An Output Validation Scheme for Stock Market Price Prediction Using ESSM-GRU

Narasimha Rao Konangi

Senior Manager Software Engineering, Capital One

DOI: 10.29322/IJSRP.13.07.2023.p13907
http://dx.doi.org/10.29322/IJSRP.13.07.2023.p13907

Paper Received Date: 13th May 2023
Paper Acceptance Date: 22nd June 2023
Paper Publication Date: 6th July 2023

Abstract- Predicting stock prices remains an important subject of big data analytics. Although many prediction models are developed in the literature, the accurate prediction of Stock-Prices is uncertain due to the underlying problem of massive amounts of data with high response time. Hence, for accurate prediction, an ESSM-GRU-based framework is proposed in this paper. Initially, the Twitter dataset is processed to separate automated twits, pre-process the separated twits, and extract features using TF-IDF. Meantime, the attributes from the historical dataset were extracted and merged with the TF-IDF features using the CK-Means-based clustering phase. Then, the ESSM-GRU-based prediction model was trained using the clustered data. Then, to reduce response time, the Stock Market (SM) users' requests are filtered using D-LS similarity and sent to the load balancing phase using KC-MRA. Finally, the predicted results are sent to the user as a notification after satisfying the output validation criteria. Finally, when compared with the prevailing models, the proposed model attained better results.

Index Terms- Chosinedistance-based K-Means (CK-Means),
Exponential Smoothened Sign Mish based Gated Recurrent Unit (ESSM-GRU),
Damerau–Levenshtein Similarity measure (D-LS),
Kendall Correlation based Mud Ring Algorithm (KC-MRA),
Deep Learning (DL).

I. Introduction

In today's scenario, big data analytics have immense potential due to the fast-developing economy. It empowers customers' experience to help organizations in achieving a better customer journey and make decisions to improve customer experience (Holmlund et al., 2020). In the era of big data, SM prediction has concerned one of the most challenging problems because of various influential factors, such as investor sentiment, social media sentiments, and economic factors (Nabipour et al., 2020), but not limited to the company's financial reports, global economy, political conditions, performance, etc.(Valle-Cruz et al., 2022).

Stock-Price Prediction is the act of discovering the future value of company stock and other financial assets traded on an exchange (Goldstein et al., 2021). The fundamental cause behind this prediction is to gain significant profits by making informed decisions on their investments (Nti et al., 2020b)and then selling stocks that are probably to fall(Vijh et al., 2020). A poor investment could easily result in considerable losses for investors, especially if they continue to make the wrong choices(J. M. T. Wu et al., 2021). Thus, to minimize the losses and maximize the profit, a useful predictive model to predict the trend over the last few years is of significant importance (Nti et al., 2020a, Patel et al., 2021).

The previous studies use machine learning techniques, which can train over the existing data and help in predicting the next set of values (Jaggi et al., 2021,Mehta et al., 2021). Some of the authors have also considered DL for stock prediction. Although these techniques can provide more reliable results, they face difficulties in rendering accurate results as speedy updates in big data mismatch the real figures(Obthong et al., 2020,Behera et al., 2020). Hence, the development of a useful stock forecasting model that delivers a quality of interest equal to independent real-world observations remains difficult.

1.1 Problem Statement

Existing SM prediction models have the following shortcomings,

- Existing work doesn't concentrate on the output validation of mismatched real-time figures and speedy updates in bigdata.
- The prediction system trained on uncertain big data information leads to a decrease in the prediction rate.
- Improper request mapping is found to be an optimization problem as it increases the users' waiting time.

In addressing these downsides, the proposed ESSM-GRU-based prediction model comprised of the following contributions,

- To match the prediction results and with the quicker data moves in realtime, an output validation framework is presented.

- To improve the prediction rate, efficient twit separation is carried out.
- To reduce waiting time, mapping user's requests to appropriate resources is performed.

The remaining paper is organized to review existing works in section 2, describe the proposed framework in section 3, experimental results in section 4, and conclude the paper in section 5.

II. LITERATURE REVIEW

(Shen & Shafiq, 2020) developed a DL model for predicting price trends. Multiple feature engineering techniques on the pre-processed data were combined with DL-based prediction. Even if the model achieved high accuracy, the technical indices that influence the irregular term lengths were not focused.

(Khan et al., 2022) used algorithms on social media data for SM prediction. Using feature selection and spam tweet reduction steps DL was used for prediction. The results showed the highest prediction accuracies of the model. The lower quality of the result was attained due to the lack of relevant stock keywords.

(Maqsood et al., 2020) introduced stock exchange forecasting model using linear regression, Support Vector Regression, and DL. The input data were processed to calculate the sentiment of each event, which is then combined with the stock exchange data to train the prediction model. Although the performance of the model was improved, the weight adjustment process was time-consuming.

(Long et al., 2020) offered a deep neural network to predict Stock-Price. The model utilized the knowledge graph to select relevant stocks. The attention-based bidirectional LSTM was used for prediction. The results showed that the method achieved the best performance. The model efficiency was limited to the deviation for historic and real-time data.

(Liu & Long, 2020) proffered a stock closing price forecasting model. The empirical wavelet transform, Long short-term memory (LSTM), and particle swarm optimization (PSO) were the components of the framework. Results showed that the framework had the best prediction accuracy. The model had increased training time due to the low convergence rate of PSO.

(S. Wu et al., 2022) presented a Stock-Price prediction method using multiple data sources. The pre-processed data were used to calculate the investor's sentiment and the Stock-Price prediction was done using LSTM. The experiments showed that the model was better. The inability to make cycle stock predictions degraded the model performance.

(Rezaei et al., 2021) employed a hybrid algorithm i.e., CEEMD-CNN-LSTM and EMD-CNN-LSTM. The extracted deep features and time sequences were applied to one-step-ahead prediction. The practical findings confirmed that the model outperformed other counterparts. The lack of relevant pre-processing steps reduced the training efficiency of the model.

III. PROPOSED SM PRICE PREDICTION MODEL

In the processing of big data analytics, an SM price prediction model is proposed to help investors. The workflow of the proposed model is shown in Figure 1,

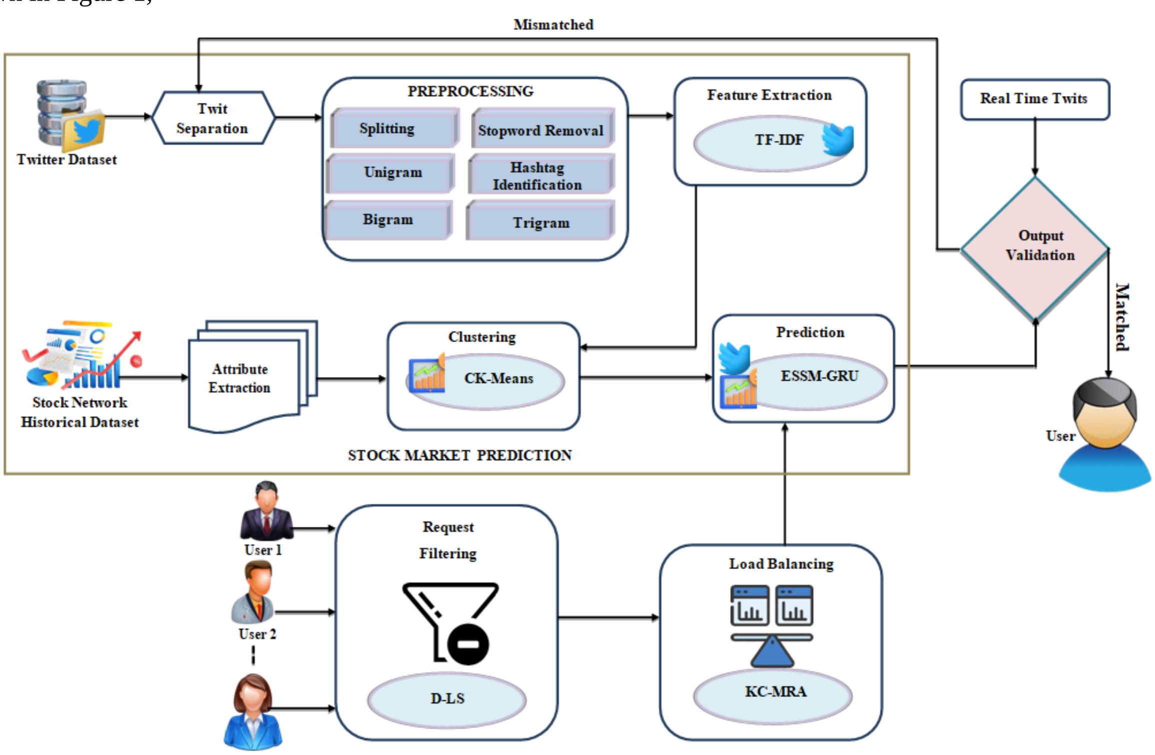


Figure 1: Block Diagram of the Proposed Methodology

3.1 SM Prediction

Two types of datasets, such as the Twitter dataset and the SM historical dataset are used to train the prediction model.

3.1.1Twitter Dataset Processing

The Twitter dataset contains Twitter tweets to identify the rise and down of share values for different companies.

Twit Separation: This step is to perform separation between the tweets $(AT_{(n)})$ that gets posted automatically and the tweets $(MT_{(n)})$ that is generated manually by an individual. The separated tweets (Tw_{sep}) are represented as,

$$Tw \xrightarrow{Separation} Tw_{sep}\left(AT_{(n)}, MT_{(n)}\right) \tag{1}$$

Where, Tw consists of the total number of tweets.

Pre-processing:Here, splitting, stop word removal, and hashtag identification are used to identify individual entities in the data, remove a set of commonly used words, and select hashtag content as keywords to match different company names.

As a result of identifying hash content $\left(\delta_{\#(n)}\right)$, the remaining word sequence (δ_n) are generated into high-precision phrases using unigram $\left(Tw_{ug}(\delta_n)\right)$, bigram $\left(Tw_{bg}(\delta_n)\right)$, and trigram $\left(Tw_{tg}(\delta_n)\right)$ functions to extract useful expressions $\left(Tw_{ug,bg,tg}\left(\delta_{(n)}\right)\right)$ from the raw data. Unigram is a single-word sequence, a bigram is a two-word sequence, and a trigram is a three-word sequence. Hence, the pre-processed data (Tw_{pre}) are generated as,

$$Tw_{pre} = \left\{ \delta_{\#(n)}, Tw_{ug,bg,tg}\left(\delta_{(n)}\right) \in \left\{ \begin{array}{ll} Tw_{ug}(\delta_n) & if\left(\tau(\delta_n)\right) = \prod_{n=1}^{N}(\delta_n) \\ Tw_{bg}(\delta_n) & if\left(\tau(\delta_n)\right) = \prod_{n=1}^{N}(\delta_n \mid \delta_{n-1}) \\ Tw_{tg}(\delta_n) & if\left(\tau(\delta_n)\right) = \prod_{n=1}^{N}(\delta_n \mid \delta_{n-2}, \delta_{n-1}) \end{array} \right. \right\} \tag{2}$$

Where, $\tau(\bullet)$ denotes the probability of (n) grams to form a sequence of words from (δ_n).

Feature Extraction: This step is to perform TF-IDF computations to gather more frequent twits of the company. TF-IDF computation (Tw_{TF-IDF}) is based on the multiplication of TF and IDF scores $(\sigma_{TF}, \sigma_{IDF})$.

$$\sigma_{TF} = \frac{\ell(\delta_n)}{n} \tag{3}$$

$$\sigma_{IDF} = \frac{Tw_{pre}}{Tw_{pre}(\delta_n)} \tag{4}$$

$$Tw_{TF-IDF} = \sigma_{TF} * \sigma_{IDF} \tag{5}$$

Where, $\ell(\bullet)$ denotes the number of times the term (δ_n) appeared in the tweet, n is the total number of terms, and $Tw_{pre}(\delta_n)$ is the number of tweets containing the term in (Tw_{pre}).

3.1.2 SM Historical Dataset Processing

Here, important attributes, such as Date Symbol, Series, Prev Close, Open, High, Low, Last, Close, Volume-Weighted Average Price (VWAP), Volume, Turnover, and Trades are extracted. The extracted attributes (κ_n) are represented as,

$$\kappa_n = \{\kappa_1, \kappa_2, \kappa_3, \ldots\ldots, \kappa_N\} \tag{6}$$

3.1.3 Clustering

Clustering is to merge the (Tw_{TF-IDF}) from the Twitter dataset to the (κ_n) extracted from the SM historical dataset which is done by the CK-Means algorithm based on the nearest values. Although K-Means is best for its speed on large datasets, it has trouble in clustering data, where the data are of varying sizes. Hence, the Chosine distance is used. The steps involved in CK-Meansare,

- Specify the number of clusters M and set of cluster centroids as $\psi_i = \{\psi_1, \psi_2, ... \psi_M\}$
- Select the number of M cluster centroids randomly to cluster feature values $(\lambda_i \in \{Tw_{TF-IDF}, \kappa_n\})$.
- Compute the ChosineDistance $C_d(\bullet)$ between the probability distributions of each data point and all centroids $(\psi(i), \lambda(i))$ to assign the data point to the closest center.

$$C_d(\psi_{(i)}, \lambda_{(i)}) = \frac{2 \times \sqrt{\sum_{i=1}^{n} \psi_{(i)}^2 - \lambda_{(i)}^2}}{\sqrt{\sum_{i=1}^{n} \psi_{(i)}^2} \sqrt{\sum_{i=1}^{n} \lambda_{(i)}^2}} \tag{7}$$

- Compute the centroid for the clusters by taking the average of all data points and repeat the above steps until the assignment of the cluster for the data point changes.

Then the clusters obtained are expressed as, $(\nabla_{cl} = \{\nabla_{cl(1)}, \nabla_{cl(2)}, \nabla_{cl(3)}, ..., \nabla_{cl(M)}\})$.

3.1.4 Prediction

Based on (∇_{cl}), predicting future values of SM using ESSM-GRUis executed in this phase. For its utilization of less memory and faster processing of time series datasets, GRU is used here. However, the GRU algorithm suffers from the problem of Exploding and Vanishing Gradients that hamper the learning of long data sequences. So, the SignMish activation function along with the Exponential Smoothing Coefficient technique is used. The architecture of ESSM-GRU is shown in Figure 2,

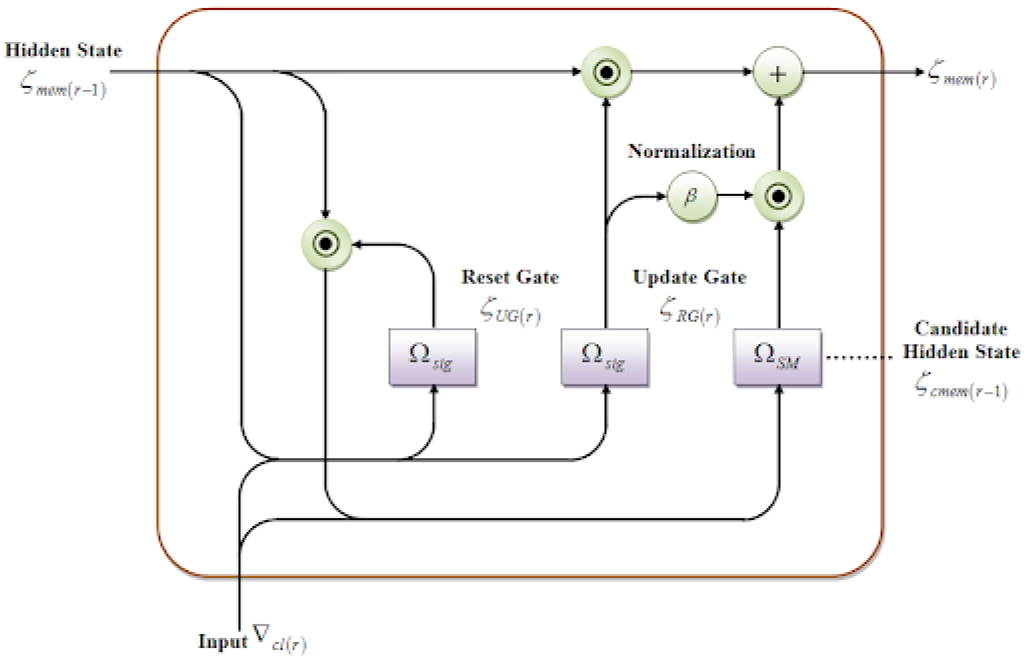


Figure 2:ESSM-GRU

Initially, the update (ξ_{UG}) and reset gates (ξ_{RG}) are determined by the current input and the previous hidden state output $(\nabla_{cl}(r), \xi_{mem}(r-1))$, which is multiplied by the weights of respective gates $(\partial_{UG}, \partial_{RG})$, added together, and applied to the sigmoid activation function (Ω_{sig}) for each time step (r).

$$\xi_{UG}(r) = \Omega_{sig}\lfloor(\nabla_{cl}(r), \xi_{mem}(r-1))\partial_{UG}\rfloor \tag{8}$$

$$\xi_{RG}(r) = \Omega_{sig} \left\lfloor \left(\nabla_{cl}(r), \xi_{mem}(r-1) \right) \partial_{RG} \right\rfloor \tag{9}$$

From the reset gate, the candidate hidden state to decide the information from the past is calculated as,

$$\xi_{cmem(r)} = \Omega_{SM} \left\lfloor \left(\xi_{RG(nor)} \nabla_{cl(r)}, \xi_{mem(r-1)} \right) \partial \right\rfloor \tag{10}$$

$$\Omega_{SM} = Y.\left(\tanh\left(\wp_{sp}\left(\frac{\sin(Y)}{Y} \right) \right) \right) \quad Where, Y = \left\lfloor \left(\xi_{RG(nor)} \nabla_{cl(r)}, \xi_{mem(r-1)} \right) \partial \right\rfloor \tag{11}$$

Hence, the new hidden state that holds the information for the current state based on the update gate and candidate hidden state is calculated as,

$$\xi_{mem(r)} = \left(1 - \xi_{nor-UG(r)}\right) \xi_{mem(r-1)} + \xi_{nor-UG(r)} \xi_{mem(r)} \tag{12}$$

$$\left(\xi_{RG(nor)}, \xi_{nor-UG} \right) \leftarrow \beta . \xi_{RG(nor)} + (1-\beta) \xi_{nor-UG} \tag{13}$$

Where, $\xi_{cmem}(r)$ is the candidate memory content, Ω_{SM} is the SM activation function, $\wp_{sp}$ is the soft plus function, $\tanh$ is the tanh function, $\left(\xi_{RG(nor)}, \xi_{nor-UG} \right)$ are the normalized outputs using the smoothing coefficient β, and $\xi_{mem}(r)$ is the output of the GRU unit.

3.2 Stock Marker User Requests

After the completion of prediction model training, SM users watching the movement of Stock-Prices in a particular company using the prediction model are initialized. While analyzing such information in the prediction model, the number of requests from various users is processed by request filtering and load balancing.

Request Filtering: In this step, multiple requests from a single user or multiple requests from multiple users for a particular company are filtered by using D-LS similarity measure $\left(dR_{(n1-1,n2)+1} \right)$.

$$\phi_{fil} = \min \begin{cases} 0 & if\ (n1, n2) = 0 \\ dR_{(n1-1,n2)+1} & if\ (n1 > 0) \\ dR_{(n1,n2-1)+1} & if\ (n2 > 0) \\ dR_{(n1-1,n2-1)+1} & if\ (n1, n2 > 0) \\ dR_{(n1-2,n2-2)+1} & if\ (n1-1, n2-1 > 1) \end{cases} \tag{14}$$

Where, ϕ_{fil} contains N number of filtered requests $\{\phi_{fil}(n)\}$.

Load Balancing: By receiving (ϕ_{fil}), load balancing is performed to reduce users waiting time. MRA is chosen for its optimization effectiveness in comparison with other meta-heuristic algorithms. It has issues in position vector updation due to the random coefficient process. So, Kendall Correlation is used in the MRA algorithm.

Initially, the population (number of filtered requests (ϕ_{fil})) begins the searching process. Dolphins justify their locations based on the best position found so far or a randomly selected dolphin at each time step.

Exploration: This mechanism is to allow a detailed search of optimum solutions by employing search agents organically in the (d) dimensional search space. The sound loudness parameter (L) and the position vector $(\phi_{fil}(i))$ based on exploration velocity (ϑ) are expressed by,

$$L = 2.\left(1 - \frac{i_{cur}}{i_{\max}} \right) (2.e - 1) \tag{15}$$

$$\phi_{fil}(i) = \phi_{fil}(i-1) + \vartheta \tag{16}$$

Where, i_{cur} is the current iteration, $i_{\max}$ is the maximum iteration, and e denotes the pulse rate changes.

Exploitation: Locating and encircling the detected prey is executed in this phase. This behaviour has been derived using the Kendal correlation function (Θ) as,

$$\phi_{fil}(i) = \phi_{fil}(i-1).\sin(2\pi\aleph) - L.\Theta \qquad (17)$$

$$\Theta = 2.e - \frac{\phi_{bst}(i-1)}{2.e} + \phi_{fil}(i-1) \qquad (18)$$

Where, $\aleph$ is the random number, and ϕ_{bst} is the best position of the dolphin updated at each time step (i).
In this way, the load balancing of filtered requests is carried out by KC-MRA, where the search process schedules responses based on the request attributes. Here, each search represents its distinct solution, as a measure of an individual's overall fitness. The fitness computation is based on the minimum response time (T_{Rs}) as,

$$\phi_{bst} = \min_{\phi_{fil}}(T_{Rs}) \qquad (19)$$

Algorithm 1: KC-MRA-based load balancing

Input: Filtered Requests (ϕ_{fil})

Output: Load Balanced Results ϕ_{bst}

Begin

Initialize population (ϕ_{fil}), velocity (ϑ), parameters (L), number of iterations $i_{\max}$
Evaluate fitness of each individual
Select best solution
While $(i < i_{\max})$
Update parameters e, L, $\aleph$
If $(L \geq 1)$
Generate new solution by $\phi_{fil}(i) = \phi_{fil}(i-1) + \vartheta$
Else
Update position based on $\phi_{fil}(i) = \phi_{fil}(i-1).\sin(2\pi\aleph) - L.\Theta$
End if
Obtain fitness functions
Update ϕ_{bst}
Set $i = i + 1$
End While
Return ϕ_{bst}
End

3.3 Output Validation

Based on (ϕ_{bst}), the scheduled responses from the prediction model are sent to the user in the form of a notification. Before sending the notification, the output from the prediction model is validated by matching with the real-time data, and processing validation results are processed are,

- If the validator result is a mismatch between the prediction result and realtimeTwitter data, the prediction model undergoes training on the real time Twitter data.
- When the match is found from the validation results, it was considered as the model actually accomplishing what it intended to accomplish. Then, the notification is sent to the user.

IV. RESULTS AND DISCUSSION

The performance evaluation of the proposed work along with prevailing works is donehere, which is implemented in the platform of PYTHON.

4.1 Database Description

The proposed framework utilized three datasets, namely financial tweets, sentiment analysis of financial tweets, and a huge SM dataset thatcontains tweets from verified users and historical daily prices.

4.2 Performance Analysis of the proposed Model

Table1 compares the proposed ESSM-GRU with existing GRU, Long Short Term Memory (LSTM), Bidirectional LSTM (BiLSTM), and Recurrent Neural Network (RNN).

Table 1: Performance Measure of ESSM-GRU

Techniques	Prediction Rate (%)
Proposed ESSM-GRU	96.1
GRU	92.8
BiLSTM	90.4
LSTM	89.2
RNN	87.7

The proposed model exhibited a 3.3% of improved prediction rate, which is higher than the existing GRU's prediction rate. Similarly, the proposed model outperformed the other existing models. This shows that the request filtering and the load balancing phase have improved the proposedmodel's performance.

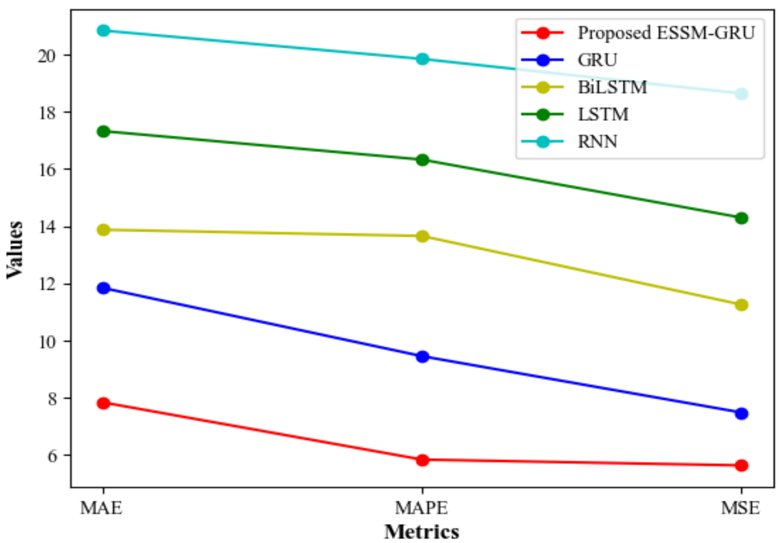


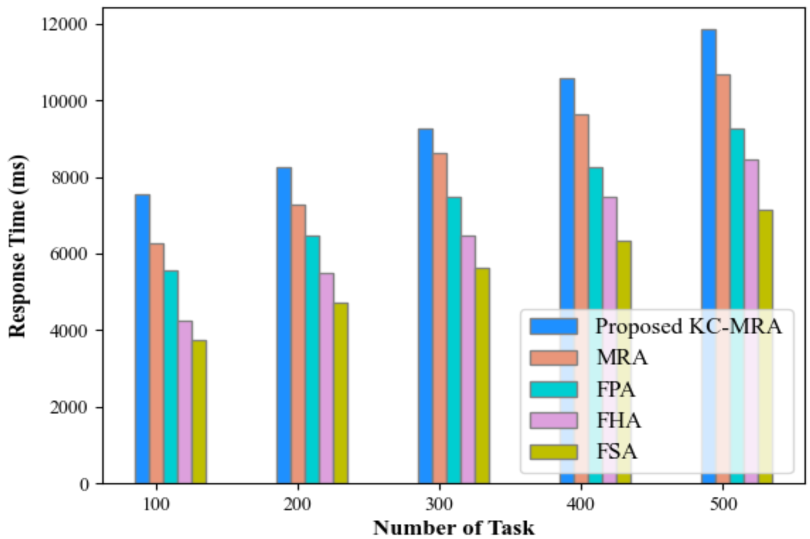
Figure 3: Prediction Error Analysis

Figure 3 unveils the Mean Absolute Error (MAE), Mean Squared Error (MSE), and Mean Absolute Percentage Error (MAPE) analysis. Here, the MAE, MSE, and MAPE of the proposed technique are 7.8457, 5.6385, and 5.8423, respectively. But, the conventional models exhibited high error. Thus, the proposedmodel has predicted the future SM with less error.

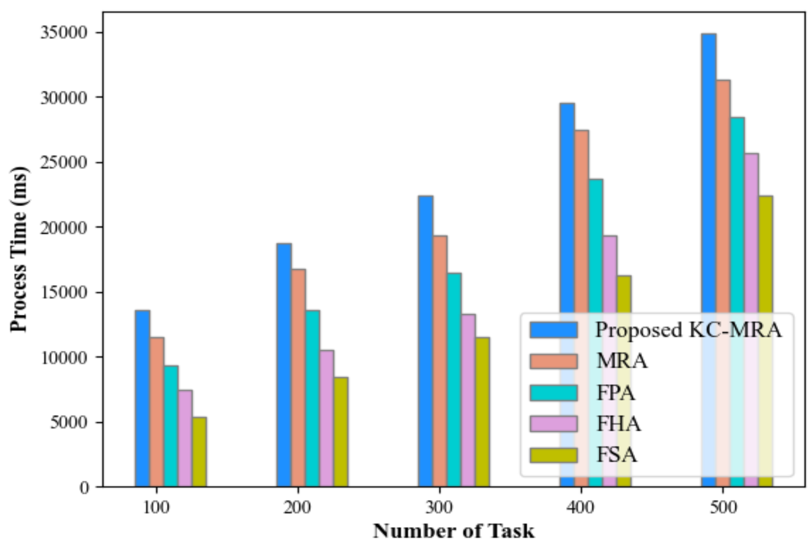
Table 2: Performance measure of proposedESSM-GRU

Techniques	RMSE	RSE
Proposed ESSM-GRU	2.37	0.836
GRU	2.74	0.653
BiLSTM	3.36	0.512
LSTM	3.78	0.423
RNN	4.31	0.385

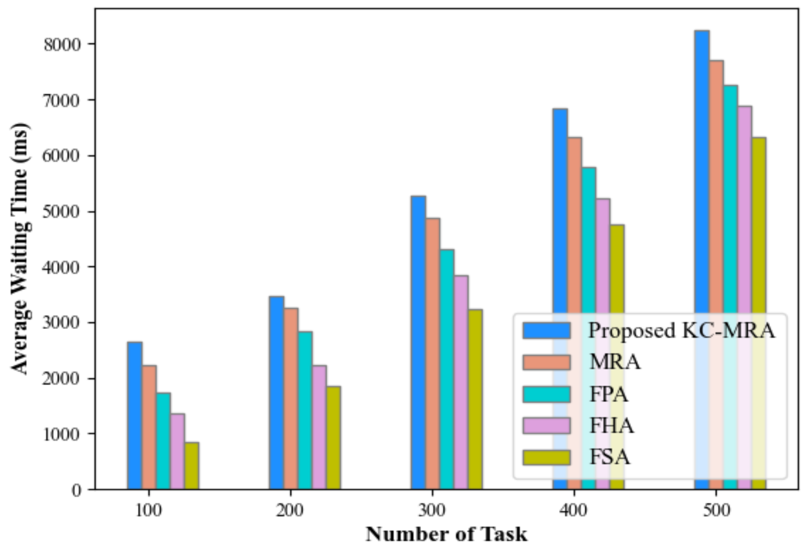
Table 2 analyses the Root Mean Square Error (RMSE) and R-Squared Error (RSE) of various methods. As shown above, the quality of predictions was measured from the proposed model by attaining a minimum value of RSE (0.836) and RMSE (2.37). Hence, with the assistance of normalization, the error rate of the proposed model has been reduced.



(a)



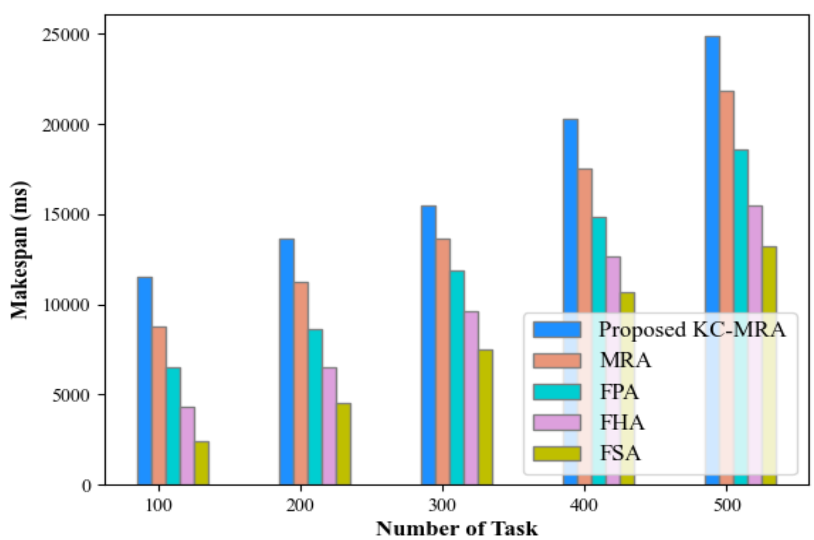
(b)



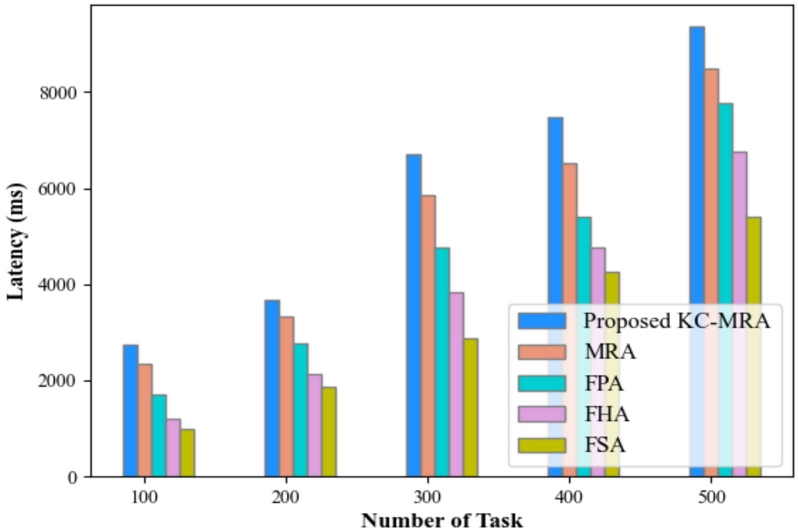
(c)

Figure 4: Performance comparison based on (a) Response (b) Process (c) Average waiting times

In Figure 4, the performance of the proposed KC-MRA and prevailing MRA, Flower Pollination Algorithm (FPA), Fox Hunting Algorithm (FHA), and Flamingo Search Algorithm (FSA) isshown. For 100 to 500requests, the response time of the proposed KC-MRA varies between 3741ms and 7158ms, whereas the response time increases for the rest of the methods. Similarly, the proposed model obtained a lower process and average waiting time of 22345ms and 6329ms, for 500 requests. This less time consumption is due to the addition of the Kendall correlation.



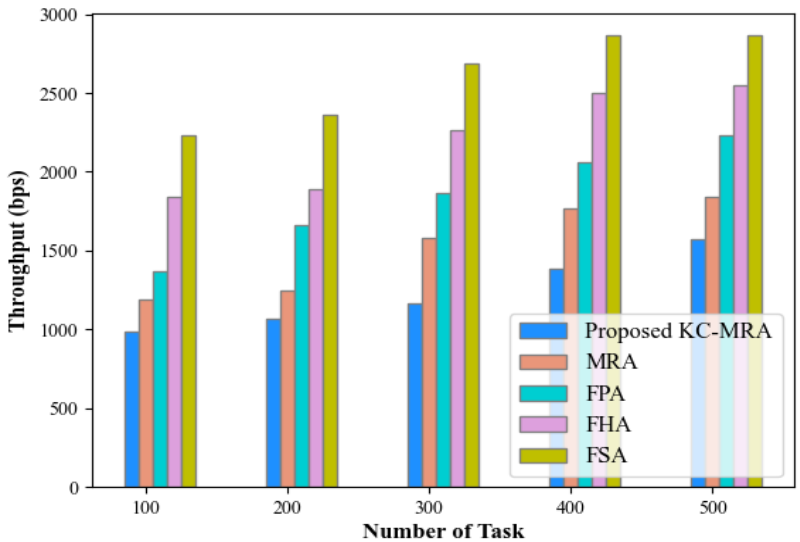
(a)



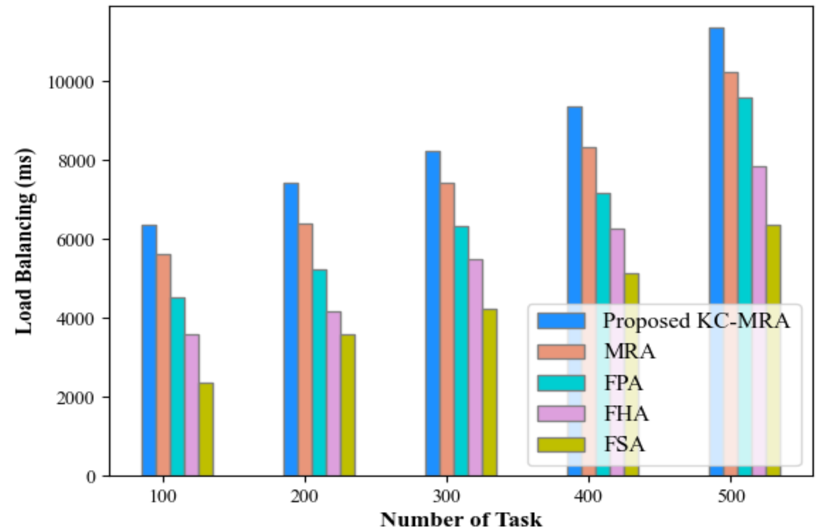
(b)

Figure 5: Performance measure based on (a) Makespan, (b) Latency

From Figure 5, the latency and makespan of the proposed method for balancing 500 requests are 1350ms and 2224.81ms lower than the traditional MRA. The other existing techniques also show higher latency and makespan values than the proposed model.Thus, the proposed method's performance is well compared to the existing methods.



(a)



(b)

Figure 6: Comparative analysis with respect to (a) Throughput (b) Load balancing Time

In Figure 5, for 500 requests, the throughput of the proposed KC-MRA is 2864bps, whereas the existing methods obtained lower throughput of 2547bps (MRA), 2234bps (FPA), 1843bps (FHA), and 1574bps (FSA). Likewise, the proposed model obtained a lower load balancing time than the prevailing models.

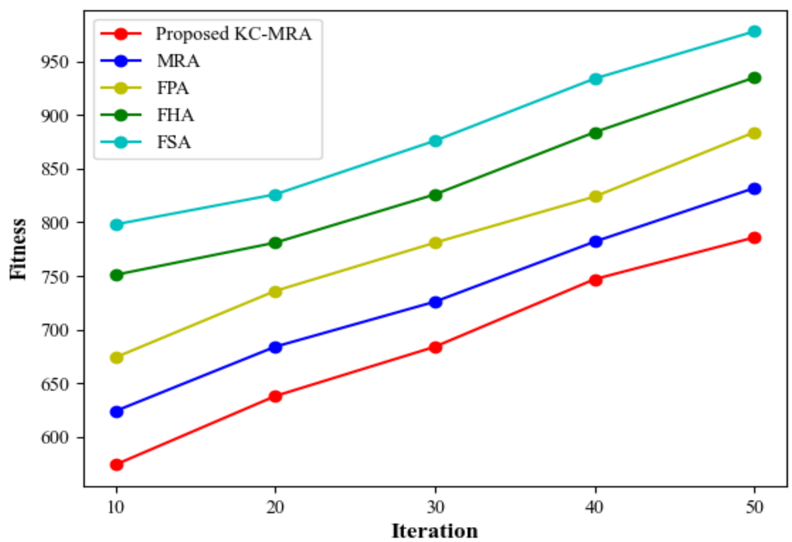


Figure 7: Fitness Vs Iteration

In Figure 7, the fitness values of the proposed KC-MRA approach varies from 798 to 978 for 10 to 50 iterations, respectively, whereas, the existing techniques obtained lower fitness. Thus, the proposed system with the KC technique is better compared to the existing methods.

Table 3: Clustering Time Analysis

Techniques	**Clustering Time (ms)**
Proposed CK-means	596
K-Means	797
K-Medoid	866
BIRCH	1048
CLARA	1345

Table3 displays the superiority of the proposed and existing K-Means, K-Medoid, Algorithm, Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH) and Clustering Large Applications (CLARA). The clustering time taken for the proposed technique is 596ms which is lesser than the conventional methods. The usage of the chosinedistance avoided inaccurate cluster formation resulting in minimal time.

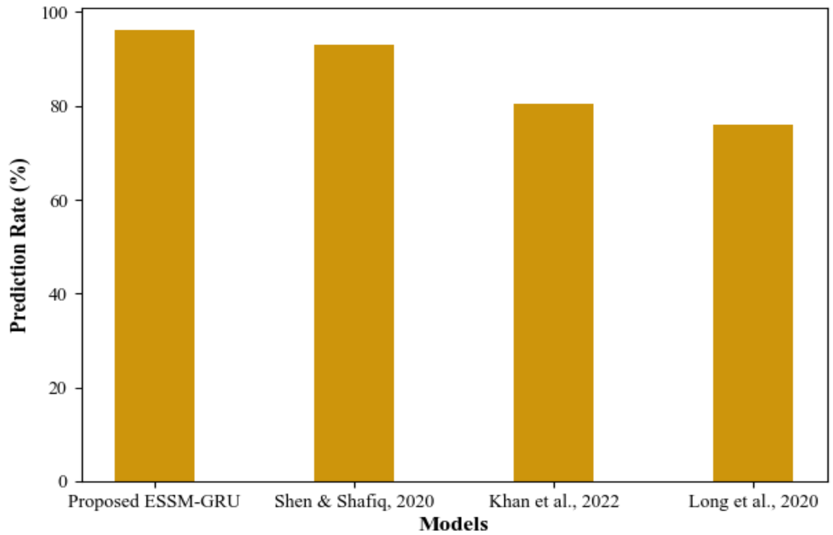


Figure 8: Comparative Analysis

Figure 8 exhibits the prediction rate obtained by various forecasting models. In the proposed model, the SM is predicted by ESSM-GRU after performing vital steps like clustering, load balancing, and request filtering, which improved their prediction rate (96.1%). But, the approaches suggested by ((Shen & Shafiq, 2020)(Long et al., 2020) and (Khan et al., 2022)obtained lower values of 93%, 75.89%, and 80.53%, respectively. This clearly shows the supremacy of the proposed model.

V. CONCLUSION

This paper proposes an SM prediction model based on ESSM-GRU and KC-MRA techniques. The proposed framework undergoes five main phases. Then, the performance analysis and the comparative analysis are carried out to evaluate the proposed system. The proposed framework achieves a prediction rate of 96.1%, and the response time required for balancing 500 tasks is 7158ms with 2864bps throughput and 5398ms latency. When compared with the existing system, the proposed model achieves better results in all metrics. Thus, it is concluded that the proposed system is better and more efficient than the other existing forecasting techniques. As of future work, the proposed model will be extended to focus on security trades of financial contexts from stock exchanges to stockbrokers and stock traders.

REFERENCES

[1] Behera, R. K., Das, S., Rath, S. K., Misra, S., & Damasevicius, R. (2020). Comparative study of real time machine learning models for stock prediction through streaming data. Journal of Universal Computer Science, 26(9), 1128–1147. https://doi.org/10.3897/jucs.2020.059

[2] Goldstein, I., Spatt, C. S., & Ye, M. (2021). Nber Working Paper Series Big Data in Finance. The Review Iof Financial Studies, 34(7), 3213–3225. https://www.wsj.com/articles/elite-m-b-a-programs-report-steep-drop-in-applications-11571130001

[3] Holmlund, M., Van Vaerenbergh, Y., Ciuchita, R., Ravald, A., Sarantopoulos, P., Ordenes, F. V., & Zaki, M. (2020). Customer experience management in the age of big data analytics: A strategic framework. Journal of Business Research, 116, 356–365. https://doi.org/10.1016/j.jbusres.2020.01.022

[4] Jaggi, M., Mandal, P., Narang, S., Naseem, U., & Khushi, M. (2021). Text mining of stocktwits data for predicting stock prices. Applied System Innovation, 4(1), 1–22. https://doi.org/10.3390/asi4010013

[5] Khan, W., Ghazanfar, M. A., Azam, M. A., Karami, A., Alyoubi, K. H., & Alfakeeh, A. S. (2022). Stock market prediction using machine learning classifiers and social media, news. Journal of Ambient Intelligence and Humanized Computing, 13(7), 3433–3456. https://doi.org/10.1007/s12652-020-01839-w

[6] Liu, H., & Long, Z. (2020). An improved deep learning model for predicting stock market price time series. Digital Signal Processing: A Review Journal, 102, 1-16. https://doi.org/10.1016/j.dsp.2020.102741

[7] Long, J., Chen, Z., He, W., Wu, T., & Ren, J. (2020). An integrated framework of deep learning and knowledge graph for prediction of stock price trend: An application in Chinese stock exchange market. Applied Soft Computing Journal, 91, 1-13. https://doi.org/10.1016/j.asoc.2020.106205

[8] Maqsood, H., Mehmood, I., Maqsood, M., Yasir, M., Afzal, S., Aadil, F., Selim, M. M., & Muhammad, K. (2020). A local and global event sentiment based efficient stock exchange forecasting using deep learning. International Journal of Information Management, 50, 432–451. https://doi.org/10.1016/j.ijinfomgt.2019.07.011

[9] Mehta, P., Pandya, S., & Kotecha, K. (2021). Harvesting social media sentiment analysis to enhance stock market prediction using deep learning. PeerJ Computer Science, 7, 1–21. https://doi.org/10.7717/peerj-cs.476

[10] Nabipour, M., Nayyeri, P., Jabani, H., Shahab, S., & Mosavi, A. (2020). Predicting Stock Market Trends Using Machine Learning and Deep Learning Algorithms Via Continuous and Binary Data; A Comparative Analysis. IEEE Access, 8, 150199–150212. https://doi.org/10.1109/ACCESS.2020.3015966

[11] Nti, I. K., Adekoya, A. F., & Weyori, B. A. (2020a). A systematic review of fundamental and technical analysis of stock market predictions. In Artificial Intelligence Review (Vol. 53, Issue 4). Springer Netherlands. https://doi.org/10.1007/s10462-019-09754-z

[12] Nti, I. K., Adekoya, A. F., & Weyori, B. A. (2020b). Predicting Stock Market Price Movement Using Sentiment Analysis: Evidence From Ghana. Applied Computer Systems, 25(1), 33–42. https://doi.org/10.2478/acss-2020-0004

[13] Obthong, M., Tantisantiwong, N., Jeamwatthanachai, W., & Wills, G. (2020). A survey on machine learning for stock price prediction: Algorithms and techniques. FEMIB 2020 - Proceedings of the 2nd International Conference on Finance, Economics, Management and IT Business, 63–71. https://doi.org/10.5220/0009340700630071

[14] Patel, R., Choudhary, V., Saxena, D., & Singh, A. K. (2021). REVIEW of STOCK PREDICTION USING MACHINE LEARNING TECHNIQUES. Proceedings of the 5th International Conference on Trends in Electronics and Informatics, ICOEI 2021, July, 840–846. https://doi.org/10.1109/ICOEI51242.2021.9453099

[15] Rezaei, H., Faaljou, H., & Mansourfar, G. (2021). Stock price prediction using deep learning and frequency decomposition. Expert Systems with Applications, 169, 1-29. https://doi.org/10.1016/j.eswa.2020.114332

[16] Shen, J., & Shafiq, M. O. (2020). Short-term stock market price trend prediction using a comprehensive deep learning system. Journal of Big Data, 7(1), 1-33. https://doi.org/10.1186/s40537-020-00333-6

[17] Valle-Cruz, D., Fernandez-Cortez, V., López-Chau, A., & Sandoval-Almazán, R. (2022). Does Twitter Affect Stock Market Decisions? Financial Sentiment Analysis During Pandemics: A Comparative Study of the H1N1 and the COVID-19 Periods. Cognitive Computation, 14(1), 372–387. https://doi.org/10.1007/s12559-021-09819-8

[18] Vijh, M., Chandola, D., Tikkiwal, V. A., & Kumar, A. (2020). Stock Closing Price Prediction using Machine Learning Techniques. Procedia Computer Science, 167(2019), 599–606. https://doi.org/10.1016/j.procs.2020.03.326

[19] Wu, J. M. T., Li, Z., Srivastava, G., Tasi, M. H., & Lin, J. C. W. (2021). A graph-based convolutional neural network stock price prediction with leading indicators. Software - Practice and Experience, 51(3), 628–644. https://doi.org/10.1002/spe.2915

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

AUTHORS

First Author – Narasimha Rao Konangi, Details: Senior Manager Software Engineering, Capital One., Email: knr2015@gmail.com