CIVIL AND ENVIRONMENTAL ENGINEERING REPORTS

E-ISSN 2450-8594

CEER 2025; 35 (3): 0313-0325
DOI: 10.59440/ceer/205939
Original Research Article

FEASIBILITY STUDY OF NEURAL NETWORK-BASED CLASSIFICATION OF CONVEYOR BELT DAMAGE USING PARTIAL DIAGBELT DATA

Aleksandra RZESZOWSKA[1], Leszek JURDZIAK, Ryszard BŁAŻEJ
Wroclaw University of Science and Technology, Faculty of Geoengineering, Mining and Geology, Department of Mining, Wroclaw, Poland

A b s t r a c t

Non-invasive methods for diagnosing conveyor belts enable effective detection of damage, significantly reducing the costs associated with belt replacement. Additionally, they allow for continuous monitoring of the belts' technical condition and degree of wear over extended periods of operation. Such solutions also enhance safety in environments where conveyor systems are used. While belt wear is an inevitable process, its rate can vary depending on specific operational conditions, such as the conveyor's location, its length, the type of material being transported, and the belt's operating speed. This article discusses an artificial intelligence-based approach to classifying conveyor belt damage. A two-layer neural network was implemented in the MATLAB environment using the Deep Learning Toolbox. By optimizing the network, a high level of operational efficiency was achieved, reaching an accuracy range of 80–90%. This solution opens new possibilities for precise diagnostics and monitoring of conveyor belts' technical state, contributing to improved durability and reduced operational costs.

Keywords: conveyor belt, magnetic method, DiagBelt system, damage detection, artificial neural networks

1. INTRODUCTION

The durability of a conveyor belt is influenced by multiple factors discussed in the literature [1]. These include the characteristics of the transported material, the design and conditions of the transfer point, as well as the conveyor's length and operational age. Figure 1 illustrates a conveyor belt commonly used in mining operations [2].

[1] Corresponding author: Aleksandra RZESZOWSKA, Wroclaw University of Science and Technology, Faculty of Geoengineering, Mining and Geology, Na Grobli 15 st, 50-421 Wrocław, Poland, aleksandra.rzeszowska@pwr.edu.pl

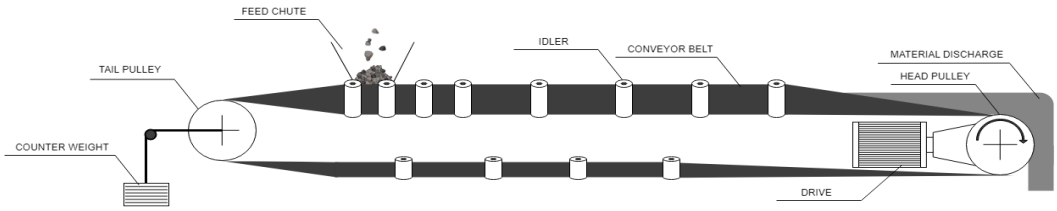


Fig. 1. Conveyor belt diagram

The conveyor belt is the most critical and failure-prone component of the system. It is estimated that the belt alone constitutes about 60% of the conveyor's total cost [3]. To meet industry standards, manufacturers are required to produce high-quality belts, which further increases their expense. This underscores the importance of effective diagnostics and prompt detection of potential damage, enabling timely repairs before problems worsen. A belt failure results in not only repair expenses but also significant costs associated with transport downtime [4]. Figure 2 presents a cross-sectional diagram of the conveyor belt.

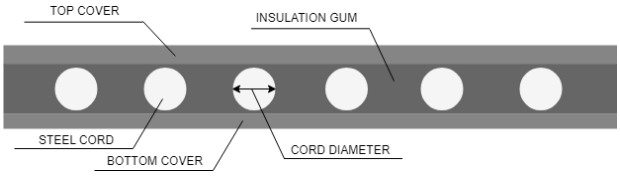


Fig. 2. Cross-section of a conveyor belt – diagram

Non-Destructive Testing (NDT) techniques ensure that the examined object—in this case, the conveyor belt—remains intact, with no changes to its structure or properties during testing. Researchers worldwide have developed numerous systems to diagnose conveyor belt cores [5–10]. Some methods focus on assessing the condition of the belt covers, while others are designed to detect damage to the steel core embedded in rubber [11]. With the advent of Industry 4.0, sensors can now be integrated into the examined object, enabling continuous data collection and advanced processing to enhance the diagnostic process [12–13].

One innovative method of non-invasive conveyor belt diagnostics involves detecting changes in the magnetic field as the belt moves beneath a measurement head spanning its entire width. This technology is being tested at the Conveyor Transport Laboratory of the Wrocław University of Science and Technology [14–15].

Magnetic methods for damage identification work by recording variations in the magnetic field generated by pre-magnetized steel cords in the belt's core. These variations can result from belt segment connections or damage such as cuts, corrosion, or missing cords. The technique involves inducing a magnetic field with a sufficiently high flux in the belt, then searching for magnetic leakage fields indicative of defects.

To enhance damage detection and optimize the setup of the measurement device (e.g., belt speed, sensor distance, sensitivity), analyses were conducted using data from the Diagbelt system [16].

Data obtained from the Diagbelt system can be visualized in a two-dimensional diagram. An example of such visualization is presented in Figure 3.

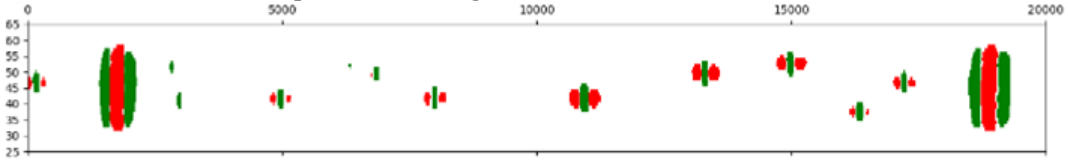


Fig. 3. Visualization of measurement data in a two-dimensional image

1.1. Neural network

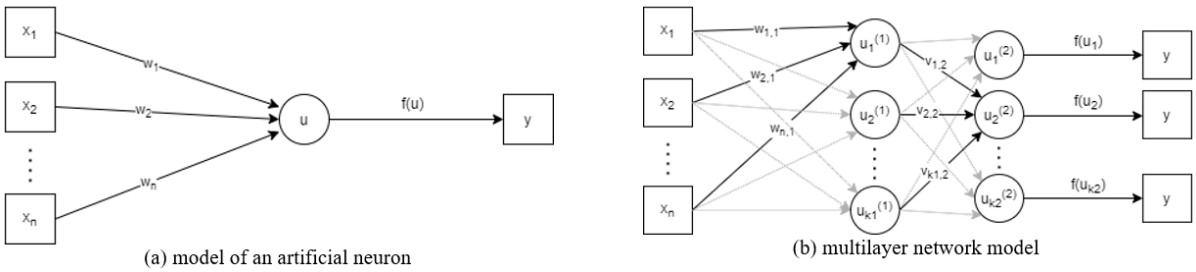
Artificial Neural Networks (ANNs) are a class of computational models inspired by the structure and functioning of the human brain. They consist of interconnected processing units (neurons) that transform input signals into output responses through a system of weighted connections and nonlinear activation functions. The origin of neural networks dates back to early models such as the McCulloch-Pitts neuron (1943), while the perceptron model proposed by Rosenblatt in the 1950s laid the foundation for learning algorithms [17–18]. Despite a temporary decline in interest due to theoretical limitations identified by Minsky and Papert [19], the field saw a resurgence with the introduction of the backpropagation algorithm for training multilayer networks [20].

The popularity of ANNs in classification tasks stems from their ability to model complex, non-linear relationships between input and output data, making them effective in pattern recognition, speech analysis, image processing, and decision-making systems. Recent advances in deep learning, which involves ANNs with many hidden layers, have further expanded their capabilities across various scientific and industrial domains [21].

In technical diagnostics and non-destructive testing (NDT), neural networks have emerged as powerful tools for detecting faults in materials or mechanical systems without causing physical damage. Their ability to learn from sensor data enables them to detect patterns that may be imperceptible to traditional analysis methods[22]. This includes successful applications in the monitoring of steel structures, pipeline corrosion, weld defects, or — as in this work — conveyor belt damage [23–24].

An artificial neural network typically consists of several layers: an input layer, one or more hidden layers, and an output layer. Each neuron in a layer computes a weighted sum of its inputs and passes the result through an activation function. The input layer has as many neurons as there are input features. In some implementations, an additional bias neuron is introduced, which always outputs a constant signal. The number of neurons in the hidden layer(s) may vary depending on the problem's complexity. The output layer's size corresponds to the number of classification categories or target values.

Figure 4a shows the structure of a single artificial neuron, which processes multiple inputs and generates a single output value. Figure 4b illustrates a simple multilayer network with one hidden layer containing k_1 neurons.



(a) model of an artificial neuron

(b) multilayer network model

Fig. 4. Neural network – diagram

Selecting the number of layers and the number of neurons in each hidden layer is one of the most important design decisions when constructing a neural network. In practice, networks with one or two hidden layers are usually sufficient for most classification tasks. Deeper networks can capture more complex patterns but are harder to train and require more data and computational power.

Beyond the architecture itself, there are other critical parameters that influence the performance of the network. These include the choice of activation functions (e.g., sigmoid, ReLU, tanh), the loss function (e.g., mean squared error for regression, cross-entropy for classification), and the optimization

algorithm used to minimize the error during training (e.g., stochastic gradient descent, Adam) [17–21]. The training process adjusts the weights of the connections between neurons using the backpropagation algorithm — a supervised learning technique where the error is propagated backward from the output layer to the input layer. Variations of this algorithm, such as those incorporating momentum, can accelerate convergence and improve generalization [21].

2. PRELIMINARY DATA ANALYSIS

Before inputting data into the neural network, it is essential to perform preliminary data processing. Image processing, presented as values of successive pixels, is primarily handled by deep neural networks, which require a large input database for the training process to proceed correctly. An image with dimensions of 480 × 600 px presented in the classical version, pixel by pixel, necessitates a network structure with 240,000 neurons in the input layer! However, this problem can be circumvented, and satisfactory results can be obtained even with a relatively small training database. In the described case of detecting damage in conveyor belts, areas in close proximity were isolated with the help of the Python programming language. The boundaries of such areas are illustrated in Figure 5.

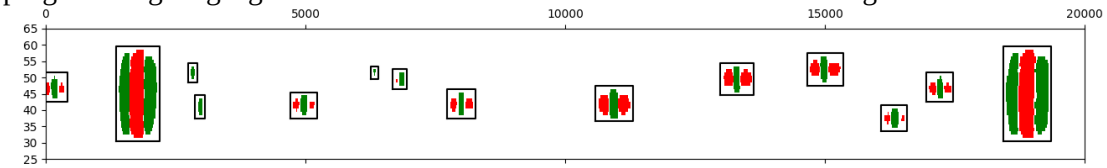
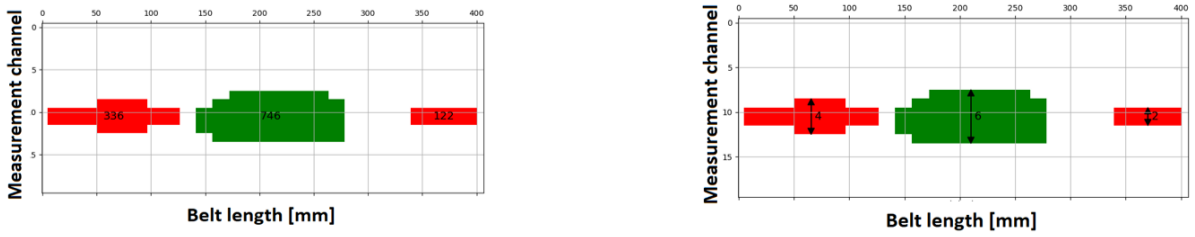


Fig. 5. Reference conveyor belt - boundaries of potential damage areas

To avoid inputting the entire data vector into the network, three values were selected to represent the areas of the subregions that make up the detected area. These values correspond to the area of the red region, the area of the green region, and the combined area of the red and green regions. Additionally, three values describe the number of channels on which the signal corresponding to the point cloud of each subregion was detected. Figure 6 illustrates the measurements taken from the point cloud. In cases where the detected area consists of one or two subregions, any missing subregion is assigned an area of 0.



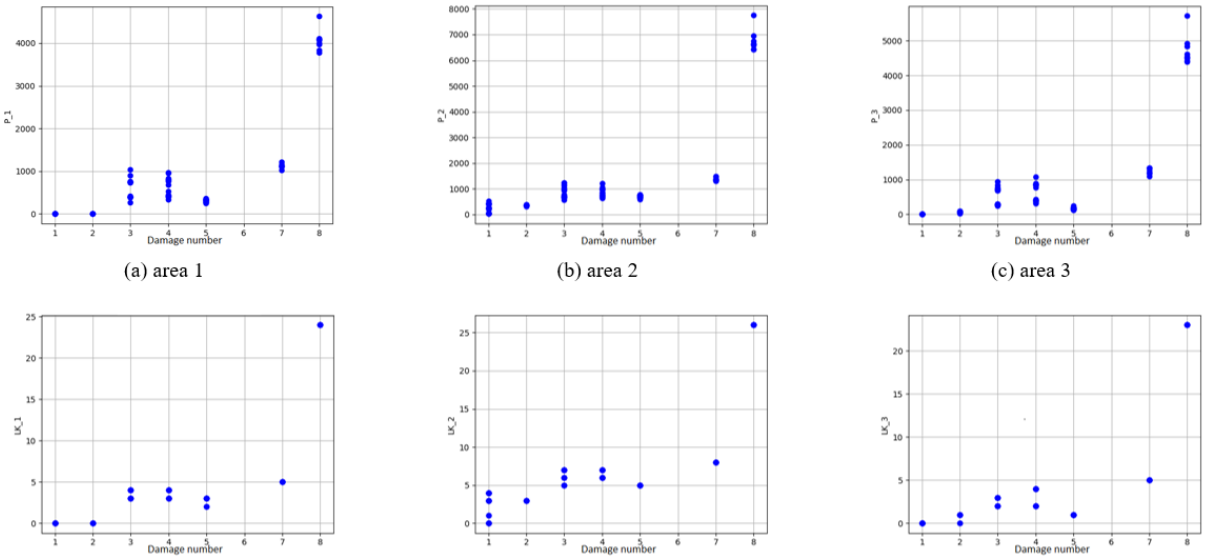
(a) areas of subregions (b) number of channels of subregions

Fig. 6. Image of damage

In this way, a large input vector was replaced with six values. Among the possible detectable damages on the examined conveyor belt, the following were distinguished: the absence of a single cord (U1), the absence of two cords (U2), the absence of three cords (U3), a cut in the strands/wires (U4), a partial cut of one cord (U5), a cut of one cord (U6), a cut of two cords (U7), a cut of three cords (U8), and a splice in the belt (U9).

To process the prepared input data and potential outputs, the neural network structure requires 6 neurons in the input layer and 9 neurons in the output layer. However, since damage U2 is not represented in the analyzed dataset, the network's output consists of an 8-element vector.

Figure 7 displays the distribution of the measured parameters based on the type of damage, offering an initial evaluation of whether the chosen parameters can effectively distinguish between the various types of failures.



(a) area 1 (b) area 2 (c) area 3

Fig. 7. Distribution of measured parameters

3. NEURAL NETWORK TRAINING PROCESS

The neural network training was carried out in the MATLAB environment using the Deep Learning Toolbox. The dataset includes 98 examples, with 8 reserved for testing—one randomly selected from each category (the category with two missing cords was excluded, as it contains no training examples). The training set consists of 90 examples. The aim of the network training process is to find the optimal weight values that ensure the correct solution for the task.

Before training the network, a hidden layer must be created (and the number of neurons defined) using an appropriate function, which also determines how to adjust the weight matrix. In this study, backpropagation was chosen as the optimization algorithm. The tool automatically trains the network while simultaneously calculating the validation error. When this error begins to rise, the training process is stopped to maintain the network's ability to generalize, i.e., to solve tasks involving examples it has not encountered during training.

3.1. Optimization of the number of neurons in the hidden layer

The first of the parameters optimized in the designed artificial neural network was the number of neurons in the hidden layer. The number of neurons in the input and output layers is determined by the size of the input vector and the expected output vector (in this case, 6 and 9, respectively).

In both the hidden and output layers, the activation function was established as a sigmoid function (this function takes values in the range <0,1>). The number of neurons in the hidden layer was varied within the range of $k_1 \in \{3,8,20,100\}$. To minimize the stochastic properties, the training process was conducted 10 times, and the results were averaged. Table 1 summarizes the averaged values of the network's responses according to the specified class of damage. The test set contained one representative from each possible class, randomly selected each time from the full database.

Table 1. Averaged results - examination of the number of neurons in the hidden layer

k_1	U1	U3	U4	U5	U6	U7	U8	U9	MSE
$k_1 = 3$	**0.71**	-0.02	0.09	0.04	0.01	0.14	0.06	0.01	0.0615
	-0.02	**0.25**	0.05	-0.02	0.07	0.12	0.15	0.11	
	0.04	0.04	**0.34**	0.09	0.13	0.04	0.03	0.04	
	0.04	-0.03	0.15	**0.75**	0.30	-0.01	-0.01	-0.01	
	0.00	0.09	0.16	0.17	**0.17**	0.08	0.10	0.01	
	0.14	0.22	0.07	-0.02	0.12	**0.30**	0.29	0.03	
	0.07	0.29	0.07	-0.01	0.16	0.31	**0.34**	0.03	
	0.00	0.11	0.05	-0.01	0.01	0.00	0.01	**0.70**	
$k_1 = 8$	**0.87**	0.02	0.02	0.02	-0.02	0.06	0.01	0.02	0.0198
	0.00	**0.89**	0.00	0.00	0.01	0.03	0.02	0.00	
	0.01	0.00	**0.91**	0.00	0.05	-0.01	-0.01	0.00	
	0.02	0.00	-0.01	**0.91**	0.21	0.00	0.00	0.00	
	0.00	0.01	0.09	0.08	**0.56**	0.03	0.04	0.01	
	0.05	0.06	-0.01	-0.01	0.10	**0.63**	0.17	-0.01	
	0.03	0.01	-0.01	0.00	0.09	0.23	**0.74**	0.00	
	0.00	-0.01	0.01	0.00	-0.01	0.00	0.00	**0.90**	
$k_1 = 20$	**0.93**	0.01	0.01	0.03	-0.01	0.04	0.02	0.00	0.0116
	0.00	**0.96**	0.00	0.01	0.00	0.01	0.00	0.00	
	0.01	0.00	**0.97**	0.00	0.02	0.00	0.00	0.00	
	0.02	0.02	0.01	**0.92**	0.13	-0.01	0.00	-0.01	
	-0.03	-0.01	0.02	0.07	**0.70**	0.06	0.02	0.00	
	0.02	0.01	0.00	-0.01	0.11	**0.74**	0.14	-0.07	
	0.02	0.00	0.00	-0.01	0.04	0.14	**0.81**	0.10	
	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**0.87**	
$k_1 = 100$	**0.89**	0.01	0.01	0.02	-0.02	0.05	0.02	0.00	0.0120
	0.00	**0.96**	0.00	0.00	-0.01	0.01	0.01	0.02	
	0.01	0.00	**0.97**	0.01	0.03	0.01	0.00	0.00	
	0.04	0.01	0.01	**0.90**	0.17	-0.02	0.00	-0.01	
	-0.02	-0.01	0.02	0.08	**0.66**	0.05	0.02	0.02	
	0.04	0.02	0.00	-0.01	0.12	**0.77**	0.17	-0.01	
	0.04	-0.01	0.00	-0.01	0.05	0.13	**0.76**	0.01	
	0.00	0.01	0.00	0.00	0.00	0.00	0.00	**0.95**	

Analyzing the data from Table 1, the number of neurons in the hidden layer was determined to be $k_1 = 20$. For this value, the network exhibited the highest effectiveness. Figure 8 shows the confusion matrices obtained for each of the examined values—values are illustrated both by numerical values and by the selection of appropriate colors—white indicates a value of 0, while navy blue indicates a value of 1. A network that correctly recognizes all given examples has a confusion matrix with values of 1 on the diagonal and values of 0 elsewhere. A value greater than 1 outside the diagonal indicates a misclassification of a given class in favor of another. The presented figures include a first column indicating the absence of a decision made by the network.

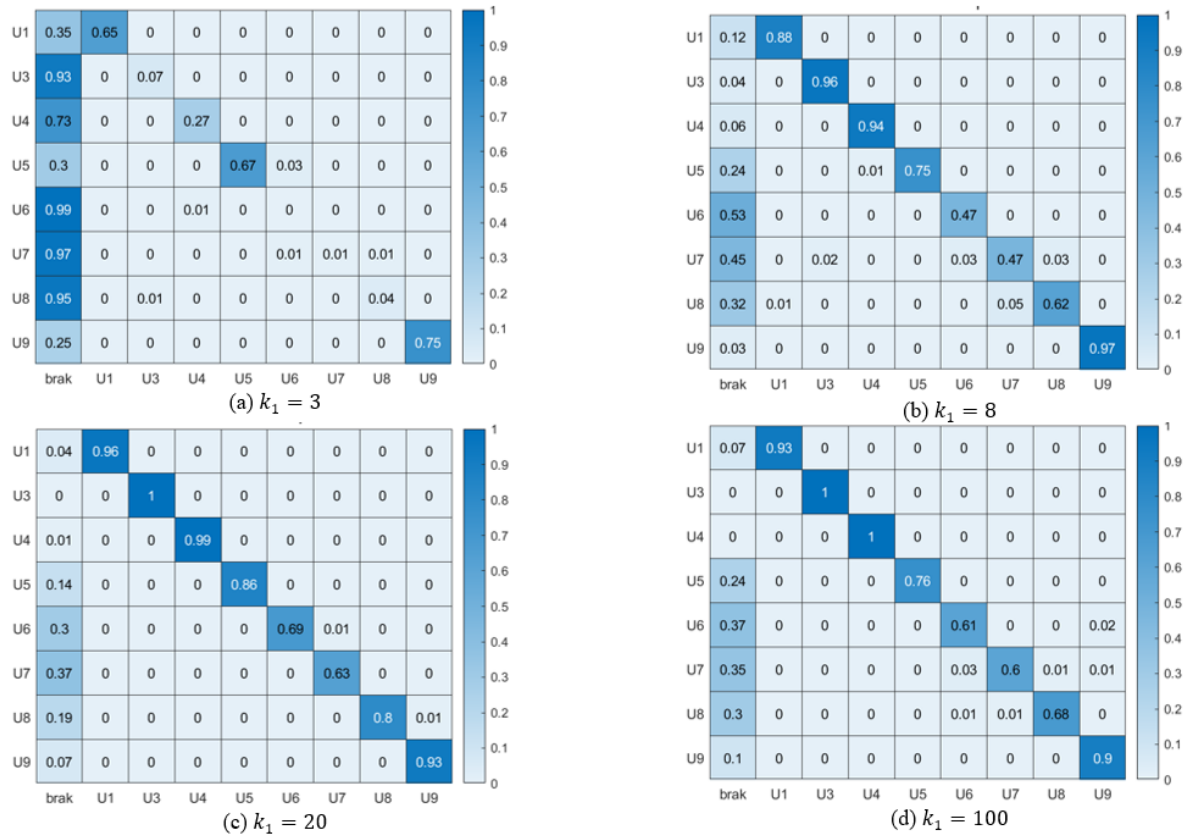


Fig. 8. Confusion matrix - examination of the number of neurons in the hidden layer

3.2. Optimization of the activation function in the hidden layer

With the number of neurons in the hidden layer set to $k_1 = 20$, an examination of the activation function in the hidden layer was conducted. These functions were varied within the range of $fun_{akt} \in \{logsig, poslin, tansig, radbas\}$. The graphs of these functions are shown in Figure 9.

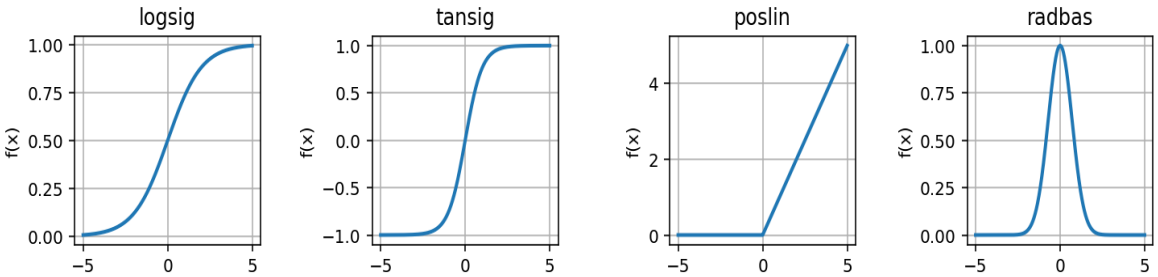


Fig. 9. Activation function curves

The choice of an appropriate activation function can be crucial in solving a given classification (or regression) problem. Table 2 summarizes the results obtained in the process of optimizing the activation function in the hidden layer.

Table 2. Averaged results - examination of the activation function in the hidden layer

fun_{akt}	U1	U3	U4	U5	U6	U7	U8	U9	MSE
logsig	**0.88**	0.01	0.02	0.02	-0.01	0.07	0.03	0.01	0.0139
	0.00	**0.94**	0.00	0.00	0.00	0.00	0.02	0.00	
	0.01	0.00	**0.94**	0.00	0.03	-0.01	-0.01	0.00	
	0.04	0.01	0.03	**0.92**	0.21	0.01	0.00	-0.01	
	-0.03	0.00	0.03	0.06	**0.65**	0.08	0.03	0.02	
	0.06	0.02	0.00	-0.01	0.09	**0.67**	0.17	-0.01	
	0.05	0.02	-0.01	0.01	0.04	0.18	**0.74**	0.01	
	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**0.91**	
tansig	**0.92**	0.01	0.03	0.03	-0.01	0.06	-0.01	0.02	0.0133
	0.01	**0.94**	0.00	0.00	0.00	0.00	0.02	-0.02	
	0.01	0.00	**0.97**	0.00	0.04	0.00	0.00	0.00	
	0.03	0.00	0.00	**0.88**	0.21	-0.02	-0.01	-0.01	
	-0.02	0.01	0.04	0.10	**0.65**	0.06	0.04	0.02	
	0.03	0.02	0.00	0.00	0.08	**0.73**	0.17	-0.02	
	0.01	0.01	-0.01	-0.01	0.04	0.15	**0.76**	0.03	
	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**0.91**	
poslin	**0.81**	-0.01	0.06	0.06	-0.14	0.18	0.07	0.02	0.0330
	0.00	**0.71**	0.00	-0.02	0.02	0.01	0.08	0.00	
	0.03	0.01	**0.91**	0.01	0.11	0.01	-0.03	0.00	
	0.06	-0.03	-0.01	**0.79**	0.33	-0.01	0.03	0.00	
	-0.06	0.03	0.09	0.16	**0.50**	0.03	0.07	0.00	
	0.14	0.07	0.02	-0.03	0.04	**0.49**	0.24	0.00	
	0.01	0.22	-0.06	0.02	0.11	0.28	**0.51**	-0.06	
	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**0.93**	
radbas	**0.92**	0.02	0.01	0.02	-0.02	0.03	0.01	-0.01	0.0119
	0.00	**0.97**	0.00	0.01	0.00	0.01	0.01	-0.01	
	0.00	0.00	**0.97**	0.00	0.02	-0.01	0.00	0.01	
	0.02	0.01	-0.02	**0.89**	0.13	-0.01	-0.01	0.03	
	0.00	0.00	0.04	0.10	**0.69**	0.06	0.03	-0.05	
	0.01	-0.02	0.00	0.00	0.12	**0.74**	0.15	0.02	
	0.05	0.02	-0.01	-0.01	0.06	0.16	**0.78**	-0.01	
	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**0.88**	

A helpful approach in selecting the best activation function is to examine the confusion matrices. Such matrices for the tested activation functions in the hidden layer are shown in Figure 10.

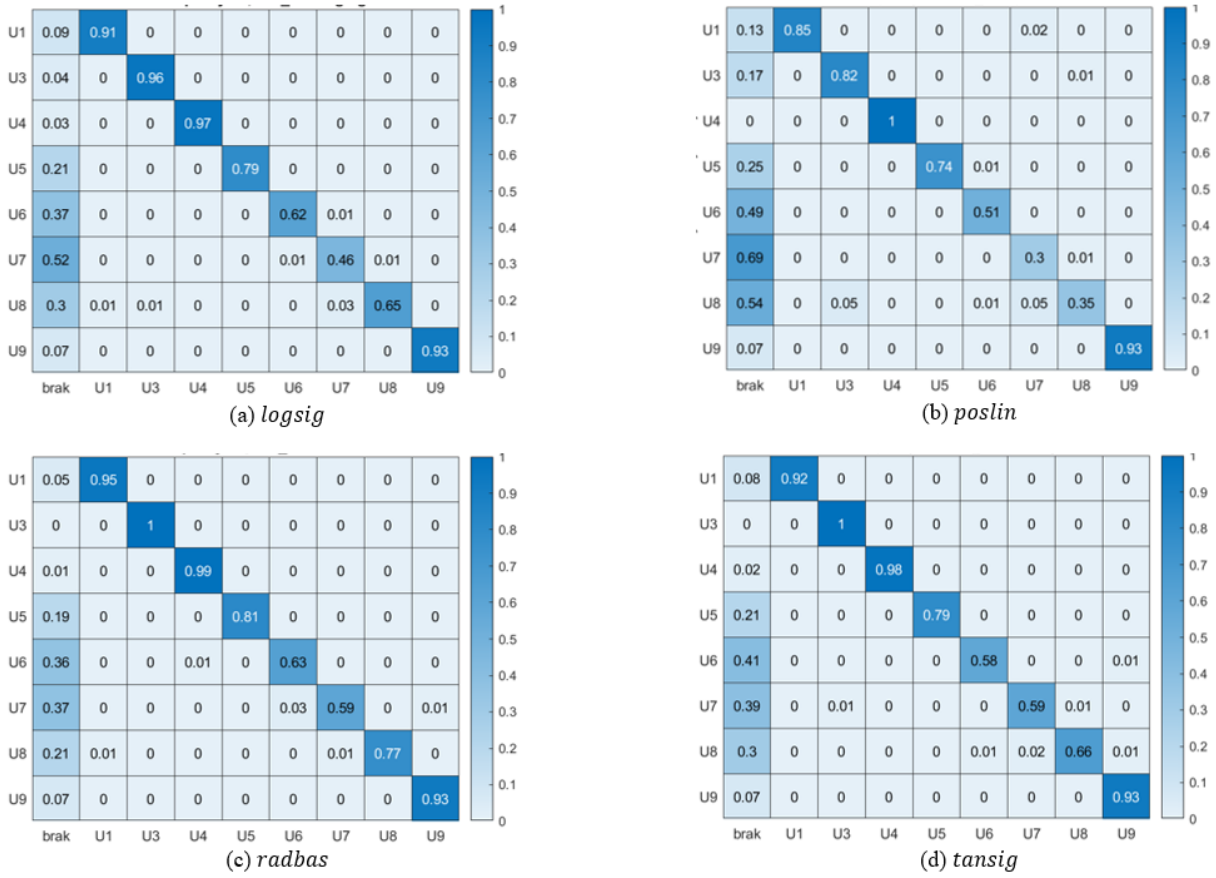


Fig. 10. Confusion matrix - examination of the activation function in the hidden layer

Based on the data collected in Table 2 and the confusion matrices presented in Figure 10, the optimal activation function in the hidden layer was determined to be radbas. This activation function was adopted for further studies.

3.3. Optimization of the activation function in the hidden layer

An important aspect of the functioning of a neural network is the selection of the appropriate activation function in the output layer. Here, it is worth noting the expected output of the network—if values from the set of real numbers are expected without a maximum and minimum (for example, in regression tasks), a linear function ($purelin$) should be used. If the expected output takes only values greater than 0, the use of the ReLU function ($poslin$) can be considered. If the output is limited to the range of values <-1, 1>, the hyperbolic tangent function ($tanh$) can be used. In classification tasks, the use of the softmax function is often also considered, which converts the value obtained by the network into the probability of accepting a given class (the sum of the values at all outputs equals 1). In the conducted study, the activation function in the output layer was varied in the range of $fun_{out} \in \{purelin, tansig, softmax\}$. The averaged results obtained from the study are presented in Table 3.

Table 3. Averaged results - study of the activation function in the output layer

fun_{out}	U1	U3	U4	U5	U6	U7	U8	U9	MSE
$purelin$	**0.93**	0.03	0.01	0.03	-0.03	0.05	0.00	0.00	0.0121
	0.00	**0.96**	0.00	0.01	0.00	0.00	0.02	0.00	
	0.01	0.00	**0.97**	0.00	0.03	0.00	0.00	0.03	
	0.02	0.01	0.00	**0.88**	0.20	-0.01	0.00	0.03	
	0.00	-0.01	0.03	0.11	**0.63**	0.09	0.03	-0.02	
	0.03	0.01	0.00	0.00	0.13	**0.74**	0.12	0.01	
	0.00	-0.02	-0.01	-0.02	0.05	0.12	**0.82**	0.01	
	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**0.86**	
$tansig$	**0.91**	0.01	0.01	0.03	0.00	0.04	0.01	0.02	0.0130
	0.00	**0.90**	0.01	0.00	0.00	0.00	0.03	0.01	
	0.00	0.01	**0.92**	0.01	0.03	0.01	0.00	0.02	
	0.02	0.01	0.01	**0.86**	0.18	0.00	0.00	0.02	
	0.00	0.01	0.05	0.09	**0.61**	0.05	0.01	0.03	
	0.04	0.02	0.01	0.00	0.10	**0.67**	0.16	0.03	
	0.02	0.04	0.00	0.01	0.07	0.18	**0.68**	0.01	
	0.00	0.01	0.01	0.00	0.00	0.00	0.00	**0.87**	
$softmax$	**0.88**	0.51	0.53	0.53	0.52	0.56	0.54	0.51	0.2533
	0.51	**0.82**	0.51	0.51	0.51	0.52	0.53	0.50	
	0.51	0.51	**0.85**	0.52	0.55	0.51	0.51	0.51	
	0.53	0.51	0.55	**0.89**	0.75	0.51	0.51	0.50	
	0.51	0.51	0.52	0.52	**0.57**	0.51	0.51	0.50	
	0.54	0.56	0.52	0.52	0.55	**0.73**	0.65	0.51	
	0.53	0.59	0.52	0.51	0.55	0.66	**0.76**	0.51	
	0.50	0.50	0.51	0.50	0.50	0.50	0.50	**0.96**	

4. SUMMARY

The popularity of using artificial neural networks to solve classification problems primarily stems from their ability to make decisions that can often be more accurate than those made by a team of specialists. They can find applications in nearly every field of science. For instance, a neural network developed by a team of specialists from the USA and China for use in medicine to diagnose patient diseases frequently achieves results that surpass those of a team of medical doctors. For example, the effectiveness of recognizing the presence of asthma is 90%, while the accuracy of diagnoses made by doctors ranges from 82% to 90%.

Magnetic sensor data is highly suitable for further processing when searching for potential damage, and recording this data in a two-dimensional format makes it easier to work with and verify the processed information. However, for accurate identification of potential issues, it is essential for the operator to be well-trained, enabling them to focus on the most characteristic points that signify the classification of a given area into a specific category. Therefore, it is worth considering the integration of artificial neural network mechanisms to automate such tasks, eliminating the need to manually search for values that might be indicative of a particular category. During the learning process, the network will automatically determine the input weights, allowing it to assess which elements of the input sequence are most significant for classifying a given category.

The use of artificial intelligence, specifically neural networks, for classifying conveyor belt damage presents a promising research avenue. Preliminary results discussed in this article support the potential of artificial neural networks in this field. To further improve the network's classification performance, it would be beneficial to incorporate additional input features, such as the displacement of a given subarea's centroid from the geometric centre, the distance between the centroids of individual subareas, or even the span of the area in both directions.

A crucial aspect of designing artificial neural networks is their optimization. Each network consists of numerous hyperparameters that must be carefully selected to achieve optimal performance. These parameters include the number of hidden layers, the number of neurons in each layer, and the appropriate choice of optimizer and loss function. The selection of some parameters depends on the nature of the problem being addressed. For instance, the cross-entropy loss function is typically used for classification tasks, while the mean squared error (MSE) function is most commonly applied in regression tasks.

Research conducted to optimize the network's performance was carried out in a manner that avoided generating multiple solutions – a set of initial parameters was established, and then only one value was changed in each experiment, with the best result from that experiment being used for further analysis. This chosen method of conducting experiments represents a compromise between time, the number of experiments conducted, and optimization capabilities. However, it may turn out that there is another parameter selection that guarantees a better solution and has not been subjected to the research process.

Due to the fact that the weights assigned to neurons are initialized randomly, it is essential to conduct multiple experiments to eliminate stochastic properties. In the research conducted as part of the project, experiments were repeated one hundred times, and the results were averaged.

The analysis of the data presented in Table 1 and Figure 8 indicates that too few neurons in the hidden layer ($k_1 = 3$) limit the network's performance, and the obtained results are not optimal. Conversely, too many neurons ($k_1 = 100$) significantly prolong the learning algorithm's runtime, and the network does not show improvement. Good results were obtained with 8 and 20 neurons in the hidden layer—here, there are slight differences favoring 20 neurons.

Changing the activation function in the hidden layer mainly causes a change between no response from the network and correct classification, although the change is not highly noticeable. The worst results among those presented in the study were obtained for the ReLU activation function ($poslin$), while the best results were for the radial basis function ($radbas$)—for this reason, the radbas function was chosen for further research. The use of different activation functions in the output layer only demonstrated that in this case, the $softmax$ function is not applicable, while both the linear function and the hyperbolic tangent function are well-suited for the classification task at hand.

It is worth noting that the improvement in the network's performance after optimizing these few parameters is significant, although many values still need to be tuned to achieve a faster and possibly better-performing network. It is also important to highlight that the training set consisted of fewer than 100 examples, and nevertheless, the recognition effectiveness oscillated around 90%. Increasing the training base will certainly improve the network's effectiveness and can be the subject of further analysis. Due to the limitations of available data, the classification results were not verified directly with manual inspection or real-world failure validation. However, future studies will aim to evaluate the correspondence between the predicted and actual damage patterns.

REFERENCES

1. Webb, C, Sikorska, J, Khan, RN and Hodkiewicz, M 2020. Developing and evaluating predictive conveyor belt wear models. *Data-Centric Engineering 1, e3.*
2. Żur, T and Hardygóra, M 1996. *Przenośniki taśmowe w górnictwie [Conveyor belts in mining]*. Katowice: Śląsk.
3. Błażej, R 2018. *Ocena stanu technicznego taśm przenośnikowych z linkami stalowymi [Assessment of the technical condition of steel cord conveyor belts]*. Wrocław: Wydział Geoinżynierii, Górnictwa i Geologii Politechniki Wrocławskiej.
4. Bugaric, U, Tanasijevic, M, Polovina, D, Ignjatovic, D and Jovancic, P 2012. Lost production costs of the overburden excavation system caused by rubber belt failure. *Eksploatacja i Niezawodnosc* 14.
5. Kumbhar, SG and Edwin Sudhagar, P 2021. Fault Diagnostics of Roller Bearings Using Dimension Theory. *Journal of Nondestructive Evaluation, Diagnostics and Prognostics of Engineering Systems* 4.
6. Skoczylas, A, Stefaniak, P, Anufriiev, S and Jachnik, B 2021. Belt conveyors rollers diagnostics based on acoustic signal collected using autonomous legged inspection robot. *Applied Sciences (Switzerland)* 11.
7. Roskosz, M, Złocki, A and Kwaśniewski, J 2020. Self Magnetic Flux Leakage as a Diagnostic Signal in the Assessment of Active Stress - Analysis of Influence Factors. *Acta Physica Polonica A* 137, 690–692.
8. Zhang, M, Shi, H, Yu, Y and Zhou, M 2020. A Computer Vision Based Conveyor Deviation Detection System. *Applied Sciences* 10, 2402.
9. Siami, M, Barszcz, T, Wodecki, J and Zimroz, R 2022. Design of an Infrared Image Processing Pipeline for Robotic Inspection of Conveyor Systems in Opencast Mining Sites. *Energies* 15, 6771.
10. Martínez-Parrales, R and Téllez-Anguiano, AdC 2022. Vibration-based Fault Detection System with IoT Capabilities for a Conveyor Machine. *Acta Polytechnica Hungarica* 19.
11. Błażej, R, Jurdziak, L, Kirjanów-Błazej, A and Kozłowski, T 2021. Identification of damage development in the core of steel cord belts with the diagnostic system. *Scientific Reports* 11.
12. Fedorko, G 2019. Implementation of Industry 4.0 in the belt conveyor transport. *MATEC Web of Conferences* 263.
13. Mendes, D, Gaspar, PD, Charrua-Santos, F and Navas, H 2023. Enhanced real-time maintenance management model - a step toward Industry 4.0 through Lean: Conveyor belt operation case study. *Electronics* 12.
14. Błażej, R, Jurdziak, L, Kozłowski, T and Kirjanów, A 2018. The use of magnetic sensors in monitoring the condition of the core in steel cord conveyor belts – Tests of the measuring probe and the design of the DiagBelt system. *Measurement: Journal of the International Measurement Confederation* 123.
15. DiagBelt+ 2024. https://diagbeltplus.pwr.edu.pl/ (accessed 01-11-2024).
16. Olchówka, D, Błażej, R and Jurdziak, L 2022. *Selection of measurement parameters using the DiagBelt magnetic system on the test conveyor*. *Journal of Physics: Conference Series* 2198, 012042.
17. Haykin, S 2009. *Neural Networks and Learning Machines*, 3rd ed. Hamilton: Pearson Education, Inc.
18. Rosenblatt, F 1958. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review* 65, 386–408.
19. Minsky, M and Papert, S 1969. *Perceptrons: An Introduction to Computational Geometry*, 1st ed. Cambridge, MA: MIT Press.

20. Rumelhart, DE, Hinton, GE and Williams, RJ 1986. Learning representations by back-propagating errors. *Nature* 323, 533–536.
21. LeCun, Y, Bengio, Y and Hinton, G 2015. Deep learning. *Nature* 521, 436–444.
22. Kim, J-W and Park, S 2018. Magnetic Flux Leakage Sensing and Artificial Neural Network Pattern Recognition-Based Automated Damage Detection and Quantification for Wire Rope Non-Destructive Evaluation. *Sensors* 18, 109.
23. Kirjanów-Błażej, A and Rzeszowska, A 2021. Conveyor belt damage detection with the use of a two-layer neural network. *Applied Sciences (Switzerland)* 11.
24. Rzeszowska, A, Jurdziak, L, Błażej, R and Kirjanów-Błażej, A 2023. Application of Clustering and SOM Analysis for Identification of Conveyor Belt Damage Based on Data from the DiagBelt+ Magnetic System. In: Burduk, A., Batako, A., Machado, J., Wyczółkowski, R., Antosz, K., Gola, A. (eds) Advances in Production. ISPEM 2023. Lecture Notes in Networks and Systems, vol 790. Springer, Cham.