Rail Vehicles/Pojazdy Szynowe

www.railvehicles.eu

Selected aspects of the diagnostic process in rail transport

Marcin **Chrzan** [a, *] , Tomasz **Ciszewski** [a] , Waldemar **Nowakowski** [a]

[a] *Faculty of Transport, Electrical Engineering and Computer Science, Radom University, Radom, Poland*

The article presents selected results of research in the area of reducing the risk of defects in railway infrastructure and traffic control devices. The first part of the article will discuss selected topics used in a defectoscope car for automated ultrasonic rail inspections related to the identification of joints and flaws. A method based on the identification of joints and flaws using a neural network will be presented. The second part of the article will cover research on the automatic collection of diagnostic data from railway traffic control devices. The solutions presented concern a simulator of railway traffic control device malfunctions, from which data is extracted to populate a database of malfunctions and then used in the inference process. The article will present partial results of research on both systems.

1. Introduction

For many years, ultrasonic testing of rails has been one of the non-destructive testing methods used to detect internal material defects such as cracks or inclusions [1–3, 6, 7]. From the point of view of rail transportation, these tests are extremely important for ensuring rail safety due to their complexity. Automatic methods of inspecting rails have become increasingly popular in recent years due to their ability to monitor large sections of rail track quickly and efficiently [11, 13]. They also significantly assist workers responsible for maintaining and inspecting rail infrastructure. Automated rail inspection systems not only increase the safety of rail traffic, but also significantly reduce the costs associated with the need for manual track inspection. Combined with modern data analysis technologies such as artificial intelligence and machine learning, they can also predict potential problems before they become a serious threat, as will be shown later in the article. The article describes the application of signal processing using neural networks to recognize patterns in measurement data from

a defectoscope wagon examining railroad rails using ultrasonic methods. This has also been described in publications [15, 16]. The wagon's measurement apparatus uses digital signal processing, enabling the recording of a large data volume. This creates the need to design good automatic evaluation procedures. By their nature, they must have the character of statistical signal estymation. Neural networks enable the construction of such procedures. On the other hand, the introduction of automatic diagnostic systems into rail traffic control equipment significantly improves operational efficiency, reduces the risk of failures and associated delays, and increases overall rail traffic safety [4, 10, 12, 14, 17]. As technology advances, such systems are becoming more sophisticated and capable of independently predicting and responding can be said in real time to potential problems. Modern automatic diagnostic systems are essential for monitoring and detecting any malfunctions in these devices. Therefore, in the second part of the article will be presented one of the elements of the research work SADEK system, which is an important element of collecting data on emergency situations.

---

2. Methodology of data analysis in the defectoscope wagon

The results of the research presented here have been implemented in the design of a defectoscope wagon operating on the PKP network (Fig. 1).

Fig. 1. Defectoscope wagon [18]

2.1. Pre-processing of data used, methodology of experiments

The results of measurements made by the defectoscope wagon have been used for the experiments. A selection was made of the part of the measurements that had already been evaluated by the operator, coming from a reference section in which artificial models of typical defects found in railroad tracks were created. This section is used to calibrate the apparatus before carrying out the measurements. Detecting artificial defects in the reference section is a guarantee of detecting real defects on the track. In accordance with the same philosophy, it was considered that the correct recognition of patterns in the measurement signals from the calibration section proves the suitability of this technology for real measurement conditions. The available records of calibration tests included test runs conducted under various conditions, in particular, with different levels of head wear, when driving at different speeds and also with different road pulser constants (and therefore with different measurement flux densities). The arrangement of the heads is shown in Fig. 2.
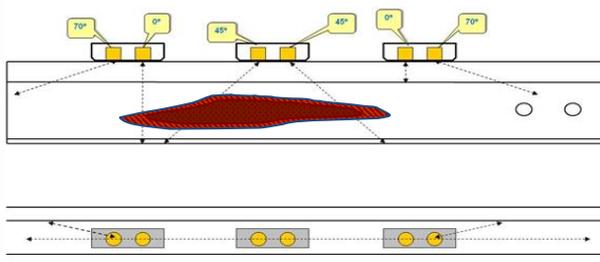

Fig. 2. Arrangement of measuring heads

As mentioned, this data is processed by the operator and equipped with a classification that determines whether the corresponding portion of the signal represents an image of a dangerous defect, a defect to be observed or a connector. The expert knowledge necessary to perform supervised network learning is therefore available. Attempts have been made to use different types of networks, including a simple layered network implementing static pattern classification. In order to prepare the appropriate data sets (training and testing), additional preprocessing tools have been prepared). The functional scope of these tools includes:

- selection of source measurements to be included in the dataset (measurement time, instrument channel)
- determination of the types of patterns to be placed in the dataset (non-safety defect, defect to be observed, connector)
- determining the subset of measurements to be placed in the dataset (e.g., only defect amplitude information, or also additionally the results of defect position measurements, etc.)
- windowing of measurement signals by specifying the length of the window and also the minimum length of the fragment of the original pattern, covered by the window
- adding noise to the signal with a Gaussian distribution, with a preset mean value and variance
- generation of data sets (including sets containing the expected values of the output of the classifier in a format acceptable to the software implementing neural network training)
- visual inspection of the generated datasets, to check the correctness of the operation carried out.

The need to carry out the various operations mentioned above resulted from the assumed methodology of the experiments.

As a task to be carried out, it was accepted to recognize the pattern of the measurement signal formed by the screw rail connector, in particular to distinguish it from other signal patterns, such as the defect pattern. It was limited to signals coming from normal heads (two channels of the apparatus). The connector signal also appears in channels with 45° angle heads, but the pattern recorded by these heads is different from the signal pattern from normal heads, so we were limited to the latter only. Success in recognizing the joint pattern in normal heads means that the neural network is able to perform an analogous operation for another signal pattern, also for the signal coming from another type of heads. In addition, the task of correctly identifying a connector pattern includes the problem of distinguishing it from the entire set of patterns

induced in the signal by defects. Thus, we have a situation anyway, where it is necessary to discriminate between whole families of patterns.

In addition to experiments involving recognition on the basis of the full information: amplitude and position of the defect, it was decided to make much more difficult attempts to discriminate only on the basis of the amplitude signal. The position of the defect is a very characteristic parameter for the connector – all the holes lie at one depth. If a neural network can create a classifier, using mainly position information for discrimination, this only speaks in favor of this method. It is good when a flexible procedure capable of creating models of high complexity constructs a simple model, if it has the ability to do so. However, extending the conclusions of the experiments to other, less obvious cases of signal patterns required testing the method under more difficult conditions. Distinguishing a connector signal from a defect signal without position information can be difficult even for a human. So this is a very demanding test for an artificial inference system. Besides, an economic aspect is included. Reducing the data stream needed to perform the necessary adjudication always entails savings. In a related line, an additional complication is assumed – the omission of information from the way counter in the classification. In other words, information about the exact spatial relationships between individual pattern samples is omitted from the training data. Only the natural ordering of the samples is saved, that is, the information on neighborhood and succession. Note that the wagon apparatus uses conditional registration of measurements – only those measurements for which the amplitude exceeds the monitor's discrimination threshold are registered. This means that adjacent samples of measurements can represent arbitrarily distant track points. Referring again to human intuition, it should be said that this makes it much more difficult to recognize the joint pattern. This is illustrated in Fig. 2, which shows the joint patterns at full registration and at conditional registration. For comparison, the defect pattern at conditional registration is also shown (Fig. 3). It can be seen that apparently, a neural network capable of discriminating patterns must recognize subtle relationships between samples due to their proximity and succession.

To evaluate the quality of the created classifier, generalization estimation was adopted. Good generalization is the real goal of the neural network learning process. A good classifier should not so much be faithful to the training data – this can almost always be achieved by increasing the complexity of the model accordingly. The real test is good classification accuracy for data that did not participate in the learning process. On the ground of machine learning, generalization error is defined as the expected value of the error on all possible data sets of a certain size and the same probability density distribution as the entire input population. In practice, this estimation is carried out by checking the classifier on an independent test set, provided that it is sufficiently numerous. The most common rule of thumb is to separate from the set of all data 30% of the elements for testing and 70% of the elements for the training set. The quality of the constructed classifier is evaluated as the error on the testing set.
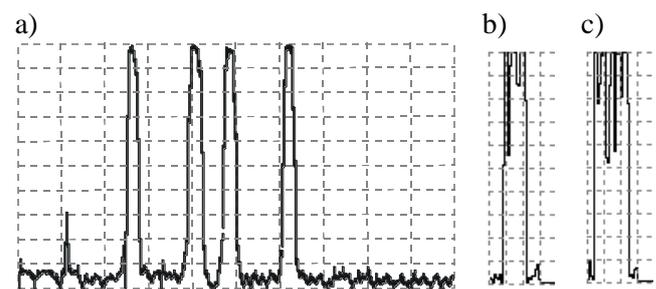

Fig. 3. Signal patterns a) connectors with preservation of spatial relations (full registration), b) connectors with conditional registration, c) dangerous defect with conditional registration. All characteristics show amplitude as a function of time

Neuro-Solutions 4.0 software, which uses an object-oriented approach to neural network modeling, was used for the simulation. This package is a very complex system in which most neural network architectures can be simulated.

2.2. Recognizing the joint using static reference classification

In this experiment, an attempt was made to teach a layered neural network for connector signal recognition based on 30-sample signal fragments, including images of connectors and defects. The window lengths were selected based on a histogram of the lengths of examples derived from the raw data, generated using one of the tool programs described earlier from measurements collected on a sample track with artificial defects. These histograms are shown in Fig. 4.

Most of the connector patterns range in length from 8 to 29 samples. By the way, this scatter shows how varied the connectors are recorded by the ultrasonic apparatus (change in the quality of coupling of the heads to the rail). All connectors with lengths in this range were accepted for windowing. As the minimum length of the pattern in the window, the value of 8 was taken. From 129 examples of connectors from the original data, 5128 windows were thus created, for which the classifier should give the answer "Connector" (encoded in the file with information on expected

values as the value 1). As for the patterns of defects, most of the histogram mass also falls within a window of length 30, but in this case the minimum accepted defect length was taken to be 1. The windowing thus resulted in 21749 examples for which the classifier should give a "Not Connected" response (encoded as zero). Each sample in the windowing is a two-dimensional vector whose components describe the amplitude expressed on one byte and the position of the defect expressed on 1 byte.
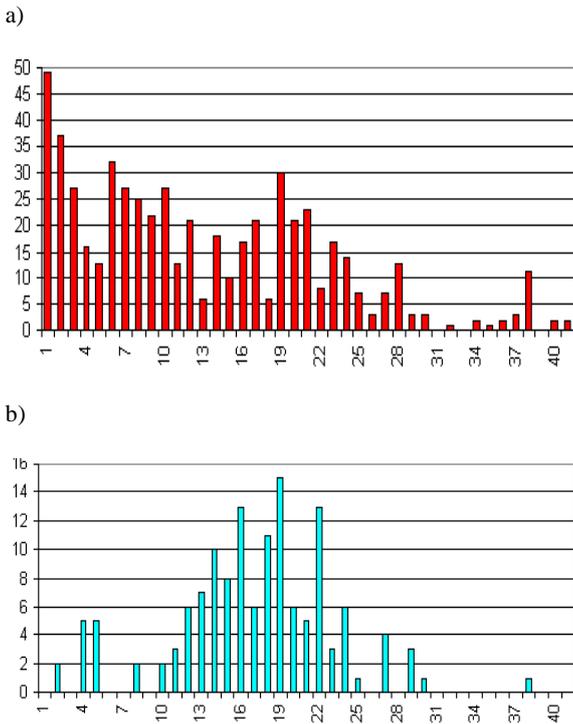
a)

b)

Fig. 4. Histograms of the length of defect and joint patterns in the raw data: a) as a function of the defect pattern, b) as a function of defect length

The data sets were split in a 70–30% ratio, resulting in a total of 18814 examples in the training set and 8063 examples in the testing set. A layered architecture with one hidden layer with five neurons was adopted. A learning algorithm called Delta-Bar-Delta – a variation of the back-propagation algorithm with complex learning rate adaptation heuristics – was used.

Figure 5 shows an excerpt from the Neurosolutions program window showing the graphical design of the network and windows displaying the status of the learning process after 791 iterations. The largest classification error (both joints and defects) is on the order of 10%, on both the training and test sets. Given that the accuracy of ultrasonic testing, especially of such a nature as wagon testing, is not exorbitant (a sign of which is, for example, the scatter in the length of the connector signal patterns), this is a good result.

Achieving such a result required 791 learning epochs. Continuing to teach from this point continues to reduce the error on the training set, but the error on the test set begins to increase, a clear sign of progressive overtraining of the network.
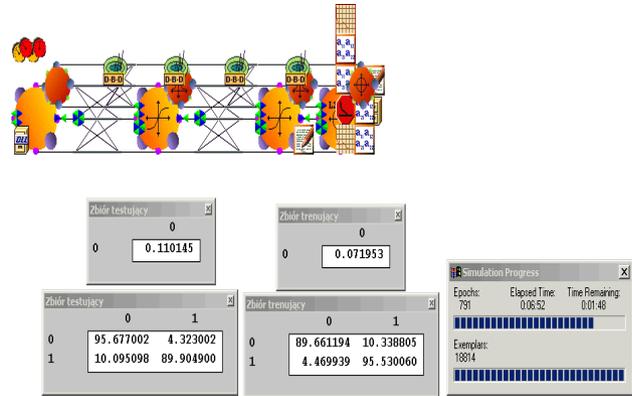

Fig. 5. An excerpt from the screen of the Neurosolutions program, showing the designed architecture of the network, as well as windows showing the averaged error on the training and testing sets, the classification matrices for both sets, and a window showing the learning process

The experiment was repeated for a reduced architecture, including three neurons in the hidden layer. The classification matrix for this trial is given in Table 1.

Table 1. Classification matrix for a network containing 3 neurons in the hidden layer; learning covered 672 epochs

	Training collection		Training collection	
	0	1	0	1
0	89.381767	10.618233	95.595436	4.404568
1	4.463670	95.536331	10.095098	89.904900

The network learned to correctly recognize connector patterns as easily as in the previous case.

Next, an attempt was made to classify based only on the amplitude signal. Initial tests showed that the learning process for the same architectures as in previous experiments was very slow and did not produce good results. Therefore, an architecture capable of producing a strongly nonlinear processing function was used – a network with two hidden layers, with 15 neurons in the first hidden layer and 5 in the second. The computational effort involved in learning such a network is much greater. Nevertheless, the results achieved are not much better than for small networks. Table 2 gives the classification matrix for this case after conducting 12600 learning epochs.

Table 2. Classification matrix for networks with two hidden layers (15 and 5 neurons), learned for 12600 epochs

	Training collection		Training collection	
	0	1	0	1
0	35.197552	64.802452	39.415585	60.584415
1	5.243790	94.756210	11.096434	88.903564

From Table 2, it can be seen that the network tends to classify everything as a joint. It should be noted that a classification efficiency of 40% (column sum of 0) is worse than a blind hit, which gives an expected error value of 50% with dichotomous discrimination. Thus, layered networks performing static pattern classification easily perform pattern recognition with full amplitude and position information, but completely fail to deal with the issue of recognition based on amplitude information alone.

3. SADEK system (System for performance data acquisition and reliability analysis of railway automatic systems)

The Faculty of Control Systems and Electronics at the University of Radom in Radom, thanks to the cooperation with Alstom ZWUS Sp. z o.o. Poland S.A. Katowice, Zakłady Automatyki KOMBUD from Radom and Scheidt & Bachmann Polska Sp. z o.o. company enriched its laboratory base with modern and unique in Europe laboratories designated for technical and functional testing of essential systems and devices of railroad traffic control. They gathered models of basic signaling systems and devices, currently produced by the above-mentioned companies, used on modernized railroad lines. All laboratory stations correspond to real railroad traffic control systems in operation on Polish and other railroads. The extensive infrastructure made the SADEK project possible.
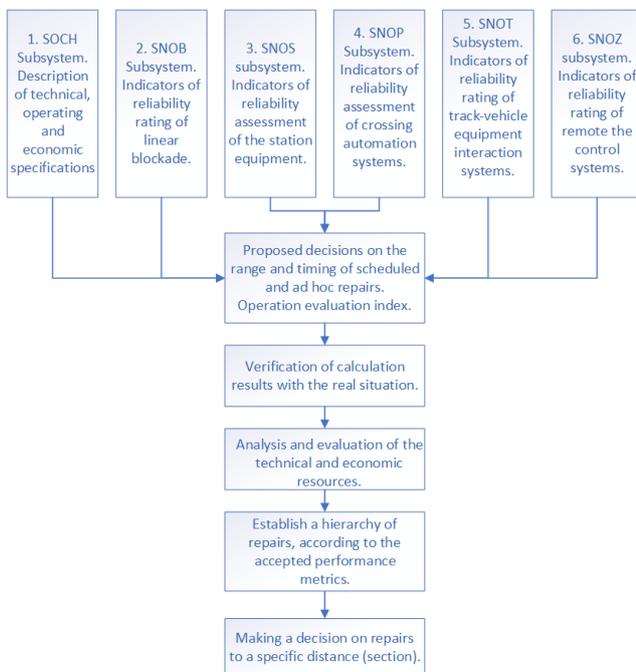

Fig. 6. Block diagram of SADEK system

3.1. Implementation of the fault simulator for the SADEK system

Due to the inability to obtain operational data, a damage simulator based on manual fault generation (Fig. 7) and an automaton described below was realized as part of one of the project stages.

Fig. 7. Manual SSP system fault simulator

The main task carried out by the automatic fault simulator is to generate random events according to the assumed distribution, resulting from the analysis of real data carried out earlier. In addition, it is envisaged that the simulator will cooperate with the rail systems using the proposed data exchange protocol [5, 8]. Therefore, the simulator software for damage simulator of railway equipment consists of two separate tools, namely: "Damage Simulator–Server" and "Damage Simulator–Client". It was assumed that the server can acquire data from multiple clients using TCP/IP protocol. To ensure security, the information exchanged between the client and the server is encrypted using AES block cipher. In addition, each component of this system has the ability to edit data manually. The structure of the fault simulator system is shown in Fig. 8.
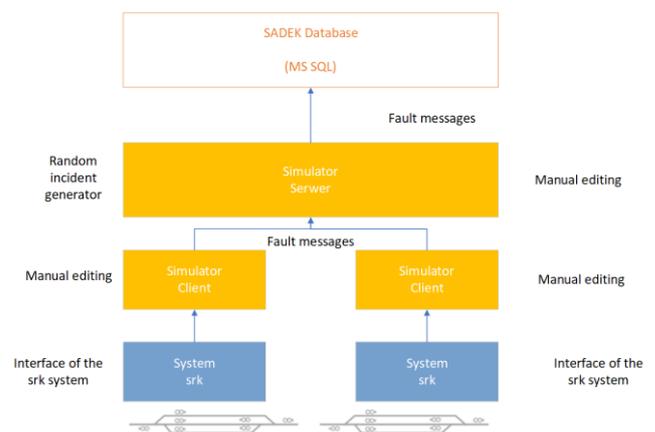

Fig. 8. Block structure of the event simulation system (own development)

3.2. Software "Fault Simulator–Server"

The "Damage Simulator–Server" software, called the server for short, is the main component of the damage simulation system. Parameterization is the first activity to be carried out after starting the server. This activity includes, among other things, parameterization of access to the MS SQL database, including definition:

– provider
– user
– password
– database server IP address
– database server port
– database name.

Other parameters that need to be defined are the TCP/IP server port, information about sending "echoes" to clients when messages are received, and encryption parameters, including password, key length and encryption mode. Parameters related to data encryption must match in the client and server. After parameterization, select the "Apply" button. The parameters will be saved in the encrypted file and you will not need to set them when you run the software again. An example screen with the "Parameters" tab active is shown in Fig. 9.
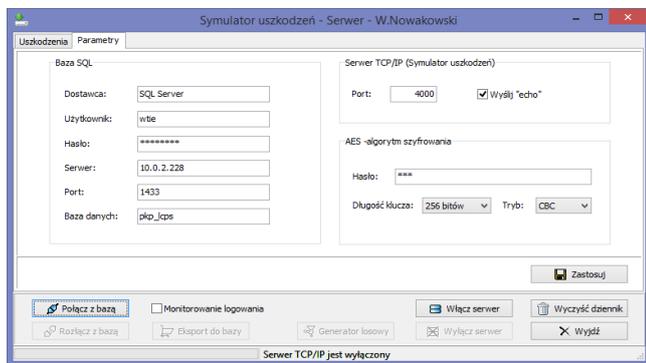

Fig. 9. Parameterization of the fault server (own development)

The main task of the server is to generate random events informing about failures of srk systems. To do this, select the "Enable server" button. As a result of performing this action, an empty table containing the following fields will appear in the main window of the application:

– subsystem
– device
– device_type
– type_element
– failure_date
– date_repair
– time_life
– activity.

The user is notified of the current status of the simulator with a message in the event log: "The TCP/IP server has been enabled" and the "Disable server" button is activated (Fig. 10).
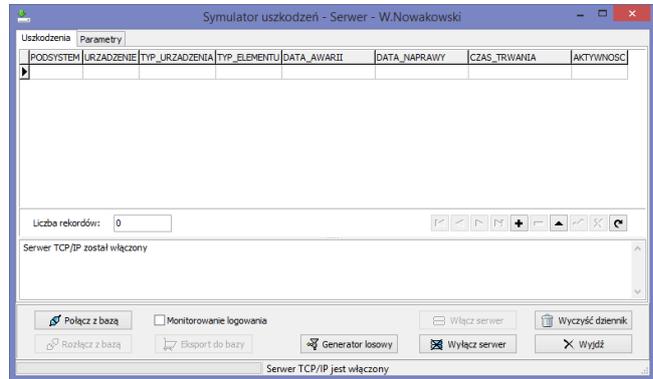

Fig. 10. Activation of the server (own development)

Once the server is activated, you can proceed to generate faults by selecting the "Random Generator" button. The generator window will then be displayed (Fig. 11).
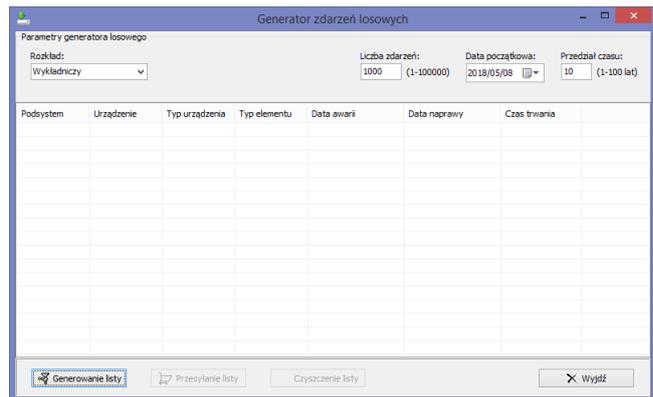

Fig. 11. Window Damage generator – exponential distribution (own development)

The user should set the following parameters:

– type of distribution (exponential, normal (Gaussian), log-normal)
– number of events (in the range from 1 to 100,000)
– starting date of the period
– time interval (in years from 1 to 100).

If a normal (Gaussian) or log-normal distribution is selected, the following can be changed: mean value and standard deviation in the range from 1–24 h and from 1–60 min (Fig. 12).

Then, after selecting the "Generate List" button, a list containing the defects will be generated, with their durations as a function described according to an assumed distribution (Fig. 13). For this purpose, the AMRandom library (by Alan Miller's), distributed on an "open source" basis, was used.
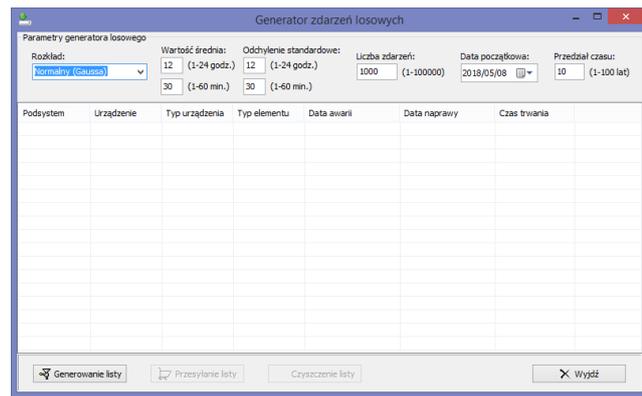
Fig. 12.Fault generator window – normal (Gaussian) distribution (own development)
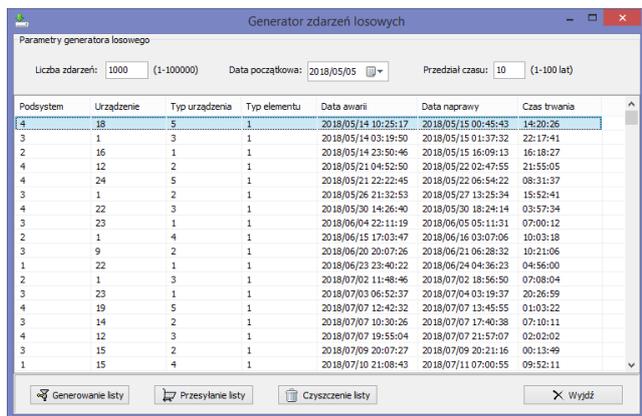

Fig. 13. List of generated random events (own study)

The data generated in this way can be transferred to the main list of the defect simulator (Fig. 14) by selecting the "Transfer list" button.
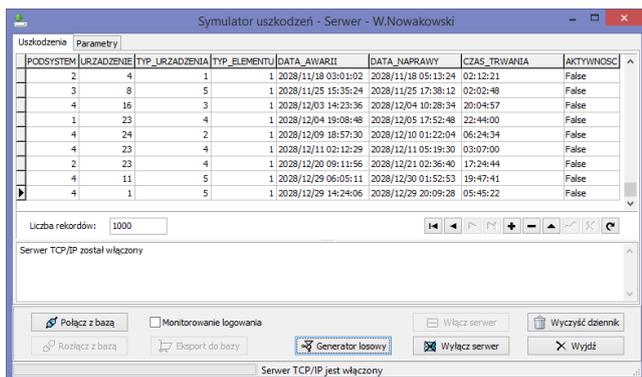

Fig. 14. List of generated random events in the main window of the simulator (own development)

Another functionality of the "Fault Simulator–Server" software is to mediate the transfer of information about events to the SADEK system database, generated by individual clients (railway systems). After activating the server (which uses the TCP/IP protocol for communication), clients can connect to it, and they can start sending messages about the failure

of srk devices. This information updates the list of faults in the application's main window, with the list records sorted by the "date_failure" field (Fig. 15). The user is given the current number of records that are on the list of faults. In addition, a group of buttons is provided in the server software that allows changing the active item in the list, manually adding, deleting and modifying the indicated records.
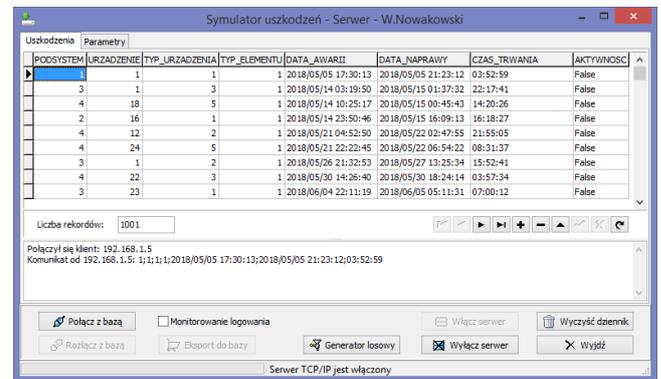

Fig. 15. Correct defect information obtained from the customer (own study)

If the server receives incorrect data from the client, resulting, for example, from incorrect failure and repair dates, such records are displayed in red (Fig. 16). Such records must be deleted or modified before exporting the data to the database. The user has an ongoing update of the number of records that are in the defect list.
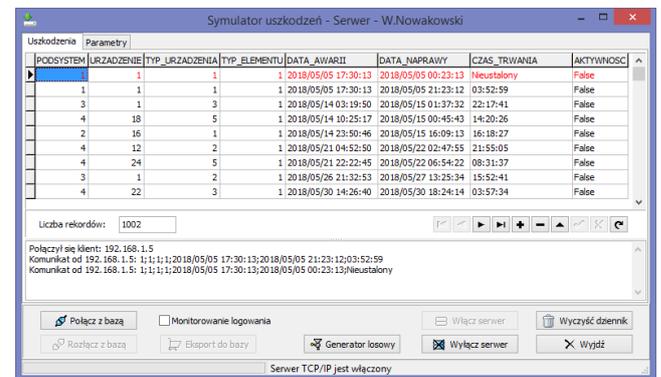

Fig. 16. Incorrect defect information obtained from the customer (own study)

In order to transfer the records contained in the defect table, a connection to the MS SQL database of the SADEK system is required. The connection can be made without logging monitoring, based on the data set during parameterization, or with logging monitoring, during which the user can edit the logging parameters.

After exporting data to the database, the fault list is cleared.If the user does not plan to perform another

data export, he should disconnect from the SADEK database, about which he will be informed by an appropriate message.

At any time the user can clear the "Log" by selecting the "Clear Log" button.

Figure 17 shows a table of the SADEK database with sample records exported by the server.

Fig. 17. Table with exported records to the SADEK database (own compilation)

3.3. Software "Fault Simulator–Client"

The "Fault Simulator–Client" software, hereafter referred to as the client, provides data exchange between the srk system and the simulator. For this purpose, it was assumed that each srk system will work with a separate client.

The first activity to be performed after launching the client is parameterization. The user must specify the IP address and port of the fault simulator server, the port of the srk system interface server, as well as parameters related to encryption, including: password, key size and encryption mode. The parameters can be changed only in the state of no connectivity with the server. Establishing a connection will block parameter editing. After parameterization, select the "Apply" button. Then the parameters will be remembered by the client software, and thus you will not need to set them when you run it again. The parameters are stored with an encrypted file, which further increases the security of this simulation environment. An example of a post-parameterization screen for the client is shown in Fig. 18.

Once the parameterization is done, you can proceed to try to establish a connection with the fault simulator server. To do this, select the "Connect" button. After connecting to the server, the "Damage" tab is activated, and the message "Connected to server: ..." appears in the "Log" window. In addition, the "Connect" button becomes inactive, while the "Disconnect" and "Send" buttons become active. The next step is to activate the interface of the srk system, which is carried out with the "Enable interface" button. As a result of these actions, the client is ready to interact with the traffic control system (Fig. 19).
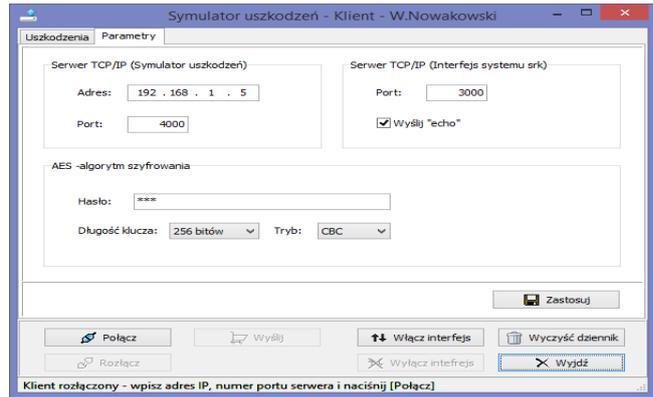

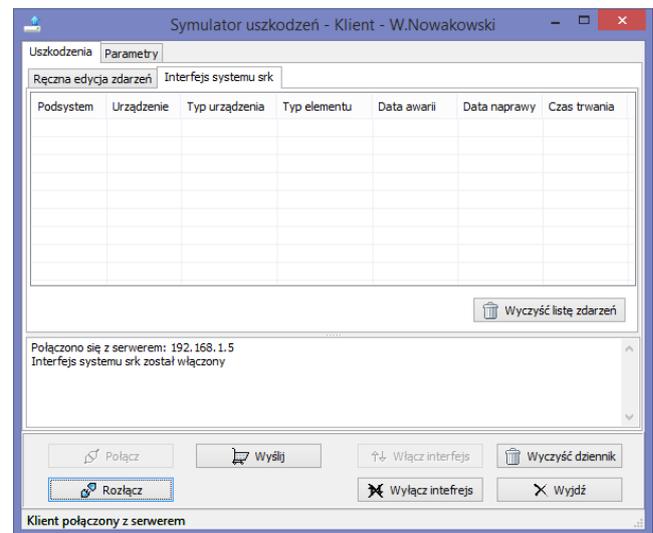Fig. 18. View of the "Parameters" tab of the client software (own development)

Fig. 19. View of the client's main window with the client software's "srk system interface" tab active (own development)

All fault messages received from the srk system are checked by the client software and automatically routed to the server (Fig. 20).
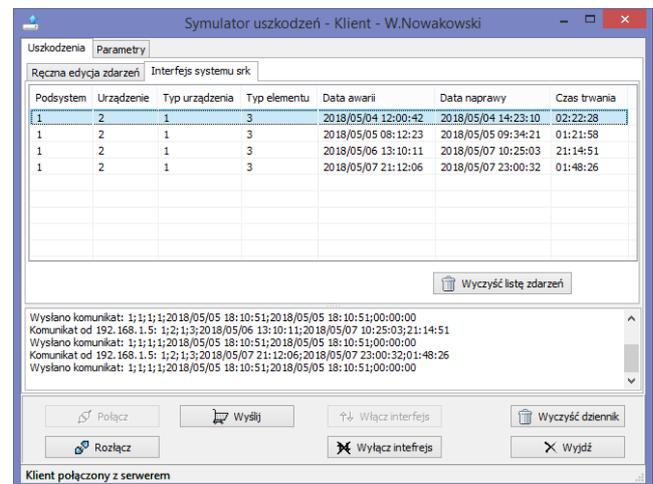

Fig. 20. Customer window with a list of defects received from the srk system (own development)

The manual simulation of events is carried out as a result of handling the fields located on the "Manual editing of events" tab. The user can select from a drop-down list:
– subsystem
– device
– device type
– item type.

Then he must indicate the failure date and repair date. These fields can be changed by editing or by selection. Incorrect indication of the dates leads to the message "Repair date earlier than the failure date". The last field, "Failure duration," is calculated automatically as a result of comparing the failure date and repair date. Any change in simulation data leads to an update of the message sent to the server, the form of which can be observed in the "Explicit message" field. The result of an example simulation is shown in Fig. 21.
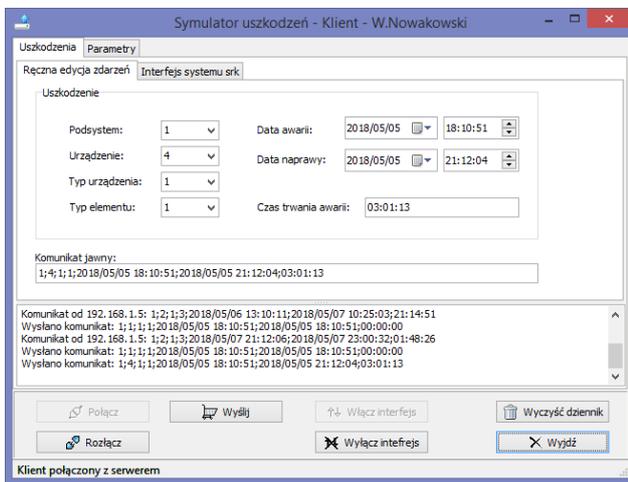

Fig. 21. View of the tab "Manual editing of events" (own elaboration)

After editing, the user can send the message to the server by selecting the "Send" button. This option is available only if the client is connected to the server. If the server has the option set to send "echo", the user will receive back the message: "Sent message ..." recorded in the "Log" window.

4. Conclusions

As a result of the conducted experiments, it has been demonstrated that adaptive non-linear signal processing schemes, such as neural networks, are well-suited for processing measurement data. The networks exhibited satisfactory classification accuracy. Unfortunately, in the case of the examined approach (static pattern recognition), the possibility of reducing the data necessary for making decisions specific to the measurement process, which would reduce the size of the data stream recorded by the system, did not become apparent. Perhaps better results could be achieved by applying dynamic classification methods and using neural networks with memory architecture. The research results were used in the defectoscope car and further developed in subsequent solutions [9].

The presented research pertains to the recognition of only two classes of patterns, but according to the authors, with a sufficient amount of classified training data, they could be applied in a system that recognizes multiple classes.

In the case of the SADEK project, a prototype system for collecting diagnostic data along with their processing using inference and prediction software for individual railway traffic control systems was obtained. The solution presented in the article was one of the components of the complex data collection subsystem. Through the implementation of physical and software solutions of a simulator for the malfunction of railway traffic control devices, it was possible to create malfunction schemes necessary to populate a central database of malfunctions. Such a solution allows for the creation of a wide-ranging database of malfunctions required in the inference and prediction process. The proposed solution serves as the basis for creating an integrated diagnostic system for all railway traffic control devices used by the Polish State Railways.

Nomenclature

IP	Internet Protocol Address
MS-SQL	Microsoft SQL Server
SADEK	System for performance data acquisition and reliability analysis of railway automatic systems

srk	traffic control system
SSP	Automatic Crossing Signals
TCP	Transmission Control Protocol

Bibliography

[1] Alahakoon S, Sun YQ, Spiryagin M, Cole C. Rail flaw detection technologies for safer, reliable transportation: a review. J Dyn Syst-T ASME. 2018;140(2):020801. https://doi.org/10.1115/1.4037295

[2] Bojarczak P, Nowakowski W. Application of deep learning networks to segmentation of surface of railway tracks. Sensors. 2021;21(12):4065. https://doi.org/10.3390/s21124065

[3] Burdzik R, Konieczny Ł, Nowak B, Rozmus J. Research on vibration employed for the train traffic control. Vibroengineering Procedia. 2018;14:227-232.
https://doi.org/10.21595/vp.2017.19237

[4] Ciszewski T, Nowakowski W, Bojarczak P. Application of RAMS methodology to computerized railway systems. Transport Means. 2022;1:212-217.

[5] Ciszewski T, Nowakowski W, Chrzan M. RailTopoModel and RailML – data exchange standards in railway sector. Archives of Transport System Telematics. 2017;10(4):10-15.

[6] Chen W, Liu W, Li K, Wang P, Zhu H, Zhang Y et al. Rail crack recognition based on adaptive weighting multi-classifier fusion decision. Measurement. 2018;123:102-114.
https://doi.org/10.1016/j.measurement.2018.03.059

[7] Chen Z, Wang Q, Yang K, Yu T, Yao J, Liu Y et al. Deep learning for the detection and recognition of rail defects in ultrasound B-scan images. Transp Res Rec. 2021: 036119812110215.
https://doi.org/10.1177/03611981211021547

[8] Drózd P, Rosiński A. Increasing the readiness of railway traffic control devices using a functional test generation method. Appl Sci. 2023;13(13):7717.
https://doi.org/10.3390/app13137717

[9] Gołabek P, Madej L. Kwalifikacja zapisów B-scan z ultradźwiękowej badania szyn kolejowych za pomocą modelu wytrenowanego w trybie głębokiego uczenia. Przegląd Elektrotechniczny. 2019;1(12):119-22.
https://doi.org/10.15199/48.2019.12.24

[10] Kokurin JM, Efanov DV. Technological foundations of traffic controller data support automation. Proceedings of 17th IEEE East-West Design & Test Symposium (EWDTS'2019), 2019:176-180.

[11] Liu X, Lovett A, Dick T, Rapik Saat M, Barkan CP. Optimization of ultrasonic rail-defect inspection for improving railway transportation safety and efficiency. J Transp Eng. 2014;140(10):04014048.
https://doi.org/10.1061/(asce)te.1943-5436.0000697

[12] Nowakowski W, Ciszewski T, Lukasik Z. Application of logical diagnosis to identify faults in computerized railway traffic control systems. In: Ginters E, Ruiz Estrada M, Piera Eroles M. (eds). ICTE in Transportation and Logistics 2019. ICTE ToL 2019. Lecture Notes in Intelligent Transportation and Infrastructure. Springer, Cham.
https://doi.org/10.1007/978-3-030-39688-6_16

[13] Sadeghi J, Rahimizadeh Y, Khajehdezfuly A, Rezaee M, Rajaei Najafabadi E. Development of rail-condition assessment model using ultrasonic technique. J Transp Eng A. 2020;146(8):04020078.
https://doi.org/10.1061/jtepbs.0000390

[14] Shamanov V. Formation of interference from power circuits to apparatus of automation and remote control. Proceedings of 16th IEEE East-West Design & Test Symposium (EWDTS'2018). 1218:140-146.
https://doi.org/10.20295/2412-9186-2022-8-03-252-265

[15] Sulimova V, Zhukov A, Krasotkina O, Mottl V, Markov A. Automatic rail flaw localization and recognition by feature-less ultrasound signal analysis. 14th International Conference, MLDM 2018. Cham, Switzerland, Springer.
https://doi.org/10.1007/978-3-319-96136-1

[16] Wu F, Li Q, Li S, Wu T. Train rail defect classification detection and its parameters learning method. Measurement. 2020;151:107246.
https://doi.org/10.1016/j.measurement.2019.107246

[17] Zabielska A, Jacyna M, Lasota M, Nehring K. Evaluation of the efficiency of the delivery process in the technical object of transport infrastructure with the application of a simulation model. Eksploat Niezawodn. 2023;25(1).
http://doi.org/10.17531/ein.2023.1.1