CGM: A hybrid model for forecasting future stock price

Ningyao Ningshen[1,*], Vinita Jindal[2] and Punam Bedi[1]

[1] *Department of Computer Science, University of Delhi, New Delhi, India, 110007*
[2] *Department of Computer Science, Keshav Mahavidyalaya, University of Delhi, New Delhi, India, 110034*

Abstract

In finance, Forecasting Stock Price (FSP) poses a significant challenge. However, the swift progress enabled by Artificial Intelligence (AI) techniques, particularly Deep Neural Network (DNN) has propelled the advancements in this sector. Consequently, researchers have investigated the application of different DNN techniques. Despite these efforts, existing models are often shallow and prone to overfitting due to the model's complexity. Consequently, there is still room for improvement in achieving accurate forecasts of the future closing price. Therefore, to advance FSP, a novel hybrid model named CGM is proposed, which is developed using a combination of Convolution, Gated Recurrent Unit (GRU), and Multi-Layer Perceptron (MLP). Thereafter, the CGM model is trained using technical features, Intrinsic Mode Function (IMF) decomposed using Empirical Mode Function (EMD), and a combination of both to exhaustively explore the ability of the CGM model, thus producing three distinct models, namely TF-CGM, IMF-CGM, and TF-IMF-CGM models. Furthermore, to automatically tailor the hyperparameters of the aforementioned models, Neural Architectural Search (NAS) is employed to automatically fine-tune the hyperparameters of the models. During the experiment, the aforementioned models are evaluated using Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE) evaluation metrics using stock indices listed in the New York Stock Exchange (NYSE) and National Stock Exchange (NSE). From the experimental results, it was observed that technical features exert a more significant influence, which leads to the TF-CGM model outperforming the IMF-CGM and TF-IMF-CGM models. Moreover, the proposed models provided better performance when compared with existing models present in the literature.

Keywords

Finance, Artificial Intelligence, Machine Learning, Deep Neural Network, Gated Recurrent Unit, Multi-Layer Perceptron[1]

Symposium on Computing & Intelligent Systems (SCI), *May 10, 2024, New Delhi, INDIA*

* Corresponding author.
† These authors contributed equally.
nningshen@cs.du.ac.in (N. Ningshen); vjindal@keshav.du.ac.in (V. Jindal); pbedi@cs.du.ac.in (P. Bedi)
0009-0003-8909-8648 (N. Ningshen); 0000-0002-0481-4840 (V. Jindal); 0000-0002-6007-7961 (P. Bedi)

1. Introduction

Predicting future stock prices and market indices is crucial for investors and traders seeking positive returns on their investments. However, it remains a challenge due to the complex dynamics influenced by economic, emotional, and political factors, as suggested by the Efficient Market Hypothesis (EMH). Despite this, efforts have been made to develop strategies for successful price prediction [1]. Traditionally, Forecasting Stock Price (FSP) involves examining historical price movement and trading volumes together with a comprehensive assessment of a company's financial health to ascertain its intrinsic value and potential for future growth. However, such methods presume the presence of exploitable trends in historical data for forecasting future prices. Therefore, an alternative method such as statistical models like Auto-Regressive Integrated Moving Average (ARIMA) and Seasonal ARIMA (SARIMA) are commonly utilized for future price forecasting [2], [3], [4]. Nevertheless, the linearity inherent in these statistical models limits their ability to capture the complex dynamics exhibited by historical market data. Consequently, various researchers have explored the usage of Deep Neural Network (DNN) in FSP [1], [5], [6], [7], [8]. However, due to the shallow nature of the existing models, they are susceptible to overfitting. Therefore, this research paper aims to address this limitation by proposing a novel hybrid model named **CGM**, which employs a combination of **C**onvolution, **G**ated Recurrent Unit (GRU), and **M**ulti-Layer (MLP) Perceptron techniques.

A convolutional layer is a building block of a Convolutional Neural Network (CNN) [9], which is widely used in image analysis. It involves a convolution operation, wherein filters slide across the input data, performing an element-wise multiplication between the filter and the input to extract localized spatial information. Meanwhile, GRU, introduced by [10], is a category of Recurrent Neural Networks (RNN), which is specifically designed to address the challenges of RNN in capturing long-range dependencies. Furthermore, MLP, a type of fully connected feedforward ANN is also utilized in the development of the CGM model. During the experiment, the hybrid CGM model is trained using three different types of inputs i.e., Technical Features (TFs), Intrinsic Mode Functions (IMFs), and a combination of both features, thereby producing three unique variants of the CGM model, namely TF-CGM, IMF-CGM, and TF-IMF-CGM models. Furthermore, the Neural Architectural Search (NAS) algorithm [11], which is a subfield in Machine Learning (ML) for streamlining the ML pipeline is introduced to facilitate automatic hyperparameter tuning in the TF-CGM, IMF-CGM, and TF-IMF-CGM models.

The manuscript is divided into six sections. Section 2 presents the summary of the existing literature centered on stock price prediction using DNN techniques. Sections 3 and 4 delve into the proposed work and its experimental study, which is followed by presenting the results and discussion of the experiment in Section 5. Finally, Section 6 presents the conclusion of the research work.

2. Literature Review

In the last decade, researchers have increasingly turned to DNN for stock price prediction, leveraging their capability to capture non-linear dependencies in financial time series data. In the research work conducted by Selvamuthu et al., (2019), three learning techniques, namely Levenberg-Marquardt (LM), Scaled Conjugate Gradient (SCG), and Bayesian Regularization (BR) were explored. Their experimental results, assessed using both tick data and 15-minute data revealed that SCG exhibited superior performance in comparison to LM and BR. However, the authors concluded that incorporating LSTM and integrating sentiment analysis could potentially yield better results. In a comparable investigation carried out by Cao et al., (2019), time-series financial data underwent decomposition into IMFs through the application of both Empirical Mode Decomposition (EMD) and Complete Ensemble Mode Decomposition with Adaptive Noise (CEEMDAN) techniques, and its influence was assessed using a two-layered LSTM model. They observed that two-layered LSTM outperformed one-layered LSTM, MLP, and Support Vector Machine (SVM). Moreover, Chen et al., (2019) explored the application of the attention mechanism in LSTM for stock market prediction. Their experiment was verified using the SSE stock index and found that LSTM with an attention mechanism has more potential, thereby achieving better results compared to standalone LSTM. In the subsequent research work conducted by Shen & Shafiq, (2020), the effectiveness of LSTM was investigated in short-term trend forecasting for market prices. Their approach encompassed feature expansion steps using min-max scaling, polarization, and computing percentage fluctuation. They contended that the superiority of their model over others stems from the comprehensive feature engineering employed in their methodology. Taking a different approach, Yang et al., (2020) leveraged three-dimensional CNN to extract features from stock data meanwhile, LSTM was used for prediction. However, they opted to exclude the pooling layers in their experiment to prevent potential information loss. Their findings underscored the significant role of ranking stock indices in enhancing the overall performance of their models.

In a more recent study conducted by Ji et al., (2021), an LSTM model was proposed for stock price prediction. Their study involved decomposing the stock data into deterministic signals through wavelet transform techniques. Additionally, sentiment analysis was incorporated by utilizing text data acquired from social media. Their experimentation demonstrated success when compared to traditional models. To emphasize the significance of utilizing DNN in forecasting future metal prices in the metal industry, Lin et al., (2022) proposed a novel model that is based on Modified Ensemble Empirical Mode Decomposition (MEEMD) and LSTM. They pointed out that MEEMD demonstrated a better decomposition effect than EMD. In [19], a novel architecture named FDGRU-transformer (Frequency Decomposition induced Gate Recurrent Unit Transformer) was proposed to tackle the stock price prediction problem. Their method involved decomposing stock data into IMFs using the EMD technique. Furthermore, a GRU, LSTM, and multi-head attention were utilized to extract temporal information. Their model's comparison with existing state-of-the-art models indicated better results. Moreover, as a consequence of ANN's popularity in FSP, Seabe et al., (2023) explored the capability of LSTM, GRU, and bi-directional LSTM in forecasting the price of Bitcoin, Ethereum, and Litecoin. Their model's evaluation illustrated that bi-directional LSTM possesses the highest capability in predicting the prices of the cryptocurrencies.

Despite the success in the application of DNN in FSP, the existing models are often characterized by limited depth, potentially leading to overfitting. Therefore, to address this limitation and enhance the prediction accuracy, this research work introduces a novel hybrid model, named CGM by integrating Convolution, GRU, and MLP techniques. Furthermore, to achieve a balance between model prediction performance and representativeness in architectural configurations, the NAS algorithm is employed to automatically optimize hyperparameters, eliminating the need for manual hyperparameter tuning. A detailed description of the proposed hybrid CGM model is presented in the following section.

3. Proposed CGM model

Conventional methods such as time series analysis and statistical approaches have established the groundwork for comprehending market dynamics. However, due to the subjective and non-linear characteristics inherent in traditional approaches, contemporary methodologies like DNN techniques offer more promising alternatives for FSP. Nonetheless, existing DNN models are often shallow, and susceptible to overfitting. Therefore, this research work aims to contribute to the ongoing discourse by proposing a novel hybrid model, named CGM, which stands for Convolution, GRU, and MLP respectively. The proposed hybrid CGM model (given in Figure 1) comprises an Input block, a Conditional block, and an Output block. The components are discussed below.

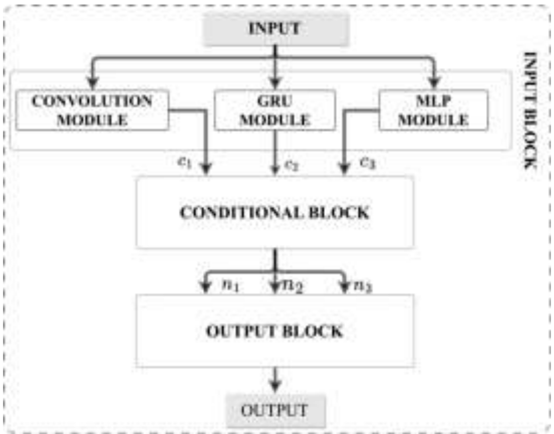

Figure 1. Visual representation of the hybrid CGM model

3.1. Input block

The Input block of the CGM model incorporates Convolution, GRU, and MLP modules. The Convolution module consists of four Convolutional layers. The initial layer establishes a connection with the input layer. Subsequently, the output is concurrently fed into three parallel Convolutional layers, capturing various aspects and representations of the input data. In tandem with the Convolution module, the Input block also integrates a GRU module to learn feature dependencies over long ranges. Similar to the Convolution module, the first

GRU layer establishes a connection with the input layer. The resulting output is then directed to two subsequent GRU layers for further processing, and their outputs are aggregated before being forwarded to subsequent layers. Additionally, to augment feature extraction capability, an MLP module is also employed, comprising three Dense layers. Analogous to the aforementioned modules, the initial Dense layer receives input from the input layer and its output is concurrently transmitted to two Dense layers. The resulting outputs are amalgamated and forwarded for subsequent processing. The visual representation of the Convolution, GRU, and MLP modules of the Input block are given in Figure 2 (a), (b), and (c) respectively.

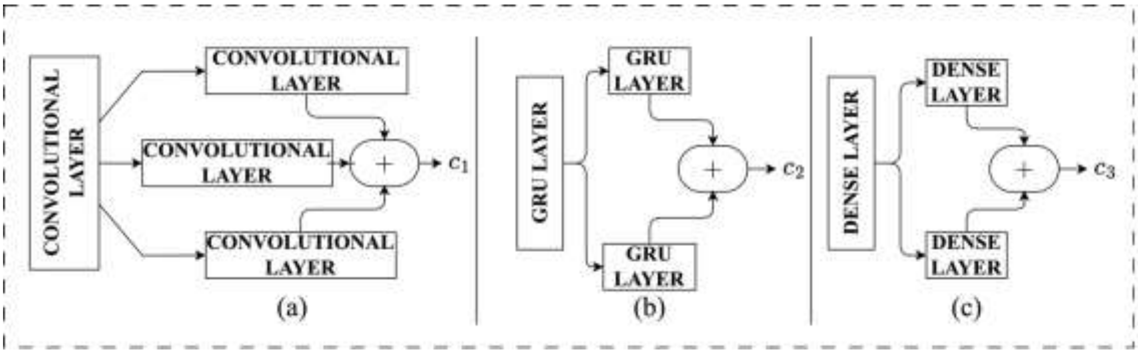

Figure 2. (a) Convolution, (b) GRU, and (c) MLP modules.

3.2. Conditional and Output block

In the Conditional block, three sets of Dropout and Normalization layers are arranged parallelly to improve training stability, prevent overfitting, and further enhance the overall performance of the model. The input of the Conditional block is derived from the Convolution, GRU, and MLP modules of the preceding block. Each of these modules feeds independently into the corresponding Dropout and Normalization layers within the Conditional block. The visual representation of the Conditional block is given in Figure 3 (a).

After the Conditional block, the outputs are forwarded to the Output block. The Output block consists of Concatenation, Bi-directional GRU, and Dense Layers. The Concatenation layer concatenates the input received from the Conditional block to create a unified feature yielding a more comprehensive feature representation of the input. The concatenation steps follow the procedure given in Equation 1.

$$x_{concat} = \begin{cases} g(n_1, n_2, n_3) \; if \; conditional \; block = True \\ g(c_1, c_2, c_3) \qquad\qquad\qquad\qquad Otherwise \end{cases} \tag{1}$$

where g is the concatenation operation, n_1, n_2, n_3 are the outputs from conditional block and c_1, c_2, c_3 are outputs produced by Convolution, GRU, and MLP modules. The visual representation of the Output block is given in Figure 3 (b).

Following the concatenation step, the combined output is passed through a bidirectional GRU layer to handle input from both forward and backward directions. This enables the model to grasp both past and future context, facilitating the model to capture bi-directional dependencies within the data. Ultimately, the resulting output is directed to the final Dense layer with a sigmoid function for predicting the future stock price.

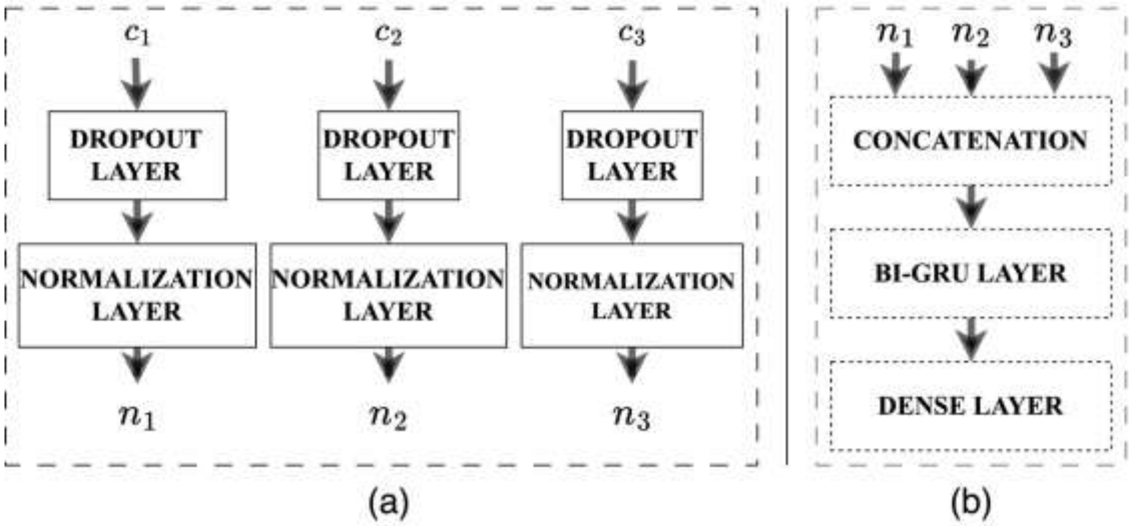

Figure 3. (a) Conditional block and (b) Output block

Throughout the experiment, three models undergo fine-tuning and training using TFs, IMFs, and combinations of both TFs and IMFs. This process leads to the development of three distinct CGM models: TF-CGM, IMF-CGM, and TF-IMF-CGM, where TFs are fed as input to TF-CGM, IMFs into IMF-CGM, and TF+IMFs into TF-IMF-CGM models. Henceforth, we will refer to these models individually as TF-CGM, IMF-CGM, and TF-IMF-CGM. Furthermore, to streamline the experimentation process and eliminate manual hyperparameter tuning for the aforementioned models, the NAS algorithm is employed. This algorithm automatically determines hyperparameters such as learning rate, activation function, hidden units, number of filters, and the decision to include or exclude a conditional block during training. The detailed results of the conducted experiment are discussed in the following section.

4. Experimentation

This section provides an in-depth exploration of the experiments carried out during the research study. The primary goal of the research is to develop a hybrid model that is expressive enough (i.e., a representable number of trainable parameters) as well as improve the performance of predicting future stock prices, thus striking a balance between complexity and performance.

4.1. Data collection and preprocessing

In the course of the study's experimentation, daily OHLCV of four stock indices listed in the New York Stock Exchange (NYSE) and four stock indices listed in the NSE are collected from Yahoo Finance for training and testing.

Furthermore, during the data preparation, NaN (not a number) values are dropped from the dataset. Subsequently, lag features with a window size of 5 were constructed from the independent features. Additionally, the stock data was decomposed into IMFs using the EMD technique as described in [21]. Subsequently, the time-series data is reorganized to embed temporal information into the dataset. Naturally, $x_{t,f} \in X$ denotes a singular input, where $x_{t,f}$ represents data at time t with features f. However, during the experiment of this research work, the input sample x_i and label y_i is reconfigured as $x_i = \{x_{t+i}, x_{t+i+1}, \dots, x_{t+i+n}\}$ and $y_i = \{y_{t+i+n+1}\}$, where the window size n is equal 10. Moreover, each sample is normalized using a Min-Max Scaler [22]. Following the data preprocessing, the data are split into training, validation, and testing subsets in a ratio of 7:2:1.

4.2. Experimental configuration and evaluation metrics

During the experimentation, the TF-CGM, IMF-CGM, and TF-IMF-CGM models were experimented on a Metal Performance Shader (MPS) device. Furthermore, throughout the experiment, Python v3.11 was used as a primary language, and TensorFlow v2.15 as the ML framework. Nonetheless, different programming languages and frameworks could be used for the experiment.

In the initial step of the experiment, the NAS algorithm was used to optimize the hyperparameters of the aforementioned models independently. The NAS algorithm utilizes a RandomSearch technique to determine the value of the hyperparameters. The process involved conducting 10 trials, each comprising four runs to optimize the loss function. Subsequently, the optimized TF-CGM, IMF-CGM, and TF-IMF-CGM were trained using TFs, IMFs, and a combination of TFs and IMFs for forecasting the future close price of daily stock data. The models were trained for 250 epochs with a batch size of 32. Moreover, an early stopping mechanism is employed during training to prevent the models from overfitting with a patience of 30 epochs i.e., the training stops if there is no sign of improvement for 30 successive epochs. Furthermore, Adam [23] optimizer was utilized to minimize the loss function associated with the models.

During the experiment, the Mean Square Error (MSE) given in Equation 2 was used as a loss function to measure the performance of each trial. However, Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE) were used to evaluate the performance of the models during the training and testing phase. The mathematical formulae for the metrics are given in Equation 3 – 5.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{2}$$

$$\mathrm{RMSE} = \sqrt{\mathrm{MSE}} \tag{3}$$

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n}|y_i - \hat{y}_i|}{n} \tag{4}$$

$$\mathrm{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{5}$$

5. Results and Discussion

This section presents the results of the hybrid CGM model in predicting the future close price of daily stock data. As previously stated, three variations of the hybrid CGM model, namely TF-CGM, IMF-CGM, and TE-IMF -CGM underwent hyperparameters tuning individually. The results of the NAS algorithm during hyperparameter optimization for the aforementioned models are given in Table 1.

Table 1. Hyperparameters selected by the NAS algorithm during hyperparameter tuning

Hyperparameters	TF-CGM	IMF -CGM	TF-IMF-CGM
Activation Function	gelu	gelu	gelu
Learning Rate	5.434e-4	1.089e-4	2.598e-4
Dropout Layer	True	False	True
Normalization Layer	True	False	False
GRU units	192	160	128
MLP units	192	160	128
Convolution filter size	160	32	64
Trainable Parameters	1, 587, 493	955, 137	678, 849

From the results presented in Table 1, it is evident that the model favors gelu as the activation function compared to other activation functions. Furthermore, the hyperparameter tuning performed by the NAS algorithm consistently opts for a low learning rate. However, the determination to include or exclude the conditional block is contingent on the variant of the model. Subsequently, the models were evaluated on NYSE and NSE stock data using RMSE, MAE, and MAPE evaluation metrics. The results of TF-CGM, IMF-CGM, and TE-IMF-CGM on the test set are given in Table 2 and 3.

The analysis of Table 2 and 3 leads to the conclusion that TF-CGM exhibited superior performance when compared to IMF-CGM and TF-IMF-CGM. Additionally, the Linear Regression Analysis (LRA) on NYSE and NSE shown in Figure 4 further substantiates the supremacy of TF-CGM over IMF-CGM and TF-IMF-CGM in forecasting the future closing prices of daily stock data. Furthermore, to bolster the claim of the proposed model's superiority, the performance of the models was compared with models documented in the existing literature given in Table 3.

Table 2. Evaluation results of TF-CGM, IMF-CGM, and TF-IMF-CGM models on NYSE stock indices.

Ticker	Model	RMSE	MAE	MAPE
	TF-CGM	3.47	2.58	0.037
AAPL	IMF-CGM	4.89	3.08	0.030
	TF-IMF-CGM	2.79	2.79	0.030
	TF-CGM	2.24	1.63	0.022
ABT	IMF-CGM	3.45	2.45	0.029
	TF-IMF-CGM	2.97	2.13	0.026
	TF-CGM	6.20	4.73	0.030

MSFT	IMF-CGM	9.75	6.34	0.030
	TF-IMF-CGM	8.29	5.42	0.026
	TF-CGM	3.82	2.41	0.063
AMD	IMF-CGM	2.84	1.53	0.041
	TF-IMF-CGM	4.09	2.38	0.060
	TF-CGM	3.93	2.83	0.038
Mean	IMF-CGM	5.23	3.35	0.032
	TF-IMF-CGM	4.53	3.18	0.035

Table 3. Evaluation results of TF-CGM, IMF-CGM, and TF-IMF-CGM models on NSE stock indices.

Ticker	Model	RMSE	MAE	MAPE
RELIANCE	TF-CGM	65.34	49.24	0.027
	IMF-CGM	106.01	82.54	0.038
	TF-IMF-CGM	91.56	72.12	0.037
TATACONSUM	TF-CGM	17.57	13.056	0.028
	IMF-CGM	31.77	22.93	0.036
	TF-IMF-CGM	22.33	16.41	0.031
SBIN	TF-CGM	14.35	10.87	0.032
	IMF-CGM	19.61	13.46	0.031
	TF-IMF-CGM	17.58	13.40	0.036
CIPLA	TF-CGM	24.19	16.84	0.022
	IMF-CGM	36.49	25.16	0.028
	TF-IMF-CGM	35.99	27.94	0.035
	TF-CGM	30.36	22.50	0.027
Mean	IMF-CGM	48.47	36.02	0.033
	TF-IMF-CGM	41.86	32.46	0.034

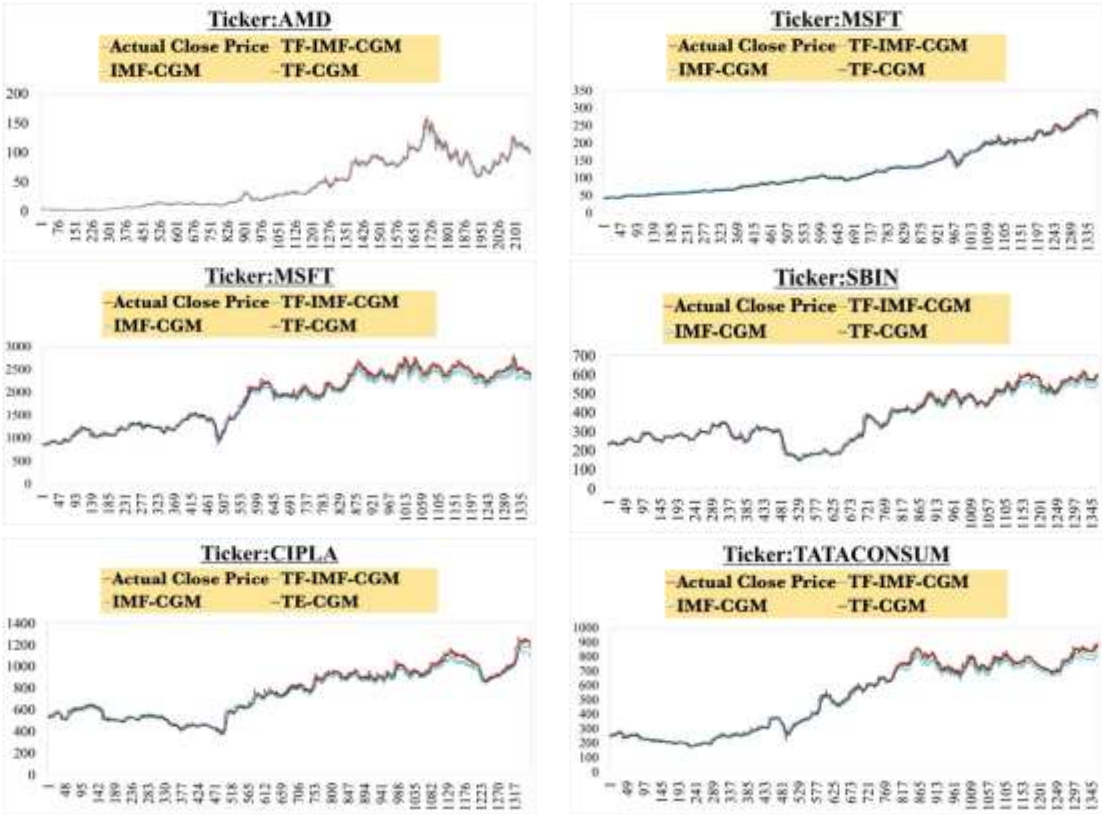

Figure 4. Linear Regression Analysis of TF-CGM, IMF-CGM, and TF-IMF-CGM on NYSE and NSE stock indices.

The analysis of Table 2 and 3 leads to the conclusion that TF-CGM exhibited superior performance when compared to IMF-CGM and TF-IMF-CGM. Additionally, the Linear Regression Analysis (LRA) on NYSE and NSE shown in Figure 4 further substantiates the supremacy of TF-CGM over IMF-CGM and TF-IMF-CGM in forecasting the future closing prices of daily stock data. Furthermore, to bolster the claim of the proposed model's superiority, the performance of the models was compared with models documented in the existing literature given in Table 4.

Table 4. Performance comparison of the proposed hybrid model with the models present in the existing literature. The given scores for TF-CGM, IMF-CGM, and TF-IMF-CGM are the mean of the scores obtained in the NYSE and NSE stock indices given in Table 2 and 3.

Model	RMSE	MAE	MAPE
[12] (SCG+ANN)	-	-	99.908
[14] (EMD+LSTM+ATTENTION)	26.10	16.39	0.66
[24] (Convolution+LSTM)	386.47	-	-
[25] (LASSO-GRU)	27.45	19.14	-
[20] (bi-LSTM)	373.77	-	0.067
[26] (GRU)	0.084	22.94	0.259

TF-CGM (proposed)	**17.14**	**12.66**	**0.032**
IMF-CGM (proposed)	**26.85**	**19.68**	**0.032**
TF-IMF-CGM (proposed)	**23.19**	**17.82**	**0.033**

From the results presented in Table 2 and 3, it can be concluded that TF-CGM outperforms IMF-CGM and TF-IMF-CGM, emphasizing the significant impact of technical factors on the model. Additionally, the models display reduced efficacy when applied to NSE data, as indicated by higher RMSE and MAE values, signifying a relatively larger margin of error. However, the models exhibit relatively similar MAPE values on NSE data, which indicates comparatively similar relative size errors in accuracy. The disparity in the scores of TF-CGM, IMF-CGM, and TF-IMF-CGM on NYSE and NSE data implies variations in the factors influencing the NYSE and NSE markets. Therefore, future research could involve exploring market dynamics and examining the variables that affect the performance of the models. Moreover, sentiment analysis could also be integrated to further enhance the predictive capability of the models.

6. Conclusion

Forecasting future stock prices poses a significant challenge in the financial sector, and addressing this challenge has been an active area of research. Hence, various researchers have contributed to this field by developing models using modern DNN techniques. However, existing models tend to be shallow and susceptible to overfitting. To address this challenge, this research paper proposes a hybrid CGM model that incorporates Convolution, GRU, and MLP techniques.

Moreover, to comprehensively assess the effectiveness hybrid CGM model, three different inputs – TFs, IMFs, and a combination of both – were used to train the CGM model, resulting in three distinct models, namely TF-CGM, IMF-CGM, and TF-IMF-CGM models. Furthermore, to tackle the challenges associated with tailoring the hyperparameters of the models, the NAS algorithm was employed to automatically optimize the hyperparameters. These models were then trained and tested using four stock indices listed in the NYSE and four stock indices listed in the NSE. Thereafter, the performance of the models was evaluated using RMSE, MAE, and MAPE metrics. From the experiment, it was found that TF-CGM outperformed the IMF-CGM and TF-IMF-CGM models by scoring 3.93, 2.83, and 0.038 on NYSE data, and 30.36, 22.50, and 0.027 on NSE data respectively for the aforementioned evaluation metrics. Moreover, the proposed models were compared with existing models, the proposed models demonstrated superior performance.

References

[1] P. M. Tsang *et al.*, "Design and implementation of NN5 for Hong Kong stock price forecasting," *Eng Appl Artif Intell*, vol. 20, pp. 453–461, 2007, doi: 10.1016/j.engappai.2006.10.002.

[2] A. A. Adebiyi, A. O. Adewumi, and C. K. Ayo, "Stock price prediction using the ARIMA model," *Proceedings - UKSim-AMSS 16th International Conference on Computer Modelling and Simulation, UKSim 2014*, pp. 106–112, 2014, doi: 10.1109/UKSIM.2014.67.

[3] Daryl, A. Winata, S. Kumara, and D. Suhartono, "Predicting Stock Market Prices using Time Series SARIMA," *Proceedings of 2021 1st International Conference on Computer Science and Artificial Intelligence, ICCSAI 2021*, pp. 92–99, 2021, doi: 10.1109/ICCSAI53272.2021.9609720.

[4] S. Khanderwal and D. Mohanty, "Stock Price Prediction Using ARIMA Model," *International Journal of Marketing & Human Resource Research*, vol. 2, no. 2, pp. 98–107, Apr. 2021, Accessed: Dec. 18, 2023. [Online]. Available: https://www.journal.jis-institute.org/index.php/ijmhrr/article/view/235

[5] K. Alkhatib, H. Najadat, I. Hmeidi, and M. K. A. Shatnawi, "Stock Price Prediction Using K-Nearest Neighbor (KNN) Algorithm," *International Journal of Business*, vol. 3, no. 3, 2013, Accessed: Dec. 18, 2023. [Online]. Available: www.ijbhtnet.com

[6] S. Mehtab, J. Sen, and A. Dutta, "Stock Price Prediction Using Machine Learning and LSTM-Based Deep Learning Models," in *SOMMA*, vol. 1366, 2021, pp. 88–106. doi: 10.1007/978-981-16-0419-5_8.

[7] J. Sen, "Stock Price Prediction Using Machine Learning and Deep Learning Frameworks," in *ICBAI*, 2018.

[8] X. Zhang, Y. Huang, K. Xu, and L. Xing, "Novel modelling strategies for high-frequency stock trading data," *Financial Innovation*, vol. 9, no. 1, pp. 1–25, Dec. 2023, doi: 10.1186/S40854-022-00431-9/TABLES/8.

[9] Goodfellow Ian, Bengio Yoshua, and Courville Aaron, *Deep Learning*, 1st ed. MIT Press, 2016. Accessed: Jan. 24, 2024. [Online]. Available: https://www.deeplearningbook.org/

[10] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *NIPS 2014 Workshop on Deep Learning, December 2014*, 2014. doi: https://doi.org/10.48550/arXiv.1412.3555.

[11] F. Hutter, H. H. Hoos, and K. Leyton-Brown, "Sequential Model-Based Optimization for General Algorithm Configuration," 2011, pp. 507–523. doi: 10.1007/978-3-642-25566-3_40.

[12] D. Selvamuthu, V. Kumar, and A. Mishra, "Indian stock market prediction using artificial neural networks on tick data," *Financial Innovation*, vol. 5, no. 1, pp. 1–12, Dec. 2019, doi: 10.1186/S40854-019-0131-7/FIGURES/5.

[13] J. Cao, Z. Li, and J. Li, "Financial time series forecasting model based on CEEMDAN and LSTM," *Physica A: Statistical Mechanics and its Applications*, vol. 519, pp. 127–139, Apr. 2019, doi: 10.1016/J.PHYSA.2018.11.061.

[14] L. Chen, Y. Chi, Y. Guan, and J. Fan, "A Hybrid Attention-Based EMD-LSTM Model for Financial Time Series Prediction," *2019 2nd International Conference on Artificial Intelligence and Big Data, ICAIBD 2019*, pp. 113–118, May 2019, doi: 10.1109/ICAIBD.2019.8837038.

[15] J. Shen and M. O. Shafiq, "Short-term stock market price trend prediction using a comprehensive deep learning system," *J Big Data*, vol. 7, no. 1, pp. 1–33, Dec. 2020, doi: 10.1186/S40537-020-00333-6/TABLES/9.

[16] C. Yang, J. Zhai, and G. Tao, "Deep Learning for Price Movement Prediction Using Convolutional Neural Network and Long Short-Term Memory," *Math Probl Eng*, vol. 2020, pp. 1–13, Jul. 2020, doi: 10.1155/2020/2746845.

[17] X. Ji, J. Wang, and Z. Yan, "A stock price prediction method based on deep learning technology," *International Journal of Crowd Science*, vol. 5, no. 1, pp. 55–72, Apr. 2021, doi: 10.1108/IJCS-05-2020-0012/FULL/PDF.

[18] Y. Lin, Q. Liao, Z. Lin, B. Tan, and Y. Yu, "A novel hybrid model integrating modified ensemble empirical mode decomposition and LSTM neural network for multi-step precious metal prices prediction," *Resources Policy*, vol. 78, p. 102884, Sep. 2022, doi: 10.1016/J.RESOURPOL.2022.102884.

[19] C. Li and G. Qian, "Stock Price Prediction Using a Frequency Decomposition Based GRU Transformer Neural Network," *Applied Sciences 2023, Vol. 13, Page 222*, vol. 13, no. 1, p. 222, Dec. 2022, doi: 10.3390/APP13010222.

[20] P. L. Seabe, C. R. B. Moutsinga, and E. Pindza, "Forecasting Cryptocurrency Prices Using LSTM, GRU, and Bi-Directional LSTM: A Deep Learning Approach," *Fractal and Fractional 2023, Vol. 7, Page 203*, vol. 7, no. 2, p. 203, Feb. 2023, doi: 10.3390/FRACTALFRACT7020203.

[21] N. E. Huang *et al.*, "The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis," *RSPSA*, vol. 454, no. 1971, pp. 903–998, 1998, doi: 10.1098/RSPA.1998.0193.

[22] M. J. Zaki and W. Meira, Jr, "Data Mining and Machine Learning: Fundamental Concepts and Algorithms," *Data Mining and Machine Learning*, Jan. 2020, doi: 10.1017/9781108564175.

[23] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *International Conference on Learning Representations*, 2014.

[24] S. Mehtab and J. Sen, "Stock Price Prediction Using CNN and LSTM-Based Deep Learning Models," *2020 International Conference on Decision Aid Sciences and Application, DASA 2020*, pp. 447–453, Nov. 2020, doi: 10.1109/DASA51403.2020.9317207.

[25] Y. Gao, R. Wang, and E. Zhou, "Stock Prediction Based on Optimized LSTM and GRU Models," *Sci Program*, vol. 2021, 2021, doi: 10.1155/2021/4055281.

[26] C. Chen, L. Xue, and W. Xing, "Research on Improved GRU-Based Stock Price Prediction Method," *Applied Sciences 2023, Vol. 13, Page 8813*, vol. 13, no. 15, p. 8813, Jul. 2023, doi: 10.3390/APP13158813.