

Finding Anomalies in the Operation of Automated Control Systems Using Machine Learning

Yurii Hodlevskiy^a, Tetiana Vakaliuk^{b, c, d}, Oleksii Chyzhmotria^b, Olena Chyzhmotria^b and Oleh Vlasenko^b

^a Infopulse Ukraine, 13a, Trypilska Street, Zhytomyr, 10003, Ukraine

^b Zhytomyr Polytechnic State University, 103 Chudnivsyka Str., Zhytomyr, 10005, Ukraine

^c Institute for Digitalisation of Education of the NAES of Ukraine, 9 M. Berlynskoho Str., Kyiv, 04060, Ukraine

^d Kryvyi Rih State Pedagogical University, 54 Gagarin Ave., Kryvyi Rih, 50086, Ukraine

Abstract

This article deals with the problem of detecting anomalies in the operation of automated systems. Anomaly detection is useful for preventing breakdowns and improving the performance of automated systems. Using various sensors in automated systems, it is possible to read information about the state of certain system parameters, which in turn helps to monitor the state of the system at the moment. But simply viewing system indicators is not a very optimal option, as human resources are wasted. It is also possible to set the limit values of the sensors and if some indicator goes beyond them, the system can show a message about it. Still, not all graphs can do with this solution to the problem, because abnormal values can be recorded in these limits and ignored by such a system, or the sensor can change the range of work and in this case, thus operators will receive a large number of false messages. In this way, it is possible to implement a system that will detect anomalies automatically using artificial intelligence, which will learn from existing previous data and notify the operator of a malfunction.

Keywords 1

Machine Learning, Gradient Descent, Learning Algorithm, Adaptive Movement Estimation, Long Short-term Memory, Diapason.

1. Introduction

The problem of detecting anomalies in complex automated control systems is widespread enough in various spheres of human activity. These problems can be tried to solve in different ways, regardless of the budget and the ability to maintain various systems.

Let's take as an example electricity generator stations in hard-to-reach places on our planet. The work control system is complicated, for example, due to weather conditions and keeping personnel in hard-to-reach places. Including the complexity of controlling data coming back from different sensors that return different values in different components. You can try to troubleshoot this problem in several ways.

The most obvious of these is 24/7 monitoring of high performance using staff. But solving the problem in this way increases the number of personnel, complicates the work schedule, and creates a human factor risk.

Another solution is a fixed minimum and maximum value for the sensors. But the problems of this approach are the difficulty of setting the values for each of the sensors separately and updating

IntelITSIS'2023: 4th International Workshop on Intelligent Information Technologies and Systems of Information Security, March 22–24, 2023, Khmelnytskyi, Ukraine

EMAIL: godlevskiy.yuriy@gmail.com (Yurii Hodlevskiy); tetianavakaliuk@gmail.com (Tetiana Vakaliuk); chov@ztu.edu.ua (Oleksii Chyzhmotria); ch-o-g@ztu.edu.ua (Olena Chyzhmotria); oleg@ztu.edu.ua (Oleh Vlasenko)

ORCID: 0000-0003-4094-0788 (Yurii Hodlevskiy); 0000-0001-6825-4697 (Tetiana Vakaliuk); 0000-0002-5515-6550 (Oleksii Chyzhmotria); 0000-0001-8597-1292 (Olena Chyzhmotria); 0000-0001-6697-2150 (Oleh Vlasenko)



© 2023 Copyright for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

the values for each separately in case of changing the operation diapason of the sensors, which slows down and complicates the operation of such systems.

One of the options is to create an application for finding anomalies in the operation of the automated control system using machine learning. Thanks to machine learning, the system will automatically adapt to the variety of diapasons and values being looked at.

In this application, it is planned to use data analysis methods in the LSTM neural network architecture to implement the search for anomalous values in the operation of the automated system, which return the values of different diapasons with different periodicity. The peculiarity of this search for anomalies is the adaptation of programs to different diapasons, and the search for anomalies in various zones, which expands the capabilities of the program supplement for various areas.

Objective – development of a system that can be used to control anomalies in the operation of automated control systems.

2. Related Works

Benjamin Lindemann, Benjamin Maschler, Nada Sahlab, Michael Weyrich in their research describe an overview of promising LSTM based approaches for anomaly detection with an additional focus on upcoming graph-based and transfer learning approaches. All approaches are evaluated based on a set of application-oriented criteria such as the detection capabilities regarding temporal anomalies, achieved accuracies, and use cases addressed in the original publication. They present some use cases which can be useful in different areas but not examples of applications that can solve the real problem of detecting anomalies on different types of power stations [13].

Gian Antonio Susto, Matteo Terzi, Alessandro Beghi describe an application which was checked anomaly detection strategies that have been tested on a real industrial dataset related to a Semiconductor Manufacturing Etching process. They show the results of their application which is a powerful tool but only for one limited area. It is not scalable for some other processes where you can customize your application for different manufacturing processes [14].

Zhe Li, Jingyue Li, Yi Wang, Kesheng Wang proposed a novel deep learning-based method for anomaly detection in mechanical equipment by combining two types of deep learning architectures, stacked autoencoders (SAE) and long short-term memory (LSTM) neural networks, to identify anomaly condition in a completely unsupervised manner. They made an experiment for anomaly detection in rotary machinery through wavelet packet decomposition (WPD) and data-driven models demonstrated the efficiency and stability of the proposed approach. But their work can be useful only for rotary machinery equipment which is not also scalable for other types of equipment [16].

Yujie Wang; Xin Du; Zhihui Lu; Qiang Duan; Jie Wu improved LSTM model to detect anomalies in equipment in rail transit systems. But their solution was implemented only for one problem and it is not scalable for other fields [18]. Mahe Zabin, Ho-Jin Choi, Jia Uddin presented hybrid DTL architecture comprising a deep convolutional neural network and long short-term memory layers for extracting both temporal and spatial features enhanced by Hilbert transform 2D images. They proposed a new customization of the model but not the application that is ready and scalable to solve problems [21].

Preeti Rajni Bala, Ram Pal Singh in their research considered the theoretical material for the analysis of time series, and also elaborated the information about the neural network with long and short-term memory [3]. Sheng Xiang, Yi Qin, Caichao Zhu, Yangyang Wang, Haizhou Chen presented material on the use cases of neural networks with long and short-term memory. Considered the use of this network in the example of predicting the resource of mechanical equipment [4]. The same scientists investigated the use of the network with long and short-term memory at a deeper level with the control state of the wear of the gears of mechanical equipment, in addition, they considered the basic concepts of the algorithms of this neural network [5].

Anuraganand Sharma conducted a review of existing neural network optimization methods. Analyzing the advantages and disadvantages of various modifications of gradient descent, he made an overview of the variation of gradient descent - stochastic gradient descent [6]. Farajtabar M., Azizan N., Mott A., Li A. considered variations of gradient descent and investigated superficial

information about orthogonal gradient descent for continuous learning [7]. Azizan, Navid & Hassibi, Babak reviewed and investigated variational gradient descent as well as stochastic gradient descent [8]. Simon Du, Jason Lee, Haochuan Li, Liwei Wang, Xiyu Zhai investigated the problem of the local minimum, as well as the possibility of using various methods to find the global minimum [9].

Lee, J.D., Simchowitz, M., Jordan, M.I., Recht, B. conducted an introduction to the problem of saddle points in the gradient descent method, and tried to optimize the usual gradient descent [10]. Soukup D., Cejka T., Hynek K. listed options for using machine learning methods to detect anomalies in various areas, and conducted an overview of anomaly detection in computer networks [11]. Grcić, M., Bevandić, P., Šegvić, S. provided an introduction to hybrid anomaly detection for dense open dataset recognition and a review of existing methods for dataset anomaly detection [12].

Based on reviewed articles the main problem is that all related works had a solution for one area and were not scalable. As result, it is necessary to create a common solution that can be used in different areas with scalable functionality which is ready to detect anomalies in almost any automation system.

3. Models & Methods & Technology

3.1. Analysis of problems and features of the anomaly detection process

Anomaly detection (or outlier detection) is the identification of rare items, events, or observations that are suspicious because of significant differences from the majority of the data. Typically, abnormal data can be related to some problem or rare event, such as bank fraud, medical problems, structural defects, malfunctioning equipment, etc. This relationship makes it very interesting to be able to choose which data points can be considered anomalies, as identifying these events is usually very interesting from a business perspective.

Any machine, whether rotating (pump, compressor, gas or steam turbine, etc.) or non-rotating machine (heat exchanger, distillation column, valve, etc.) will eventually reach a point of breakage. This point may not be an actual failure or shutdown, but the point at which the equipment is no longer operating in an optimal state. It means that some maintenance may be required to restore its full operating potential. Simply put, determining the "health" of our equipment is the realm of health monitoring.

The most common way to perform condition monitoring is to look at each machine sensor measurement and set a minimum and maximum value limit for it. If the current value is within the limits, then the machine is working. If the current value is out of range, it means the machine is faulty and an alarm is sent. But this procedure for setting hard-coded alarm limits is known to send a large number of false alarms, that is, alarms for situations that are actually healthy for the machine. There are also no alarms, i.e. situations that are problematic but not alarming. The first problem leads to an additional expenditure of time and effort. The second problem is more important because it leads to real damage along with repair costs and loss of productivity.

Both problems can be the result of the same cause: human error. Even if you seat several operators to view all sensor values, the station will not achieve energy-efficient monitoring of the equipment, because a human may miss a value, and it takes a lot of time to view volumetric data. That is why it is an optimal case to use automated methods of finding anomalies. In this way, you can avoid human error, and focus the attention of service personnel on the anomalies already found by the application, instead of reviewing the entire volume of data.

Let's imagine a certain station, each of them, with a high probability, will have certain equipment with completely different parameters, units of measurement, number of sensors, etc. In this way, the issue of controlling all the equipment, keeping reports, and the process of detecting abnormal values is complicated, if we take into account the proportional growth of stations to the growth of personnel. As a result, with the increase in equipment, the process of detecting incorrect behavior of certain automated systems becomes more and more difficult.

Usually, in this case, more personnel are hired, and more operators keep reports and try to visually detect the incorrect operation of this or that automated system. Or various applications are being developed that are configured specifically for certain automated systems. Therefore, it should be noted that the versatility of use plays a very important role in the further design and further

development of the software product, as the above-mentioned different equipment have a completely different set of sensors, from simple temperature sensors to complex sensors that can measure pressure, voltage, speed, etc.

3.2. Selection of tools of implementation of the program

Many different methods are used for data analysis, but it is advisable to approach the choice reasonably, as for each problem there is a more suitable solution that will affect the further operation of the software application and how the result can affect the operation of the business itself.

An LSTM neural network is well-suited for anomaly detection.

LSTM (long short-term memory) is an artificial neural network used in the field of artificial intelligence and deep learning. Unlike standard feedforward neural networks, LSTM has feedback. Such a recurrent neural network (RNN) can process not only individual data points (such as images), but also entire data sequences (such as speech or video). Due to the fact that this neural network can remember sequences of data, it helps to determine the behavior of a certain sensor and further check whether subsequent data matches this behavior. This means that this neural network can be used for many completely different graphs, which have a certain consistent behavior that helps to unify the application for use in completely different equipment.

Unlike any other data analysis method, LSTM is very good at detecting outliers in a graph. But in order to conduct data analysis, you first need to prepare the data. First, you need to standardize the data. This is necessary for the correct operation of the neural network.

The standard evaluation of sample x is calculated by the formula $z = (x - u) / s$, where u is the mean of the training samples and s is the standard deviation of the training samples. Centering and scaling are performed independently for each feature by computing appropriate statistics on the samples in the training set. The mean and standard deviation are then stored for further use in the data using a transformation.

Dataset standardization is a common requirement for many machine learning estimators: they can perform poorly if individual features do not in some way resemble standard normally distributed data (eg, a Gaussian with zero mean and unit variance).

After the standardization of the data set, it should be divided into training and test samples. Usually, the sample for train data is 70% of the total number, and for testing - 30%. The training data is used to train the model. Tests for checking the trained model.

In Figure 1, we can see the sensor data, which cannot be analyzed for anomalies using a fixed minimum and maximum.

Thanks to this neural network, we can work with different graphs, even with similar graphs, which are shown in Figure 1, where it is impossible to simply set the minimum and maximum, beyond which the value of the sensor cannot go, since there is a certain interval sequence, decreasing the interval, and increasing it. An example of an anomaly on a similar sequence can be the examples shown in Figure 2.

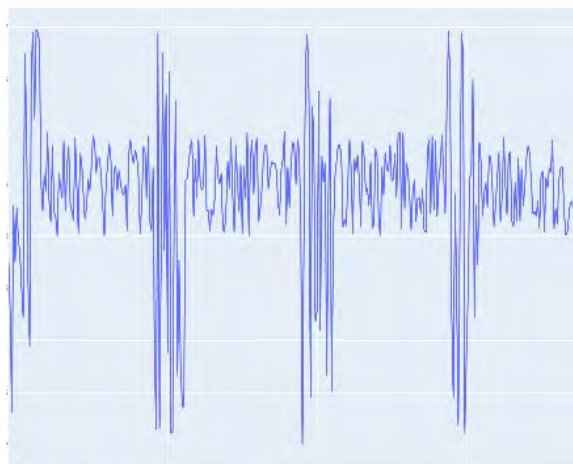


Figure 1: Data before standatrisation

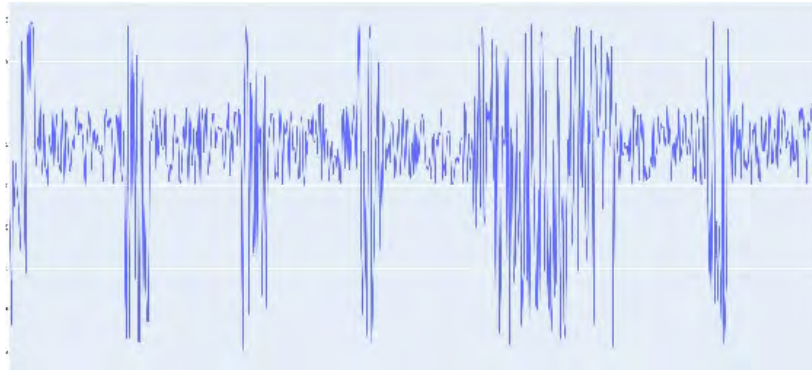


Figure 2: Anomaly example

3.3. Development of an anomaly analyzer using a neural network

Python should be used as the main programming language.

For implementation, it is suggested to use the following modules:

- scikit-learn — Python module for machine learning, based on SciPy.
- pandas — a software library is written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series.
- Numpy — module that adds support for large multidimensional arrays and matrices, along with a large library of high-level math functions for manipulating these arrays.
- Plotly — for data visualization.
- TensorFlow — for model realization.

Let's take the LSTM neural network as a basis - it is an artificial neural network, the advantages of which are that it avoids the problems of explosion or fading of the gradient since it does not change the weights of the paths as with the usual method of back error propagation. In this way, we can analyze quite different data graphs, and prepare a fairly universal software application that will be useful in many areas.

3.4. Neural network

To understand the work of the LSTM neural network, let's first familiarize ourselves with its main components, Figure 3 shows the basic structure of the LSTM.

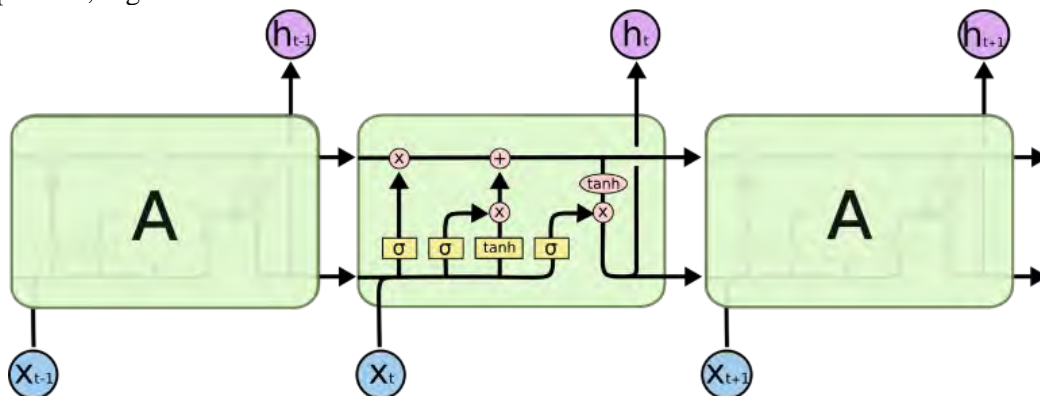


Figure 3: Structure of LSTM

Consider the main elements of the structure shown in Figure 4. " σ " refers to the sigmoid.

A sigmoid is a continuously differentiable monotonic non-linear S-shaped function that is often used to "smooth" the values of some quantity. It is determined by the formula:

$$S(x) = \frac{1}{1 + e^{-x}} \quad (1)$$

With the help of sigmoids, it is easy to get a number in the range from 1 to 0 depending on the number that will be input as x. The graphic view of the sigmoid is shown in Figure 4.

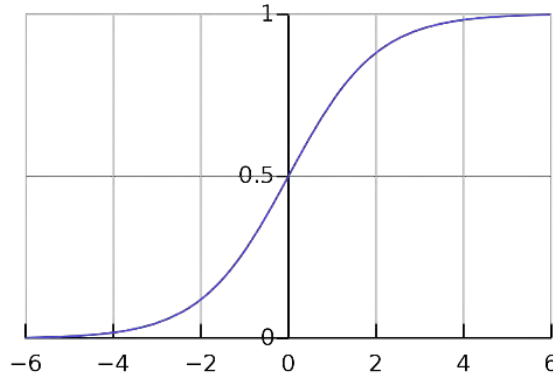


Figure 4: Sigmoid function

The notation “tanh” means the hyperbolic tangent function, which returns a result in the range from -1 to 1. The hyperbolic tangent function is the hyperbolic counterpart of the circular tangent function, which is used throughout trigonometry. The graphic representation is shown in Figure 5.

It is determined by the formula:

$$f(x) = \frac{(e^x - e^{-x})}{(e^x + e^{-x})} \quad (2)$$

Conventional notation “+” and “×” denote the operations of addition and multiplication, respectively. Having analyzed the symbols on the general diagram, let's move on to a detailed review of the work of the LSTM neural network.

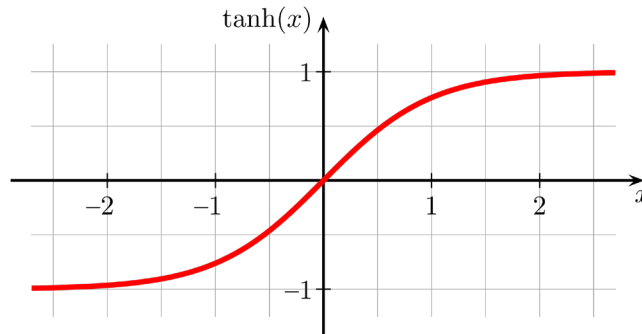


Figure 5: Hyperbolic tangent function

The Figure 6 shows the first stage of the neural network. The first layer can also be called the Forget gate layer. It is in this layer that it is determined which information can be forgotten and which can be left. The value of the previous output h_{t-1} and current input x_t are passed through the sigmoid layer. The obtained values are in the range $[0; 1]$. Values closer to 0 will be forgotten, and values closer to 1 will be save.

The formula of current step.

$$f_t = \sigma(W_f \times [h_{t-1}, x_t] + b_f) \quad (3)$$

Next, it is decided what new information will be stored in the state of the cell. This stage consists of two parts. First, a sigmoidal layer called input layer gate which determines which values should be updated. The hyperbolic tangent layer then constructs a vector of new candidate values that can be added to the cell. In Figure 7 schematically depicts the work of the second stage.

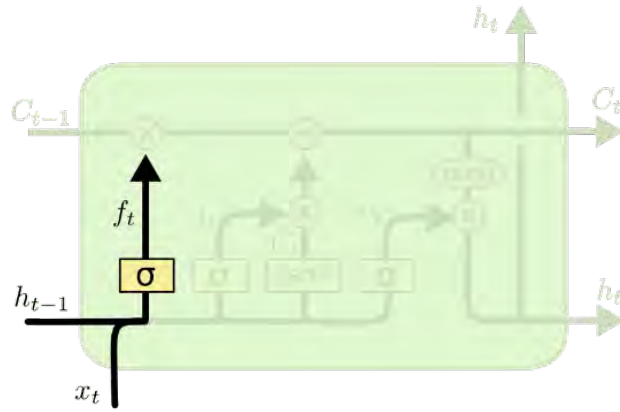


Figure 6: First step of LSTM

The formula of current step.

$$\begin{aligned} i_t &= \sigma(W_i \times [h_{t-1}, x_t] + b_i) \\ \tilde{C}_t &= \tanh(W_c \times [h_{t-1}, x_t] + b_c) \end{aligned} \quad (4)$$

At the third stage, to replace the old state of the cell C_{t-1} with new condition C_t . It is necessary to multiply the old state with f_t , forgetting what had decided to forget earlier. Then add an expression $i_t * \tilde{C}_t$. As result we have a new candidates for the value. In Figure 8 schematically depicts the work of the third stage.

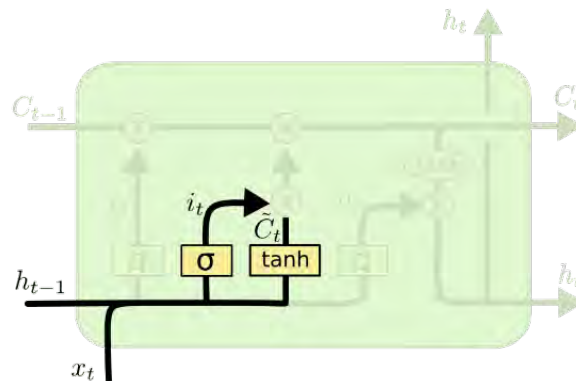


Figure 7: Second step of LSTM

The formula of current step.

$$C_t = f_t \times C_{t-1} + i_t \times \tilde{C}_t \quad (5)$$

At the last stage, it is determined what information will be received at the output. The output will be based on our cell state, with some filters applied to it.

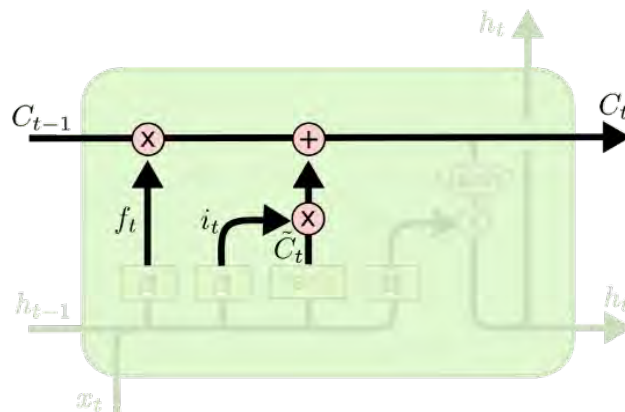


Figure 8: Third step of LSTM

First, the value of the previous output h_{t-1} and current input x_t are passed through a sigmoid layer that decides what cell state information will be output. Then the values of the state of the cell are passed through the hyperbolic tangent layer to obtain at the output values from the range from -1 to 1, and are multiplied with the output values of the sigmoid layer, which allows to output only the necessary information. In Figure 9 schematically depicts the work of the last fourth stage.

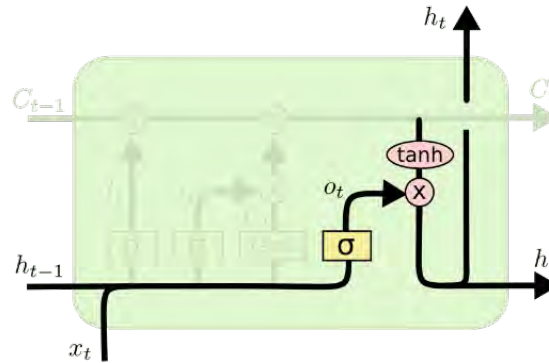


Figure 9: Fourth step of LSTM

The formula of current step.

$$\begin{aligned} o_t &= \sigma(W_o[h_{t-1}, x_t] + b_o) \\ h_t &= o_t \times \tanh(C_t) \end{aligned} \quad (6)$$

Obtained in this way h_t i C_t are passed down the chain. Iterative gradient descent with error backpropagation could be used to minimize the total error, but the main problem with gradient descent for standard recurrent neural networks is that the error gradients decrease at an exponential rate as the time delay between important events increases, which was discovered in 1991 [1,2]. With LSTM blocks, however, as the error magnitudes propagate backward from the source layer, the error appears locked in the block's memory. Thus, regular error backpropagation is effective for training an LSTM block to remember values for very long time intervals.

3.5. Dataset creation

For a real check of the algorithm, it is desirable to have real equipment that would return certain values, but since the equipment can be different, it was decided to analyze Internet resources and find out what types of graphs can be returned from automated systems.

After analyzing Internet resources, it was decided to generate data in a certain range, periodic data with repetition of oscillations, and sinusoidal data for examples. Data in a certain range are shown in Figure 10. An example of periodic data is shown in Figure 11. An example of sinusoidal data is shown in Figure 12.

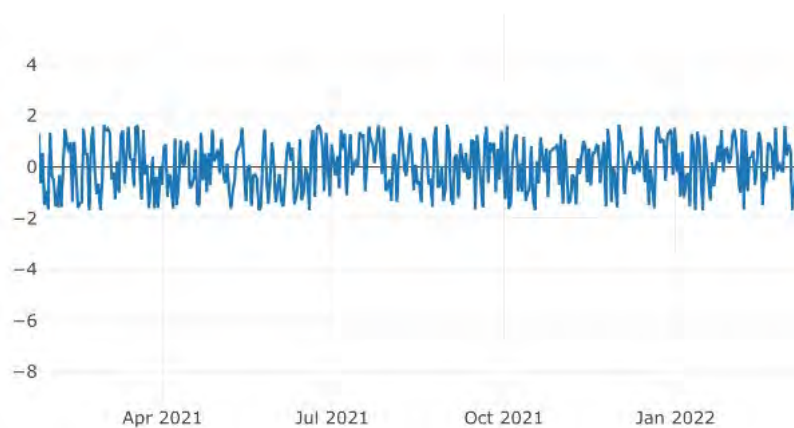


Figure 1: Dataset example

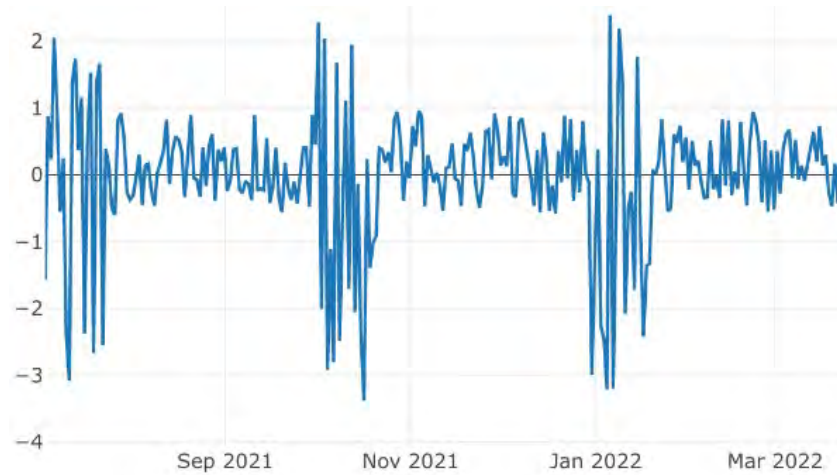


Figure 2: An example of periodic dataset

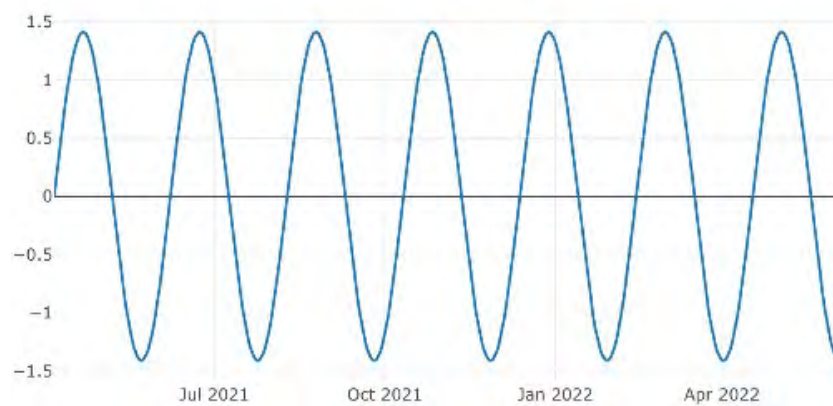


Figure 3: Sinusoidal dataset

3.6. Preparation for data analysis

To work with artificial intelligence algorithms, you need to prepare data in the appropriate format. After generating the dataset and saving the data in a csv file, you should select only the necessary lines and perform data standardization.

Data standardization is the process of converting data into a common format so that analysts can process and analyze it. Most organizations use data from multiple sources; this can include on-premises data storage, cloud storage, and various databases. However, data from different sources can be problematic if it is not homogeneous, leading to difficulties later (for example, when you use this data to create dashboards, visualizations, etc.).

Data standardization is critical for many reasons. Above all, it helps establish clear, coherently defined elements and attributes, providing a complete catalog of your data. No matter what statistics we're trying to get or what problems we're trying to solve, getting the data right is an important starting point.

This requires converting this data into a single format with logical and consistent definitions. These definitions will form metadata—labels that identify different aspects of the data. This is the basis of the data standardization process.

In terms of accuracy, standardizing the way data is labeled will improve access to the most relevant pieces of information. This will simplify analytics and reporting. It is calculated according to the formula:

$$Z = \frac{x - \mu}{\sigma} \quad (7)$$

In the formula mentioned above, x is a point from the dataset, μ is the arithmetic mean of the dataset, σ is the standard deviation of the dataset.

The formula for calculating the standard deviation can be presented as:

$$\sqrt{\frac{\sum(x_i - \bar{x})^2}{n - 1}} \quad (8)$$

where \bar{x} – is the arithmetic mean of the dataset, n – is the number of values of the dataset, x_i – is the i th value from the dataset.

After the standardization of the data set, it should be divided into training and test samples. Usually, the sample for training is 70% of the total number, and for testing - 30%. The training data is used to train the model. Tests for checking the trained model.

3.7. Interface

After successful registration, the user can familiarize himself with the main functionality of the software application. On the stations page, a user with manager rights can create stations, change or delete one of the stations. The station page is shown in Figure 13.

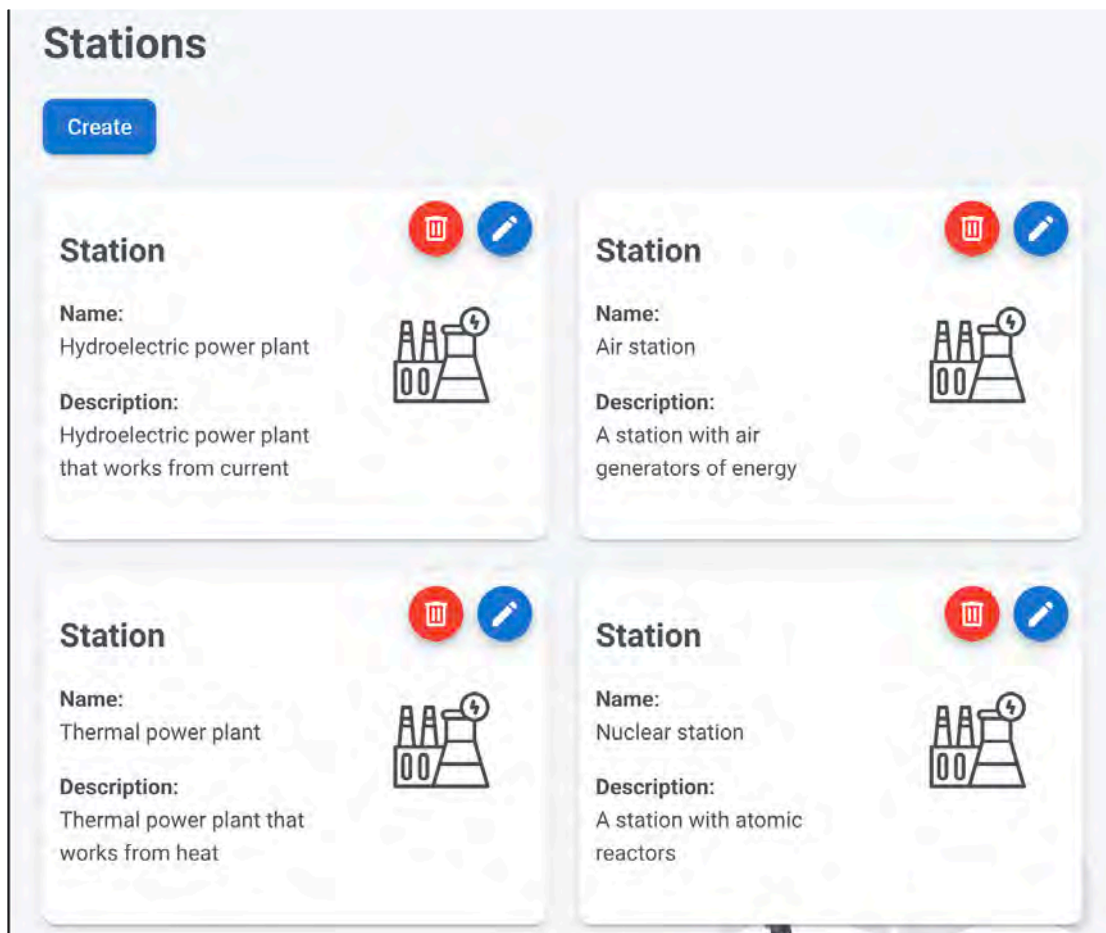


Figure 4: Stations page

Before the user creates station equipment and equipment sensors, it is necessary to create equipment types, which are actually a certain automated system of this station. Since different equipment can have similar properties at different stations these are common features that should be included in the type of equipment. The same logic should be used when creating types for sensors, since, for example, temperature sensors can be installed on different equipment and it does not make sense to create the same information for different types of equipment every time. The equipment types page is shown in Figure 14. The sensor types page is similar to the previous one.

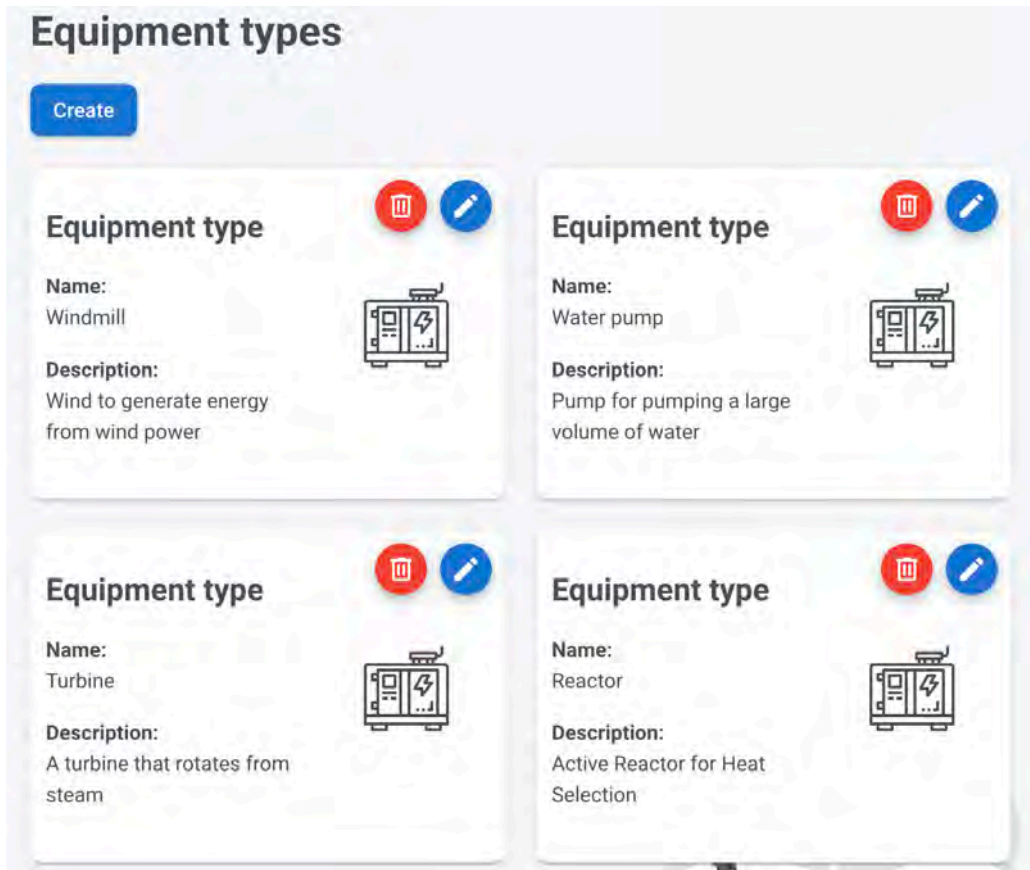


Figure 5: Equipment type page

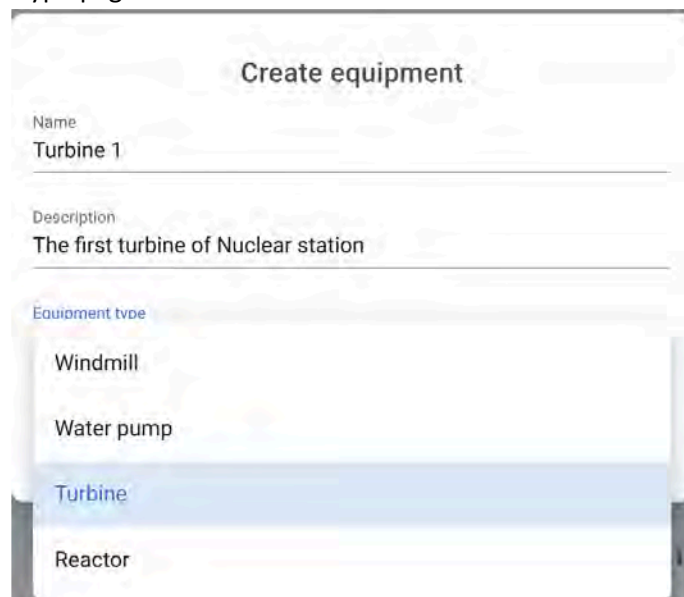


Figure 6: Equipment type context menu

After creating equipment types and sensor types, the user can start creating equipments and sensors using pre-prepared types. The user can select the appropriate type from the drop-down menu shown in Figure 15. The logic of the work of creating sensors is similar, with the exception of the choice of the type of graph and the presence or absence of anomalies, since the dataset is generated, and not taken from a real automated system, since there is no access to such at the development stage.

After that user can check the equipment list with all the necessary details, and edit or delete them, equipment page is shown in Figure 16.

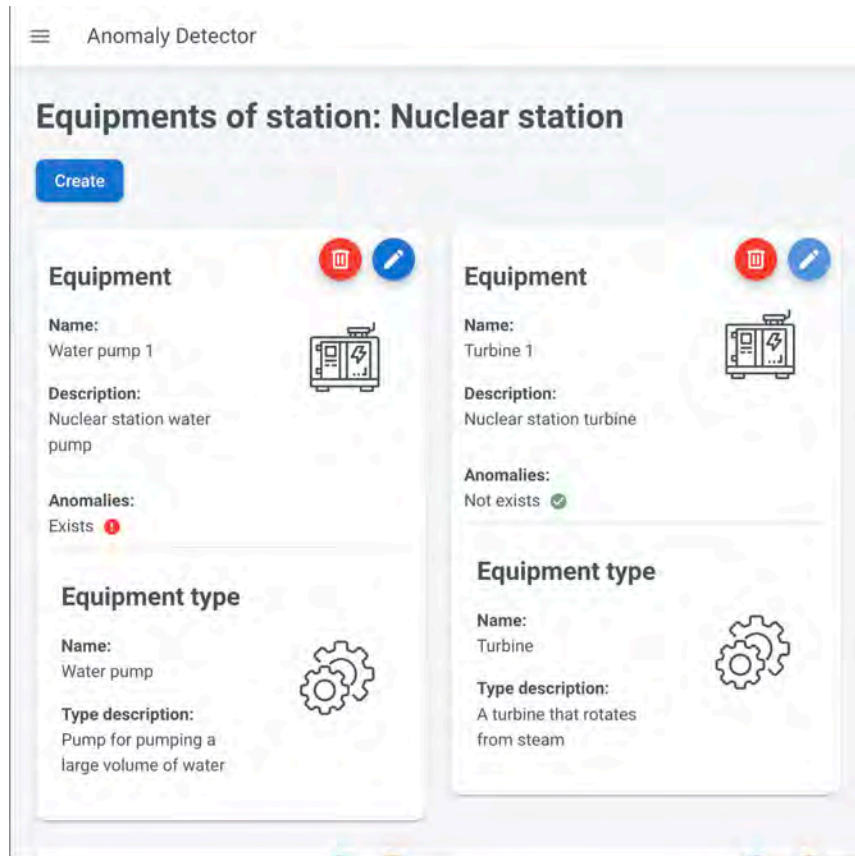


Figure 7: Equipment page

Inside every equipment the user can create sensors. Among the graph types, you can choose periodic, with a certain period of repetition of data ranges, normal, where there is a certain range and values that do not go beyond it, and sine wave. You can also choose the type in the drop-down menu, which is shown in Figure 17. The sensors page is similar to the equipments page.

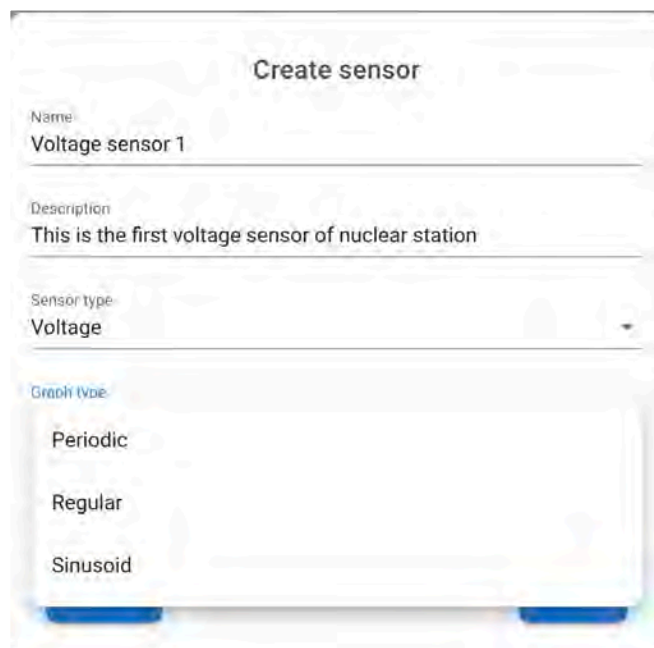


Figure 8: Selection of graph types

4. Experiments

For experiments researched internet resources for checking the data that different equipment can return from different sensors. Picked up a regular graph which should return data in diapason from -2 to 2, shown in Figure 18, a repeating graph that has some repeating trend where the big part has a value between -4 to 3 and the small part from -1 to 1, shown in Figure 19, and sinusoid graph, shown in Figure 20.

After that was imitated anomalies which is shown in Figure 21. For application training data was split for train and test batches. After all training and testing, the application analyzes data, find anomalies, and show the results, anomalies were highlighted as orange dots, shown in Figure 21.

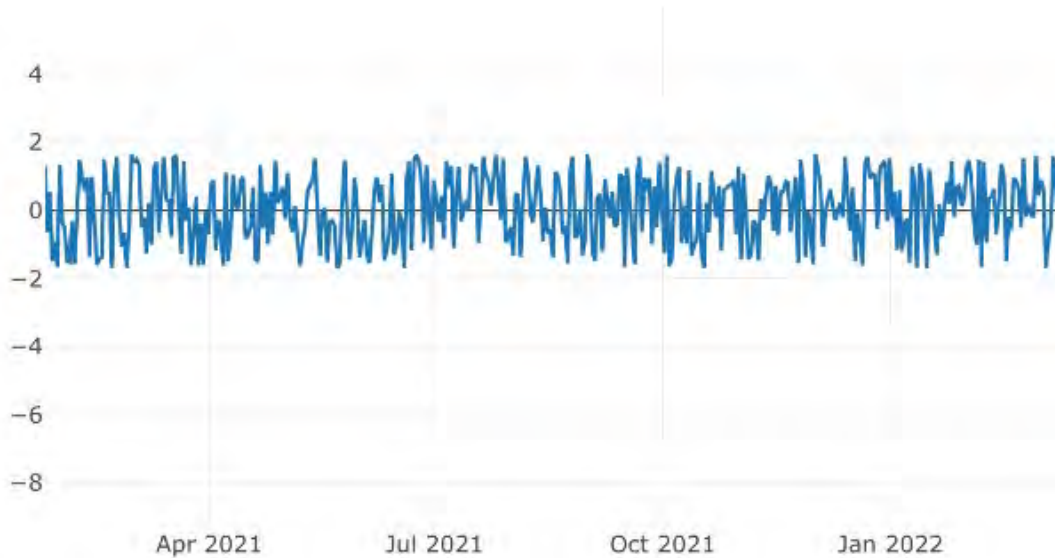


Figure 9: Regular graph

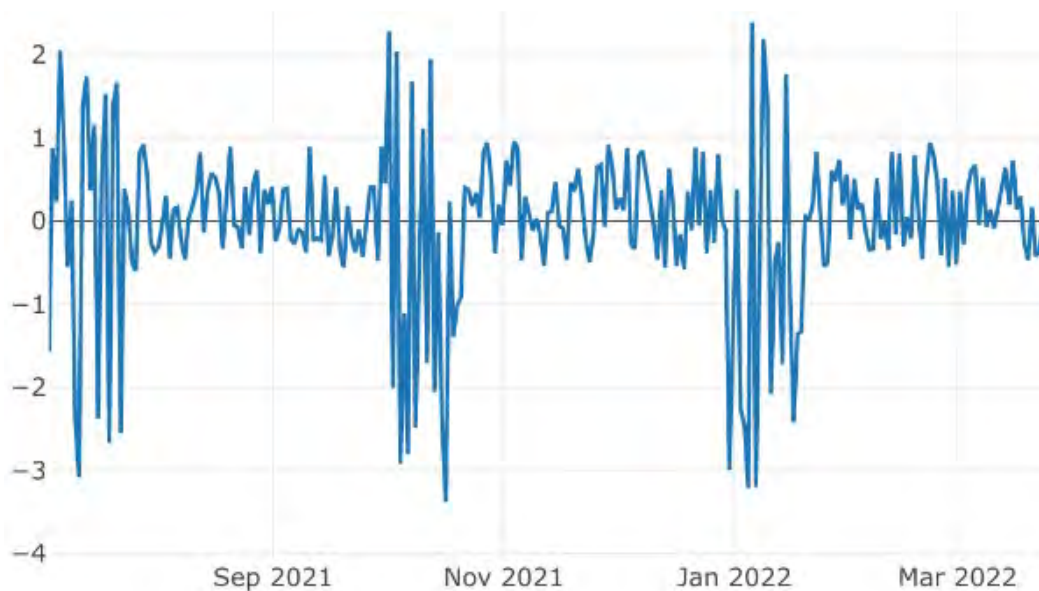


Figure 19: Repeating graph that has repeating trend

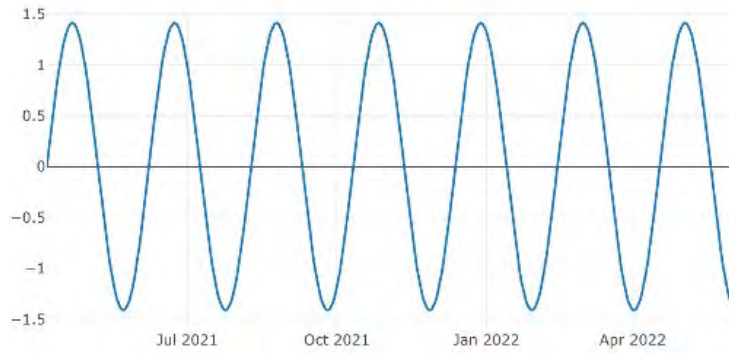


Figure 10: Sinusoid graph

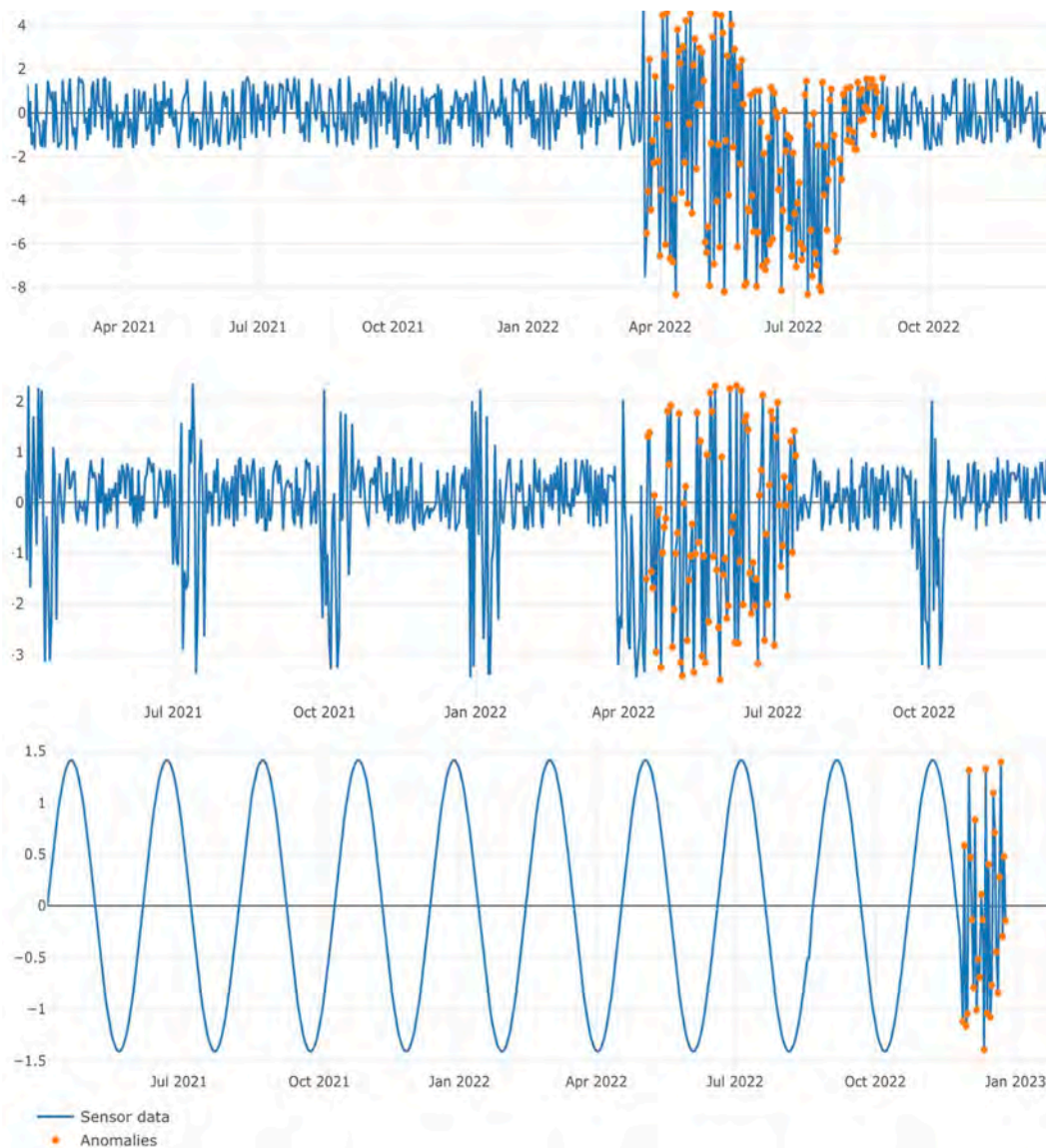


Figure 11: Experiments results

The application is ready to analyze and return results for radically different graphs, which can solve the problem of previously considered studies that were prepared only for certain equipment. This application has solutions for completely different equipment, which as a result is flexible in further use at various stations using different equipment, from pumps and turbines to nuclear reactors.

If the system did not detect any anomalies, the user will be informed by the inscription that there are no anomalies in the specific equipment on all sensors, shown in Figure 22.

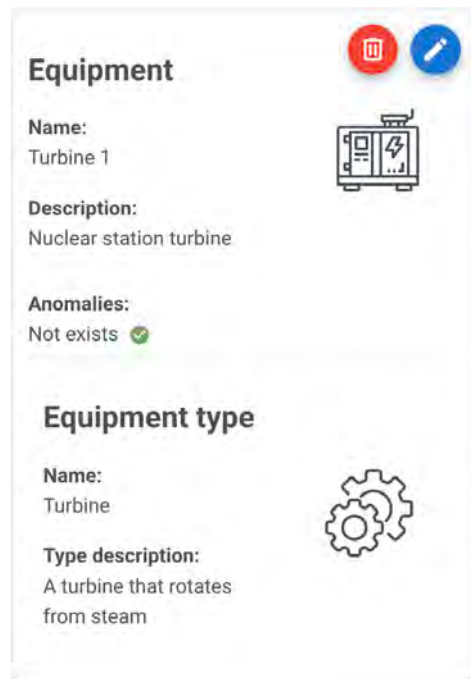


Figure 12: Equipment without anomalies

If anomalies were detected, the status will change to "Exist". To study in detail where exactly the anomaly occurred, the user needs to click on the equipment that he needs to examine, and after going to the page of all sensors of this equipment, analyze the sensors for the presence of the "Exist" status. It is shown in Figure 23.

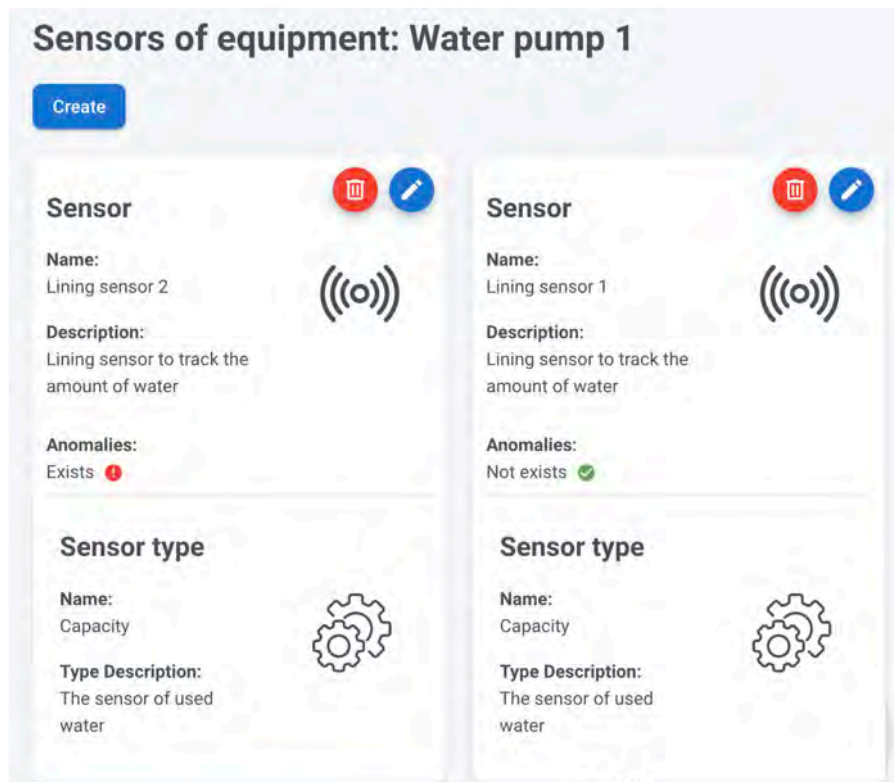


Figure 13: Page of sensors of this equipment with the present anomalies

After that, the user can go to the sensor that he needs to investigate and analyze the results of the anomaly detection in this automated system. The graph of similar results is shown in Figure 24.

If anomalies were detected, they are indicated by orange dots, as shown in Figure 24. If there are no anomalies, then the graph will consist only of the indicators of the current sensor of this equipment, shown in Figure 25. In this way, the program can analyze quite different graphs, an example is shown in Figure 21.



Figure 14: The page of the sensor of this equipment with the exist anomalies

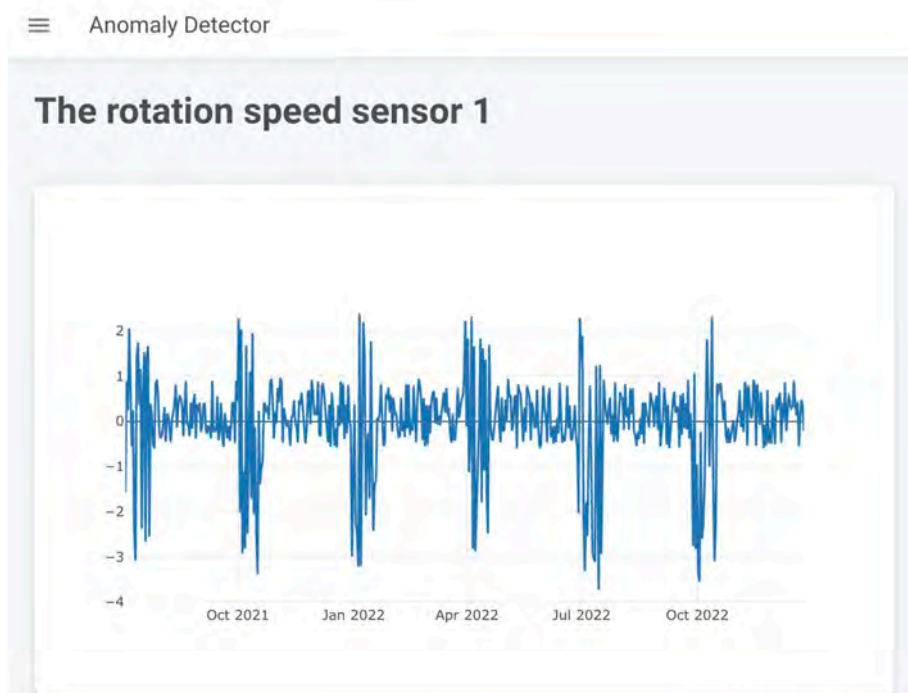


Figure 15: The page of the sensor of this equipment without anomalies

5. Conclusions

In this work, a user-friendly software application was implemented. What is convenient for keeping records of various automated station systems and a mechanism for automatic detection of anomalies in them has been implemented. The technical task was prepared and the correct tools for the implementation of the program were chosen. Analyzed different researcher works with their advantages and disadvantages. Was implemented a scalable application that can be used in different equipment which is the main novelty. The application was tested with different experiments which showed acceptable results for future use.

The main role in the detection of anomalies is played by the LSTM neural network. This work describes the process of data preparation for its use, its creation and training. To train the neural network, a proprietary dataset generator was created after analyzing various Internet resources. The process of choosing a specific neural network architecture is described. A number of algorithms for preliminary data preparation were also implemented.

Prospects for further research should be in-depth investigations of the performance of an LSTM natural network in this context. It is necessary to check the current application in various areas. It is worth conducting a study on the success of this software application using a large amount of real data from equipment indicators of automated systems.

6. References

- [1] X. Yang, M. Xu, S. Xu and X. Han, "Day-ahead forecasting of photovoltaic output power with similar cloud space fusion based on incomplete historical data mining", *Appl. Energy*, vol. 206, pp. 683-696, 2017. <https://www.sciencedirect.com/science/article/abs/pii/S0306261917312564?via%3Dihub>
- [2] M. Ceci, R. Corizzo, F. Fumarola, D. Malerba and A. Rashkovska, "Predictive modeling of PV energy production: How to set up the learning task for a better prediction?", *IEEE Trans. Ind. Informat.*, vol. 13, no. 3, pp. 956-966, Jun. 2017. <https://ieeexplore.ieee.org/document/7556989>
- [3] Preeti Rajni Bala, Ram Pal Singh. A dual-stage advanced deep learning algorithm for long-term and long-sequence prediction for multivariate financial time series. *Applied Soft Computing*. Volume 126, 2022, 109317. <https://doi.org/10.1016/j.asoc.2022.109317>.
- [4] Sheng Xiang, Yi Qin, Caichao Zhu, Yangyang Wang, Haizhou Chen. LSTM networks based on attention ordered neurons for gear remaining life prediction. *ISA Transactions*. Volume 106, 2020. Pp. 343-354. <https://doi.org/10.1016/j.isatra.2020.06.023>.
- [5] Sheng Xiang, Yi Qin, Caichao Zhu, Yangyang Wang, Haizhou Chen. Long short-term memory neural network with weight amplification and its application into gear remaining useful life prediction. *Engineering Applications of Artificial Intelligence*. Volume 91, 2020, 103587, <https://doi.org/10.1016/j.engappai.2020.103587>.
- [6] Anuraganand Sharma. Guided Stochastic Gradient Descent Algorithm for inconsistent datasets. *Applied Soft Computing*. Volume 73, 2018, pp. 1068-1080. <https://doi.org/10.1016/j.asoc.2018.09.038>.
- [7] Farajtabar M., Azizan N., Mott A., Li A. Orthogonal gradient descent for continual learning. *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy*. PMLR: Volume 108. <https://core.ac.uk/download/pdf/345075797.pdf>
- [8] Azizan, Navid & Hassibi, Babak. (2018). Stochastic Gradient/Mirror Descent: Minimax Optimality and Implicit Regularization. *ICLR 2019*. <https://openreview.net/pdf?id=HJf9ZhC9FX>
- [9] Simon Du, Jason Lee, Haochuan Li, Liwei Wang, Xiyu Zhai. Gradient Descent Finds Global Minima of Deep Neural Networks. *Proceedings of the 36th International Conference on Machine Learning*, PMLR 97:1675-1685, 2019. <http://proceedings.mlr.press/v97/du19c.html>
- [10] Lee, J.D., Simchowitz, M., Jordan, M.I. & Recht, B.. (2016). Gradient Descent Only Converges to Minimizers. *29th Annual Conference on Learning Theory*, in *Proceedings of Machine Learning Research*. 49: 1246-1257. <https://proceedings.mlr.press/v49/lee16.html>.

- [11] Soukup D., Cejka T., Hynek K. (2020) Behavior Anomaly Detection in IoT Networks. Lecture Notes on Data Engineering and Communications Technologies, 49, pp. 465 - 473. DOI: 10.1007/978-3-030-43192-1_53
- [12] Grcić, M., Bevandić, P., Šegvić, S. (2022). DenseHybrid: Hybrid Anomaly Detection for Dense Open-Set Recognition. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds) *Computer Vision – ECCV 2022*. ECCV 2022. Lecture Notes in Computer Science, vol 13685. Springer, Cham. https://doi.org/10.1007/978-3-031-19806-9_29
- [13] Benjamin Lindemann, Benjamin Maschler, Nada Sahlab, Michael Weyrich. (2021). A survey on anomaly detection for technical systems using LSTM networks. *Computers in Industry*, Volume 131, <https://doi.org/10.1016/j.compind.2021.103498>
- [14] Gian Antonio Susto, Matteo Terzi, Alessandro Beghi. (2017). Anomaly Detection Approaches for Semiconductor Manufacturing. *Procedia Manufacturing*, Volume 11, Pages 2018-2024, <https://doi.org/10.1016/j.promfg.2017.07.353>.
- [15] Bashar M. Haddad, Sen Yang, Lina J. Karam, Jieping Ye, Nital S. Patel, Martin W. Braun. (2018). Multifeature, Sparse-Based Approach for Defects Detection and Classification in Semiconductor Units. *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 1, pp. 145-159, Jan. 2018, doi: 10.1109/TASE.2016.2594288.
- [16] Li, Z., Li, J., Wang, Y. et al. (2019) A deep learning approach for anomaly detection based on SAE and LSTM in mechanical equipment. *The International Journal of Advanced Manufacturing Technology*, volume 103, 499–510. <https://doi.org/10.1007/s00170-019-03557-w>
- [17] Yaguo Lei, Jing Lin, Zhengjia He, Ming J. Zuo (2013). A review on empirical mode decomposition in fault diagnosis of rotating machinery. *Mechanical Systems and Signal Processing*, Volume 35, Issues 1–2, Pp. 108-126. <https://doi.org/10.1016/j.ymssp.2012.09.015>.
- [18] Y. Wang, X. Du, Z. Lu, Q. Duan and J. Wu. (2022). Improved LSTM-Based Time-Series Anomaly Detection in Rail Transit Operation Environments. *IEEE Transactions on Industrial Informatics*, vol. 18, no. 12, pp. 9027-9036, doi: 10.1109/TII.2022.3164087.
- [19] M. Abdel-Nasser, K. Mahmoud and M. Lehtonen (2021) Reliable Solar Irradiance Forecasting Approach Based on Choquet Integral and Deep LSTMs. *IEEE Transactions on Industrial Informatics*, vol. 17, no. 3, pp. 1873-1881, doi: 10.1109/TII.2020.2996235.
- [20] R. Quan, L. Zhu, Y. Wu and Y. Yang (2021) Holistic LSTM for pedestrian trajectory prediction. *IEEE Trans. Image Process.*, vol. 30, pp. 3229-3239, <https://ieeexplore.ieee.org/document/9361440>
- [21] Zabin, M., Choi, HJ. & Uddin, J. (2023) Hybrid deep transfer learning architecture for industrial fault diagnosis using Hilbert transform and DCNN–LSTM. *J Supercomput* 79, 5181–5200. <https://doi.org/10.1007/s11227-022-04830-8>