UDK 004.773
IRSTI 20.15.05

https://doi.org/10.55452/1998-6688-2024-21-2-65-82

[1]Uvaliyeva I.,
PhD, ORCID: 0000-0002-2117-5390, e-mail: iuvalieva@ektu.kz.
[2]Rustamov S.,
PhD, Associate Professor, ORCID: 0000-0002-3247-5882, e-mail: Srustamov@ada.edu.az
[3]*Belginova S.,
PhD, ORCID: 0000-0002-7238-6016, e-mail: sbelginova@gmail.com
[1]Rakhmetullina Zh.,
Candidate of Physical- Mathematical Sciences, ORCID: 0000-0002-0554-7684,
e-mail: zhrakhmetullina@ektu.kz

[1]D. Serikbayev East Kazakhstan Technical University, 070004, Ust-Kamenogorsk, Kazakhstan
[2]ADA University, Baku, Azerbaijan
[3]University «Turan», 050013, Almaty, Kazakhstan

DEVELOPMENT OF A WEB APPLICATION FOR DATA ANALYSIS AND VISUALIZATION TO ENSURE THE SAFETY OF TECHNOLOGICAL PROCESSES

Abstract

Modern technologies have developed quite well lately, and in addition to the banal data collection and processing, modern algorithms are able to predict the behavior of equipment and prevent the occurrence of malfunctions and failures in advance and take actions to reduce the likelihood of an accident and warn the relevant employees. Within the framework of this study, the authors proposed the development of a web-application for the analysis and visualization of data from UK-TPP to ensure the safety of technological processes. Ust-Kamenogorsk TPP is a regional thermal power plant, heat and electricity generation covers 80% of the load of the housing and communal sector of the city and industrial enterprises. The TPP is part of the unified energy system of Kazakhstan. The application being developed has a web interface that will display and visualize data from the main sensors and instruments of the UK TPP, as well as prevent equipment malfunctions and analyze the behavior of sensors using machine learning. Node.js was chosen as the main software platform used, built on the java script programming language, using additional modules that expand the capabilities of this programming language, written in c ++, which speeds up working with data arrays, receiving, and sending them.

Key words: data analysis, visualization, safety of technological processes, UK-TPP, device, Web-application, monitoring, node.js, java script.

Introduction

Analysis and visualization of large data arrays is a topical issue in many areas of activity that use data arrays in their work. The issue of automation of work with data is an important aspect in the modern world, the arrays of data used are only increasing every year and there are not enough human resources for a one-time continuous analysis and monitoring of indicators [1]. The issue of automation is extremely acute in such enterprises as a combined heat and power plant since the issue of analyzing and tracking data in such enterprises is the most important aspect of work. Analysis and timely search for critical indicators can prevent equipment breakdown and emergency situations. Monitoring the behavior of equipment, timely transmission of data, the use of visualization modules, ensuring the safety of technological processes by means of alerting users are key in terms of solving this issue [2].

Among the latest modern tools, advanced technologies based on machine learning are used, which, with the help of machine code learning, are able to solve the problems of analyzing and predicting the behavior of large data sets at a level close to human thinking, but at the same time, many times higher than the performance and speed of human thinking, which is many times more effective than an ordinary employee [3].

In addition, the program can independently analyze critical data and inform the relevant employees about the behavior of faulty equipment, visualize the change in sensors over time [4]. To analyze the emergencies that have occurred, the arrays of processed data are recorded in various databases, using backup, so that even in the event of global emergencies, the picture of the events that have occurred can be restored.

To optimize the analysis of indications of various equipment, web interfaces are created so that employees can analyze the behavior of the equipment they need live. Modern interfaces can be viewed using mobile devices, thanks to which employees are not "tied" to the workplace because of the computer, which greatly optimizes the work with data [5]. For the same purpose, to alert about the occurrence of emergencies, bots are used that send alerts in various popular social networks. This is an effective solution, since every modern person has a mobile device at hand, which will significantly speed up the notification of the relevant employees and taking measures to eliminate existing problems.

The aim of the study is to develop a web-application for the analysis and visualization of data from UK-TPP to ensure the safety of technological processes. The application being developed has a web interface that will display and visualize the data of the main sensors and instruments of the UK TPP [6], prevent the occurrence of equipment malfunctions, and analyze the behavior of sensors using machine learning.

The main task of the thermal power plant is to provide hot water and electricity supply services for consumers. Node.js was chosen as the main software platform used [7], built on the java script programming language, using additional modules that expand the capabilities of this programming language, written in C ++ [8], which speeds up working with data arrays, receiving, and sending them.

As part of the study, the node-red visual programming interface was chosen [9], since it is quite easy to use and configure, which does not require great programming skills and work with other software, therefore, to work with this web interface, you do not need to have great skills in working with programs, which will simplify the introduction of a new program. Java Script is the main programming language of this platform [10], and all the code of each module is written in it, also, using additional nodes, you can connect third-party libraries of other programming languages, such as python [11].

Methology

The role of machine learning in the technological process of ensuring security

Machine learning plays a key role in the creation of intelligent monitoring and control systems that help prevent equipment failures and ensure the continuous and safe operation of production processes, such as:

- Anomaly detection: Machine learning allows you to create models that can analyze data from sensors and identify abnormal behavior of equipment that may indicate a potential malfunction. This may be a deviation from normal operating parameters, as well as the appearance of specific signals or patterns preceding a malfunction.
- Predicting the probability of failure: Machine learning models can be used to predict the probability of equipment failure in the future based on its current state and already existing data. This helps to take proactive measures to prevent breakdowns and plan maintenance.

- Adaptive Hardware Management: Machine learning models can be integrated into hardware management systems for real-time decision making. For example, based on the analysis of sensor data, the model can automatically adjust the operating parameters of the equipment to minimize the risk of malfunctions.
- Maintenance optimization: Machine learning helps to optimize the maintenance schedule of equipment. Based on predicting the probability of failure and analyzing data on the condition of equipment, optimal maintenance strategies can be developed, for example, to carry out maintenance when it is really necessary, and not according to a strict schedule.
- Improving production efficiency: Fault prevention using machine learning helps to reduce equipment downtime and avoid loss of production capacity due to unexpected failures. This helps to increase the efficiency and cost-effectiveness of production processes.

The process of preventing equipment malfunction using machine learning and analyzing sensor behavior can be described as follows:

1) Sensor selection and equipment installation. At the beginning of the process, the necessary sensors are determined to monitor various equipment parameters. These include: temperature, vibration, pressure, liquid level sensors and others, depending on the type of equipment.
2) Data collection. Sensors are installed on the equipment, and data collection on its operation begins.
3) Data preprocessing. In order for the data to be ready for analysis, it must be preprocessed. This includes noise filtering, elimination of outliers, normalization of values.
4) Modeling and training. At this stage, machine learning methods are used to create a model capable of predicting possible malfunctions. Classification and regression algorithms are used, depending on the specific task. The model is trained on existing data that contains information about the condition of the equipment and actual malfunctions.
5) Model implementation. After training the model, it is ready for use in real time. The model can be integrated with an equipment monitoring system for continuous analysis of data coming from sensors.
6) Anomaly detection and warning. When the equipment is in operation, the model analyzes data from sensors in real time and identifies abnormal or unusual indicators that may indicate a potential malfunction. If such anomalies are detected, a warning or an automatic signal is generated to the operator or control system to take the necessary measures.
7) Model update. New data on the operation of the equipment is constantly being collected, and this data can be used to continuously improve the model. The model is periodically updated and retrained on new data, which improves its accuracy and efficiency.

This process allows you to quickly detect potential problems with the equipment, minimizing the risk of accidents and increasing the efficiency of its operation.

Technological processes of UK TPP

Ust-Kamenogorsk TPP is a regional thermal power plant, in which the generation of heat and electricity covers 80% of the load of the housing and communal sector of the city and industrial enterprises. The TPP is part of the unified energy system of Kazakhstan [12].

Ust-Kamenogorsk TPP was launched in 1947, for the first time in Kazakhstan, high-pressure equipment was installed on it. In Soviet times, it was part of the regional energy management department, covering the territories of the East Kazakhstan and Semipalatinsk regions [13]. In 1997, it was among the six power plants transferred to the American energy corporation AES. In April 2017, AES sold the Ust-Kamenogorsk and Sogrinsk TPPs to the Kazakh company Kazakhstan Utility Systems [14].

The main purpose of the combined heat and power plant, as the name suggests, is the generation of electricity and hot water to cover the needs of consumers.

The full cycle of operation of the UK TPP is shown in Figure 1 (p. 68).

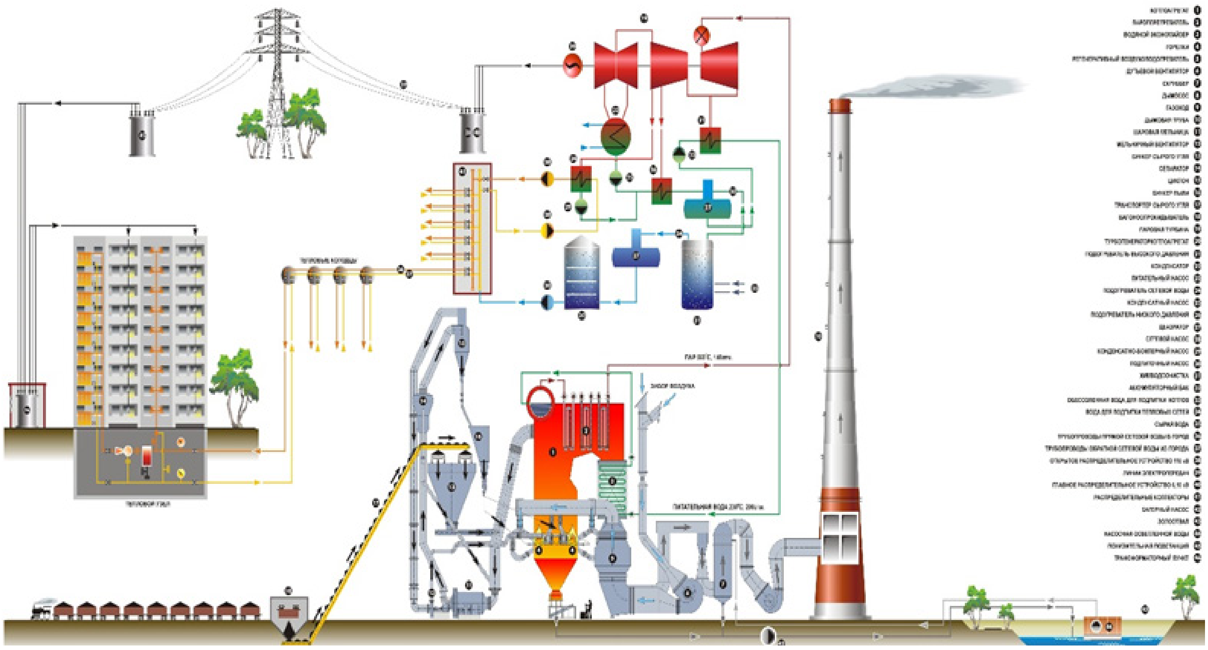


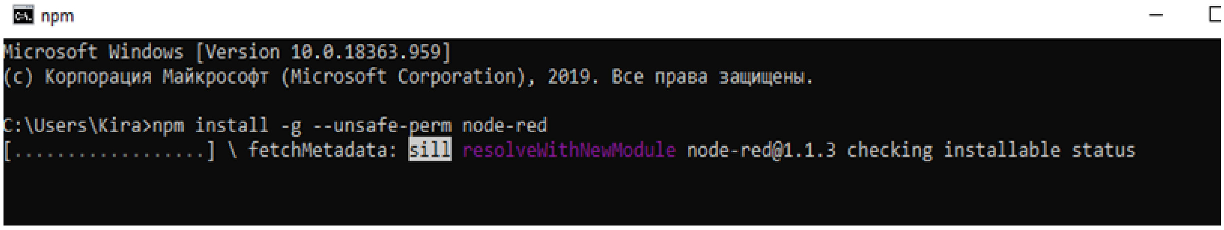
Figure 1 – Operation cycle of Ust-Kamenogorsk TPP

Simplistically, the operation of a combined heat and power plant can be described as follows – the cycle begins with the operation of a pumping station that pumps water from the Ulba River and also takes water from the return flow of hot water, then the process of distillation of the resulting water takes place using chemical water treatment, the resulting distilled water enters the boiler, where the water is heated with the help of coal obtained by conveyors, then the water is heated and converted into steam, the high-pressure steam is conveyed through the pipeline to the turbine, where, using steam pressure, turbines rotate, the turbines, in turn, rotate generators that convert mechanical energy into electricity, after which the generated electricity is passed through transformers, a stable alternating voltage is created and transmitted to the consumer [15]. The resulting steam passes further through the pipes and is transferred to the condensate, where the steam is converted into water and re-filtered, then the resulting water is mixed with that which has not evaporated in the boiler and enters the heat units for the final preparation of the coolant for supply to the consumer. Further, by means of pipes and thermal wells, hot water is transferred to the consumer. Unused hot water, having passed through the consumer, returns to the TPP, and repeats the entire chain again.

Within the framework of this work study, only the aspect of thermal generation of a combined heat and power plant will be used.

Results and Discussion

The main requirement for the functioning of the web interface is the presence of its software platform, namely, the key element of the application is Node.js [16]. This platform is installed using the usual installer of the latest stable version of the program. It is also necessary to install all the required software packages and libraries. The next condition is to install the main extension, which acts as the web interface itself, namely node-red. This extension will act as the main editor of various chains of nodes and program code, will act as a link for all elements. This extension is installed using the package installation manager – NPM, using the appropriate command (Figure 2, p. 69).

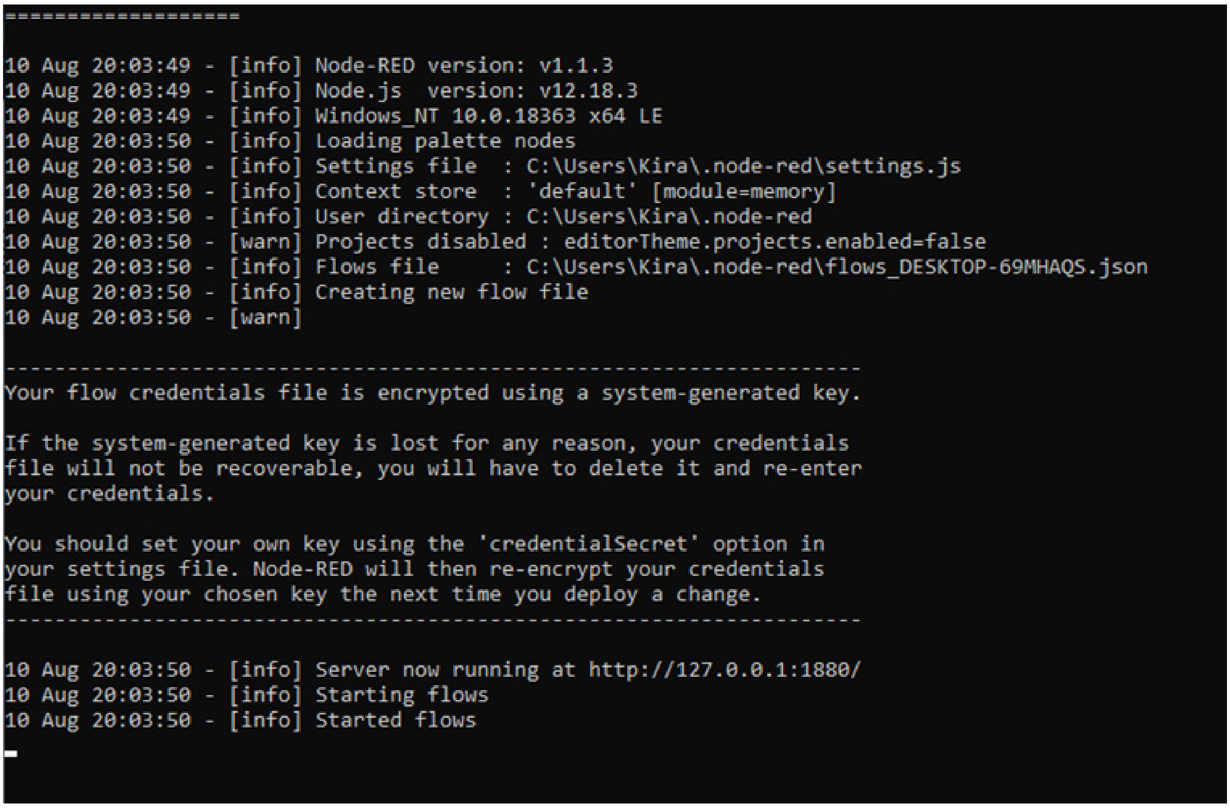


```
npm
Microsoft Windows [Version 10.0.18363.959]
(c) Корпорация Майкрософт (Microsoft Corporation), 2019. Все права защищены.

C:\Users\Kira>npm install -g --unsafe-perm node-red
[..................] \ fetchMetadata: sill resolveWithNewModule node-red@1.1.3 checking installable status
```

Figure 2 – Launching the node red package installation manager using the command line

To start working with the web interface, you need to run the executable module on the command line using the appropriate command (Figure 3).



```
===================

10 Aug 20:03:49 - [info] Node-RED version: v1.1.3
10 Aug 20:03:49 - [info] Node.js  version: v12.18.3
10 Aug 20:03:49 - [info] Windows_NT 10.0.18363 x64 LE
10 Aug 20:03:50 - [info] Loading palette nodes
10 Aug 20:03:50 - [info] Settings file  : C:\Users\Kira\.node-red\settings.js
10 Aug 20:03:50 - [info] Context store  : 'default' [module=memory]
10 Aug 20:03:50 - [info] User directory : C:\Users\Kira\.node-red
10 Aug 20:03:50 - [warn] Projects disabled : editorTheme.projects.enabled=false
10 Aug 20:03:50 - [info] Flows file     : C:\Users\Kira\.node-red\flows_DESKTOP-69MHAQS.json
10 Aug 20:03:50 - [info] Creating new flow file
10 Aug 20:03:50 - [warn]

---------------------------------------------------------------------
Your flow credentials file is encrypted using a system-generated key.

If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.

You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
---------------------------------------------------------------------

10 Aug 20:03:50 - [info] Server now running at http://127.0.0.1:1880/
10 Aug 20:03:50 - [info] Starting flows
10 Aug 20:03:50 - [info] Started flows
```

Figure 3 – Running node-red using the command line

The editing menu of this web interface (Figure 4, p. 70) is launched on the local host using port 1880 (http://127.0.0.1:1880). You can open this menu in any browser, but it is preferable to use the Google Chrome browser, as some functions may not be supported in other browsers, or the displayed information may not be correct. As mentioned earlier, using this interface, you can create a sequence of nodes that will execute the program code sequentially, according to the location of the nodes in the chain. For each block of this chain (node), you can prescribe certain parameters, set your own program code, customize the appearance of the displayed element, and specify the behavior algorithm. Program code is written primarily in the java script programming language.

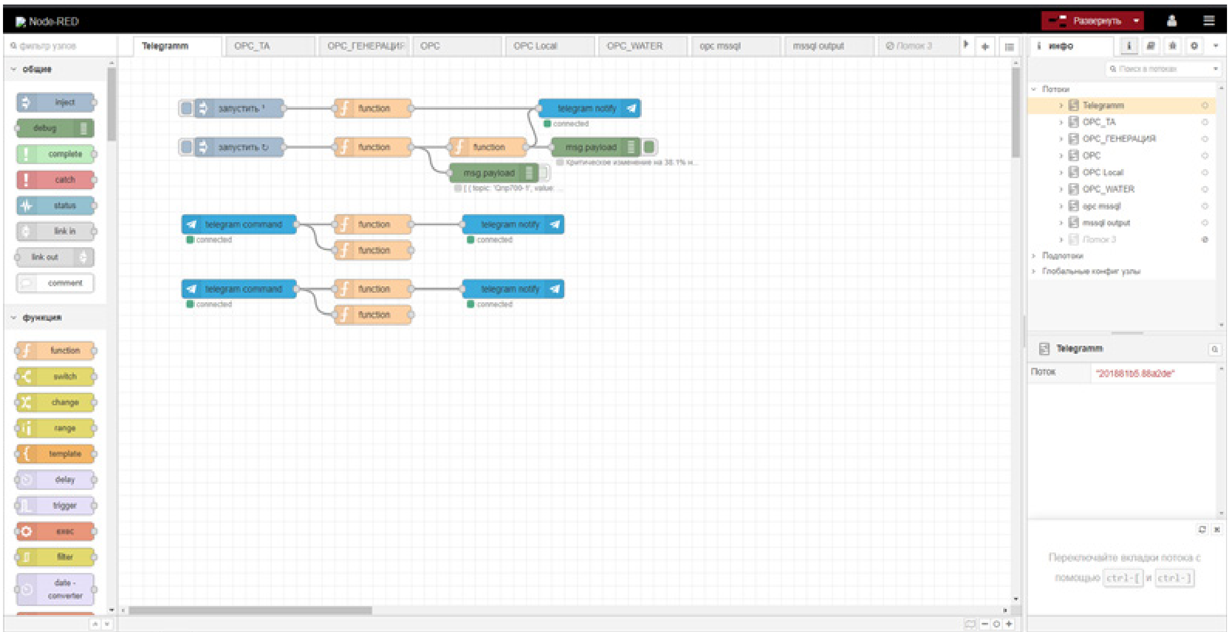


Figure 4 – Node-red interface window

For all aspects of the program to function correctly, it is necessary to install all the main node libraries (Figure 5).

The chains of program nodes are built from the nodes listed by the libraries. Nodes can act as various structural blocks responsible for various tasks: receiving and outputting information, outputting data of various formats to a web page, executing special program code, processing data, interacting with programs and servers using various data transfer protocols, interacting with databases, etc.

The web interface is a tool for providing a user interface, by means of dynamic web pages, for the purpose of managing the program, by means of the http protocol [17]. In this case, node-red is selected as the web interface (Figure 6). Node-red is a web interface that acts as a visual programming tool that uses various libraries and extensions, of which there are a huge number of variations, which greatly simplifies the development of various programs [18]. This web interface was chosen because it does not require much knowledge in programming and any employee can easily interact with this interface. Working with the interface, as the name says, is carried out by means of working with a dynamic web page that acts as a web interface.

The Node-red web interface consists of several parts. On the left side is a list of all installed nodes that can be used as modules to create node chains. In the center is located directly the workspace for the location of these very nodes, called the flow. For the convenience of debugging and load distribution, several threads should be used, so that in the event of a malfunction, a qualified employee can turn off a certain thread so that the program can continue its work, and the employee can fix the malfunction without harm to users.

Customizable web interface tabs:

1. The “Info” tab is used to display detailed information about streams and the global node configuration.

2. The “Help” tab displays brief information about the selected node.

3. The “Debug” tab is used for debugging and troubleshooting.

4. The “Configuration” tab is used to display information about the elements used (the number of elements used on streams).

“MSSQL”

• node used to connect to the database, create and receive SQL queries using the TDS protocol.

“Dashboard”

• a dashboard used to display information in the form of various graphs, instrument indicators, tables and diagrams. Also, with the help of this node, tools for interaction are used: buttons, sliders, input fields, etc.

“Bool-gate”

• a node that is a switch using boolean values.

“Date-converter”

• a node used to convert time from various formats.

“Modbus”

• a node used to connect to a server built on OPC technology that transmits and receives information via the modbus protocol.

“OPC-DA”

• node used to connect and interact with servers that work according to the OPC-DA standard. This standard is used for real-time data exchange.

“Time”

• the node used to get the current date and time.

“Telegrambot”

• host used to manage the telegram bot.

“Random”

• node used to generate random numbers.

“PythonShell”

• node used to connect third-party libraries built in the python programming language.

Figure 5 – Main node libraries

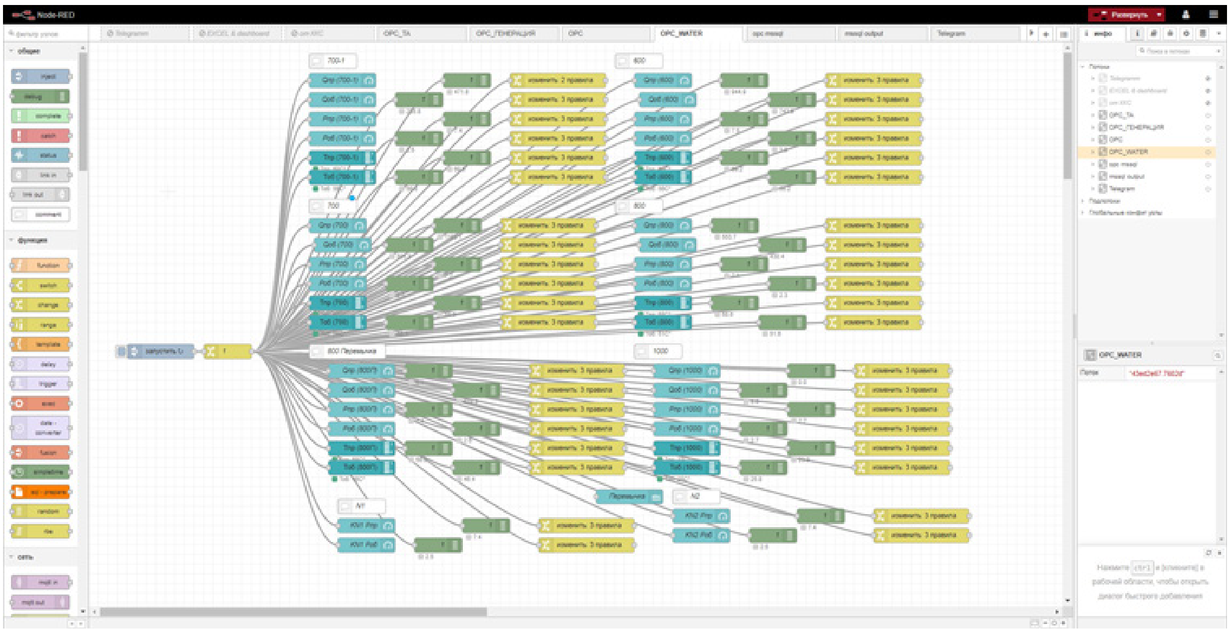


Figure 6 – Node-red web interface window

The main node used to display information from the sensors and devices of the UK TPP is the Dashboard node. This node is a graphical panel consisting of several elements that can be used to display sensor data in the form of various diagrams, graphs, tables, etc. The main elements of this node are: “Text” (output of static and dynamic text), “Graph” (output of graphs), “Audio” (audio output), “Dashboard” (output of sensor readings in the form of a dashboard), “Alert” (display an alert), “Number” (select numerical values using the interface arrows), “Dropdown Menu” (select from multiple pop-up panels, when clicked), “Select Color” (select a color from the color palette), “Switch ” (boolean switch), “Slider” (horizontal slider with which you can adjust the transmitted value), “Form” (fillable form), “Text” (text input), “Date” (date selection).

This dashboard will be displayed on the local host by the link http://localhost:1880/ui . It is this page that all users will use to display information and interact with the interface, except that to connect, users will need to specify the IP address of the server instead of the “localhost” value. You can automate this process using the local DNS of the current server by creating a record of the ratio of the IP address of the dashboard [19] and, for example, the name of the page “UkTPPDashboard”. Thus, when entering this combination of words in the browser, the server will find the ratio of the record and open the dashboard page.

To display indicators, it is necessary to create sections with information output and display them according to special templates. The templates should also be optimized for mobile devices. The main sections are:

1. “Generation” – This section is used to display basic data on heat generation at the UK-TPP, namely indicators of total generation, loss and net return to consumers. Basic graphs and data from the database.
2. “Ta” The main indications of all heat units and a general graph of all indicators.
3. “Water” – Readings of all pipes and jumpers, such as: flow rate, pressure, temperature, etc.
4. “Water (graphs)” – Displaying pipe indicators on general graphs.
5. “Table” – Output of any indicators from the database for a certain period using the selection drop-down menu.
6. “Graph” – Displays graphs for any indicators for a certain period using the drop-down menu of choice.

Main modules of the web application

a) Module “Ta”

This section is used to display the performance of all heat units. The heat unit is a key link in the heat supply system, designed to prepare the heat carrier in accordance with consumer demand. This heating system is centralized and has a high efficiency. The heat carrier in this case is hot water, with the necessary indication of temperature and pressure. If the values of the unit heat indicators are equal to 0, this means that this heat unit is turned off due to repair or another reason. General readings of all heat units go to the “Generation” section.

Nodes used in this chain:

1. Injector – a node used to launch a chain of actions, it initializes the launch of the chain with the required interval and action parameters.
2. Change – a node used to write data to local and global variables for further use in other chains of nodes.
3. Function – a node used to execute program code written in the java script programming language.
4. Dashboard – elements of this node are used to visualize information in the form of various graphs, tables, dashboards, text, etc. To use them, certain parameters of input data and displayed elements are filled. The dashboard itself with graphs and indicators “Dashboard” will be a dynamic web page that opens at the link http://127.0.0.1:1880/ui.
5. Debug – this node is used to debug the input data and indicate the operation of the node.
6. Delay – a node used to create a delay before transmitting data to the next node.

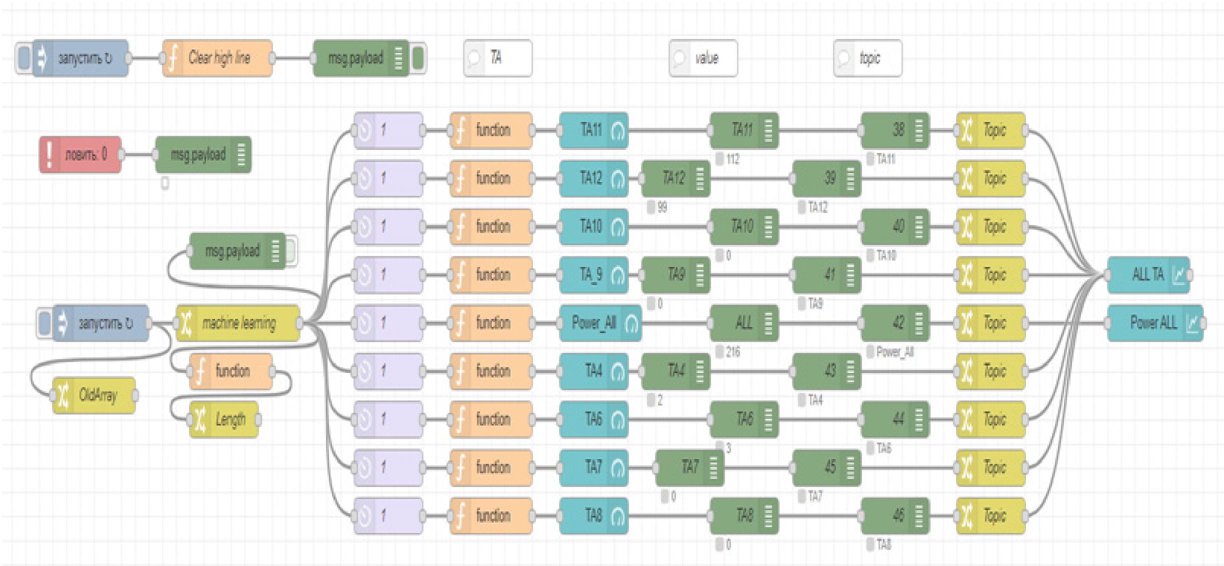


Figure 7 – Scheme of the “TA” module

For data visualization nodes to work, they need to set: a navigation bar tab and a group (where the element will be displayed on the dashboard), set the size corresponding to the size of the average screen of a mobile device, the input data format (values that will be transferred from the local array), set units of measurement, set the color of the dashboard to change when it goes beyond the norm (Figures 8).

In addition, this chain of nodes uses the delay node, this node sets the delay before executing the next node. The node was added in order to sequentially update the dashboard data, to optimize the updated data, since with a one-time update of all dashboard sensors, on weak devices, the load increases at that moment, as a result of which the browser freezes slightly, sequential data updating solves this problem.

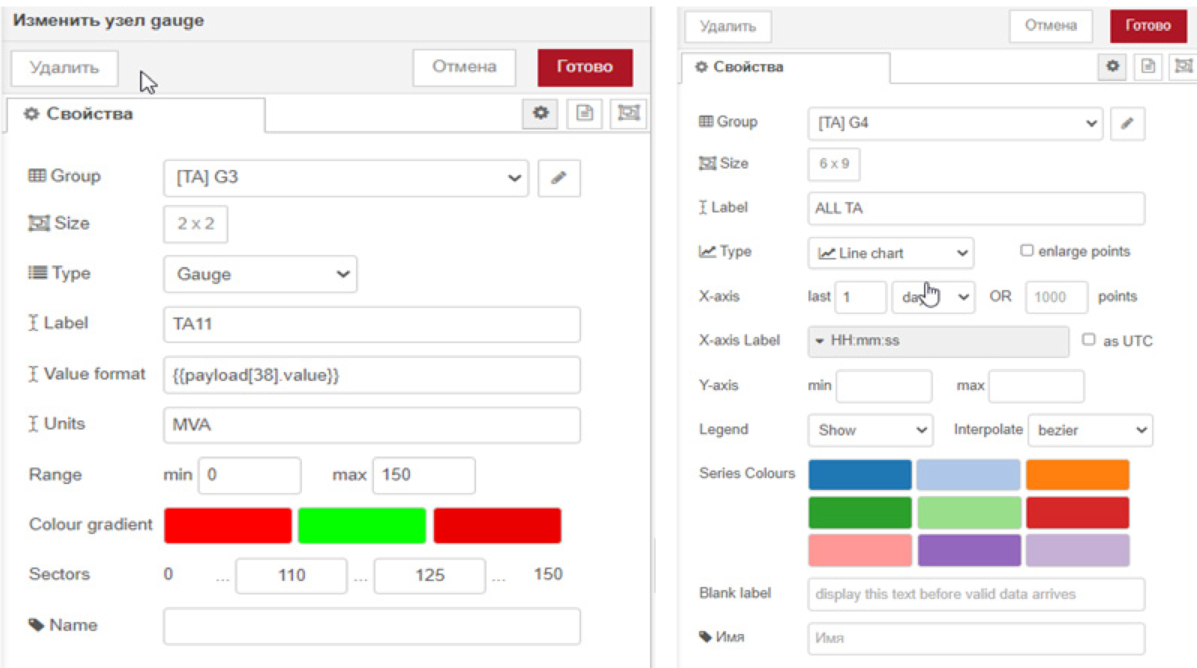


Figure 8 – Dashboard Gauge Setting Window and Graph Setting Window

The display of the section on the computer and mobile devices can be seen in Figure 9 and Figure 10.

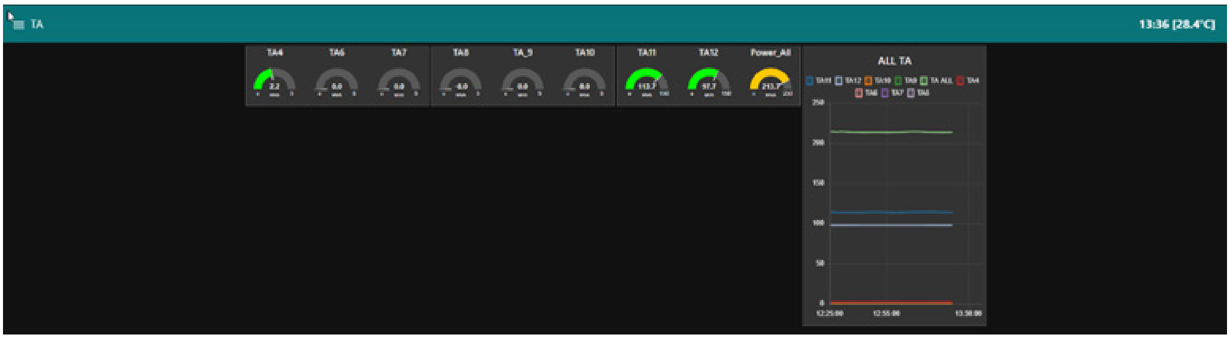


Figure 9 – Display window of the “TA” module on the computer

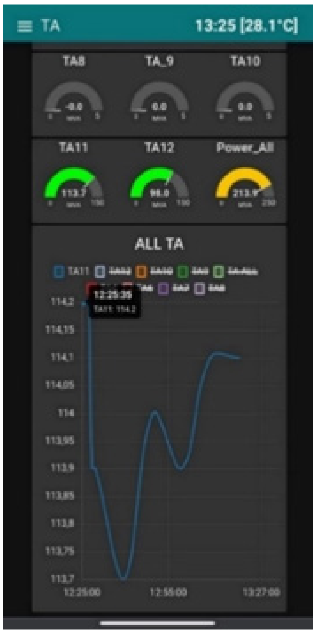


Figure 10 – Display window of the “TA” module on the smartphone

b) “Generation” module

This section (Figure 11) is used to display the main statistics for warm water. The main calculation indicators are derived: losses, given to consumers and generation. These readings are also displayed in a schematic circular graph.

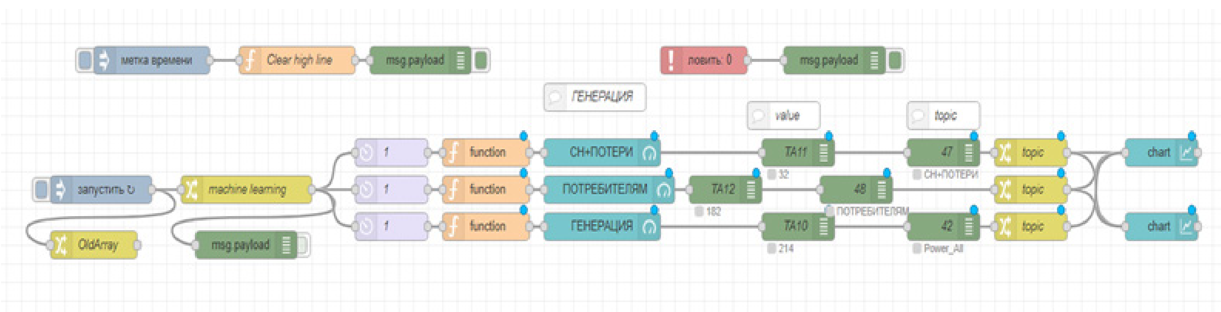


Figure 11 – Scheme of the “Generation” module

The display of the “Generation” section on a computer and mobile devices can be seen in Figures 12 and 13.

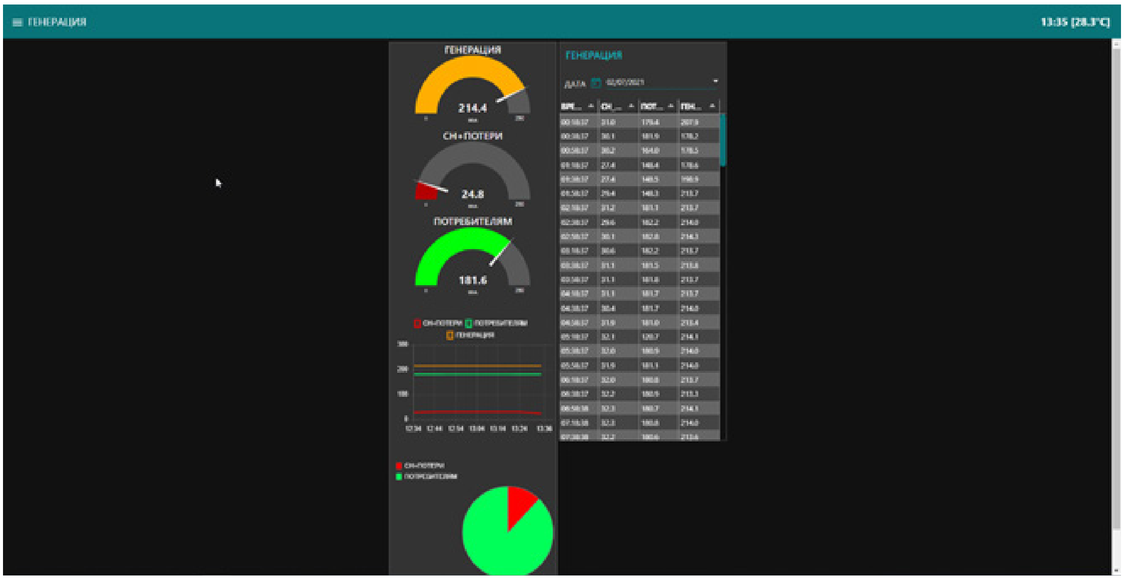


Figure 12 – Display window of the “Generation” module on the computer

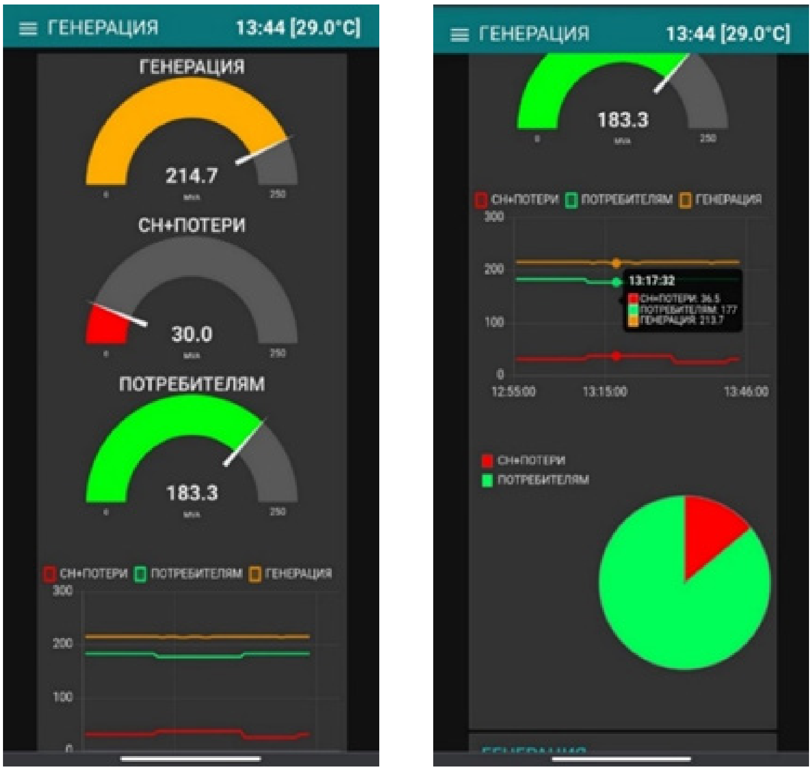


Figure 13 – Display window of the “Generation” module on mobile devices

c) “Water” module

This section (Figures 14 and 15, p. 76) displays the indicators of the main pipes responsible for supplying hot water to the consumer, various jumpers, and collectors. The section displays pipe indicators, namely: temperature, density, and water flow rate. A feature of this section is that for data visualization, in addition to the dashboard node, the “Charts” node is used, which is responsible for temperature visualization.

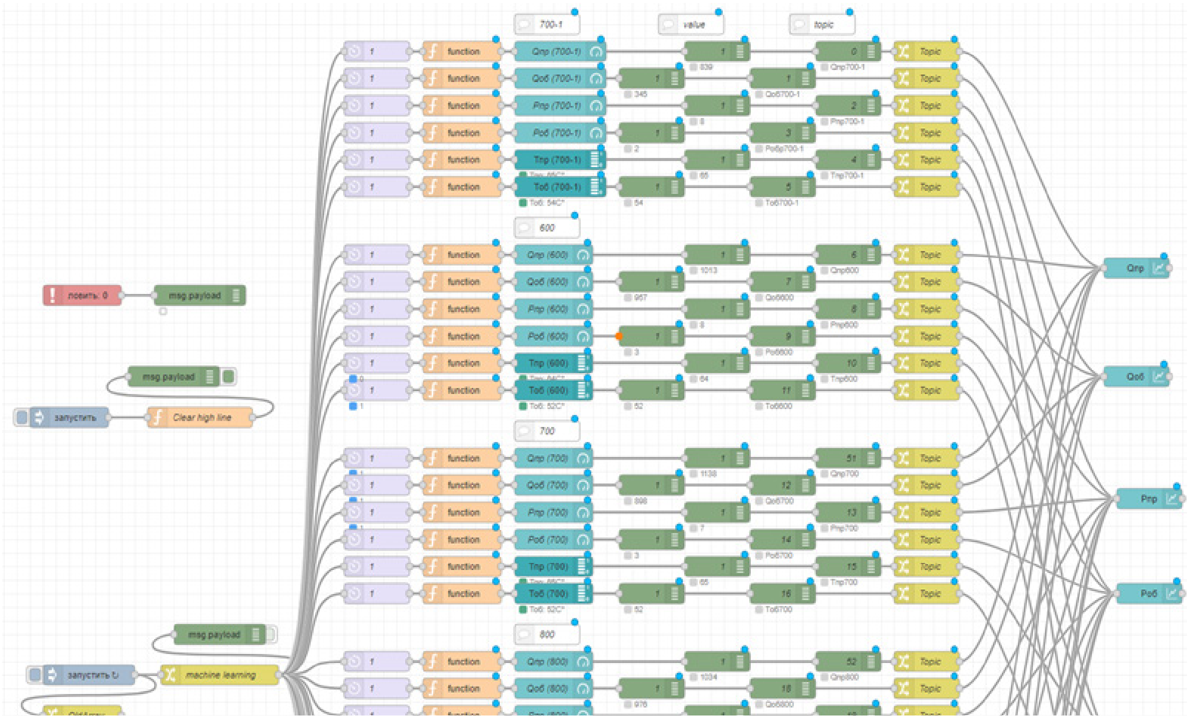


Figure 14 – Scheme of the “Water” module, part 1

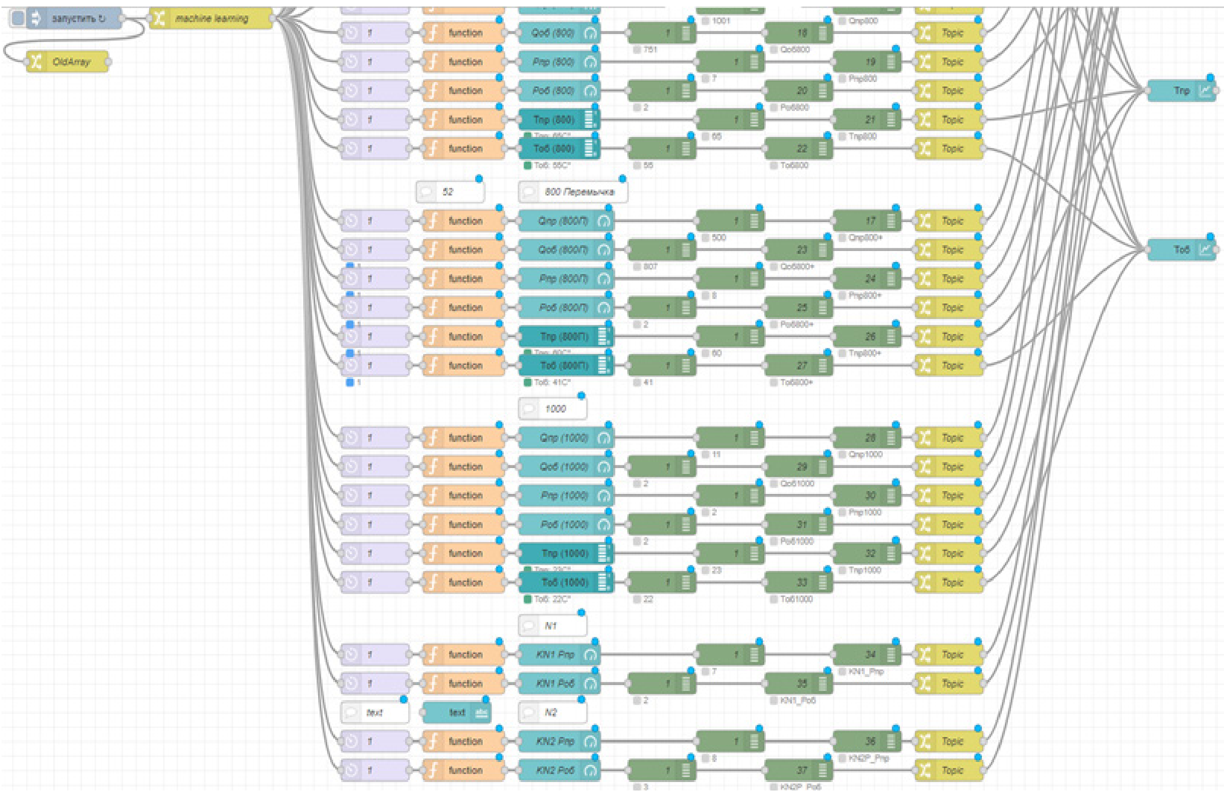


Figure 15 – Scheme of the “Water” module, part 2

The chain of this node begins by running the loop along with the program and then polling the chain once every ten seconds. Next comes the process of obtaining an array of machine learning data and writing to a local one, to further process the data. With the help of functions, the node analyzes the received data and compares them with the corresponding sensors, creates local arrays to configure each sensor and graph. The resulting data is passed to the dashboard and ladder nodes to visualize the corresponding values. Values such as: the name of the current sensor, its value, critical data ranges, maximum sensor ranges, whether the value is critical, etc. After visualizing the data, debug nodes follow, displaying the correctness of the transmitted data and signaling the operation of the node. Further, if necessary, a more precise name is specified for the sensor, to correctly display it on the chart. The last element in the chain is the dashboard node responsible for plotting the graph.

For the data visualization nodes to work, they need to set: a navigation panel tab and a group, set the size corresponding to the size of the average screen of a mobile device, the input data format, set the units of measurement, set the dashboard color to change when it goes beyond the norm. In addition, this chain of nodes uses the delay node, the node sets the delay before executing the next node. This node was added to sequentially update the dashboard data, to optimize the updated data, since with a one-time update of all dashboard sensors, the load on weak devices increases at this moment, because of which the browser freezes slightly, sequential data updating solves this problem.

The readings of one pipe are a block [20] consisting of the name of this pipe and its indicators (temperature, pressure, flow rate), this block can be seen in Figure 16.

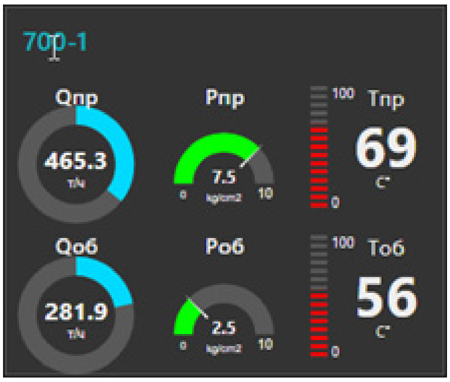


Figure 16 – Example of a block of one pipe in the dashboard

The display of the section on a computer and mobile devices is shown in Figures 17 and 18 (p. 78).

Figure 17 – The window for displaying indicators on computers.

Figure 18 – The window for displaying indicators on mobile devices

d) Web application navigation bar

"Header" – is the top strip of the dashboard, in which you can display any dynamic and static information. The dashboard header contains the name of the selected section, time, and outdoor temperature. The operation of this section is based on the chain of nodes that form the circuit shown in Figure 19.

This scheme consists of:

1. An injector that is set to launch along with node-red with a certain frequency of polling a chain of nodes.
2. A time converter that converts the time to a standard format and transfers it to the original array.
3. The change node, which receives the air temperature from the sensor from the global OPC-DA array.
4. A function node that rounds the data and writes it in t°C format.
5. The join node, which combines time and temperature into a single array to output data through the template node.
6. The template node, with a customized html template specifically for the header, to correctly display the text.
7. Debug – this node is used for debugging input data and indicating node operation.

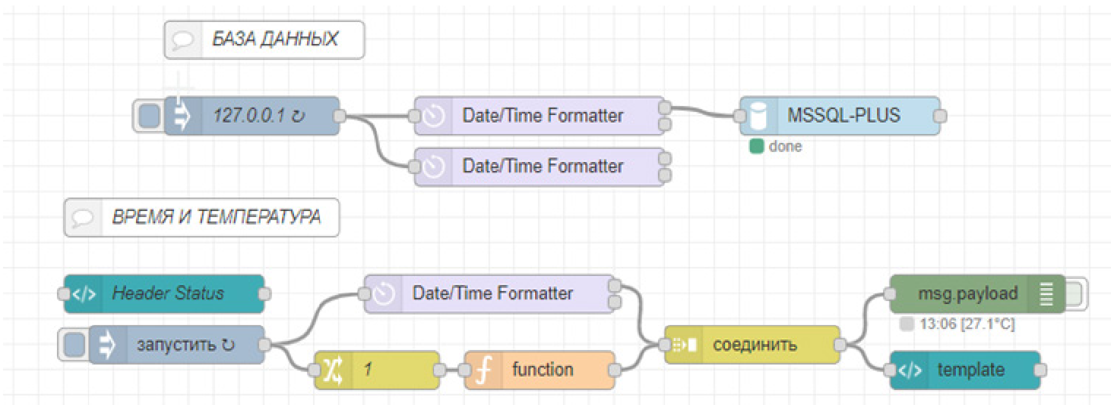


Figure 19 – Time and temperature display diagram

The chain of this node begins by running the loop along with the program and then polling the chain once every ten seconds. Next comes the process of obtaining an array of machine learning data and writing this array to a local one, to further process the data. There is also a conversion of time from UNIX format to standard [21], for the correct display of data. Further, the initial data is combined into a single data array and transferred to the node for data substitution in the html template to display data in the header (Figure 20) of the dashboard.

Template

```
<style>
    #headerStatus {
        font-weight: bold;
        text-transform: uppercase;
    }
    #headerStatus.online {
        color: lime;
    }
    #headerStatus.offline {
        color: tomato;
    }
</style>

<script id="titleScript" type="text/javascript">
$(function() {
    if($('.md-toolbar-tools').length){
        initHeader();
    } else setTimeout(initHeader, 500)
});

function initHeader(){
    if (!$('#headerStatus').length) {
        var toolbar = $('.md-toolbar-tools');
        var div = $('<div/>');
        var p = $('<p id="headerStatus" />');
        div[0].style.margin = '5px 5px 5px auto';
        div.append(p);
        toolbar.append(div);
    }
}
</script>
```

Figure 20 – Html “template” containing java script code

Sections of the navigation bar are created using, setting in the settings of group nodes and sections of the navigation bar.

Conclusion

The created application is an effective tool for solving the tasks set by the UK TPP. All requirements were met, threats to information security were foreseen, issues of optimization, machine learning and data analysis were resolved. All the main aspects of the functioning of this application were considered and analyzed. This work is an effective tool for optimizing the work of various departments and workshops of the UK TPP. One of the main tasks was to prevent critical situations related to the equipment of the workshops, analyze the behavior of the equipment, and increase the efficiency of the devices, thanks to the timely replacement of the necessary equipment. If the indicators of the devices go beyond their norm, the relevant employees will be immediately informed, in addition, thanks to the database used. The database is systematically backed up, which in the event of a hardware breakdown or other malfunction will save data from loss. Also, the application optimizes the visualization of key indicators for employees of various workshops to increase labor productivity. The software used is clear and intuitive for all employees, so there should be no problems using the dashboard and interface.

REFERENCES

1 Pérez J. B., Falótico A. J. A. (2019) Various perspectives of la bor and human resources challenges and changes due to automation and artificial intelligence, Academicus, no.20, p. 106.

2 Koopman P., Wagner M. (2017) Autonomous vehicle safety: An interdisciplinary challenge, IEEE Intelligent Transportation Systems Magazine, vol. 9, no.1, pp. 90–96.

3 Sadoun A.M. et al. (2022) Utilization of improved machine learning method based on artificial hummingbird algorithm to predict the tribological behavior of Cu-Al2O3 nanocomposites synthesized by in situ method, Mathematics, vol. 10, no. 8, p. 1266.

4 Hao L. et al. (2022) When is silence golden? A meta-analysis on antecedents and outcomes of employee silence, Journal of Business and Psychology, vol. 37, no. 5, pp. 1039–1063.

5 Oviatt S., Cohen P.R. (2022.)The paradigm shift to multimodality in contemporary computer interfaces, Springer Nature.

6 Zhamalkanov T.A. (2023) Razvitiye sistemy tsentralizovannogo teplosnabzheniya goroda Ust'-Kamenogorsk, Sostav redaktsionnoy kollegii i organizatsionnogo komiteta [in Russian].

7 Tilkov S., Vinoski S. (2010) Node. js: Using JavaScript to build high-performance network programs, IEEE Internet Computing, vol. 14, no. 6, pp. 80–83.

8 Josuttis N.M. (2012) The C++ standard library: a tutorial and reference.

9 Ferencz K., Domokos J. (2019) Using Node-RED platform in an industrial environment, XXXV. Jubileumi Kandó Konferencia, Budapest, pp. 52–63.

10 Malik R.S., Patra J., Pradel M. (2019) NL2Type: inferring JavaScript function types from natural language information, 2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE), IEEE, pp. 304–315.

11 Stančin I., Jović A. (2019)An overview and comparison of free Python libraries for data mining and big data analysis, 2019 42nd International convention on information and communication technology, electronics, and microelectronics (MIPRO), IEEE, pp. 977–982.

12 Turarov ZH. T., Sedelev V.A. (2019) Razrabotka nauchnykh osnov sberezheniya energeticheskikh resursov TOO "Ust'-Kamenogorskaya TETS", Vestnik Vostochno-Kazakhstanskogo gosudarstvennogo tekhnicheskogo universiteta im. D. Serikbayeva, no. 1, pp. 204–207.

13 Shair J. et al. (2021) Power system stability issues, classifications, and research prospects in the context of high penetration of renewables and power electronics, Renewable and Sustainable Energy Reviews, vol. 145, p. 111111.

14 Bayazitova Z.E. et al. (2022) Relevance of Environmental Surveys on the Design of a New Municipal Waste Management System on the City of Kokshetau (Kazakhstan), Sustainability, vol. 14, no. 21, p. 14368.

15 Dudjak V. et al. (2021) Impact of local energy markets integration in power systems layer: A comprehensive review, Applied Energy, vol. 301, p. 117434.

16 Cantelon M. et al. (2014) Node. js in Action, Greenwich, Manning, 2014, pp. 17–20.

17 King G., Wang H. (2023) Httpa: Https attestable protocol, Future of Information and Communication Conference. – Cham : Springer Nature Switzerland, pp. 811–823.

18 Larrinaga F. et al. (2022) Node-red workflow manager for edge service orchestration, NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium, IEEE, pp. 1–6.

19 Lencse G. (2020) Benchmarking authoritative DNS servers, IEEE Access., vol. 8, pp. 130224–130238.

20 Rennels D.C. (2022) Pipe flow: A practical and comprehensive guide, John Wiley & Sons.

21 Ardiansah I. et al. (2021) Design of micro-climate data monitoring system for tropical greenhouse based on arduino UNO and raspberry pi, IOP Conference Series: Earth and Environmental Science, IOP Publishing.

[1]Увалиева И.,
PhD, ORCID: 0000-0002-2117-5390, e-mail: iuvalieva@ektu.kz
[2]Рустамов С.,
PhD, қауымдастырылған профессор, ORCID: 0000-0002-3247-5882,
e-mail: Srustamov@ada.edu.az
[3]*Бельгинова С.,
PhD, ORCID: 0000-0002-7238-6016, e-mail: sbelginova@gmail.com
[1]Рахметуллина Ж.,
ф.-м.ғ.к., ORCID: 0000-0002-0554-7684, e-mail: zhrakhmetullina@ektu.kz

[1]Д. Серікбаев атындағы Шығыс Қазақстан техникалық университеті,
070004, Өскемен қ., Қазақстан
[2]ADA университеті, Баку қ., Әзірбайжан
[3]«Тұран» университеті, 050013, Алматы қ., Қазақстан

ТЕХНОЛОГИЯЛЫҚ ҮРДІСТЕРДІҢ ҚАУІПСІЗДІГІН ҚАМТАМАСЫЗ ЕТУ ҮШІН ДЕРЕКТЕРДІ ТАЛДАУ МЕН ВИЗУАЛИЗАЦИЯЛАУҒА АРНАЛҒАН WEB-ҚОЛДАНБА ӘЗІРЛЕУ

Аңдатпа

Соңғы уақытта заманауи технологиялар қарқынды түрде дамуда және қарапайым деректерді жинап, өңдеп қана қоймай, қазіргі алгоритмдер жабдықтың әрекетін болжауға, түрлі ақаулардың алдын алуға, апаттың ықтималдығын азайту үшін шаралар қабылдауға, тиісті қызметкерлерге ескерту сияқты мүмкіндіктерге ие болды. Осы зерттеу аясында авторлар технологиялық үрдістердің қауіпсіздігін қамтамасыз ету мақсатында Өскемен жылу электр орталығының деректерін талдау және визуализациялау үшін веб-қосымшаны әзірлеуді ұсынады. Өскемен ЖЭО облыстық жылу электр орталығы – қаланың тұрғын үй-коммуналдық секторы мен өнеркәсіптік кәсіпорындар жүктемесінің 80%-ын қамтамасыз ететін жылу және электр энергиясын өндіруші. ЖЭО – Қазақстанның біртұтас энергетикалық жүйесінің бөлігі. Әзірленіп жатқан қосымшада Өскемен ЖЭО негізгі сенсорлары мен аспаптарынан алынған деректерді көрсететін және визуализациялайтын, сондай-ақ жабдықтың ақауларын болдырмайтын, машиналық оқытуды қолдана отырып, сенсорлардың әрекетін талдайтын веб-интерфейс қарастырылады. Пайдаланылатын негізгі бағдарламалық платформа ретінде Node.js таңдалды, java скрипттік бағдарламалау тілінде құрастырылып, бағдарламалау тілінің мүмкіндіктерін кеңейтетін с ++ тілінде жазылған, сондай-ақ деректер ауқымымен жұмыс істеуді, оларды қабылдауды және жіберуді жылдамдататын қосымша модульдер пайдаланылды.

Тірек сөздер: деректерді талдау, визуализациялау, технологиялық процестердің қауіпсіздігі, Өскемен-ЖЭО, құрылғы, Web-қосымша, мониторинг, node.js, java script.

[1]**Увалиева И.,**
PhD, ORCID: 0000-0002-2117-5390, e-mail: iuvalieva@ektu.kz
[2]**Рустамов С.,**
PhD, ассоциированный профессор, ORCID: 0000-0002-3247-5882,
e-mail: Srustamov@ada.edu.az
[3]***Бельгинова С.,**
PhD, ORCID: 0000-0002-7238-6016, e-mail: sbelginova@gmail.com
[1]**Рахметуллина Ж.,**
канд.физ.-мат.наук, ORCID: 0000-0002-0554-7684, e-mail: zhrakhmetullina@ektu.kz

[1]Восточно-Казахстанский технический университет им. Д. Серикбаева,
070004, г. Усть-Каменогорск, Казахстан,
[2]Университет ADA, г. Баку, Азербайджан
[3]Университет «Туран», 050013, г. Алматы, Казахстан

РАЗРАБОТКА WEB-ПРИЛОЖЕНИЯ АНАЛИЗА И ВИЗУАЛИЗАЦИИ ДАННЫХ ДЛЯ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ

Аннотация

Современные технологии довольно хорошо развились за последнее время, и кроме банального сбора и обработки данных современные алгоритмы способны предугадывать поведение оборудования и предупреждать возникновение неполадок и выход из строя заблаговременно и предпринимать действия по снижению вероятности возникновения аварии и предупреждению соответствующих сотрудников. В рамках данного исследования авторами предложена разработка web-приложения анализа и визуализации данных УК-ТЭЦ для обеспечения безопасности технологических процессов. Усть-Каменогорская ТЭЦ является тепловой электростанцией регионального назначения, выработка тепла и электричества покрывает 80% нагрузки жилищно-коммунального сектора города и промышленных предприятий. ТЭЦ входит в единую энергосистему Казахстана. Разрабатываемое приложение имеет веб-интерфейс, который будет отображать и визуализировать данные основных датчиков и приборов УК ТЭЦ, а также предупреждать возникновение неисправности оборудования и анализировать поведение датчиков с помощью машинного обучения. В качестве основной используемой программной платформы была выбрана node.js, построенная на языке программирования java script, с использованием дополнительных модулей, расширяющих возможности данного языка программирования, написанных на c++, что ускоряет работу с массивами данных, их получение и отправку.

Ключевые слова: анализ данных, визуализация, безопасность технологических процессов, УК-ТЭЦ, прибор, Web-приложение, мониторинг, node.js, java script.