Neuroevolution methods for organizing the search for anomalies in time series

Serhii Leoshchenko[a], Andrii Oliinyk[a], Sergey Subbotin[a], Matviy Ilyashenko[a] and Tetiana Kolpakova[a]

[a] *National University "Zaporizhzhia Polytechnic", Zhukovskogo street 64, Zaporizhzhia, 69063, Ukraine*

Abstract

This paper is devoted to the problem of detecting and classifying anomalies for time series data. Some of the important applications of time series anomaly detection are healthcare, fraud detection, and system failure recognition.

Despite scientists extensive experience in detecting anomalies in time series, most methods look for individual objects that differ from ordinary objects, but do not take into account the specifics of the data sequence [1].

In this paper, a method for detecting anomalies and clastering time series based on neuro-evolutionary approaches is proposed. Prediction-based methods are used to detect anomalies: statistical and deep neural networks are used. Classical clustering methods that accept statistical parameters of series were used for clustering.

Keywords 1

Forecasting, time series, clustering, neuroevolution, genetic method

1. Introduction

Effective operation of complex technological systems requires monitoring and various analytical methods that allow you to monitor, manage, and preemptively change parameters. Monitoring is usually provided by standard tools (in most cases, a fairly reliable data collection and visualization system) [2, 3]. But creating effective analytical tools requires additional research, experiments, and a good knowledge of the subject area. As a rule, there are four main types of data analytics [4]:

- descriptive analytics visualizes accumulated data, including transformations for clarity and interpretability. Descriptive analytics is the simplest type of analysis, but also the most important one for applying other analysis methods;
- with the help of diagnostic analytics, they investigate the cause of events in the past, while identifying trends, anomalies, the most characteristic features of the described process, looking for causes and correlations (relationships);
- predictive analytics predicts likely outcomes based on identified trends and statistical models derived from historical data;
- administrative analytics allows you to get the optimal solution to a production problem based on predictive analytics. For example, it can be optimizing the operation parameters of equipment or business processes, or a list of measures to prevent an emergency.

Modeling methods, including machine learning, are usually used for predictive and prescriptive analytics. The effectiveness of these models depends on the high-quality organization of data

The Sixth International Workshop on Computer Modeling and Intelligent Systems (CMIS-2023), May 3, 2023, Zaporizhzhia, Ukraine
EMAIL: sergleo.zntu@gmail.com (S. Leoshchenko); olejnikaa@gmail.com (A. Oliinyk); subbotin@zntu.edu.ua (S. Subbotin); matviy.ilyashenko@gmail.com (M. Ilyashenko); t.o.kolpakova@gmail.com (T. Kolpakova)
ORCID: 0000-0001-5099-5518 (S. Leoshchenko); 0000-0002-6740-6078 (A. Oliinyk); 0000-0001-5814-8268 (S. Subbotin); 0000-0003-4624-4687 (M. Ilyashenko); 0000-0001-8307-8134 (T. Kolpakova)

collection, processing, and preliminary analysis. The listed types of analytics differ both in the complexity of the models used and in the degree of human participation [2-6].

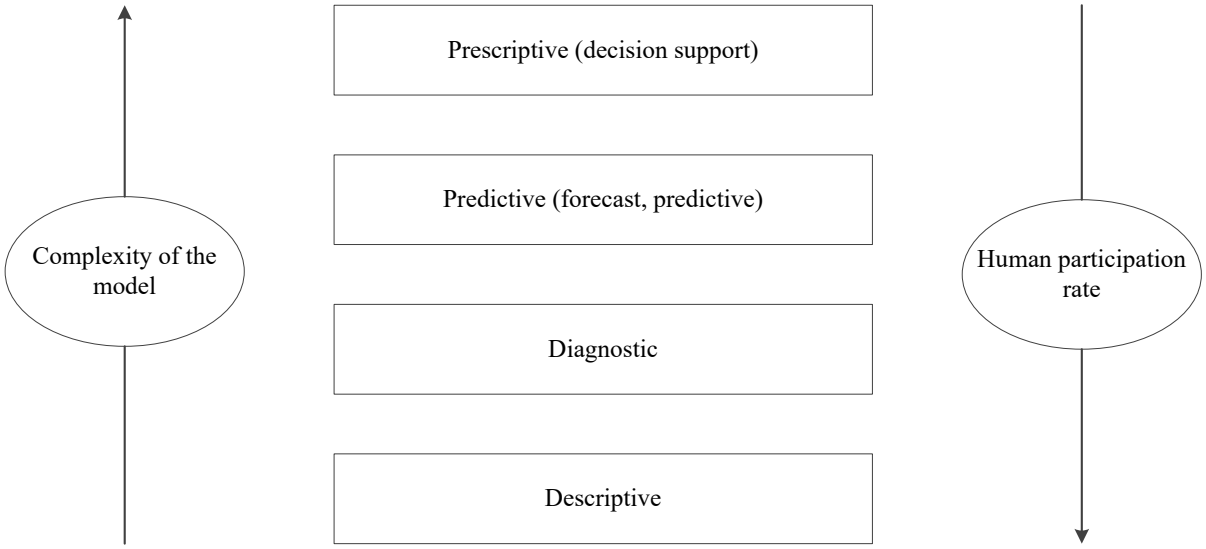

Figure 1: Comparison of model differences based on human participation and the complexity of the models themselves

There are a lot of areas of application of analytics tools-information security, banking sector, public administration, medicine and many other subject areas. Often, the same method works effectively for different subject areas, so developers of analytics systems create universal modules containing different algorithms [5].

For many technological systems, monitoring results can be represented as time series. The properties of the time series are [6]:

- linking each measurement (sample, discrete) to the time of its occurrence;
- equal time distance between measurements;
- ability to restore the behavior of the process in the current and subsequent periods from data from the previous period.

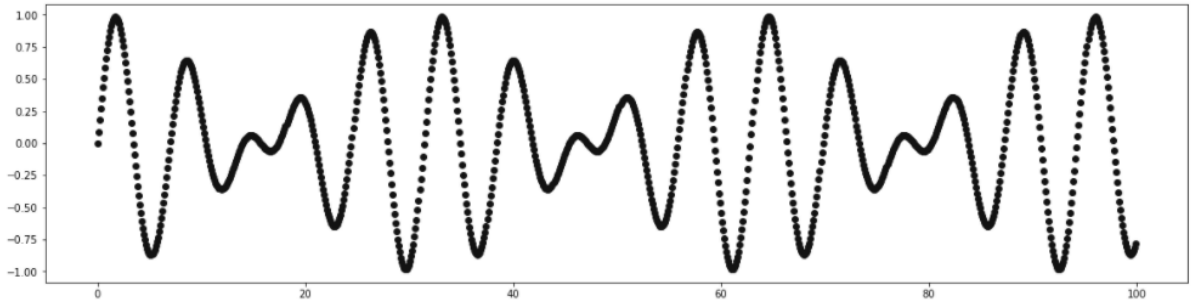

Figure 2: Example of a time series

Time series can describe more than just numerically measurable processes. The use of various methods and model architectures, including deep neural networks, allows you to work with data from natural language processing, computer vision tasks, etc. For example, a chat message can be converted into numeric vectors (embedding) that appear sequentially at a certain time, and the video is nothing more than a matrix of numbers that changes over time [7].

So, time series are very useful for describing the operation of complex devices and are often used for typical tasks: modeling, forecasting, feature selection, classification, clustering, pattern search, anomaly search [6]. Examples of such usage include an electrocardiogram, changes in the value of

shares or currency, weather forecast values, changes in network traffic, engine operation parameters, and much more.

Time series have typical characteristics that accurately describe the nature of the time series [2-6]:

- period: a time interval of constant length for the entire series, at the ends of which the series takes similar values,
- seasonality: periodicity property (season the same as period),
- cycle: characteristic changes in a series related to global causes (for example, cycles in the economy), there is no constant period,
- trend: a trend towards a long-term increase or decrease in the values of a series.

Time series may contain anomalies. An anomaly is a deviation in the standard behavior of a process. The method of machine search for anomalies uses data about the operation of the process (data sets) [7]. Depending on the subject area, there may be different types of anomalies in the dataset. It is customary to distinguish between several types of anomalies [8].

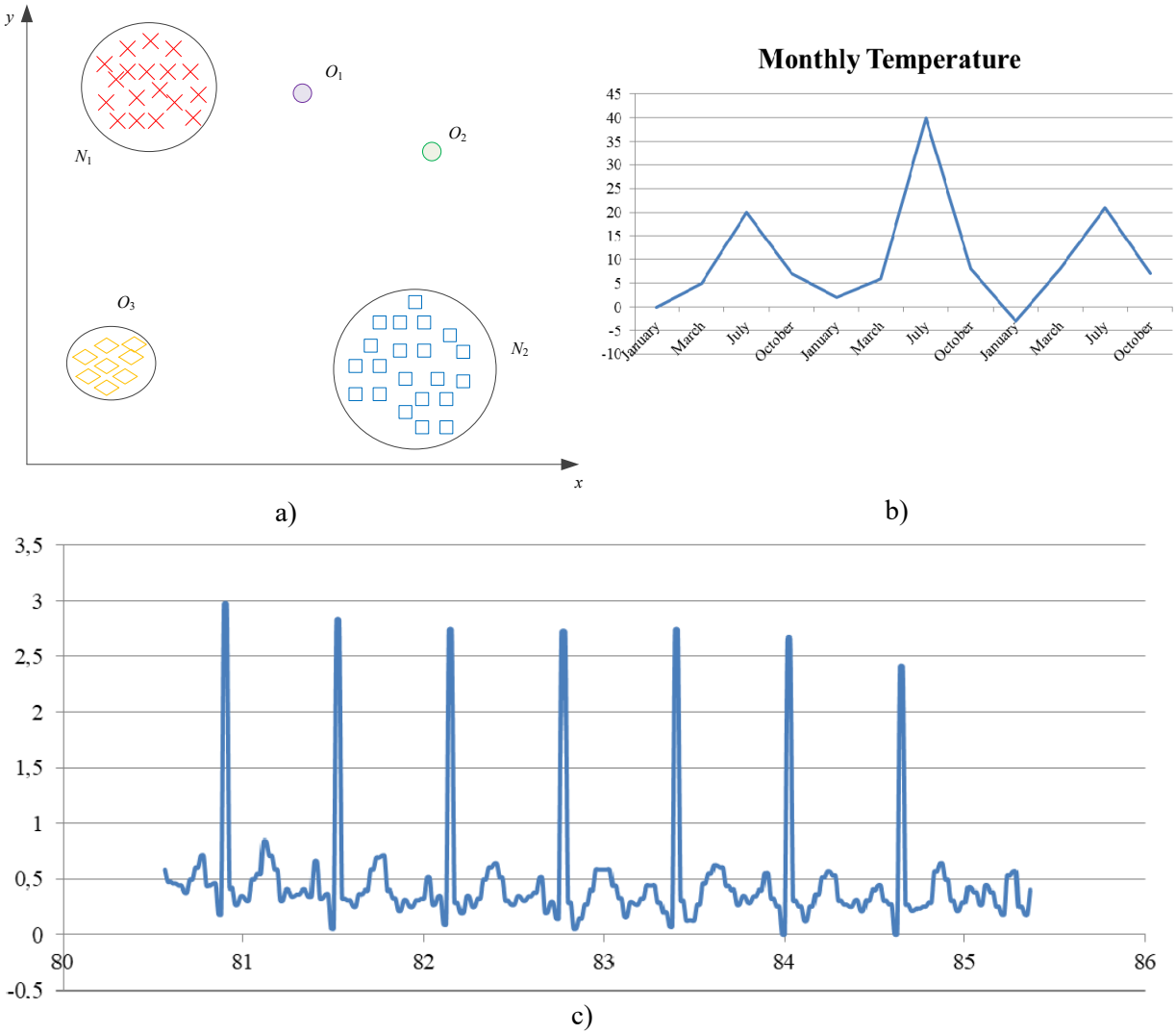

Figure 3: Examples of various anomalies in time series: a) point O_1 and O_2 and collective O_3 anomalies in two-dimensional data; b) contextual anomaly in the time series; c) collective anomaly in the time series (ECG): arrhythmia

1. Point anomalies. They occur in situations where a single instance of data can be considered as absolutely abnormal in relation to others [8].
2. Contextual anomalies. They are observed if the instance is abnormal in a certain context, or when a certain condition is met (therefore also called conditional) [8].

3. Collective anomalies. Occur when a sequence of related data instances (for example, a time series graph) is abnormal relative to the rest of the data [8].

An individual instance may not be a deviation, but the joint appearance of such instances will be a collective anomaly.

1.1. Anomaly detection strategies

Detecting point anomalies often requires some kind of system model. If the system does not have a deterministic mathematical model or such a model is too difficult to build, then a statistical model must be available. Depending on the method of constructing a statistical model, the following approaches are distinguished [8-10].

1. Recognize anomalies in case supervised learning. This technique requires a full-fledged training sample, including enough representatives of the normal and abnormal classes of values.

The technique is applied in 2 stages: first, training takes place on data that manually indicates normal and abnormal points. Then recognition occurs when new data is classified based on the constructed model [8-10].

It is usually assumed that the statistical properties of the model do not change over time, and such a change often requires repeated training.

The main difficulty of such methods is the formation of data for training. In addition to the obvious labor costs, often the abnormal class is also worse represented than the normal one, which can lead to inaccuracies in the resulting model [8-10].

2. Recognition of anomalies partially supervised learning. Similar to the previous one, but the training data represents only a normal class. A system trained in a normal class can determine whether data belongs to it, thus identifying abnormal data by exclusion.

3. Recognition in case unsupervised learning. In the absence of a priori information, this is the only possible option. Recognition in unsupervised learning: free mode is based on the assumption that abnormal data is quite rare. Therefore, only those that are farthest from the average values are indicated as anomalous [8-10]. Applying this technique to streaming data is difficult because it is necessary to have an idea of the entire data array to have a good estimate of the average and expected deviations.

1.2. Anomaly recognition methods

Classification. This method is based on the fact that the normal behavior of a system can be determined by one or more classes. Then an instance that does not belong to any of the classes is anomalous. This method usually uses the “partially supervised learning” approach. Basic mechanisms: neural networks, Bayesian networks, rule-based reference vector method [7, 9].

Clustering. This method is based on grouping similar values into clusters, and does not require knowledge of the properties of possible deviations. Anomaly detection is based on the following assumptions [8, 10]:

- normal data instances belong to a cluster
- normal data is closer to the center of the cluster, abnormal data is further away
- normal data forms large dense clusters, while abnormal data forms small and scattered ones.
- one of the simplest clustering methods is the *k*–means algorithm.

Statistical analysis. Using this approach, a statistical model of the process is constructed, which is then compared with the actual behavior. If the actual behavior differs from the model by more than a certain threshold, it is concluded that there are anomalies. Statistical analysis methods are divided into two groups [7]:

- parametric methods. Normal data is assumed to have a probability density function $\rho(x,\theta)$, where θ is the parameter vector, x is the data instance (observation);
- nonparametric method. The model structure is not defined a priori, but is determined from the data provided.

Nearest neighbor method. When using this technique, a metric is introduced (a measure of similarity between objects). Then two approaches are possible [10]:

- distance to the k^{th}-nearest neighbor. Abnormal data is the most distant from all other data;
- using relative density. Samples in areas with low relative density are evaluated as abnormal. (e.g. local emission level method).

Spectral methods. Based on the spectral (frequency) characteristics of the data, a model is constructed that is designed to take into account most of the variability in the data.

Let's compare the existing methods and present the results as a table.

Table 1

Comparison of anomaly recognition methods

Method	Result	Strategy	Classification of anomalies
Classification	Tag	Supervised learning, partially supervised learning	Yes
Clustering	Tag	Supervised learning, partially supervised learning	No
Statistical analysis	Degree	Partially supervised learning	No
Nearest neighbor	Degree	Unsupervised learning	No
Spectral methods	Tag	Unsupervised learning, partially supervised learning	No

In general, from the comparison of methods, we can conclude that there is an insufficient qualitative level of existing methods, because most methods are not able to use all strategies (approaches) to machine learning, and also from the results they do not provide an unambiguous answer about the assessment of detected anomalies. Moreover, most papers note an insufficient level of accuracy when using cool methods with newer model topologies. That is why the task of developing new approaches and methods for detecting anomalies in time series remains an urgent task.

2. Related Works

Point anomalies are most easily recognized-these are individual points where the behavior of the process differs dramatically from other points. For example, you can observe a sharp deviation of parameter values at a particular point [7-10].

Such values are called “outliers”, they strongly influence the statistical indicators of the process and are easily detected by setting thresholds for the observed value.

It is more difficult to detect an anomaly in a situation where the process behaves “normally” at each point, but collectively the values at several points behave “strangely”. Such abnormal behavior can include, for example, a change in the waveform, a change in statistical indicators (mean, mode, median, variance), the appearance of a mutual correlation between two parameters, small or short-term abnormal changes in the amplitude, and so on. And in this case, the task is to recognize abnormal behavior of parameters that cannot be detected by conventional statistical methods [7-10].

Finding anomalies is very important. In one situation, data should be cleared of anomalies in order to get a more realistic picture, while in another situation, anomalies should be carefully examined, as they may indicate a possible rapid transition of the device to emergency mode.

Finding anomalies in time series is not easy (unclear definition of anomalies, lack of markup, non-obvious correlation). So far, the Self Organizing Tree Algorithm (SOTA-algorithms) for searching for anomalies in time series has a high level of False Positive [7-10].

Only a small number of anomalies, mostly point-based, can be detected manually if good data visualization is available. Group anomalies are harder to detect manually, especially when it comes to large amounts of data and analyzing information about multiple devices. Also difficult to detect is the case of an "anomaly in time", when a normal signal appears at the "wrong" time. Therefore, when searching for anomalies in time series, it is advisable to use automation methods [7-10].

A big problem in finding anomalies on real data is that the data is usually not marked up, so initially it is not strictly defined what an anomaly is, there are no rules for searching. In such situations, it is necessary to apply methods of teaching without a teacher (unsupported learning), while models independently determine the relationships and characteristic laws in the data.

The methods used to search for anomalies in time series are usually divided into groups [7-10]:

- proximity-based: anomaly detection based on proximity parameter information or a sequence of fixed-length parameters, suitable for detecting point anomalies and outliers, but will not detect changes in the waveform;
- prediction-based: build a forecast model and compare the forecast and actual value, best applied to time series with pronounced periods, cycles, or seasonality;
- reconstruction-based: methods based on Data Fragment reconstruction use data fragment recovery (reconstruction), so it can detect both point anomalies and group anomalies, including changes in the waveform.

Proximity-based methods are focused on finding values that significantly deviate from the behavior of all other points. The simplest and most obvious example of implementing this method is monitoring whether a given threshold of values is exceeded [8].

In prediction-based methods, the main task is to build a qualitative process model in order to simulate the signal and compare the obtained simulated values with the original (true) ones. If the predicted and true signal are close, then the behavior is considered "normal", and if the values in the model are very different from the true ones, then the behavior of the system in this area is declared abnormal [9].

The most common time series modeling methods are SARIMA [11] and periodic neural networks.

The original approach is used in reconstruction-based models - first, the model is taught to encode and decode signals from an existing sample, while the encoded signal has a much smaller dimension than the original one, so the model has to learn to "compress" information. An example of such compression for 32-by-32-pixel images is given in [12].

After training, the models give input signals that are segments of the time series under study, and if encoding and decoding is successful, then the behavior of the process is considered "normal", otherwise the behavior is declared abnormal.

One of the newly developed reconstruction-based methods that show good results in detecting anomalies is TadGAN [13-15], developed by MIT researchers in late 2020. The TadGAN Method Architecture contains elements of an auto-encoder and Generative Adversarial Networks.

The network ε acts as an encoder that translates time series segments x into hidden space vectors z, and ς is a decoder that recovers time series segments from the hidden z representation. C_x is the critic who evaluates the recovery quality of $\varsigma(\varepsilon(x))$, and C_z is the critic who evaluates the similarity of the hidden representation of $z = \varepsilon(x)$ to white noise. In addition, there is a control of the "similarity" of the original and restored samples using the L2 measure according to the "Cycle consistency loss" ideology (which ensures the overall similarity of the generated samples to the original samples in GAN) [13]. The final objective function is a combination of all metrics for evaluating the quality of work of critics C_x, C_z and a measure of similarity between the original and restored signal.

$$\min_{\{\varepsilon,\varsigma\}} \max_{\{C_x \in C_x, C_z \in C_z\}} V_X(C_x,\varsigma) + V_Z(C_z,\varepsilon) + V_{L2}(\varepsilon,\varsigma). \quad (1)$$

To create and train a neural network, you can use various standard packages (for example, TensorFlow or PyTorch) that have a high-level API. An example of implementing a TadGAN-like architecture using the TensorFlow package for weight training can be found in the repository [14, 15]. When training this model, five metrics were optimized:

- aeLoss is the root-mean-square deviation between the original and restored time series, i.e. the difference between x and $\varsigma(\varepsilon(x))$;
- cxLoss-binary cross-entropy critique of C_x, which determines the difference between a true Time Series segment and an artificially generated one,
- cx_g_loss-binary cross-entropy, oscillator error $\varsigma(\varepsilon(x))$, which characterizes its inability to "deceive" the critic C_x,
- czloss-binary cross-entropy critique of C_z, which determines the difference between the hidden vector generated by the encoder and white noise, provides similarity of the hidden vector ‡ (x) to a random vector, preventing the model from "memorizing" individual patterns in the source data,
- cz_g_Loss is a binary cross-entropy, an error of the oscillator $\varepsilon(x)$, which characterizes its inability to create hidden vectors similar to random ones, and thus "deceive" the critic C_z.

After training the model, individual segments of the time series under study are reconstructed and the original and reconstructed series are compared, which can be performed using one of the following methods [13]:

- streaming comparison;
- comparison of curve areas in a given area around each sample (area length is a hyperparameter);
- Dynamic Time Warping.

Quality is evaluated using the f1 metric for the binary classification problem, "positive" (null hypothesis): there is an anomaly, "negative" (alternative hypothesis): no anomaly.

Table 2
Quality is evaluated using the F1 metric for the binary classification problem

	The anomaly is predicted by the model	The model predicted the absence of an anomaly
There is an anomaly	TP correctly predicted anomaly	FN there is an anomaly, but it was not found
There is no anomaly	FP predicted an anomaly where it doesn't exist	TN there is no anomaly and the model does not see it

To demonstrate the operation of the method, we use a synthetic (artificially generated) series without anomalies, which is the sum of two sinusoids, the values of which vary in the range from -1 to 1: $y(x)=\frac{1}{2}\sin(x)+\frac{1}{2}\sin(0.8x)$.

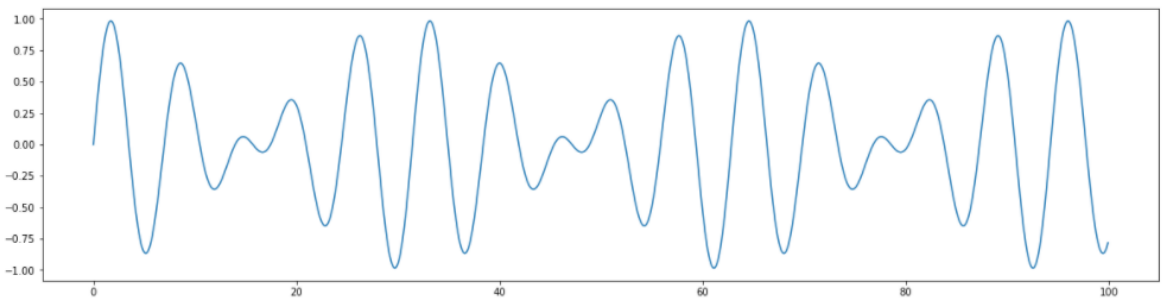

Figure 4: Graph of such a series

It can be seen that the model has quite accurately learned to predict the main patterns in the data. Let's try adding various anomalies to the data and then detect them using the tadgan model. First, let's add a few point anomalies [13].

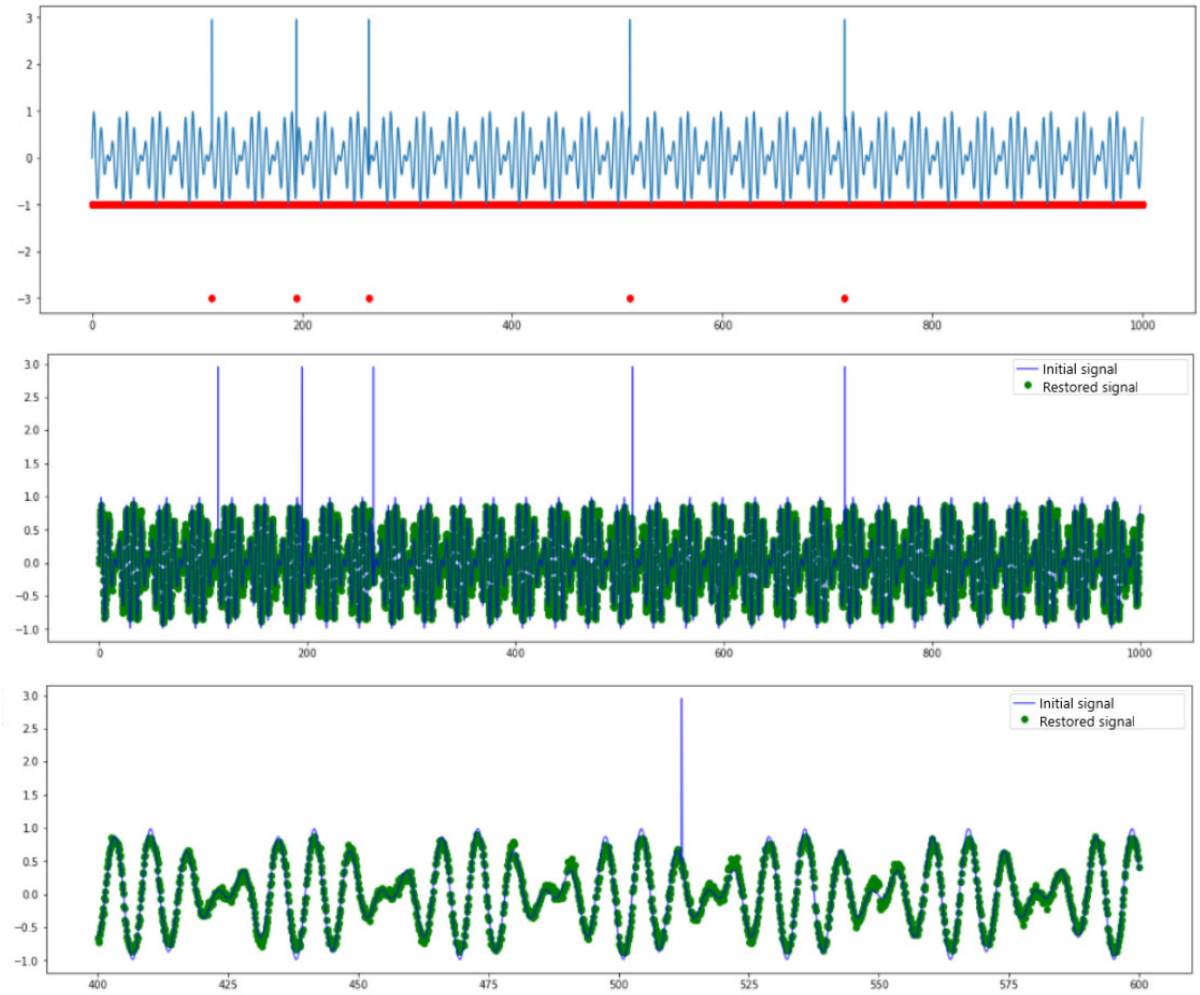

Figure 5: Spot Anomaly Detection using TadGAN

The graph of the original and predicted signals shows that the model cannot restore the “peaks” of anomalous values, which can be used with high accuracy to determine point anomalies. However, in such a situation, the crust of the complex TadGAN model is not obvious-such anomalies can also be detected by estimating the excess of thresholds [13].

Now consider a signal with a different type of anomaly: a periodic signal with an abnormal frequency change. In this case, there is no excess of the threshold: from the point of view of amplitude, all elements of the series are “normal” values, and the anomaly is detected only in the group behavior of several points. In this case, TadGAN also cannot restore the signal (as can be seen in the figure) and can be used as a sign of the presence of a group anomaly [13].

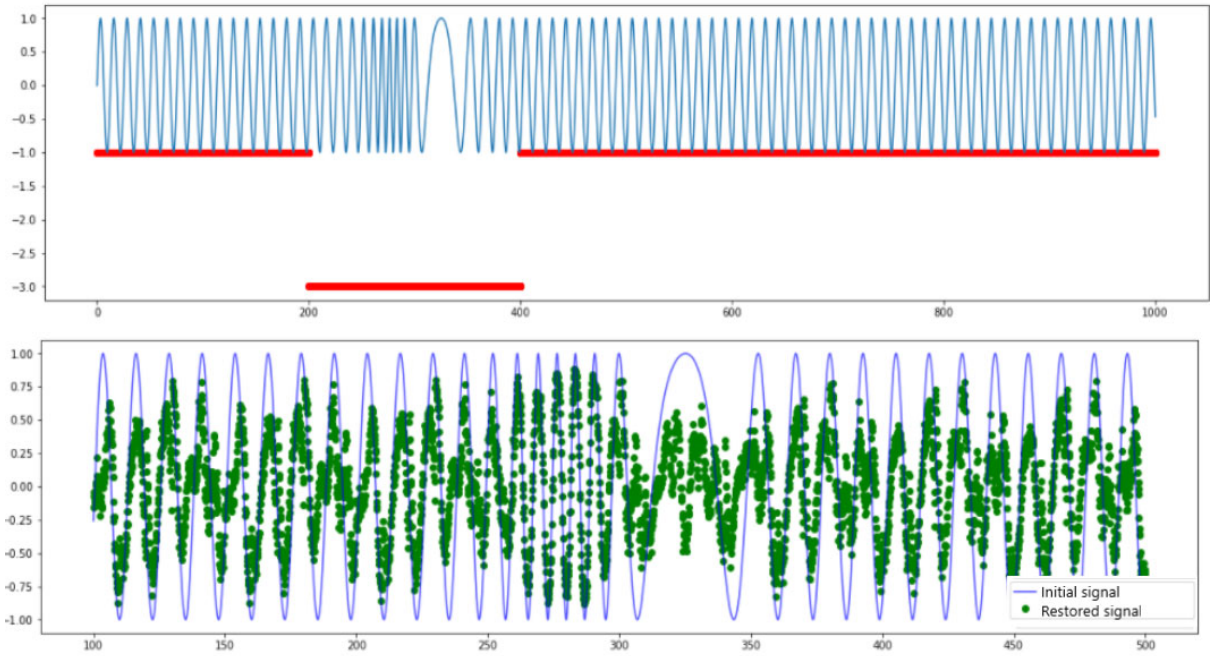

Figure 6: The result of the TadGAN operation on a dataset with an abnormal frequency change.

These two examples illustrate the work of the method. The reader can also try to create their own data sets and test the model's capabilities in various situations.

More complex examples of datasets can be found in the article by the authors of the TadGAN Method [13]. There is also a link to the Orion library, which is developed by MIT specialists, which uses machine learning to recognize rare anomalies in time series, using the unsupported learning approach [14].

However, most scientists agree that each case requires its own method of signal reconstruction and a conductive method of training the model, which significantly slows down the practical implementation.

3. Proposed method

The prototype of our solution is shown in Fig. 7, consists of two separate groups of the same population. The first is a set of models, and the second is a set of data individuals.

Each model is a sequence of encoder and decoder levels that describe the input multidimensional signal [16]. The framework allows you to create an ensemble model using an approach similar to the method based on packaging in packages. A whole ensemble model is a set of sub-models that work with subgroups of input signals. Ensemble models form a set.

The work of the solution begins with generating initial groups of input signals using correlation; the system then updates the models and groups using a genetic algorithm [17-19]. In parallel, genetic operators optimize individual models in each subgroup by making changes to the topology of neural models (for example, model length and layer parameters) [17-19]. The end result of these actions is an ensemble model optimized for anomaly detection. The ensemble model is defined as follows.

A number of papers note that despite different models, it is noticeable that almost all models show a similar set of anomalies, despite changes in their hyperparameters [16]. Of course, the results differed depending on the hyperparameters, but none of the changes significantly affected detection. Because of this, an ensemble model is proposed, based on the simultaneous division of the search space into subgroups and the evolution of models for such subgroups. As a result, the models were able to identify more specific dependencies and relationships between signals [20-22].

Models within each subgroup are optimized by changing their internal structure. The evolution of a single model is carried out in five main stages:

1. clustering the search space [23-25];

2. crossing, mutating, and selecting the best models for individual clusters;
3. syncing solutions;
4. multi-parent crossover [17-19];
5. evaluation of the ensemble solution.

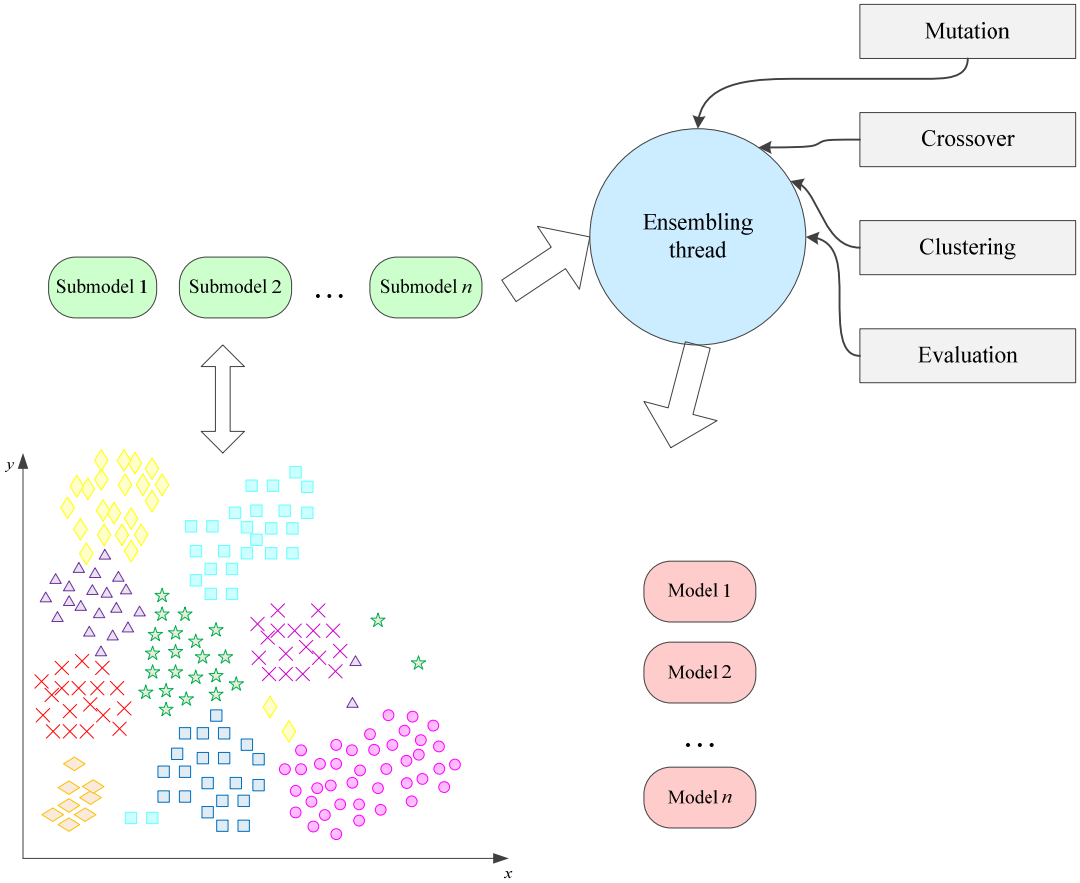

Figure 7: The proposed method

4. Experimental research

For the experimental research of proposed method was be used the following as the training and testing data:

- The Secure Water Treatment (SWaT) Dataset [26], [16]. This dataset contains data gathered from a scaled-down version of a real water treatment plant. The data were collected for 11 d in two modes – 7 d of normal operation of the plant and 4 d during which there were cyber and physical attacks executed;
- The Water Distribution (WADI) Dataset [27], [16]. This dataset contains data from a scaled-down version of a water distribution network in a city. The collected data contain 14 d of normal operation and 2 d during which there were 15 attacks executed.

The general information about datasets present in Table 3.

Table 3
General information about datasets

Datasets	Number of Input Signals	Number of Trainings	Number of Tests	Number of Anomalies
SWAT	51	49,668	44,981	11.97%
WADI-2017	123	1,048,571	172,801	5.99%
WADI-2019	123	784,571	172,801	5.77%

The meta-parameters for neuroevolution synthesis of models demonstrate at Table 4.

Table 4
The meta-parameters for neuroevolution synthesis

Metaparameter	Value
Population size	100
Elite size	5%
Activation function (fitness functions)	hyperbolic tangent
Mutation probability	25%
Crossover type	uniform
Types of mutation	deleting an interneuronal connection removing a neuron adding interneuronal connection adding a neuron changing the activation function
Clustering method	*k*-nearest neighbors
Neighbors number	7

The results of the work present at Table 5.

Table 5
The work results of proposed method

Datasets	Precision, %	Recall, %	f1, %
SWAT	94.41	55.35	0.74
WADI-2017	90.28	70.64	0.82
WADI-2019	89.53	71.47	0.83

5. Discussions of results

The experimental results demonstrate that parallel clustering of data and synthesis of a model based on the processed data using an ensemble system can significantly increase the efficiency of the anomaly detection process. The process of neuroevolution helps to synthesize models and develop them in stages based on updated information about input data, without separating the process of data preprocessing. Tests based on samples of WADI and SWAT data. In both cases, the results of anomaly detection demonstrated satisfactory values, raising the qualitative level of detection. The improvements in the WADI dataset were more significant than in the SWAT dataset because there are more sensors and samples in the WADI dataset than in the SWAT dataset.

The article proved that the neuroevolution approach can have a positive impact on the results. Our future work will focus on further improvements to the method. A combination of different solution topologies can significantly improve the results of the work.

6. Conclusion

At the paper a research and comparative analysis of existing strategies and methods solving the problem of detecting and classifying anomalies were carried out, and a detection method based on neuroevolutionary methods was also proposed. As can be seen from the results of the study, the method of detecting anomalies based on neuroevolution has shown much greater effectiveness. The proposed method has shown itself to be viable and can be improved.

The result of the work is not only a comprehensive study and theoretical justification of the theory associated with the analysis of time series, but also the proposed solution. His work consists of two

stages: the separation of the search space and the synthesis of models. At the training stage, the method processes and separates data about the behavior of the system. In the synthesis mode, the method gradually adjusts the models in order to get the final solution from them in the future. The resulting model is synthesized using uniform crossing, which makes it possible to increase the size of the parent pool from two individuals to a much larger number.

7. Acknowledgements

The work was carried out with the support of the state budget research projects of the state budget of the National University "Zaporozhzhia Polytechnic" “Development of methods and tools for analysis and prediction of dynamic behavior of nonlinear objects” (state registration number 0121U107499) and “Intelligent methods and tools for diagnosing and predicting the state of complex objects” (state registration number 0122U000972).

8. References

[1] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: A survey, ACM Computing Surveys, 41(3) (2009) 1-58. doi: 10.1145/1541880.1541882.
[2] What Is Data Analytics?, 2020. URL: https://www.intel.com/content/www/us/en/artificial-intelligence/what-is-data-analytics.html#:~:text=Data%20analytics%20is%20the%20process,data%20for%20practically%20any%20purpose
[3] Cycle Consistency Loss, 2017. URL: https://paperswithcode.com/method/cycle-consistency-loss
[4] Search for anomalies in time series based on the estimation of their parameters, 2021. URL: https://openarchive.nure.ua/items/7c8e2e76-10fa-4044-907b-e51d05bd7cbf [In Ukrainian]
[5] J. Ma, S. Perkins, Time-series novelty detection using one-class support vector machines, in: Proceedings of the International Joint Conference on Neural Networks, Portland, OR, IEEE, 2003, pp. 1741-1745. doi: 10.1109/IJCNN.2003.1223670.
[6] MIT – Data to AI Lab, Time series anomaly detection — in the era of deep learning, 2020. URL: https://medium.com/mit-data-to-ai-lab/time-series-anomaly-detection-in-the-era-of-deep-learning-dccb2fb58fd
[7] Anomaly Detection in Time Series, 2023. URL: https://neptune.ai/blog/anomaly-detection-in-time-series
[8] A. Bhattacharya, Effective Approaches for Time Series Anomaly Detection, 2020. URL: https://towardsdatascience.com/effective-approaches-for-time-series-anomaly-detection-9485b40077f1
[9] S. Schmidl, P. Wenig, T. Papenbrock, Anomaly detection in time series: a comprehensive evaluation, in: Proceedings of the Proceedings of the VLDB Endowment, Vol. 15 (9), Sydney, ACM, 2022, pp. 1779–1797. doi: 10.14778/3538598.3538602.
[10] Anomaly detection and forecasting in Azure Data Explorer, 2023. URL: https://learn.microsoft.com/en-us/azure/data-explorer/kusto/query/anomaly-detection
[11] B. Artley, Time Series Forecasting with ARIMA , SARIMA and SARIMAX, 2022. URL: https://towardsdatascience.com/time-series-forecasting-with-arima-sarima-and-sarimax-ee61099e78f6
[12] M. K. Mandal, Implementing PCA, Feedforward and Convolutional Autoencoders and using it for Image Reconstruction, Retrieval & Compression, 2018. URL: https://blog.manash.io/implementing-pca-feedforward-and-convolutional-autoencoders-and-using-it-for-image-reconstruction-8ee44198ea55
[13] A. Geiger, D. Liu, S. Alnegheimish, A. Cuesta-Infante, K. Veeramachaneni, TadGAN: Time Series Anomaly Detection Using Generative Adversarial Networks, in: Proceedings of the IEEE International Conference on Big Data (Big Data), IEEE, 2020. doi: 10.1109/BigData50022.2020.9378139.
[14] TadGAN, 2021. URL: https://github.com/gusty1g/TadGAN

[15] Examples. TadGAN, 2022. URL: https://github.com/CyberLympha/Examples/tree/main/%D0%A0%D0%B0%D0%B7%D0%B1%D0%BE%D1%80%20%D1%81%D1%82%D0%B0%D1%82%D0%B5%D0%B9/TadGAN
[16] K. Faber, M. Pietron, D. Zurek, Ensemble Neuroevolution-Based Approach for Multivariate Time Series Anomaly Detection, Entropy, 23 (2021). doi: 10.3390/e23111466.
[17] S. Leoshchenko, S. Subbotin, A. Oliinyk, V. Lytvyn, M. Ilyashenko, Smart crossover mechanism for parallel neuroevolution method of medical diagnostic models synthesis, in: Proceedings of the Third International Workshop on Computer Modeling and Intelligent Systems, CMIS-2020, CEUR-WS, Zaporizhzhia, 2020, pp. 57-69.
[18] S. Leoshchenko, A. Oliinyk, S. Subbotin, Adaptive Mechanisms for Parallelization of the Genetic Method of Neural Network Synthesis, in: Proceedings of the 10th International Conference on Advanced Computer Information Technologies, ACIT 2020, Deggendorf, Ternopil, IEEE, 2020, oo. 446-450. doi: 10.1109/ACIT49673.2020.9208905.
[19] S. Leoshchenko, A. Oliinyk, S. Subbotin, T. Zaiko, S. Shylo, V. Lytvyn, Sequencing for encoding in neuroevolutionary synthesis of neural network models for medical diagnosis , in: Proceedings of the 3rd International Conference on Informatics & Data-Driven Medicine, IDDM 2020, Växjö, Lviv, CEUR-WS, 2020, pp. 62-71.
[20] J.A.J. Alsayaydeh, Irianto, M. Zainon, H. Baskaran, S. G. Herawan, Intelligent Interfaces for Assisting Blind People using Object Recognition Methods, International Journal of Advanced Computer Science and Applications(IJACSA), 13(5) (2022) 734-741. doi: 10.14569/IJACSA.2022.0130584.
[21] J.A.J. Alsayaydeh, Irianto, A. Aziz, C.K. Xin, A. K. M. Zakir Hossain, S.G. Herawan, Face Recognition System Design and Implementation using Neural Networks, International Journal of Advanced Computer Science and Applications(IJACSA), 13(6) (2022) 519-526. doi: 10.14569/IJACSA.2022.0130663.
[22] V. Shkarupylo, I. Blinov, A. Chemeris, J.A.J. Alsayaydeh, A. Oliinyk, Iterative approach to TLC model checker application, in: Proceedings of the 2nd KhPI Week on Advanced Technology, KhPI Week 2021, IEEE, Kyiv, 2021, pp. 283-287. doi: 10.1109/KhPIWeek53812.2021.9570055.
[23] R. Pawar, k-NN based Time Series Classification, 2021. URL: https://towardsdatascience.com/k-nn-based-time-series-classification-e5d761d01ea2
[24] S. Tajmouati, B. Wahbi, A. Bedoui, A. Abarda, M. Dakkon, Applying k-nearest neighbors to time series forecasting : two new approaches, 2021. URL: https://arxiv.org/abs/2103.14200
[25] F. Martínez, M.P. Frías, M. Pérez, A. J. Rivera Rivas, A methodology for applying k-nearest neighbor to time series forecasting, Artificial Intelligence Review, 52 (2019) 2019–2037. doi: 10.1007/s10462-017-9593-z.
[26] A. P. Mathur, N. O. Tippenhauer, SWaT: a water treatment testbed for research and training on ICS security, in: Proceedings of the International Workshop on Cyber-physical Systems for Smart Water Networks, CySWater, Vienna, IEEE, 2016, pp. 31-36, doi: 10.1109/CySWater.2016.7469060.
[27] C. M. Ahmed, V. R. Palleti, A. P. Mathur, WADI: a water distribution testbed for research in the design of secure cyber physical systems, in: Proceedings of the 3rd International Workshop on Cyber-Physical Systems for Smart Water Networks, CySWATER '17, Pittsburgh, PA, ACM, 2017, pp. 25-28. doi: 10.1145/3055366.3055375.