UDC 004.048
DOI https://doi.org/10.32782/2663-5941/2024.2/10

Hladka O.M.
National University of Water and Environmental Engineering

Karpovich I.M.
National University of Water and Environmental Engineering

Sabaniuk A.Yu.
National University of Water and Environmental Engineering

ARTIFICIAL NEURAL NETWORKS FOR ANALYSIS OF STOCK MARKET DYNAMICS

A software application has been developed that implements the methods of artificial intelligence and machine learning, and simulates the operation of different types of artificial neural network models. We used the developed software product to forecast the dynamics of the company's stock indicators. Many different factors can affect stock prices, including company news and results, industry indicators, investor sentiment and others economic factors. Because of this complexity, there is great interest in the application of machine learning methods to evaluate large historical data sets of enterprise stock indicators. Predictive models based on machine learning are widely used in industry and research to facilitate the prediction of time and resource allocation, in particular, in business planning, cryptocurrency markets, stock markets, disease spread modeling, pattern recognition, resource allocation, sports analytics, statistics, weather forecasts, etc. We experimented with the following models: Recurrent Neural Networks, Model of Long Short-Term Memory, Multi-Layer Perceptron. Analysis and research of artificial neural networks with different architectures was performed. The features of various models of artificial neural networks and their areas of application are studied. We used the Python programming language and the Jupyter Notebook development environment for the software implementation of the project. The advantage of the Jupyter Notebook development environment, which became decisive in the choice, is the ability to break the code into parts and execute them separately or change the order of code execution. The Python programming language has many built-in frameworks, libraries and extensions that simplify the implementation of various functions, which allowed us to easily create a software application to realize the goals of machine learning. A number of Python libraries and built-in functions for artificial intelligence and machine learning were used in the implementation of the project.

Key words: *artificial neural network, time series, machine learning, Deep Learning, Recurrent Neural Networks, Long Short-Term Memory, Multi-Layer Perceptron, Python, Jupyter Notebook.*

Formulation of the problem. Time series forecasting is one of the key topics in machine learning. Predictive models based on machine learning are widely used by enterprises to facilitate forecasting of time and resource allocations. After analyzing the data that was saved in the past, it is possible to make informed decisions [1]. Many industries and scientific fields use time series forecasting: business planning; cryptocurrencies; stock markets; modeling of the spread of diseases; pattern recognition; allocation of resources; sports analytics; statistics; weather forecast etc.

The stock market is the basis of any economy. The main goals of any investment in the stock market are to maximize profit and minimize risk. It is very difficult to predict stock prices because of their uncertain behavior. Investing in the stock market can lead to a quick return on investment, so forecasting is one of the strategies for making a profit. Many different factors can affect stock prices, including company news and results, industry indicators, investor sentiment and economic factors. Because of this complexity, there is great interest in the application of machine learning methods to evaluate large historical data sets of enterprise stock indicators.

Analysis of recent research and publications. The stock market attracts a lot of attention from investors. Capturing the regularity of stock market changes has always been a key point for investors and investment companies. Investors are very interested in the research area of stock price forecasting. To make a successful investment, many investors want to know the future of the stock market [2].

Machine learning algorithms can recognize patterns and relationships between the data they are trained on, build mathematical models related to those patterns, and use those models to make predictions or make decisions. In addition, the more information machine learning-based systems can process, the more sophisticated these models will become, allowing algorithms to improve their analytical and predictive performance [3–4].

Such opportunities are important for financial companies. By examining inputs including stock trends, corporate metrics, financial news, investor behavior, and social media information, these models can pinpoint the subtlest, non-linear relationships between all of these variables. Based on such findings, they will create realistic forecasts of the value of securities and provide market players with useful information and recommendations for future investments [5].

Evaluating machine learning algorithms for stock market forecasting is a task that should be approached with due caution for two good reasons. First, research is still ongoing and far from reaching universally accepted results, since the range of algorithms suitable for this purpose is quite wide, and evaluating their accuracy in different scenarios can be quite difficult. Second, corporations and investment firms are generally reluctant to disclose their machine learning algorithms in order to maintain a competitive advantage. Nevertheless, a general idea of the progress in the development and implementation of algorithms can be obtained from academic studies and reports of scientific societies focusing on different methods of predicting the value of securities [6].

Task statement. The purpose of the work is to develop a software implementation and study the results of training of the most common artificial neural networks. We took the data of the Amazon company's stock indicators from the Yahoo! Finance for 5 years as an object of forecasting. The subject of the research is forecasting the dynamics of the stock market using artificial neural networks.

An artificial neural network is an information processing system that arose as a generalization of mathematical models of neural biology. An artificial neural network is characterized by:

- a pattern of connections between neurons (architecture);
- the method of determining the weighting coefficients of connections (learning algorithm);
- the internal state of neurons (activation function).

An artificial neuron or neural node is a mathematical model. In most cases, it calculates a weighted average of the input and then applies a bias to it. It then passes the resulting signal through the activation function. This activation function is a non-linear function, such as a sigmoid function, which takes a linear input and produces a non-linear output [7].

Outline of the main material of the study. In our work, we investigated the following models of artificial neural networks: Recurrent Neural Networks, Model of Long Short-Term Memory, Multi-Layer Perceptron.

Recurrent Neural Network (RNN) is a type of artificial neural network that implements deep learning algorithms, typically used for ordinal or temporal problems such as translation, speech recognition and processing, image captioning, etc. They are distinguished by their memory because they take information from previous inputs to influence current inputs and outputs. In traditional deep neural networks, the inputs and outputs are independent of each other, but the outputs of recurrent neural networks depends on the previous elements in the sequence [8].

Another distinguishing characteristic of recurrent networks is that they share parameters at each layer of the network. While feed-forward networks have different weights for each node, recurrent neural networks have the same weight parameter at each level of the network. However, these weights are adjusted using backpropagation and gradient descent processes to facilitate reinforcement learning.

Recurrent Neural Networks use a time-based backpropagation algorithm to determine gradients, which is slightly different from traditional backpropagation because it is specific to sequence data. The principles of the time backpropagation algorithm are the same as traditional backpropagation, where the model is trained by computing the errors from the output to the input layer. These calculations allow you to correctly adjust and fit the model parameters.

The time backpropagation algorithm differs from the traditional approach in that it accumulates the errors at each time step, whereas other networks do not need to sum the errors because they do not share the parameters at each level. Due to this process in RNN, there are two problems such as exploding gradients and vanishing gradients. These problems are defined by the size of the gradient, which is the slope of the loss function along the error curve. When the gradient is too small, it continues to decrease, updating the weight parameters until they become negligible, i.e. 0. When this happens, the algorithm no longer learns.

Explosive gradients occur when the gradient is too large, creating an unstable model. In this case, the model weights will grow too large and will eventually be represented as NaN. One of the solutions to these problems is to reduce the number of hidden layers in the neural network, eliminating part of the complexity in the RNN model [8].

There are different RNN architectures that are used in machine learning tasks [9]:

- Bidirectional Recurrent Neural Networks (BRNN) – inputs from future time steps are used to improve the accuracy of the network;
- Gated Recurrent Units (GRU) – developed to solve the problem of gradient disappearance; have a reset and refresh gateway and determine what information should be stored for future predictions;
- Long short-term memory (LSTM) – designed to solve the problem of gradient disappearance.

Long Short-Term Memory (LSTM) is a subtype of recurrent neural networks. They are used to recognize patterns in data sequences appearing in sensor data, stock prices or natural language. Recurrent neural networks store previous data in their short-term memory. When memory runs out, they simply delete the longest stored information and replace it with new data. The LSTM model tries to avoid this problem by storing selected information in long-term memory. This long-term memory is stored in the so-called cell state. In addition, there is also a hidden state in which short-term information from previous steps of the calculation is stored. At each step of the calculation, the current input, the previous state of the short-term memory and the previous state of the hidden state are used [10].

Multi-Layer Perceptron (MLP) is the most widely used form of feed-forward neural networks. Input patterns are represented at the input layer, and the resulting block activations are propagated through hidden layers to the output layer (forward network). Blocks in hidden layers usually have non-linear activation functions. Since the behavior of the function is parameterized by the connection weights, the MLP can approximate any continuous function on a compact input domain with arbitrary precision [9].

In the work, the Mean Absolute Error (MAE) and Mean Square Error (MSE) were used as ***metrics to assess the quality of forecasting***.

The average absolute error (MAE) is the average value of errors in the predictions of the model:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|Y_i - \widehat{Y}_i\right|.$$

The mean squared error (MSE) is used in cases where it is necessary to emphasize large errors and choose a model that gives fewer particularly large errors:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \widehat{Y}_i\right)^2.$$

Here, Y_i are the actual values, $\widehat{Y_l}$ are the predicted values, and n is the number of observations.

Optimization algorithms are aimed at increasing the effectiveness of learning algorithms by increasing their accuracy in forecasting tasks. We analyzed the following optimization algorithms:

- *Stochastic Gradient Descent* (SGD) uses only one sample from the entire data set for each iteration, which avoids redundant computations in a large data set to update a parameter;
- *Root Mean Square Propagation* (RMSprop) is an optimization technique based on momentum gradient descent that limits vertical fluctuations and dynamically adjusts the learning rate by choosing different rates for different parameters;
- *Adam* is an optimization algorithm that calculates the rate of adaptive learning of individual parameters using only first-order gradients; it combines the capabilities of optimization algorithms such as RMSprop (works well with mobile and online environments) and AdaGrad (works well with sparse gradients);
- *Adamax* is a variant of the Adam optimization method; it proved to be stable in the form of infinite order, which distinguishes it from other optimizers [11].

In the work, the Adam optimization method was chosen for the prediction of neural networks.

Activation functions in neural networks and deep learning algorithms play an important role in igniting hidden nodes to obtain a more desirable result [12]. To train neural networks, we used ReLU (*Rectified linear unit*), which is an activation function that introduces the property of nonlinearity into the deep learning model and solves the problem of vanishing gradients.

Development tools and environment. The Python programming language and the Jupyter Notebook development environment were chosen for the construction and training of artificial neural networks. The main advantage of the Jupyter Notebook environment over traditional development tools, which was decisive in the choice, is the ability to break the code into parts and execute them separately or change the order of code execution.

Although machine learning and artificial intelligence are complex algorithms and various workflows, the Python programming language allows developers to build reliable systems. Python has many frameworks, libraries, and extensions that make it easy to implement various features and allows you to quickly create prototypes that allow you to test your product for machine learning purposes. The following libraries for artificial intelligence and machine learning were used in the work:

- *TensorFlow* is a machine and deep learning framework used for numerical calculations and determination of data flows; allows training and running deep neural networks for image recognition, handwritten digit classification, recurrent neural network, word embedding, natural language processing, video detection, etc. [13];
- *NumPy*, *SciPy* are basic packages for high-performance scientific computing and data analysis;
- *Pandas* is a software library for general purpose data analysis and processing [14];
- *Keras* is a special library for neural networks that works on top of TensorFlow; supports numerous neural network server calculations [15];
- *Seaborn* for data visualization;
- *Scikit-learn* contains a variety of classification, regression, and clustering algorithms, including support vector machines, random forests, gradient boosting, k-means, etc.

Results of computational experiments. The work used the closing prices of shares of Amazon (AMZN) over the past 5 years (05-15-2018 to 05-12-2023). Analysis data was downloaded from Yahoo!Finance, which offers historical and real-time stock market data. An analysis of non-zero values and data types in the set was carried out, basic statistics of numerical series were obtained and a check was made for the absence of missing values in the data set.

Each constructed neural network was divided into training and test sets, where 80% of the data is used for training and 20% for testing the model. All models were compiled with the *Adam* optimizer and the *Mean_Squared_Error* loss function, and the *ReLU* activation function was used.

A Recurrent Neural Network (RNN) is a deep learning network framework that has the advantage of taking the context of the data into account during the learning process. The indicators of model evaluations for some built architectures (the number of epochs is 100) are listed in the table 1:

1. RNN consisting of one layer with 64 neurons, where a Dense layer with one neuron is added after the RNN layer.
2. RNN with two layers of 64 neurons each.
3. RNN consists of three layers, where the first one with 128 neurons takes serial input and returns serial output, the next layer with 64 neurons, which takes input from the first layer and returns only the last output state, also added a Dense layer with 1 neuron (Fig. 1).

Long Short-Term Memory (LSTM) is a powerful time series algorithm that can capture historical trends and predict future values with high accuracy. The LSTM model evaluation indicators for some built architectures (the number of epochs is 200) are listed in the table. 2.

The best LSTM result with architecture: one layer with 128 neurons this layer returns sequential outputs for further use, a second layer with 64 neurons that receives sequential outputs from the previous one and one Dense layer with one output, to get the final prediction (Fig. 2).

In the table 3 shows the evaluation indicators of the Multi-Layer Perceptron (MLP) model with 6 different architectures with different number of hidden layers.

The best result obtained is the MLP model, which has one hidden layer with 128 neurons and uses the ReLU activation function. The last layer has one neuron without an activation function, which is designed to predict the share price (Fig. 3).

Therefore, with the increase in the number of layers and neurons, the models show more accurate results with the smallest MAE and MSE errors.

Conclusions. Methods of forecasting the dynamics of the stock market using artificial neural networks are investigated in the work. Predicting the value of securities using machine learning helps to know the price of a company's stock and other financial assets.

Stock market forecasting aims to determine the future movement of the value of financial exchange securities. Accurately predicting the price movement will result in higher profits that investors can

Table 1

RNN model evaluation indicators

No	units	Train MAE	Train MSE	Test MAE	Test MSE
1	64	0.2886	0.1255	0.3016	0.1308
2	64, 64	0.1081	0.0185	0.1242	0.0197
3	**128, 64**	**0.0429**	**0.0038**	**0.0718**	**0.0061**

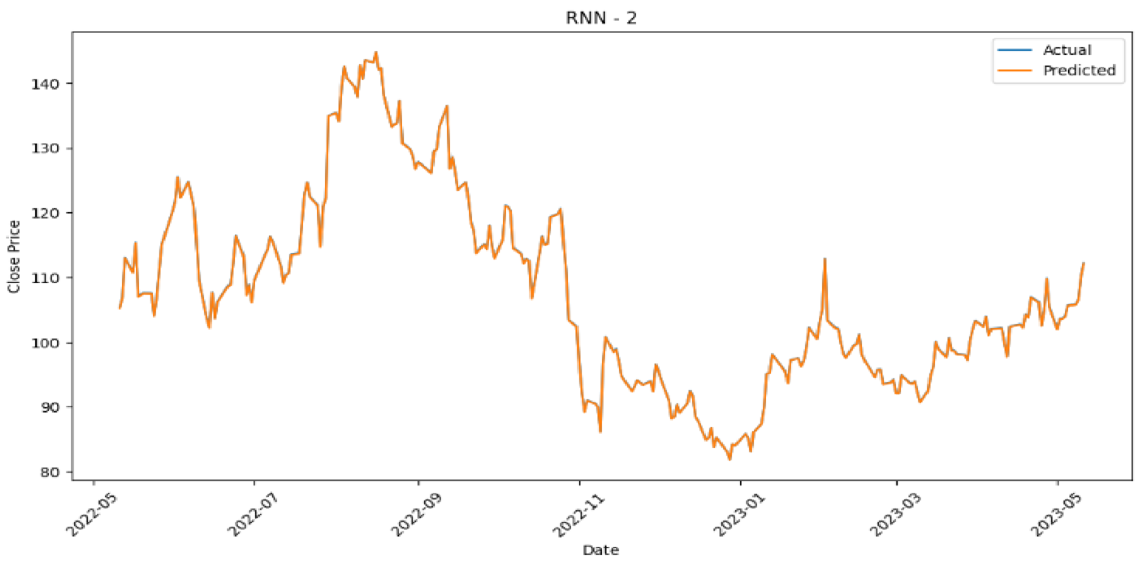

Fig. 1. Forecast and actual values of the RNN model

Table 2

LSTM model evaluation indicators

No	units	Train MAE	Train MSE	Test MAE	Test MSE
1	64	0.7383	0.9647	1.0006	1.4501
2	**128, 64**	**0.0826**	**0.0137**	**0.0117**	**0.0220**
3	64, 64	0.9887	2.3996	0.6743	0.7013
4	128	0.4454	0.3528	0.4491	0.3038

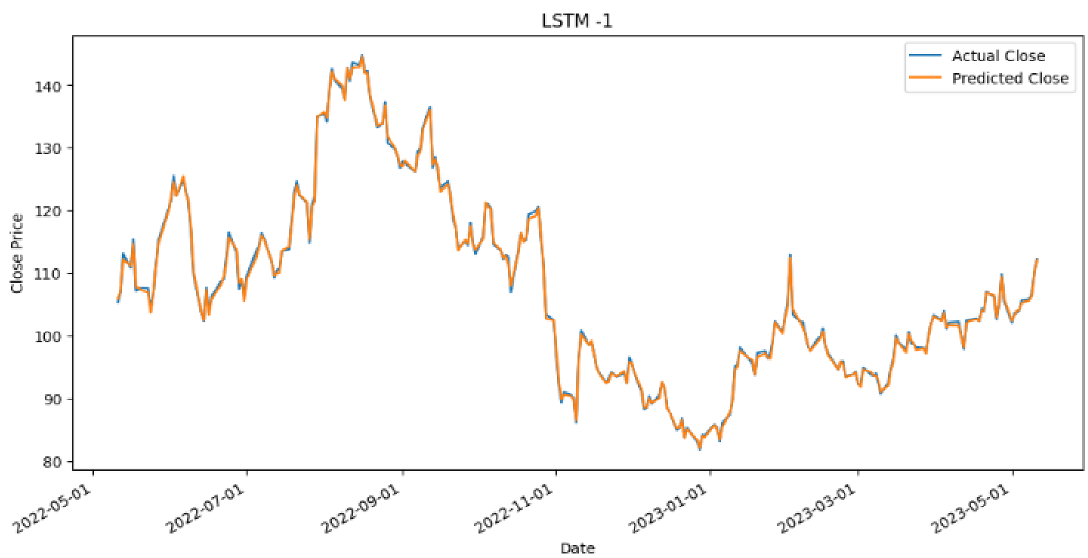

Fig. 2. Forecast and actual values of the LSTM model

make. Predicting how the stock market will move is one of the most difficult matters as it is influenced by many factors such as interest rates, politics, economic growth, which make the stock market volatile and very difficult to predict accurately. Stock forecasting offers huge chances of profit and is the main motivation for research in this field, knowing about the movement of stocks in a fraction of a second can lead to high profits. Since investment in securities is the main activity of the financial market, lack of accu-

Table 3

MLP model evaluation indicators

No	units	epochs	Train MAE	Train MSE	Test MAE	Test MSE
1	32	100	0.2294	0.0976	0.2711	0.1231
2	64	100	0.5496	0.5473	0.6909	0.7395
3	64	200	0.1971	0.0751	0.2331	0.0831
4	64, 64	100	0.0158	0.0004	0.0236	0.0011
5	128	100	0.2890	0.1590	0.3458	0.1827
6	**128**	**200**	**0.0261**	**0.0015**	**0.0253**	**0.0009**

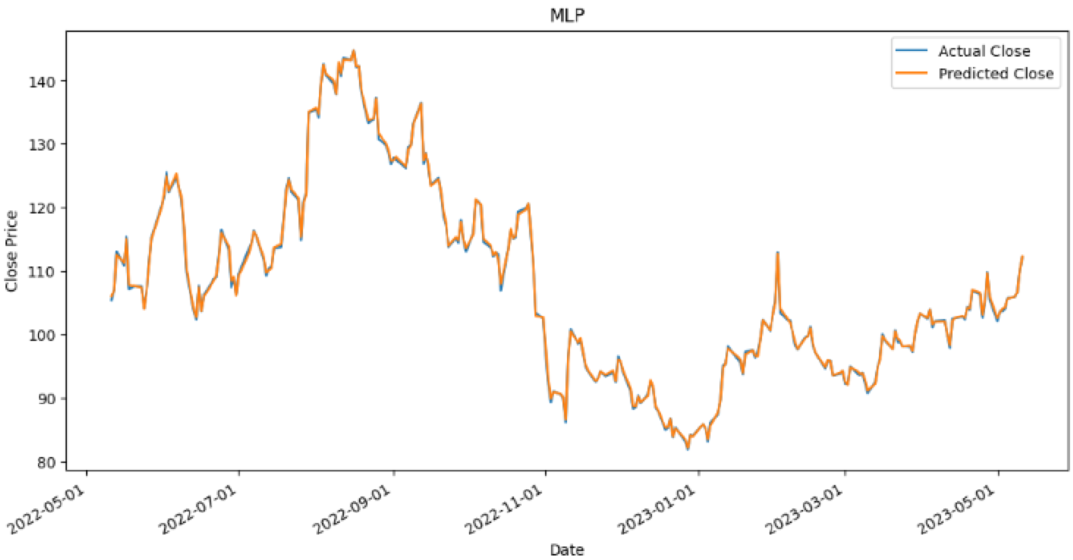

Fig. 3. Forecast and actual values of the MLP model

rate knowledge and detailed information will lead to loss of investment. Forecasting the stock market is a difficult task because market movements are always subject to uncertainty.

The software implementation of the most common neural network models with different architectures was created using the Python programming language and in the Jupyter Notebook development environment. We used to create models: Recurrent Neural Networks, Long Short-Term Memory, Multi-Layer Perceptron.

Developed models can be retrained for new data.

Bibliography:

1. Fundamental Analysis of Stocks: Explained, 2021. URL: https://www.mstock.com/articles/fundamental-analysis (Referred 24.03.2024).

2. K. Gülen, Applying machine learning in financial markets: A review of state-of-the-art methods, 2023. URL: https://dataconomy.com/category/topics/data-science/machine-learning/ (Referred 24.03/2024).

3. S. Asadi, E. Hadavandi, F. Mehmanpazir, M. M. Nakhostin, "Hybridization of evolutionary Levenberg–Marquardt neural networks and data pre-processing for stock market prediction." Knowledge-Based Systems 35 (2012): 245-258. DOI: 10.1016/j.knosys.2012.05.003.

4. P. Hrytsiuk, T. Babych, S. Baranovsky, M. Havryliuk. Assessing of Climate Impact on Wheat Yield using Machine Learning Techniques. CEUR Workshop Proceedings, Vol-3513. Proceedings of the 11-th International Conference "Information Control Systems & Technologies" (ICST 2023), Odesa, Ukraine, September 21–23, 2023. pp. 314-329.

5. F. S. Alzazah and X. Cheng, Recent Advances in Stock Market Prediction Using Text Mining: A Survey, in: E-Business – Higher Education and Intelligence Applications, 2020. DOI: 10.5772/intechopen.92253.

6. K. Li, Predicting Stock Prices Using Machine Learning, 2023. URL: https://neptune.ai/blog/predicting-stock-prices-using-machine-learning (Referred 24.03/2024).

7. N. Bhargav, Neurons in Neural Networks, 2022. URL: https://www.baeldung.com/cs/neural-networks-neurons (Referred 24.03/2024).

8. What are recurrent neural networks? URL: https://www.ibm.com/topics/recurrent-neural-networks (Referred 24.03/2024).

9. A. Graves, Supervised Sequence Labelling with Recurrent Neural Networks, Springer, Berlin, Heidelberg, 2012. DOI: 10.1007/978-3-642-24797-2

10. Long Short-Term Memory Networks (LSTM) – simply explained!, 2022. URL: https://databasecamp.de/en/ml/lstms (Referred 24.03/2024).

11. N. Saurabh, "LSTM-RNN model to predict future stock prices using an efficient optimizer." International Research Journal of Engineering and Technology (IRJET), 7.11, 2020, 672-677.

12. J. Brownlee, A Gentle Introduction to the Rectified Linear Unit (ReLU), 2020. URL: https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks (Referred 24.03/2024).

13. TensorFlow. URL: https://www.javatpoint.com/tensorflow (Referred 24.03/2024).

14. Pandas, 2021. URL: https://www.nvidia.com/en-us/glossary/data-science/pandas-python (Referred 24.03/2024).

15. R. Kumar, What is Keras and How it works? An Overview and Its Use Cases, 2022. URL: https://www.devopsschool.com/blog/what-is-keras-and-how-it-works-an-overview-and-its-use-cases (Referred 24.03/2024).

Гладка О.М., Карпович І.М., Сабанюк А.Ю. ШТУЧНІ НЕЙРОННІ МЕРЕЖІ ДЛЯ АНАЛІЗУ ДИНАМІКИ ФОНДОВОГО РИНКУ

Розроблено програмний застосунок, який реалізує методи штучного інтелекту та машинного навчання, симулює роботу різних типів моделей штучних нейронних мереж. Розроблений програмний продукт використано для прогнозування динаміки біржових показників підприємства. На ціни акцій можуть впливати багато різних чинників, зокрема новини та результати діяльності компаній, показники галузі, настрої інвесторів та інші економічні чинники. Через таку складність існує великий інтерес до застосування методів машинного навчання для оцінки великих історичних наборів даних фондових показників підприємств. Прогнозні моделі на основі машинного навчання широко використовуються у промисловості та наукових дослідженнях для полегшення прогнозування розподілу часу та ресурсів, зокрема, у бізнес плануванні, на ринку криптовалют, фондових ринках, при моделюванні поширення хвороб, розпізнаванні образів, розподілу ресурсів, у спортивній аналітиці, статистиці, прогнозі погоди тощо. Ми експериментували з наступними моделями: Рекурентні нейронні мережі, Модель довготривалої короткочасної пам'яті, Багатошаровий персептрон. Досліджено особливості різних моделей штучних нейронних мереж та сфер їхнього застосування. Проведено аналіз та дослідження штучних нейронних мереж з різною архітектурою. Для програмної реалізації проекту використано мову програмування Python та середовище розробки Jupyter Notebook. Перевагою середовища Jupyter Notebook, що стала вирішальною при виборі, є можливість розбити код на частини та виконувати їх окремо чи змінювати порядок виконання коду. Мова програмування Python має багато вбудованих фреймворків, бібліотек і розширень, які спрощують реалізацію різних функцій, що дозволило без зайвих зусиль створити програмний застосунок для реалізації цілей машинного навчання. В реалізації проекту використано низку бібліотек і вбудованих функцій Python для штучного інтелекту та машинного навчання.

Ключові слова: *штучна нейронна мережа, часові ряди, машинне навчання, глибоке навчання, Рекурентні нейронні мережі, Модель довготривалої короткочасної пам'яті, Багатошаровий персептрон, Python, Jupyter Notebook.*