DOI: 10.31319/2519-8106.1(50)2024.304779
UDC 004.8

Vokhmianin Hlib[1], Student of Systems Software Department
Вохмянін Г.Я., здобувач вищої освіти першого (бакалаврського) рівня кафедри програмного забезпечення систем
ORCID: 0000-0002-9582-5990
e-mail: vohmyanin.yleb@gmail.com

Zhulkovska Inna[2], Candidate of technical sciences, Associate Professor of Department of Cybersecurity and Information Technologies
Жульковська І.І., кандидат технічних наук, доцент кафедри кібербезпеки та інформаційних технологій
ORCID: 0000-0002-6462-4299
e-mail: inivzh@gmail.com

Zhulkovskyi Oleg[1], Candidate of technical sciences, Associate Professor of Systems Software Department
Жульковський О.О., кандидат технічних наук, доцент кафедри програмного забезпечення систем
ORCID: 0000-0003-0910-1150
e-mail: olalzh@ukr.net

Ulianovska Yuliia[2], Candidate of technical sciences, Head of Department of Computer Science and Software Engineering
Ульяновська Ю.В., кандидат технічних наук, завідувач кафедри комп'ютерних наук та інженерії програмного забезпечення
ORCID: 0000-0001-5945-5251
e-mail: yuliyauyv@gmail.com

Mala Yuliia[2], Candidate of technical sciences, Associate Professor of Department of Computer Science and Software Engineering
Мала Ю.А., кандидат технічних наук, доцент кафедри комп'ютерних наук та інженерії програмного забезпечення
ORCID: 0000-0002-2539-4793
e-mail: malaya.ua@gmail.com

[1]Dniprovsky State Technical University, Kamianske
Дніпровський державний технічний університет, Кам'янське
[2]University of Customs and Finance, Dnipro
Університет митної справи та фінансів, Дніпро

FORECASTING DEMAND FOR PRODUCTS USING NEURAL MODELS AND TIME SERIES

ПРОГНОЗУВАННЯ ПОПИТУ НА ПРОДУКЦІЮ ЗА ДОПОМОГОЮ НЕЙРОННИХ МОДЕЛЕЙ ТА ВИКОРИСТАННЯМ ЧАСОВИХ РЯДІВ

The demand for modern manufacturing products is becoming more unpredictable due to rapid changes in consumer tastes, globalization, technological and economic changes, as well as the impact of external factors. In this context, the use of forecasting methods is becoming a critical element of production management strategies. Effective demand forecasts allow not only to adapt production activities to current market conditions, but also to ensure the sustainability of product supply to the mar-

ket, minimizing costs and risks. This paper considers the problem of forecasting demand for modern manufactured products on the example of agricultural products using methods of analyzing available data. The emphasis is on understanding and using statistical and mathematical models for forecasting trends in consumer demand for goods and services. As a result of the study, the ARIMA neural model was trained to forecast demand based on available data for the past period of time using modern tools and the Python programming language. The developed algorithm for training the neural forecasting model is universal and can be further used in metallurgy, machine building, chemical industry and other sectors of the economy.

***Keywords**: smart factory, Industry 4.0, data forecasting, time series, ARIMA neural model.*

Попит на продукцію сучасного виробництва стає більш непередбачуваним через швидкі зміни смаків споживачів, глобалізацію, технологічні і економічні зміни, а також вплив зовнішніх чинників. В цьому контексті використання методів прогнозування стає критично важливим елементом стратегій управління виробництвом. Ефективні прогнози попиту дозволяють не лише адаптувати свою виробничу діяльність до поточних ринкових умов, але і забезпечувати сталість постачання продукції на ринок, мінімізуючи витрати і ризики. У роботі розглядається проблема прогнозування попиту на продукцію сучасного виробництва на прикладі сільськогосподарської продукції із використанням методів аналізу наявних даних. Акцент робиться на розумінні та використанні статистичних та математичних моделей прогнозування тенденцій у споживчому попиті на товари та послуги. Враховуючи розвиток сучасних технологій обробки даних та штучного інтелекту, висвітлено можливості застосування таких новітніх підходів, як машинне навчання для поліпшення точності та ефективності прогнозування. В результаті проведеного дослідження була навчена нейронна модель ARIMA для прогнозування попиту на основі наявних даних за минулий період часу за допомогою сучасних інструментів та мови програмування Python. Стаціонарність вхідного часового ряду була підтверджена за допомогою статистичного тесту Дікі-Фуллера, значення якого склало -4.58, а також отриманого p-значення, яке дорівнює 8.7×10^{-19}. *Якість навченої нейронної моделі була оцінена за допомогою середньої абсолютної помилки та середньоквадратичної помилки. Значення MAE дорівнює 13.23, а MSE — 173.84. Відхилення σ у прогнозуванні склало ~13.19 тон. Також якість моделі підтверджується отриманими прогнозованими результатами на певний період часу в майбутньому. Для візуалізації додаткових характеристик навченої нейронної моделі був побудований графік розподілу частоти даних та Q-Q графік. Розроблений алгоритм навчання нейронної моделі прогнозування на основі наявних даних за минулий період часу є універсальним та може бути у подальшому використаний у металургії, машинобудуванні, хімічній промисловості та інших галузях економіки. Доцільно також порівняти отримані результати з тими, які можуть бути отримані з використанням модифікованої моделі ARIMA — SARIMA, яка враховує сезонність, а отже дозволяє краще прогнозувати попит у певні періоди, наприклад, у відповідь на сезонні тренди чи піки попиту.*

***Ключові слова**: інтелектуальна фабрика, Індустрія 4.0, прогнозування даних, часові ряди, нейронна модель ARIMA.*

Problem's Formulation

In the context of the study of product demand forecasting, special attention is paid to the transitional phenomenon that defines a new stage in economic development — Industry 4.0. This concept is based on the use of modern technologies, such as the Internet of Things (IoT), artificial intelligence (AI), data analytics, blockchain, and others to automate and optimize production processes [1, 2]. The smart factory, which is also part of the Industry 4.0 concept, is a modern approach to production organization that uses advanced technologies and intelligent systems to automate and optimize all stages of the production process. It creates an integrated ecosystem where devices, equipment and systems exchange data in real time to increase production efficiency and simplify management [3]. Intelligent systems can use the data to automatically control equipment. Of particular importance is the ability of systems to adapt production processes in real time to changes in external factors that may affect production productivity and product quality. Additionally, the Intelligent Factory helps to increase automation and reduce costs by optimizing the use of resources, including energy and water. It can implement energy-efficient technologies and management systems to ensure efficient use of resources and reduce the environmental impact of production [4, 5]. The most common use case of the smart factory concept is in an enterprise where the entire production chain, from design to production and management, is fully based on technology and AI. Production processes are subject to optimization, with AI predicting equipment malfunctions, managing inventory, and developing more efficient production methods [2].

To implement the forecasting process, time series are used, which are a sequence of data measured or observed at successive points in time [6, 7]. They are collected in a chronological order, which allows analyzing changes over time and identifying relevant patterns, trends or cyclic fluctuations. Based on the studied data dynamics, they provide an opportunity to make certain decisions. The common characteristics of time series include time, variables, trends, seasonality, and noise.

Smart factories that create an integrated system using Industry 4.0 technologies help to optimize production processes, reducing costs and increasing efficiency [8]. This opens up the possibility to respond quickly to changes that affect production processes. The relevance of the study lies in the use of time series for demand forecasting, allowing to automatically analyze changes over time and identify patterns that are used in making decisions on the production and supply of products.

Analysis of recent research and publications

When applying data processing methods in the field of the Internet of Things, the main focus is on the problems associated with recording data from various sources and possible errors in the electronics of embedded devices, which leads to the emergence of high-dimensional data sets and «concept drift» events. Therefore, we propose a new approach to processing high-dimensional non-stationary time series within the IoT: projecting the original high-dimensional data into a low-dimensional nested space and using the Fuzzy Time Series (FTS) approach for further analysis. The main improvement lies in a more efficient representation of the complex content of non-stationary multidimensional time series and more accurate forecasting. It is claimed [1] that the model is able to explain 98 % of the variance and achieves high forecast accuracy rates, including Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE).

The transformation of traditional production into intelligent production is of interest to manufacturing companies at the global level. The main goal of the Industry 4.0 concept is to create digitalization of production processes, where the flow of information between different devices is controlled with minimal human intervention. Various physical and cybernetic technologies make Industry 4.0 a way to improve productivity, quality, manageability, management and transparency of information in production processes. Despite the numerous potential opportunities provided by Industry 4.0, many companies still lack an understanding of the variety of technologies available within the mentioned concept, as well as the challenges and risks it faces during implementation [3]. The main features of Industry 4.0 involve the use of digital technologies to improve all aspects of industrial production, including processes, products and interaction with customers; connecting systems and devices on production lines to the Internet for data exchange and real-time monitoring; implementation of AI technologies for analysis of large volumes of data, decision-making, optimization of production and prediction of possible problems; combining physical systems with digital counterparts, creating «cyber-physical systems» for more effective management and monitoring; quick response to changes in demand or market requirements; creation of virtual copies of physical objects and processes for more effective management and control [3, 4].

Modern technologies including IoT, Edge Computing, Cloud Computing, AI, Deep Learning and Machine Learning can be used and integrated to create practical implementation examples in manufacturing and other industries. The emphasis is on research into how a combination of these technologies can work together. Instead of traditional methods of tracking inventory, such as ledgers and spreadsheets, solutions [4] are proposed to implement real-time tracking and management of inventory status, even remotely. Inventory data is monitored by deep learning systems and processed on an IoT platform deployed in a cloud service for easy access to tracking and subsequent analysis. So, the technologies of Industry 4.0 technologies are IoT, AI, and supervisory control and management systems (SCADA), which provide real-time monitoring and control of industrial processes, allowing for optimized production operations; technologies for processing and analyzing huge amounts of data collected from various sources in the production process to identify trends, forecast and optimize; cyber-physical systems (CPS), which determine the integration of physical processes with digital information, creating a single system where computers and networks interact with the physical world; storage and processing technologies; and These innovative technologies and tools work together to create an intelligent factory where production processes become more efficient, flexible and sustainable. The result is improved product quality, optimized costs and environmentally sustainable production practices.

One of the existing methods for forecasting future time series uses multivariate historical time series [9]. Historical time series data covers the time variations of various phenomena such as stock prices and economic indicators, as opposed to physical time series data such as voice recordings. Historical time series data is characterized by complex interdependencies between many factors, resembling a complex network of interactions. Unlike physical time series, establishing causal relationships between these factors is often impractical, which creates significant challenges for forecasting tasks. However, through the use of advanced statistical methodologies, the system proposed in [9] seeks to uncover the patterns inherent in historical time series data, facilitating accurate forecasting of future trends. Experimental verification of the concept's effectiveness highlights its potential usefulness in practical forecasting scenarios.

In the process of time series modelling, it is advisable to use the Autoregressive Integrated Moving Average (ARIMA) model with a special emphasis on variance analysis [10]. Variance testing is conducted to assess the suitability of the ARIMA model for modelling seasonal data, which helps in determining the appropriate time periods. Time series analysis and modelling is a fast-growing area of financial econometrics, especially in the field of high-frequency data, which refers to financial information collected at hourly, minute, or even second intervals [10]. The dynamic nature of the data content requires data mining algorithms that can cope with conceptual drift and reflect the evolution of the data. The model itself is a combination of three main components: autoregression (AR), integration (I), and moving average (MA). Autoregression describes the relationship between the current value of a time series and the previous values of the same series. In the AR(p) model, the current value of the series is represented by a linear combination of p previous values of the series, where p is the order of autoregression [11, 12]. Integration (I) is used to stabilize the time series to make it stationary. In a stationary series, the statistical characteristics of the moving average and variance remain constant over time. Integrating a series means subtracting the current value from the previous value, which helps to remove the trend and make the series stationary. Moving average (MA) is a model in which the current value of a time series depends on random errors at previous points in time. In the MA(q) model, the current value of the series is a linear combination of q previous random errors, where q is the order of the moving average. By combining the three components discussed above, the ARIMA(p, d, q) model allows you to model a wide range of time series. Here, p, d, and q stand for the autoregressive, integration, and moving average orders, respectively [13].

Graphs provide a powerful framework for analyzing complex datasets, allowing you to effectively explore relationships within the data [14]. Stationarity, a key characteristic of the analysis of random time signals, helps in their processing and interpretation. However, traditional definitions of stationarity are based on temporal characteristics and cannot be directly applied to graphs due to their irregular structure.

Recent studies have proposed the use of nonlinear time series analysis (NLTSA) [15]. The fundamental concept of NLTSA is stationarity, although its formal definition cannot be directly applied to experimental data. Nevertheless, most NLTSA methods assume stationarity for accurate analysis. For example, the Cross-Prediction Error (CPE) algorithm can be used to estimate the stationarity of time series. The effectiveness of this approach is evaluated [15] in comparison with well-known methods such as PSA and PRPD, as well as NLTSA.

Non-linear methods play a crucial role in the analysis and processing of random sequences, especially when dealing with outliers or impulse noise [16]. Among them are methods based on order statistics, which use rank information to solve problems associated with different problem sizes [16]. For example, it is advisable to use a quadratic model of measuring the first rank using thumbnails and apply it to the order statistics. This approach involves the introduction of a new method for estimating the correlation matrix based on a reduced number of measurements. The estimation is achieved by means of a convex relaxation problem that exploits the structural characteristics of ordered data sequences. To verify the effectiveness of the proposed method, simulations are performed to illustrate the behavior of the estimators. These simulations demonstrate the robustness of the described approach, especially in scenarios where homogeneous noise is present. The proposed approach [16] provides a practical solution for processing random sequences with outliers and impulsive noise, increasing the reliability and accuracy of statistical estimates in various applications.

Based on the discussions on the characteristics of time series data, temporal attention mechanisms, and deep learning methods for time series prediction, an understanding of open datasets, experimental environments, and parameter settings is being developed [17]. In this context, an advanced time series forecasting model PA-LSTM (Predictive Attention Long Short-Term Memory) based on deep learning methodology is being developed. The proposed PA-LSTM model incorporates innovative features and architecture improvements to enhance forecasting accuracy. The effectiveness of the PA-LSTM model has been evaluated [17] using a thorough experimental analysis. The results show that the root mean square logarithmic error and mean absolute error achieved by the PA-LSTM forecasting method as developed in this paper are extremely low, with appropriate values. These figures are superior to those of alternative forecasting methods, which emphasizes the advantages of the PA-LSTM approach.

Formulation of the study purpose

The purpose of this study is to develop and train a neural model for forecasting product demand to improve the efficiency of modern production by using advanced technologies for working with neural models and networks using the Python programming language. To achieve the described goal, the following tasks are defined in the paper:

– collect data on product demand for a certain period of time. To perform preliminary data processing (trimming of anomalous values, data normalization, combining different sources);

– using the existing tools of the Python programming language, develop and train a neural model based on the collected data to predict product demand for a certain period of time in the future;

– using various metrics, it is necessary to check the time series for stationarity and evaluate the trained model, drawing conclusions about the quality and feasibility of its further use.

Presenting main material

One of the most common tools used in time series analysis for data forecasting is the ARIMA model. To apply it to a time series, the optimal values of p, d and q need to be selected, which can be done using selection methods or statistical criteria such as the Akaike Information Criterion (AIC) or the Bayesian Information Criterion (BIC). Among the limitations of the ARIMA model, it is advisable to highlight the linear dependencies between the variables. They may not be sufficient for accurate modelling of some time series. It also does not take into account seasonal changes or changes in the structure of the series, which leads to underestimation of complex data patterns. To take into account seasonality and nonlinear dependencies between variables, appropriate extensions of the ARIMA model have been developed: the seasonal ARIMA model — SARIMA and the autoregressive model with an integrated moving average and seasonal component — SARIMAX [10, 11].

Since many methods and models require stationary data to work correctly, it is advisable to check for unit roots in a time series model using the Dickey-Fuller test (DF test). The P-value is responsible for the probability of obtaining the observed data or more extreme values, provided that the null hypothesis is true. In the context of statistical tests, the smaller the p-value, the stronger the evidence against the null hypothesis and in favor of the stationarity of the time series. If the DF test statistic is less than the critical values, the main hypothesis is rejected in favor of the alternative hypothesis, which indicates that the time series is stationary. Otherwise, the series is considered non-stationary [1].

In order to visualize additional characteristics of a trained neural model, a visual method is often used, which involves the construction of various graphs. A data distribution histogram is a graphical representation of the frequency distribution or relative frequency of variable values, allowing you to get an idea of the data structure and its distribution. The principle of creating a histogram is based on dividing the range of values of a variable into several bins. Then, for each bin, the number of observations falling into that interval is counted and a bar chart is drawn, where the height of each bar corresponds to the frequency or relative frequency of the values in the corresponding interval [18]. The main purpose of this histogram is to visualize the distribution of data and identify the main characteristics, such as mode, mean, median, shape of the distribution, etc. A symmetrical histogram is characterized by an even distribution of values around the central point, indicating that there are no strong asymmetries in the data. An asymmetric histogram, on the other hand, is skewed to one side and may indicate a predominance of values in a particular area. A unimodal histogram has a single mode, while a multimodal histogram has two or more modes. In addition, a histogram can also show outliers and anomalies in the data, which can be sensitive when exploring and analyzing data. Outliers are usually values that are significantly different from the bulk of the data and may indicate the presence of errors or unusual situations.

Another graph is the Q-Q (Quantile-Quantile) graph. This is a graphical visualization method used to compare the theoretical distribution of a variable with the empirical distribution of data [18]. It is a dot plot where the quantiles of the theoretical distribution are displayed on the horizontal axis and the quantiles of the empirical distribution of the data are displayed on the vertical axis. If the data conform to the theoretical distribution, the points on the graph will lie approximately on a straight line with an angular coefficient of 1. Such graphs are widely used to test hypotheses about whether the data conform to a particular theoretical distribution, such as normal, exponential, gamma, and others. They allow you to visually assess how well the data conforms to the assumed theoretical distribution and identify deviations or anomalies in the data. To create a scatter plot, the quantiles of the theoretical distribution and the quantiles of the empirical distribution of the data are first calculated. These values are then ordered in ascending order and points are plotted on the graph, where each point corresponds to a pair of quantiles. If the data fit the theoretical distribution well, the points will lie on a straight line through the origin. The main advantage of Q-Q plots is their ability to quickly and visually check whether data fit a theoretical distribution without the need to calculate statistical tests beforehand. They also help to identify deviations in the data, such as skewness, heavy tails, or outliers, which may not be visible with other methods of analysis. However, it should be borne in mind that Q-Q plots can be sensitive to sample size and may give false positives or false negatives with small samples. Therefore, it is recommended that they be used in conjunction with other methods of checking the distribution of data, such as statistical tests of normality.

Mean absolute error and mean squared error are used in machine learning to assess the quality of models. MAE is the average of the absolute differences between the model's predictions and their actual values. That is, MAE measures the average absolute deviation of predictions from their true values. MSE also measures the difference between the model's predictions and their actual values, but it takes into account the squares of the deviations, which makes it more sensitive to large deviations [1].

Python is the undisputed leader among modern programming languages for training neural models. It has a large number of libraries and tools specifically designed for working with data, including pandas, numpy, scipy, scikit-learn, statsmodels, etc. These libraries provide all the tools you need to work with time series, including functions for loading, processing, visualizing, and modelling data. ARIMA model training methods in Python are based on various libraries and tools, including the above, which provide functionality for time series analysis and ARIMA model building [19]. The aforementioned statsmodels library includes autoregressive, moving average, ARIMA, and their seasonal variants SARIMA models.

To train an ARIMA model in Python using the statsmodels library, there are usually several key steps. First of all, the time series needs to be prepared for analysis by ensuring its stationarity by applying various transformations, such as differentiation, logarithmization, etc. For this purpose, functions and methods provided by the pandas library are used. Then it is advisable to proceed to determining the optimal parameters of the ARIMA model [19]. For this purpose, methods of automatic parameter selection are used, such as, for example, the auto_arima function from the pmdarima library. This method allows you to select the optimal values of the ARIMA model parameters based on the information criteria AIC or BIC, and by searching for different combinations of parameters. After determining the optimal parameters of the ARIMA model, you can start training it. For this purpose, the ARIMA class from the statsmodels library is used. To create an instance of the class, you need to specify the model parameters (p, d, q). After that, the model can be trained using the fit method by passing it a time series as input. After training the model, it is advisable to analyze its quality and evaluate its accuracy. For this purpose, various model quality metrics are used, such as mean squared error, mean absolute error, coefficient of determination R-squared, etc. If the respective metrics demonstrate the inefficiency of the model, it can be refined. Its further use for forecasting time series values for the future is possible using the predict method provided in the ARIMA class. It allows you to get forecasts of time series values for a certain forecasting horizon.

Thus, the methods for training an ARIMA model in Python include data preparation, determining the optimal model parameters, training the model, evaluating its quality, and forecasting time series values. To perform these steps, you should use various libraries: pandas, statsmodels, pmdarima, etc.

For conducting experiments, the goal was formulated to implement and research the process of forecasting demand on the example of agricultural products based on the analysis of available data for a certain period of time using modern technologies for working with neural networks and models. In order to achieve the set goal, tasks were formulated to collect data on the production of agricultural products for a certain period of time; pre-processing of data, such as clipping of anomalous values, normalization of data, merging of different sources, etc.; determination of types of neural networks, models that will be used to forecast the demand for agricultural products; development, training and optimization of the model based on the collected data; testing and validating the model, taking into account performance metrics; development of a system for forecasting the demand for agricultural products based on a trained model; fixation of the obtained results.

The developed Python code uses the pandas, numpy, matplotlib, statsmodels, seaborn and sklearn libraries for data processing, graphing and time series modelling. To process and analyze data, the pandas library is used, which provides various data structures, such as DataFrame, for convenient work with tabular data. It is used to create and manipulate time series data in code, including indexing, data merging, and other operations. To organize work with numeric data and arrays, you can use the NumPy library, which provides functionality for efficient work with arrays and matrices. To visualize the obtained data, the matplotlib library is used to create static, interactive, and animated graphs in Python. The model is built and trained using ARIMA from the statsmodels library for estimating statistical models, including time series. The scikit-learn library is used to calculate metrics to assess the quality of the model. The program code uses mean_squared_error and mean_absolute_error to estimate model errors. The functionality of the sklearn library was used to split the data into training and test sets. 80 % of the data was allocated to the training set and 20 % to the test set. Such proportions are one of the most common options.

The combination of all these technologies allows to analyze time series, train the ARIMA model, make a forecast, and evaluate performance.

The collected input data for training the ARIMA neural model are shown in Tabl. 1. They reflect the daily demand for agricultural products at the enterprise during the year. Based on the input data, a time series is built, which is shown in Fig. 1.

Table 1. Input data for training a neural model

№	Date	Demand, tons
1	01.01.2023	1013,45
2	02.01.2023	995,14
3	03.01.2023	1002,82
4	04.01.2023	1021,57
…		
302	29.10.2023	984,26
303	30.10.2023	944,51
304	31.10.2023	996,23
305	01.11.2023	971,10

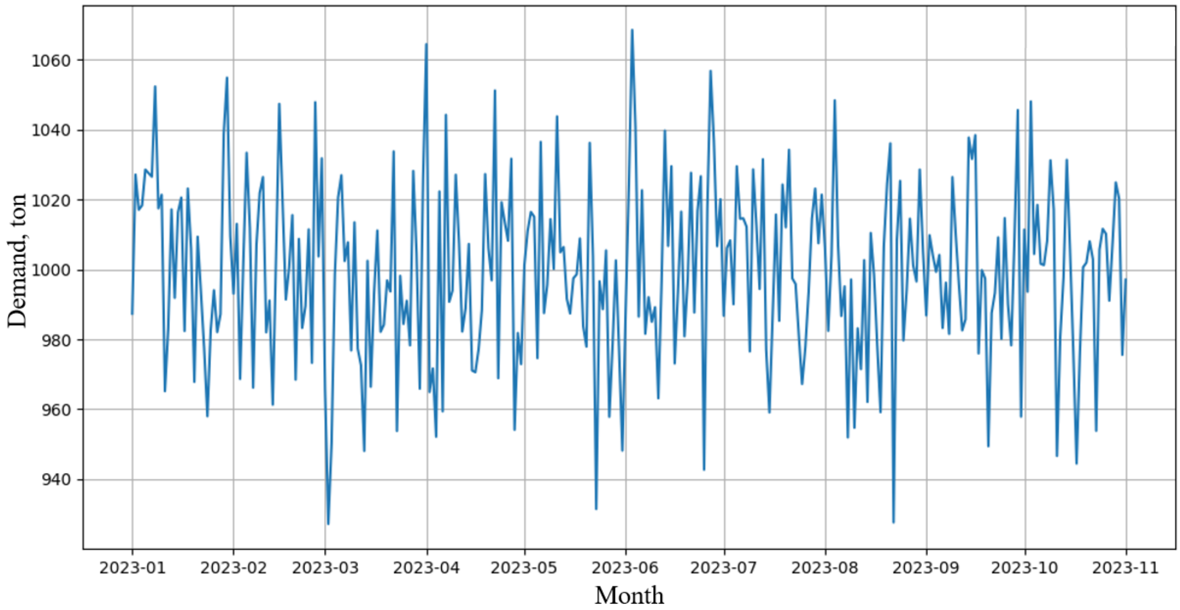

Fig. 1. Time series of input data collected during the year

The Dickey-Fuller test on the collected and structured product demand data yielded a value of -4.58. The P-value is 8.7×10^{-19}. Since the p-value is < 0.05 and the DF test statistic is less than the critical value, the null hypothesis of a single root is rejected, and thus the time series is stationary. After the neural model training process was completed, the MSE error was 173.84 and the MAE was 13.23. The deviation of the trained neural model σ in forecasting is ~13.19 tons, since MSE = σ2.

In order to visualize additional characteristics of the trained neural model, the corresponding graphs were constructed. The histogram with the estimated density (Fig. 2) is a graphical representation of the frequency distribution of the variable values divided into several intervals. At the same time, the estimated probability density is a smooth function that approximates the actual data distribution. The second visual tool is the Q-Q (Quantile-Quantile) plot, a graphical visualization method used to compare the theoretical distribution of a variable with the empirical distribution of data. It is a dot plot (Fig. 3), where the quantiles of the theoretical distribution are displayed on the horizontal axis and the quantiles of the empirical distribution are displayed on the vertical axis.

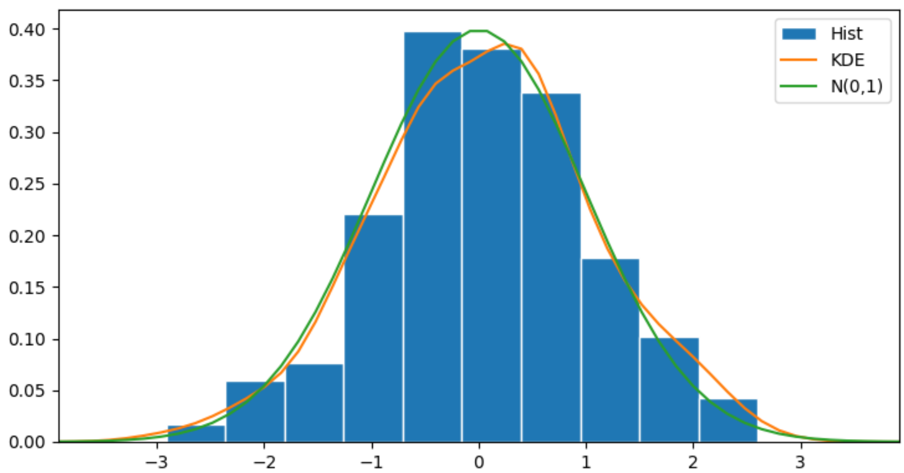

Fig. 2. Histogram of data frequency distribution with estimated density

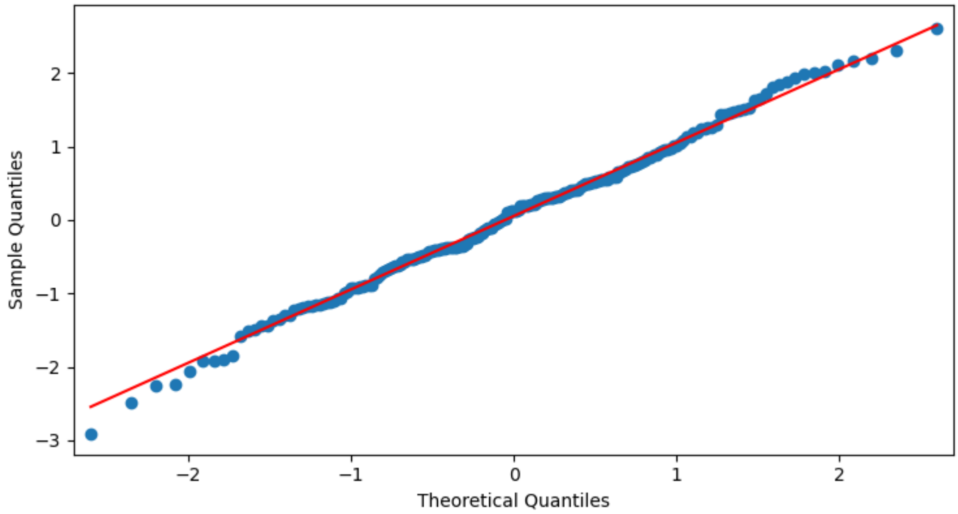

Fig. 3. Quantile-Quantile plot comparing the theoretical distribution of a variable with the empirical distribution of data

From Fig. 2 and Fig. 3 shows that the data distribution is close to normal, which means that the neural model was well trained on the available data. Accordingly, it can be used to forecast product demand for the future period of time, which is confirmed by the MSE and MAE error values and the deviation σ.

Thus, Fig. 4 shows the resulting graph, which reflects the predicted data on product demand for the next month, which were obtained using a pre-trained neural model. The corresponding table with the resulting data (Tabl. 2), as well as the graph (Fig. 4), confirm the quality and reliability of the data obtained as a result of the study.

Table 2. Input data of the forecasted product demand for the next month, obtained using a pre-trained neural model

№	Date	Demand, tons
306	02.11.2023	1039,17
307	03.11.2023	1001,73
308	04.11.2023	982,04
309	05.11.2023	1004,35
310	06.11.2023	989,28
…		
331	27.11.2023	957,84
332	28.11.2023	989,56
333	29.11.2023	1008,47
334	30.11.2023	1022,96
335	01.12.2023	1024,42

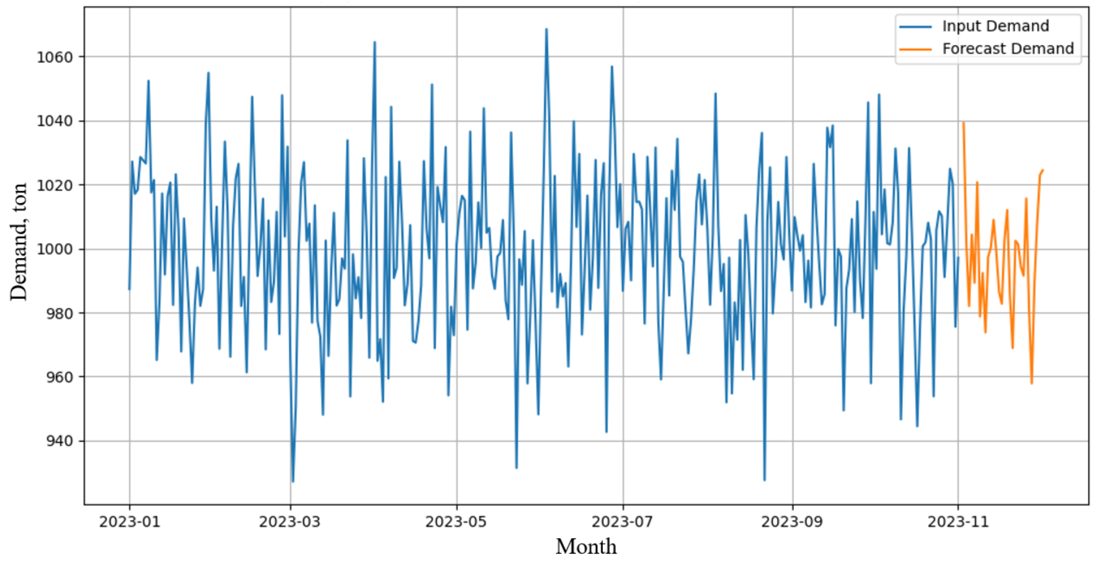

Fig. 4. Results of forecasting product demand for the next month

Conclusions

As a result of the study, the ARIMA neural model was trained to forecast product demand based on available data for the past period of time using modern tools, including the use of the modern Python programming language and its libraries, such as pandas, numpy, scipy, scikit-learn, statsmodels, etc. The developed algorithm for training a neural forecasting model based on available data for the past period of time is universal and can be used in the metallurgy, mechanical engineering, chemical industry, etc.

The stationarity of the input time series was established and confirmed using the Dickey-Fuller statistical test, the value of which was -4.58, and the resulting p-value of 8.7×10^{-19}. The quality of the trained neural model was evaluated using the mean absolute error and the root mean square error. The MAE value is 13.23 and the MSE is 173.84. The deviation of the trained neural model σ in forecasting was ~13.19 tons, which opens up room for further model improvement to reduce the deviation by incorporating exogenous variables into the machine learning process.

To visualize additional characteristics of the trained neural model, we plotted the data frequency distribution (Fig. 2) and the Q-Q plot (Fig. 3). The quality of the model is also confirmed by the predicted results obtained, which were displayed on the resulting graph (Fig. 4) and are given in Table 2.

The developed algorithm for training a neural forecasting model based on the available data for the past period of time is universal and can be further used in metallurgy, machine building, chemical industry and other sectors of the economy. It is also worth comparing the results obtained with those that can be obtained using a modified ARIMA model, SARIMA, which takes into account seasonality and therefore allows for better forecasting of demand in certain periods, for example, in response to seasonal trends or demand peaks.

References

[1] Bitencourt, H. V., Guimaraes, F. G. (2021). High-dimensional Multivariate Time Series Forecasting in IoT Applications using Embedding Non-stationary Fuzzy Time Series. *2021 IEEE Latin American Conference on Computational Intelligence (LA-CCI)* [in English]. DOI: https://doi.org/10.1109/la-cci48322.2021.9769792

[2] Zhao, T., Liu, L., Liu, S., Xie, Y., Qiu, X., Chen, Y. (2022). Research on Production Simulation and scheduling of printing Shop for Intelligent Factory. *2022 IEEE 10th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)* [in English]. DOI: https://doi.org/10.1109/itaic54216.2022.9836632

[3] Kalsoom, T., Ramzan, N., Ahmed, S. (2020). Societal Impact of IoT-Lead Smart Factory in the Context of Industry 4.0. *2020 International Conference on UK-China Emerging Technologies (UCET)* [in English]. DOI: https://doi.org/10.1109/ucet51115.2020.9205484

[4] Vaddadi, S., Srinivas, V., Reddy, N. A., Devipriya, A. (2022). Factory Inventory Automation using Industry 4.0 Technologies. *2022 IEEE IAS Global Conference on Emerging Technologies (GlobConET)* [in English]. DOI: https://doi.org/10.1109/globconet53749.2022.9872416

[5] Cui, S., Chen, M., Zhang, Y., He, L. (2020). Study on Decoupling Control System of Temperature and Humidity in Intelligent Plant Factory. *2020 IEEE 9th Joint International Information Technology and Artificial Intelligence Conference* (ITAIC) [in English]. DOI: https://doi.org/10.1109/itaic49862.2020.9339036

[6] Banerjee, S., Martin, R. R., Pardo, A. (2022). Frequency-aware Time Series Forecasting, Anomaly Detection, Classification and Granger Causality. *2022 14th International Conference on COMmunication Systems & NETworkS (COMSNETS)* [in English]. DOI: https://doi.org/10.1109/comsnets53615.2022.9668359

[7] Ostroski, D., Slovenec, K., Brajdic, I., Mikuc, M. (2021). Anomaly Correction in Time Series Data for Improved Forecasting. *2021 16th International Conference on Telecommunications (ConTEL)* [in English]. DOI: https://doi.org/10.23919/contel52528.2021.9495986

[8] Chen, X., Yu, Y., Huang, Q., Shen, M., Wang, S., Zhang, L. (2023). Time Series-Based Electric Load Forecasting with Mixture of Expert System. *2023 4th International Conference on Intelligent Computing and Human-Computer Interaction* (ICHCI) [in English]. DOI: https://doi.org/10.1109/ichci58871.2023.10277743

[9] Rokui, J. (2021). Historical time series prediction framework based on recurrent neural network using multivariate time series. *2021 10th International Congress on Advanced Applied Informatics (IIAI-AAI)* [in English]. DOI: https://doi.org/10.1109/iiai-aai53430.2021.00084

[10] Zhang, M. (2023). Financial Time Series Frequent Pattern Mining Algorithm Based on Time Series ARIMA Model. *2023 International Conference on Networking, Informatics and Computing (ICNETIC)* [in English]. DOI: https://doi.org/10.1109/icnetic59568.2023.00057

[11] Gupta, A., Kumar, A. (2020). Mid Term Daily Load Forecasting using ARIMA, Wavelet-ARIMA and Machine Learning. *2020 IEEE International Conference on Environment and Electrical Engineering and 2020 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe)* [in English]. DOI: https://doi.org/10.1109/eeeic/icpseurope49358.2020.9160563

[12] Lu, F., Lv, J., Zhang, Y., Liu, H., Zheng, S., Li, Y., Hong, M. (2021). Ultra-Short-Term Prediction of EV Aggregator's Demond Response Flexibility Using ARIMA, Gaussian-ARIMA, LSTM and Gaussian-LSTM. *2021 3rd International Academic Exchange Conference on Science and Technology Innovation (IAECST)* [in English]. DOI: https://doi.org/10.1109/iaecst54258.2021.9695933

[13] Gupta, A., Sharma, K. C., Vijayvargia, A., Bhakar, R. (2019). Very Short term Wind Power Prediction Using Hybrid Univariate ARIMA-GARCH Model. *2019 8th International Conference on Power Systems (ICPS)* [in English]. DOI: https://doi.org/10.1109/icps48983.2019.9067611

[14] Güneyi, E. T., Vural, E. (2020). Investigation of Stationarity for Graph Time Series Data Sets. *2020 28th Signal Processing and Communications Applications Conference (SIU)* [in English]. DOI: https://doi.org/10.1109/siu49456.2020.9302376

[15] Donoso, P., Schurch, R., Ardila-Rey, J., Montana, J. (2022). Stationarity in Partial Discharge Time Series of Electrical Trees. *2022 IEEE Conference on Electrical Insulation and Dielectric Phenomena (CEIDP)* [in English]. DOI: https://doi.org/10.1109/ceidp55452.2022.9985359

[16] Hoyos S., J. P., Pazos, S., Jojoa, P., Arce, G. (2017). Correlation matrix estimation of ordered data using sketches. *2017 XVII Workshop on Information Processing and Control (RPIC)* [in English]. DOI: https://doi.org/10.23919/rpic.2017.8214327

[17] Sun, J., Guo, W. (2023). Time Series Prediction Based on Time Attention Mechanism and LSTM Neural Network. *2023 IEEE International Conference on Integrated Circuits and Communication Systems (ICICACS)* [in English]. DOI: https://doi.org/10.1109/icicacs57338.2023.10099498
[18] Chudo, S. B. (2022). Multiplicative Seasonal ARIMA Modeling and Forecasting of COVID_19 Daily Deaths in Hungary. *2022 10th International Conference on Bioinformatics and Computational Biology (ICBCB)* [in English]. DOI: https://doi.org/10.1109/icbcb55259.2022.9802498
[19] Shoaga, G. O., Ikuzwe, A., Gupta, A. (2022). Forecasting of Monthly Hydroelectric and Solar Energy in Rwanda using SARIMA. *2022 IEEE PES/IAS PowerAfrica* [in English]. DOI: https://doi.org/10.1109/powerafrica53997.2022.9905311

Список використаної літератури

1. Bitencourt H. V., Guimaraes F. G. High-dimensional Multivariate Time Series Forecasting in IoT Applications using Embedding Non-stationary Fuzzy Time Series. *2021 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*, Temuco, Chile, 2–4 November 2021. 2021. URL: https://doi.org/10.1109/la-cci48322.2021.9769792
2. Zhao T., Liu L., Liu S., Xie Y., Qiu X., Chen Y. Research on Production Simulation and scheduling of printing Shop for Intelligent Factory. *2022 IEEE 10th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, Chongqing, China, 17–19 June 2022. 2022. URL: https://doi.org/10.1109/itaic54216.2022.9836632
3. Kalsoom T., Ramzan N., Ahmed S. Societal Impact of IoT-Lead Smart Factory in the Context of Industry 4.0. *2020 International Conference on UK-China Emerging Technologies (UCET)*, Glasgow, United Kingdom, 20–21 August 2020. 2020. URL: https://doi.org/10.1109/ucet51115.2020.9205484
4. Vaddadi S., Srinivas V., Reddy N. A., Devipriya A. Factory Inventory Automation using Industry 4.0 Technologies. *2022 IEEE IAS Global Conference on Emerging Technologies (GlobConET)*, Arad, Romania, 20–22 May 2022. 2022. URL: https://doi.org/10.1109/globconet53749.2022.9872416
5. Cui S., Chen M., Zhang Y., He L. Study on Decoupling Control System of Temperature and Humidity in Intelligent Plant Factory. *2020 IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, Chongqing, China, 11–13 December 2020. 2020. URL: https://doi.org/10.1109/itaic49862.2020.9339036
6. Banerjee S., Martin R. R., Pardo A. Frequency-aware Time Series Forecasting, Anomaly Detection, Classification and Granger Causality. *2022 14th International Conference on COMmunication Systems & NETworkS (COMSNETS)*, Bangalore, India, 4–8 January 2022. 2022. URL: https://doi.org/10.1109/comsnets53615.2022.9668359
7. Ostroski D., Slovenec K., Brajdic I., Mikuc M. Anomaly Correction in Time Series Data for Improved Forecasting. *2021 16th International Conference on Telecommunications (ConTEL)*, Zagreb, Croatia, 30 June – 2 July 2021. 2021. URL: https://doi.org/10.23919/contel52528.2021.9495986
8. Chen X., Yu Y., Huang Q., Shen M., Wang S., Zhang L. Time Series-Based Electric Load Forecasting with Mixture of Expert System. *2023 4th International Conference on Intelligent Computing and Human-Computer Interaction (ICHCI)*, Guangzhou, China, 4–6 August 2023. 2023. URL: https://doi.org/10.1109/ichci58871.2023.10277743
9. Rokui J. Historical time series prediction framework based on recurrent neural network using multivariate time series. *2021 10th International Congress on Advanced Applied Informatics (IIAI-AAI)*, Niigata, Japan, 11–16 July 2021. 2021. URL: https://doi.org/10.1109/iiai-aai53430.2021.00084
10. Zhang M. Financial Time Series Frequent Pattern Mining Algorithm Based on Time Series ARIMA Model. *2023 International Conference on Networking, Informatics and Computing (ICNETIC)*, Palermo, Italy, 29–31 May 2023. 2023. URL: https://doi.org/10.1109/icnetic59568.2023.00057

11. Gupta A., Kumar A. Mid Term Daily Load Forecasting using ARIMA, Wavelet-ARIMA and Machine Learning. *2020 IEEE Ind. Commercial Power Syst. Europe (EEEIC / I&CPS Europe)*, Madrid, Spain, 9–12 June 2020. 2020. URL: https://doi.org/10.1109/eeeic/icpseurope49358.2020.9160563
12. Lu F. Ultra-Short-Term Prediction of EV Aggregator's Demond Response Flexibility Using ARIMA, Gaussian-ARIMA, LSTM and Gaussian-LSTM. *2021 3rd Int. Academic Exchange Conf. Sci. Technol. Innov. (IAECST)*, Guangzhou, China, 10–12 December 2021. 2021. URL: https://doi.org/10.1109/iaecst54258.2021.9695933
13. Gupta A., Sharma K. C., Vijayvargia A., Bhakar R. Very Short term Wind Power Prediction Using Hybrid Univariate ARIMA-GARCH Model. *2019 8th Int. Conf. Power Syst. (ICPS)*, Jaipur, India, 20–22 December 2019. 2019. URL: https://doi.org/10.1109/icps48983.2019.9067611
14. Güneyi E. T., Vural E. Investigation of Stationarity for Graph Time Series Data Sets. *2020 28th Signal Processing and Communications Applications Conference (SIU)*, Gaziantep, Turkey, 5–7 October 2020. 2020. URL: https://doi.org/10.1109/siu49456.2020.9302376
15. Donoso P., Schurch R., Ardila-Rey J, Montana J. Stationarity in Partial Discharge Time Series of Electrical Trees. *2022 IEEE Conference on Electrical Insulation and Dielectric Phenomena (CEIDP)*, Denver, CO, USA, 30 October – 2 November 2022. 2022. URL: https://doi.org/10.1109/ceidp55452.2022.9985359
16. Hoyos J. P., Pazos S., Jojoa P., Arce G. Correlation matrix estimation of ordered data using sketches. *2017 XVII Workshop on Information Processing and Control (RPIC)*, Mar del Plata, 20–22 September 2017. 2017. URL: https://doi.org/10.23919/rpic.2017.8214327
17. Sun J., Guo W. Time Series Prediction Based on Time Attention Mechanism and LSTM Neural Network. *2023 IEEE International Conference on Integrated Circuits and Communication Systems (ICICACS)*, Raichur, India, 24–25 February 2023. 2023. URL: https://doi.org/10.1109/icicacs57338.2023.10099498
18. Chudo S. B. Multiplicative Seasonal ARIMA Modeling and Forecasting of COVID_19 Daily Deaths in Hungary. *2022 10th Int. Conf. Bioinf. Comput. Biol. (ICBCB)*, Hangzhou, China, 13–15 May 2022. 2022. URL: https://doi.org/10.1109/icbcb55259.2022.9802498
19. Shoaga G. O., Ikuzwe A., Gupta A. Forecasting of Monthly Hydroelectric and Solar Energy in Rwanda using SARIMA. *2022 IEEE PES/IAS PowerAfr*, Kigali, Rwanda, 22–26 August 2022. 2022. URL: https://doi.org/10.1109/powerafrica53997.2022.9905311

Надійшла до редколегії 01.04.2024