Geomechanics and Engineering, *Vol. 37, No. 3 (2024) 263-277*
https://doi.org/10.12989/gae.2024.37.3.263

Optimizing artificial neural network architectures for enhanced soil type classification

Yaren Aydın[1], Gebrail Bekdaş*[1], Ümit Işıkdağ[2], Sinan Melih Nigdeli[1] and Zong Woo Geem[3]

[1]Department of Civil Engineering, Istanbul University-Cerrahpaşa, 34320 Istanbul, Turkey
[2]Department of Informatics, Mimar Sinan Fine Arts University, Istanbul 34427, Turkey
[3]College of IT Convergence, Gachon University, Seongnam 13120, Korea

(*Received January 21, 2024, Revised April 21, 2024, Accepted April 22, 2024*)

Abstract. Artificial Neural Networks (ANNs) are artificial learning algorithms that provide successful results in solving many machine learning problems such as classification, prediction, object detection, object segmentation, image and video classification. There is an increasing number of studies that use ANNs as a prediction tool in soil classification. The aim of this research was to understand the role of hyperparameter optimization in enhancing the accuracy of ANNs for soil type classification. The research results has shown that the hyperparameter optimization and hyperparamter optimized ANNs can be utilized as an efficient mechanism for increasing the estimation accuracy for this problem. It is observed that the developed hyperparameter tool (HyperNetExplorer) that is utilizing the Covariance Matrix Adaptation Evolution Strategy (CMAES), Genetic Algorithm (GA) and Jaya Algorithm (JA) optimization techniques can be successfully used for the discovery of hyperparameter optimized ANNs, which can accomplish soil classification with 100% accuracy.

Keywords: artificial neural networks; bio-inspired methods; hyperparameter optimization; soil classification

1. Introduction

Developing smart communities and built environment with the help of Artificial Intelligence (AI) is crucial for sustainability. AI plays a key role in optimizing resource management and improving overall quality of life. The main areas of focus where AI can facilitate the processes include energy management, transportation, waste management, water management, urban planning, public safety, healthcare, and community engagement.

In energy management, AI helps in optimization of energy consumption, predicts demand, and manages renewable sources for a more sustainable energy infrastructure. In transportation, AI-driven traffic management reduces congestion, fuel consumption, and promotes eco-friendly modes of transport. Waste management benefits from AI optimization, improving waste collection efficiency, sorting materials, and encouraging sustainable disposal practices.

Water management benefits from AI through analyzing water consumption, detecting leaks, and optimizing irrigation, contributing to efficient water usage and conservation. In urban planning, AI-driven data analysis aids in planning sustainable cities, optimizing layouts, green spaces, and energy-efficient buildings. Public safety benefits from AI surveillance and predictive modeling, enhancing real-time threat detection and optimizing emergency response for disaster resilience.

*Corresponding author, Professor
E-mail: bekdas@iuc.edu.tr

AI in healthcare supports predictive analytics, early disease detection, and personalized health insights for a healthier population. Community engagement is facilitated by AI platforms, which gather citizen feedback and promote sustainability initiatives through effective communication.

Sustainable Construction forms the key element of the Sustainable Built Environment. As all structures interact with soil, information about the type and characteristics of the soil is a crucial element for design, construction and maintenance of the buildings. Furthermore, understanding soil type and characteristics aids in optimizing water management, vegetation planning, and minimizing the negative environmental impact of structures through more sustainable design and construction of buildings. Soil-type classification complements AI-driven initiatives, contributing to the long-term sustainability of Smart Built Environment. By the prediction of soil types via AI accurately, the need for high-cost and energy-consuming soil type determination tests may be eliminated. In addition, determining soil behavior is of great importance in terms of the safety and durability of the structure. Therefore, for all the reasons identified above, the soil characteristics, type and properties must be well-known before the design and construction of a building.

Traditional methods for soil classification are expensive and take a lot of effort and time. Therefore, new technologies and procedures are needed to produce faster and better results with improved methods. Previous research indicates that Artificial Neural Networks can perform very well in tabular classification problems, such as soil type classification. ANN is one of the newest technologies in soil classification and can be implemented to complete soil classification in a faster, more efficient and accurate manner.

http://www.techno-press.org/?journal=gae&subpage=7 ISSN: 2005-307X (Print), 2092-6219 (Online)

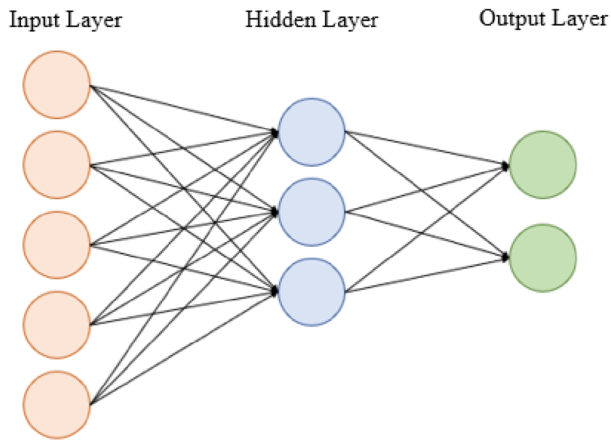


Fig. 1 ANN layers

Artificial neural networks (ANN) are one of the most widely used AI techniques and simulate the learning mechanism of the human brain. Since the ANN algorithm is capable of self-learning, a large number of data indicates that it will give results closer to reality (Elshorbagy *et al.* 2000). With the development of technology, Artificial Neural Networks are widely used in engineering fields in many different ways; fault detection (Xie *et al.* 2022), financial distress prediction (Wu *et al.* 2022) credit card fraud detection (Kasasbeh *et al.* 2022) and disease diagnosis (Ahmadian *et al.* 2022).

Physiologists McCulloch and Pitts proposed the first model of an artificial neural network model in 1943 (McCulloch and Pitts 1943). ANN is a method that imitates the working principle of neurons in the human brain. ANNs have the ability to learn from historical data. This is a powerful way to reveal complex nonlinear relationships in a data set (Arı and Berberler 2017). ANNs are complex systems in which artificial neural cells, like neurons in the human brain, are connected by different connection geometries (Topçu and Sarıdemir 2008) ANNs, which can describe complex and relationships between input and output in the data set, are flexible mathematical structures that can predict output values based on education and learning processes (Tachi *et al.* 2016). An ANN consists of many processing units called neurons and layers (Fig. 1) (Kukreja *et al.* 2016).

The most widely used model of ANNs today is multilayer perceptron (MLP) networks. These networks can produce solutions to different engineering problems (Ghunimat *et al.* 2023, Verma *et al.* 2022, Sharma *et al.* 2022, Patil-Shinde *et al.* 2023). MLP networks emerged because of the studies carried out to solve the Exclusively-OR (XOR) problem. The input layer of the MLP is the layer where information from the outside world is received and there is no processing. The hidden layers process the information from the input layer and send it to the output layer. It is possible to solve many problems with one middle layer. If the relationship between the input and output of the problem that the network is asked to learn is not linear and the complexity increases, there may be more than one intermediate layer for a network. The output layer processes the information from the hidden and finds the output produced by the network for the input presented to the network from the input layer. This output produced is transmitted to the outside world (Öztemel 2006).

Artificial Neural Networks are trained in light of a set of rules that either provide constraints or options related to the architecture of the network or regarding the training process itself. These sets of rules are termed the hyperparameters of the ANN. The purpose of hyperparameter optimization is to find the ideally near-optimal model possible, given the time and computational resources that are available for training and evaluating alternate models.

The adaptability of ANNs and their ability to remember very well the information presented while training makes them suitable and popular tools in various research fields (Zendehboudi and Saidur 2019). Adeli and Yeh published the first journal article on civil/structural engineering applications of ANNs in 1989 (Adeli and Yeh 1989, Singh *et al.* 2019). Then, many researchers have developed methods using neural networks in civil engineering problems. There are many studies using ANN in civil engineering., including transportation (Owusu-Ababio 1998, Jabamony and Shanmugavel 2019, Amarasiri and Muhunthan 2022), building materials (Zhang *et al*. 2020, Isleem *et al.* 2021, Hema *et al.* 2022, Soyer *et al.* 2022, Sridhar *et al.* 2023), seismic performance (Lee *et al.* 2022, Menon and Nair 2022), Jia and Wu 2022), hydrology (Karunanithi *et al.* 1994, Birikundavyi *et al.* 2002, Chen *et al.* 2013), geotechnic (Bai *et al.* 2021, Alzabeebee *et al.* 2022, Dinarvand and Ardakani 2022, Dehghanbanadaki *et al.* 2022, Lee *et al.* 2022, Kwak and Ko 2022, Lawal *et al.* 2022, Zhang *et al.* 2022, Esmaeili-Falak and Benemaran 2023, Fereidooni and Karimi 2023, Pereira *et al.* 2023), construction management (Cheng and Darsa 2021, Mahmoodzadeh *et al.* 2022, Anirudh and Reddy 2022, Sanni-Anibire *et al.* 2022, Alshboul *et al.* 2022), optimum design (Yucel *et al.* 2018, Yucel *et al.* 2019, Bekdaş *et al.* 2021, Yücel *et al.* 2021, Nigdeli *et al*. 2022, Yucel *et al.* 2022, Aydın *et al.* 2023).

Similar to other engineering fields, the use of ANNs has also become common in geotechnical engineering. Tinoco *et al.* (2009) used three different data mining models, including ANNs, to predict the Uniaxial Compressive Strength of Jet Grout laboratory formulations. These models were able to learn the relationships between the Uniaxial Compressive Strength of Jet Grout laboratory formulations and their contributing factors with high success. Samui and Sitharam (2010) modeled site characterization using ANN and a geo-statistical model based on the ordinary kriging technique. A comparison between the ANN and geostatistical model demonstrated that the ANN model is better than the Geo-statistical model in predicting SPT-corrected values in the subsurface of Bangalore, India. Das *et al.* (2011) discussed the application of different ANN models to slope stability analysis based on the actual slope failure database. As a result, the differential evolution neural network (DENN) results were found to be more suitable for describing the physical model. Sivrikaya and Soycan (2011) predicted compaction parameters for fine-grained soils related to compaction energy using ANN. As a result, ANN models showed higher success than regression methods. Hasanzadehshooiili *et al.* (2014) used artificial neural networks to calculate the value of collapse settlement in embankment dams. It is shown that the developed ANN is well-behaved with geotechnical engineering considerations. Yücel *et al.* (2021) developed a hybrid model artificial neural network (FPA-ANN) to predict the optimum dimensions of reinforced concrete cantilever retaining walls. According to the results, the optimum values found via FPA and the predicted results via the ANN model showed a great match for cases. Baghbani *et al.* (2022) conducted a statistically detailed review of 1235 published papers on the performance of Artificial Intelligence (AI) methods and algorithms used in geotechnical engineering. They identified the main areas where AI methods are applied in geotechnical engineering and stated that the most widely used model is Artificial Neural Network (ANN). Amâncio *et al.* (2022) developed prediction models for the shaft and end-bearing capacity of single piles using ANN. As a result, ANN predictions were closer to the bearing capacity values measured in single pile tests and pile groups. Wrzesiński and Markiewicz (2022) presented a method of application of an ANN to predict the permeability coefficient k in sandy soils. The proposed ANN predicted the true value of the permeability coefficient using the data.

In this study, ANN was used to accurately model soil classification. Correct soil classification is critical to a reliable design. After a satisfactory ANN model was developed, two similar approaches were applied and compared. These two similar approaches focus on finding the network hyperparameters to hyperparameter optimization for achieving the most accurate classification results. Suggested techniques for classifying soil type and using advanced techniques are termed the hyperparameters of the ANN. This study investigates how well two different methods of hyperparameter optimization work for optimizing ANNs to produce accurate results only for soil classification.

2. Materials and methods

2.1 The dataset

The dataset includes 805 soil sample records obtained from the ground soundings made in the Gayrettepe-Istanbul Airport underground construction, the details of the dataset can be found in (Aydın *et al.* 2023). The problems encountered during the data preprocessing phase were related to missing data and class imbalance. Different approaches were tested to tackle these problems. The missing data was observed in retaining no.4 sieve, liquid limit, plastic limit, and plasticity index variables. For the missing data problem, the missing data points were imputed with the K-Nearest Neighbor (KNN) Imputer. While performing the KNN Imputation, the k value was chosen as 10. The composition of the dataset enabled working with the data without losing the information about the rarely occurring classes. For the data imbalance problem, Synthetic Minority Oversampling Technique (SMOTE) in combination with selective sampling was used to create a new balanced class distribution (Table 1) (Aydın *et al.* 2023). Tests performed in the balancing phase showed that the prediction accuracy was positively correlated to the balance in the class distribution.

In the dataset, the predictor variables were Retaining No. 4 Sieve (for particles), Passing No. 200 Sieve (for particles), the Liquid Limit, the Plastic Limit, and the Plasticity Index, the target variable was the Soil Class. The maximum, mean and standard deviation values of the values in the data set in this study are provided in Table 2. The standard deviation determines how much of the data is close to the mean.

In our previous study (Aydın *et al.* 2023), we have observed that tree-based base classifiers and gradient boost-based ensemble classifiers have achieved good performance this dataset. The correlation matrix for a better understanding of the dataset is given in Fig. 2.

Fig. 2 shows that the highest correlation is between LL and PI. The lowest correlation is between No4 and No200. We would like to mention here that although there is a moderate/high level of dependency between LL/PI and LL/PL , it can be observed that there is no very high degree of dependence (i.e., > 0.90) between any independent variable.

2.2 Soil classification

The traditional soil classification process involves several stages including Sieve Analysis, Determination of Atterberg Limits and utilizing charts with pre-determined value ranges of soil properties. The details of these stages are explained in this section.

Table 1 Class distribution (Before/After Balancing)

Soil Classes	Number of Rows (Before Balancing)	Number of Rows (After Balancing)
SC	19	60
CI	25	60
MH	25	60
CL	169	60
CH	567	60

Table 2 Details about the collected data set in this study

Features	Type	Unit	Min	Max	Mean	Standard Deviation
Retaining No. 4 sieve	Input	%	0	29.4	0.4366	2.4988
Passing No. 200 sieve	Input	%	13	100	90.8600	12.9337
Liquid Limit	Input	%	23.1	90	53.4744	11.7505
Plastic Limit	Input	%	3.39	36.9	23.3159	29.6083
Plasticity Index	Input	%	7.3	62	30.4820	11.0532

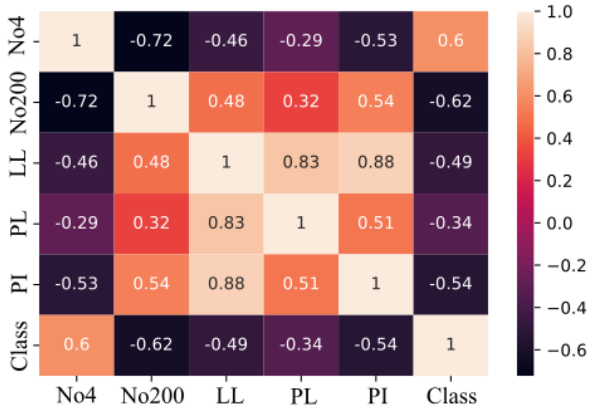


Fig. 2 Correlation matrix of dataset

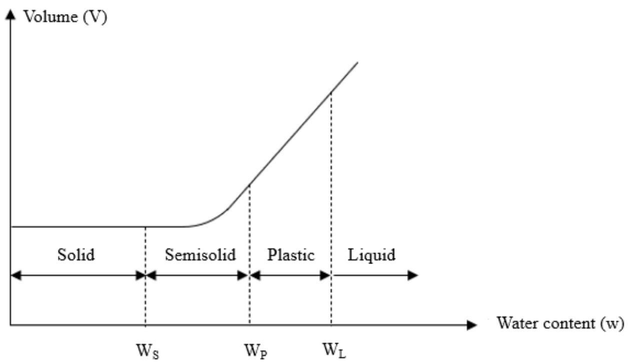


Fig. 3 Volume change and consistency limits of cohesive soils during drying (Yılmaz *et al.* 2016)

2.2.1 Sieve analysis

The grain size distribution of soils is defined as well-graded and poorly graded. Determining the grain sizes of soils, and determining the percentage of grains (gravel, sand, silt, clay); classification of soils, determination of uniformity and gradation coefficients, selection of core and filter materials in dams, etc. is important. The analyzes made to determine the grain sizes of the material of any soil and the weight ratios of these different-sized grains are called 'sieve analyses'. As a result of the sieve analyzes, the weight percent amounts of the grains of different sizes that make up the material are determined and the grain size on the logarithmic axis and the grain size on the arithmetic axis of the semi-logarithmic paper. A grain size distribution curve is drawn using percent sieve weights. By using this curve, the amounts of gravel, sand, clay and silt that make up the material are determined, and soil classification is made for different uses (Yılmaz *et al.* 2016).

Since fine and coarse-grained soils can be mixed in soils, the part with grain diameters between 76.2 mm and 0.075 mm is subjected to sieve analysis, while methods called wet analysis (hydrometer or pipette analysis) are used for soils with diameters less than 0.075 mm The soil whose granulometry curve is to be determined is dried and sieved through a series of sieves of different diameters. Sieve series given by German (DIN) and American (ASTM) standards used as standard are used for this experiment. Sieve numbers indicate the number of holes per square inch. For example, sieve #40 (425 μm) has 40 holes in 1 square inch (Yılmaz *et al.* 2016).

2.2.2 Determination of Atterberg limits

The different behavior of soils depending on the water content was experimentally described by Atterberg in 1911. The boundary water contents, defined by Atterberg, are

Table 4 ASTM D 2487:1969 Soil Classification (ASTM D2487, 1975)

Main groups		Properties	Grading	Group symbol	Group name [B]
Coarse grained soils (More than 50% bigger than 75µm)	GRAVELS More than 50% of coarse grains (2-60 mm)	Clean gravels (Less than 5% fines)	Cu>4 and 1<Cr<3	GW	Gravel[C] with uniform grain distribution
			Cu<4 or 1>Cr>3	GP	Uniform Gravel[C]
		Gravels with fines (More than 5% fines)[A]	Fines are ML,MI or MH	GM	Silty Gravel[C]
			Fines are CL,CI or CH	GC	Clayey Gravel[C]
	SANDS More than 50% of coarse grains (75 µm-2 mm)	Gravels with fines (More than 5% fines)[A]	Cu>6 and 1<Cr<3	SW	Sand[D] with uniform grain distribution
			Cu<6 and/or 1>Cr>3	SP	Uniform Sand[D]
		Sands with fines (More than 5% fines)[A]	Fines are ML,MI or MH	SM	Silty Sand[D]
			Fines are CL,CI or CH	SC	Clayey Sand[D]
Fine grained soils (More than 50% smaller than 75µm)	Silts and clays (LL<35)	Inorganic	PI>4 and plots on 'A' line	CL[E]	Low Plasticity Clay
			PI<4 and plots below 'A' line	ML[E]	Low Plasticity Silt
	Silts and clays (35<LL<50)	Inorganic	PI, plots on 'A' line	CI[E]	Medium Plasticity Clay
			PI, plots below 'A' line	MI[E]	Medium Plasticity Silt
	Silts and clays (LL>50)	Inorganic	PI, plots on 'A' line	CH[E]	High Plasticity (Fat) Clay
			PI, plots below 'A' line	MH[E]	High Plasticity (Elastic) Clay
	Contains dark, fragrant, excessive organic material			PT	Peat

A line: Line A separates clay and silt soils.
A: If the proportion of fines is between 5% and 12%, the soil is expressed with a double symbol.
B: If the soil contains rubble, B is added to the group name.
C: If there is more than 15% sand in the sample, the term 'Sandy' is added to the group name.
D: If there is more than 15% gravel in the sample, the term 'Gravel' is added to the group name.
E: If LL(dried) / LL(natural) < 0.75, the term 'O' is added

called Atterberg limits or consistency limits. If the consistency (Atterberg) limits of the soil are defined with the water content of the cohesive soils in certain physical states and evaluated well, information about many properties of the soil can be obtained. Atterberg limits are mainly 3 (liquid limit, plastic limit and shrinkage limit). Depending on the decrease in the water content of the soil, the changes described in Fig. 3 occur.

The water content required for the soil to flow under its weight is called the liquid limit. It can be obtained in two different ways, the Casagrande method or the falling cone method. In the Casagrande method, the liquid limit is defined as the water content required to close a standard cavity in the soil sample in the standard tested (ASTM) Casagrande instrument in 25 strokes. Laboratory definition of plastic limit; It is the water content in which the ground takes the form of a cylindrical pasta with a diameter of 3 mm without breaking. The purpose of the plastic limit test; is the determination of the lowest water content where the soil has a plastic consistency yet. The shrinkage limit of soil is the water content of the soil at the moment when more water loss does not result in further volume reduction (Yılmaz *et al.* 2016).

Table 3 Plasticity classification of soils (Yılmaz *et al.* 2016)

Plasticity index (%)	Definition
<1	Non-plastic
1-7	Low plasticity
7-17	Medium plasticity
17-35	High plasticity
>35	Extreme plasticity

2.2.3 Plasticity index

Soil Classification is conducted taking into account both qualitative and quantitative and/or semi-quantitative properties of soils. For instance, the plasticity index (PI) of soils is a very important parameter used in the soil classification. The consistency limits of soils and the plasticity index (PI=LL-PL) determined according to these consistencies are widely used especially in the classification of clays. The classification of soils according to the plasticity index is given in Table 3.

2.2.4 Soil classification charts

The chart is used in the classification of fine-grained soils (material below sieve 200 > 50%). The classes defined by the terms in the chart are given in Table 4.

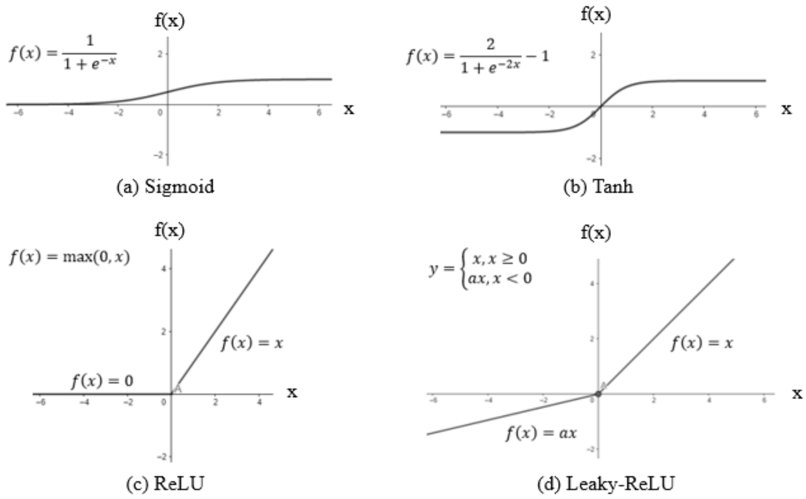


Fig. 4 Activation functions (Karlik and Olgac 2011)

2.3 ANN model structure

ANNs are structures inspired by neurons in the brain. Soma, dendrite, axon and synapse in the biological nervous system correspond to neurons, inputs, outputs and weights, respectively. The artificial neural network produces the output by collecting the information coming to the nerve cells and passing it through the determined activation function. When we look at the neuron structure, the nodes where the inputs are kept, the weights of these inputs, and the additional function obtained by multiplying all the inputs and weights, consist of an activation function that produces results by using the values from the addition function, and the output (Gülcü and Kuş 2019). In this research, we focused on generating ANNs composed of 1,2, or 3 hidden layers. Depending on the architecture each layer of the network could contain up to 512 neurons in our experiments.

2.3.1 Activation functions

As a result of matrix operations in artificial neural networks, the ANN and its elements are linear (İnik and Ü lker 2017).

The resulting linear structure is converted into a non-linear structure by activation functions. This allows the network to be derived easily. Selecting an easily distinguished activation function is very important to accelerate the calculation of the network (Gülcü and Kuş 2019). Widely used activation functions in ANNs are shown in Fig. 4. The graphics shown in Fig. 4 are drawn via GeoGebra (GeoGebra Graphing Calculator).

Sigmoid Activation Function: Sigmoid, which is frequently used in classification problems in artificial neural networks, converts incoming inputs into outputs between 0 and 1 as shown in Fig. 4(a) (Gülcü and Kuş 2019).

Log-Sigmoid Activation Function: This activation function is based on the logarithm of Sigmoid Activation Function.

Tangent Hyperbolic Activation Function (Tanh): The tangent hyperbolic activation function converts inputs to outputs between -1 and 1. As shown in Fig. 4(b), the outputs in the range [-1, 1] are zero-centered and this function and the sigmoid activation function have the problem of approaching the zero of the derived elements after a certain point (Gülcü and Kuş 2019).

Rectified Linear Unit (ReLU): As shown in Fig. 4(c), the ReLU activation function converts inputs into outputs between zero and infinite. Therefore, ReLU is called a function that does not reach satisfaction. The main advantage of this function is that the calculation is very fast and tolerates the missing gradient problem (Gülcü and Kuş 2019).

Exponential Linear Unit (ELU): ELU is very similar to RELU for positive in-puts(f(x)=x). In fact if the input x is negative, the output is f(x)=exp(x) – 1 in the ELU function.

Leaky Rectified Linear Unit (Leaky ReLU): ReLU is called dead ReLU because it equates negative values directly to zero. This prevents the units in the hierarchy from entering the hierarchy when the negative value is high. To prevent such situations, a leaking relu activation function has been proposed. As shown in Fig. 4(d), a small fixed parameter (ai) is selected to add a gradient of negative values to the function. If the ai parameter receives a small positive value (for example, 0.01)., the method is called Leaky-Relu (Gülcü and Kuş 2019).

Mish: The formula of Mish function is f(x)=x.tanh.softplus(x) where softplus(x) is defined as ln(1+ex) . The Mish is known as a self-regularized non-monothonic activation function.

In our experiments, we have implemented LeakyReLU, Sig-moid, Tanh, ReLU, LogSigmoid, ELU, Mish as activation functions. These activation functions are implemented using Python PyTorch (Paszke *et al.* 2019) package.

2.4 Evaluation strategies

Train-Test Split is a frequently used method to evaluate and validate the performance of models. This method

Table 5 Confusion matrix

Predicted \ Actual	True	False
Positive	True Positive (TP)	False Positive (FP)
Negative	False Negative (FN)	True Negative (TN)

divides the data set into training and test subsets. Dividing the data set into two parts in training and test allows the model to be trained on the training data and then its performance can be measured on the (unseen) test data. (Su *et al.* 2020). If the model is evaluated by randomly splitting the data into two parts(i.e., split1) , when training with another random split(i.e. split2), the model has a very low probability of providing the same accuracy as the previous split(i.e. split1), as the model will be trained and tested by different parts of the data. So the accuracy obtained by each train/test split(i.e. split1,split2…) will be different, and thus train/test evaluation approach is not considered as a very reliable one.

Cross Validation is a technique used to overcome this barrier. (Li *et al.* 2023). When k-fold cross-validation is used for evaluation, the model is evaluated in k rounds. This is done by firstly dividing the data into k groups, groups from 2…k are used as the training set and group 1 is used as the test set in round 1, in round 2, groups from 1,3….k are used as training set and group 2 is used as test set, and finally in round k, groups from 1….k-1 are used as the training set and group k is used as the test set. The model's accuracy is then determined as the mean of accuracy values obtained from all test sets in k rounds. In k-fold cross-validation, as all data in the dataset is used both for training and testing, this allows for better evaluation of the performance of the model (Yadav *et al.* 2016). In this study, the k-fold cross validation is chosen as the validation strategy, and the k value was chosen as 10, and thus 10-fold cross-validation was implemented as the model evaluation strategy. We have used mean values of true and false positives /negatives for 10-folds obtained from the tests in the construction confusion matrices and we have calculated accuracy values based on these matrices. In order to evaluate the performance of the developed model, firstly confusion matrices were used. A confusion matrix is used to correlate the predicted values of the model with the actual values. A confusion matrix basically consists of four basic index values. These are true-positive (TP), true-negative (TN), false-positive (FP) and false-negative (FN). TP represents the number of instances predicted by the model that are actually positive, while FP corresponds to the number of instances that the model predicts as positive but are actually negative. Similarly, TN represents the number of instances that the model predicts as negative and are actually negative, while FN corresponds to the number of instances that the model predicts as negative but are actually positive (Ruuska *et al.* 2018).

The metric considered when analyzing the performance of the classifiers used in the study is Accuracy. From the frequencies mentioned above (TP, FP, TN and FN), one can calculate classification performance indicators that reflect how the classifier performs in detecting the given class. The accuracy value is the ratio of correctly predicted data to all predicted values. Calculation of the accuracy is shown in Eq. (1) (Ruuska *et al.* 2018).

$$\boldsymbol{Accuracy} = \frac{\boldsymbol{TP + TN}}{\boldsymbol{TP + TN + FP + FN}} \quad (1)$$

2.5 Hyperparameter optimization

One of the difficulties encountered in artificial neural networks is choosing the most successful network model for the problem. In order to obtain successful results in artificial neural networks, the hyperparameters to be used must be selected correctly. However, since the optimization of hyperparameters requires high processing power, it is very difficult in practice to evaluate all possible possibilities for hyperparameter values in experiments to be carried out. For this reason, meta-heuristics are preferred for hyperparameter optimization In machine learning, deep learning and artificial neural networks, hyperparameter optimization or tuning is the problem of choosing a set of optimal hyperparameters for the learning algorithm. The value of a hyperparameter is a parameter used to control the learning process, and the values of other parameters (typically node weights) are learned (Atlan *et al.* 2020, Karabaş 2019).

The same type of machine learning or deep learning model may require different constraints, weights or learning rates to generalize different data patterns. These are called hyperparameters and need to be adjusted so that the model can best solve the machine-learning problem (Karabaş 2019). Many architectural features of an ANN (number of hidden layers, number of nodes per layer, etc.) are considered hyperparameters that can be tuned as part of the overall hyperparameter optimization process.

One of the goals of machine learning is to build a model that works well and makes accurate predictions. Therefore, the machine learning algorithm needs to be optimized. Optimization in machine learning is the process of optimizing hyperparameters using an optimization technique to minimize monetary costs, time costs. Hyperparameter optimization is the search process performed to determine the best fit among the defined value range for a model's parameter in order to reach a better estimation result. The aim is to determine the optimal values for the model parameters and thus improve machine learning. In other words, parameter optimization refers to optimizing the parameters of the default model to achieve better performance (Chon 2019).

In this study a tool developed by the authors, (HyperNetExplorer) which utilizes several (Covariance Matrix Adaptation Evolution Strategy (CMAES)", "Genetic Algorithm (GA)" and "Jaya Algorithm (JA)" optimizers for hyperparameter optimization were investigated.

The structure of the model is defined by hyperparameters. The performance of each combination of hyperparameter values is determined by evaluating a model's performance after it has been trained. In a classification algorithm, hyperparameters are used with their defined default values. Finding the optimal value for these parameters offers better results and efficiency (Chon 2019).

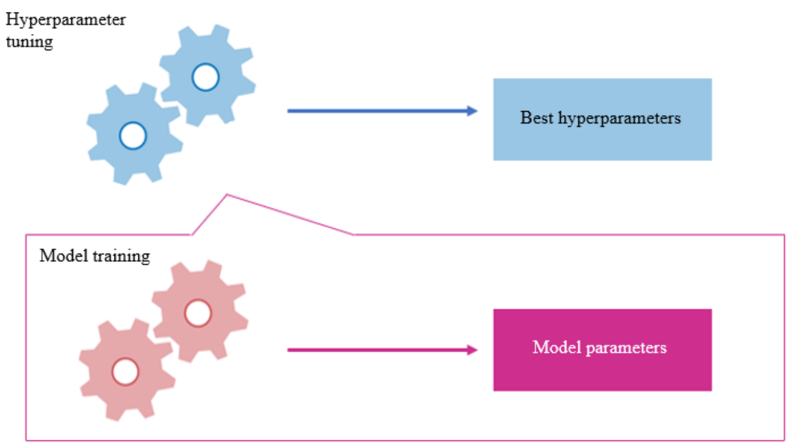


Fig. 5 Performing hyperparameter optimization on the model (Ramli *et al.* 2020)

Hyperparameters are parameters that differ according to the data set and model used to solve a problem. The purpose of hyperparameter optimization is to optimize the result obtained from the desired success criterion in any neural network model and many different types of machine learning algorithms. Choosing the right model and/or choosing hyperparameters is not enough to build a good model. It is also necessary to find and know the combination that will give the best result among these hyperparameters (Atlan *et al.* 2020).

Hyperparameter optimization which is a meta-optimization task is shown in Fig. 5. The training of a model - an inner optimization process is involved with each trial of a particular hyperparameter setting. The outcome of the hyperparameter optimization process is the best performing set of hyperparameter values among all of the various combinations that were considered. Also, the outcome of model training is the best model parameter setting (Zheng 2015). Hyperparameter optimization is a process in which the most appropriate model parameters are determined and the best hyperparameter values are determined by means of machine learning (Cinaroglu and Baser 2020).

2.5.1 Metaheuristics used for hyperparameter optimization

Since optimization of hyperparameters requires high processing power, it is very difficult from a practical point of view to evaluate all possible possibilities for hyperparameter values in experiments to be carried out. For this reason, metaheuristics are preferred for hyperparameter optimization (Gülcü and Kuş 2019). When a neural network is to be trained, an optimizer is needed to help minimize the objective function and achieve the best possible neural network. In this section, 3 metaheuristic optimization algorithms are briefly summarized for the problem.

2.5.1.1 Covariance Matrix Adaptation Evolution Strategy (CMAES)

Covariance Matrix Adaptation Evolution Strategy (CMAES) was introduced by Hansen and Ostermeier (1996). The E in the name of this algorithm stands for evolution and is actually an evolutionary algorithm that works by mimicking the way evolution works in species. After the children are selected according to the objective function, the first individuals with the lowest fitness value form the next generation population. It starts with the first generation to find the best solution and randomly selects solutions from the first generation. If this were a species, then the species would undergo natural selection and the weakest members would die out. The weak and strong members of the new generation of survivors would undergo natural selection and the weak members would die. The result is achieved when the maximum iteration is met (Chen *et al.* 2023).

The CMAES algorithm starts with the parameter initialization. Population mutation is controlled with the algorithm parameters. These are the mean value (m^g), step size (σ^g), and covariance matrix (C_g) as shown in Eq. (2). g is the population number.

$$x_k{}^{g+1} = m^g + \vartheta^g N(0, C^g),\ \ k = 1, \dots, \lambda \qquad (2)$$

CMAES has a wide range of applications in engineering. Some of these are: Kaveh *et al.* (2012) used the covariance matrix adaptation evolution strategy algorithm for the size optimisation problem of steel truss structures subjected to ground motions and a generalised regression neural network for the fitness approximation. This method is an effective approach for the optimal design of large-scale truss structures subjected to time domain loading. Liang *et al.* (2019) proposed a new variant of covariance matrix adaptive evolution strategy (CMA-ES) for single objective numerical optimisation problems in continuous domain. The proposed variant method AEALSCE is competitive with other algorithms in terms of accuracy and convergence.

2.5.1.2 Genetic Algorithm (GA)

Genetic Algorithm (GA) developed by Holland (1975). Genetic algorithms have three operators: selection, crossover and mutation. At each iteration or generation, these operators are used on the population of all possible solutions to improve the fitness function. Each solution is

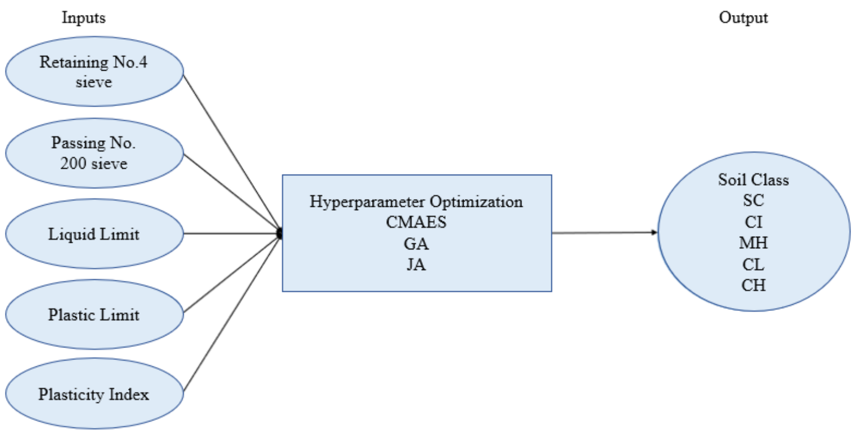


Fig. 6 Foundational approach for hyperparameter optimization process

defined by a sequence, and these sequences are often the original chromosomes, hence the name genetic algorithm. The population is randomly generated at the beginning and continues until the termination criteria are met (Neeraja 2017). The algorithm includes five biological processes called mating, reproduction, cloning, crossover and mutation for optimization applications (Bekdaş *et al.* 2021)

In the optimization process with GA, each candidate solution corresponds to a chromosome, and each variable corresponds to different genes on the chromosomes. Therefore, a natural genetic process operates on a set of chromosomes in order to converge to the best result for any problem. Although it is the last stage of optimization, the mutation is needed, since individuals may have similar characteristics and crossover may be insufficient to ensure individual diversity. Eq. (3) is applied for the mutation process [74]. In Eq. (3), mr, $X_{q,new}$ $X_{q,min}$, and $X_{q,max}$ are is mutation rate, new, lower, and upper limit values of qth parameter, respectively. In Eq. (4) q is a randomly selected gene and nc represents the number of chromosomes.

$$X_{q,new} = mr > rand(), X_{q,min} + rand()(X_{q,max} - X_{q,min}) \quad (3)$$

$$q = ceil(rand()) \times nc \quad (4)$$

2.5.1.3 Jaya Algorithm (JA)

The Jaya Algorithm developed by Rao (2016) works based on the approach of always reaching the optimum solution. Therefore, the algorithm aims to reach the best solution and move away from the worst solution at the same time. The algorithm is named after the Sanskrit word Jaya, which means victory.

Jaya algorithm works on the basis of fewer functions than other methods. Therefore, fewer operations are required to reach the optimum result. In addition, the algorithm implementation is simple and does not contain special parameters, which makes Jaya Algorithm relatively superior to other methods (Rao 2016). Jaya's algorithm is expressed by Eq. (5).

$$X_{i,new} = X_{i,j} + rand()\left(X_{i,g_{best}} - |X_{i,j}|\right) - rand()\left(X_{i,g_{worst}} - |X_{i,j}|\right) \quad (5)$$

In Eq. (5), $X_{i,new}$ is the new value to be determined for variable i. $X_{i,j}$ is the value of variable i in the initial matrix of the jth candidate solution. $X_{i,gbest}$ is the value of variable i of the best solution in terms of the objective function. $X_{i,gworst}$ is the value of variable i in the vector containing the worst solution in terms of the objective function.

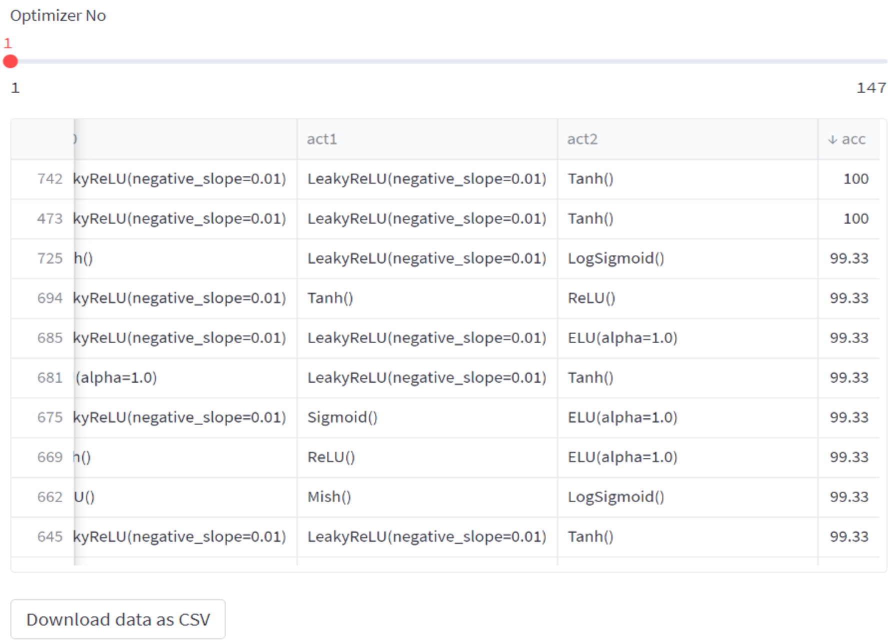
2.5.2 HyperNetExplorer

HyperNetExplorer is a tool that uses various optimization algorithms from the MealPy (Van Thieu and Mirjalili 2023) package for hyperparameter optimization of ANN. The tool was developed using Python and Pytorch as ANN framework and Streamlit as Graphical User Interface (GUI). The tool was first introduced by Aydın *et al.* (2023) and thus the hyperparameters that are optimized with their lower and upper values can be found in Aydın *et al.* (2023).

In this study, MealPy's CMAES, GA and Original JAYA algorithms were used for the optimization of ANN hyperparameters. Once the dataset is loaded the HyperNetExplorer starts to generate different ANN architectures (i.e., with different numbers of layers, with different numbers of neurons in each layer and different activation functions with each layer.). For each generated ANN, its accuracy is determined with k-fold CV(k=10) and recorded and persisted, along with the actual ANN itself. The accuracy rate of each generated ANN is then fed into the optimization algorithm, such as GA, where the optimization algorithm evaluates this result and generates a new ANN architecture. This generation-evaluation loop continues until a pre-determined iteration or time. In our tests, we have generated 1000+ ANN architectures with HyperNetExplorer.

3. Results

In this study, three different optimizers (CMAES, GA and JA) for soil classification based on MealPy package were utilized within the HyperNetExplorer. The aim was to explore the best ANN architecture among the generated ANNs.The schema defining the foundational approach implemented in the study is depicted in Fig. 6.


Optimizer No

1

1 147

	)	act1	act2	↓ acc
742	kyReLU(negative_slope=0.01)	LeakyReLU(negative_slope=0.01)	Tanh()	100
473	kyReLU(negative_slope=0.01)	LeakyReLU(negative_slope=0.01)	Tanh()	100
725	h()	LeakyReLU(negative_slope=0.01)	LogSigmoid()	99.33
694	kyReLU(negative_slope=0.01)	Tanh()	ReLU()	99.33
685	kyReLU(negative_slope=0.01)	LeakyReLU(negative_slope=0.01)	ELU(alpha=1.0)	99.33
681	(alpha=1.0)	LeakyReLU(negative_slope=0.01)	Tanh()	99.33
675	kyReLU(negative_slope=0.01)	Sigmoid()	ELU(alpha=1.0)	99.33
669	h()	ReLU()	ELU(alpha=1.0)	99.33
662	U()	Mish()	LogSigmoid()	99.33
645	kyReLU(negative_slope=0.01)	LeakyReLU(negative_slope=0.01)	Tanh()	99.33

Download data as CSV

Fig. 7 10 Best Performing ANNs discovered by CMAES optimizer

Table 6 Neural network structure and parameters of the soil dataset (CMAES)

Parameter	Value
Total number ANNs Generated	1050
Iteration No. of Best ANN	742
Best ANN	LeakyRelu-Tanh
Hidden layers	3
Neurons per layer	256-512-256

Fig. 7 is a screenshot illustrating the web-based user interface of HyperNetExplorer showing the accuracy rates (acc) of the 10 best-performing ANNs obtained using the Covariance Matrix Adaptation Evolution Strategy (CMAES) in descending order. For the soil dataset, the best mean accuracy obtained as a result of 10-fold CV of all discovered ANNs (ANN architectures) generated with CMAES optimizer was 100%. The accuracy range of the 10 best-performing ANN architectures was 99.33%-100.

Table 6 provides i.) training details and ii.) the details of the ANN architecture achieving 100% mean accuracy in 10-fold cross-validation, discovered through the hyperparameter optimization with CMAES algorithm.

Fig. 8 shows the web-based user interface of HyperNetExplorer showing all metadata regarding the 10 best-performing networks obtained using the Genetic Algortihm (GA) in descending order. For the soil dataset, best mean accuracy of 10-fold CV achieved by ANNs (ANN architectures) generated with GA was 100%. The accuracy range of the best performing 10 ANN architectures was between 99.67%-100.

Table 7 provides i.) training details and ii.) the details of the ANN architecture achieving 100% mean accuracy in 10-fold cross-validation, discovered through the hyperparameter optimization with GA.

Table 7 Neural network structure and parameters of the soil dataset (GA)

Parameter	Value
Total number ANNs Generated	1050
Iteration No. of Best ANN	375
Best ANN	LeakyRelu-Tanh
Hidden layers	3
Neurons per layer	512-512-512

Table 8 Neural network structure and parameters of the soil dataset (GA)

Parameter	Value
Total number ANNs Generated	1050
Iteration No. of Best ANN	1009
Best ANN	LeakyRelu-Mish
Hidden layers	3
Neurons per layer	512-512-512

Fig. 9 illustrates a similar screenshot taken from the web-based UI of the HyperNetExplorer showing the metadata of the 10 best performing ANNs obtained this time using the Jaya Algorithm (JA). For the soil dataset, best mean accuracy of 10-fold CV achieved by ANNs (ANN architectures) generated with JA was 100%. The accuracy range of the best-performing ANN architectures was between 99.67%-100 for the JA.

Table 8 provides i.) training details and ii.) the details of the ANN architecture achieving 100% mean accuracy in 10-fold cross-validation, discovered through the hyperparameter optimization with JA.

Table 9 provides a summary of the best accuracy rates achieved by the discovered ANN architectures for the soil dataset using all three CMAES, GA and JA optimizers. In

Optimizer No

1

1 147

	nl	hl0	hl1	hl2	act0	act1	act2	↓ acc
375	2	512	512	512	LeakyReLU(nega	LeakyReLU(neg	Tanh()	100
285	2	512	512	512	LeakyReLU(nega	ReLU()	ReLU()	100
1,048	2	512	512	512	LeakyReLU(nega	ELU(alpha=1.0)	Tanh()	99.67
1,047	2	512	512	512	LeakyReLU(nega	LeakyReLU(neg	LogSigmoid()	99.67
1,037	2	256	512	512	LeakyReLU(nega	LeakyReLU(neg	LogSigmoid()	99.67
1,028	2	512	512	512	LeakyReLU(nega	LeakyReLU(neg	LeakyReLU(nega	99.67
1,000	2	256	512	512	LeakyReLU(nega	LeakyReLU(neg	LogSigmoid()	99.67
946	2	512	512	512	LeakyReLU(nega	LeakyReLU(neg	LogSigmoid()	99.67
849	2	256	512	512	LeakyReLU(nega	Mish()	Tanh()	99.67
814	2	512	512	512	LeakyReLU(nega	LeakyReLU(neg	Tanh()	99.67

Download data as CSV

Fig. 8 10 Best Performing ANNs discovered by GA optimizer

Optimizer No

1

1 147

	nl	hl0	hl1	hl2	act0	act1	act2	↓ acc
1,009	2	512	512	512	LeakyReLU(negative_slope=0.01)	LeakyReLU(nega	Mish()	100
991	2	512	512	512	LeakyReLU(negative_slope=0.01)	LeakyReLU(nega	Mish()	100
632	2	512	512	512	LeakyReLU(negative_slope=0.01)	LeakyReLU(nega	Mish()	100
261	2	256	512	512	ReLU()	LeakyReLU(nega	LeakyReLU(neg	100
987	2	512	512	512	LeakyReLU(negative_slope=0.01)	LeakyReLU(nega	Mish()	99.67
909	2	512	512	512	LeakyReLU(negative_slope=0.01)	LeakyReLU(nega	LogSigmoid()	99.67
903	2	512	512	512	LeakyReLU(negative_slope=0.01)	LeakyReLU(nega	Mish()	99.67
883	2	512	512	512	LeakyReLU(negative_slope=0.01)	LeakyReLU(nega	ELU(alpha=1.0)	99.67
813	2	512	512	512	LeakyReLU(negative_slope=0.01)	LeakyReLU(nega	Tanh()	99.67
799	2	512	512	512	LeakyReLU(negative_slope=0.01)	LeakyReLU(nega	LogSigmoid()	99.67

Download data as CSV

Fig. 9 10 Best Performing ANNs discovered by JA optimizer

addition, the table also provides the best accuracy values observed in our earlier research conducted based on Conventional Machine Learning Algorithms using the same dataset in Aydın *et al.* (2023).

When Table 9 is examined, it is seen that all ANN architectures discovered were able to achieve the same perfect accuracy(acc=100%) for the soil classification dataset. When the number of iterations in which the optimizers find the best architectures, is compared, GA appeared as the quickest algorithm (in terms of iteration steps) to find the best ANN architecture. The discovered ANNs provided much better classification accuracies when compared with our previous publication (Aydın *et al.* 2023) where we tested several ML algorithms for the same problem. A radar chart of the prediction accuracies and speed of convergence is given in Fig. 10.

In our previous research (Aydin *et al.* 2023), the highest accuracy achieved with decision tree (CART) classifier was

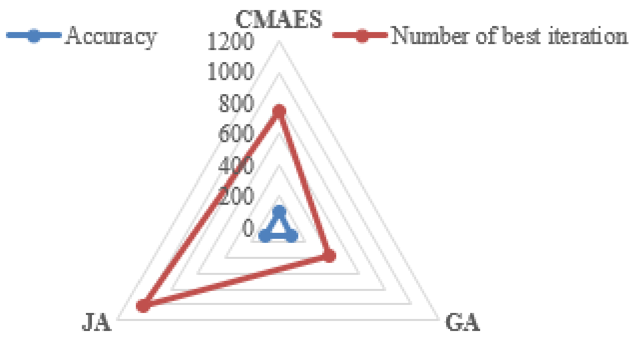


Fig. 10 Optimizer performances

Table 9 Best accuracy achieved with each optimizer vs. conventional machine learning algorithms

ANN Hyperparameter Optimization		
Algorithm-Optimizer	Best Accuracy	No.of.Iterations to discover
ANN-CMAES	100	742
ANN-GA	100	375
ANN-JA	100	1009
Conventional ML Algorithms (Aydın *et al.* 2023)	Best Accuracy	
DecisionTreeClassifier	90.66	
KNeighborsClassifier	79.33	
XGBClassifier	90.33	

90.66%. In this study, HyperNetExplorer was able to achieve 100% accuracy for the same dataset. As the aim of HyperNetExplorer was to discover the best/most accurate ANN architecture (model) among many, for a given dataset, thus the models identified/discovered with HyperNetExplorer achieved better performance than the classical (hyperparamter-optimized) ML models, such as Decision Trees or SVMs .

4. Discussion

This research aimed to understand the role of hyperparameter optimization in enhancing the accuracy of ANNs that are being applied to the task of soil type classification, with a specific focus on discovering the applicability of bio-inspired algorithms for this purpose. The results of the experiments conducted demonstrated that hyperparameter-optimized ANNs greatly contribute to the prediction ability of ANNs for the soil type classification problem. The discovered ANNs were able to classify soil types with perfect 100% accuracy. This finding indicates that hyperparameter optimization can lead to the discovery very efficient ANN architectures. It is also observed that bio-inspired algorithms can be successfully utilized for discovering hyperparameter-optimized shallow ANN architectures which can achieve very high prediction accuracies for the soil-type classification problem. In addition, as the ANN architectures were shallow and the number of total parameters to be optimized was very few (when compared with vision or language models), this type of hyperparameter optimization can performed without the need for expensive hardware resources such as high-end GPUs. Despite the fact that the optimization tool makes use of PyTorch and CUDA framework, a low-end Intel Xeon E-2224 CPU was sufficient to complete the search for the best ANN architecture in less than 1 hour. This finding points out that the methodology and the tool were energy efficient as well. This optimized model offers convenience to engineers by automating time-consuming and costly soil classification.

5. Conclusions

Soil classification is an important and time-consuming and costly process. Machine learning can be very helpful to facilitate the soil classification process. This study focuses on the utilization of a new tool HyperNetExplorer developed for hyperparameter optimization, to discover and identify ANN architectures that provide maximum accuracy in classifying different soil types. The results obtained have proved that classification accuracy in this problem can be greatly enhanced through hyperparameter optimization of ANNs. For instance, our recent study (Aydın *et al.* 2023) which we employed and tested different machine learning methods, has resulted in 90% as the best accuracy achieved by utilizing DecisionTree classifier. In fact, in this study, we were able to achieve a perfect 100% accuracy for the exact same problem (and dataset) as a result of the hyperparameter-optimized ANN architectures discovered. The results have also proven that along with optimization techniques such as covariance matrix adaptation evolution strategy, genetic algorithm and Jaya algorithm can also be successfully employed to support hyperparameter optimization to discover the best ANN architectures.

References

Adeli, H. and Yeh, C. (1989), "Perceptron learning in engineering design", *Comput. - Aided Civil Infrastruct. Eng*., **4**(4), 247-256. https://doi.org/10.1111/j.1467-8667.1989.tb00026.x.

Ahmadian, S., Jalali, S.M.J., Raziani, S. and Chalechale, A. (2022), "An efficient cardiovascular disease detection model based on multilayer perceptron and moth-flame optimization", *Exp. Syst*., **39**(4), e12914. https://doi.org/10.1111/exsy.12914.

Alshboul, O., Shehadeh, A., Mamlook, R.E.A., Almasabha, G., Almuflih, A.S. and Alghamdi, S.Y. (2022), "Prediction liquidated damages via ensemble machine learning model: Towards sustainable highway construction projects", *Sustainability*, **14**(15), 9303. https://doi.org/10.3390/su14159303.

Alzabeebee, S., Zuhaira, A.A. and Al-Hamd, R.K.S. (2022), "Development of an optimized model to compute the undrained shaft friction adhesion factor of bored piles", *Geomech. Eng*., **28**(4), 397-404. https://doi.org/10.12989/gae.2022.28.4.397.

Amarasiri, S. and Muhunthan, B. (2022), "Evaluating cracking deterioration of preventive maintenance–treated pavements using machine learning", *J. Transport. Eng. Part B: Pavements*, **148**(2), 04022014. https://doi.org/10.1061/JPEODX.0000354.

Amâncio, L.B., Dantas Neto, S.A. and Cunha, R.P.D. (2022), "Estimative of shaft and tip bearing capacities of single piles using multilayer perceptrons", *Soils Rocks*, 45. https://doi.org/10.28927/SR.2022.077821.

Anirudh, N., Padala, S.S. and Reddy, H.E. (2022), "Development of ANN-based risk prediction model in construction projects", In Recent Advances in Sustainable Environment: Select *Proceedings of the RAiSE 2022*, Singapore: Nature Singapore. https://doi.org/10.1007/978-981-19-5077-3_9.

Arı, A. and Berberler, M.E. (2017), "Interface design for solving prediction and classification problems with artificial neural networks", *Acta Infologica*, **1**(2), 55-73.

ASTM D2487 (1975), Standard Test Method for Classification of Soils for Engineering Purposes. 1969: R1975. ASTM: West Con-shohocken, PA, USA.

Atlan, F., Hançer, E. and Pençe, İ. (2020), "Evaluation of hyper parameter optimization effect on nuclei segmentation with U-Net", *Eur. J. Sci. Tech*., 60-69.

Aydın, Y., Işıkdağ, Ü., Bekdaş, G., Nigdeli, S.M. and Geem, Z.W. (2023), "Use of machine learning techniques in soil classification", *Sustainability*, **15**(3), 2374. https://doi.org/10.3390/su15032374.

Aydın, Y., Bekdaş, G., Nigdeli, S.M., Işıkdağ, Ü. and Geem, Z.W. (2023), "Comparison of multilayer perceptron and other methods for prediction of sustainable optimum design of reinforced concrete columns", (Eds., Bekdaş, G. and Nigdeli, S.M.), Hybrid Metaheuristics in Structural Engineering. Studies in Systems, Decision and Control, **480**, Springer, Cham. https://doi.org/10.1007/978-3-031-34728-3_12.

Aydın, Y., Cakiroglu, C., Bekdaş, G., Işıkdağ, Ü., Kim, S., Hong, J. and Geem, Z.W. (2024), "Neural network predictive models for Alkali-activated concrete carbon emission using metaheuristic optimization algorithms", *Sustainability*, **16**, 142. https://doi.org/10.3390/su16010142.

Baghbani, A., Choudhury, T., Costa, S. and Reiner, J. (2022), Application of artificial intelligence in geotechnical engineering: "A state-of-the-art review", *Earth-Sci. Reviews*, **228**, 103991. https://doi.org/10.1016/j.earscirev.2022.103991.

Bai, X.D., Cheng, W.C., Ong, D.E. and Li, G. (2021), "Evaluation of geological conditions and clogging of tunneling using machine learning", *Geomech. Eng*., **25**(1), 59-73. https://doi.org/10.12989/gae.2021.25.1.059.

Bekdaş, G., Nigdeli, S.M., Yücel, M. and Kayabekir, A.E. (2021), Artificial intelligence optimization algorithms and engineering applications, Seçkin Publishing: Ankara, Turkey.

Bekdaş, G., Yücel, M. and Nigdeli, S.M. (2021), "Estimation of optimum design of structural systems via machine learning", *Front. Struct. Civil Eng*., 1-12. https://doi.org/10.1007/s11709-021-0774-0.

Birikundavyi, S., Labib, R., Trung, H.T. and Rousselle, J. (2002), "Performance of neural networks in daily streamflow forecasting", *J. Hydrol. Eng*., **7**(5), 392-398. https://doi.org/10.1061/(ASCE)1084-0699(2002)7:5(392).

Chen, G., Yin, J. and Yang, S. (2023), "Ship autonomous berthing simulation based on covariance matrix adaptation evolution strategy", *J. Mar. Sci. Eng*., **11**(7), 1400. https://doi.org/10.3390/jmse11071400.

Chen, S.M., Wang, Y.M. and Tsou, I. (2013), "Using artificial neural network approach for modelling rainfall–runoff due to typhoon", *J. Earth Syst. Sci*., **122**, 399-405. https://doi.org/10.1007/s12040-013-0289-8.

Cheng, M.Y. and Darsa, M.H. (2021), "Construction schedule risk assessment and management strategy for foreign general contractors working in the Ethiopian construction industry", *Sustainability*, **13**(14), 7830. https://doi.org/10.3390/su13147830.

Chon, S.H. (2019), Hyperparameter Optimization of a Convolutional Neural Network. Air Force Institute of Technology Wright-Patterson AFB Oh Wright-Patterson United States.

Cinaroglu, S. andBaser, O. (2020), "Comparative regression performances of machine learning methods optimising hyperparameters: application to health expenditures", *Int. J. Bioinform. Res. Appl*., **16**(4), 387. https://doi.org/10.1504/ijbra.2020.113022.

Das, S.K., Biswal, R.K., Sivakugan, N. and Das, B. (2011), "Classification of slopes and prediction of factor of safety using differential evolution neural networks", *Environ. Earth Sci*., **64**(1), 201-210. https://doi.org/10.1007/s12665-010-0839-1.

Dehghanbanadaki, A., Rashid, A.S.A., Ahmad, K., Yunus, N.Z.M. and Said, K.N.M. (2022), "A computational estimation model for the subgrade reaction modulus of soil improved with DCM columns", *Geomech. Eng*., **28**(4), 385-396. https://doi.org/10.12989/gae.2022.28.4.385.

Dinarvand, R. and Ardakani, A. (2022), "Shear behavior of geotextile-encased gravel columns in silty sand-Experimental and SVM modeling", *Geomech. Eng*., **28**(5), 505-520. https://doi.org/10.12989/gae.2022.28.5.505.

Elshorbagy, A., Simonovic, S.P. and Panu, U.S. (2000), "Performance evaluation of artificial neural networks for runoff prediction", *J. Hydrol. Eng*., **5**(4), 424-427. https://doi.org/10.1061/(ASCE)1084-0699(2000)5:4(424).

Esmaeili-Falak, M. and Benemaran, R.S. (2023), "Ensemble deep learning-based models to predict the resilient modulus of modified base materials subjected to wet-dry cycles", *Geomech. Eng.*, **32**(6), 583-600. https://doi.org/10.12989/gae.2023.32.6.583.

Fereidooni, D. and Karimi, Z. (2023), "Predicting rock brittleness indices from simple laboratory test results using some machine learning methods", *Geomech. Eng*., **34**(6), 697-726. https://doi.org/10.12989/gae.2023.34.6.697.

Ghunimat, D., Alzoubi, A.E., Alzboon, A. and Hanandeh, S. (2023), "Prediction of concrete compressive strength with GGBFS and fly ash using multilayer perceptron algorithm, random forest regression and k-nearest neighbor regression", *Asian J. Civil Eng*., **24**(1), 169-177. https://doi.org/10.1007/s42107-022-00495-z.

GeoGebra Graphing Calculator. Available online: httpss://www.geogebra.org/graphing?lang=en (accessed on 24 September 2023).

Gülcü, A. and Kuş, Z. (2019), "A Survey of hyper-parameter optimization methods in convolutional nural networks", *Gazi Univ. J. Sci. Part C: Des. Tech*., **7**(2), 503-522.

Hansen, N. and Ostermeier, A. (1996), "Adapting arbitrary normal mutation distributions in evolution strategies: the covariance matrix adaptation", *Proceedings of IEEE International Conference on Evolutionary Computation*, https://doi.org/10.1109/icec.1996.542381

Hasanzadehshooiili, H., Mahinroosta, R., Lakirouhani, A. and Oshtaghi, V. (2013), "Using artificial neural network (ANN) in prediction of collapse settlements of sandy gravels", *Arabian J. Geosci*., **7**(6), 2303-2314. https://doi.org/10.1007/s12517-013-0858-9.

Hema, H., Chakravarthy, H.G.N. amd Naganna, S.R. (2022), "Prediction of ultimate load carrying capacity of short cold-formed steel built-up lipped channel columns using machine learning approach", *Sādhanā*, **47**(4). https://doi.org/10.1007/s12046-022-01979-z.

Holland, J.H. (1975), Adaptation in Natural and Artificial Systems; University of Michigan Press: Ann Arbor, MI, USA.

Isleem, H.F., Tayeh, B.A., Alaloul, W.S., Musarat, M.A. and Raza, A. (2021), "Artificial Neural Network (ANN) and Finite Element (FEM) models for GFRP-reinforced concrete Columns under axial compression", *Materials*, **14**(23), 7172. https://doi.org/10.3390/ma14237172 .

İnik, Ö. and Ü lker, E. (2017), "Deep learning and deep learning models used in image analysis", *J. Gaziosmanpasa Scientific Res.*, **6**(3), 85-104.

Jabamony, J. and Shanmugavel, G. (2020), "IoT based bus arrival time prediction using Artificial Neural Network (ANN) for Smart Public Transport System (SPTS)", *Int. J. Intell. Eng. Syst*., **13**(1), 312-323. https://doi.org/10.22266/ijies2020.0229.29.

Jia, D.W. and Wu, Z.Y. (2022), "Structural probabilistic seismic risk analysis and damage prediction based on artificial neural network", *Structures*, **41**, 982-996. https://doi.org/10.1016/j.istruc.2022.05.056

Karabaş, A. (2019), Irony detection in Turkish microblogs. Master's Thesis, Yildiz Technical University, Istanbul, Turkey.

Karunanithi, N., Grenney, W.J., Whitley, D. and Bovee, K. (1994), "Neural networks for river flow prediction", *J. Comput. Civil Eng*., **8**(2), 201-220. https://doi.org/10.1061/(asce)0887-3801(1994)8:2(201).

Karlik, B. and Olgac, A.V. (2011), "Performance analysis of various activation functions in generalized MLP architectures of neural networks", *Int. J. Artif. Intell. Exp. Syst*., **1**(4), 111-122.

Kasasbeh, B., Aldabaybah, B. and Ahmad, H. (2022), "Multilayer perceptron artificial neural networks-based model for credit card fraud detection", *Indonesian J. Elec. Eng. Comput. Sci*., **26**(1), 362-373. https://doi.org/10.11591/ijeecs.v26.i1.pp362-373.

Kaveh, A., Fahimi-Farzam, M. and Kalateh-Ahani, M. (2012), "Time-history analysis based optimal design of space trusses: The CMA evolution strategy approach using GRNN and WA", *Struct. Eng. Mech*., **44**(3), 379-403. http://doi.org/10.12989/sem.2012.44.3.379.

Kukreja, H., Bharath, N., Siddesh, C.S. and Kuldeep, S. (2016), "An introduction to artificial neural network", *Int. J. Adv. Res. Innov. Ideas Educ*., **1**, 27-30.

Kwak, N.S. and Ko, T.Y. (2022), "Machine learning-based regression analysis for estimating Cerchar abrasivity index", *Geomech. Eng*., **29**(3), 219-228. https://doi.org/10.12989/gae.2022.29.3.219.

Lawal, A.I., Kwon, S., Aladejare, A.E. and Oniyide, G.O. (2022), "Prediction of the static and dynamic mechanical properties of sedimentary rock using soft computing methods", *Geomech. Eng*., **28**(3), 313-324. https://doi.org/10.12989/gae.2022.28.3.313.

Lee, J.S., Park, J., Kim, J. and Yoon, H.K. (2022), "Study of oversampling algorithms for soil classifications by field velocity resistivity probe", *Geomech. Eng*., **30**(3), 247-258. https://doi.org/10.12989/gae.2022.30.3.247.

Lee, K.S., Ahn, J.H., Park, H.Y., Seo, Y.D. and Kim, S.C. (2022), "Seismic acceleration estimation method at arbitrary position using observations and machine learning", *KSCE J. Civil Eng*., **27**(2), 712-726. https://doi.org/10.1007/s12205-022-1235-6.

Li, J., Gao, F., Lin, S., Guo, M., Li, Y., Liu, H. and Wen, Q. (2023), "Quantum k-fold cross-validation for nearest neighbor classification algorithm", *Physica A: Statist. Mech. Appl*., **611**, 128435. https://doi.org/10.1016/j.physa.2022.128435.

Liang, Y., Wang, X., Zhao, H., Han, T., Wei, Z. and Li, Y. (2019), "A covariance matrix adaptation evolution strategy variant and its engineering application", *Appl. Soft Comput*., **83**, 105680. https://doi.org/10.1016/j.asoc.2019.105680.

Mahmoodzadeh, A., Mohammadi, M., Abdulhamid, S.N., Ibrahim, H.H., Ali, H.F.H., Nejati, H.R. and Rashidi, S. (2022), "Prediction of duration and construction cost of road tunnels using Gaussian process regression", *Geomech. Eng*., **28**(1), 65-75. https://doi.org/10.12989/gae.2021.28.1.065.

Menon, U.A. and Nair, D.S. (2022), "Seismic response prediction of RC buildings using artificial Neural Network", *Proceedings of the SECON'22*, 403-413. https://doi.org/10.1007/978-3-031-12011-4_31.

Neeraja, D., Kamireddy, T., Kumar, P.S. and Reddy, V.S. (2017), "Weight optimization of plane truss using genetic algorithm", *Proceedings of the IOP Conference Series: Materials Science and Engineering*, 263, 032015. https://doi.org/10.1088/1757-899x/263/3/032015.

Nigdeli, S.M., Yücel, M. and Bekdaş, G. (2022), "A hybrid artificial intelligence model for design of reinforced concrete columns", *Neural Comput. Appl*., **35**(10), 7867-7875. https://doi.org/10.1007/s00521-022-08164-7.

Owusu-Ababio, S. (1998), "Effect of neural network topology on flexible pavement cracking prediction", *Comput.-Aided Civil Infrastruct. Eng*., **13**(5), 349-355. https://doi.org/10.1111/0885-9507.00113

Ö ztemel, E. (2006), Artificial Neural Networks. Papatya.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, S., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J. and Chintala, S. (2019), "PyTorch: An imperative style, high-performance deep learning library", *Adv. Neural Inform. Process. Syst.*, **32**, 8024-8035. Curran Associates, Inc. Retrieved from https://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-

Patil-Shinde, V., Lakshete, V.A. and Gadekar-Shinde, S. (2023), "Bio-diesel synthesis from waste sunflower oil: Experimental and multilayer perceptron neural network modeling", *Materials Today: Proceedings*. https://doi.org/10.1016/j.matpr.2023.01.117

Pereira, L., Godinho, L. and Branco, F.G. (2023), "Predicting unconfined compression strength and split tensile strength of soil-cement via artificial neural networks", *Geomech. Eng*., **33**(6), 611-624. http://doi.org/10.12989/gae.2023.33.6.611.

Ramli, A.S., Rashid, M.I.M. and Ahmad, M.A. (2020), "Energy management strategy of hev based on simulated annealing", *Int. J. Integrated Eng*., **12**(2), 30-37.

Venkata Rao, R. (2016), "Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems", *Int. J. Ind. Eng. Comput*., 19-34. https://doi.org/10.5267/j.ijiec.2015.8.004.

Ruuska, S., Hämäläinen, W., Kajava, S., Mughal, M., Matilainen, P. and Mononen, J. (2018), "Evaluation of the confusion matrix method in the validation of an automated system for measuring feeding behaviour of cattle", *Behavioural Processes*, **148**, 56-62. https://doi.org/10.1016/j.beproc.2018.01.004.

Samui, P. and Sitharam, T.G. (2010), "Site characterization model using artificial neural network and Kriging", *Int. J. Geomech*., **10**(5), 171-180. https://doi.org/10.1061/(asce)1532-3641(2010)10:5(171).

Sanni-Anibire, M.O., Zin, R.M. and Olatunji, S.O. (2020), Machine learning model for delay risk assessment in tall building projects", *Int. J. Constr. Management*, **22**(11), 2134-2143. https://doi.org/10.1080/15623599.2020.1768326.

Sharma, V., Pratap Singh Chouhan, A. and Bisen, D. (2022), "Prediction of activation energy of biomass wastes by using multilayer perceptron neural network with Weka", *Materials Today: Proceedings*, **57**, 1944–1949. https://doi.org/10.1016/j.matpr.2022.03.051.

Singh, V., Bano, S., Ahmad, A.K. and Ahmad, S. (2019), "Feasibility of artificial neural network in civil engineering", Int. *J. Trend Scientific Res. Development*, **3**(3), 724-728. https://doi.org/10.31142/ijtsrd22985.

Sivrikaya, O. and Soycan, T.Y. (2010), "Estimation of compaction parameters of fine-grained soils in terms of compaction energy using artificial neural networks", *Int. J. Numer. Anal. Method. Geomech.*, **35**(17), 1830-1841. https://doi.org/10.1002/nag.981.

Soyer, M.A., Kalaycı, C.B. and Karakaş, Ö. (2022), "Low-cycle fatigue parameters and fatigue life estimation of high-strength steels with artificial neural networks", *Fatigue Fract. Eng. M.*, **45**(12), 3764-3785. Portico. https://doi.org/10.1111/ffe.13847.

Sridhar, J., Balaji, S., Jegatheeswaran, D. and Awoyera, P. (2023), "Prediction of the mechanical properties of fibre-reinforced quarry dust concrete using response surface and artificial Neural network techniques", *Adv. Civil Eng*., **2023**, 1-13. https://doi.org/10.1155/2023/8267639.

Su, M., Feng, G., Liu, Z., Li, Y. and Wang, R. (2020), "Tapping on the black box: how is the scoring power of a machine-learning scoring function dependent on the training set?", J. Chem. Inform. Model., **60**(3), 1122-1136. https://doi.org/10.1021/acs.jcim.9b00714.

Tachi, S.E., Ouerdachi, L., Remaoun, M., Derdous, O. and Boutaghane, H. (2016), "Forecasting suspended sediment load using regularized neural network: Case study of the Isser River (Algeria)", *J. Water Land Development*, **29**(1), 75. htts://doi.org/10.1515/jwld-2016-0014.

Tinoco, J., Correia, A.G. and Cortez, P. (2009), "A data mining approach for Jet Grouting Uniaxial Compressive Strength prediction", *Proceedings of the 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*. https://doi.org/10.1109/nabic.2009.5393401.

Topçu, İ.B. and Sarıdemir, M. (2008), "Prediction of rubberized concrete properties using artificial neural network and fuzzy logic", *Constr. Build. Mater*., **22**(4), 532-540. https://doi.org/10.1016/j.conbuildmat.2006.11.007.

McCulloch, W.S. and Pitts, W. (1943), "A logical calculus of the ideas immanent in nervous activity", *Bull. Math. Biophys*., **5**(4), 115-133. https://doi.org/10.1007/bf02478259.

Van Thieu, N. and Mirjalili, S. (2023), "MEALPY: An open-source library for latest meta-heuristic algorithms in Python", *J. Syst. Architect*., **139**, 102871. https://doi.org/10.1016/j.sysarc.2023.102871.

Verma, G. and Kumar, B. (2022), "Multi-layer perceptron (MLP) neural network for predicting the modified compaction parameters of coarse-grained and fine-grained soils", *Innov. Infrastruct. Solution.*, 7(1), 78. https://doi.org/10.1007/s41062-021-00679-7.

Wrzesiński, G. and Markiewicz, A. (2022), "Prediction of permeability coefficient k in sandy soils using ANN", *Sustainability*, **14**(11), 6736. https://doi.org/10.3390/su14116736.

Wu, D., Ma, X. and Olson, D.L. (2022), "Financial distress prediction using integrated Z-score and multilayer perceptron neural networks", *Decision Support Systems*, **159**, 113814. https://doi.org/10.1016/j.dss.2022.113814.

Xie, S., Li, Y., Tan, H., Liu, R. and Zhang, F. (2022), "Multi-scale and multi-layer perceptron hybrid method for bearings fault diagnosis", *Int. J. Mech. Sci*., **235**, 107708. https://doi.org/10.1016/j.ijmecsci.2022.107708.

Yadav, S. and Shukla, S. (2016), "Analysis of k-fold cross-validation over hold-out validation on colossal datasets for quality classification", *Proceedings of the 2016 IEEE 6th International conference on advanced computing (IACC)*, IEEE. https://doi.org/10.1109/IACC.2016.25.

Yılmaz, I., Yıldırım, M. and Keskin, İ. (2016), Soil mechanics laboratory experiments and solved problems, Seçkin Publishing.

Yucel, M., Bekdas, G., Nigdeli, S.M. and Sevgen, S. (2018), "Artificial neural network model for optimum design of tubular columns", *Int. J. Theor. Appl. Mech.*, 3.

Yucel, M., Nigdeli, S.M. and Bekdaş, G. (2019), "Generation of an artificial neural network model for optimum design of I-beam with minimum vertical deflection", *Proceedings of the 12th HSTAM international congress on mechanics*, Thessaloniki, Greece.

Yucel, M., Kayabekir, A.E., Nigdeli, S.M. and Bekdaş, G. (2022), "Optimum design of Carbon Fiber-Reinforced Polymer (CFRP) beams for shear capacity via machine learning methods", *Res. Anthology on Machine Learning Techniques, Method. Appl*., 308-326. https://doi.org/10.4018/978-1-6684-6291-1.ch018.

Yücel, M., Nigdeli, S.M., Kayabekir, A.E. and Bekdaş, G. (2021), "Optimization and artificial neural network models for reinforced concrete members", Nature-Inspired Metaheuristic Algorithms for Engineering Optimization Applications, 181-199. https://doi.org/10.1007/978-981-33-6773-9_9.

Yücel, M., Bekdaş, G., Nigdeli, S.M. and Kayabekir, A.E. (2021), "An artificial intelligence-based prediction model for optimum design variables of reinforced concrete retaining walls", *Int. J. Geomech*., **21**(12). https://doi.org/10.1061/(asce)gm.1943-5622.0002234.

Zendehboudi, A. and Saidur, R. (2019), "A reliable model to estimate the effective thermal conductivity of nanofluids", *Heat Mass Transfer*., **55**, 397-411.

Zhang, G., Ali, Z.H., Aldlemy, M.S., Mussa, M.H., Salih, S.Q., Hameed, M.M., Al-Khafaji, Z.S. and Yaseen, Z.M. (2020), "Reinforced concrete deep beam shear strength capacity modelling using an integrative bio-inspired algorithm with an artificial intelligence model", *Eng. with Comput*., 1-14.

Zhang, G., Chen, C., Zhang, Y., Zhao, H., Wang, Y. and Wang, X. (2022), "Optimised neural network prediction of interface bond strength for GFRP tendon reinforced cemented soil", *Geomech. Eng*., **28**(6), 599-611. https://doi.org/10.12989/gae.2022.28.6.599.

Zheng, A. (2015), Evaluating Machine Learning Models A Beginner's Guide to Key Concepts and Pitfalls, O'Reilly.

CC