An intelligent system for forecasting time series based on a neural network with LSTM-blocks

Vasyl Lytvyn[1,†], Ivan Peleshchak[1,*,†], Yurii Futryk[1,*,†], Roman Peleshchak[1,†], Andriy Khudyy[1,†], Andriy Senyk[1,†]

[1] *Lviv Polytechnic National University, Stepana Bandery Street, 12, 79000, Lviv, Ukraine*

Abstract

Improving time series forecasting methods is a critical task for various industries, including finance, manufacturing, military applications, medicine, mine clearance processes, including time-based analysis of GPR and magnetometer data, and energy in the era of Industry 4.0. The use of recurrent neural networks with LSTM units is an effective approach for predicting long-term dependencies in data. However, the optimal configuration of the long short-term memory (LSTM) architecture remains an open question, in particular, the choice of the number of blocks, the dropout rate, and the use of technical indicators to improve prediction accuracy. This study presents a detailed analysis of the impact of key hyperparameters in LSTM models, including the number of blocks (100-400), the dropout rate (0.1-0.3), and the role of technical indicators such as EMA (Exponential Moving Average) and RSI (Relative Strength Index) in generating accurate forecasts. The obtained results show that EMA_20 has the highest correlation coefficient (0.99) with the closing price, while RSI demonstrates a weaker relationship (0.04-0.05), which emphasizes its secondary role. The training algorithm for the neural network with LSTM blocks was optimized using the Nadam optimizer, which allowed us to determine the most effective combination of hyperparameters for forecasting financial time series. The training data was obtained from the Yahoo Finance (yfinance) library and included historical data on the Google (GOOGL) stock price for the period from 2011 to 2024. The model performance was evaluated using the MSE, RMSE and MAPE metrics, which allowed us to objectively assess the level of forecasting accuracy. The analysis of the obtained results showed that the optimal configuration of the neural network consists of 350 LSTM units, a Dropout level of 0.05, and the Nadam optimizer. This configuration achieved a minimum average absolute percentage error (MAPE) of 1.64%, which is lower than the results obtained in previous studies. The study confirms that increasing the number of LSTM blocks beyond 350 does not improve accuracy and may lead to overfitting.

Keywords

LSTM recurrent neural network, time series forecasting, MAPE, Dropout, Nadam optimizer, technical indicators, EMA, RSI.

1. Introduction

Time series forecasting is a key task in the era of Industry 4.0, encompassing finance, marketing, energy, medicine, and landmine clearance processes, including time-series analysis of ground-penetrating radar and magnetometer data. One of the most effective tools in this domain is the Long Short-Term Memory (LSTM) recurrent neural network, which has the ability to retain long-term temporal dependencies between data points.

In the context of Intelligent Systems and Technologies in Industry, LSTM models are actively utilized for early failure detection in industrial equipment, forecasting production line loads, identifying anomalous patterns in sensor data, and enhancing the efficiency of logistics management. Conversely, in humanitarian and military fields, LSTM models are applied to mine clearance operations, including time-series analysis of ground-penetrating radar and magnetometer data, forecasting potential landmine displacement due to weather and geological factors, and optimizing

[1]SMARTINDUSTRY-2025: 2nd International Conference on Smart Automation & Robotics for Future Industry, April 03-05, 2025, Lviv, Ukraine

* Corresponding author.
† These authors contributed equally.
vasyl.v.lytvyn@lpnu.ua (V. Lytvyn); ivan.r.peleshchak@lpnu.ua (I. Peleshchak); yurii.v.futryk@lpnu.ua (Y. Futryk); roman.m.peleshchak@lpnu.ua (R. Peleshchak); khudyy@ukr.net (A. Khudyy); andrij.p.senyk@lpnu.ua (A. Senyk)
0000-0002-9676-0180 (V. Lytvyn); 0000-0002-7481-8628 (I. Peleshchak); 0000-0001-5271-9883 (Y. Futryk); 0000-0002-0536-3252 (R. Peleshchak); 0000-0003-2029-7270 (A. Khudyy); 0000-0002-1614-512X (A. Senyk)

resource allocation in demining missions. Due to LSTM's ability to retain long-term temporal contexts, these tasks can be solved with greater accuracy and efficiency.

2. Review and analysis of recent studies

Recurrent neural networks (RNNs) incorporating Long Short-Term Memory (LSTM) units are extensively employed for time series prediction in modern computational research [1]. Despite significant progress in this field, multiple unresolved challenges persist. The application of LSTM-based models for forecasting sequential data remains a pivotal area in contemporary neural engineering [2]. While various architectural modifications of LSTM networks yield promising outcomes, certain aspects of model optimization still require refinement.

The study in [3] explores a methodology where stacked autoencoders are combined with LSTM layers to improve financial time series predictions. The authors report a forecasting precision of MAPE ≈ 2.2%. However, the model demonstrates constrained adaptability to fluctuations in market conditions due to the absence of technical indicators during the training phase.

In [4], an LSTM model is enhanced with an attention mechanism, enabling the neural network to prioritize significant temporal markers. While this modification improves forecast accuracy (MAPE ≈ 2.0%), the model lacks optimization in terms of computational efficiency and does not account for the impact of Dropout regularization on training stability.

The work presented in [5] examines the integration of technical indicators such as the Exponential Moving Average (EMA) and the Relative Strength Index (RSI) into LSTM-based stock price forecasting models. Despite achieving a forecasting accuracy of MAPE ≈ 1.95%, the model does not incorporate adaptive optimization algorithms, particularly Nadam, which could improve performance in volatile financial markets.

A comparative evaluation of LSTM and Gated Recurrent Unit (GRU) models is conducted in [6], where the authors establish that LSTM-based architectures are more effective for financial predictions, reaching MAPE ≈ 1.9%. However, this study does not explore the influence of varying LSTM block counts or Dropout levels on model robustness.

In [8], researchers assess the effectiveness of low-value Dropout regularization in improving LSTM model stability. Although the study reports a forecasting precision of MAPE ≈ 1.9%, it does not consider the impact of LSTM block count variation and adaptive loss functions on overall prediction reliability.

Findings from studies [9, 10] suggest that machine learning methodologies are progressively supplanting traditional forecasting models, particularly in cases involving extensive datasets with complex interdependencies. Ensemble learning techniques such as Random Forest and XGBoost, alongside regression-based models incorporating supplementary features (e.g., lagging indicators, seasonality, and market anomalies), have demonstrated enhanced predictive accuracy. However, the challenge of selecting optimal hyperparameters and ensuring model interpretability remains.

Several studies [11, 12] investigate hybrid modeling strategies that integrate statistical and neural network approaches (e.g., ARIMA+LSTM, SARIMA+MLP). These models leverage the explainability of statistical trend analysis while capitalizing on the adaptability of deep learning methods. However, the complexity of hybrid pipelines increases computational costs and necessitates extensive data preprocessing.

Research efforts detailed in [13, 14] focus on time series prediction, particularly the forecasting of Yahoo Finance stock price data using LSTM networks trained with Adam [13] and Nadam [14] optimization algorithms. The reported prediction accuracy, measured by MAPE, reached 1.9%.

It is noteworthy that Adam, as referenced in multiple studies, utilizes a conventional adaptive weight adjustment approach based on the mean squared gradients and their exponentially smoothed moments. However, Adam does not account for a "lookahead" gradient component. In contrast, the Nadam optimizer incorporates a projected gradient update mechanism, adjusting weights based on anticipated rather than current gradient values. This methodological distinction enables Nadam to enhance the accuracy of LSTM-based time series forecasting. The primary factors influencing forecasting performance include the number of LSTM units, the level of Dropout regularization, and the incorporation of additional technical indicators such as EMA and RSI.

3. The purpose and objectives of the research

A review of existing literature [1–14] reveals that LSTM model accuracy is typically constrained by a Mean Absolute Percentage Error (MAPE) exceeding 1.9%, indicating the need for further advancements in both network architecture and training methodologies. Several key challenges persist in time series forecasting using LSTM-based neural networks:

- Development of an optimal model architecture and training algorithm with different LSTM-block configurations (100, 200, 300, 400) and trained with the Nadam optimizer. The study aims to identify the most effective Dropout regularization value (0.1, 0.2, 0.3) to prevent overfitting while balancing generalization and accuracy.
- Impact of technical indicators such as the Exponential Moving Average (EMA_20) and the Relative Strength Index (RSI) on forecasting accuracy, as these indicators help smooth stochastic fluctuations in data;
- Evaluation of model accuracy using metrics such as MSE, RMSE and MAPE.

Thus, the critical issue is designing an optimal LSTM architecture configuration and training algorithm using the Nadam optimizer, selecting the appropriate number of blocks, tuning the Dropout level, and incorporating technical indicators to enhance time series forecasting accuracy.

To achieve this objective, the study focuses on solving the following tasks:

- Determining the optimal number of LSTM-blocks in a sequential network to achieve a forecasting error below 1.8%;
- Optimizing the training process of the LSTM neural network using the Nadam optimizer;
- Investigating the role of Dropout regularization in improving the predictive performance of LSTM models;
- Analyzing the impact of EMA and RSI on forecasting accuracy;
- Assessing model accuracy using evaluation metrics such as MSE, RMSE and MAPE.

4. Neural network architecture with LSTM-blocks

Fig 1. illustrates the architectural design of a recurrent neural network (RNN) [15], structured with sequentially connected LSTM units. Each LSTM block processes data at a discrete time step, accumulating contextual information through inter-block interactions. This architecture enables the network to refine both its output predictions and internal state updates, ensuring greater accuracy in long-term forecasting.

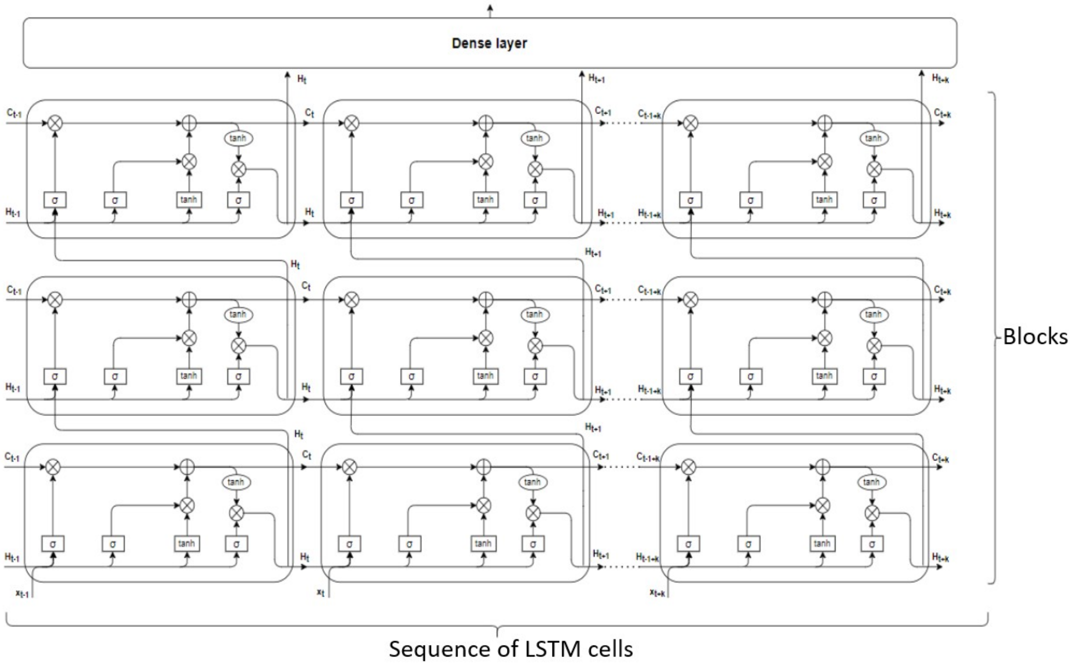


Figure 1: Morphology of a neural network with LSTM-blocks.

The processed output from LSTM blocks is subsequently fed into a fully connected (Dense) layer, responsible for generating final prediction outputs. Depending on the specific network configuration, activation functions in LSTM layers may include tanh or sigmoid, while the output Dense layer employs linear or ReLU, depending on the necessity for output scaling.

On Fig. 2 illustrates the internal structure of an individual LSTM unit utilized in this research.

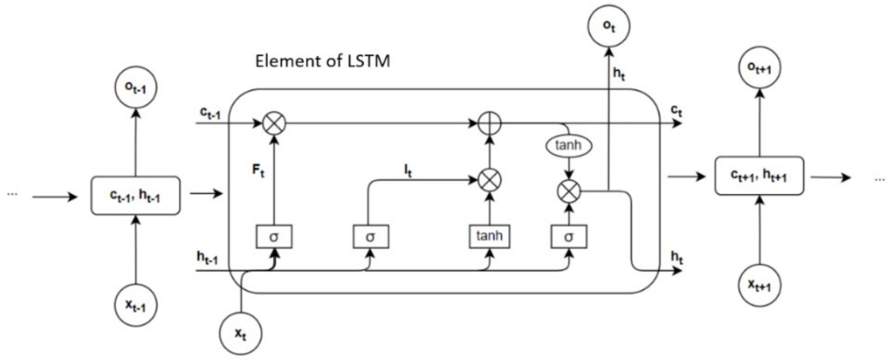


Figure 2: LSTM unit architecture.

1. Forget gate — This step allows the network to decide which elements of the current memory cell should be discarded based on the forget gate layer. The decision-making process is governed by the sigmoid activation functionσ, which evaluates information from the previous hidden state h_{t-1} and the current time step x_t. The output value ranges between 0 (indicating data removal) and 1 (indicating retention), ensuring selective memory update mechanisms cell C_{t-1}.

$$f_t = \sigma\left(W_t * \left[h_{t-1}, x_t\right] + b_f\right), \quad (4)$$

where:

f_t – represents a vector of values in the range [0,1], specifying the fraction of retained or discarded information within the memory cell.

W_t, b_f – denote trainable weight matrices and bias vectors, updated dynamically during the training process.

σ – denotes the sigmoid activation function used to regulate information flow.

2. Input gate — At this stage, the network decides whether to store new information in the current memory state. Specifically, it determines which input values should be used to modify memory. The network utilizes the previous hidden state and the sequence value at the current time step, applying them to a sigmoid function. This process consists of two layers: sigmoid activation layer that determines which values can be updated and tanh activation layer that generates a vector of new candidate values for updating the memory cell

$$i_t = \sigma\left(W_i * \left[h_{t-1}, x_t\right] + b_i\right), \tag{5}$$

$$C_t = \tan h\left(W_C * \left[h_{t-1}, x_t\right] + b_C\right), \tag{6}$$

where:

i_t – represents activation vector that determines memory updates.

C_t – represents vector of new candidate values.

3. Output gate — At this stage, the network processes previously computed and stored information to generate a new hidden state, deciding what will be returned as the output of the current memory cell. First, a sigmoid activation layer determines which part of the current state should be output. Then, the tanh function is applied to compute the candidates for the output state. Finally, all layers' results are combined, and only the relevant information is returned.

$$o_t = \sigma\left(W_0 * \left[h_{t-1}, x_t\right] + b_0\right), \tag{7}$$

$$h_t = o_t * \tan h\left(C_t\right), \tag{8}$$

where:

o_t – represents vector of output signal values.

h_t – represents updated hidden state, which is passed to the next time step.

The LSTM-block structure [14] is designed to efficiently preserve long-term dependencies by combining mechanisms for storing essential information and filtering out irrelevant data. This makes LSTM-blocks one of the most effective tools for time series forecasting.

The core idea is that the learning process occurs within a memory-based context. The network forgets, learns, and extracts relevant parts of the information for the next step. However, at the next step, the same process is repeated again. Essentially, this approach attempts to mimic how the human brain learns and retains information through the internal gating mechanisms of LSTM (although this is not necessarily an exact representation, it is an attempt to apply different strategies to improve learning).

5. LSTM model evaluation metrics

The performance of LSTM models was assessed using the following metrics [17]:

Mean Squared Error (MSE) – is an indicator that reflects the average of the squared differences between the actual and predicted values. The use of error squares amplifies the impact of large deviations, making this metric sensitive to significant errors. Formula for calculation MSE:

$$MSE = \frac{1}{n}\sum_{i=0}^{n}\left(y_i - \hat{y}_i\right)^2, \tag{1}$$

Root mean squared error (RMSE) – is the square root of MSE, which allows the error to be expressed in the same units as the actual values. This metric provides an intuitive interpretation of forecasting accuracy. The RMSE formula is:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=0}^{n}\left(y_i - \hat{y}_i\right)^2},$$
(2)

Mean absolute percentage error (MAPE) – measures the average percentage deviation between actual and predicted values. This metric evaluates forecasting accuracy independently of scale. The MAPE formula is:

$$MAPE = \frac{1}{n}\sum_{i=0}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| * 100\,\%.,$$
(3)

6. Computer experiment and analysis of results

The computer experiment on time series forecasting was conducted based on a neural network algorithm with LSTM-blocks and the Nadam optimizer, represented in the block diagram in Fig. 3.

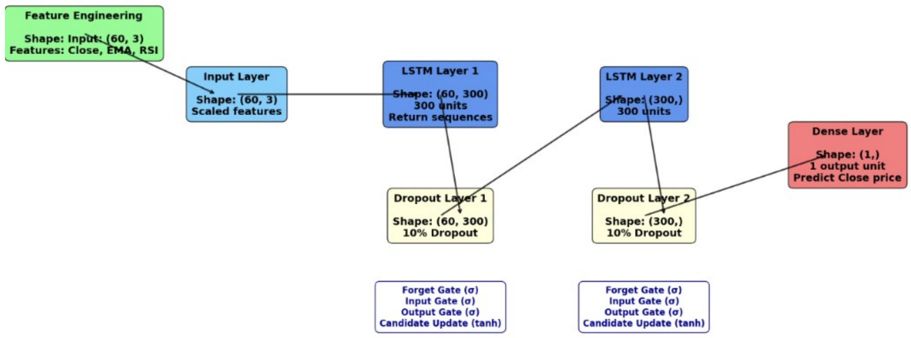


Figure 3: Block diagram of the neural network algorithm with different numbers of LSTM-blocks and hyperparameter configurations (Case A: 300 LSTM-blocks, Dropout = 0.1; Case B: 350 LSTM-blocks, Dropout = 0.05) for time series forecasting.

The experiment was performed for two different sets of hyperparameters in the LSTM-based neural network.

Case A: First set of parameters:

- Number of LSTM-blocks: 300
- Dropout: 0.1
- Optimizer: Nadam
- Technical indicators: Exponential Moving Average (EMA) and Relative Strength Index (RSI) were included to enhance the understanding of market trends.

Case B: Second set of parameters:

- Number of LSTM-blocks: 350
- Dropout: 0.05
- Optimizer: Nadam
- Technical indicators: Exponential Moving Average (EMA) and Relative Strength Index (RSI) were included to enhance the understanding of market trends.

To train the LSTM-based neural network, a dataset was obtained from the publicly available Yahoo Finance source using the yfinance library. The experiment focused on Google (GOOGL) stock market data, forming a dataset that spans January 2011 to January 2024. In total, the analyzed time period includes 4,748 calendar days, while the dataset contains 3,270 records. Each record contains the following characteristics:

- Open – the opening price of the stock at the beginning of the trading day;
- High – the highest price recorded during the day;
- Low – the lowest price at which the stock was traded during the day;
- Close – the closing price of a stock at the end of the trading session;
- Adj Close – is the adjusted closing price taking into account corporate events;
- Volume – the volume of shares that were sold or bought during the day.

The data was retrieved using the yfinance library, which provides a convenient interface for accessing financial data. For this experiment, the "Close" column was selected as it is the most relevant for time series analysis and forecasting.

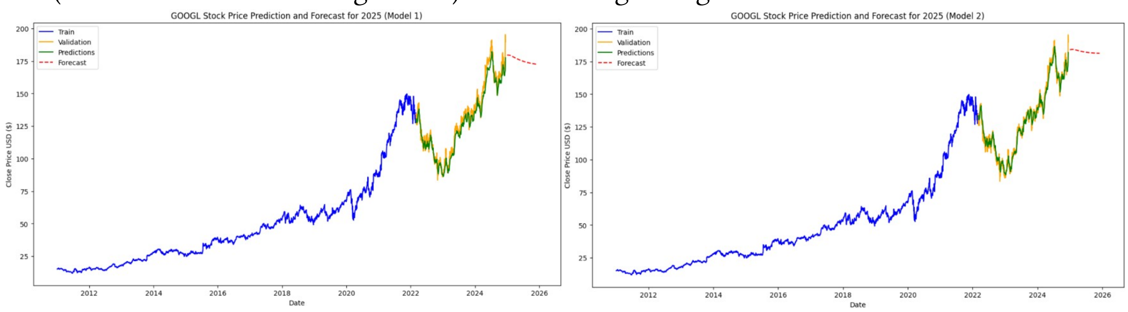
6.1. Data preparation and neural network training

The model was implemented in Python programming language, using the following libraries:

- NumPy – for efficient numerical computations.
- pandas – for handling tabular data and preprocessing.
- Matplotlib – for graphical visualization of results.
- keras – a framework for developing and training neural networks.
- yfinance – a library for retrieving historical financial data.

The dataset was divided into training and test samples in the ratio of 80:20. The Min-Max Scaling method was used to normalize the values. The dataset was prepared in such a way that each forecast was formed on the basis of 60 previous time points. The scaling was performed using the Min-Max Scaler, which contributes to stable model training.

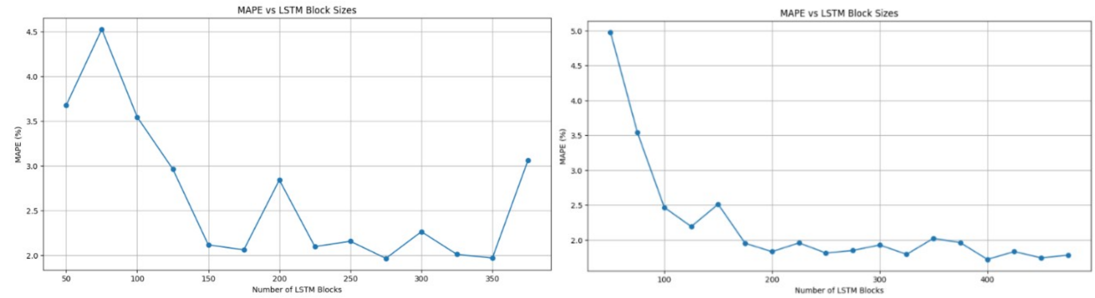
Fig. 4 (a, b) presents the results of the computer experiment on stock price forecasting using an LSTM-based neural network with the first (Case A) and second (Case B) sets of hyperparameters. Fig. 5 (a, b) illustrates the dependency of MAPE metric values on the number of LSTM-blocks for Cases A and B. Fig. 6 (a, b) visualizes the MSE (Mean Squared Error), RMSE (Root Mean Squared Error), and MAPE (Mean Absolute Percentage Error) metrics using histograms.



a. Case A; b. Case B;

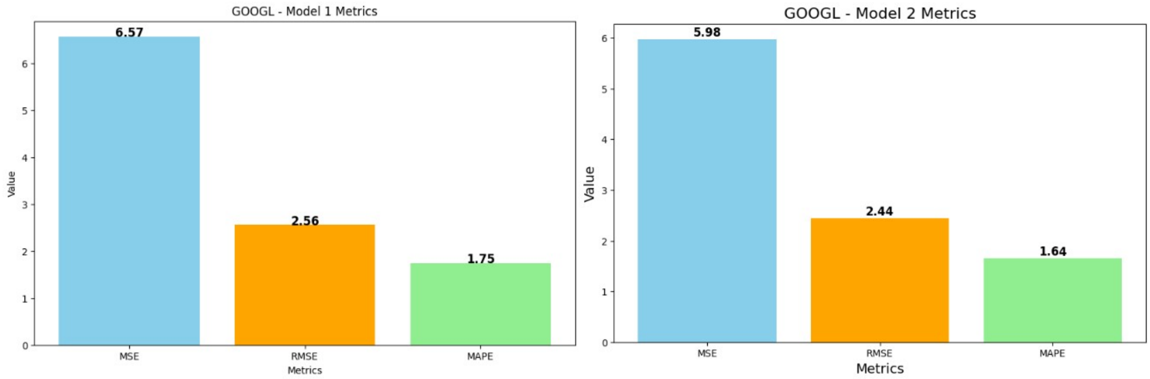
Figure 4 (a, b): Stock price forecast graphs for the first and second sets of parameters.

- Blue line: Training data.
- Yellow line: Validation data.
- Green line: Predicted values on validation.



a. Case A; b. Case B;

Figure 5 (a, b): Dependency of MAPE metric values on the number of LSTM-blocks for cases A and B.



a. Case A – for 275 LSTM-blocks b. Case B – for 325 LSTM-blocks

Figure 6 (a, b): Visualization of MSE, RMSE and MAPE metrics using histograms.

Significantly higher MAPE values were observed when using fewer LSTM-blocks (50–100). A more gradual decrease in MAPE with an increasing number of LSTM-blocks indicates the stability of the neural network.

The network morphology in Case A is well-suited for stock price forecasting due to its low error rate and stable predictions. The choice of 300 LSTM-blocks and Dropout 0.1 ensures an optimal balance between model complexity and overfitting prevention. However, despite achieving good accuracy (MAPE = 1.75%), this neural network lags behind Case B in performance.

The network morphology in Case B (350 LSTM-blocks, Dropout 0.05) demonstrated a consistent MAPE value across 180–400 LSTM-blocks, indicating strong generalization capability. In particular, the Case A configuration (300 LSTM-blocks, Dropout 0.1) resulted in minimal MAPE values (~1.64%). Reducing the Dropout rate to 0.05 in Case B helped prevent excessive weight nullification, positively impacting prediction accuracy. This neural network (Case B) outperformed the model from Case A, showing higher efficiency but also greater sensitivity to overfitting with a large number of LSTM-blocks (>350). The lowest MAPE value was observed in Case B with 325 LSTM-blocks, indicating that this configuration achieved the highest accuracy and stability. Thus, the best-performing network had 325 LSTM-blocks and a Dropout of 0.05, yielding the lowest MAPE (~1.64%) while minimizing the risk of overfitting.

Histograms (Fig. 6 a, b) provide a visual representation of forecasting accuracy for the neural network models across three key metrics: MSE (Mean Squared Error), RMSE (Root Mean Squared Error) and MAPE (Mean Absolute Percentage Error).

For the first set of hyperparameters (Case A):

- MSE = 6.57
- RMSE = 2.56
- MAPE = 1.75%

For the second set of hyperparameters (Case B):

- MSE = 5.98
- RMSE = 2.44
- MAPE = 1.64%

The experimental results confirmed that using LSTM-based neural networks significantly improves time series forecasting accuracy compared to traditional approaches such as ARIMA and GRU.

In particular, the integration of technical indicators (EMA and RSI) with adaptive optimizers enabled the models to better adapt to complex fluctuations in time series, resulting in more accurate predictions.

Advantages of the proposed LSTM-based neural network in case B configuration:

- High accuracy: The MAPE of 1.64% confirms the model's ability to provide precise forecasts, even in the presence of stochastic fluctuations in the time series.
- Robustness to noise: The use of technical indicators allows the model to process time series data more effectively.
- Flexibility: The ability to adjust the Dropout rate and the number of LSTM-blocks makes the model adaptable to different forecasting tasks and datasets.

6.2. Analysis of the impact of the number of LSTM-blocks and Dropout

During the experiments, the impact of different LSTM-block counts (100, 200, 300, 400) and Dropout levels (0.1, 0.2, 0.3) on the accuracy of Google stock price forecasting was analyzed.

Dropout is a regularization method [18] used to prevent overfitting in a neural network by randomly deactivating (zeroing out) a portion of neurons during training. In our LSTM model, Dropout was applied twice:

After the first LSTM layer (Dropout Layer 1):

- 10% of neurons were deactivated before passing to the next LSTM layer.
- This reduces dependencies between neurons and improves the network’s generalization ability.

After the second LSTM layer (Dropout Layer 2):

- Another 10% of neurons were deactivated before passing to the final Dense layer.
- This helps prevent excessive adaptation of the model to the training data.

Thus, Dropout improves the model’s robustness to input data variability and enhances its generalization on test data.

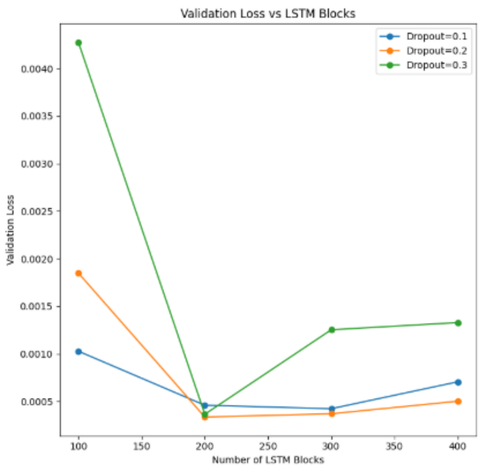


Figure 7: Dependence of validation loss on the number of LSTM-blocks.

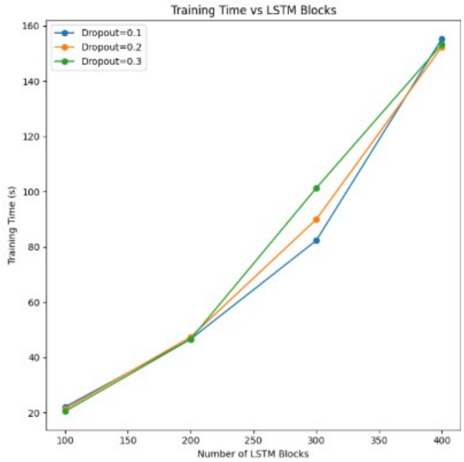


Figure 8: Dependence of training time on the number of LSTM-blocks.

1. Validation Loss vs. LSTM-blocks (Fig. 7) – This graph illustrates the change in validation loss depending on the number of LSTM-blocks and Dropout values:

- Increasing the number of LSTM-blocks up to 200 helps minimize validation loss.
- At 300 LSTM-blocks, the model exhibits the lowest loss values when Dropout 0.2, indicating optimal stability.
- Dropout 0.3 leads to significant fluctuations, suggesting weaker generalization ability of the model.

2. Training Time vs. LSTM-blocks (Fig. 8) – This graph shows that increasing the number of LSTM-blocks directly impacts the model's training time:

- There is an almost linear relationship between the number of LSTM-blocks and training time.
- Dropout has a minimal effect on training time, but minor differences can be observed when the number of blocks exceeds 200.

Fig. 9 presents a 3D-visualization of the stability analysis of LSTM models based on validation loss in the coordinates: number of LSTM-blocks and Dropout.

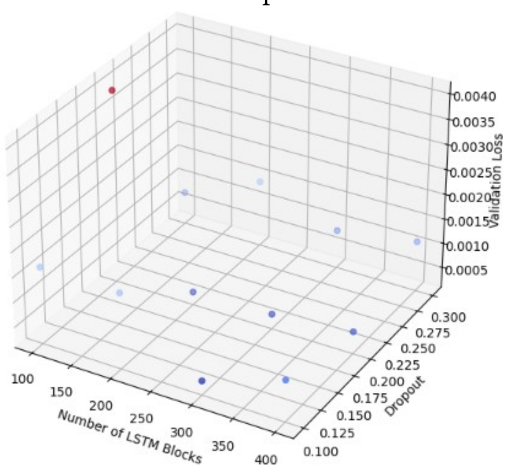


Figure 9: Validation losses by a neural network when predicting validation data in 3D-space.

This 3D graph illustrates validation loss variations depending on two key parameters – Dropout and the number of LSTM-blocks (Fig. 9). Red areas indicate higher loss values, observed at high Dropout values and a low number of LSTM-blocks. Light and deep blue areas indicate minimal loss values, particularly when the LSTM model has 300–350 blocks with Dropout = 0.05–0.1.

- Optimal Dropout values (0.05–0.1) significantly reduce validation loss during time series forecasting.

- The LSTM model with 300 blocks demonstrates the best performance in validation forecasting.

The analysis of experimental results confirms that proper tuning of LSTM parameters [19], specifically the number of blocks and Dropout, can significantly improve model accuracy and stability.

6.3. Impact of technical indicators (EMA, RSI) on forecast accuracy

3D-visualization of technical parameter influence on forecasting

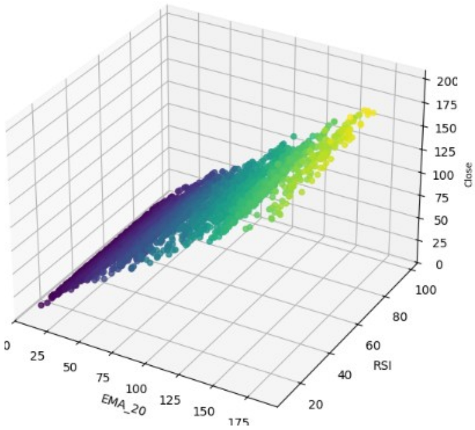


Figure 10: 3D-visualization of parameters: EMA, RSI, and Close Price.

A 3D graph (Fig. 10) was constructed to visualize the relationships between EMA_20, RSI, and the closing price:

- EMA_20 exhibits the most significant impact on predicted stock price changes.
- RSI shows a much more dispersed influence, confirming its secondary role in forecasting long-term trends.

Trend analysis of Close, EMA, and RSI for LSTM models

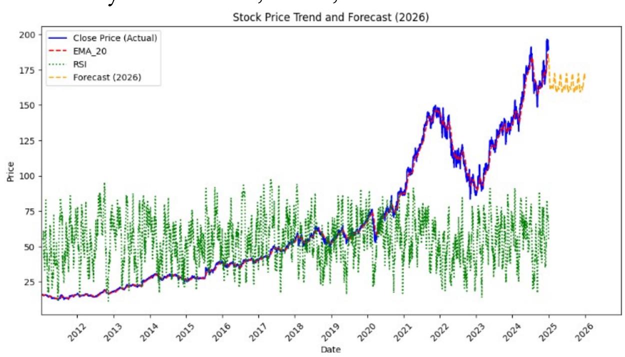


Figure 11: Graph of trend analysis for Close, EMA, and RSI parameters.

A comparison of LSTM models with different numbers of blocks revealed certain trends, as illustrated in Fig. 11:

- The EMA_20 indicator (red line) closely follows the Close price trend (blue line) with almost no deviations, confirming their strong correlation.
- The RSI indicator (green line) exhibits significant fluctuations, particularly during sharp price changes.

Thus:

- EMA_20 is well-suited for forecasting long-term trends.
- RSI is useful for short-term market fluctuation analysis, especially during periods of market instability.

7. Conclusions

The architecture of a neural network with LSTM-blocks and its training algorithm were optimized using the Nadam optimizer for time series forecasting and analysis of the impact of hyperparameters and technical indicators on forecasting accuracy.

A block diagram of the neural network algorithm with different numbers of LSTM-blocks and hyperparameter configurations was developed for time series forecasting.

Based on the conducted experiments, the role of the number of LSTM-blocks, Dropout levels, and technical indicators (EMA and RSI) was analyzed:

- The optimal configuration was achieved with 300–350 LSTM-blocks and Dropout values in the range of 0.05–0.1, minimizing prediction errors.
- EMA_20 was identified as the key predictor for closing price forecasting, whereas RSI exhibited a weaker correlation, but can be useful for detecting anomalous market movements.
- Training time increases linearly with the number of LSTM-blocks.
- The Nadam optimizer ensured stable model training.
- The EMA_20 trend almost perfectly follows the Close price trend, confirming its efficiency in forecasting.
- 3D-vizualization of validation loss demonstrated that Dropout = 0.3 is less effective, as it reduces prediction accuracy.

Practical implications of the study:

- The proposed model can be applied to mine clearance operations, including temporal analysis of georadar and magnetometer data, as well as energy forecasting in the Industry 4.0 era, stock prices, currency exchange rates, sales volume predictions, product demand, and other time series tasks.
- The study established that using 325 LSTM-blocks is optimal, achieving a minimum forecast error (MAPE = 1.64%), surpassing previous studies where the error exceeded 1.8%.
- This model can also be adapted for forecasting drone trajectories, meteorological changes, and object recognition based on electromagnetic signals.
- It was confirmed that applying the Nadam optimizer and low Dropout values (0.05–0.1) ensures training stability and high-speed model learning.

Thus, this study confirms that a properly optimized LSTM architecture, incorporating the optimal number of blocks, Dropout levels, and technical indicators, can significantly enhance the accuracy of time series forecasting. The results obtained open new opportunities for applying LSTM networks in financial analysis, market trend forecasting, and artificial intelligence tasks. Additionally, the proposed architectural and morphological solutions can be directly applied to predicting the trajectories of drones and forecasting the potential movement of mines due to weather and geological factors, as well as optimizing resource planning in demining missions.

Acknowledgements

The research was carried out with the grant support of the National Research Fund of Ukraine "Methods and means of active and passive recognition of mines based on deep neural networks", project registration number 273/0024 from 1/08/2024 (2023.04/0024). Also, we would like to thank the reviewers for their precise and concise recommendations that improved the presentation of the results obtained.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] Abbasimehr, H., Paki, R. Improving time series forecasting using LSTM and attention models. J Ambient Intell Human Comput 2022; 13, 673–691. https://doi.org/10.1007/s12652-020-02761-x

[2] Fischer, T., & Krauss, C. Deep learning with long short-term memory networks for financial market predictions. European Journal of Operational Research, 2018; 270(2), 654-669. https://doi.org/10.1016/j.ejor.2017.11.054.

[3] Bao W., Yue J., Rao Y. A Deep Learning Framework for Financial Time Series Using Stacked Autoencoders and Long-Short Term Memory. PLoS ONE 12(7): e0180944. 2017; https://doi.org/10.1371/journal.pone.0180944.

[4] Abbasimehr, H., Paki, R. Improving Time Series Forecasting Using LSTM and Attention Models. Proceedings of the International Conference on Machine Learning. Journal of Ambient Intelligence and Humanized Computing 2022; 13, 673–691. https://doi.org/10.1007/s12652-020-02761-x

[5] Kurniawan A., Indrabayu, Yusuf M. Stock Price Prediction Using Technical Data and Sentiment Score. 2024 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT). 2024; https://doi.org/10.1109/IAICT62357.2024.10617768

[6] Peter T. Yamak, Li Yujian, and Pius K. Gadosey. A Comparison between ARIMA, LSTM, and GRU for Time Series Forecasting. ACAI '19: Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence. 2020; https://doi.org/10.1145/3377713.3377722

[7] Livieris, I. E., Pintelas, E., & Pintelas, P. "A CNN–LSTM model for gold price time-series forecasting." Neural Computing and Applications, 2020; 32, 17351–17360. https://doi.org/10.1007/s00521-020-04867-x

[8] Smyl, S. "A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting." International Journal of Forecasting, 2020; 36(1), 75–85. https://doi.org/10.1016/j.ijforecast.2019.03.017

[9] Makridakis, S. et al. "The M5 competition: Background, methods and results." International Journal of Forecasting, 2022; 38(3), 897–906. https://doi.org/10.1016/j.ijforecast.2021.07.007

[10] Zhang, G. P. "Forecasting inflation using univariate continuous-time stochastic models." Journal of Forecasting, 2019; 38(8), 655–676. https://doi.org/10.1002/for.2603

[11] Zhou, H. et al. "Informer: Beyond efficient transformer for long sequence time-series forecasting." AAAI Conference on Artificial Intelligence, 2021; 35(12), 11106–11115. https://doi.org/10.1609/aaai.v35i12.17325

[12] Kim, K., & Won, C. H. Forecasting the volatility of stock price index: A hybrid model integrating LSTM with multiple GARCH-type models. Expert Systems with Applications, 2018; 103, 25-37. https://doi.org/10.4108/eai.6-1-2023.2330313.

[13] Peleshchak I. Lytvynenko Yu. Vysokotochne prohnozuvannia chasovykh riadiv. Proceedings of the 8th International Scientific and Practical Conference «International Scientific Discussion: Problems, Tasks and Prospects». 2024; https://doi.org/10.51582/interconf.19-20.05.2024.063

[14] Peleshchak, I., Futryk Y. Time series forecasting using a neural network with serially connected LSTM blocks. Herald of Khmelnytskyi National University. Technical Sciences, 2025; 347(1), 432-441. https://doi.org/10.31891/2307-5732-2024-347-1-59

[15] Bishesh, B. Neural Networks for Big Data Analytics. In M. Mangla, S. Satpathy, B. Nayak & S. N. Mohanty (Eds.), Integration of Cloud Computing with Internet of Things: Foundations, Analytics, and Applications 2021; (pp. 277–298). https://doi.org/10.1002/9781119769323.ch16

[16] Farrag, Tamer & Elattar, Ehab. Optimized Deep Stacked Long Short-Term Memory Network for Long-Term Load Forecasting. IEEE Access 2021; 9, 68511–685222021; http://dx.doi.org/10.1109/ACCESS.2021.3077275

[17] Chicco D, Warrens MJ, Jurman G. The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. PeerJ Comput. Sci. 7: e 623. – 2021. https://doi.org/10.7717/peerj-cs.623

[18] A Theoretically Grounded Application of Dropout in Recurrent Neural Networks. Advances in Neural Information Processing Systems (NeurIPS), 2016; 29. ttps://doi.org/10.48550/arXiv.1512.05287

[19] Nelson, D. M., Pereira, A. C. M., & de Oliveira, R. A. Stock market's price movement prediction with LSTM neural networks. International Joint Conference on Neural Networks (IJCNN), 2017; 1419-1426. https://doi.org/10.1109/IJCNN.2017.7966019