign that the uptrend is accelerating.

**Daily Returns Percentage**

**A blue sound wave graph

Description automatically generated**

* The y-axis starts at -15% and goes up to 20%. This suggests that there has been some variability in the daily returns, with some days experiencing losses (negative returns) and other days experiencing gains (positive returns).
* The scale of the y-axis is not uniform. The space between 0% and 5% is much larger than the space between 15% and 20%. This makes it difficult to precisely judge the magnitude of the changes in daily return, especially for higher return values.

**50 Day vs 200 Day Moving Average**

**A graph of a line graph with numbers and text

Description automatically generated with medium confidence**

* **Long-term trend:** The general trend in Intel stock price appears to be **upward** over the period shown in the graph, though with fluctuations. This is because the price line (blue) is generally increasing over time.
* **Short-term trend:** It's difficult to say for sure what the short-term trend is based on this graph alone. However, we can see that the price is currently below the 50-day moving average (yellow line), which suggests that the price has been **decreasing** over the past 50 days.

**True vs Predicted values**

**A graph with numbers and lines

Description automatically generated**

# Future Works

For future extensions, implementing LSTM models could involve exploring more sophisticated architectures, experimenting with different hyperparameters, and incorporating additional features or data sources to enhance prediction accuracy and robustness. Additionally, techniques such as attention mechanisms or advanced regularization methods could be employed to improve the performance of LSTM models further.

Regarding the implementation of a critique agent in stock prediction, future work could focus on developing a feedback mechanism that evaluates the performance of the prediction models in real-time and provides constructive feedback. This critique agent could leverage techniques from reinforcement learning or online learning to adaptively adjust the model parameters or strategies based on the feedback received. By continuously refining the prediction models based on the critique agent's insights, the overall accuracy and reliability of the stock prediction system could be enhanced, leading to more informed decision-making in financial markets.

# References

1. Masoud, Najeb MH. (2017) “The impact of stock market performance upon economic growth.” International Journal of Economics and Financial Issues 3 (4) : 788–798.
2. Murkute, Amod, and Tanuja Sarode. (2015) “Forecasting market price of stock using artificial neural network.” International Journal of Computer Applications 124 (12) : 11-15.
3. Hur, Jung, Manoj Raj, and Yohanes E. Riyanto. (2006) “Finance and trade: A cross-country empirical analysis on the impact of financial development and asset tangibility on international trade.” World Development 34 (10) : 1728-1741.
4. Li, Lei, Yabin Wu, Yihang Ou, Qi Li, Yanquan Zhou, and Daoxin Chen. (2017) “Research on machine learning algorithms and feature extraction for time series.” IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC): 1-5.
5. Seber, George AF and Lee, Alan J. (2012) “Linear regression analysis.” John Wiley & Sons 329
6. Reichek, Nathaniel, and Richard B. Devereux. (1982) “Reliable estimation of peak left ventricular systolic pressure by M-mode echographicdetermined end-diastolic relative wall thickness: identification of severe valvular aortic stenosis in adult patients.” American heart journal 103 (2) : 202-209.
7. Chong, Terence Tai-Leung, and Wing-Kam Ng. (2008) “Technical analysis and the London stock exchange: testing the MACD and RSI rules using the FT30.” Applied Economics Letters 15 (14) : 1111-1114.
8. Zhang, G. Peter. (2003) “Time series forecasting using a hybrid ARIMA and neural network mode.” Neurocomputing 50 : 159-175.
9. Suykens, Johan AK, and Joos Vandewalle. (1999) “Least squares support vector machine classifiers.” Neural processing letters 9 (3) : 293-300.
10. Liaw, Andy, and Matthew Wiener. (2002) “Classification and regression by Random Forest.” R news 2 (3) : 18-22.
11. Oyeyemi, Elijah O., Lee-Anne McKinnell, and Allon WV Poole. (2007) “Neural network-based prediction techniques for global modeling of M (3000) F2 ionospheric parameter.” Advances in Space Research 39 (5) : 643-650. 606 Mehar Vijh et al. / Procedia Computer Science 167 (2020) 599–606 8 Author / Procedia Computer Science 00 (2019) 000–000
12. Huang, Wei, Yoshiteru Nakamori, and Shou-Yang Wang. (2005) “Forecasting stock market movement direction with support vector machine.” Computers & Operations Research 32 (10) : 2513-2522.
13. Selvin, Sreelekshmy, R. Vinayakumar, E. A. Gopalakrishnan, Vijay Krishna Menon, and K. P. Soman. (2017) “Stock price prediction using LSTM, RNN and CNN-sliding window mode.” International Conference on Advances in Computing, Communications and Informatics (ICACCI): 1643-1647.
14. Hamzac¸ebi, Cos¸kun, Diyar Akay, and Fevzi Kutay. (2009) “Comparison of direct and iterative artificial neural network forecast approaches in multi-periodic time series forecasting.” Expert Systems with Applications 36 (2) : 3839-3844.
15. Rout, Ajit Kumar, P. K. Dash, Rajashree Dash, and Ranjeeta Bisoi. (2017) “Forecasting financial time series using a low complexity recurrent neural network and evolutionary learning approach.” Journal of King Saud University-Computer and Information Sciences 29 (4) : 536-552.
16. Yetis, Yunus, Halid Kaplan, and Mo Jamshidi. (2014) “Stock market prediction by using artificial neural network.” in2014 World Automation Congress (WAC): 718-722.
17. Roman, Jovina, and Akhtar Jameel. (1996) “Backpropagation and recurrent neural networks in financial analysis of multiple stock market return.” Proceedings of HICSS-29: 29th Hawaii International Conference on System Sciences 2: 454-460.
18. Mizuno, Hirotaka, Michitaka Kosaka, Hiroshi Yajima, and Norihisa Komoda. (1998) “Application of neural network to technical analysis of stock market prediction.” Studies in Informatic and control 7 (3) : 111-120.
19. Kumar, Manish, and M. Thenmozhi. (2006) “Forecasting stock index movement: A comparison of support vector machines and random forest” In Indian institute of capital markets 9th capital markets conference paper
20. Mei, Jie, Dawei He, Ronald Harley, Thomas Habetler, and Guannan Qu. (2014) “A random forest method for real-time price forecasting in New York electricity market.” IEEE PES General Meeting Conference & Exposition: 1-5.
21. Herrera, Manuel, Lu´ıs Torgo, Joaqu´ın Izquierdo, and Rafael Perez-Garc ´ ´ıa. (2010) “Predictive models for forecasting hourly urban water demand.” Journal of hydrology 387 (1-2) : 141-150. [22] Dicle, Mehmet F., and John Levendis. (2011) “Importing financial data.” The Stata Journal 14 (4) : 620-626.
22. Yahoo Finance - Business Finance Stock Market News, [ Accessed on August 16,2018] [24] Zurada, Jacek M. (1992) “Introduction to artificial neural systems.” St. Paul: West publishing company 8.