Journal of Research Administration
Society of Research Administrators International

ATTENTION-DRIVEN GRU MODEL WITH FEATUREFUSION FOR ROBUST STOCK PRICE PREDICTION

Boni Murali Manohar[1*] and Dr. Harshlata Vishwakarma[2†]

[1*]Master of Technology, Artificial Intelligence and Data Science, VIT Bhopal, Madhya Pradesh, India.

[2] Assistant Professor Grade 1, School of Computer Science and Engineering, VIT Bhopal, Madhya Pradesh, India.

*Corresponding author(s). Boni Murali Manohar;
Contributing authors: Dr. Harshlata Vishwakarma;
†These authors contributed equally to this work.

Abstract

In the realm of stock price prediction, an integration of advanced Deep Learning architectures with comprehensive data sources has become imperative for achiev- ing heightened accuracy. This study proposes an innovative approach leveraging an attention-driven Gated Recurrent Unit (GRU) model, enhanced by feature fusion techniques, to predict stock prices. The model incorporates historical price data and technical indicators, capitalizing on their collective insights into market trends and patterns. The attention mechanism embedded within the GRU archi- tecture dynamically emphasizes salient features from the input sequence, which allows the GRU model to focus on the most relevant information for prediction. Furthermore, the integration of feature fusion enables the seamless combina- tion of diverse data streams, facilitating a more holistic understanding of market dynamics. Through extensive experimentation and validation on real-world finan- cial datasets, our proposed model demonstrates superior predictive performance compared to baseline models, showcasing its efficacy in capturing intricate mar- ket behaviors and enhancing robustness in stock price forecasting. This research study contributes to an enhancement of predictive analysis in financial markets, offering a promising avenue for informed decision-making and risk management strategies.

Keywords: Stock price prediction, Attention-mechanism, EMA, GRU, LSTM, RMSE.

1 Introduction

Investing in stock market[1] is like trying to navigate a rough sea, where each wave can either propel you towards success or leave you adrift in uncertainty. Imagine possessing the ability to anticipate these waves, to foresee the market's direction with precision. This is the tantalizing promise of stock market prediction, a domain where advanced technologies and data-driven models[2] hold the key to unlocking lucrative opportunities.

Nowadays, the area of stock market prediction has affirmed a surge of interest and innovation. The allure of accurately forecasting stock prices[3] has captivated both academic researchers and industry professionals, driven by the potential for substantial profits. However, traditional methods often struggle to capture the intricate dynamics of market behavior, prompting a quest for more sophisticated approaches.

Amidst the cacophony of market noise and volatility, the challenge of accurately predicting stock movements remains a daunting task. The question that lingers is: How can we improve the precision and resilience of stock market prediction by utilizing cutting edge technologies and abundant data sources?

This paper's main objective is to explore new breakthroughs in stock market prediction by combining feature fusion approaches with an attention-driven Gated Recurrent Unit (GRU) model. We aim to enhance predictive accuracy by leveraging historical price data and technical indicators, embed an attention mechanism within the GRU[4] architecture to highlight key features from input sequences, and explore the advantages of feature fusion to seamlessly merge diverse data streams and gain a comprehensive understanding of market dynamics.

This study is significant in the realm of stock market prediction and finan- cial analytics. Introducing our innovative model, the Attention-Driven GRU Model with Feature Fusion, we aspire to propel the advancement of stock market predic- tion methodologies, provide insightful perspectives on the integration of advanced deep learning[5] architectures with extensive data sources, showcase the potential for improved accuracy and robustness in stock price forecasting, and offer a promis- ing avenue for informed decision-making and effective risk management strategies in financial markets.

As we embark on this journey, we will delve into the architecture, mechanisms, and outcomes of our proposed model. Through experimentation and validation on authentic financial datasets, we endeavor to reveal the efficacy and potential influence of innovative methods in the ever-evolving realm of stock market forecasting.

2 Related Techniques and Literature review

In the dynamic world of stock market prediction, the fusion of advanced deep learning architectures with extensive data sources is crucial for achieving precise forecasts. Our study introduces an innovative approach by harnessing an attention-driven Gated Recurrent

Unit (GRU) model, enriched with feature fusion techniques, to predict stock prices. Through a thorough exploration of existing literature, we aim to delve into the significance of attention mechanisms and feature fusion in bolstering the robustness and performance of stock price forecasting.

Lee et al.[6] introduced the AttBiLSTM model for stock trading strategy analysis, integrating it with Technical Analysis (TA) and Technical Indicators (TIs) including RSI, stochastic oscillators, BIAS, MACD, and W%R. The increasing accessibility of stock market information via the Internet aids investor decisions. The study achieves an impressive 68.83% accuracy in trend prediction and a remarkable 42.74% Return on Investment (ROI) on the TPE0050 stock, demonstrating AttBiLSTM's effectiveness in merging TA with Deep Neural Networks (DNNs) [7]for stock price forecasting.
Huicheng Liu[8] explores the integration of deep learning techniques with financial news analysis. Traditional methods rely on historical market data, while the internet's rise has democratized access to stock information. Technical Analysis (TA) has been a staple tool for investors, yet existing methods often overlook the context in financial news. To gather semantic features in news content, the paper presents a new Attention- based LSTM (At-LSTM) model that combines bidirectional-LSTM and self-attention techniques. The model shows the potential of NLP advances in computational finance by competitively predicting movements within the S&P 500 index and individual firm stock prices.
Karan Pardeshi et al.[9] delve into stock market prediction's core, emphasizing the need for accurate forecasts in investment decisions. Their focus lies on the Long Short- Term Memory with Sequential Self-Attention Mechanism model, aiming for precise future stock price prediction. Through rigorous experiments on SBIN, HDFCBANK, and BANKBARODA datasets, they validate the model's effectiveness with superior root-mean-squared error and R-square results. LSTM-SSAM outperforms traditional time series models, offering a promising framework for stock market forecasting.
Jilin Zhang[10] and colleagues present a groundbreaking stock market prediction method by applying a CNN-BiLSTM-attention-based model. This model combines temporal feature extraction from a CNN and a BiLSTM network to enhance stock price and index prediction accuracy. An attention mechanism automatically assigns weights to information features, followed by prediction through a dense layer. Their approach outperforms traditional methods like LSTM, CNN-LSTM, and Attention based CNN-LSTM in predicting the CSI300 index.
Yazeed Alsubaie[11], with his colleagues, explores the selection of technical indi- cators (TIs) for stock market forecasting, aiming to improve prediction accuracy and investment returns. Ranking fifty TIs using diverse feature selection methods, they introduce a cost-sensitive fine-tuned na¨ıve Bayes classifier, surpassing other models in investment performance. Results emphasize that an excessive number of TIs can reduce accuracy and increase misclassification costs. The CSFTNB classifier strikes a balance

between investment return and risk, showing promise for stock market trend prediction. This study offers valuable insights into TI selection, providing a frame- work for practitioners to optimize prediction accuracy and investment returns while considering misclassification costs.
Taha Bugra Celik[12] and colleagues propose an innovative approach to stock mar- ket prediction, emphasizing the importance of prediction accuracy and reliability. They address the limitations of machine learning techniques[13] in providing explainable pre- dictions and the challenges of achieving incremental accuracy improvements. Instead, they advocate the 'eXplainable Artificial Intelligence' approach to assess prediction

reliability and guide decision-making. Their two-stage stacking ensemble model inte- grates machine learning, empirical mode decomposition (EMD), and XAI, achieving high accuracy on various stock market indices. By incorporating the LIME algorithm, their framework offers reliable predictions, making it a valuable tool for capital market players to make decisions. The author's contribution lies in the integration of LIME with machine learning techniques, providing original insights into enhancing predic- tion reliability and accuracy. Their proposed framework demonstrates distinct success in facilitating informed decision-making for investors.

3 Algorithm study

3.1 Attention-mechanism

Attention mechanisms[14], inspired by how humans focus on specific details when pro- cessing information, have become incredibly popular in deep learning. Using attention mechanisms primarily serves to enable models that assist in focusing on significant categories among the input data throughout the prediction process. The attention processes allow the model to give varying amounts of attention to distinct elements based on their significance, rather than considering all data equally. In time-series fore- casting, self-attention[9] models have become popular because they enable each time step to consider others within the same sequence. Unlike traditional models, which rely on an encoder-decoder setup, self-attention models[15] efficiently capture global dependencies.
The Transformer architecture is frequently used by researchers to create self- attention models. This design consists of feed-forward neural networks and numerous layers of self-attention. As seen in Fig. 1, the self-attention mechanism determines weights for attention by comparing every set of time steps in sequence. The encoded hidden states will be represented by H = [H1, H2, . . . , HT].
The hidden state encoded Hi and the previous hidden state decoder is given: $\text{Score}(t) = V \cdot \tanh(W1 \cdot HT + W2 \cdot \text{prev hidden dec})$ (1)
where the hyperbolic tangent function (tanh) adds non-linearity to the weighted sum

of the hidden state encoded and the previous hidden state decoder, W1 and W2 were learnable weight matrices, and V represents a learnable vector.
The scores then moved on to a softmax function, which was used to determine the weights for attention (alpha1, alpha2,..., alphaT). In order to make the weights of attention comprehensible as probabilities, the softmax function ensures that their sum equals 1.
Softmax function is described below:

$$\mathrm{softmax}(x) = \frac{e^{x}}{\sum e^{x}} \quad (2)$$

where, x indicates the input vector.

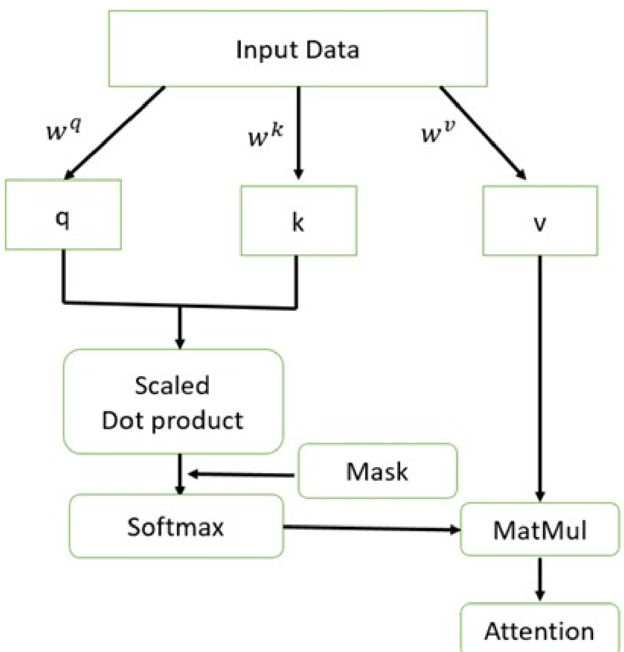


Fig. 1: Self-Attention mechanism[9]

The sum of weights of the encoded hidden states helps in calculating the context vector (context):

$$\mathrm{context} = \alpha_1 \cdot H_1 + \alpha_2 \cdot H_2 + \ldots + \alpha_T \cdot H_T \quad (3)$$

where - $alpha_i$ represents a weight or coefficient, - H_i represents a hidden state vector.

3.2 Gated Recurrent Unit

Gated Recurrent Unit is a type of RNN architecture designed to address the challenges faced by traditional RNNs, particularly in handling long-term dependencies within sequences. The information flows via the network are maintained by specialized processes called gates, which are incorporated into GRUs. As seen in Fig. 2, these gates include an update gate and the reset gate.

In a GRU, the update gate chooses how much new data to blend in with the prior hidden state, while the reset gate helps choose which portions of the previous hidden state to forget according to the current input. As a kind of temporary memory, the model generates a candidate hidden state by fusing the new input with the old hidden state. The new concealed state for the current time step is then generated by passing this candidate state through the update gate. GRUs are renowned for their effectiveness in effectively capturing long-term relationships in data.

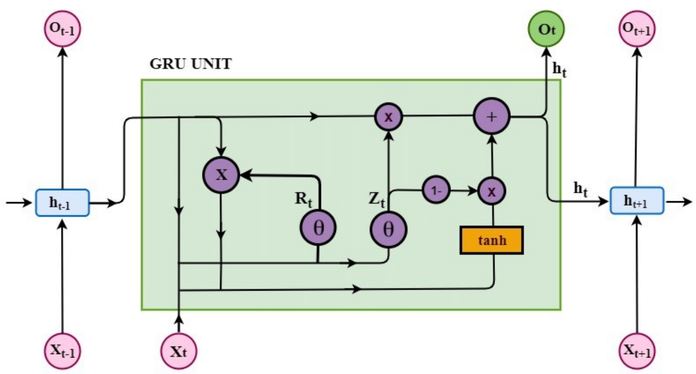


Fig. 2: Gated-Recurrent Unit(GRU)[16]

In GRU architecture[17], at each time step, it recieves an input X_t and the previous hidden state H_{t-1} from the before time step t 1, prodūcing the next hidden state H_t for the next time step on the basis of last state. Both Reset Gate and Update Gate are the two primary gates of a GRU.

Reset Gate:

Update Gate:

where:

$$r_t = \sigma(W_r \cdot [H_{t-1}, X_t]) \quad (4)$$

$$u_t = \sigma(W_u \cdot [H_{t-1}, X_t])$$

(5)

- For reset and update gates, the weight matrices are W_r and W_u,
- σ denotes the sigmoid activation function.

To compute the hidden state H_t, GRU follows a two-step process. First, it generates candidate's hidden state:

where

$$\tilde{H}t = \tanh(W_h \cdot [r_t \odot H_{t-1}, X_t]) \quad (6)$$

- $\odot$ represents element-wise multiplication.

- r_t controls the influence of the last hidden state H_{t-1} on the candidate state $\tilde{H}_t$.

$$H_t = (1 - u_t) \odot H_{t-1} + u_t \odot \tilde{H}t \quad (7)$$

Here

- u_t is the candidate state of $\tilde{H}t$, the last hidden state H_{t-1} which contributes to
current hidden state H_t.
- It ranges from 0 to 1, controlling the balance between historical and new information.

6

3.3 Long Short-Term Memory

LSTM,[18] stands as a crucial component in the concept of Deep Learning, mainly in Recurrent Neural Networks. Its forte lies in its ability to apprehend long-term dependencies, rendering it invaluable for tasks involving the prediction of sequences. Unlike conventional neural networks, LSTM is adept at processing entire sequences of data instead of processing individual data points. This attribute makes it especially proficient in discerning and prognosticating patterns within sequential data, such as time series, textual information, and spoken language.
Functionally, LSTM[19] operates akin to a typical RNN cell, comprising three inte- gral components known as gates: the first one is Forget gate, the second is an Input gate, and the last one is an Output gate as shown in Fig 3. The information flow into and out of the memory cell is controlled by these gates, with each gate serving a distinct purpose in determining what information to retain, learn, or transmit to sub- sequent steps. Essentially,

LSTM undergoes cyclical processes wherein it evaluates the relevance of past information, acquires new insights from current input, and propa- gates updated information to future timestamps. With provisions for both short-term memory (represented by the hidden state) and long-term memory (reflected in the cell state), LSTM networks possess the capability to retain and accumulate knowledge over time. In essence, LSTM's utilization of specialized gates to govern information flow addresses long-term dependency challenges prevalent in traditional neural networks, thus proving instrumental in predictive tasks involving sequential data.

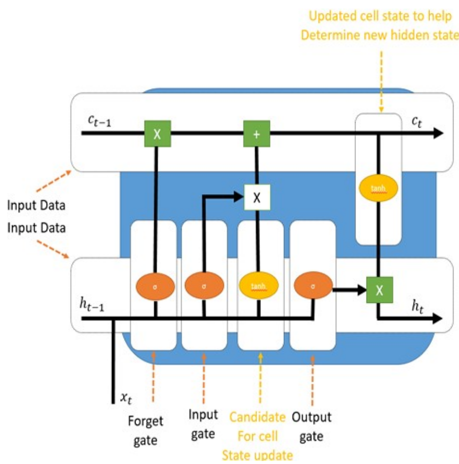


Fig. 3: LSTM Architecture[9]

Forget Gate:

7This gate makes sure that which information from the past should be remembered or forgotten. The formula for this gate is:

$$f_t = (W \cdot H_{t-1} + U_f X_t) \qquad (8)$$

where: ft - Forget gate at timestamp t, Xt- -input at the current time,
Wf, Uf - weight matrices associated with these inputs,
$\sigma(x)$- sigmoid function

Input Gate:

This gate decides how important new information is. The formula for this gate is:

$$\boldsymbol{i}_t = (\mathbf{W}_i[H_{t-1}, X_t] + \mathbf{U}_i) \qquad (9)$$

New Information Update:

This part combines the new information with the old information to update the cell state. The formula is:

$$N = \tanh(W_N [H_{t-1}, X_t] + U_N) \tag{10}$$

Cell State Update:

This part updates the state of a cell according to the new information and forget gate. The formula is:

$$C_r = f_r Cr_{r-1} + N_e \tag{11}$$

Output Gate:

The information to be output from the current time step is determined by this gate. This gate's formula is:

$$O_t = o(W_o[H_{t-1}, X_t] + U_o) \tag{12}$$

Hidden State Update:

The hidden state for the current time step is created in this section by combining the output gate and the cell state. The equation is:

$$H_t = o_t \cdot tanh(C_t) \tag{13}$$

3.4 Recurrent Neural Network

RNN[4] is like our brain's way of understanding and predicting sequences of infor- mation. They're super smart at handling things like text, speech, or even time-based data. Think of them as a tool that remembers what came before, just like how our

8brains recall past experiences to make sense of the present. This memory is sort of like a hidden state in the network, keeping track of important details from the sequence. The cool thing about RNNs is how they loop information through a layer, allowing what happened in the past to influence what they predict for the future. This looping mechanism makes them great for dealing with sequences of different lengths, which can be tricky for other models. When we teach an RNN, we use a method called backpropagation through time. This means we give it data over multiple steps, so it can learn from its mistakes and get better. So, in simple terms, RNNs are like our brain's clever way of processing sequences, remembering what happened, and using
that to guess what comes next.

Hidden State Calculation in RNN:

$$h_t = f(W_{hx}x_t + W_{hh}h_{t-1} + b_h) \quad (14)$$

where:

- x_t is input at time step t
- h_t is hidden state at time step t
- The weight matrices for the input and hidden states are shown by W_{hx} and W_{hh}
- b_h represents the bias
- f is activation function

Output Calculation in RNN:

$$y_t = f(W_{hy}h_t + b_y) \quad (15)$$

where:

- y_t shows an output at time step t
- W_{hy} describes weight matrix for the output
- b_y is basically a bias

Backpropagation Through Time (BPTT) for updating weights:

where:

-ΔW is the change in weights.

-η is the learning rate

∂D

$\Delta W = -\eta$ ∂E

(16)

$\frac{\partial E}{\partial W}$ -

is the gradient of error w.r.t weights.

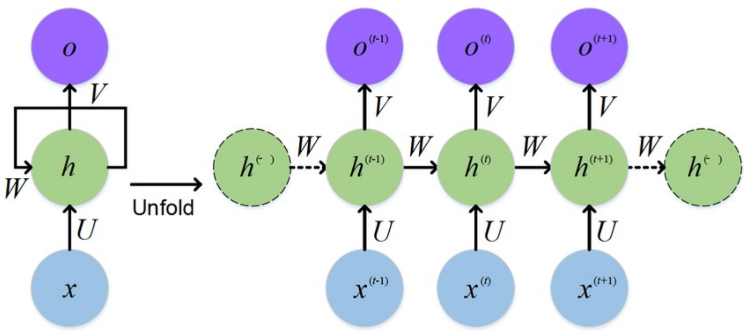


Fig. 4: RNN Architecture[20]

4 Methodology

4.1 Data set

The dataset utilized in this investigation study spans a decade of TATA MOTORS stock data of the National Stock Exchange (NSE)[21] India, which covers the duration from January 2014 to December 2023. It contains a range of historical data, including trade volume, open and close prices, and high and low values. Along with this, technical indicators like Relative Strength Index, Moving Average Convergence Divergence, and Exponential Moving Averages for 5, 10, and 20 days are computed from this historical data. With a total of 2451 carefully curated data points, the dataset is categorized into three parts for model development and evaluation. This extensive dataset forms a robust basis for analyzing TATA MOTORS' stock performance and developing accurate predictive models for stock price forecasting, contributing valuable insights to the realm of financial analytics and prediction.

4.2 Feature Engineering

When analyzing historical stock data, it's common to use technical indicators to gain insights into potential market trends. The Relative Strength Index is a popular and effective indicator that determines if a company is overbought or oversold. By ana- lyzing an average of gains and losses over a specific period, the RSI is calculated. The stock may be overbought, indicating a possible sell signal, if the RSI number is higher than 70. On the other hand, if the RSI value is less than 30, it can be oversold, indicating a potential buying opportunity.

$$RSI = 100 - \frac{100}{1 + \frac{AverageGain}{AverageLoss}} \quad (17)$$

where

- Average Gain: The average of price gains over a given period of time.
- Average Loss: It denotes the average of price losses over the same time duration.

- specified period (usually 14 days) for which you're calculating the RSI.

1The Moving Average Convergence Divergence is another crucial indicator that may be used to spot shifts in momentum and possible trend reversals. The difference between two Exponential Moving Averages with varying time periods. The mathematical formula of MACD is mentioned below:

$$MACD = EMA_{short} - EMA_{long} \tag{18}$$

where

- Short-term Exponential Moving Average (EMA): Usually based on 12 periods.
- Long-term Exponential Moving Average (EMA): Usually based on 26 periods.

At last, Exponential Moving Averages (EMAs) serve to refine price data and detect trends with greater precision compared to simple moving averages. By assigning higher significance to recent prices, EMAs exhibit increased responsiveness to current market dynamics. Typical periods for EMAs encompass intervals such as 5 days, 20 days, and 30 days, as depicted in Fig 5.

$$EMA = \frac{ClosingPrice \times Multiplier + PreviousEMA \times (1 - Multiplier)}{Period}$$

(19)

where

- Close Price is the price of stock at the end of the trading day.
- Multiplier is a smoothing factor calculated based on the chosen period.
- Period is the number of days used in the calculation.

Investors might potentially optimize their investment strategies by using techni- cal indicators to calculate and analyze historical stock data and make well-informed judgments about when to buy or sell shares. Indicators like moving averages, RSI, and stochastic oscillator offer insights into market trends and price movements. Mov- ing averages smooth price data, indicating uptrends or downtrends. RSI identifies overbought (above 70) or oversold (below 30) conditions. Stochastic oscillator reveals momentum and reversal points. Combined with historical analysis[22], investors gain a comprehensive view, reducing emotional biases. These data-driven strategies enable systematic and disciplined trading, helping navigate market volatility with confidence. Ultimately, investors aim for better risk-adjusted returns over the long term.

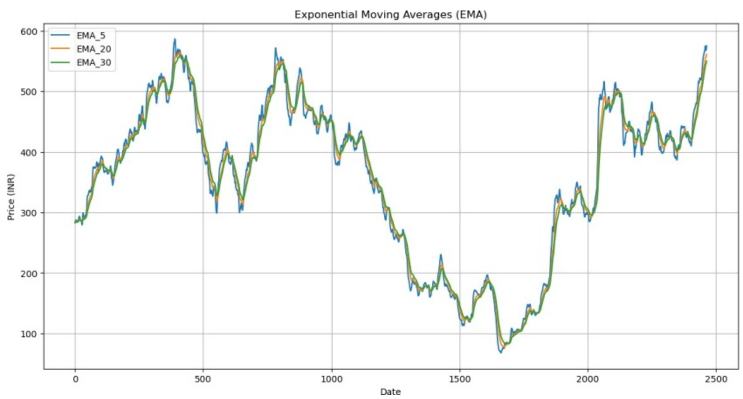


Fig. 5: EMA of 5, 20, 30 days

4.3 Experimental Design

This stock market research study is structured into six different phases: data collection, feature extraction, data pre-processing, model training, model testing, and the last one is prediction findings. A conceptual flowchart illustrating the method's analysis and design is depicted in Fig 6.

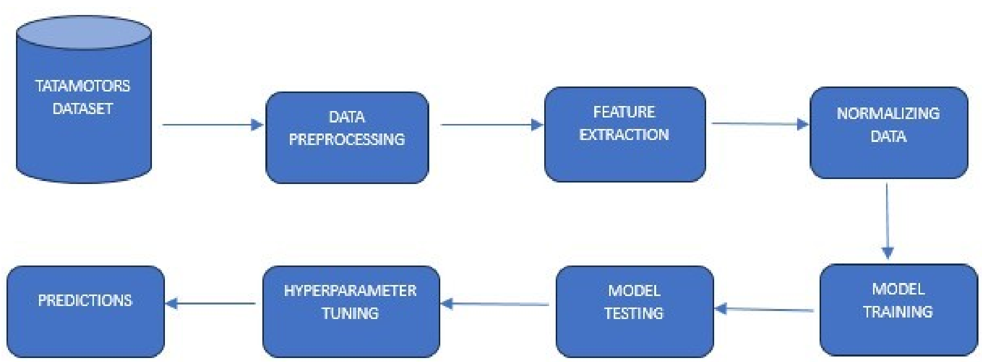


Fig. 6: Process flowchart

Due to the unsuitability of the original Yahoo Finance dataset for direct model training and testing, data preprocessing becomes imperative. To tackle the issue of inconsistent data magnitudes, data normalization is implemented. This process involves transforming the data into a specified interval and scaling it to a particu- lar range, thereby creating a more useful dataset for the experiment. Specifically, the data's values are adjusted, enhancing the model's accuracy and convergence speed. By employing data normalization, the study ensures that the model is well-trained

and tested on a given dataset, which aids in producing reliable and consistent results throughout the analysis.

4.4 Model Analysis

4.4.1 Proposed Algorithm - GRU using a self Attention mechanism

Integrating a self-attention mechanism with a Gated Recurrent Unit (GRU) can make it better at understanding long sequences of data and paying attention to important parts.

Self-Attention Mechanism:

Integrating self-attention with a GRU enhances the model's contextual understand- ing. At each step, the model evaluates Query, Key, and Value for each word, identifying crucial elements with attention weights. These vital words' Values are then fused with the GRU's input, improving predictions. This integration guides the GRU to focus on essential sequence parts, boosting performance and context comprehension.

Integration for Enhanced GRU:

Self-attention integration with a GRU improves context understanding. The model assesses Query, Key, and Value for each word, recognizing important elements using attention weights. Merging these crucial words' Values with the GRU's input enhances predictions. This mechanism directs the GRU to focus on key sequence elements, improving performance and context understanding.

Training:

During training, the model refines its weights by analyzing examples and correct- ing errors through backpropagation. Backpropagation allows the model to recognize and rectify previous prediction mistakes, improving accuracy over time. This iterative learning from errors enhances the model's performance, akin to how we learn from our mistakes to improve.

Step I: The 'attention layer' function is a crucial component in neural networks, specifically designed to create an Attention Mechanism layer. This mechanism is vital for

assigning weights to each element in an input sequence, indicating their relative importance. The function accepts input data in the format of (batch size, time steps, input dim), where batch size represents the number of samples in each batch, time steps is the length of the input sequence, and input dim is the dimensionality of each element.
The function first extracts the input dim from the input data to define the dimen- sions for the subsequent Dense layer. It then constructs a Dense[4] layer with input dim units, applying the softmax activation function to compute attention scores for every element in the sequence. The softmax activation ensures that these scores are nor- malized, ranging between 0 and 1 and summing up to 1. These scores act as weights, determining the importance of each element.
After computing the attention scores, the function calculates the context vector by taking the dot product of these scores with the input sequence. This process generates a weighted sum of the input elements, with the weights being determined by the

attention scores. Ultimately, the context vector serves as a focused representation of the input sequence, emphasizing the significant elements based on the computed attention weights.
In essence, this attention mechanism enables the neural network to prioritize cer- tain elements of the input sequence, enhancing its ability to understand context and make more informed predictions. This functionality is particularly beneficial for tasks where understanding the importance of each part of the sequence is critical for accurate processing.

Table 1: Attention-GRU Architecture

Layer (type)	**Output Shape**	**Param #**
$input_1$(*InputLayer*)	(None, 9, 1)	0
gru (GRU)	(None, 9, 128)	50304
dense (Dense)	(None, 9, 128)	16512
$dense_1$(*Dense*)	(None, 9, 1)	129
activation (Activation)	(None, 9, 1)	0
dot (Dot)	(None, 1, 128)	0
concatenate (Concatenate)	(None, 10 128)	0
$dense_2$(*Dense*)	(None, 10, 64)	8256
$dense_3$(*Dense*)	(None, 10, 1)	65
Total params	**17089 (66.75 KB)**	

Trainable params	**17089 (66.75 KB)**	
Non-trainable params	**0 (0.0(Byte)**	

Step II: The attention-based GRU model is a powerful tool for predicting sequences effectively. Its secret lies in the attention layer, which assigns weights to sequence elements, emphasizing their importance. When creating the model, we start by defining its architecture: an input layer followed by a GRU layer with 64 units, enabling the model to process sequences efficiently. The attention mechanism, crafted with the 'attention layer', enhances the model's understanding by focusing on crucial elements within the sequence.
After compiling the model with the Adam optimizer and MAE loss function, we train it on the training data for 50 epochs with a batch size of 64. Evaluating its performance on validation and test data ensures its accuracy in predicting the next values in the sequence. This process empowers the model to learn from unseen data, making it adept at forecasting future sequences.

4.5 Evaluation Metrics

Evaluating[1] the performance of machine learning models, such as those with inte- grated self-attention mechanisms in GRU, is crucial to understanding their effective- ness. Several evaluation metrics can be employed to assess these model's performance in tasks such as sequence prediction:

MAE(Mean Absolute Error)

14It measures the deviation between two predicted values, commonly employed in evaluating the precision of forecasts in machine learning regression tasks. Essentially, MAE computes the absolute mean values of the differences between the values that were predicted and the actual values. This aids in quantifying the average extent of our predictions divergence from actual outcomes.
Mathematically, the formula for MAE,

where

n

$MAE = \frac{1}{} \quad A \qquad \Sigma_{|}$

n t

t=1

— F_t|

(20).

- MAE represents the Mean Absolute Error.
- n is the total number of data samples.

-The actual value at time ’t’ is represented by A_t.

- The predicted value at time ’t’ is represented by F_t.

RMSE (Root-mean-squared error)

It is an additional crucial statistic that is used in regression analysis, namely, to evaluate predictors. An indicator of the average size of prediction mistakes, expressed in the same unit as the target variable, Root Mean Square Error is derived from Mean Squared Error. RMSE represents the square root of MSE, providing a concise summary of prediction accuracy.

Mathematically, the formula for RMSE can be written as:

$$\sqrt{\frac{1}{\ } \Sigma}$$

where

$RMSE = ,_{n}$

t=1

$(A_t - F_t)^2$

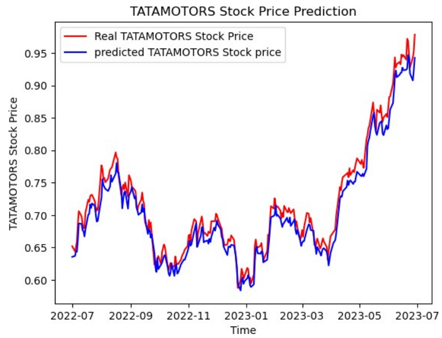
- A_t shows the observed value,
- F_t shows the predicted value,
- n shows the no. of data samples.

5 Result

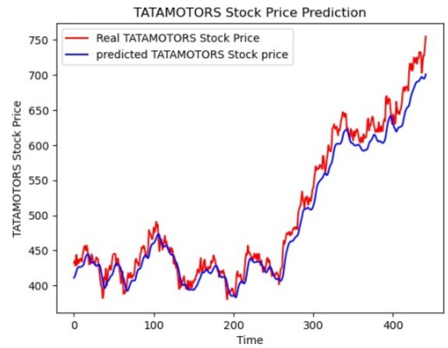
To determine the performance of the proposed Attention-GRU model, we compared with various models such as LSTM, GRU, and RNN using the TATAMOTORS stock dataset of NSE India. The outcomes are depicted in Fig 7, where the x-axis shows the dates, and the y-axis shows the closing price. In the given graph, the red line represents the actual data tested for the closing price, while the other line depicts the predicted values for the Closing price. This visualization allows us to see how well our models perform in predicting the stock prices compared to the actual values. It's crucial to analyze these predictions to understand the strengths and weaknesses of each model and determine which one provides the most accurate forecasts for TATAMOTORS stock.

Table 2: Evalution Measures

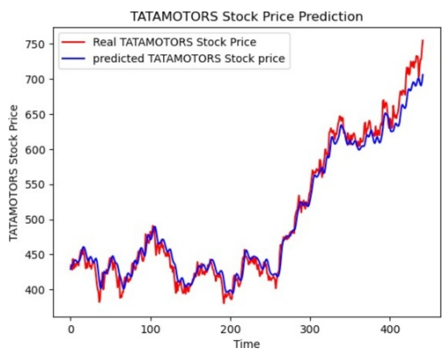
Model/Evaluation metrics	MSE	RMSE	MAE
RNN	0.000737	0.026108	0.020019
LSTM	0.001030	0.032093	0.023605
GRU	0.000590	0.024699	0.018037
Attention-GRU	**0.000284**	**0.01688**	**0.014409**



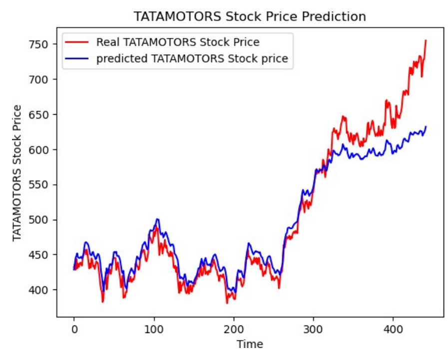
(a) Attention-GRU Result Graph



(b) GRU Result Graph



(c) LSTM Result Graph



(d) RNN Result Graph

Fig. 7: Comparison of Different Models

Table 2 displays the prediction errors of various models on the TATAMOTORS stock dataset. Notably, our proposed Attention-GRU model exhibits superior predic- tive power compared to others. It outperforms LSTM, RNN, and GRU models across all metrics including RMSE, MSE, and MAE. Additionally, the actual and predicted data values fall within the same range, with lower error differences as compared to alternative algorithms. This indicates the robustness of the Attention-GRU model in accurately predicting TATAMOTORS stock prices. Understanding these results is

16

vital for making informed decisions in financial forecasting, as they explained the reli- ability and efficiency of our proposed Attention-GRU model in capturing stock market dynamics.

6 Conclusion and Future work

In conclusion, our research study on the comparative analysis of LSTM, GRU, RNN, and Attention-GRU models using the TATAMOTORS stock dataset from NSE India reveals valuable insights into their predictive performance. Fig 7 showcases the mod- els' abilities to forecast TATAMOTORS stock prices against actual values, with the Attention-GRU model emerging as superior, as evidenced by its consistently accu- rate predictions depicted in the graph. Table 2 further supports this, illustrating the Attention-GRU model's significantly lower prediction errors compared to LSTM, GRU, and RNN. And also demonstrating the Attention-GRU model's consistent out-performance across metrics like MSE, RMSE, and MAE. These results are pivotal for financial forecasting decisions, highlighting the reliability and efficiency of the Attention-GRU model as a powerful tool for capturing stock market dynamics and providing accurate predictions for TATAMOTORS stock.

In the Future, this study would be beneficial to upgrade the potential of Attention- GRU, to further enhance prediction accuracy. Additionally, integrating external factors such as economic indicators, market trends, and news sentiment analysis into the model helps in providing a more comprehensive understanding of stock price move- ments. Conducting a more in-depth analysis of different time intervals and market conditions could also offer valuable insights into the model's adaptability and robust- ness in varying scenarios. Moreover, exploring deep learning techniques for anomaly detection in stock prices could be a promising avenue for future research.

References

[1] Nabipour, M., Nayyeri, P., Jabani, H., Mosavi, A., Salwana, E., S., S.: Deep learning for stock market prediction. Entropy **22**(8), 840 (2020) https://doi.org/10.3390/e22080840

[2] Kusuma, R.M.I., Ho, T.-T., Kao, W.-C., Ou, Y.-Y., Hua, K.-L.: Using Deep Learning Neural Networks and Candlestick Chart Representation to Predict Stock Market. arXiv (2019). https://doi.org/10.48550/ARXIV.1903.12258

[3] Lien Minh, D., Sadeghi-Niaraki, A., Huy, H.D., Min, K., Moon, H.: Deep learning approach for short-term stock trends prediction based on two-stream gated recurrent unit network. IEEE Access **6**, 55392–55404 (2018) https://doi.org/10.1109/access.2018.2868970

17

[4] Jiang, W.: Applications of deep learning in stock market prediction: Recent

progress. Expert Systems with Applications **184**, 115537 (2021) https://doi.org/10.1016/j.eswa.2021.115537

[5] M, H., E.A., G., Menon, V.K., K.P., S.: Nse stock market prediction using deep- learning models. Procedia Computer Science **132**, 1351–1362 (2018) https://doi.org/10.1016/j.procs.2018.05.050

[6] Lee, M.-C., Chang, J.-W., Yeh, S.-C., Chia, T.-L., Liao, J.-S., Chen, X.-M.: Apply- ing attention-based bilstm and technical indicators in the design and performance analysis of stock trading strategies. Neural Computing and Applications **34**(16), 13267–13279 (2022) https://doi.org/10.1007/s00521-021-06828-4

[7] Sawhney, R., Agarwal, S., Wadhwa, A., Shah, R.R.: Deep attentive learning for stock movement prediction from social media text and company correlations. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, ??? (2020). https://doi.org/10.18653/v1/2020.emnlp-main.676

[8] Liu, H.: Leveraging Financial News for Stock Trend Prediction with Attention-Based Recurrent Neural Network. arXiv (2018). https://doi.org/10.48550/ARXIV.1811.06173

[9] Pardeshi, K., Gill, S.S., Abdelmoniem, A.M.: Stock Market Price Prediction: A Hybrid LSTM and Sequential Self-Attention based Approach. arXiv (2023). https://doi.org/10.48550/ARXIV.2308.04419

[10] Zhang, J., Ye, L., Lai, Y.: Stock price prediction using cnn-bilstm-attention model.
Mathematics **11**(9), 1985 (2023) https://doi.org/10.3390/math11091985

[11] Alsubaie, Y., Hindi, K.E., Alsalman, H.: Cost-sensitive prediction of stock price direction: Selection of technical indicators. IEEE Access **7**, 146876–146892 (2019) https://doi.org/10.1109/access.2019.2945907

[12] Ç elik, .B.E. Taha Buğra / İ can: Extending machine learning prediction capabili- ties by explainable ai in financial time series prediction. Applied Soft Computing**7**, 109876 (2023) https://doi.org/10.24433/CO.1813338.v1

[13] Yoo, P.D., Kim, M.H., Jan, T.: Machine learning techniques and use of event information for stock market prediction: A survey and evaluation. In: Inter- national Conference on Computational Intelligence for Modelling, Control and Automation and

International Conference on Intelligent Agents, Web Technolo- gies and Internet Commerce (CIMCA-IAWTIC'06). IEEE. https://doi.org/10. 1109/cimca.2005.1631572

[14] Li, Y., Li, L., Zhao, X., Ma, T., Zou, Y., Chen, M.: An attention-based lstmmodel for stock price trend prediction using limit order books. Journal of Physics:

18

Conference Series **1575**(1), 012124 (2020) https://doi.org/10.1088/1742-6596/1575/1/012124

[15] He, S., Gu, S.: Multi-modal Attention Network for Stock Movements Prediction. arXiv (2021). https://doi.org/10.48550/ARXIV.2112.13593

[16] Liu, Y., Wang, Z., Zheng, B.: Application of regularized gru-lstm model in stock price prediction. IEEE Access (2019) https://doi.org/10.1109/ICCC47050.2019.9064035

[17] Li, C., Qian, G.: Stock price prediction using a frequency decomposition based gru transformer neur. Applied Sciences **13**, 222 (2022) https://doi.org/10.3390/app13010222

[18] Vishwakarma, V.K., Bhosale, N.: Stock price prediction using lstm: An advanced review. SSRN Electronic Journal (2022) https://doi.org/10.2139/ssrn.4097211

[19] Ghosh, P., Neufeld, A., Sahoo, J.K.: Forecasting directional movements of stock prices for intraday trading using LSTM and random forests. arXiv (2020). https://doi.org/10.48550/ARXIV.2004.10178

[20] Feng, W., Guan, N., Li, Y., Zhang, X., Luo, Z.: Audio visual speech recognition with multimodal recurrent neural networks. In: 2017 International Joint Confer- ence on Neural Networks (IJCNN). IEEE, ??? (2017). https://doi.org/10.1109/ijcnn.2017.7965918

[21] Tata Motors Limited (TATAMOTORS.NS). https://finance.yahoo.com/quote/TATAMOTORS.NS/history Accessed February 22, 2024

[22] Xu, Y., Cohen, S.B.: Stock movement prediction from tweets and historical prices. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics,

??? (2018). https://doi.org/10.18653/v1/p18-1183