Intelligent automation system of the railway company’s purchasing activity based on artificial neural networks

D K Gek¹, V V Kukartsev¹,², I V Markevich¹, D S Shalaeva¹, A S Yadrov² and D V Rogova¹

¹Reshetnev Siberian State University of Science and Technology, 31, Krasnoyarsky Rabochy Av., Krasnoyarsk, 660037, Russia
²Siberian Federal University, 79, Svobodny pr., Krasnoyarsk, 660041, Russia

E-mail: vlad_saa_2000@mail.ru

Abstract. A person is responsible for quality control of the purchase forecast of commercial enterprises, which entails a number of certain consequences. In this regard, we decided to create an intelligent system that will determine the quality of the forecast. Such a system will be a human assistant in deciding on the quality of the forecast. The system will also be able, if necessary, to indicate the necessary adjustments for its forecasting method. As part of this work, a software product will be created that will determine the quality of the created trade procurement plan, and, if necessary, help the user adjust the forecasting method used by the outlet. The aim of the work is to create an intelligent procurement system that will generate its forecast using a neural network, and then determine the forecast error using XYZ analysis.

1. Introduction
To understand what will happen as a result of development, it is necessary to define an intelligent system.

An intelligent system (IS) is a technical or software system capable of solving problems traditionally considered creative [1 - 3], belonging to a specific subject area, the knowledge about which is stored in the memory of such a system [4 - 6]. The structure of an intelligent system includes three main blocks - a knowledge base, a mechanism for choosing solutions and a program interface [7 - 9].

The knowledge base is an important component of an intelligent system. The most famous class of such programs is expert systems [10 - 12]. They are designed to search for ways to solve problems from a certain subject area, based on knowledge base records and a user description of the situation [13 - 15].

Simple knowledge bases can be used to create expert data storage systems in an organization: documentation, manuals, technical articles. The main purpose of creating such databases is to help less experienced people find an already existing description of a way to solve a problem [3, 7, 16].

The decision-making mechanism is a generalized procedure for finding a solution to a problem, which, based on the knowledge base and in accordance with the user's information needs, builds a chain of reasoning (logically related units of knowledge) leading to a specific result [1, 5].

The user interface provides the transfer of information by the user of the application and the software and hardware complex [6, 8, 11]. The user enters the necessary data, which the system processes, and then, using the interface, displays information that is understandable to the user.

Predicting the purchase of an item is a regression problem. In a regression problem, the sample usually looks like:
\[ S = \{(x_1, y_1), \ldots, (x_i, y_i)\} \subseteq \mathbb{R}^n \times \mathbb{R}. \]  

(1)

The answers are real numbers (maybe also vectors from real numbers). The regression task is to predict the values of this function at other points using the known values of \( y_i \) of some unknown function at given points \( x_i \) (\( i = 1, \ldots, n \)).

The creation of a procurement plan was and remains an urgent problem; therefore, there are currently many products that are engaged in the creation of products. The most successful companies charge a lot of money for their product, but the main amount goes to the fact that they have to apply an individual approach to each client.

The modern market is very changeable, in order to create the most accurate product, they conduct a point-of-sale analysis to find the most accurate parameters that the system will use in forecasting.

Supply of goods is a very difficult regression task, the solution of which depends on many factors.

The simplest and most accessible methods can only indicate a trend when creating a forecast. This conclusion was made thanks to a review of software products. Results of the survey are given below.

2. Review of programs and methods for forecasting purchases

In the course of analyzing the market for ready-made software products, a large number of ready-made software solutions were found for creating a purchase forecast for a commercial enterprise by various methods. Some ready-made solutions are Excel tools for inventory management, BUY © ER system, inventory management system on Access, analytical Internet service for inventory management and procurement planning, procurement assistant, KORUS - Inventory management, SCM: Purchase planning, FORECAST & SUPPLY, KAIS AutoStandard, oremax.

Such a large number of programs prove that the topic of forecasting is in demand, and also that there is no universal program, because these programs have many differences, starting from the price and ending with the forecasting method.

Each method has its own percentage of forecast error. There are applications in this list that, just like the system being developed, use neural networks to create a forecast. Other methods worth noting are OLAP cubes, trendline, least squares, etc.

The more and more varied data a trading enterprise has, the more difficult it is to forecast. Simple methods are not capable of making accurate predictions with large and varied amounts of data, and programs with more modern methods are very expensive.

One of the most popular and complex forecasting methods from the point of view of implementation is a neural network. The developed system generates its forecast using an artificial neural network.

To implement the neural network of the project, the Python programming language was used due to the large availability of ready-made libraries for creating neural networks. For the basis, an open library from Google - Tensorflow and an open-source library for deep neural networks Keras were chosen.

To create the application interface, QtDesigner will be used - a cross-platform free environment for developing graphical interfaces (GUI) programs using the Qt library, a wrapper for the Tkinter Python PySimpleGUI API. The MatLab library for Python will be used to create graphs. First, you need to download the data for training the neural network. The data was previously digitized. After loading, the data needs to be transformed for more accurate neural network operation. We replace all data types with numeric so that the neural network does not have problems during data normalization. Baseline data is the sale of supermarket goods for a certain period.

Input data is entered into the program, after that they are transformed, unnecessary columns are removed, and the products are grouped. Part of the input transformed data is shown in Figure 1.
After loading, output part of the data, as well as the description of the input data, the next step is to normalize the input data. Before normalization, it is necessary to split the DataSet into training and test samples.

Input normalization is a process in which all input data goes through a process of "flattening", i.e. reduction to a certain interval, for example [0,1] or [-1,1] [20]. Normalization can significantly increase the convergence rate of the neural network learning algorithm. If you do not perform normalization, then the input data will have an additional effect on the neuron, which can lead to incorrect decisions. In other words, you cannot compare quantities of different orders.

For normalization, the Min-Max function was used:

$$X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}}$$  \hspace{1cm} (2)

We separate target values or "labels" from features. Now we need to create a neural network model. Our model will be sequential, each layer will be the input for a new layer. Next there is the training of the neural network directly.

Figure 2 shows the indicators of the mean square error and the mean absolute error when training a neural network.

After testing, we need to check the created neural network against test data. The result of the neural network forecast is shown in Figure 3.
The prediction result is very accurate, now you need to look at the graph of the distribution of prediction errors, which is shown in Figure 4.

**Figure 4.** Prediction error distribution.

3. **XYZ Analysis Module Implementation**

To understand the work of XYZ analysis, it is necessary to introduce two concepts of the deviation of the variation series and the coefficient of variation.

The variation series deviation is the standard deviation of a product over a known period. The coefficient of variation is the stability of product sales over a known period. It is calculated using the formula:

$$ v = \frac{\sigma}{x} \times 100\%, $$  \hspace{1cm} (3)

where $\sigma$ is the standard deviation; $x$ is the average number of items.

After all calculations, our products are divided into categories X, Y, Z. The neural network made a forecast, the system finds deviations between real data and those received from the neural network. If
the result falls within the specified scope of the product category, then the system can recommend the user to purchase the product based on the neural network forecast.

For example, a standard deviation of 100 can have a product with an average monthly sale of 200 or 20,000. In the first case, the significance of monthly fluctuations will be 50%, in the other case - 0.5%. It is obvious that the sales of the second product are much more stable and, as a result, more predictable, which means that the forecast for them needs to be as accurate as possible. The first product is poorly predictable, so the system can only suggest the sales trend for product number 1.

Figure 5. Product variation coefficient.

Figure 5 shows the coefficient of variation for each product from the original data.

Now you need to design your user interface layouts. To get started, the user must download the initial data for creating a forecast, as well as data on the forecast results of the custom forecasting method.

Figure 6 shows the layout of the application start window.

Figure 6. Data loading window prototype for application operation.

Next, the user should look at the forecast for each of the selected products, which have already been divided into groups thanks to the XYZ analysis.

Figure 7 shows the layout of the application window, where the user can see the distribution of the product by groups X, Y, Z.
Further, the system should produce a result with a recommendation about the quality of the forecast and the forecasting method used. The layout of the application window, where the system did not find deviations in errors, is shown in Figure 8.

Figure 8. Example of a "Successful Result" result.

Figure 9 shows the possible results when the system found an error in the forecast, as well as an example of the result when the system advises to change the forecasting method for a given outlet.

Figure 9. An example of a "negative" result output layout.

4. Experimental study
For ease of use and to increase the speed of the program, the layouts of the application have been slightly modified. The test launch was performed on the data from Figure 1. After starting the program, the user sees the start screen of the program, which is shown in Figure 10.
One needs to download the initial data for the developed network, as well as the data of the available forecast. If necessary, one needs to indicate the date of the marketing campaign.

Then the neural network is trained, and also creates a forecast.

Before creating a forecast, the system divides products into 3 groups according to the XYZ analysis algorithm. The result of data splitting is shown in Figure 11.

The user can view the graphs of the error and the forecast result by clicking on the menu items.

Then the system analyzes the forecast and initial data and gives out the problems found. Figure 12 shows the result of analyzing the program when running on test data.

As a result, the system did not find any problems in the created forecast, so the user can order goods according to the system forecast.

Now it is necessary to take stock of the research and evaluate the created program.

5. Conclusion
After conducting market research of programs for creating a procurement forecast, improvements in programs and methods can be noted, but there are still problems that do not allow creating the most accurate forecast.

Until now, a person has to make a decision about the quality of the created forecast and whether it is worth purchasing goods according to the created forecast.

The developed program should help outlets in making this decision.
References

[1] Meystel A M and Albus J S 1997 Intelligent Systems (Architecture: Design and Control)
[2] Grosan C and Abraham A 2011 Intelligent systems (Berlin: Springer)
[3] Rudas I J and Fodor J 2008 Intelligent systems International Journal of Computers, Communications & Control 3(3) 132-8
[4] Tynchenko V S, Tynchenko V V, Bukhtoyarov V V, Tynchenko S V and Petrovskyi E A 2016 The multi-objective optimization of complex objects neural network models Indian Journal of Science and Technology 9(29) 99467
[5] Milov A V, Tynchenko V S and Petrenko V E 2018 Algorithmic and software to identify errors in measuring equipment during the formation of permanent joints In: 2018 Int. Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon) pp. 1-5
[6] Hayes-Roth B 1995 An architecture for adaptive intelligent systems Artificial intelligence 72(1-2) 329-65
[7] Aronson J E, Liang T P and MacCarthy R V 2005 Decision support systems and intelligent systems (Upper Saddle River, NJ, USA: Pearson Prentice-Hall)
[8] Stephanopoulos G and Han C 1996 Intelligent systems in process engineering: A review Computers & Chemical Engineering 20(6-7) 743-91
[9] Tynchenko V, Milov A, Tynchenko V, Bukhtoyarov V and Antamoshkin O 2018 Application of artificial neural networks for identification of non-normative errors in measuring instruments for controlling the induction soldering process International Multidisciplinary Scientific GeoConference: SGEM 18(2.1) 117-24
[10] Bukhtoyarov V V, Tynchenko V S, Petrovskyi E A, Tynchenko S V and Milov A V 2019 Intelligently informed control over the process variables of oil and gas equipment maintenance International Review of Automatic Control 12(2) 59-66
[11] Gregor S and Benbasat I 1999 Explanations from intelligent systems: Theoretical foundations and implications for practice MIS quarterly 497-530
[12] Abraham A 2003 Intelligent systems: Architectures and perspectives Recent advances in intelligent paradigms and applications (Physica: Heidelberg) 1-35
[13] Sfidari E, Kadkhodaie-Ilkchi A and Najjari S 2012 Comparison of intelligent and statistical clustering approaches to predicting total organic carbon using intelligent systems Journal of Petroleum Science and Engineering 86 190-205
[14] Bukhtoyarov V V, Tynchenko V S, Petrovsky E A, Kukartsev V V and Kuklina A I 2018 Evolutionary method for automated design of models of vortex flowmeters transformation function Journal of Physics: Conference Series 1118(1) 012041
[15] Subhadra K and Vikas B 2019 Neural network based intelligent system for predicting heart disease International Journal of Innovative Technology and Exploring Engineering 8(5) 484-7
[16] Milov A V, Tynchenko V S, Kukartsev V V, Tynchenko V V and Bukhtoyarov V V 2018 Use of artificial neural networks to correct non-standard errors of measuring instruments when creating integral joints Journal of Physics: Conference Series 1118(1) 012037