

Live to learn: learning rules-based artificial neural network

Aseel Shakir I. Hilaiwah¹, Hanan Abed Alwally Abed Allah², Basim Akhudir Abbas³, Tole Sutikno⁴

¹Imam A'adhum University College, Department of Fundamentals of Relegions For Girls, Baghdad, Iraq

^{2,3}Mustansiriya University, College of Science, Department of Computer Science, Baghdad, Iraq

⁴Department of Electrical Engineering, Universitas Ahmad Dahlan, Yogyakarta, Indonesia

⁴Embedded System and Power Electronics Research Group (ESPERG), Yogyakarta, Indonesia

Article Info

Article history:

Received May 5, 2020

Revised Aug 9, 2020

Accepted Aug 30, 2020

Keywords:

Artificial neural networks

Deep learning

Lifelong learning

Neural network

Training

ABSTRACT

An extensive review of the artificial neural network (ANN) is presented in this paper. Previous studies review the artificial neural network (ANN) based on the approaches (algorithms) used or based on the types of the artificial neural network (ANN). The presented paper reviews the ANN based on the goal of the ANN (methods, and layers), which become the main objective of this paper. As a famous artificial intelligent model, ANN mimics the human nervous system in handling the information transmitted by different nodes (also known as neurons) in this model. These nodes are stacked in layers and work collectively to bring about solution to complex problems. Numerous structures exist for ANN and each of these structures is designed to address a specific task. Basically, the ANN architecture is comprised of 3 different layers wherein the first layer represents the input layer that consist of several input nodes that represent the input parameter for the model. The hidden layer is the second layer and consists of a hidden layer of neurons. The neurons in this layer are directly connected to the neurons in the output layer. The third layer is the output layer which is the models' response layer. The output layer neurons have the activation functions for the calculation of the ANN final output. The connection between the nodes in the ANN model is mediated by synaptic weights. This paper is a comprehensive study of ANN models and their layers.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Aseel Shakir I. Hilaiwah

Department of Fundamentals of Relegions For Girls

Imam A'adhum University College, Baghdad, Iraq

Email: asshakir1983@gmail.com

1. INTRODUCTION

The currently existing ANN models allows for adjustment of the network behavior by altering the weights that connects the neurons to each other; however, the network designer is expected to set up the number of neurons, as well as the structural inter-neuron relationship which once designed, is sustained throughout the lifetime of the network [1, 2]. This is one of the constraints on the applicability of ANNs. "The ability to interpret and manipulate internal workings of neural network is a major breakthrough in the ANN theoretical research [3-6]. The author proves that most of the feed forward neural networks are functions; thus, many properties of functions can be applied to analyze neural network behaviors. Based on this proof, a graphical mapping technique was proposed to interpret the internal activities of ANN models. With this technique, it is possible to study the impact of noisy data on ANN modeling, and several key features of ANN models such as memorization, robustness, and sensitivity from the perspective of artificial neurons and their connection weights [7-10].

The decision-making capability of ANNs is reliant on a set of weights as the weights serve as a store for important network information. Thus, it is important to ensure a careful selection of the network weights because the accuracy of the model decision is dependent on the detection of the optimal weight [11-20]. The aim of the training phase is to determine the optimal weight for the whole network and the training process is reliant on a set of rules. Several algorithms have been built for ANN weights adjustment based on specific measures [21, 22]. During the training phase, the weights are first assigned randomly before the estimation of each neurons output based on specified rules. Then, the weights are adjusted by repetitively matcing with the expected output until the optimal weights are reached. The learning frameworks can either be unsupervised, supervised, or reinforcement algorithm [23-25].

In this study, one of the key components is the sensitivity analysis that will be conducted using the ANN models. The aim of the sensitivity analysis is to examine the consistency of the models' performance with known oil sands extraction behavior; it basically portrays the level of trust in the ANN model. The ANN models have been proven successful in areas where standard statistical linear regression models have failed to identify the existence of any form of relationship between the key input parameters from the extraction database [26-28].

This paper contributes the following:

- a) Review of previous studies on ANNs.
- b) Review of ANNs based on the goal and methods of the ANN methods.
- c) Benchmarking of dimensionality reduction methods for malware classification based on network behaviour.

1) Supervised method

The neural network system in a supervised method requires an external training phase; the network is presented with the expected output for each input and the network weights are adjusted by comparing the actual output with the expected output for each node [29-31]. The output error rate is progressively decreased; this process is repeated until the ANN output is close to the expected output. Backpropagation and Delta rule are the mostly used training instances for supervised learning method [32].

2) Unsupervised method

For this network, the learning phase is not reliant on any external training phase in this method. The ANN model leads the learning process for the whole model based on specific criteria and such criteria is formed within the neural network without any external assistance [32].

3) Reinforcement learning method

Here, the NN can learn its behaviours from the external world. It resembles the supervised method of learning just that the reinforcement learning method is dependent on less pre-information and does rely less on exactness of the output. However, the models' output is considered as true or false [33].

4) Artificial Neural Network Models

The ANN models consist of two models that form the main infrastructure of ANN. Figure 1 shows the ANN models.

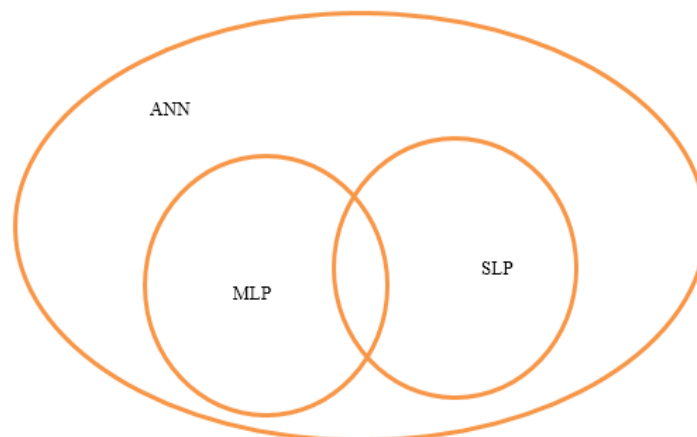


Figure 1. The ANN models

1.1. Single layer perceptron (SLP)

The basic ANN model is represented as a feed forward model. Single layer perceptron models are comprised of only the input & output layers as they lack the feedback process. Additionally, they lack a hidden layer in their structure but relies on supervised learning to determine the network output. They are typically used in solving simple problems that involve simple computations. The SLP model structure is depicted in Figure 2 [34-37].

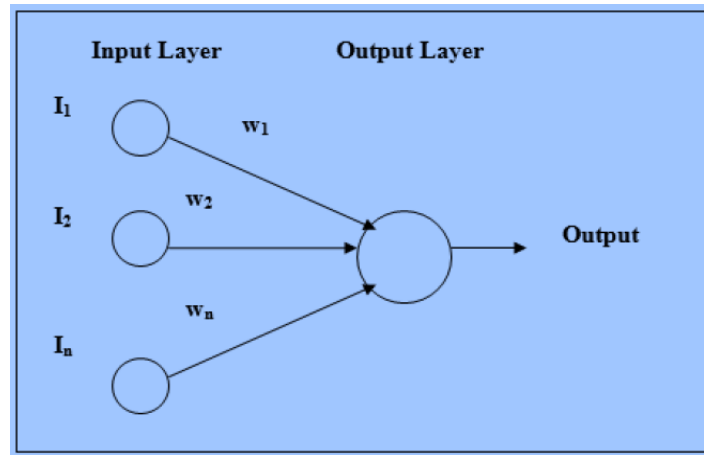


Figure 2. The structure of SLP model

1.2. Multilayer perceptron (MLP)

The difference between MLP and SLP is the presence of a hidden layer in MLP; hence, the MLP has three basic layers which are the input, hidden, and output layers. The number of hidden layers in the MLP varies. Figure 3 depicts the MLP structure [34].

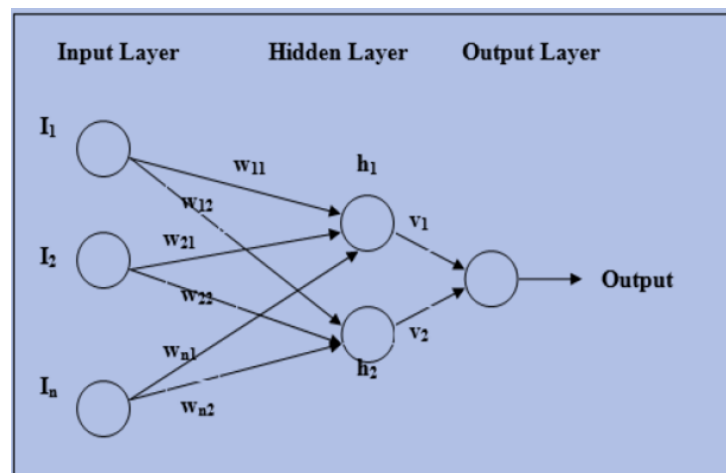


Figure 3. The structure of multilayer perceptron (MLP)

As a feed forward neural network, the MLP is commonly used to solve nonlinear problems. It allows one-way information flow through the network without feedback, meaning that there is no loop or cycle. A typical FFNN is made up of 3 layers and each of these layers contains some nodes based on the considered problem type. The input data is fed into the model via the input layer and forwarded to the hidden layer for onward processing. After processing the data by the hidden layer, the output is forwarded to the output layer to determine the final model result. The MLP model is normally trained using the backpropagation algorithm [38].

2. TRAINING PROCESS

The following processes are involved in the overall training of the ANN model; the input data are weighed and fed into the model via the input layer; the data is forwarded to the hidden layer for the hidden layer neurons to produce the outputs by applying an activation function to the sum of the weighted input values. Then, considering the hidden layer-output layer connections, the resulting outputs are weighed [39]. The output layer generates the expected model results [40]. The expected learning is achieved by continuously adjusting the interconnection weights of the network until the real neuron output matches closely with the target output neuron based on the training data. The variation between the actual and predicted outputs is called the error value. It is a difficult task to determine whether a network has been sufficiently trained in order to terminate the training process; this problem can be addressed a network calibration and this can be done in 2 ways: (i) Checking the number of events that has occurred since the occurrence of the minimum error factor (usually between 20,000 – 40,000 for BPN); (ii) by calculating the calibration test interval (it regulates the convergence rate of the iteration processes). The test set is indirectly involved in the determination of the appropriate time to terminate network training [41, 42].

2.1. Kohonen neural network

This network is a self-organizing map network that can learn without the need for an output data. It exploits the clustering principle to separate data into identical categories. It consists of only an input layer and an output layer [43].

2.2. Probabilistic neural network

This network can train on few data sets; its training phase is so fast that it can be sufficiently trained in just one part of the training set. The PNN also clusters data into specific number of output categories as shown in Figure 4.

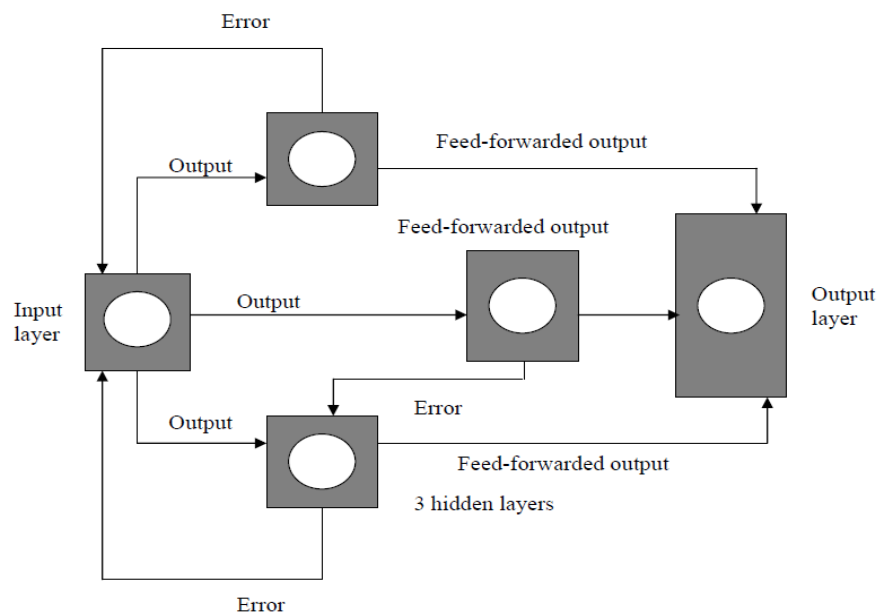


Figure 4. Structure of back propagation network (BPN) with 3 hidden layers

2.3. Learning rules

Various rules are available for learning ANN models; the aim of using these rules is to detect a given set of network weights. Some researchers rely on several learning rules (such as biological learning) while others focus on the evaluation of their perception of training. It is still more tedious learning ANN compared to the facilitation provided by the learning rules. Among the available common learning rules are The Delta Rule, Hebb rule, Hopfield rule, & Kohonen's learning rule [43].

- a) Hebb's Rule: This is a popular learning rule proposed by Hebb in 1949. The Hebb rule provides that when a node is fed by any other node within a network, and all the other nodes are active, these other nodes are considered to have reinforced weights [44].

- b) Hopfield Rule: This rule determines the level of strengths and weaknesses. This rule provides that if the input & output for nodes interlinked to a common weight have a similar sign (active/passive), then, the weight should be increased/decreased by using the learning rate [44].
- c) Delta Law: Also among the commonest rules used in learning processes; it is also referred to as least mean square (LMS) learning rule. The aim of this law is to reduce the rate of error (delta) between the expected and actual outputs for each of the networks' output layer neurons [44, 45].
- d) Gradient Descent Law: This is the commonest rule used for ANN learning; it is similar to the delta law but in this rule, the set of synaptic weights is updated by transferring the error rate through the network. This rule uses proportional constant in consideration of the learning rate that is effective on the set of weights [44]. Sometimes, training performance is improved by using different learning rates for different network layers. The specified learning rate for layers near the output layer is normally smaller in some neural network models compared to the specified learning rate for layer near the input [44].
- e) Kohonen Learning Rule: This rule was adopted by Teuvo Kohonen as a biological inspired approach. It is also called a self-organizing topology; its concept is based on the fact that the processing elements compete to improve their weights, with the best performing element in terms of the best output value being considered the learner that the close elements will learn from. However, this learner strives to keep the competitors from improving their own weights. The number of processing elements within the neighbourhood is varied during the learning phase and is usually kept low [44].

3. NNIDS-NEURAL NETWORK BASED INTRUSION DETECTION SYSTEM

In this work, a system that depends on a novel approach for real-time traffic analysis was developed for the detection and classification of malware. The system was no developed to analyze the traffic itself but to extract the secondary data features that contains much information about the general state of the network [46]. The current system was built on a data set that has 90 dimensions which is more than the number used for DNN classifiers; hence, the accuracy of the analysis is ensured. The use of this large malware dataset to train the DNN makes it possible to detect malware in the dataset and malware that exhibit similar network behavior. This approach used two smaller dimensional DNN systems to detect malware; this reduced the computational power when compared to one larger dimensional DNN. Therefore, a a system for real-time malware monitoring can be developed. The accuracy and performance of four machine learning classifiers were also analyzed and determined; additionally, a heuristic prediction evaluation was performed [47].

3.1. Architecture

Figure 5 shows the overall structure of the proposed system. The data packets are captured by a data packet sniffer and transferred in the protected networks. The obtained data from the sniffer is first preprocessed with the analyzer to extracts the important data features. The DNN classifier is then used to analyze the set of saved features; the chances of the maliciousness of the data stream in the network is considered the output of the system.

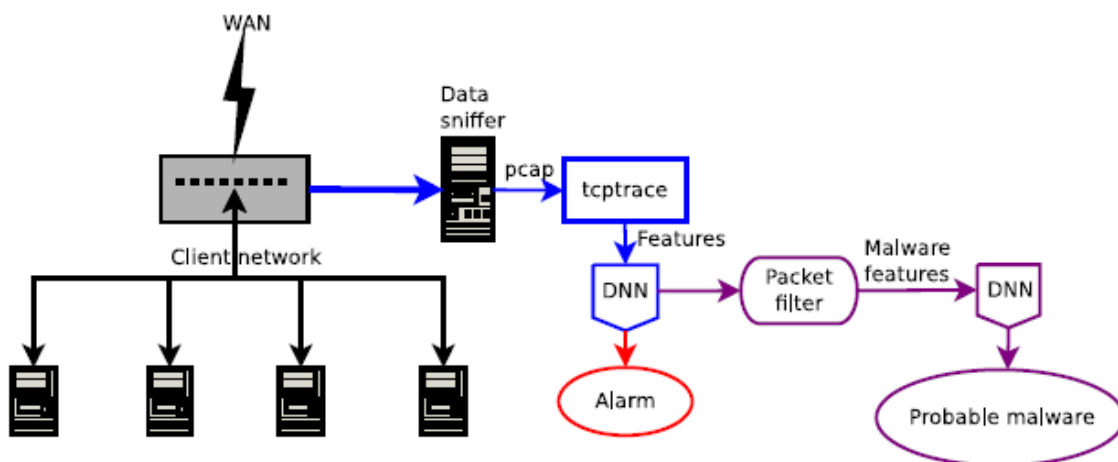


Figure 5. System overview

3.2. Dual dnn malware detection and classification

After extracting the 90 data features from the PCAP file, they were fed into the DNN classifier which was built using the Keras framework [48]. This work proposed the use of two DNN classifiers; the first DNN classifier only determines the status of the traffic (normal or suspicious) and when a suspicious activity is detected, the second DNN classifier will be required to further classify the traffic based on the malware dataset. As a multiclass classifier, the second DNN classifier strives to assign the name of already known malware to the input data based on the dataset it has been trained on. The chances of malware traffic belonging to any of the known malware are generated and they are helpful for heuristic detection [48, 49].

3.3. Benchmarking dimensionality reduction methods for malware classification based on network behavior

Malware classification based on network behavior is an important step in the analysis for malware detection. The efficiency of this step is crucial for better performance of the IDS. The deep neural network (DNN) was the best option in terms of accuracy, but not in terms of performance. We propose two ways of increasing performance by reducing the dimensionality of the datasets and selecting the optimal features [50, 51].

4. ADVANTAGES OF ANN

The robustness of the ANN as a prediction tool is attributed to its following capabilities;

- a) ANN makes a rapid and confident prediction once a new dataset is presented to the constructed model.
- b) As data-driven models, ANNs do not need a pre-knowledge of the data that they will be applied.
- c) ANN learns the data behavior by self-tuning its parameters in a manner that the trained ANN will accurately match the employed data.
- d) ANN can establish the hidden non-linear input-output data relationship and this is suitable for heterogeneous scenarios as commonly experienced in oil & gas reservoirs.
- e) ANN can accurately generalize over a range of input data owing to its self-adaptability.
- f) ANNs can rapidly process data and serve as an easy way of applying an already existing model to a new system.

5. CONCLUSION

ANN has various structures and each of these structures is designed to address a specific task. There are three layers in the basic architecture of ANN (input layer, hidden layer, and output layer) and each layer has a different function in the network system. The final output of the ANN is calculated by the neurons contained in the output layer as they have the activation functions required for this role. The connection between the nodes in an ANN model is mediated by the synaptic weights which are randomly assigned and updated during the learning period using any of the existing training algorithms. This study primarily aims to perform sensitivity analysis using the ANN models for the sake of examining the consistency of the model's performance with known oil sands extraction behavior. It also provides a level of confidence in the performance of the ANN approach. Moreover, ANN models have been proven successful in applications where the standard statistical linear regression methods had not been able to identify the existence of the expected relationship between the major input parameters from the extraction database.

REFERENCES

- [1] D. J. Amit and D. J. Amit, *Modeling brain function: The world of attractor neural networks*, Cambridge university press, 1992.
- [2] H. M. T. Al-Hilfi, M. T. M. Al Hashimi, and I. Marghescu, "WiMAX and WiFi network integration as backhaul on emergency communications," in *2018 14th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET)*, pp. 1063-1068, 2018.
- [3] K. Bayouhdh, "From Machine Learning To Deep Learning," ed: Ebook, 2017.
- [4] R. A. Hasan, M. A. Mohammed, Z. H. Salih, M. A. B. Ameen, N. Țăpuș, and M. N. Mohammed, "HSO: A Hybrid Swarm Optimization Algorithm for Reducing Energy Consumption in the Cloudlets," *Telkomika*, vol. 16, pp. 2144-2154, 2018.
- [5] R. A. Hasan, M. A. Mohammed, N. Țăpuș, and O. A. Hammood, "A comprehensive study: Ant Colony Optimization (ACO) for facility layout problem," in *2017 16th RoEduNet Conference: Networking in Education and Research (RoEduNet)*, 2017, pp. 1-8.
- [6] R. A. Hasan and M. N. Mohammed, "A krill herd behaviour inspired load balancing of tasks in cloud computing," *Studies in Informatics and Control*, vol. 26, pp. 413-424, 2017.

- [7] M. A. Mohammed and R. A. Hasan, "Particle swarm optimization for facility layout problems FLP-A comprehensive study," in *2017 13th IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*, 2017, pp. 93-99.
- [8] M. A. Mohammed, R. A. Hasan, M. A. Ahmed, N. Tapus, M. A. Shanan, M. K. Khaleel, *et al.*, "A Focal load balancer based algorithm for task assignment in cloud environment," in *2018 10th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, 2018, pp. 1-4.
- [9] M. A. Mohammed, A. A. Kamil, R. A. Hasan, and N. Tapus, "An Effective Context Sensitive Offloading System for Mobile Cloud Environments using Support Value-based Classification," *Scalable Computing: Practice and Experience*, vol. 20, pp. 687-698, 2019.
- [10] M. A. Mohammed, I. A. Mohammed, R. A. Hasan, N. Tãpuş, A. H. Ali, and O. A. Hammood, "Green Energy Sources: Issues and Challenges," in *2019 18th RoEduNet Conference: Networking in Education and Research (RoEduNet)*, 2019, pp. 1-8.
- [11] M. A. Ahmed, R. A. Hasan, A. H. Ali, and M. A. Mohammed, "The classification of the modern arabic poetry using machine learning," *Telkomnika*, vol. 17, pp. 2667-2674, 2019.
- [12] A. H. Ali, "A Survey on Vertical and Horizontal Scaling Platforms for Big Data Analytics," *International Journal of Integrated Engineering*, vol. 11, pp. 138-150, 2019.
- [13] O. A. Hammood, M. N. M. Kahar, W. A. Hammood, R. A. Hasan, M. A. Mohammed, A. A. Yoob, *et al.*, "An effective transmit packet coding with trust-based relay nodes in VANETs," *Bulletin of Electrical Engineering and Informatics*, vol. 9, pp. 685-697, 2020.
- [14] O. A. Hammood, N. Nizam, M. Nafaa, and W. A. Hammood, "RESP: Relay Suitability-based Routing Protocol for Video Streaming in Vehicular Ad Hoc Networks," *International Journal of Computers, Communications & Control*, vol. 14, 2019.
- [15] M. A. Mohammed, Z. H. Salih, N. Tãpuş, and R. A. K. Hasan, "Security and accountability for sharing the data stored in the cloud," in *2016 15th RoEduNet Conference: Networking in Education and Research*, 2016, pp. 1-5.
- [16] M. A. Mohammed and N. TãPUŞ, "A Novel Approach of Reducing Energy Consumption by Utilizing Enthalpy in Mobile Cloud Computing," *Studies in Informatics and Control*, vol. 26, pp. 425-434, 2017.
- [17] N. Q. Mohammed, M. S. Ahmed, M. A. Mohammed, O. A. Hammood, H. A. N. Alshara, and A. A. Kamil, "Comparative Analysis between Solar and Wind Turbine Energy Sources in IoT Based on Economical and Efficiency Considerations," in *2019 22nd International Conference on Control Systems and Computer Science (CSCS)*, 2019, pp. 448-452.
- [18] M. A. A. Royida A. Ibrahim Alhayali, Yasmin Makki Mohialden, Ahmed H. Ali, "Efficient method for breast cancer classification based on ensemble hoeffding tree and naïve Bayes," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 18, pp. 1074-1080, 2020.
- [19] Z. H. Salih, G. T. Hasan, and M. A. Mohammed, "Investigate and analyze the levels of electromagnetic radiations emitted from underground power cables extended in modern cities," in *2017 9th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, 2017.
- [20] Z. H. Salih, G. T. Hasan, M. A. Mohammed, M. A. S. Klib, A. H. Ali, and R. A. Ibrahim, "Study the Effect of Integrating the Solar Energy Source on Stability of Electrical Distribution System," in *2019 22nd International Conference on Control Systems and Computer Science (CSCS)*, 2019, pp. 443-447.
- [21] Y. Bodyanskiy, A. Dolotov, I. Pliss, and M. Malyar, "A fast learning algorithm of self-learning spiking neural network," in *2016 IEEE First International Conference on Data Stream Mining & Processing (DSMP)*, 2016, pp. 104-107.
- [22] D. T. Bui, P. Tsangaratos, V.-T. Nguyen, N. V. Liem, and P. T. Trinh, "Comparing the prediction performance of a Deep Learning Neural Network model with conventional machine learning models in landslide susceptibility assessment," *CATENA*, vol. 188, pp. 104426, 2020.
- [23] C. Bishop, "Neural networks for pattern recognition Oxford University Press Oxford Google Scholar," 1995.
- [24] S. van Cranenburgh and A. Alwosheel, "An artificial neural network based approach to investigate travellers' decision rules," *Transportation Research Part C: Emerging Technologies*, vol. 98, pp. 152-166, 2019.
- [25] D. Venkatesan, K. Kannan, and R. Saravanan, "A genetic algorithm-based artificial neural network model for the optimization of machining processes," *Neural Computing and Applications*, vol. 18, pp. 135-140, 2009.
- [26] X. Wang, X. Lin, and X. Dang, "Supervised learning in spiking neural networks: A review of algorithms and evaluations," *Neural Networks*, vol. 125, pp. 258-280, 2020.
- [27] Y. Wang, "A Machine-Learning Model Using Artificial Neural Network to Facilitate Liver SBRT Prescription Selection By Predicting Normal Liver Geud Based on Geometric Properties of Liver and PTV," *International Journal of Radiation Oncology*Biolog*Physics*, vol. 105, pp. E139, 2019.
- [28] X. Yao, "A review of evolutionary artificial neural networks," *International journal of intelligent systems*, vol. 8, pp. 539-567, 1993.
- [29] J. R. Busemeyer, E. Byun, E. L. Delosh, and M. A. McDaniel, "Learning functional relations based on experience with input-output pairs by humans and artificial neural networks," 1997.
- [30] A. Yastrebov, S. Gad, G. Slon, A. Kulakowski, and E. Vinogradova, "Artificial Neural Networks in Rule-based Decision Systems," in *EUROCON 2007 - The International Conference on "Computer as a Tool"*, 2007, pp. 686-691.
- [31] A. Zhang, H. Zhou, X. Li, and W. Zhu, "Fast and robust learning in Spiking Feed-forward Neural Networks based on Intrinsic Plasticity mechanism," *Neurocomputing*, vol. 365, pp. 102-112, 2019.

- [32] T. Ejima, Y. Kato, Y. Tan, and K. Ishigaki, "Character Recognition By A Neural Network With Fuzzy Partitioning Units," in *Artificial Neural Networks*, T. Kohonen, K. MÄKisara, O. Simula, and J. Kangas, Eds., ed Amsterdam: North-Holland, 1991, pp. 1777-1780.
- [33] R. N. Naguib and G. V. Sherbet, *Artificial neural networks in cancer diagnosis, prognosis, and patient management*: CRC press, 2001.
- [34] G. Bakırlı, "An incremental genetic algorithm and neural network for classification and sensitivity analysis of their parameters," 2009.
- [35] A. H. Ali and M. Z. Abdullah, "Recent trends in distributed online stream processing platform for big data: Survey," in *2018 1st Annual International Conference on Information and Sciences (AiCIS)*, 2018, pp. 140-145.
- [36] A. H. Ali and M. Z. Abdullah, "A novel approach for big data classification based on hybrid parallel dimensionality reduction using spark cluster," *Computer Science*, vol. 20, 2019.
- [37] M. A. H. Ali, "An Efficient Model for Data Classification Based on SVM Grid Parameter Optimization and PSO Feature Weight Selection," *International Journal of Integrated Engineering*, 2018.
- [38] A. Esposito, S. Rampone, and R. Tagliaferri, "Addition and subtraction in neural nets as results of a learning process This work was supported in part by CNR, Progetto Finalizzato "Sistemi Informatici e Calcolo Parallelo", by MPI 40% and by IIASS," in *Artificial Neural Networks*, T. Kohonen, K. MÄKisara, O. Simula, and J. Kangas, Eds., ed Amsterdam: North-Holland, 1991, pp. 1789-1792.
- [39] C. He, M. Ma, and P. Wang, "Extract interpretability-accuracy balanced rules from artificial neural networks: A review," *Neurocomputing*, vol. 387, pp. 346-358, 2020.
- [40] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," *arXiv preprint arXiv:1510.00149*, 2015.
- [41] M. Ishikawa, "Neural networks approach to rule extraction," in *Proceedings 1995 Second New Zealand International Two-Stream Conference on Artificial Neural Networks and Expert Systems*, 1995, pp. 6-9.
- [42] S. H. Kim, S. G. Shin, S. Han, M. H. Kim, and C. H. Pyeon, "Feasibility study on application of an artificial neural network for automatic design of a reactor core at the Kyoto University Critical Assembly," *Progress in Nuclear Energy*, vol. 119, pp. 103183, 2020.
- [43] M. Nickel, K. Murphy, V. Tresp, and E. Gabrilovich, "A review of relational machine learning for knowledge graphs," *Proceedings of the IEEE*, vol. 104, pp. 11-33, 2015.
- [44] M. BİLGİLİ, "Predictions of Wind Speed and Wind Power Potential Using Artificial Neural Networks," ed: Doktora Tezi, Çukurova Üniversitesi, Fen Bilimleri Enstitüsü, 2007.
- [45] H. Ogi, H. Tanaka, Y. Akimoto, and Y. Izui, "APPROXIMATION CAPABILITIES OF NEURAL NETWORKS USING SAMPLING FUNCTIONS," in *Artificial Neural Networks*, T. Kohonen, K. MÄKisara, O. Simula, and J. Kangas, Eds., ed Amsterdam: North-Holland, 1991, pp. 1785-1788.
- [46] K. Patan and M. Patan, "Neural-network-based iterative learning control of nonlinear systems," *ISA Transactions*, vol. 98, pp. 445-453, 2020.
- [47] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85-117, 2015.
- [48] H. R. Ibraheem, Z. F. Hussain, S. M. Ali, M. Aljanabi, M. A. Mohammed, and T. Sutikno, "A new model for large dataset dimensionality reduction based on teaching learning-based optimization and logistic regression," *Telkomnika*, vol. 18, 2020.
- [49] N. D. Zaki, N. Y. Hashim, Y. M. Mohialden, M. A. Mohammed, T. Sutikno, and A. H. Ali, "A real-time big data sentiment analysis for iraqi tweets using spark streaming," *Bulletin of Electrical Engineering and Informatics*, vol. 9, pp. 1411-1419, 2020.
- [50] M. Sharifzadeh, A. Sikinioti-Lock, and N. Shah, "Machine-learning methods for integrated renewable power generation: A comparative study of artificial neural networks, support vector regression, and Gaussian Process Regression," *Renewable and Sustainable Energy Reviews*, vol. 108, pp. 513-538, 2019.
- [51] M. Sharma and R. Garg, "An artificial neural network based approach for energy efficient task scheduling in cloud data centers," *Sustainable Computing: Informatics and Systems*, vol. 26, p. 100373, 2020.