Research Article

USING DEEP LEARNING ON SYSTEM IDENTIFICATION OF THE RETAINING WALLMODEL

SERTAÇ TUHTA[1]

[1]Department of Civil Engineering, Ondokuz Mayis University, Samsun, Turkey, stuhta@omu.edu.tr, ORCID: 0000-0003-2671-6894

ABSTRACT

Structures have been adversely affected by dynamic effects from past to present. This has alwaysbeen a problem for structural engineering. Structural engineers strive to design structures to be least affected by dynamic effects. The biggest challenge in these designs is the exact and realistic calculation of the response of dynamic effects on the structure. There are various methods for calculating the dynamic effects affecting the structures. The system identification method is one of the methods used to calculate the responses of the structures to the dynamic effects affecting. In the other words, a Mathematical model of the structure system is obtained by the system identification method. Today, the use of artificial intelligence has increased considerably in every field. In the field of system identification, methods using artificial intelligence (neural network) have been used in many studies in training data. For this reason, in this study, the system identification of the retaining wall model was made with the deep learning method. In the light of the data obtained, it was seen that the system identification of the model was realized close to one hundred percent. As a result of this study, it is seen that deep learning will be useful on system identification methods in the civil engineering field.

Keywords: *System Identification, Deep Learning, Artificial Intelligence, Neural Network, Retaining Wall.*

1.INTRODUCTION

Today, studies and developments about earthquake resistance of structures are

Received: 23.04.2022 - Accepted: 12.07.2022
DOI: 10.17932/EJEAS.2021.024/ejeas_v02i2002

very popular. There are many old and new methods used especially when determining the earthquake performance of structures. The methods used in the past century and the beginning of the 21st century have been replaced by newer and more reliable methods. The old methods are *generally theoretical and based on* some assumptions. The new methods, on the other hand, include more experimental methods and the data represent more real situations. The system identification method is one of these methods. System identification (SI) is a modeling process for an unknown system based on a set of input outputs and is used in various engineering fields [1], [2]. With system identification, a mathematical model of the system is created. Effects and reactions on the created model are determined realistically by the mathematical model. In determining the earthquake performance, it is of great importance to obtain the mathematical model of the structure or model correctly. It is known that it is possible to obtain correct earthquake performance only with the correct mathematical model.

The system identification method also includes various improvements. Especially today, the use of artificial intelligence is seen in many areas. In the field of system identification, its use in obtaining parameters and processing data is seen in current academic studies. The authors have many studies [3-19] on system identification and artificial neural network given in the sources. It has been clearly seen that the limits of deep learning are pushed by covering various models in these academic studies. Based on all this information, the very up-to-date deep learning method was used on system identification in this study. It was decided to choose the retaining wall model as the model. Also, it is aimed to interpret the data obtained in the study by sharing it clearly in the results section.

2.DEEP LEARNING

Deep learning is part of a broader family of machine learning methods based on artificial neural networks with representation learning. Artificial neural networks as defined above are implemented in computer software as learning systems that match inputs to outputs. They are designed to perform the mathematical operation of "simultaneously organizing low-level visual information in a hierarchical structure and then mapping the resulting structure back into the input," in contrast to, say, a traditional vector or matroid, in which the input would be treated linearly or an idealized monoid where the model has an associative structure. Deep learning itself often requires some form of pre-training, a period of trial-and-error based on artificial neural network models where training data are fed into the model through layers of progressively more complex activation functions. This involves defining rules about how to generate neural network representations for particular data, then generating thousands or even millions of models for a data set. A network, which might contain millions of connected neural layers, is trained by comparing its predicted outputs with the actual outputs of the network. The model is trained by feeding it with more and more examples, progressively reducing its accuracy until the output matches the input. The accuracy, or the ability of the network to correctly classify a data item, increases with the size of the input set and with a decreasing number of layers in the network. The ability of the network to adapt to new examples can be measured by the number of times it is classified on average when the data are compared with new examples.

For example, a network with three layers that can accurately classify an input with two features could perform better than a network with a single layer that would only successfully classify the input with three features. The best networks perform so well that, as the number of training examples increases, the network becomes "smooth". For example, a network that can classify a new image with only four examples will show a nearly perfect function until the network is labeled with 64 examples. This is called "lateral inhibition". The number of samples usually used to achieve this is referred to as the "generative model size". Note that if the output of the network matches the actual input a model with a larger number of layers is considered to be a more robust representation of the real data, and can be used to classify it on the fly, without the need to redo all the work in a previous layer. Deep learning is a subset of machine learning. It usually describes the connection between an input signal (the input layer), a hidden layer (the hidden layer or the hidden- mean layer), and an output signal (the output layer or the output layer). However, other machine learning techniques work similarly. For example, finite element models can generate complex systems by considering several different possible solution parameters and then feeding the output from one of these parameters into another, with subsequent evaluations until a solution is found. In contrast, a deep model starts with some hidden state and then works backward through time to find a possible solution. This is the first major concept in deep learning: identifying the output signal and making it the input. If this input is a pre-trained model, we can return to the neural network and select the output that best matches the input. For example, consider a network trained to discriminate between an image of a cat and an image of a dog. Let the hidden layer be the set of multi-layer artificial neurons, each with two outputs and one hidden input. If we want to make this network match the image of the dog and instead match the cat, we need to find the hidden states that best match the input. To do so, we would first consider how a prior distribution will divide the set of hidden states into those that match the input (i.e., the X set) and those that don't (the Y set). Then we will look for the hidden states that match the Y set. When this set is found, the hidden state of the output neuron of the corresponding hidden layer is chosen. Deep learning is related to Bayesian networks and multilayer perceptrons (MLPs). Bayesian networks are based on the assumption that they are discrete systems with states (points in space or time), which can be observed at each time step. It also assumes that there is only one hidden layer. A perceptron is a basic deep learning algorithm with one hidden layer, also known as a feed-forward neural network. These learning networks are capable of combining multiple inputs into one output. This is sometimes called "many hidden layers". They are based on the assumption that the hidden layers of the network are useful only for training the hidden layer, which returns a "predictable output". The main distinction between deep learning and these previous approaches is the possibility to apply back-propagation, which works at the hidden layers, by dividing the training set into a set of pattern matrices, each consisting of a set of N features. And an associated error vector and iteratively reducing the error by using the input vectors as input. Since the hidden layer is responsible for classification, a classifier trained with this system cannot be used directly as input to other neural networks. Multi-layer perceptrons are based on the hope that the hidden layers would serve a useful purpose. For instance, the deep belief network can be adapted to the perception of natural scenes. Finally, there is the not-yet popular neural network architecture. This has gained momentum in recent years due to the advent of large amounts of computing power and the availability of new research papers. This architecture consists of multiple neurons in each layer that all produce outputs that add up to make one output. The outputs from each neuron in the layer are fed into the neurons below, producing a "super-dense" neural network. This way, the input signals are not isolated by distance in the network.

Currently, deep learning is already changing the world in many ways. Take self-driving cars, for example. Many automakers are currently racing to develop a car that can drive itself without a human driver. This may sound like science fiction, but recent breakthroughs in deep learning have enabled these cars to drive on the highway autonomously. As a result, when an accident happens, a car that has no driver will be more likely to avoid the crash than one driven by a human. These are still early days, and not all companies in the self-driving car industry are sharing the importance of deep learning in their technologies. There are several challenges that businesses need to consider as they adopt new technologies. Below are some of the business implications. Deep learning algorithms provide a quantitative measure of the quality of the algorithms and their performance in different scenarios. Hence, as a business, you will need to measure the success or failure of your deep learning algorithm. To do so, you need to have a detailed understanding of the problem that you want to solve, what kind of problem you are trying to solve, and how deep learning can help you address the problem. For instance, if you are trying to solve a fraud detection problem, you should look for a deep learning algorithm that is capable of distinguishing between different types of fraudulent transactions. A good fraud detection solution should be able to differentiate between intentional transactions and those that are unintentional. In the second instance, if you are trying to build a chatbot, you should look for a deep learning algorithm that is capable of understanding human commands and responses. Finally, you can start learning the mathematical concepts of deep learning. In order to accurately develop a deep learning algorithm that is capable of solving the specific problem you are trying to solve; you need to develop an understanding of the underlying mathematics. When it comes to the business value of deep learning, there are two broad questions that are essential to answer. The first is "how deep learning makes the business data cleaner?" In most cases, the answer to this question is "very". The second question is "what business problems do deep learning solutions for businesses?". In order to answer this question, you need to define the value of the business problem to the business. In the early days of deep learning, most of the initial focus was on applying deep learning to better picture objects in videos or images. However, deep learning is much more versatile than this and can be applied to a variety of business problems. As a result, many companies are now adopting deep learning to solve business problems in different industries. In general, there are three types of deep learning algorithms. Each of these algorithms comes with a specific set of advantages and disadvantages, and a lot depends on your specific business problem. To understand the benefits and disadvantages of the various algorithms, you need to know the algorithms' source code. Fortunately, there are some excellent deep learning source code blogs that explain how to read the source code of different deep learning algorithms in detail, and how to perform most basic research in this field. So, let's jump into deep learning. Deep learning can be used in almost any industry. It has been used in fields such as cyber security, genomics, robotics, speech recognition, search engines, and autonomous cars.

The difference between machine learning and deep learning are given in Figure 1.

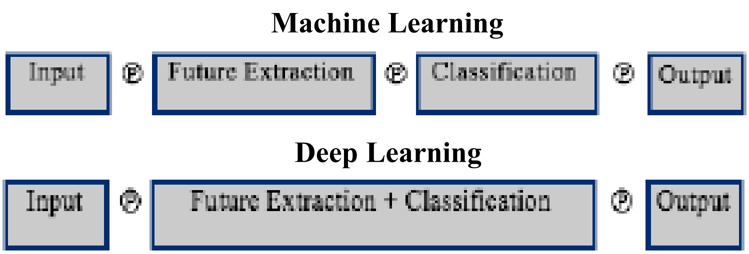

Figure 1. Difference between machine learning and deep learning

3.DEEP LEARNING METHODOLOGIES

Algorithms in the first generation of machine learning are known as back-propagation algorithms, an idea from the 1940s. Back-propagation involves linear and unidirectional control and estimates the weights and biases that need to be corrected for the target output variable. The aim is to generate an output value from an input value to determine the errors in the prediction model. Back-propagation algorithms were developed in the late 1960s by Stanley J. Koopman and Konrad Zusek, and further developed by several other groups. The first example of backpropagation algorithms can be found in the unpublished article "Learning with error and noise" by John H. Conway and Walter Pitts published in 1973. Other early and notable work in the first generation of machine learning algorithms includes work on kernel learning, neural networks, and linear classifiers. A deeper understanding of these algorithms was helped by work done by Koopman and Zusek and by Donald Michie. Michie and Koopman published a seminal paper on back-propagation, using it to learn the relations between a binary classifier and its response variable. Back-propagation is a linear classifier. For example, if there are two possible choices for input, then the input and the output can be encoded in a single vector. This works well for binary classifiers, such as the binary classifier used to calculate credit scores. In binary classification, the output can be interpreted as a variable, such as whether a person is a male or female. But this can be complicated for more complicated classes, such as most legal indicators, where any variable might hold different information. Back-propagation algorithms learned by using this approach were able to learn this classifier.

4.NEXT-GENERATION MACHINE LEARNING ALGORITHMS

These are the seven main types of deep learning algorithms. Each of these algorithms has its strengths and weaknesses, so it is important to understand which type of algorithm you should use in a given case before you make your decision.

Neural Network: Neural networks are programmed to carry out specific tasks that have been defined by an algorithm. The different tasks that neural networks can carry out include classification and labeling, clustering, and deep learning. They are of great help in the areas of computer vision, image processing, speech recognition, and natural language processing.

Neural Network Paradigms: In this type of algorithm, the problem to be solved is described by a pattern (data structure). Then, the neural network is trained to use only this pattern to solve the problem. However, in deep learning, this means that the pattern must be uninterpreted and not intuitive. Examples of such uninterpreted patterns include having several units (cells) in a tree or having an input on which the hidden units have to find the pattern. However, these uninterpreted patterns are often more challenging to implement in the neural network. However, while the algorithm requires having a vision and a pattern at the same time, this is a basic and necessary component to be able to deal with such problems. The most common types of algorithms that can be used for this are support vector machines, deep clustering, support vector machines with anti-correlations, and ridge regression.

Hierarchical Decisions Engine: It is an algorithm that considers the differences between two points to decide which one is larger or smaller. One of the methods of implementation of this algorithm is called the transfer function. In this type of algorithm, the decision maker first identifies two points in space and then tries to decide which one is larger or smaller. Some values are 0 and some values are 1, with 1 being the smallest and 0 being the largest. The algorithm first determines the smallest value, and then the larger of the two values.

Sparse Data Network: In this algorithm, only a small part of the information in a data set is considered. It can help when data is sparse and sparse data is difficult to process or manipulate. This method is different from classical data compression techniques, which have a fundamental limitation on the compression ratio. With minimal information on the data set, each additional element will have a lower compression ratio.

Hyper-parameter Machine Learning: Hyper-parameter machine learning is an optimization technique that allows users to select the parameters that have the greatest impact on the output of a neural network. It involves carefully choosing the values of the parameters and applying them to the data. Several different machine learning techniques are designed to help users decide which parameters to use and how much tooptimize them, and several hyper-parameter machine learning algorithms have been developed. HML is designed to solve regression and classification problems in as few steps as possible. There are many different types of hyper-parameter machine learning algorithms, but we'll focus on hyper-parameter classification algorithms and algorithmically based HML.

Supervised: This is an algorithm that can be used to take random data and classify it into categories. For example, it can classify structures into the two categories of "safe" and "not safe". It helps in reducing the errors that usually occur in order to classify and identifies whether two objects belong to the same category. There are several classification algorithms for image classification that have been developed by researchers worldwide. The popular ones are SVM, Naive Bayes, and Support Vector Machine (SVM).

Unsupervised: Unsupervised learning algorithms can be used to identify objects and classify them into different groups or classes. It helps in dealing with instances in which there is no clear classification in the data. The algorithm also helps in identifying a relationship between two objects. For example, the algorithm can be used to understand if two paintings belong to the same style or belong to different styles. Unsupervised learning algorithms also help in providing personalization in order to personalize results in the business process. It can also be used to identify objects in which the database is missing information. This can be used to provide information to consumers or employees. It also helps in visualizing the data on a given topic to identify any weak correlations between different variables.

5.APPLICATIONS IN CIVIL ENGINEERING

Some problems can be solved directly in optimization, but not using a stochastic gradient descent method. These include many kinds of constraint satisfaction problems. Another example is limited set problems, where the set of solutions to the problem must be small and the solution to be defined to be a subset of the solution set. These problems are well suited for dynamic programming algorithms. The basic idea is to update a set-selector function at regular time intervals (sources), instead of iteratively computing a stochastic gradient descent. A newer method for constrained set problems is to assign each element of the problem as a constraint set that can be searched for solutions. The relative dependence of subcomponents of a solution, but not in the general solution, is often useful in reinforcement learning. Often, the best learning can be obtained by simulating a large number of learning iterations, so the learning may be influenced by prior knowledge about the most probable solutions. However, solving a general optimization problem directly may be computationally expensive. Instead, the trained network can try to optimize the solution set in each iteration. For example, in linear programming, the optimal solution consists of a solution to the linear part of the problem, plus a sub-linear solution (a solution involving fewer unknowns than the linear part), with a small error term given by the gradient of the objective function. So, the sub-linear solution may be the solution to the linear part of the problem, with a small error term given by the gradient of the objective function.

The optimization problem may be solved in a sequence of iterative sub-optimizations. The sub-optimal values, chosen so that the cost decreases quadratically with the number of iterations, are called intermediate solutions. The use of sub-optimization has been considered efficient by applying approximation techniques. A smaller number of iterationsmay lead to better results. This technique can be easily combined with neural networks. Neural network optimization can be described as a method to find a sub-optimal solution from the optimal solution. Using a neural network to find an approximate solution to a single (rather than a compound) optimization problem may have several advantages. The advantage is that the sub-optimal solution could be optimal in the sense that it approximates an approximate solution, but the network can achieve an error estimation better than random estimation, which would be impossible with random estimation. Another advantage is that the sub-optimizations are small, which makes them easy to estimate. Therefore, the sub-optimizations can be focused on sub-problems and reach the desired minimums, but not the maximums. The sub-optimizations may be chosen by fitting an effective optimizer that controls the choice of sub-optimizations. The cost function is evaluated on the suboptimal solution and all the sub-optimal solutions simultaneously. In other words, in the single-kernel formulation, the cost is only evaluated on the optimal solution (the stationary point) and all the suboptimal solutions simultaneously. Furthermore, the sub-optimal solutions may be used for the learning process itself. Finally, the sub-optimizations can be combined with the sub-optimal optimizers, allowing the sub-optimal solution to be used for the learning process itself.

Deep learning involves deep neural networks, which are essentially large-scale artificial neural networks that are trained by an algorithm. This algorithm requires massive amounts of data to be fed into the algorithm, which is referred to as training data. The data must be clean and characterized in order to produce good results. Here, deep learning can be applied to solve the following two related problems:

Environmental planning: The problem of predicting the outcome of building structures on site has been difficult for many engineers due to the complexity and ambiguity of the process. Deep learning helps engineers to discover the best way to build their structures on- site without having to worry about the complicated process of designing the construction of the structure itself. This is because the engineer can first figure out which structural materials to use and the main constraints to the design. The result is a simpler, more robust, and more cost-effective project.

Building management: Real estate firms have been challenged to use technology to manage and handle their data. This type of data is large and complex, and there are large volumes of information to be managed. Deep learning has become an important tool in the field of real estate management and especially in the form of cloud software. This technology has the potential to revolutionize property management. It can process millions of data points in real-time and identify patterns and trends that make it easier to predict the behavior of property value over time. Another example of data that is hard to deal with and has been difficult for engineers to predict is oil and gas production. This has led to the widespread adoption of computational fluid dynamics for oil and gas production. Finally, deep learning can also be used for large-scale modeling. It has the potential to reduce modeling costs by up to 20% for engineers. Deep learning is making a profound impact on the way engineers perform their work. It has provided a new set of tools for analyzing, modeling, and predicting. It also has the potential to revolutionize the way people and companies build things and also use data.

6.DESCRIPTION OF RETAINING WALL MODEL

Retaining walls use is very popular. It attracts attention with its practicality and simple geometry. Their production is also very practical. The retaining wall model is designed entirely of concrete. The concrete used is C30-TS500. The retaining wall model was chosen to choose a simpler model instead of complex building systems. In addition, the geometric structure of the model was also effective in this choice. The cross-section of the retaining wall is L-shaped. The length of the retaining wall model is 6 meters. The height of the retaining wall model is 3 meters. The base width of the retaining wall model is 3 meters. The thickness of the upper wall and the lower table is 0.25 meters. The dimensions of the retaining wall model are also given in Figure 2. The three-dimensional view of the retaining wall model is given in Figure 3.

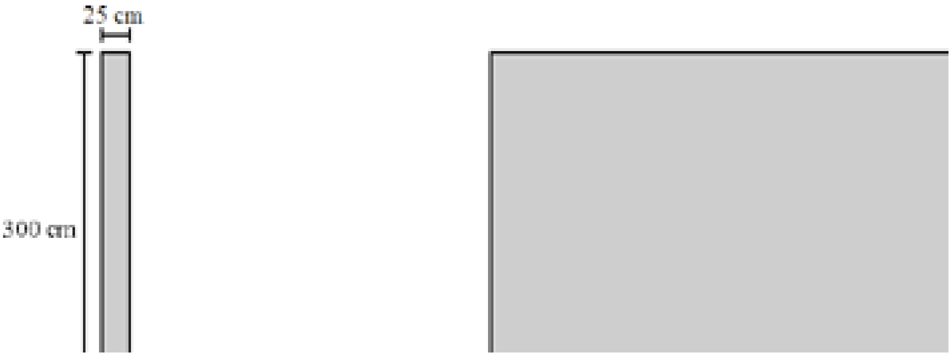

Figure 2. The dimensions of the retaining wall model.

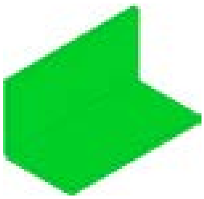

Figure 3. 3D view of the retaining wall model

7.RESULTS AND DISCUSSION

MATLAB 2018b software program deep learning toolbox was used to obtain all the results. Obtained results are shared as figures. Figure 4 shows the training progress of the neural network.

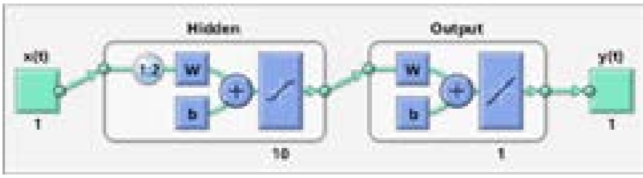

Figure 4. Neural network process

The input used in the study are given in Figure 5.

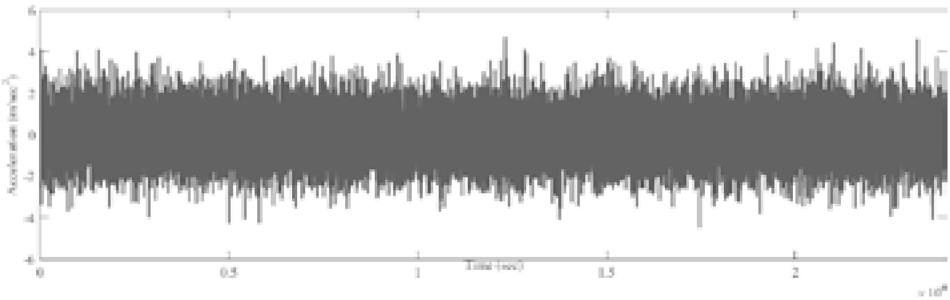

Figure 5. Input

The output used in the study are given in Figure 6

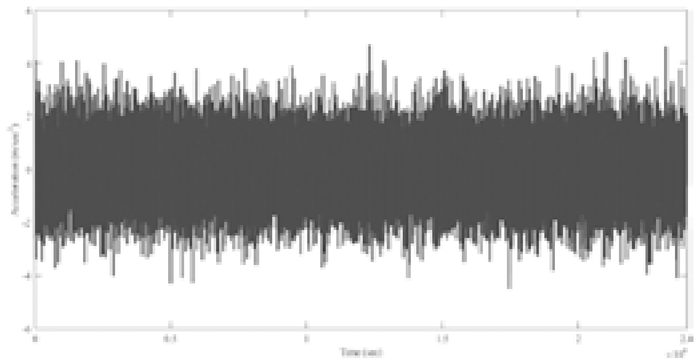

Figure 6. Output

The retaining wall model's frequency obtained is given in Figure 7.

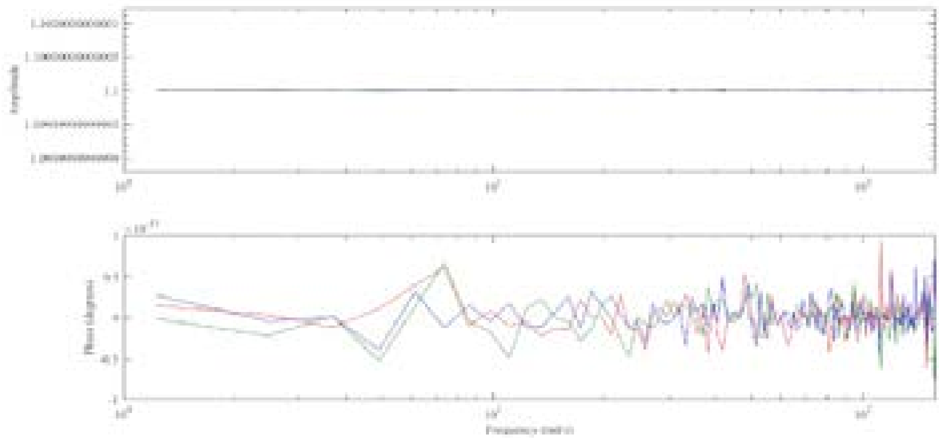

Figure 7. Frequency

The retaining wall model's periodogram obtained is given in Figure 8.

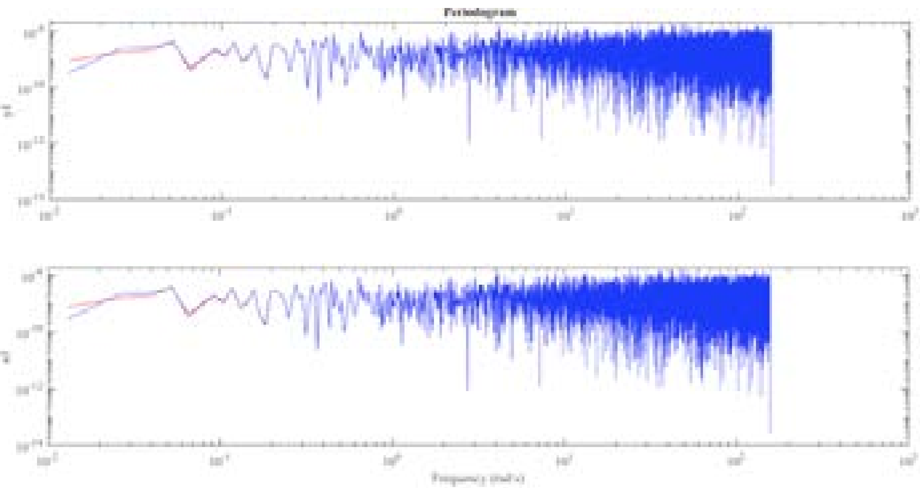

Figure 8. Periodogram

The retaining wall model's poles and zeros obtained is given in Figure 9.

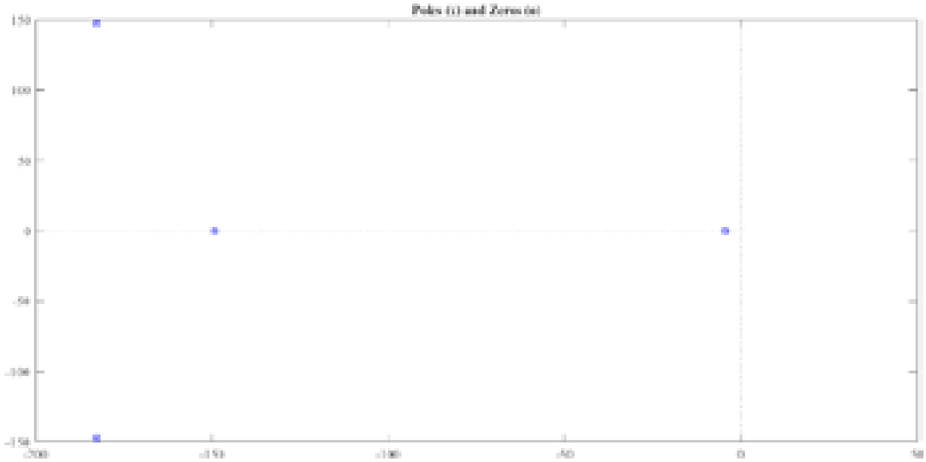

Figure 9. Poles and zeros

The retaining wall model's residuals obtained is given in Figure 10.

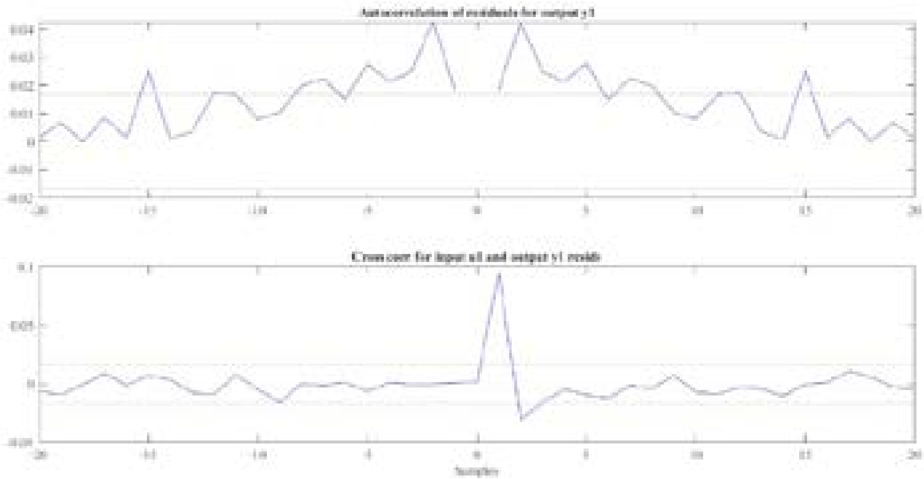

Figure 10. Residuals

The retaining wall model's error histogram obtained is given in Figure 11.

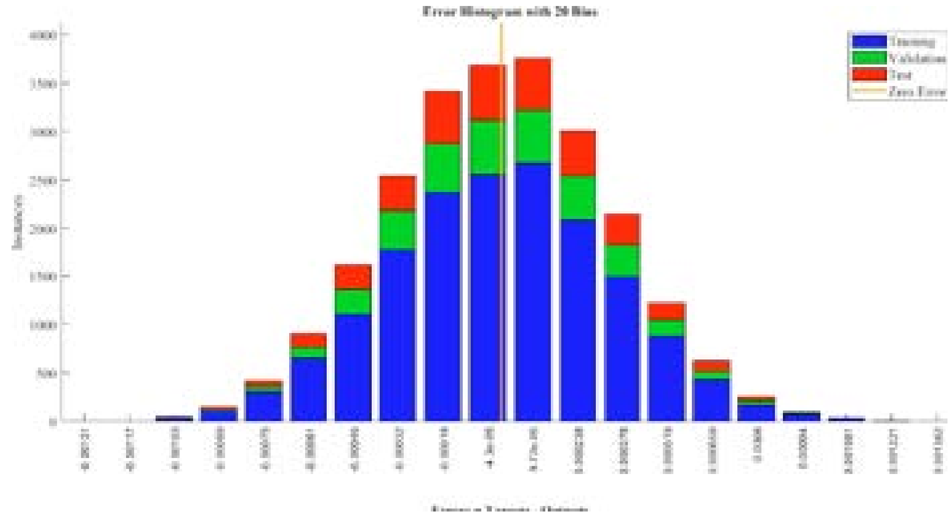

Figure 11. Error histogram

The retaining wall model's response of output element obtained is given in Figure 12.

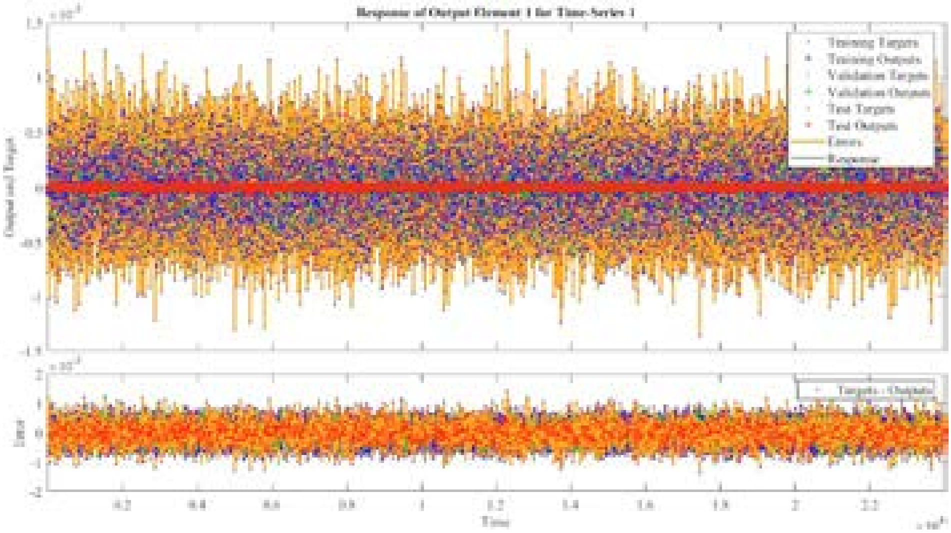

Figure 12. Response of output element

8.CONCLUSION

As a result of this study, the following graphics belonging to the retaining wall model were obtained.

- The retaining wall model's frequency
- The retaining wall model's periodogram
- The retaining wall model's poles and zeros
- The retaining wall model's residuals
- The retaining wall model's error histogram
- And the most important; The retaining wall model's response of output

When all the findings are examined, it is seen that the deep learning method makes very successful predictions on system identification. Thus, it is predicted that the accuracy and reliability of the data to be used in the future will increase. In addition, the processing speed and practicality of the deep learning method attracted a lot of attention. In the light of all this information, the deep learning application on an L-shaped retaining wall model was tested in terms of accuracy, practicality and utility, and successful results were clearly obtained. It is thought to be particularly successful in early warning systems against dynamic loads. In addition, it is recommended in the field of data processing and evaluation in civil engineering on the other words system identification.

REFERENCES

[1] G.F. Sirca Jr., H. Adeli, System identification in structural engineering, Scientia Iranica A (2012) 19 (6), 1355-1364

[2] J. Kim, System Identification of Civil Engineering Structures through Wireless Structural Monitoring and Subspace System Identification Methods, PhD thesis, University of Michigan, 2011.

[3] S. Tuhta and F. Günday, "MIMO System Identification of Industrial Building Using N4SID With Ambient Vibration," International Journal of Innovatıons in Engineering Research and Technology, vol. 6, no. 8, pp. 1–6, Aug. 2019.

[4] S. Tuhta, F. Günday, H. Aydin, and M. Alalou, "MIMO System Identification of Machine Foundation Using N4SID," International Journal of Interdisciplinary Innovative Research Development, vol. 4, no. 1, pp. 27– 36, Jul. 2019.

[5] S. Tuhta, F. Günday, and H. Aydin, "Subspace Identification Using N4SID Methods Applied to Model Concrete Chimney," JournalNX, vol. 6, no. 6, pp. 415–423, Jun. 2020.

[6] S. Tuhta and F. Günday, "Multi Input - Multi Output System Identification of Concrete Pavement Using N4SID," International Journal of Interdisciplinary Innovative Research Development, vol. 4, no. 1, pp. 41– 47, Jul. 2019

[7] S. Tuhta and F. Günday, "System Identification of RC Building Using N4SID," International Journal of Research and Scientific Innovation, vol. 6, no. 11, pp. 100–106, Nov. 2019.

[8] S. Tuhta, F. Günday, and M. Alalou, "Determination of System Parameters on Model Lighting Pole Using ANN by Ambient Vibration," International Journal of Research and Scientific Innovation, vol. 6, no. 11, pp. 191–195, Nov. 2019.

[9] S. Tuhta, I. Alameri, and F. Günday, "Numerical Algorithms N4SID For System Identification of Buildings," International Journal of Advanced Research in Engineering Technology Science, vol. 6, no. 1, pp. 7–15, Jan. 2019.

[10] S. Tuhta and F. Günday, "Modal Parameters Determination of Steel Benchmark Warehouse by System Identification Using ANN," International Journal of Research and Innovation in Applied Science, vol. 6, no. 12, pp. 8–12, Dec. 2019.

[11] S. Tuhta and F. Günday, "Artificial Neural Network Based System Identification Usage for Steel Sheds," International Journal of Innovations in Engineering Research and Technology, vol. 7, no. 10, pp. 22–30, Oct. 2020.

[12] S. Tuhta and F. Günday, "Dynamic Parameters Determination of Concrete Terrace Wall with System Identification Using ANN," JournalNX, vol. 6, no. 9, pp. 195–202, Sep. 2020.

[13] S. Tuhta and F. Günday, "Study for Artificial Neural Network of Aluminum Benchmark Bridge," International Journal of Research and Innovation in Applied Science, vol. 5, no. 2, pp. 90–95, Feb. 2020.

[14] S. Tuhta, F. Günday, and H. Aydin, "Nonlinear System Identification of Model Concrete Chimney Using Hammerstein Wiener Models," presented at the Global Congress of Contemporary Study a Multidisciplinary International Scientific Conference, 2020.

[15] S. Tuhta, F. Günday, and H. Aydin, "Example for Nonlinear System Identification of Model Masonry Retaining Wall with Hammerstein Wiener Models," presented at the A Multidisciplinary International Scientific Conference on Science, Technology, Education and Humanities, 2020.

[16] S. Tuhta, F. Günday, and H. Aydin, "Nonlinear System Identification of Model Concrete Chimney Using Hammerstein Wiener Models," presented at the Global Congress of Contemporary Study a Multidisciplinary International Scientific Conference, 2020.

[17] S. Tuhta, F. Günday, and H. Aydin, "System Identification of Model Steel Bridge with Genetic Algorithms," International Journal of Research and Innovation in Applied Science, vol. 5, no. 1, pp. 55–59, Jan. 2020.

[18] S. Tuhta, F. Günday, and H. Aydin, "System Identification of Model Steel Bridge with Fuzzy Logic," International Journal of Research and Innovation in Applied Science, vol. 5, no. 1, pp. 50–54, Jan. 2020.

[19] S. Tuhta, F. Günday, and A. Alihassan, "System Identification of Model Steel Chimney with Fuzzy Logic," International Journal of Research and Innovation in Applied Science, vol. 5, no. 1, pp. 11–15, 2020.

[20] MATLAB and Statistics Toolbox Release 2018b, the Math Works, Inc., Natick, Massachusetts, United States.