MATHEMATICAL MODELING AND COMPUTING, Vol. 12, No. 1, pp. 158–167 (2025)

Dynamic learning rate adjustment using volatility in LSTM models for KLCI forecasting

Shakawi A. M. H. A.[1,2], Shabri A.[2]

[1] *Centre for Pre University Studies, University Malaysia Sarawak, 94300, Kota Samarahan, Sarawak, Malaysia*

[2] *Department of Mathematical Sciences, Faculty of Science, University Technology Malaysia, 81310, Johor, Malaysia*

(Received 19 November 2024; Revised 8 February 2025; Accepted 16 February 2025)

The prediction of financial market behaviour constitutes a multifaceted challenge, attributable to the underlying volatility and non-linear characteristics inherent within market data. Long Short-Term Memory (LSTM) models have demonstrated efficacy in capturing these complexities. This study proposes a novel approach to enhance LSTM model performance by modulating the learning rate adaptively based on market volatility. We apply this method to forecast the Kuala Lumpur Composite Index (KLCI), leveraging volatility as a key input to adapt the learning rate during training. By integrating volatility into the learning process, the model can better accommodate market fluctuations, potentially leading to more accurate and robust predictions. The proposed dynamic learning rate adjustment mechanism operates by scaling the learning rate according to the most recent volatility measurements, ensuring that the model adapts swiftly to changing market conditions. This approach contrasts with traditional static learning rates, that may fail to sufficiently account for the dynamic of financial markets. We conduct extensive experiments using historical KLCI data, comparing our proposed model with standard LSTM and other baseline models. The results demonstrate that our volatility-adjusted learning rates outperform conventional LSTM models with fixed learning rates with respect to predictive performance and stability. The findings suggest that incorporating volatility into learning rate adjustments can significantly enhance the predictive capability of LSTM models for stock market forecasting. The improved forecasting accuracy of the KLCI index highlights the potential of this approach for broader applications in financial markets.

Keywords: *average true range; Bursa Malaysia; LSTM; machine learning; time series; volatility-adjusted learning rates.*

2010 MSC: 37M10, 62P20, 68T99, 91B84 **DOI:** 10.23939/mmc2025.01.158

1. Introduction

Financial markets are characterized by their dynamic and volatile nature, posing significant challenges for accurate prediction and analysis [1]. The Kuala Lumpur Composite Index (KLCI), representing the performance of the Malaysian stock market, is no exception. Traditional statistical models often struggle to capture the non-linear patterns and abrupt changes present in financial time series data. To address these limitations, machine learning approaches, particularly Long Short-Term Memory (LSTM) networks, have garnered substantial attention due to their proficiency in modeling sequential data and capturing long-term dependencies [2].

LSTM networks, a specialized form of recurrent neural networks (RNN), are engineered to address the vanishing gradient problem, rendering them well-equipped for the task of time series forecasting. Despite their advantages, LSTM models are sensitive to the selection of hyper-parameters, as the learning rate is one of the most crucial factors [3]. The learning rate parameter controls the magnitude of updates to the model parameters during the gradient-based optimization process, influencing the convergence speed and stability of the model. An optimal learning rate can lead to faster convergence and better performance, whereas a suboptimal learning rate can impede efficient model convergence or cause model divergence. In traditional training setups, the learning rate is often set manually and remains fixed throughout the entire training process. This static approach is straightforward but may not be optimal as the model's performance might suffer from issues like slow convergence or overshooting [4, 5].

In contrast, dynamic learning rate management adjusts the learning rate during training based on various criteria, aiming to optimize model performance and training efficiency [6]. Dynamic management can be categorized as predefined schedulers, adaptive optimizers and custom dynamic schedulers. Various learning rate schedulers have been designed to dynamically tune the learning rate during training. Step decay scheduler (SDS) is one of the simplest and most widely used learning rate schedulers. In this approach, the learning rate is adaptively scaled down by a factor at specific intervals, typically every few epochs. This approach helps in stabilizing the training process as the model gets closer to convergence by reducing the step size, thus preventing overshooting the minimum of the loss function. For instance, reducing the learning rate by half every ten epochs is a common practice.

Exponential decay scheduler (EDS) provides a more gradual reduction in the learning rate. It decreases the learning rate exponentially over time. This method allows for a smooth transition in learning rates, helping the model to adjust its weight updates progressively as it learns. Polynomial decay scheduler (PDS) is another variant where the learning rate follows a polynomial function. This scheduler is characterized by a learning rate that decreases over time according to a polynomial formula. This approach provides flexibility in controlling the rate of decay, offering a more tailored reduction based on the training requirements. Cosine annealing scheduler (CAS) is another sophisticated technique that reduces the learning rate according to a cosine function. This methodology commences with a comparatively elevated learning rate and progressively diminishes to a lower bound in a cosine shape, often resetting the learning rate periodically in a cyclical manner. This technique has shown to be effective in escaping local minima and achieving better performance in certain scenarios [7, 8].

Adaptive optimizers, such as Adaptive Moment Estimation (Adam) [9] and Root Mean Square Propagation (RMSprop) [10], take a different approach by automatically adjusting the learning rate based on observed gradients during training. These algorithms calculate the learning rate dynamically, often considering the magnitude and direction of recent gradient updates. Among variation of Adam optimizer include AdaGrad [11], where the learning rate is updated as a function of the cumulative squared magnitudes of the historical gradients, AdaDelta [12] where adaptation of AdaGrad utilizing a sliding window of previous gradient updates, AdamW [13], Adamax [9], where Adam was used with infinity norm and NAdam [14], where Adam was combined with Nesterov's accelerated gradient.

Custom dynamic schedulers represent a sophisticated evolution in learning rate management, tailored to enhance model performance by integrating real-time data characteristics and external factors into the training process [3, 15]. Unlike predefined schedulers that follow static, predetermined patterns, custom dynamic schedulers adjust the learning rate dynamically in response to the model's performance or environmental conditions. Introducing external factors or real-time data characteristics into the learning rate adjustment allows models to be more responsive and adaptive to the conditions they encounter during training. By dynamically modifying the learning rate based on these external signals, custom dynamic schedulers can better manage the learning process, avoiding the pitfalls of overly aggressive or overly conservative updates [16].

Volatility represents the degree of fluctuation in the price of a financial asset or market index over time, often indicating the level of uncertainty or risk in the market. High volatility is characterized by sharp price swings, while low volatility suggests more stable price movements. This makes predicting financial market behaviour particularly challenging due to the unpredictable nature of volatility and the non-linear patterns in the data. Volatility is also a key reflection of market sentiment, capturing the collective perception of investors during periods of uncertainty or stress. Incorporating volatility into stock index forecasting models helps to identify these critical periods, providing deeper insights into investor behaviour and market sentiment, which can anticipate potential market trends and shifts [17]. As volatility often accompanies significant price movements, integrating it into forecasting models allows for better adaptation to sudden changes in market conditions. This leads to more accurate predictions, especially during volatile periods, that are crucial for making timely and effective decisions [18].

Despite the success of LSTM models in forecasting time series data, traditional implementations often employ fixed learning rates, which do not account for the dynamic nature of financial market volatility. This limitation results in models that may be too rigid, struggling to adapt quickly enough during periods of high market fluctuation. Stock indices often experience periods of high volatility driven by economic events, market sentiment, or geopolitical developments [19]. Traditional learning rate schedules may fail to adapt swiftly to these rapid changes, leading to suboptimal model performance.

Incorporating volatility into the learning rate adjustment is crucial, particularly for models forecasting stock index data, due to the inherently unpredictable and fluctuating nature of financial markets [20]. Through the dynamic adjustment of the learning rate parameter based on real-time volatility, models can become more agile and responsive to the market's behaviour [21]. This adaptive approach allows the model to slow down learning during periods of high volatility to avoid overfitting to transient noise and speed up learning when the market is stable to capture underlying trends more efficiently [22]. Consequently, incorporating volatility into the learning rate helps improve the model's ability to extrapolate insights from historical data, enhancing the accuracy and robustness of stock index forecasts in the face of market dynamics.

This study introduces a novel approach to LSTM model training by integrating a volatility-driven learning rate adjustment mechanism. AdamVol (Adam optimizer incorporating volatility metrics) represents an approach by integrating market volatility directly into the tuning of the learning rate process. Through the dynamic adjustment of the learning rate based on current volatility levels, AdamVol aims to address the gap in traditional LSTM models that rely on static learning rates. By incorporating market volatility into the learning process, AdamVol enhances the model's ability to better capture and respond to rapid market fluctuations, improving prediction accuracy and adaptability.

This study makes several key contributions to the field of stock market prediction using deep learning models. At first, it introduces a novel mechanism for dynamically adjusting the learning rate of LSTM models based on market volatility, specifically leveraging the ATR as a volatility measure. This method allows the model to adapt in real time to market changes, a feature absent in traditional LSTM models with static learning rates. Secondly, comprehensive experiments are conducted to compare this dynamic approach with standard LSTM models and baseline methods, evaluating its effectiveness in improving prediction accuracy and robustness. Finally, this paper addresses a critical gap in stock market forecasting by demonstrating the significance of incorporating volatility measures into the learning process of deep learning models.

2. Method

2.1. Data collection and preparation

This study utilized the daily closing prices of the KLCI from January 2, 2018, to January 31, 2023, comprising a total of 1 225 observations obtained from the Yahoo Finance website. The dataset was divided into two segments: the training set, spanning from January 2, 2018, to April 20, 2022 (90% of the data), and the testing set, covering the period from April 22, 2022, to January 31, 2023 (10% of the data).

2.2. Average true range

ATR is a flexible measure of volatility that quantifies the average magnitude of price fluctuations over a given timeframe [23]. Unlike traditional measures that only consider closing prices, ATR incorporates both intraday price highs and lows, providing a more comprehensive view of market volatility [24]. It is particularly valuable for setting stop-loss levels and determining the potential range of price movements within a given trading session or timeframe.

The ATR is calculated as the mean of the absolute differences between consecutive daily closing prices over a specified period as follows:

$$\mathrm{ATR} = \frac{1}{p}\sum_{i=1}^{n} \mathrm{TR}_i, \tag{1}$$

where TR_i represents true range of each day i, p represents the number of periods

2.3. LSTM model

The LSTM model represents a variation of RNN that incorporates a sequence of iterative computational components. LSTMs are particularly well-suited for tasks involving time series analysis, natural language processing, and other applications where temporal dependencies are crucial [25]. LSTMs incorporate unique memory cells enabling them to maintain information over long periods. As illustrated in Figure 1, an LSTM unit consists of a memory cell, an input gate, an output gate, and a forget gate, that function collaboratively to process and preserve information over temporal periods [26].

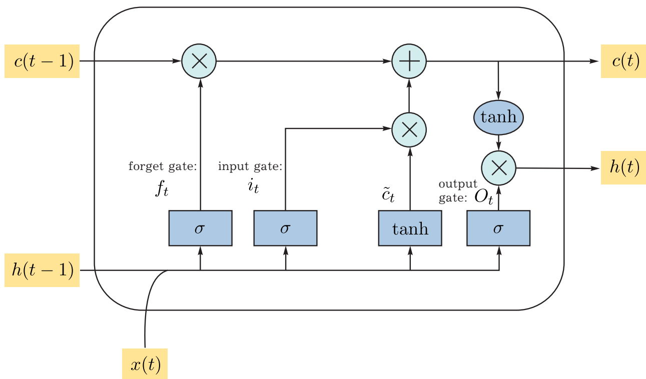

Fig. 1. Architecture of the LSTM unit.

The forget gate f_t dictates the degree to which the previous cell state c_{t-1} is discarded, computed as $f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right)$. The input gate i_t determines what new information is retained in the cell state, given by $i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right)$, and the candidate cell state, $\widetilde{c}_t$ is calculated using $\widetilde{c}_t = \tanh\left(W_c \cdot [h_{t-1}, x_t] + b_c\right)$, with hyperbolic tangent function. The current cell state C_t is updated using the formula $C_t = f_t \odot c_{t-1} + i_t \odot \widetilde{c}_t$. Finally, the output gate O_t governs the output of the LSTM cell, calculated as $O_t = \sigma\left(W_O \cdot [h_{t-1}, x_t] + b_0\right)$, and the hidden state, h_t is derived from $h_t = O_t \odot \tanh\left(c_t\right)$. Here, W_f, W_i, W_c and W_O represents the weight matrices that connect the input to the respective gates while b_f, b_i, b_c and b_O represents the bias terms that are added to the respective gates to adjust the activations, helping the LSTM model account for inherent shifts in the data and improve the gating mechanism's flexibility. The Hadamard product, $\odot$ is an element-wise multiplication, allowing the model to selectively control the flow of information. The σ (sigma) symbol represents the sigmoid activation function used in the input, forget, and output gates to control the flow of information by producing values between 0 and 1, determining how much information should be passed through or discarded.

2.4. LSTM model with AdamVol optimizer (LSTM-AdamVol)

The Adam optimization algorithm has gained widespread adoption owing to its adaptive learning rate mechanism, that integrates the benefits of both AdaGrad and RMSProp techniques [9]. The learning rate is dynamically adjusted based on the first and second moments of the gradients denoted as

$$\theta_{t+1} = \theta_t - \alpha \cdot \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \varepsilon}, \tag{2}$$

where θ represents the parameters, α represents the learning rate, $\hat{m}_t$ represents the bias-corrected first moment estimate, $\hat{v}_t$ represents the bias-corrected second moment estimate.

The proposed AdamVol optimizer incorporates market volatility into the learning rate adjustment, denoted as follows:

$$\alpha_t = \alpha \times (1 + \eta \times \mathrm{ATR}), \tag{3}$$

where α represents the base learning rate, η represents the adjustment factor, ATR represents the Average True Range.

The initial learning rate is represented by α. This is the base learning rate that the model would use if there were no volatility adjustment. This value is typically determined through standard hyperparameter tuning, starting with a base value, commonly between 0.001 and 0.01 for LSTM models. The ATR is used to quantify the recent market volatility. A higher ATR value indicates higher volatility. η is the sensitivity factor, determining how much the volatility (as represented by ATR) affects the learning rate. If η is set to a higher value, small changes in volatility will lead to larger changes in the learning rate. The value of η can be chosen empirically by observing how different values affect the performance during model training. Cross-validation is performed to find the η that provides the best trade-off between rapid adaptation and stable convergence. The term $(1 + \eta \times \mathrm{ATR})$ adjusts the learning rate based on the most recent volatility. When ATR is high, the learning rate increases to help the model quickly adapt to rapid market changes, and when ATR is low, the learning rate remains closer to the base α value, allowing the model to make more stable, incremental adjustments.

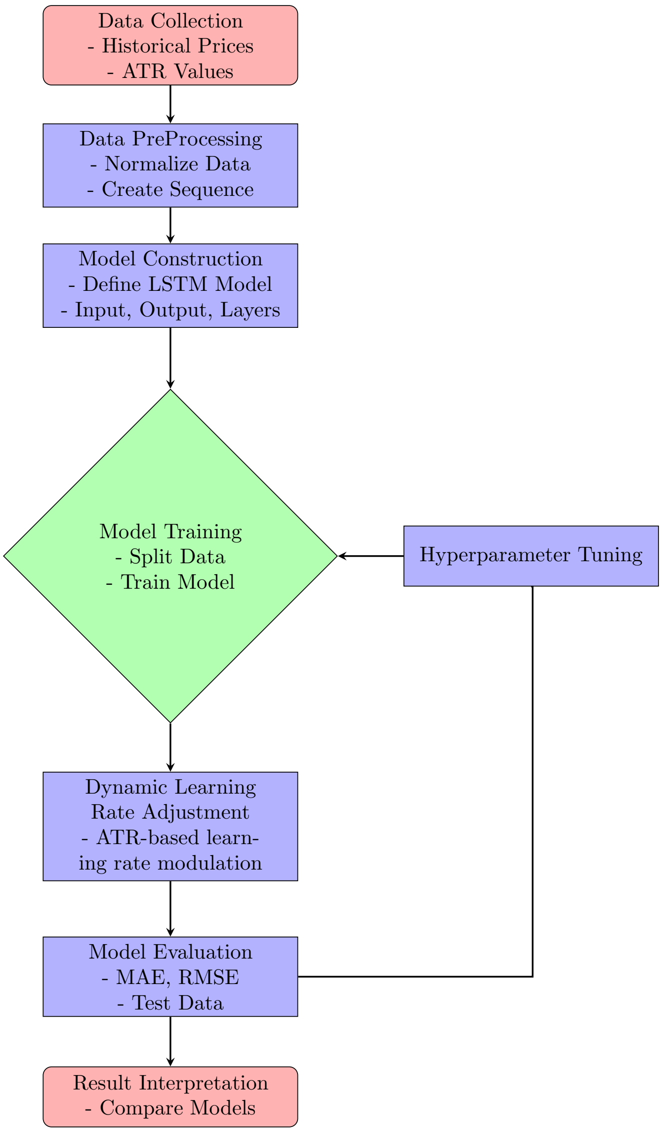

Fig. 2. Workflow of the LSTM model with AdamVol optimizer.

This adjustment allows the learning rate to dynamically respond to market conditions, ensuring that the learning process is more adaptive and sensitive to the current state of the financial market. Specifically, during periods of high volatility, the learning rate can be adjusted to be more conservative, reducing the risk of large, erratic updates that could destabilize the learning process. Conversely, during periods of low volatility, the learning rate can be increased to accelerate the learning process, taking advantage of the relatively stable market conditions to make more significant updates to the model's parameters [18].

The process of incorporating volatility into the LSTM model through dynamic learning rate can be visualized in Figure 2.

2.5. Performance evaluation

The forecasting models were evaluated using two performance measures: Mean Absolute Error (MAE) and Root Mean Square Error (RMSE),

$$\mathrm{MAE} = \frac{1}{n} \sum_{t=1}^{n} \left| y_t - \widehat{y}_t \right|, \tag{4}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{n} \sum_{t=1}^{n} \left(y_t - \widehat{y}_t \right)^2}. \tag{5}$$

The total number of observations is represented by n, where y_t denotes the actual observed value and $\widehat{y}_t$ refers to the forecasted value. These performance metrics will be utilized to evaluate the forecasted values generated by the LSTM model for the out-of-sample data.

3. Results and discussion

3.1. Volatility analysis and market movement

The analysis of the ATR values alongside the closing price reveals important insights into the relationship between volatility and stock price behaviour. As shown in Figure 3, the closing price (blue

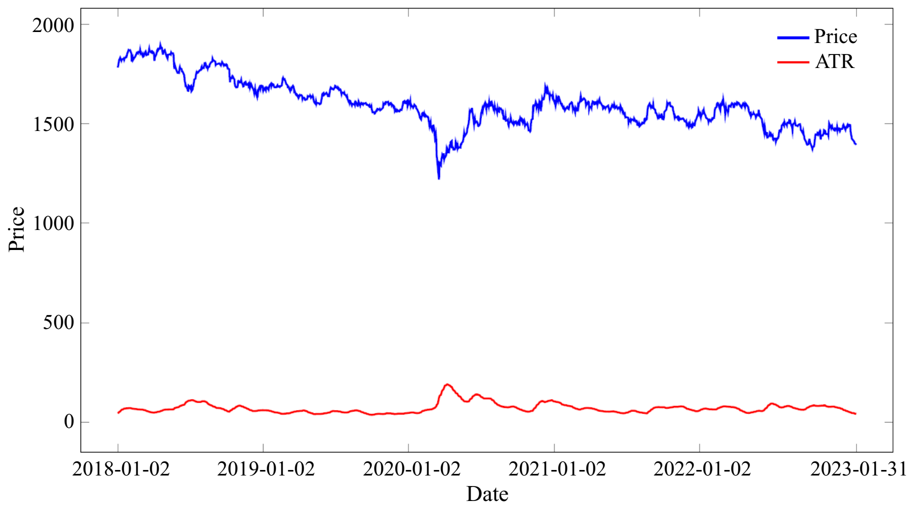

Fig. 3. Price and ATR values over trading dates.

line) fluctuates throughout the period, with a sharp decline observed around early 2020, likely corresponding to the global market crash due to the COVID-19 pandemic. In contrast, the ATR values (red line), which measure market volatility, remain relatively low during periods of stability but show a significant spike during this market downturn. This indicates that volatility increases when the market experiences large price movements, particularly during periods of uncertainty. After the sharp rise in ATR during early 2020, the volatility values gradually decrease, reflecting market stabilization. This reinforces the rationale for incorporating ATR-based weighting into the LSTM model, as it allows the model to better account for volatility-driven market shifts.

3.2. Hyperparameter setting on LSTM and LSTM-AdamVol model

The hyper-parameter values were empirically tuned through a iterative process in order to achieve optimal model performance during the training phase. Following multiple experimental iterations, the hyper-parameter settings for both the LSTM and LSTM-AdamVol models include one hidden layer with 200 hidden neurons. A dropout rate of 0.2 is applied to prevent overfitting. The models use a timestep of 10, a batch size of 32, and are trained for 100 epochs. The activation function employed is the hyperbolic tangent (tanh), while the recurrent function is the sigmoid function. The mean squared error serves as the loss function. The Adam (for standard LSTM) and AdamVol optimizer are leveraged to adjust the model parameters during the training phase.

The MAE and RMSE were calculated using the specified hyper-parameter configurations to evaluate the predictive accuracy of the LSTM-AdamVol models on the testing data. These performance metrics were also compared to those of other benchmark models, including ARIMA, artificial neural networks (ANN), and a standard LSTM model, as presented in Table 1.

Table 1. Forecasting performance of LSTM-AdamVol and other models.

Model	MAE Training	RMSE Training	MAE Testing	RMSE Testing
ARIMA (0,1,0)	8.8733	12.262	9.1305	11.9292
ANN (3,1,1)	8.8729	12.2626	8.9962	11.6962
Standard LSTM	8.8053	12.1604	8.9870	11.6266
LSTM-AdamVol	**8.7795**	**12.0178**	**8.9411**	**9.2053**

As reported in Table 1, the study's findings are consistent with previous research by [27–29] which have also recognized the LSTM model as the most effective choice for forecasting the KLCI, followed by the ANN and ARIMA models. Table 1 also shows that the LSTM-AdamVol model has recorded the smallest value for the evaluation metrics. The experimental results show that the LSTM-AdamVol model outperforms the standard LSTM, ANN, and ARIMA in terms of both training and testing performance.

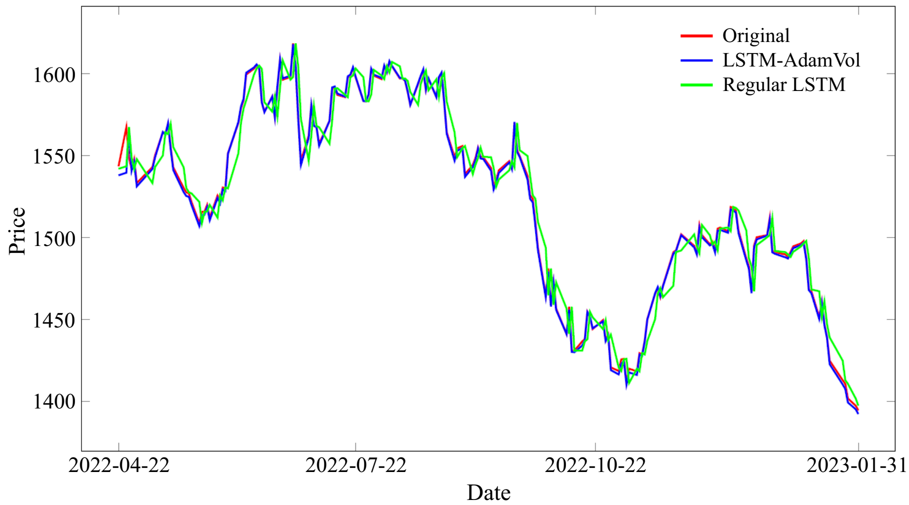

Fig. 4. Comparison of Original, LSTM-AdamVol, and Regular LSTM.

Figure 4 displays the original closing prices alongside the predictions from both the LSTM-AdamVol and the regular LSTM models, based on the testing data. The frequent alignment between the original data and the LSTM-AdamVol predictions in the graph underscores the superior forecasting performance of the proposed model in comparison to the regular LSTM. The KLCI index underwent a notable decrease in its volatility levels during the year 2022, which can be attributed to extraneous influences, such as political upheaval and economic recovery phase [19], as illustrated in Figure 4. This dynamic adjustment mechanism in LSTM-AdamVol model leverages real-time market data to continuously fine-tune the learning rate, making the model more resilient to the inherent fluctuations in financial markets. This flexibility makes the model more robust, as it can adjust its learning pace according to the market's volatility. This results in a more balanced and efficient learning process, capturing the underlying market trends more accurately.

This methodology adheres to the principles of adaptive learning, in which the learning parameters are adjusted in accordance with the inherent attributes of the data, culminating in more efficient and effective training procedures. By continuously adapting the learning rate, the model can maintain high performance and reliability, even as market conditions change. This dynamic adjustment not only augments the model's forecasting precision but also improves its ability to generalize across different market regimes, providing a powerful tool for financial forecasting and analysis.

4. Conclusion

In conclusion, this study has underscored the significance of integrating volatility into the learning rate adjustment of LSTM models for financial forecasting. This dynamic mechanism, which we term AdamVol, leverages real-time volatility data to fine-tune the learning rate, allowing the model to adapt to changing market conditions more effectively. Our results demonstrate that this method improves the model's robustness and predictive accuracy, particularly in the volatile environment of the KLCI. By aligning the learning rate with market dynamics, we achieve a more resilient and responsive training process, reducing the risk of overfitting and underfitting. This research underscores the potential of adaptive learning techniques in financial modeling, offering a valuable instrument that assists investors and analysists in effectively navigating the intricate dynamics of financial markets. Future work could explore the integration of other market indicators into the learning rate adjustment mechanism, further enhancing the model's adaptability and performance.

Acknowledgement

The authors acknowledged the Ministry of Higher Education Malaysia and Universiti Malaysia Sarawak for financial support.

[1] Idrees S. M., Alam M. A., Agarwal P. A prediction approach for stock market volatility based on time series data. IEEE Access. **7**, 17287–17298 (2019).

[2] Van Houdt G., Mosquera C., Nápoles G. A review on the long short-term memory model. Artificial Intelligence Review. **53** (8), 5929–5955 (2020).

[3] Al-Selwi S. M., Hassan M. F., Abdulkadir S. J., Muneer A., Sumiea E. H., Alqushaibi A., Ragab M. G. RNN-LSTM: From applications to modeling techniques and beyond-Systematic review. Journal of King Saud University – Computer and Information Sciences. **36** (5), 102068 (2024).

[4] Nacson M. S., Srebro N., Soudry D. Stochastic gradient descent on separable data: Exact convergence with a fixed learning rate. Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS). **PMLR 89**, 3051–3059 (2019).

[5] Huang H., Huang X., Ding W., Zhang S., Pang J. Optimization of electric vehicle sound package based on LSTM with an adaptive learning rate forest and multiple-level multiple-object method. Mechanical Systems and Signal Processing. **187**, 109932 (2023).

[6] Yu C., Qi X., Ma H., He X., Wang C., Zhao Y. LLR: Learning learning rates by LSTM for training neural networks. Neurocomputing. **394**, 41–50 (2020).

[7] Iiduka H. Appropriate learning rates of adaptive learning rate optimization algorithms for training deep neural networks. IEEE Transactions on Cybernetics. **52** (12), 13250–13261 (2021).

[8] Loizou N., Vaswani S., Laradji I. H., Lacoste-Julien S. Stochastic Polyak Step-size for SGD: An Adaptive Learning Rate for Fast Convergence. Proceedings of Machine Learning Research. **130**, 1306–1314 (2021).

[9] Kingma D., Ba J. Adam: A Method for Stochastic Optimization. Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015), San Diego (2015).

[10] Tieleman T., Hinton G. Lecture 6.5-rmsprop: Divide the Gradient by a Running Average of Its Recent Magnitude. COURSERA: Neural Networks for Machine Learning. **4**, 26–31 (2012).

[11] Duchi J., Hazan E., Singer Y. Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research. **12** (61), 2121–2159 (2011).

[12] Zeiler M. D. ADADELTA: An Adaptive Learning Rate Method. Preprint arXiv:1212.5701 (2012).

[13] Loshchilov I., Hutter F. Decoupled Weight Decay Regularization. Preprint arXiv:1711.05101 (2017).

[14] Dozat T. Incorporating Nesterov momentum into Adam. Proceedings of 4th International Conference on Learning Representations (ICLR), Workshop Track. 1–4 (2016).

[15] Xie Z., Wang X., Zhang H., Sato I., Sugiyama M. Adaptive inertia: Disentangling the effects of adaptive learning rate and momentum. International Conference on Machine Learning. 24430–24459 (2022).

[16] Tong Q., Liang G., Bi J. Calibrating the adaptive learning rate to improve convergence of ADAM. Neurocomputing. **481**, 333–356 (2022).

[17] Hao X., Ma Y., Pan D. Geopolitical risk and the predictability of spillovers between exchange, commodity and stock markets. Journal of Multinational Financial Management. **73**, 100843 (2024).

[18] Zhang F., Zhang Y., Xu Y., Chen Y. Dynamic relationship between volume and volatility in the Chinese stock market: Evidence from the MS-VAR model. Data Science and Management. **7** (1), 17–24 (2024).

[19] Shakawi A. M. H. A., Shabri A. Improving Prediction of Bursa Malaysia Stock Index Using Time Series and Deep Learning Hybrid Model. Advances in Intelligent Computing Techniques and Applications. 119–128 (2024).

[20] Liang C., Wang L., Duong D. More attention and better volatility forecast accuracy: How does war attention affect stock volatility predictability? Journal of Economic Behavior & Organization. **218**, 1–19 (2024).

[21] Jepkoech J., Mugo D. M., Kenduiywo B. K., Too E. C. The effect of adaptive learning rate on the accuracy of neural networks. International Journal of Advanced Computer Science and Applications. **12** (8), 736–751 (2021).

[22] Park J., Yi D., Ji S. A novel learning rate schedule in optimization for neural networks and it's convergence. Symmetry. **12** (4), 660 (2020).

[23] Wilder J. W. New concepts in technical trading systems. Greensboro, NC: Trend Research (1978).

[24] Atkins A., Niranjan M., Gerding E. Financial news predicts stock market volatility better than close price. The Journal of Finance and Data Science. **4** (2), 120–137 (2018).

[25] Mienye I. D., Swart T. G., Obaido G. Recurrent Neural Networks: A Comprehensive Review of Architectures, Variants, and Applications. Information. **15** (9), 517 (2024).

[26] Hochreiter S., Schmidhuber J. Long Short-Term Memory. Neural Computation. **9** (8), 1735–1780 (1997).

[27] Ho M. K., Darman H., Musa S. Stock price prediction using ARIMA, neural network and LSTM models. Journal of Physics: Conference Series. **1988** (1), 012041 (2021).

[28] Malim T. N. A. B. T., Kamarudin S. A., Ahad N. A., Mamat N. A. M. G. Prediction of FTSE Bursa Malaysia KLCI Stock Market using LSTM Recurrent Neural Network. 2022 IEEE International Conference on Computing (ICOCO). 415–418 (2022).

[29] Khalil M. R. A., Bakar A. A. A Comparative Study of Deep Learning Algorithms in Univariate and Multivariate Forecasting of the Malaysian Stock Market. Sains Malaysiana. **52** (3), 993–1009 (2023).

Динамічне коригування швидкості навчання з використанням волатильності в моделях LSTM для прогнозування KLCI

Шакаві А. М. Х. А.[1,2], Шабрі А.[2]

[1] *Центр доуніверситетських досліджень, Університет Малайзії Саравак, 94300, Кота Самарахан, Саравак, Малайзія*
[2] *Кафедра математичних наук, факультет природничих наук, Університет технологій Малайзії, 81310, Джохор, Малайзія*

Прогнозування поведінки фінансового ринку є багатогранною проблемою, яка обумовлена основною волатильністю та нелінійними характеристиками, які властиві ринковим даним. Моделі довготривалої короткочасної пам'яті (LSTM) продемонстрували ефективність у врахуванні цих складнощів. У цьому дослідженні пропонується новий підхід до підвищення продуктивності моделі LSTM шляхом адаптивного модулювання швидкості навчання на основі волатильності ринку. Цей метод застосовується для прогнозування Kuala Lumpur Composite Index (KLCI), використовуючи волатильність як ключову вхідну інформацію для адаптації швидкості навчання під час тренування. Інтегруючи волатильність у процес навчання, модель може краще враховувати ринкові коливання, що потенційно може приведе до точніших і надійніших прогнозів. Запропонований механізм регулювання динамічної швидкості навчання працює шляхом масштабування швидкості навчання відповідно до останніх вимірювань волатильності, забезпечуючи швидку адаптацію моделі до мінливих умов ринку. Цей підхід контрастує з традиційними статичними швидкостями навчання, які можуть недостатньо врахувати динаміку фінансових ринків. Проведено масштабні експерименти, використовуючи історичні дані KLCI, порівнюючи запропоновану модель зі стандартною LSTM та іншими базовими моделями. Результати показують, що швидкості навчання з поправкою на волатильність перевершують звичайні моделі LSTM із фіксованими швидкостями навчання щодо прогнозної продуктивності та стабільності. Отримані дані свідчать про те, що включення волатильності в коригування швидкості навчання може значно підвищити передбачувані можливості моделей LSTM для прогнозування фондового ринку. Підвищена точність прогнозування індексу KLCI підкреслює потенціал цього підходу для ширшого застосування на фінансових ринках.

Ключові слова: *середній справжній діапазон; Бурса Малайзія; LSTM; машинне навчання; часові ряди; швидкість навчання з поправкою на волатильність.*