https://doi.org/10.53360/2788-7995-2025-3(19)-8

IRSTI: 28.23.37

D. Amrin[1], S. Mukhanov[2*], S. Amanzholova[2], B. Amirgaliyev[2]
[1]International Information Technology University,
050000, Republic of Kazakhstan, Almaty, 34/1 Manas Street
[2]Astana IT University,
010000, Republic of Kazakhstan, Astana, 55/11 Mangilik El Avenue
*e-mail: s.mukhanov@astanait.edu.kz

A COMPARATIVE ANALYSIS OF NEURAL NETWORK MODELS ON PREDICTING STOCK PRICES

Abstract: *Due to their complex and unpredictable nature, stock market movements were always challenging to predict. Factors like economic indicators, market sentiment, and political and global events significantly contribute to stock price unpredictability. There are different methods to analyze risks, returns, and average price movements, based on which investors make assumptions. Identifying patterns and making the right decision on large amounts of data is very difficult, but nowadays, with the advancement of neural networks, we can solve prediction problems by identifying patterns of high-dimensional sequential data. We will analyze and compare five neural network architectures, including Recurrent Neural Networks (RNNs), Long Short-Term Memory networks (LSTMs), Gated Recurrent Units (GRUs), Convolutional Neural Networks (CNNs), and Artificial Neural Networks (ANNs), to try to predict stock prices using historical data taken from Yahoo Finance API, which is widely used and reliable for financial data analysis. We will separate historical data into two parts, 80% of which will be trained and 20% will be tested. For each model, we will use different hyperparameters we selected as the most effective training. Popular Python libraries such as TensorFlow, Keras, and NumPy are used for efficient implementation. Additionally, we used preprocessing for data, such as data cleaning and normalization, to avoid errors and enhance model performance. The models are evaluated based on prediction accuracy using metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared (R^2). Additionally, we use classification metrics such as the confusion matrix and Receiver Operating Characteristic - Area Under the Curve (ROC-AUC) to analyze each model's performance in predicting price movement directions. We concluded that the GRU model achieves the highest accuracy and reliability in our analysis, with notable performance in classification metrics. Conversely, the simple ANN model shows the worst results, highlighting the variability in predictive capabilities across different neural network architectures.*

Key words: *neural networks, deep learning, time-series forecasting, stock market prediction, financial data analysis.*

Introduction

With the advancement of machine learning and deep learning, stock market prediction and pattern recognition have shifted to complex methods [1, 2]. The stock market is known for its complexity and unpredictability, making accurate price prediction a challenging task that researchers, investors, financial institutions, and governments are always interested in [3, 4]. The volatility and randomness of stock prices are driven by dynamic factors that make price movements more unpredictable [5]. Traditional statistical and econometric models often lack in accurately capturing the complex patterns in financial data [6].

Neural networks show good results in processing large amounts of sequential data, recognizing complex patterns, and making predictions with improved accuracy [7]. Among these architectures, Recurrent Neural Networks (RNNs) and their advanced alternatives have also been shown to effectively process time series data, making them suitable for financial forecasting.

For example, Gao et al. [8] compared LSTM to GRU for stock price prediction, both optimized by least absolute shrinkage and selection operator and principal component analysis.

Additionally, convolutional neural networks (CNNs), typically used for image processing, are gaining attention due to their ability to identify local patterns in sequential data. [9].

The work of Mukherjee et al. [10], showed that CNN have better prediction compared to ANN. Durairaj et al. [11] implemented CNN in combination with Chaos Theory and Polynomial Regression (PR) for financial time series prediction in stocks indices, commodity prices and foreign exchange rates, and concluded that this hybrid approach is better than other models.

In other work, comparisons between hybrid CNN-RNN, CNN-LSTM, and CNN-GRU were made but lacked a standalone model for comprehensive analysis [12].

Artificial Neural Networks (ANNs), on the other hand, serve as simpler benchmarks for evaluating the performance of more complex models [13].

By finding the most effective neural network model for stock price prediction we will help other researchers and practitioners to choose the right model for implementation of neural networks in predictive tasks.

Materials and Methods

We use different neural network architectures to predict stock market price movements and outline the data sources, preprocessing steps, model architectures. Timeframe of 5 years were used.

Data preprocessing is a necessary step for preparing time-series data because it will help suit our data for neural network training [14]. Steps include data cleaning and normalization, which means scaling the input features for optimization of training for neural networks [15].

In our study we compare five different neural network models that is used to evaluate different aspects and efficiency of stock price behavior:

1. Recurrent Neural Networks (RNN): a model used for sequential data, that is capable of capturing time series dependencies [16].

2. Long Short-Term Memory (LSTM): an advanced variant of RNN that solves vanishing gradient issues which is ideal for long-term dependency learning in stock prices [17]. Gate units are stored in layers, and processed data is stored in hidden state or cell's state [18].

3. Gated Recurrent Units (GRU): another RNN variant like LSTM mode that is optimized for reducing computational complexity, while retaining memory of past prices, which is very useful in our analysis [19].

4. Convolutional Neural Networks (CNN): Applied to identify local patterns in the data using convolutional calculations [20].

5. Artificial Neural Networks (ANN): Used to establish a benchmark, this model captures simple relationships between inputs and outputs, providing a baseline comparison [21].

To assess model performance, we use Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), R-Squared (R^2).

Mean Absolute Error (MAE) measures the average absolute differences between predicted ($\hat{y}$) and actual (y_i) stock prices [22].

Mean Squared Error (MSE) emphasizes larger errors, and is useful for penalizing significant deviations:

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

where n is the number of predictions.

Root Mean Squared Error (RMSE): Provides an overall measure of model accuracy in original units and is the square root of MSE.

R-Squared (R^2): Indicates the proportion of variance explained by the model, offering insights into model fit quality:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2}{\sum_{i=1}^{n}\left(y_i - \bar{y}\right)^2} \quad (2)$$

These metrics are important, because they provide quantitative measurement for the model's prediction accuracy, enabling objective evaluation and comparison of model performance across different neural network architectures [23].

Other crucial metrics are the confusion matrix and ROC-AUC (Receiver Operating Characteristic - Area Under the Curve).

By analyzing these metrics across multiple neural network models, we will identify an optimal model for stock market prediction, and based on the results, we will choose the better model and will identify what we need to include for better performance.

Results and Discussion

Figure 1 shows Microsoft's (MSFT) stock price movements and its moving averages. The 50-day moving average is more responsive to short-term price fluctuations. The 200-day moving average is less sensitive to short-term noise and provides a smoother representation of the long-term price trend. It is often used as a key indicator of market sentiment and potential trend reversals. However, the moving average is less useful for forecasting long-term time series [24].

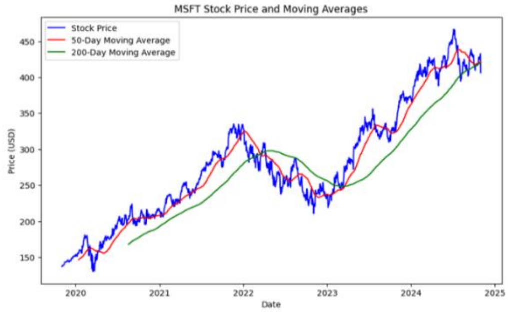

Figure 1 – Microsoft stock prices and moving averages

While moving averages offer valuable insights into market trends, their effectiveness can be limited by their dependence on historical data and their inability to adapt to rapidly changing market dynamics. To solve this problem, we propose integrating neural networks into technical analysis, which is a promising approach, as it will provide better results.

Firstly, we split our dataset into training set (80%) and a test set (20%), and built each model with Keras sequential model, adding different types of layers individually.

Figure 2 illustrates model layers. For the RNN model we use an RNN layer with 60 units for sequential dependencies in the input data, dropout to reduce overfitting, a second RNN with 120 units followed by dropout, a dense layer with 20 units and ReLU activation, and 1 unit output layer. LSTM model uses 2 LSTM layers with 60 units to capture long-term dependencies in the input data followed by a dropout layer to prevent overfitting. The next layer is LSTM with 60 units followed by a dropout layer, a dense layer with 20 units with ReLU activation, and 1 unit output layer.

RNN model:

Model: "sequential"

Layer (type)	Output Shape	Param #
simple_rnn (SimpleRNN)	(None, 60, 60)	3,720
dropout (Dropout)	(None, 60, 60)	0
simple_rnn_1 (SimpleRNN)	(None, 120)	21,720
dropout_1 (Dropout)	(None, 120)	0
dense (Dense)	(None, 20)	2,420
dense_1 (Dense)	(None, 1)	21

Total params: 27,881 (108.91 KB)
Trainable params: 27,881 (108.91 KB)
Non-trainable params: 0 (0.00 B)

LSTM model:

Model: "sequential"

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 60, 60)	14,880
dropout (Dropout)	(None, 60, 60)	0
lstm_1 (LSTM)	(None, 120)	86,880
dropout_1 (Dropout)	(None, 120)	0
dense (Dense)	(None, 20)	2,420
dense_1 (Dense)	(None, 1)	21

Total params: 104,201 (407.04 KB)
Trainable params: 104,201 (407.04 KB)
Non-trainable params: 0 (0.00 B)

GRU model:

Model: "sequential"

Layer (type)	Output Shape	Param #
gru (GRU)	(None, 60, 60)	11,340
dropout (Dropout)	(None, 60, 60)	0
gru_1 (GRU)	(None, 120)	65,520
dropout_1 (Dropout)	(None, 120)	0
dense (Dense)	(None, 20)	2,420
dense_1 (Dense)	(None, 1)	21

Total params: 79,301 (309.77 KB)
Trainable params: 79,301 (309.77 KB)
Non-trainable params: 0 (0.00 B)

CNN model:

Model: "sequential"

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 58, 64)	256
max_pooling1d (MaxPooling1D)	(None, 29, 64)	0
dropout (Dropout)	(None, 29, 64)	0
flatten (Flatten)	(None, 1856)	0
dense (Dense)	(None, 50)	92,850
dense_1 (Dense)	(None, 1)	51

Total params: 93,157 (363.89 KB)
Trainable params: 93,157 (363.89 KB)
Non-trainable params: 0 (0.00 B)

ANN model:

Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 64)	3,904
dense_1 (Dense)	(None, 64)	4,160
dropout (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 1)	65

Total params: 8,129 (31.75 KB)
Trainable params: 8,129 (31.75 KB)
Non-trainable params: 0 (0.00 B)

Figure 2 – Model layers

In our GRU model, we have a GRU layer with 60 units for long-term dependencies, a dropout layer to prevent overfitting, another GRU layer with 120 units, a dropout layer, a dense layer with 20 units followed by ReLU activation, and 1 one-unit output layer.

For the CNN model, we use a 1D convolutional layer, a 1D max pooling layer that helps reduce overfitting and computational cost, a dropout layer to reduce overfitting, a flatten layer that flattens the 2D feature maps into a 1D array, a layer with 50 units followed by a ReLU activation function, and an output layer with 1 unit.

For a simple ANN model, we use 2 dense layers with 64 units and ReLU activation, a dropout layer to reduce overfitting, and a dense output layer with 1 unit.

Figure 3 illustrates stock price prediction results against actual prices. While the RNN model is accurate in trend prediction overall, it may be bad at predicting sharp market movements. The LSTM model's strength in maintaining trend alignment with actual prices, but it may not fully capture the rapid price fluctuations as in the previous model. The GRU model's performance shows its strong potential for accurate stock price prediction, making it a valuable neural network model for financial predictions. CNN model shows good results in trend prediction but like other models, it performs poorly in case of abrupt price changes. It is still a useful model for stock price prediction. ANN model's performance is less efficient for predicting MSFT stock prices, because of its inability to closely follow the actual price trends.

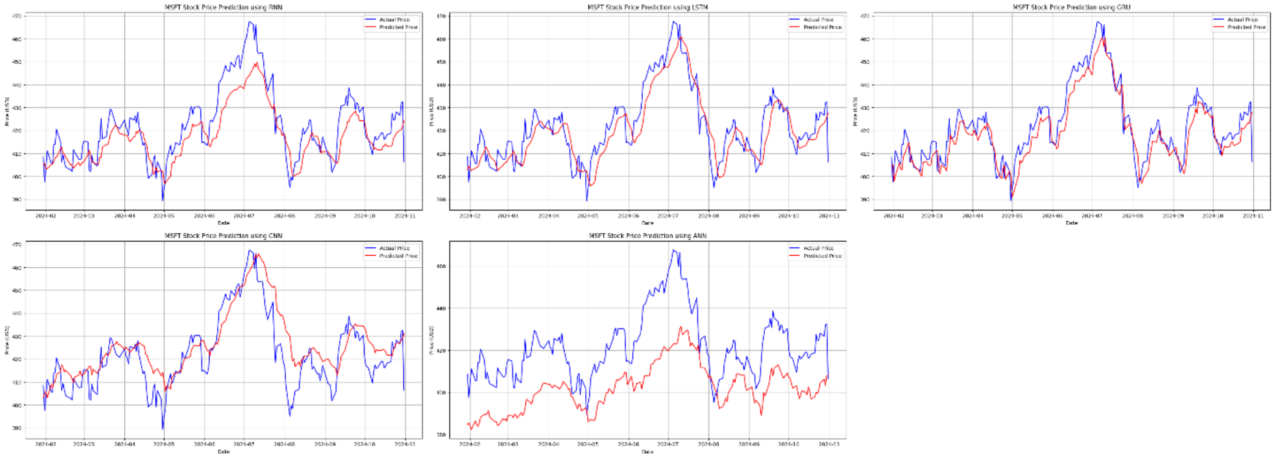

Figure 3 – Microsoft stock price prediction using neural network models

Figure 4 illustrates confusion matrices for models. The RNN model correctly predicted stock price increases (TP = 57) more often than it correctly predicted price drops (TN = 41), but it has a significant number of incorrect positive predictions (FP = 51) and false negatives (FN = 42). This shows us that the RNN model overpredicts increases in stock prices very often, which can lead to less reliable information for traders to make a decision. The LSTM model shows improved performance with a higher number of true positives (TP = 60) and slightly fewer false negatives (FN = 39) compared to the RNN model. It also maintains a balance between true negatives (TN = 42) and false positives (FP = 50). This means that the LSTM model is better at predicting stock price increases accurately and also maintains a reasonable false positive value. The GRU model demonstrates a balanced performance with true positives (TP = 55) and false negatives (FN = 44), and is comparable in number, as well as true negatives (TN = 44) and false positives (FP = 48). This balance indicates that the GRU model can reliably predict both increases and decreases in stock prices, making it a moderately effective model for stock price prediction. The CNN model has a higher number of false positives (FP = 55) than true negatives (TN = 37), which means that it often incorrectly predicts stock price increases. However, it maintains a higher number of true positives (TP = 57) than false negatives (FN = 42), showing that while it can accurately predict some increases, it is less reliable overall because of the high false positive rate. The ANN model shows an equal number of true negatives (TN = 42) and false positives (FP = 50), indicating a balanced, but it still a poor performance. The true positives (TP = 51) are only slightly higher than the false negatives (FN = 48), which means that the ANN model struggles to provide accurate predictions and is the least reliable among the models tested.

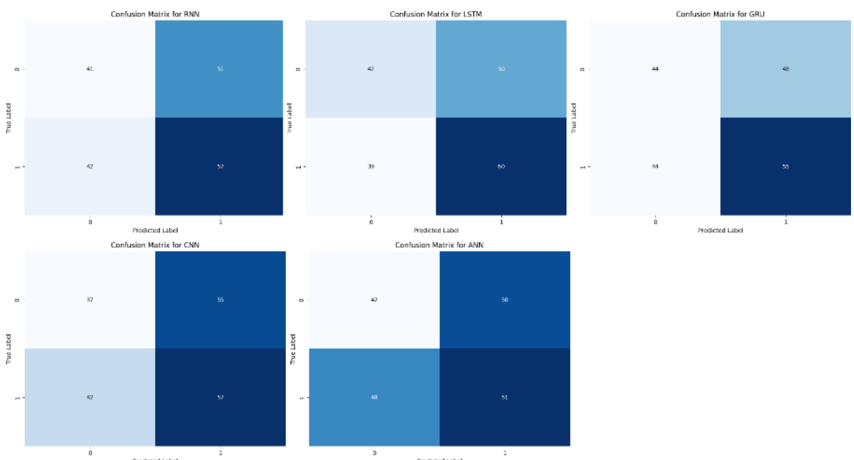

Figure 4 – Confusion matrix for neural network models

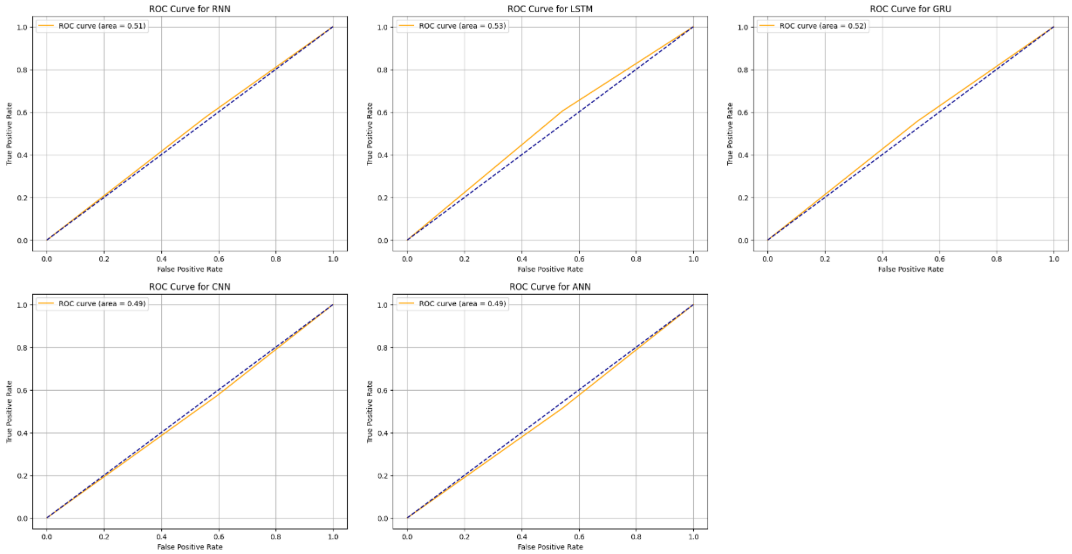

Figure 5 – ROC-AUC for neural network models

The ROC curve of RNN is very close to the diagonal reference line, suggesting that the model struggles to distinguish positive from negative classes effectively (Figure 5). This significantly above-average AUC value implies limited predictive capability for stock price movements. With an AUC value of 0.53, the LSTM model performs better than the RNN model. The LSTM model has a slightly better prediction, but it remains relatively weak in accurately predicting stock price movements. The ROC curve's proximity of the GRU model to the diagonal line reflects its limited effectiveness in distinguishing between classes. This AUC value highlights the need for further model improvement to enhance its predictive accuracy. The ROC curve of CNN model lying below the diagonal reference line means that it is bad with classification task. This AUC value highlights the CNN model's inadequacy in predicting stock price movements accurately, compared to previous models. The ROC curve of ANN is below the diagonal line means that the ANN model is ineffective in distinguishing between positive and negative classes. This AUC value underscores the ANN model's poor suitability for stock price prediction tasks.

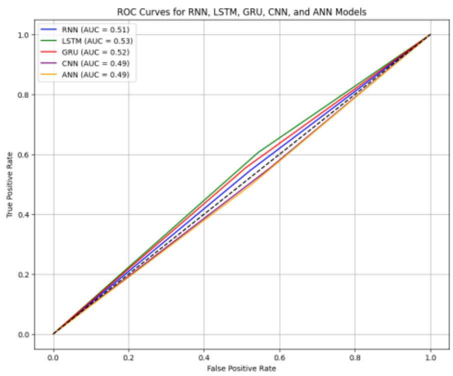

Figure 6 – ROC-AUC for RNN, LSTM, GRU, CNN, ANN

Figure 6 shows the ROC curves together. The ROC analysis shows us that none of the models demonstrate strong predictive power, with AUC values being around 0.5. The LSTM model shows the highest AUC at 0.53, indicating slight but insufficient improvement over random classification. The GRU and RNN models follow closely, while the CNN and ANN models underperform with AUC values of 0.49. Our analysis shows that none of these models can adequately predict if stock will go up or down, which means that we have the potential to improve our model's performance and that we should focus on increasing the AUC value for all of them.

We analyzed metrics, like MAE, MSE, RMSE, and R squared, as shown in Table 1. Based on the analysis of the performance metrics across various models, we observe distinct differences in accuracy and prediction quality. The GRU model outperforms the other models, showing the lowest error rates and highest R squared value. This demonstrates that in our analysis of metrics, GRU is the most precise and reliable model among those tested, because it has the lowest MAE of

5.36, MSE of 43.53, and RMSE of 5.81, which means that there are small differences between predicted and actual values. The R squared value of 0.82 also means that the GRU model's prediction fits the data most. On the other hand, the poorest performance is demonstrated by the ANN, and it shows the highest error metrics overall: an MAE of 19.84, MSE of 472.89, and RMSE of 21.74, which reflect large prediction errors. The negative R squared value of -0.98 further demonstrates its inadequacy in prediction in our analysis. Other models, such as RNN, LSTM, and CNN, also show different degrees of prediction accuracy. The LSTM and CNN models perform well as well, in which LSTM achieves MAE of 6.18 and an R squared value achieves 0.76, the CNN showing MAE of 7.43 and an R squared of 0.61. These metrics show that while they are more accurate than the RNN, their performances are not better than the GRU's one. As shown in confusion matrices, within all models a common difficult task includes balancing false positives and false negatives, which highlights the difficulty in predicting stock price movements. While LSTM and GRU generally perform better than CNN and ANN because of their ability to keep information about previous time data, they still produce a significant number of incorrect classifications, especially in volatile market conditions.

Table 1 – sequential neural network models performance comparison

Models	MAE	MSE	RMSE	R^2
RNN	7.02	73.42	8.57	0.69
LSTM	6.18	56.26	7.5	0.76
GRU	5.36	43.53	6.59	0.82
CNN	7.43	93.52	9.67	0.61
ANN	19.84	472.89	21.74	-0.98

Conclusions

In our paper we provided a comprehensive analysis of several neural network architectures for stock price prediction, highlighting the advantages and disadvantages of each model in their predictive accuracy and efficiency. With the help of Python libraries, we could initialize each model and test them. Our findings demonstrate that the Gated Recurrent Unit (GRU) model slightly outperforms other architectures, demonstrating superior accuracy and reliability compared to them, as shown by its low error metrics (MAE, MSE, RMSE), high R-squared (R^2) value, and good performance in classification metrics, including a confusion matrix and Area Under the Curve - Receiver Operating Characteristic (AUC-ROC) score.

The GRU model's relatively low false positive rate means that it better fit for real-world trading applications where minimizing risk is as essential as maximizing profit.

In contrast, the Artificial Neural Network (ANN) model showed less effectiveness, with its high error metrics, a negative R^2 value, and lower AUC-ROC performance. Overall, ANN demonstrated worse results for our study.

Moderate predictive capabilities are demonstrated by other models, like the Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), and Convolutional Neural Network (CNN). While the LSTM and CNN models showed better performance than the RNN, they are still behind GRU.
Our results show that we need to focus on improving the performance of ROC-AUC scores for all models because, due to the stock market's dynamic nature, it is still very difficult for neural network models to predict movements, as trying to buy or sell a share is very similar to flipping the coin. In our future studies, we will try to implement a newer neural network model or add additional input data and change hyperparameters.

References

1. Comparative analysis of neural network models for gesture recognition methods hands / Bulletin of NIA RK / S.B. Mukhanov et al // Information and communication technologies. – 2023 – № 2. – P. 88.
2. Olorunnimbe K. Deep learning in the stock market – a systematic survey of practice, backtesting, and applications / K. Olorunnimbe, H. Viktor // Artif. Intell. Rev. – 2023. – Vol. 56, № 3. – P. 2057-2109.
3. Cohen G. Algorithmic Trading and Financial Forecasting Using Advanced Artificial Intelligence Methodologies: 18 / G. Cohen // Mathematics. Multidisciplinary Digital Publishing Institute. – 2022. – Vol. 10, № 18. – P. 3302.

4. Predicting stock price movement using a DBN-RNN / X. Zhang et al // Appl. Artif. Intell. – 2021. Vol. 35, № 12. – P. 876-892.
5. An innovative neural network approach for stock market prediction / X. Pang et al // J. Supercomput. – 2020. – Vol. 76, № 3. – P. 2098-2118.
6. Chong E. Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies / E. Chong, C. Han, F.C. Park // Expert Syst. Appl. – 2017. – Vol. 83. – P. 187-205.
7. Sarker I.H. Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions / I.H. Sarker // SN Comput. Sci. – 2021. – Vol. 2, № 6. – P. 420.
8. Gao Y. Stock Prediction Based on Optimized LSTM and GRU Models / Y. Gao, R. Wang, E. Zhou // Sci. Program. ed. Gupta P. – 2021. – Vol. 2021. – P. 1-8.
9. A comparison of convolutional neural networks for Kazakh sign language recognition / C. Kenshimov et al // East.-Eur. J. Enterp. Technol. – 2021. – Vol. 5, № 2(113). – P. 44-54.
10. Stock market prediction using deep learning algorithms / S. Mukherjee et al // CAAI Trans. Intell. Technol. – 2023. – Vol. 8, № 1. – P. 82-94.
11. Durairaj Dr.M. A convolutional neural network based approach to financial time series prediction / Dr.M. Durairaj, B.H.K. Mohan // Neural Comput. Appl. – 2022. – Vol. 34, № 16. – P. 13319-13337.
12. Jaiswal R. A Hybrid Convolutional Recurrent (CNN-GRU) Model for Stock Price Prediction / R. Jaiswal, B. Singh // 2022 IEEE 11th International Conference on Communication Systems and Network Technologies (CSNT). Indore, India: IEEE. – 2022. – P. 299-304.
13. Chahuán-Jiménez K. Neural Network-Based Predictive Models for Stock Market Index Forecasting / K. Chahuán-Jiménez // J. Risk Financ. Manag. – 2024. – Vol. 17, № 6. – P. 242.
14. Deep Learning for Stock Market Prediction / Nabipour M. et al. // Entropy. – 2020. – Vol. 22, № 8. – P. 840.
15. Normalization Techniques in Training DNNs: Methodology, Analysis and Application / L. Huang et al // IEEE Trans. Pattern Anal. Mach. Intell. – 2023. – Vol. 45, № 8. – P. 10173-10196.
16. Deep causal speech enhancement and recognition using efficient long-short term memory Recurrent Neural Network / Z. Li et al // PLOS ONE. Public Library of Science. – 2024. – Vol. 19, № 1. – P. e0291240.
17. Mienye I.D. Recurrent Neural Networks: A Comprehensive Review of Architectures, Variants, and Applications: 9 / I.D. Mienye, T.G. Swart, G. Obaido // Information. Multidisciplinary Digital Publishing Institute. – 2024. – Vol. 15, № 9. – P. 517.
18. GESTURE RECOGNITION OF MACHINE LEARNING AND CONVOLUTIONAL NEURAL NETWORK METHODS FOR KAZAKH SIGN LANGUAGE / S. Mukhanov et al // Sci. J. Astana IT Univ. – 2023. – P. 85-100.
19. Patil P. Long Short-Term Memory and Gated Recurrent Unit Networks for Accurate Stock Price Prediction / P. Patil, S.J. Patel // International Journal of Computer Science Trends and Technology (IJCST). – 2023. – Vol. 11, № 4. – P. 9-12.
20. A comparative analysis of neural network models for hand gesture recognition methods / Y.N. Amirgaliyev et al // Bull. Natl. Eng. Acad. Repub. Kazakhstan. – 2023. – Vol. 88, № 2. – P. 16.
21. CNN features with bi-directional LSTM for real-time anomaly detection in surveillance networks / W. Ullah et al // Multimed. Tools Appl. – 2021. – Vol. 80, № 11. – P. 16979-16995.
22. Vallarino D. A Dynamic Approach to Stock Price Prediction: Comparing RNN and Mixture of Experts Models Across Different Volatility Profiles. arXiv, 2024.
23. Chicco D. The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation / D. Chicco, M.J. Warrens, G. Jurman // PeerJ Comput. Sci. – 2021. – Vol. 7. – P. e623.
24. Kumar R. Analysis of Financial Time Series Forecasting using Deep Learning Model / R. Kumar, P. Kumar, Y. Kumar // 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence). Noida, India: IEEE. – 2021. – P. 877-881.

Acknowledgement: *This research has been funded by the Committee of Science of the Ministry of Science and Higher Education of the Republic of Kazakhstan (Grant No. BR28713411 «Development of Methods and Models in Computational Social Sciences for the Analysis and Forecasting of Social Processes)».*

Д. Амрин[1], С. Муханов[2*], С. Аманжолова[2], Б. Амиргалиев[2]
[1]Международный университет информационных технологий,
050000, Республика Казахстан, г. Алматы, ул. Манаса, 34/1
[2] Astana IT University,
010000, Республика Казахстан, г. Астана, проспект Мангилик Ел, 55/11
*e-mail: s.mukhanov@astanait.edu.kz

СРАВНИТЕЛЬНЫЙ АНАЛИЗ МОДЕЛЕЙ НЕЙРОННЫХ СЕТЕЙ ДЛЯ ПРОГНОЗИРОВАНИЯ ЦЕН АКЦИЙ

Прогнозирование динамики фондового рынка остается сложной задачей из-за его волатильности и непредсказуемости. Экономические показатели, рыночные настроения и глобальные события оказывают значительное влияние на колебания цен акций, что затрудняет точное прогнозирование трендов инвесторами. Традиционные методы основаны на анализе исторических рисков, доходности и ценовых паттернов, однако эти подходы имеют ограничения при обработке больших объемов данных. С развитием глубинного обучения нейронные сети стали мощным инструментом для прогнозирования цен акций. В этом исследовании мы сравниваем пять архитектур нейронных сетей: рекуррентные нейронные сети (RNN), сети долговременной кратковременной памяти (LSTM), управляемые рекуррентные блоки (GRU), сверточные нейронные сети (CNN) и искусственные нейронные сети (ANN). В качестве источника данных используется Yahoo Finance API, который является надежной и широко используемой платформой для финансового анализа. Исторические данные разделены на обучающую выборку (80%) и тестовую выборку (20%), а гиперпараметры подбираются для достижения оптимальных результатов. Мы проводим предварительную обработку данных, включая очистку и нормализацию, чтобы повысить точность и эффективность моделей. Для оценки работы моделей используются средняя абсолютная ошибка (MAE), среднеквадратичная ошибка (MSE), корень из среднеквадратичной ошибки (RMSE) и коэффициент детерминации (R^2). Кроме того, мы оцениваем точность классификации с использованием матрицы ошибок и площади под кривой ROC (ROC-AUC). Результаты показывают, что модель GRU превосходит другие, обеспечивая наивысшую точность и надежность как в регрессионных, так и в классификационных метриках. В то же время простая модель ANN демонстрирует худшие результаты, что подчеркивает значительные различия в предсказательной способности различных архитектур нейросетей. Эти выводы подтверждают важность выбора правильной модели для финансового прогнозирования, поскольку методы глубинного обучения продолжают развиваться и повышать точность предсказаний фондового рынка.

Ключевые слова: *нейронные сети, глубинное обучение, прогнозирование временных рядов, прогнозирование фондового рынка, анализ финансовых данных.*

Д. Амрин[1], С.Муханов[2*], С. Аманжолова[2], Б. Амиргалиев[2]
[1]Халықаралық ақпараттық технологиялар университеті,
050000, Қазақстан Республикасы, Алматы қ., Манас көшесі, 34/1.
[2]Астана IT University,
010000, Қазақстан Республикасы, Астана қ., Мәңгілік Ел даңғылы, 55/11
*e-mail: s.mukhanov@astanait.edu.kz

ҚОР НАРЫҒЫ БАҒАЛАРЫН БОЛЖАУҒА АРНАЛҒАН НЕЙРОНДЫҚ ЖЕЛІ МОДЕЛЬДЕРІНІҢ САЛЫСТЫРМАЛЫ ТАЛДАУЫ

Қор нарығының қозғалысын болжау оның тұрақсыз және болжанбайтын табиғатына байланысты күрделі міндет болып қала береді. Экономикалық көрсеткіштер, нарықтық көңіл-күй және жаһандық оқиғалар акция бағасының өзгеруіне айтарлықтай әсер етеді, бұл инвесторлардың трендтерді дәл болжауын қиындатады. Дәстүрлі әдістер тарихи тәуекелдерді, кірістерді және баға үлгілерін талдауға негізделген, бірақ олар үлкен көлемдегі деректерді тиімді өңдеуде шектеулерге ие. Терең оқытудың дамуының арқасында нейрондық желілер акция бағасын болжаудың қуатты құралына айналды. Бұл зерттеуде біз бес нейрондық желі архитектурасын салыстырамыз: қайталанатын нейрондық желілер (RNN), ұзақ қысқа мерзімді жады желілері (LSTM), басқарылатын қайталанатын блоктар (GRU), сверткіш нейрондық желілер (CNN) және жасанды нейрондық желілер (ANN). Деректер көзі ретінде Yahoo Finance API пайдаланылады, ол қаржылық талдау үшін кеңінен қолданылатын сенімді платформа болып табылады. Тарихи деректер оқыту жиынтығына (80%) және сынақ жиынтығына (20%) бөлінеді, ал ең жақсы нәтижеге қол жеткізу үшін гиперпараметрлер реттеледі. Деректерді алдын ала өңдеу тазарту және қалыпқа келтіру арқылы жүзеге асырылады, бұл модельдердің дәлдігі мен тиімділігін арттырады. Модельдердің жұмысын *бағалау үшін орташа абсолюттік қате (MAE), орташа квадраттық қате (MSE), орташа квадраттық қатенің түбірі (RMSE) және детерминация коэффициенті (R^2) қолданылады. Сонымен қатар, қателер матрицасы және ROC қисығы астындағы аудан (ROC-AUC) арқылы классификация дәлдігі бағаланады. Зерттеу нәтижелері GRU моделінің ең жоғары дәлдік пен сенімділікке ие екенін көрсетеді, ол регрессиялық және классификациялық метрикалар бойынша үздік нәтиже береді. Керісінше, қарапайым ANN моделі ең әлсіз нәтижелерді көрсетеді, бұл әртүрлі нейрондық желі архитектураларының болжамдық мүмкіндіктеріндегі айырмашылықтарды айқындайды. Бұл қорытындылар қаржылық болжам жасау үшін дұрыс модельді таңдаудың маңыздылығын растайды, өйткені терең оқыту әдістері үнемі дамып, қор нарығының болжамдарын жақсарта береді.*

***Түйін сөздер**: нейрондық желілер, терең оқыту, уақыттық қатарларды болжау, қор нарығын болжау, қаржылық деректерді талдау.*

Information About Authors

Daryn Amrin – Master's student in Software Engineering, Department of Computer Engineering, International Information Technology University, Almaty, Kazakhstan; e-mail: 41376@iitu.edu.kz. ORCID: https://orcid.org/0009-0003-0684-6947.

Samat Mukhanov* – PhD, Assistant-professor, Astana IT University, Astana, Kazakhstan; e-mail: s.mukhanov@astanait.edu.kz. ORCID: https://orcid.org/0000-0001-8761-4272.

Saule Amanzholova – Candidate of technical sciences, Head of Intelligent Systems and Cybersecurity Department, associate professor, Astana IT University, Astana, Kazakhstan; e-mail: beibut.amirgaliyev@astanait.edu.kz. ORCID: https://orcid.org/0000-0002-6779-9393.

Beibut Amirgaliyev – PhD, Professor, Astana IT University, Astana, Kazakhstan; e-mail: beibut.amirgaliyev@astanait.edu.kz. ORCID: https://orcid.org/0000-0003-0355-5856.

Информация об авторах:

Дарын Амрин – магистрант ОП «Программная инженерия», кафедра «Компьютерная инженерия», Международный университет информационных технологий, Алматы, Казахстан; e-mail: 41376@iitu.edu.kz. ORCID: https://orcid.org/0009-0003-0684-6947.

Самат Муханов* – PhD, Astana IT University, Астана, Казахстан; e-mail: s.mukhanov@iitu.edu.kz. ORCID: https://orcid.org/0000-0001-8761-4272.

Сауле Аманжолова – кандидат технических наук, заведующая кафедрой интеллектуальных систем и кибербезопасности, доцент, Astana IT University, Астана, Казахстан; e-mail: beibut.amirgaliyev@astanait.edu.kz. ORCID: https://orcid.org/0000-0002-6779-9393.

Бейбут Амиргалиев – доктор философии, профессор, Astana IT University, Астана, Казахстан; e-mail: beibut.amirgaliyev@astanait.edu.kz. ORCID: https://orcid.org/0000-0003-0355-5856.

Авторлар туралы мәліметтер:

Дарын Амрин – Халықаралық ақпараттық технологиялар университеті, компьютерлік инженерия кафедрасы, Программалық инженерия ББ магистранты, Алматы, Қазақстан; e-mail: 41376@iitu.edu.kz. ORCID: https://orcid.org/0009-0003-0684-6947.

Самат Мұханов* – PhD, Astana IT University, Астана, Қазақстан; e-mail: s.mukhanov@iitu.edu.kz. ORCID: https://orcid.org/0000-0001-8761-4272.

Сәуле Аманжолова – техника ғылымдарының кандидаты, зияткерлік жүйелер және киберқауіпсіздік кафедрасының меңгерушісі, доцент, Astana IT University, Астана, Қазақстан; e-mail: beibut.amirgaliyev@astanait.edu.kz. ORCID: https://orcid.org/0000-0002-6779-9393.

Бейбіт Әмірғалиев – философия докторы, профессор, Astana IT University, Астана, Қазақстан; e-mail: beibut.amirgaliyev@astanait.edu.kz. ORCID: https://orcid.org/0000-0003-0355-5856.

Received 03.04.2025
Revised 02.06.2025
Accepted 04.08.2025