Educational Administration: Theory and Practice
2024, 30(5), 13637-13646
ISSN: 2148-2403
https://kuey.net/
Research Article

Foresight In Finance: Elevating Predictions With Enhanced Rnn-Lstm And Adam Optimizer

P. Dineshkumar[1*], Dr. B. Subramani[2]

[1]Research Scholar, Snmv College of Arts & Science, Coimbatore, Tamilnadu, India.
[2]Principal, Snmv College of Arts & Science, Coimbatore, Tamilnadu, India.

Citation: P. Dineshkumar (2024) Foresight In Finance: Elevating Predictions With Enhanced Rnn-Lstm And Adam Optimizer, *Educational Administration: Theory and Practice, 30(5),* 13637-13646
Doi: 10.53555/kuey.v30i5.5931

ARTICLE INFO

ABSTRACT

In the ever-evolving landscape of financial markets, accurate stock price predictions play a pivotal role in informed decision-making. This paper explores the application of advanced deep learning techniques to enhance the precision of future stock data predictions. Specifically, research focus on leveraging the power of Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) architectures, bolstered by the adaptive learning capabilities of the Adam optimizer. The evaluation metrics include Mean Absolute Error and Root Mean Squared Error, providing insights into the model's accuracy and robustness. The findings of this paper contribute to the ongoing discourse on the application of deep learning in finance, offering a promising avenue for refining stock price predictions. As financial markets continue to demand sophisticated forecasting methodologies, the Elevated RNN-LSTM with Adam Optimizer emerges as a valuable tool for stakeholders seeking foresight and precision in their financial decision-making processes.

Keywords: Stock Market, deep learning, Prediction, Financial Analytics, Recurrent Neural Networks, Long Short-Term Memory and adaptive learning;

1. Introduction

Establishing a competitive edge in the fast-paced and dynamic world of financial markets requires making use of cutting-edge technologies. Data mining is one such technical advancement that has had a big impact on the world of financial decision-making. When applied to stock market data, this intricate process which entails the extraction of insightful patterns and knowledge from enormous datasets becomes an effective tool for prediction and categorization. Because of its inherent volatility and complexity, the stock market is difficult for both experts and investors to navigate. The incorporation of data mining tools has become essential for navigating this complex terrain, including insightful information on market trends, risk assessment, and predictive modeling. A wide range of factors are included in the enormous and continuously growing field of stock market data, such as historical stock prices, trading volumes, economic indicators, and other financial measures.
Advanced computational approaches are required since it would be extremely difficult, if not impossible, to analyze this enormous amount of data manually. As a subfield of artificial intelligence, data mining makes use of statistical models and algorithms to sort through this massive amount of data and find hidden patterns and relationships that would not be seen using more conventional analytical techniques. Financial experts can use this approach to make well-informed decisions based on past trends and new patterns, which makes it very useful for classification and prediction. When it comes to stock market data, classification is the process of grouping stocks or other financial instruments according to certain standards. By finding assets with comparable traits or behaviors, this can help investors make more informed strategic decisions. Classifying stocks according to industry, market capitalization, past performance, and other pertinent attributes is a crucial function of data mining algorithms like decision trees, support vector machines, and neural networks.

1.1 Prediction and Classification

For stock market classification or prediction, apply the trained model to new data. Using algorithms like Random Forests, Decision Trees, or Neural Networks, the model assesses unknown market variables and provides predictions based on past trends. To help with decision-making, this procedure entails using the model to evaluate possible future stock movements. Regularly updating and fine-tuning the model with the most recent data is essential to improve prediction accuracy. Keep in mind that making investment decisions based on these models should only be done after consulting a financial advisor, as market predictions are inherently risky.

2. Literature Survey

2.1 Stock Market Random Forest-Text Mining (SMRFTM)

M. N. Elagamy (2018) et.al proposed Stock market random forest-text mining system mining critical indicators of stock market movements. This research highlights the critical function that stock markets (SM) play in a free market economy by focusing on how it directly affect trade and the expansion of industries. The research uses text mining and the Random Forest algorithm to extract crucial indicators and categorize linked news stories, addressing the demand for early warning indications for stock markets crises. This innovative technique increases the number of crucial indicator classes from three to eight, in contrast to conventional data mining techniques. According to the research, Random Forest outperforms other classifiers and demonstrates its excellent classification accuracy when it comes to news item classification using bigram characteristics. The suggested method might be expanded to examine financial news from several sources, improving the identification of crucial indications for better stock market movement prediction.

2.2 Random Forest Model (RFM)

S. S. Maini (2017) et.al proposed Stock market prediction using data mining techniques. In this research, a unique approach to machine learning models specifically, Random Forest and Support Vector Machine for stock market trend prediction is presented. The research uses these models, which acknowledge the complicated and volatile character of the stock market, to predict, based on previous data, whether the price of a stock will rise above its current level. Prediction accuracy is improved by using natural language processing to analyze news articles and other pertinent sources. Using these models, the research uses the Dow Jones Industrial Average Index, which is controlled by S&P Global, to positively impact traders' decisions about instrument trades. The models' remarkable performance in forecasting the direction of the stock index is demonstrated by the results, which offer investors important information.

2.3 Support Vector Machine (SVM)

Z. Hu (2013) et.al proposed Stocks market prediction using Support Vector Machine. The present research investigates the shortcomings of conventional predictive regression models in out-of-sample predictability tests and proposes the Support Vector Machine (SVM) method as a solution to these issues. Six macroeconomic and four company-specific parameters are used by the SVM to illustrate how well it can analyze the intricate, nonlinear dynamics present in the financial sector. The findings demonstrate the effectiveness of SVM as a stock market trend prediction tool. The research highlights the significance of data-intensive, noise-resistant models for investment decisions while acknowledging the complexities of financial forecasting. Although the authors express pleasure with the classifier's high prediction percentage for a variety of stocks, it suggest improving it even more by adding more variables, particularly those that represent factors other than profitability. It emphasizes how multivariate statistics may be used to examine a company's stock market performance in relation to its past financial success as well as the state of the global economy.

2.4 Spatio-Temporal Hyper-Graph Convolution Network (STHGCN)

R. Sawhney (2020) et.al proposed Spatio-temporal Hyper-graph Convolution Network for Stock Movement Forecasting. Predicting stock movements is essential for quantitative trading and financial decision-making in the fields of computer science and finance. The market is quite volatile and stochastic, which presents a difficult problem. Promising developments have been made recently in neural stock forecasting, especially with deep learning models. But inter-stock connections are frequently oversimplified in current studies. For time-aware capture of inter-stock relations and stock price evolution, STHGCN uses a gated temporal convolution over hyper graphs. STHGCN outperforms the state-of-the-art techniques and shows promise for algorithmic trading with enhanced prediction performance, higher profit generation, and significant model latency reduction.

2.5 Stock Sequence Array Convolution Long–Short-Term Memory (SACLSTM)

Wu JM (2023) et.al proposed a graph-based CNN-LSTM stock price prediction algorithm with leading indicators. The management of investment wealth, which includes stocks, bonds, real estate, and other financial instruments, is becoming more and more common in today's society. In order to anticipate stock prices, this research presents a novel method called the Stock Sequence Array Convolutional LSTM (SACLSTM), which combines Long-Short-Term Memory Neural Network (LSTM) with Convolutional Neural Network (CNN). The model use a sequence array of historical data as input, employing convolutional layers and pooling to extract features, which are then fed into an LSTM for prediction. Using Taiwanese and American stock data, SACLSTM performs better than earlier techniques. By combining news, options, past data, and futures, the model improves upon conventional techniques in terms of prediction. The framework uses long-short-term memory units and convolution to effectively advance stock price prediction.

3. Proposed Methodology

In the dynamic landscape of stock markets, effective classification and prediction methodologies are crucial for informed investment decisions. This proposed methodology leverages the power of data mining techniques to navigate the intricate patterns within stock market data. The primary objective is to develop a robust framework that integrates advanced classification algorithms and prediction models, aiming to enhance the accuracy of forecasting future market movements. By employing sophisticated data mining methods, this methodology seeks to extract meaningful insights from historical stock data, exploring patterns, trends, and key indicators that contribute to predictive accuracy. The fusion of classification and prediction techniques aims to provide investors and financial analysts with a valuable tool for making informed and timely decisions in the volatile realm of stock markets. This methodology aspires to contribute to the evolution of sophisticated and reliable tools for investment wealth management in today's dynamic financial environment.

3.1 Recurrent Neural Networks (RNNs)

For classification and prediction tasks, Recurrent Neural Networks (RNNs) are an effective way to analyse data from the stock market. RNNs, with their ability to capture sequential dependencies, can model time-series dynamics in stock prices. Trained on historical data, it learns patterns and trends, aiding in predicting market movements. By incorporating past information, RNNs enhance classification accuracy and enable insightful predictions, making them valuable tools for leveraging temporal dependencies in financial datasets. However, the inherent unpredictability of markets necessitates cautious interpretation of results, acknowledging the dynamic nature of financial landscapes.

RNNs are a type of neural network architecture that is well-suited for sequence data. The equation for a basic RNN cell can be expressed as follows:

$$h_t = f(W_{ih} \cdot x_t + b_{ih} + W_{hh} \cdot h_{t-1} + b_{hh})$$

- ht is the hidden state or output at time t
- x_t is the input at time t
- W_{ih} and W_{hh} are the weight matrices for the input and hidden state, respectively
- b_{ih} and b_{hh} are the bias terms
- f is the activation function

The RNN equations express how information is passed through time, allowing the network to capture sequential dependencies in the data. In the context of stock price prediction, you would train an RNN on historical stock prices, and the network would learn to capture patterns and dependencies in the sequential data to make predictions for future stock prices.

3.2 Proposed Enhanced RNN-LSTM and Adam Optimizer for Stock Market Prediction

The proposed methodology for utilizing data mining and recurrent neural networks (RNNs) in stock market data analysis for classification and prediction involves several key steps. Firstly, historical stock market data is collected and preprocessed to ensure consistency and reliability. For feature extraction, relevant financial indicators and technical indicators are selected to capture market dynamics. These features are used to train the RNN model, which is chosen for its ability to recognize temporal dependencies in sequential data. The model is configured with appropriate architecture and hyperparameters, and training is performed on a portion of the dataset.

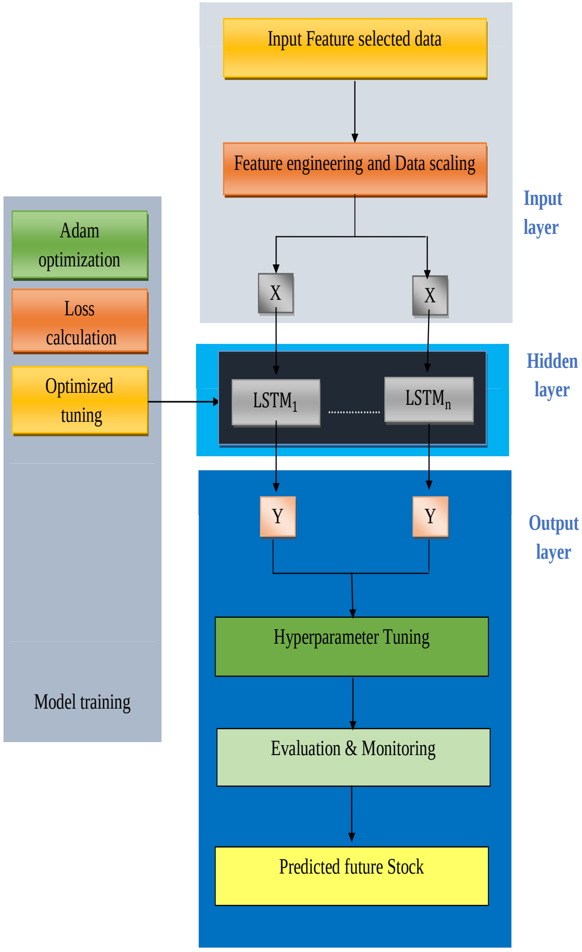

Figure 1.Proposed Work flow Diagram

Enhancing Recurrent Neural Networks (RNNs) for stock market prediction often involves incorporating optimization algorithms to improve training efficiency and convergence. One popular optimization algorithm is the Adam optimizer.

3.2.1 Choose an Advanced RNN Architecture:

Begin by using a more advanced RNN architecture such as Long Short-Term Memory (LSTM) or Gated Recurrent Unit (GRU). The proposed RNN with LSTM unit is designed to predict the forecasting demand of auto company spare parts dataset. The RNN are the NN which use feedback connections to remember the prior time steps. For instance, the value at prior time 1 is computed as,

$$\boldsymbol{h_{j,t}^{m}} = \boldsymbol{w_1}\emptyset\left(\boldsymbol{h_{j,t-1}^{m}}\right) + \boldsymbol{w_2}\emptyset\left(\boldsymbol{h_{j,t}^{m-1}}\right)$$

w_1 , w_2 are the weights and $h_{j,t}^{m}$ denotes output of activation function of node j in layer m of time step t. The values of the node output in a given time step links with the earlier time steps by adopting back propagation via time algorithm. Moreover, during long intervals the gradients of those values cease to exist. This makes the long term process challenging. These complications are resolved by the entry of LSTM. The function of LSTM, it adaptively measures the input, then remember or forgets the transient cell state value and finally activate the output function. The Fig. 2 illustrates the architecture of the proposed RNN/LSTM model to forecast the Stock market Prediction. Here, the circles in layer 1 represent the inputs to the model. In layers 2 &3, each circle represent an output vector at a given time step after passing through an LSTM activation function. The layer 4 gives scalar values at each time step which follows linear combination and concatenation. The layer 3 is MLP and Layer 5 is the hidden layer. The final layer 6 is the output layer.

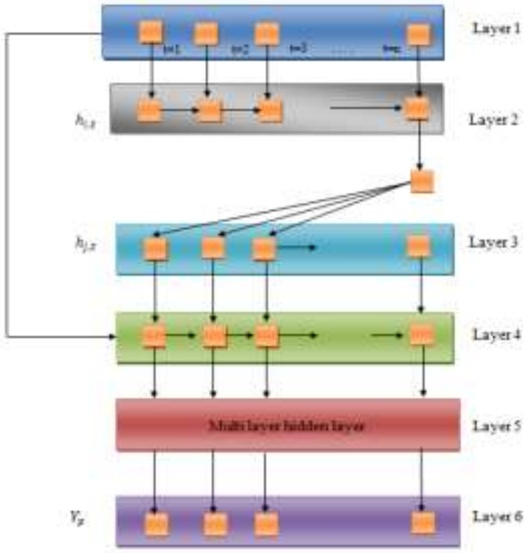

Figure 2.Architecture RNN/LSTM model

The function of LSTM is as follows: initially, it scales the input then forgets or remembers the transient cell state and finally, measures the activation function output. The mathematical formulation of LSTM is,

$$\boldsymbol{c\tilde{t}} = \boldsymbol{tanh}(\boldsymbol{W_c x_t} + \boldsymbol{U_c h_{t-1}} + \boldsymbol{b_c})$$
$$\boldsymbol{C_t} = \boldsymbol{f_t} * \boldsymbol{C_{t-1}} + \boldsymbol{i_t} * \boldsymbol{c\tilde{t}}$$
$$\boldsymbol{h_t} = \boldsymbol{0_t} * \boldsymbol{tanh}(\boldsymbol{c^t})$$

Where $c_t$the cell state at timestamp t is , $c\tilde{t}$ is the candidate for cell state at timestamp t. $i_t$denotes the input gate, $f_t$resembles the forgot gate, $o_t$denotes the output gate, w_x is the weight for the gate (x) neurons, h_{t-1} resembles the previous output of the blocks of LSTM, $x_t$is the current timestamp input, and b_x is the biases of the respective gates (x).
These functions are performed using the gates input, forgot and output. Let the input passes through the LSTM cell, it separate the input gate I and activation function g.
Where σ the sigmoid is fucntion, x_t is the input to LSTM activation corresponding to a previous layer and time step t.
To evaluate the transient value of the activation function c_t(product of I and g) requires the details of forgot gate f. It is expressed as,

$$\boldsymbol{f} = \boldsymbol{\sigma}(\boldsymbol{wf_1 h_{t-1}} + \boldsymbol{wf_2 x_t} + \boldsymbol{b_f})$$

The output Gate o is calculated as,

$$\boldsymbol{o} = \boldsymbol{\sigma}(\boldsymbol{w_{o1} h_{t-1}} + \boldsymbol{w_{o2} x_t} + \boldsymbol{b_o})$$

Then the transient memory value $C_1$is,

$$\boldsymbol{C_t} = \boldsymbol{iog} + \boldsymbol{c_{t-1} of}$$

Where o denotes element wise multiplier, Here depending on the input of the LSTM functions g q, $c_{t-1}$are scaled by the functions of i, f. At last it is calculated as

$$\boldsymbol{h_t} = \boldsymbol{Oo\emptyset(C_t)}$$

Where, $h_t$scaled by O is given as

$$\boldsymbol{h_t} = \boldsymbol{Oo\psi(C_t)}$$

Then the output from layer 3 concatenated along with the original input and the final layer predicts the forecasting demand which is denoted as $Y_p$is expressed below,

$$\boldsymbol{Y_p} = \boldsymbol{MLP} \left(\sum_{j=1}^{j} \boldsymbol{w_j X'_t, ij}\right)$$

Where, X' denotes linear combination of input at each time steps.
At training stage, the weights $w = [w_{g1}, w_{g2}, w_{i1}, w_{i2}, w_{f1}, w_{f2}, w_{o1}, w_{o2},]$and the bias vectors $b = [b_g, b_i, b_f, b_o]$are generated through the back-propagation. Then these weights are optimized using modified ADAM algorithms.

3.2.2 Feature Engineering:

Feature engineering involves creating new features or modifying existing ones to improve the performance of a machine learning model. Select and engineer relevant features, including historical stock prices, technical indicators, moving averages, and external factors like economic indicators or news sentiment. The goal of feature engineering is to provide the model with more relevant and informative input features, leading to better predictive performance.

3.2.3 Normalization and Scaling:

Normalize and scale input features, including stock prices and additional features, to ensure effective learning from different scales of data. These steps help ensure that all features are on a comparable scale, preventing issues where certain features with larger magnitudes dominate others during the training process. For many machine learning algorithms, normalization and scaling contribute to better convergence and improved model performance.

3.2.4 Use the Adam Optimizer:

Implement the Adam optimizer for training the RNN. AdaGrad and RMSProp's benefits are combined in Adam, an adaptive learning rate optimization approach. By modifying the learning rates for every parameter separately, it promotes more effective and steady convergence.

The Adam optimizer is particularly beneficial for its adaptive learning rate properties, aiding in efficient and stable convergence. The Adam update rule involves computing adaptive learning rates for each parameter by combining first-order momentum and second-order scaling of gradients. The equations for updating weights w and bias b using Adam are as follows:

$$\boldsymbol{m_t = \beta_1 \cdot m_{t-1} + (1-\beta_1) \cdot \nabla J(w,b,t)}$$
$$\boldsymbol{m_t = \beta_2 \cdot v_{t-1} + (1-\beta_2) \cdot \nabla J(w,b,t)^2}$$
$$\boldsymbol{\widehat{m}_t = \frac{m_t}{1-\beta_1^t}}$$
$$\boldsymbol{\widehat{v}_t = \frac{v_t}{1-\beta_2^t}}$$
$$\boldsymbol{w_{t+1} = w_t - \alpha \cdot \frac{\widehat{m}_t}{\sqrt{\widehat{v}_t + \epsilon}}}$$
$$\boldsymbol{b_{t+1} = b_t - \alpha \cdot \frac{\widehat{m}_t}{\sqrt{\widehat{v}_t + \epsilon}}}$$

Here, t represents the time step, $\nabla J(w,b,t)$ is the gradient of the loss function with respect to weights and biases at time step t, $\beta_1$and β_2 are exponential decay rates for the first and second moments, α is the learning rate, and ϵ is a small constant to prevent division by zero, integrating the Adam optimizer into the training process prevent division by zero. Integrating the Adam optimizer into the training process enhances the RNN's ability to learn from financial time series data efficiently, especially when combined with advanced architectures, feature engineering, and regularization techniques. The model's performance should be continually monitored and fine-tuned to adapt to the dynamic nature of stock market data.

3.2.5 Model Training:

Train the RNN model using the Adam optimizer. During training, the model learns to adjust its weights and biases to minimize the loss function, which measures the difference between predicted and actual stock prices.

3.2.6 Hyperparameter Tuning:

Perform hyperparameter tuning to find the optimal configuration for the RNN model. Tune the number of layers, the number of neurons in each layer, dropout rates, and other relevant parameters.

3.2.7 Regularization Techniques:

Apply regularization techniques such as dropout to prevent overfitting. Regularization helps improve the generalization ability of the model, especially when dealing with a limited amount of financial data.

3.2.8 Evaluation and Monitoring:

Evaluate the model's performance using appropriate metrics (e.g., Mean Squared Error or Root Mean Squared Error) on a validation set. Continuously monitor the model's performance and adapt it as needed, considering the dynamic nature of stock market data.

3.2.9 Prediction:
Once satisfied with the model's performance, make predictions on the test set or future data to assess its effectiveness in predicting stock prices.

4. Experimental Results

4.1 Accuracy

Dataset	SVM	RFM	Proposed Enhanced RNN-LSTM and Adam Optimizer
100	77	84	95
200	75	87	96
300	80	70	90
400	85	83	98
500	92	76	99

Table 1.Comparison Table of Accuracy

The Comparison table 1 of Accuracy demonstrates the different values of existing SVM, RFM and Proposed Enhanced RNN-LSTM and Adam Optimizer. While comparing the Existing algorithm and Proposed Enhanced RNN-LSTM and Adam Optimizer, provides the better results. The existing algorithm values start from 75 to 92, 70 to 83 and Proposed Enhanced RNN-LSTM and Adam Optimizer values starts from 90 to 99. The proposed method provides the great results.

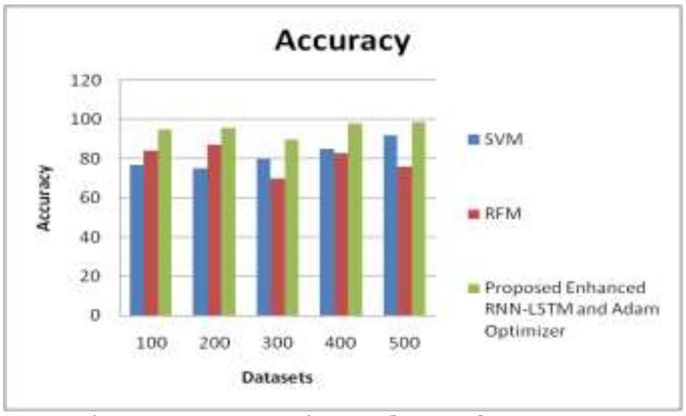

Figure 3.Comparison Chart of Accuracy

The Figure 3 Shows the comparison chart of Accuracy demonstrates the existing SVM, RFM and Proposed Enhanced RNN-LSTM and Adam Optimizer. X axis denote the Dataset and y axis denotes the Accuracy. The Proposed Enhanced RNN-LSTM and Adam Optimizer values are better than the existing algorithm. The existing algorithm values start from 75 to 92, 70 to 83 and Proposed Enhanced RNN-LSTM and Adam Optimizer values starts from 90 to 99. The proposed method provides the great results.

4.2 Precision

Dataset	SVM	RFM	Proposed Enhanced RNN-LSTM and Adam Optimizer
100	80.12	78.37	98.67
200	82.69	90.82	97.26
300	79.62	88.54	99.21
400	74.55	80.63	96.58
500	71.94	75.72	91.87

Table 2.Comparison Table of Precision

The Comparison table 2 of Precision demonstrates the different values of existing SVM, RFM and Proposed Enhanced RNN-LSTM and Adam Optimizer. While comparing the Existing algorithm and Proposed Enhanced RNN-LSTM and Adam Optimizer, provides the better results. The existing algorithm values start from 71.94 to 82.69, 75.72 to 90.82 and Proposed Enhanced RNN-LSTM and Adam Optimizer values starts from 91.87 to 99.21. The proposed method provides the great results.

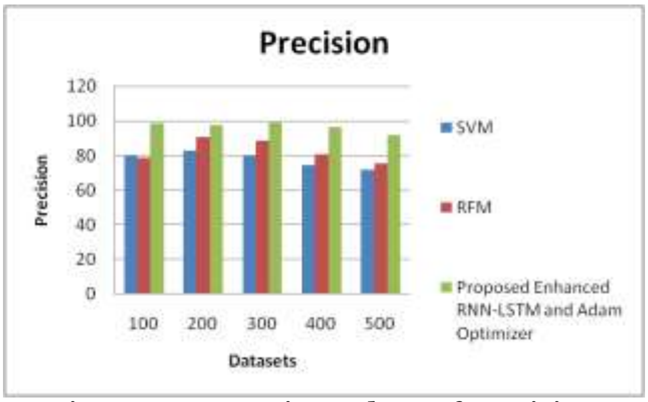

Figure 4.Comparison Chart of Precision

The Figure 4 Shows the comparison chart of Precision demonstrates the existing SVM, RFM and Proposed Enhanced RNN-LSTM and Adam Optimizer. X axis denote the Dataset and y axis denotes the Precision ratio. The Proposed Enhanced RNN-LSTM and Adam Optimizer values are better than the existing algorithm. The existing algorithm values start from 71.94 to 82.69, 75.72 to 90.82 and Proposed Enhanced RNN-LSTM and Adam Optimizer values starts from 91.87 to 99.21. The proposed method provides the great results.

4.3 Mean Absolute Error

Dataset	SVM	RFM	Proposed Enhanced RNN-LSTM and Adam Optimizer
100	0.75	0.88	0.68
200	0.77	0.95	0.70
300	0.83	0.96	0.73
400	0.86	0.97	0.75
500	0.87	0.99	0.78

Table 3.Comparison Table of Mean Absolute Error

The Comparison table 3 of Mean Absolute Error demonstrates the different values of existing SVM, RFM and Proposed Enhanced RNN-LSTM and Adam Optimizer. While comparing the Existing algorithm and Proposed Enhanced RNN-LSTM and Adam Optimizer, provides the better results. The existing algorithm values start from 0.75 to 0.87, 0.88 to 0.99 and Proposed Enhanced RNN-LSTM and Adam Optimizer values starts from 0.68 to 0.78. The proposed method provides the great results.

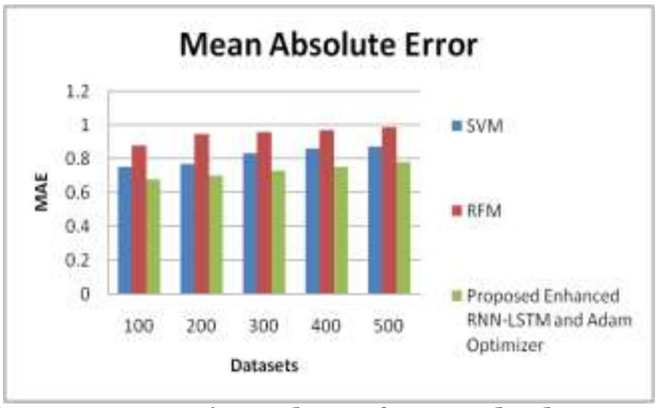

Figure 5.Comparison Chart of Mean Absolute Error

The Figure 5 Shows the comparison chart of Mean Absolute Error demonstrates the existing SVM, RFM and Proposed Enhanced RNN-LSTM and Adam Optimizer. X axis denote the Dataset and y axis denotes the MAE ratio. The Proposed Enhanced RNN-LSTM and Adam Optimizer values are better than the existing algorithm. The existing algorithm values start from 0.75 to 0.87, 0.88 to 0.99 and Proposed Enhanced RNN-LSTM and Adam Optimizer values starts from 0.68 to 0.78. The proposed method provides the great results.

4.4 Mean Square Error

Dataset	SVM	RFM	Proposed Enhanced RNN-LSTM and Adam Optimizer
100	0.93	0.98	0.84
200	0.95	0.99	0.82

300	0.88	0.90	0.75
400	0.97	0.96	0.73
500	0.86	0.94	0.74

Table 4.Comparison Table of Mean Square Error

The Comparison table 4 of Mean Square Error Values explains the different values of existing SVM, RFM and Proposed Enhanced RNN-LSTM and Adam Optimizer. While comparing the Existing algorithm and Proposed Enhanced RNN-LSTM and Adam Optimizer, provides the better results. The existing algorithm values start from 0.86 to 0.97, 0.90 to 0.99 and Proposed Enhanced RNN-LSTM and Adam Optimizer values starts from 0.73 to 0.84. The proposed method provides the great results.

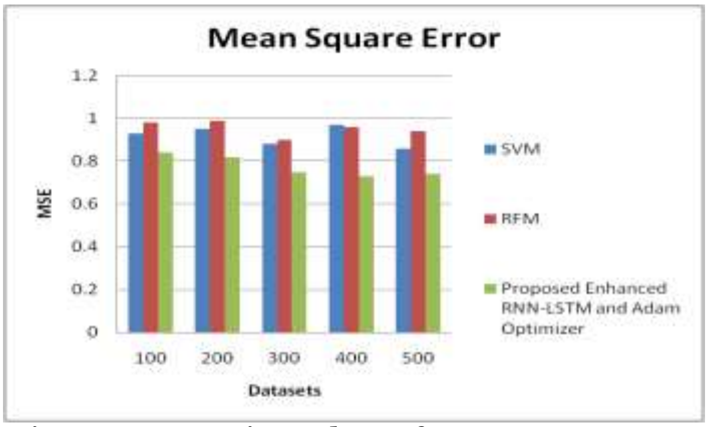

Figure 6.Comparison Chart of Mean Square Error

The Figure 6 Shows the comparison chart of Mean Square Error demonstrates the existing SVM, RFM and Proposed Enhanced RNN-LSTM and Adam Optimizer. X axis denote the Dataset and y axis denotes the MSE ratio. The Proposed Enhanced RNN-LSTM and Adam Optimizer values are better than the existing algorithm. The existing algorithm values start from 0.86 to 0.97, 0.90 to 0.99 and Proposed Enhanced RNN-LSTM and Adam Optimizer values starts from 0.73 to 0.84. The proposed method provides the great results.

5. Conclusion

In this paper, the deployment of an Enhanced Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) model, fortified by the adaptive learning prowess of the Adam optimizer, presents a compelling advancement in the domain of stock price prediction. The validation on unseen datasets gauged by performance metrics such as Mean Absolute Error and Root Mean Squared Error and underscores the model's accuracy in forecasting future stock prices. This research contributes to the evolving landscape of financial forecasting methodologies, highlighting the efficacy of advanced deep learning techniques in capturing nuanced patterns within historical stock data. The Enhanced RNN-LSTM with Adam Optimizer emerges as a promising tool for financial stakeholders seeking refined foresight and precision in decision-making. As financial markets continue to demand sophisticated predictive models, the findings of this study advocate for the adoption of this enhanced approach, paving the way for more informed and strategic decision-making processes in the unpredictable milieu of financial markets.

References

1. J. Sun, Y. Qing, C. Liu and J. Lin, "Self-FTS: A Self-Supervised Learning Method for Financial Time Series Representation in Stock Intraday Trading," 2022 IEEE 20th International Conference on Industrial Informatics (INDIN), Perth, Australia, 2022, pp. 501-506, doi: 10.1109/INDIN51773.2022.9976077.
2. K. -Y. Chen, P. -J. Lee and S. -C. Liu, "Poster:Stock Price Prediction Using Machine Learning," 2023 IEEE 43rd International Conference on Distributed Computing Systems (ICDCS), Hong Kong, Hong Kong, 2023, pp. 1067-1068, doi: 10.1109/ICDCS57875.2023.00123.
3. M. Jeyakarthic and R. Ramesh, "An Effective Stock Market Direction Using Hybrid WWO-MKELM technique," 2023 Fifth International Conference on Electrical, Computer and Communication Technologies (ICECCT), Erode, India, 2023, pp. 1-4, doi: 10.1109/ICECCT56650.2023.10179847.
4. M. N. Elagamy, C. Stanier and B. Sharp, "Stock market random forest-text mining system mining critical indicators of stock market movements," 2018 2nd International Conference on Natural Language and Speech Processing (ICNLSP), Algiers, Algeria, 2018, pp. 1-8, doi: 10.1109/ICNLSP.2018.8374370.
5. N. Korivi, K. S. Naveen, G. C. Keerthi and V. M. Manikandan, "A Novel Stock Price Prediction Scheme from Twitter Data by using Weighted Sentiment Analysis," 2022 12th International Conference on Cloud

Computing, Data Science & Engineering (Confluence), Noida, India, 2022, pp. 623-628, doi: 10.1109/Confluence52989.2022.9734139.

6. N. Poornima, D. Abilash and M. Theodaniel, "Improvising the Stock Prediction by Integrating with roBERTa and LSTM," 2023 International Conference on Signal Processing, Computation, Electronics, Power and Telecommunication (IConSCEPT), Karaikal, India, 2023, pp. 1-8, doi: 10.1109/IConSCEPT57958.2023.10169904.
7. N. W. Xiang and M. Dabbagh, "Stock Market Price Prediction: Text Analytics of the GameStop Short Squeeze," 2022 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET), Kota Kinabalu, Malaysia, 2022, pp. 1-6, doi: 10.1109/IICAIET55139.2022.9936756.
8. P. Deivendran, N. J. Kumar, R. Yashwanth, R. Raghul and D. Naresh, "Predict the Price in Stock Market Based on Heuristic Search Techniques Using Tensor Representation," 2023 International Conference on Networking and Communications (ICNWC), Chennai, India, 2023, pp. 1-6, doi: 10.1109/ICNWC57852.2023.10127251.
9. R. Sawhney, S. Agarwal, A. Wadhwa and R. R. Shah, "Spatiotemporal Hypergraph Convolution Network for Stock Movement Forecasting," 2020 IEEE International Conference on Data Mining (ICDM), Sorrento, Italy, 2020, pp. 482-491, doi: 10.1109/ICDM50108.2020.00057.
10. S. Agrawal and D. K. Sharma, "Feature Extraction and Selection Techniques for Time Series Data Classification: A Comparative Analysis," 2022 9th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 2022, pp. 860-865, doi: 10.23919/INDIACom54597.2022.9763125.