International Journal of Electrical and Computer Engineering (IJECE)
Vol. 13, No. 2, April 2023, pp. 2123~2130
ISSN: 2088-8708, DOI: 10.11591/ijece.v13i2.pp2123-2130

Enhancing the stability of the deep neural network using a non-constant learning rate for data stream

Hussein Abdul Ameer Abbas Al-Khamees, Nabeel Al-A'araji, Eman Salih Al-Shamery
Department of Software, Faculty of Information Technology, University of Babylon, Babil, Iraq

Article Info

Article history:

Received Dec 4, 2021
Revised Sep 30, 2022
Accepted Oct 30, 2022

Keywords:

Data stream
Deep neural network
Learning rate
Machine learning
Network performance

ABSTRACT

The data stream is considered the backbone of many real-world applications. These applications are most effective when using modern techniques of machine learning like deep neural networks (DNNs). DNNs are very sensitive to set parameters, the most prominent one is the learning rate. Choosing an appropriate learning rate value is critical because it is able to control the overall network performance. This paper presents a new developing DNN model using a multi-layer perceptron (MLP) structure that includes network training based on the optimal learning rate. Thereupon, this model consists of three hidden layers and does not adopt the stability of the learning rate but has a non-constant value (varying over time) to obtain the optimal learning rate which is able to reduce the error in each iteration and increase the model accuracy. This is done by deriving a new parameter that is added to and subtracted from the learning rate. The proposed model is evaluated by three streaming datasets: electricity, network security layer-knowledge discovery in database (NSL-KDD), and human gait database (HuGaDB) datasets. The results proved that the proposed model achieves better results than the constant model and outperforms previous models in terms of accuracy, where it achieved 88.16%, 98.67%, and 97.63% respectively.

Corresponding Author:

Hussein Abdul Ameer Abbas Al-Khamees
Software Department, Faculty of Information Technology, University of Babylon
Najaf Road, Babil, Iraq
Email: hussein.alkhamees7@gmail.com

1. INTRODUCTION

Recently, real-world applications such as sensor networks, different monitoring systems, social networks, and others are able to generate data streams, which are defined as huge data that have many different characteristics from traditional data, including boundless size (that cannot be stored in whole), high-speed, the appearance of concept drift (that is, data is not static but rather evolves over time) [1]. Neural networks are either shallow or deep which can be distinguished by having multiple hidden layers instead of a single layer. Deep learning techniques that use deep neural networks (DNN) have many distinctive features, such as strong and influential learning abilities, powerful generalization, the ability to train big data, and premium performance. Accordingly, deep learning ranks as the fastest growing and most successful among other machine learning techniques [2], [3]. Moreover, learning can be defined as a procedure for estimating the model parameters. So, deep learning can be seen as a universal learning rate that nearly is able to solve different problems [4].

Evidently, neural networks are very sensitive to set the parameters, the most notable one is the learning rate, which, like many other parameters, may change over time [5]. Lately, the learning rate issue has become a center of interest and attraction for researchers as it has a clear impact on achieving network stability and their results, thus leads to increase the DNN model accuracy [6]. Choosing an appropriate learning rate value is critical and essential since it is able to control network performance. For example, when the learning rate value is small, the network is easy to get stuck in the local minimum, while a large value avoids the local minimum. Subsequently, obtaining an optimal learning rate value is still an open challenge for DNN models as long as often, the learning rate is set to a constant value along the work [7].

Lewkowycz *et al.* [8] tested two values of the learning rate, the first according to a large learning rate and the second for a small one. Therefore, there are two regimes based on these values and a transitional phase separates these two regimes. They generally proved the best performance of the neural networks when the learning rate is large; this is known as the catapult phase. This phase avoids the divergence as well as the high curvature minimum.

The network is trained through a single cycle of learning rate that has a large learning rate [9]. This processing cycle ensures linearly to increase the learning rate to the highest value and then start decreasing all the way to the end. Practically speaking, in this method, the performance is better than standard methods, especially in the case of limited training data. Furthermore, it is generally characterized by limited training periods, hence, increasing the accuracy of the model.

Leclerc *et al.* [10] suggested a method for separating two training phases (regimes). The first one is the large step regime: this regime reflects the highest learning rate which does not lead to divergence, whilst its performance is poor from the optimization aspect. The second one is the small step regime: it reflects the greatest learning rate and from it, the loss begins to decrease constantly, and it is also poor from the generalization aspect. Therefore, for each regime, the processing was detached based on a specific algorithm. To explain deeply the learning rate and how it affects deep learning, Nakkiran [11] proposed a method to compare the error of the test data between a large and small value of learning rate. Through this work, the stability at a large value of the learning rate is proven to be impracticable. Alternatively, the process starts with a large learning rate value and continuously decreases this value until it reaches the target.

This paper presents a new developing DNN model using a multi-layer perceptron (MLP) structure that includes network training based on the optimal learning rate. Therefore, this model consists of three hidden layers and does not adopt the stability of the learning rate, but rather has a non-constant value (varying over time) to obtain the optimal learning rate which is able to reduce the error in each iteration and increase the model accuracy. This is done by deriving a new parameter that is first added to the learning rate value and then subtracted from it to get the lowest error.

The proposed DNN model is evaluated by different datasets that are; electricity, network security layer-knowledge discovery database (NSL-KDD), and four sub-datasets from the human gait database (HuGaDB). The results proved that the proposed model outperforms constant models where it achieved an accuracy of 88.16%, 98.67%, and 97.63%, respectively. Furthermore, the proposed DNN method outperformed the previous models. In addition to accuracy, three other measurements, precision, recall, and F1-score were used.

2. RESEARCH METHOD

This section describes the methods of this research in two sub-sections. The first sub-section clarifies the neural network types and the main difference between them. Then, it explains the MLP structure. While the second sub-section illustrates the importance of the learning rate and its impact on neural network performance.

2.1. Neural network types

Architecturally, the neural network is either shallow or deep. The main difference between them is that the shallow uses a single hidden layer, whereas the DNN use multiple hidden layers. Therefore, a DNN is an artificial neural network that has multi-hidden layers located between the input and output layers where every layer utilizes the former layer output as an input so, the neurons in DNN layers form the hierarchy [12]. Therefore, when deep learning first appeared it was known as hierarchical learning [13]. MLP which is sometimes known as back-propagation is a neural network framework that uses more than a hidden layer and these layers are connected on a feed-forward network [14]. The general architecture of the proposed MLP model consists of three hidden layers in addition to one input and output layer as depicted in Figure 1.

2.2. Learning rate value

The learning rate is able to control the network performance, so it is carefully chosen to ensure the best performance [15] where it controls the amount of change in the model based on the estimated error every time an update of the model parameters has been updated, such as the weights [16]. There is no general way to assign a specific learning rate value to all models, each one has a specific value that varies according to the model's task and data behavior [17].

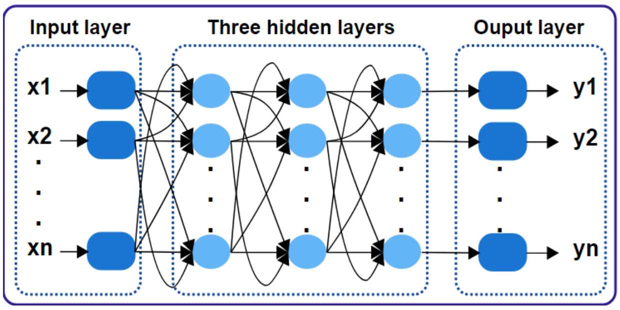


Figure 1. The general architecture of the MLP model

Many previous models set an initial value of the learning rate, and then this value increases or decreases linearly or exponentially according to the model structure. As a rule, the value of the learning rate is a small positive, typically in the range [0 to 1] and the most common values are 0.1, 0.001 and 0.0001. Generally, when the learning rate is small, the network convergence achieves a satisfactory level, but it needs many training epochs and a lot of time. In contrast, if the learning rate is large, the network is diverging and needs a little training epoch and less time.

Nevertheless, the algorithms that determine the learning rate value can classify into, constant and adaptive algorithms [18] besides, it can be classified according to the batch size numbers, batch gradient descent (if all training examples are treated as a single batch), stochastic gradient descent (if batch size just one) and minibatch gradient descent (if batch size more than a batch) [19].

3. THE PROPOSED METHOD

The proposed DNN model consists of four phases as shown in Figure 2.

1) Pre-processing phase: The normalization technique used in this phase and min-max method is implemented. Mathematically, if there is a set of matching scores (*Ms*) where, *s=1,2, ..., n* the normalized scores (*Ms'*) calculate by (1).

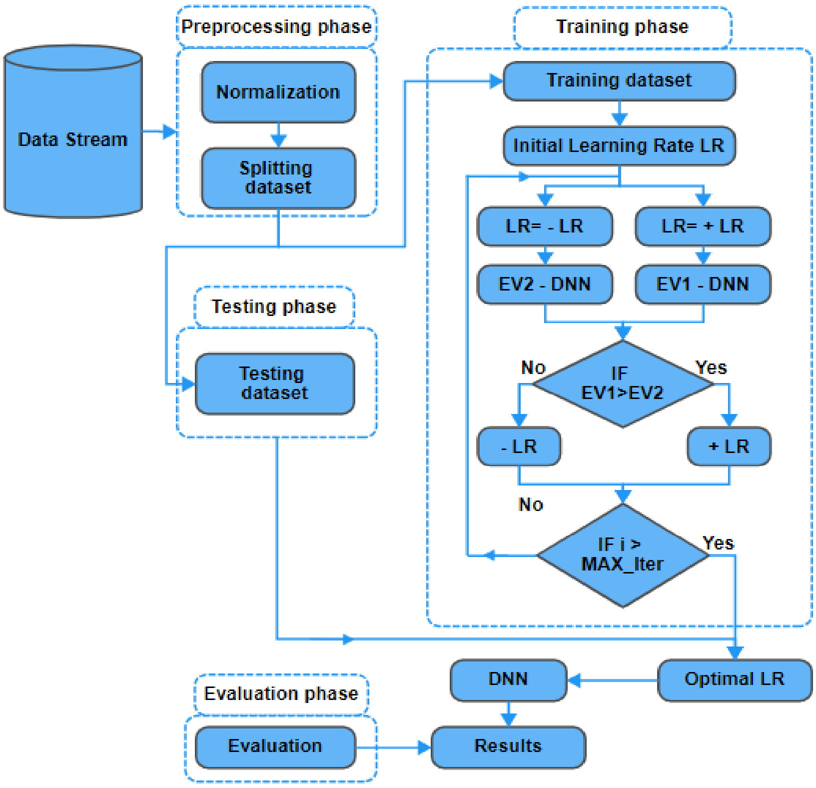


Figure 2. The block diagram of the proposed model

$$Ms' = (Ms - min)/(max - min) \quad (1)$$

where *min* is the minimum value and *max* is the maximum value. Then, the stream dataset is divided into training data as 80% and testing data as 20%.

2) Training phase: Implementation of the proposed model by applying a non-constant learning rate using the training data from the previous step to get the optimal learning rate.
3) Testing phase: It tests the ability of the model if it is trained accurately after the optimal learning rate has been obtained.
4) Evaluation phase: It is done by applying different measurements.

In the training phase, we derive a new parameter that is a lambda (λ). However, this phase starts with setting the initial value to the learning rates then the value of lambda is added and subtracted to this learning rate value respectively. Thereafter, each process is evaluated by the DNN to determine which is the best (adding or subtracting process). This determination is done by *IF* condition. All the above processes are performed as long as the current iteration (i) is less than the maximum iteration (Max_{iter}). Otherwise, the optimal learning rate is returned. The pseudocode of the proposed model is described in Figure 3.

```
Input: Stream dataset D, Initial Learning Rate
Output: optimal learning rate (OLR)
Set parameters as: λ_min= 0.0005, λ_max= 0.0009, Max_LR=0.1, Min_LR= 0.0, LR= 0.001,
i=1, Max_iter=100.
While i < Max_iter do
   Υ=(λ max - λ min)
   λ=λ max - Υ * (current_iter / max_iter)^2
  Temp_1LR ← 0.001
   Temp_2LR ← 0.001
   Temp_1LR ← Temp_1LR + λ
   Temp_2LR ← Temp_1LR - λ
      IF Temp_1LR > Max_LR Then
         Temp_1LR ← Max_LR
        EV_1 ← DNN (Temp_1LR, data)
      End if
      IF Temp_2LR < Min_LR Then
         Temp_2LR ← Min_LR
        EV_2 ← DNN (Temp_2LR, data)
      End if
        IF EV_1 > EV_2 Then
           LR ← Temp_1LR
         Else
           LR ← Temp_2LR
       End if
 i= i+1
End While
 Return OLR
```

Figure 3. The pseudocode of proposed model

The λ_{max} and λ_{min} refer to the maximum and minimum values of λ parameter respectively i.e., λ boundary. However, this model implements in the learning rate range [0.0, 0.1] so, two learning rate boundaries are set as follows, Max_{LR}=0.1, Min_{LR}=0.0, besides, the initial learning rate (LR) is 0.001 and also Max_{iter} refers to the maximum iteration of the algorithm, which set to 100. $Temp_{1LR}$ and $Temp_{2LR}$ refer to the temporary saving of learning rate values. Finally, EV_1 and EV_2 indicate the evaluation step of the selected learning rate values by DNN. The testing phase used the testing data that represents the input to the optimal learning rate. Thereafter, a DNN is applied to produce the results.

The last phase in the current model is the evaluation phase, which represents the result's evaluation. Four different measurements were applied, which are: accuracy, precision, recall, and F1-score. The streaming datasets to this model as described in Table 1 [20]–[23].

Based on the change in the learning rate during the first epochs, and then determining the optimal one for network training, the proposed model certainly will be stable after a number of epochs. In fact, the proposed idea might be generalizable; therefore, several types of stream data have been tested in this paper. In this diversity of stream datasets, we have taken care to include both balanced and unbalanced data.

Table 1. The description of streaming datasets

No.	Dataset name	Year	No. of samples	No. of features	No. of classes	Dataset symbol
1.	Electricity	1996	45,312	8	2	Electricity
2.	NSL-KDD	2009	148,517	41	5	NSL-KDD
3.	HuGaDB-v2-various-01-01	2017	2,435	39	4	DS1
4.	HuGaDB-v2-various-05-12	2017	4,393	39	3	DS2
5.	HuGaDB-v2-various-13-11	2017	5,272	39	3	DS3
6.	HuGaDB-v2-various-14-05	2017	2,392	39	2	DS4

4. RESULTS AND DISCUSSION

This section explains the results attained by applying the proposed model (which consists of three hidden layers) to train the network by the OLR and hence, reducing the error rate as the training progress (this is obviously explained by the difference between the constant and the proposed models). The first dataset is the electricity, the error rate in the constant model is 0.1424, while in the proposed model it is 0.1183 so, the error enhancement through the proposed model is 0.0241. The OLR is 0.0012.

The second dataset is the NSL-KDD, the error rate resulting from the constant model is 0.0165, whereas it through the proposed model is 0.0132. Therefore, the amount of error enhancement by the proposed model is 0.0033 with OLR as 0.0015. The third dataset is HuGaDB, as we mentioned above, we tested four sub-datasets. In DS1, the error is reduced from 0.0164 to 0.0136 by this model thus, the amount of error enhancement is 0.0027, with 0.0015 as OLR. For DS2, the resulting error rate of the constant model is 0.0295, whilst in the proposed model it 0.0288. There is 0.0007 reduced error rate and OLR is 0.0009. For DS3, the error decreased from 0.1649 to 0.0493 by applying our model and 0.1156 is the error enhancement. However, the OLR is 0.0017. Finally, in DS4, the error rate in constant model is 0.0055 that decreased to 0.0027 by our model i.e., the error enhancing is 0.0027 with OLR as 0.0009. All these decreasing error rates are depicted in Figure 4.

As seen in Figure 4, the results achieved by the proposed model (which are highlighted in bold font) are actually less than the results of the constant model. Because the network is trained by OLR, the accuracy of the proposed model is better than that of the constant model. Figure 5 illustrates the accuracy results. Moreover, Table 2 displays the results of other measurements: precision, recall, and F1-score.

Furthermore, based on the accuracy of the electricity dataset which is 88.16%, this model superiors the MLP model of [24] which achieved an accuracy of 81.06%. For the accuracy of the NSL-KDD dataset which is 98.67%, the proposed model outperforms the model presented by [25] that attained an accuracy of 97.05% and also the model proposed by [26] which achieved an accuracy of 97.97%. In terms of the accuracy of HuGaDB dataset which is 97.63%, the proposed model outperforms previous models such as [6] that attained an accuracy of 92.5%, [12] which achieved an accuracy of 88.0%, and [27] that obtained 91.7% as an accuracy. After implementing our proposed idea to a previous MLP model which has two hidden layers and a constant learning rate (0.001) that attained an accuracy rate of 50.2% [28], the accuracy rate increased to 85.80%.

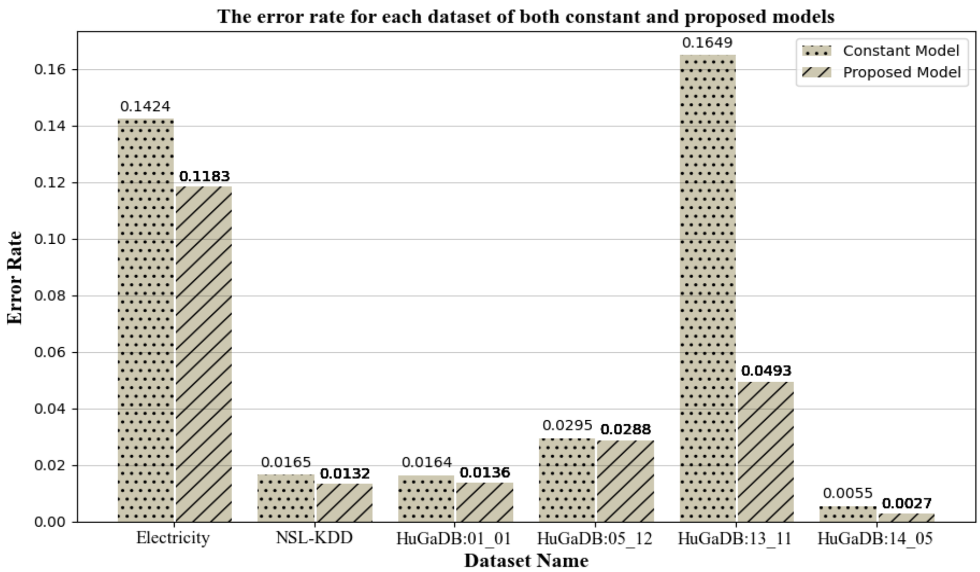


Figure 4. The error rate of both constant and proposed models

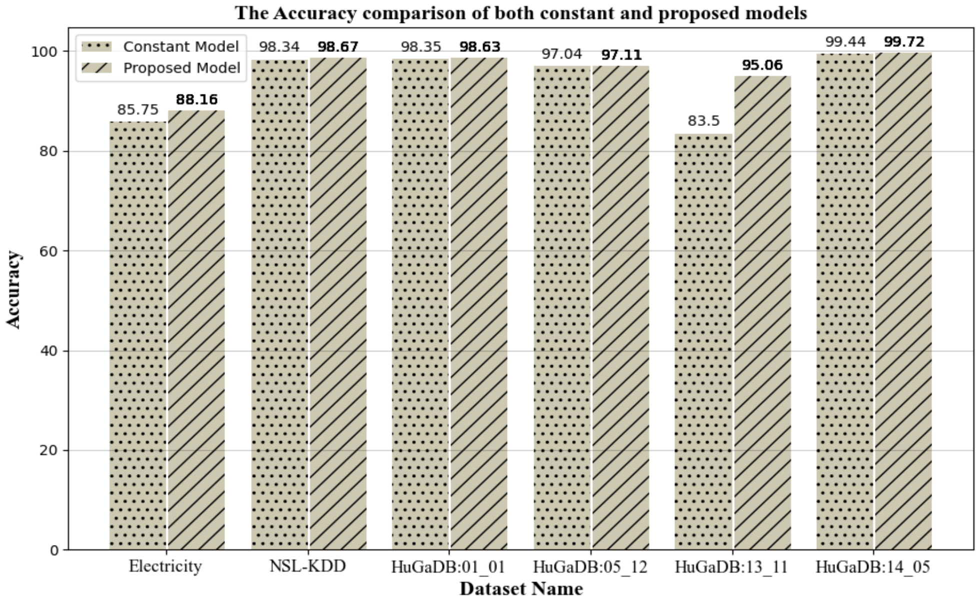


Figure 5. The accuracy of both constant and proposed models

Table 2. The measurements of both constant and proposed models

Dataset name	Constant model			Proposed model		
	Precision	Recall	F1-score	Precision	Recall	F1-score
Electricity	85.67	85.75	85.71	88.37	88.16	88.27
NSL-KDD	98.44	98.34	98.39	98.67	98.67	98.67
DS1	98.38	98.35	98.37	98.64	98.63	98.63
DS2	97.06	97.04	97.05	97.12	97.11	97.12
DS3	83.56	83.50	83.53	95.11	95.06	95.09
DS4	99.44	99.44	99.44	99.72	99.72	99.72

5. CONCLUSION

The data stream can be generated by many real-world applications. This type of data appears as a modern type that is defined as a huge amount of data arriving with a high speed that is not static but evolving over time, this causes the appearance of concept drift. Deep learning is one of the most important and successful machine learning techniques that are very sensitive to set parameters including the learning rate, which like other parameters may not be constant all the time, this is to achieve the best network performance. Remarkably, obtaining the optimal learning rate remains a major challenge for deep learning techniques. This paper presents a new developing DNN model that aims to get the optimal learning rate through several iterations in a step to reduce the error generated by the network thus, increase the model accuracy. The core idea is to derive a new parameter that will be added to the learning rate and then subtracted from it to get the least error. Practically, the proposed model proved to be effective and outperformed both the constant models (which adopt constant learning rate values) as well as the previous models tested by the same streaming datasets. This model is tested by different streaming datasets that are electricity, NSL-KDD and four sub-datasets from HuGaDB and it achieved an accuracy of 88.16%, 98.67%, and 97.63% respectively.

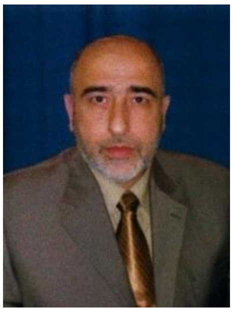
REFERENCES

[1] H. A. Al-Khamees, N. Al-A'araji, and E. S. Al-Shamery, "Survey: clustering techniques of data stream," in *2021 1st Babylon International Conference on Information Technology and Science (BICITS)*, Apr. 2021, pp. 113–119, doi: 10.1109/BICITS51482.2021.9509923.
[2] P. Saikia, R. D. Baruah, S. K. Singh, and P. K. Chaudhuri, "Artificial neural networks in the domain of reservoir characterization: A review from shallow to deep models," *Computers and Geosciences*, vol. 135, Feb. 2020, doi: 10.1016/j.cageo.2019.104357.
[3] H. Alkhamees, W. Al-Jwaid, and E. Al-Shamery, "The impact of using convolutional neural networks in COVID-19 tasks: A survey," *International Journal of Computing and Digital Systems*, vol. 11, no. 1, pp. 1157–1165, Mar. 2022, doi: 10.12785/ijcds/110194.
[4] A. Boukhalfa, A. Abdellaoui, N. Hmina, and H. Chaoui, "LSTM deep learning method for network intrusion detection system," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 3, pp. 3315–3322, Jun. 2020, doi: 10.11591/ijece.v10i3.pp3315-3322.

[5] T. Elsken, J. H. Metzen, and F. Hutter, “Simple and efficient architecture search for convolutional neural networks,” in *ICLR 2018 Conference*, 2018, pp. 1–14.
[6] M. Javeed, M. Gochoo, A. Jalal, and K. Kim, “HF-SPHR: Hybrid features for sustainable physical healthcare pattern recognition using deep belief networks,” *Sustainability*, vol. 13, no. 4, Feb. 2021, doi: 10.3390/su13041699.
[7] N. Jinsakul, C.-F. Tsai, and C.-E. Tsai, “Anomaly chicken cell identification using deep learning techniques,” *Journal of Information Science and Engineering*, vol. 37, no. 4, pp. 827–838, 2021, doi: 10.6688/JISE.202107.
[8] A. Lewkowycz, Y. Bahri, E. Dyer, J. Sohl-Dickstein, and G. Gur-Ari, “The large learning rate phase of deep learning: the catapult mechanism,” in *ICLR 2021 Conference*, 2020, pp. 1–28.
[9] L. N. Smith and N. Topin, “Super-convergence: very fast training of neural networks using large learning rates,” in *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, May 2019, vol. 110061, doi: 10.1117/12.2520589.
[10] G. Leclerc and A. Madry, “The two regimes of deep network training,” *arXiv:2002.10376*, 2020.
[11] P. Nakkiran, “Learning rate annealing can provably help generalization, even for convex problems,” in *OPT2020: 12th Annual Workshop on Optimization for Machine Learning*, 2020, pp. 1–1.
[12] Y. Sun, G. Yang, and B. Lo, “An artificial neural network framework for lower limb motion signal estimation with foot-mounted inertial sensors,” in *2018 IEEE 15th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*, Mar. 2018, pp. 132–135, doi: 10.1109/BSN.2018.8329676.
[13] R. Vargas and L. Ruiz, “Deep learning: Previous and present applications,” *Journal of awareness*, vol. 2, no. 3, pp. 11–20, 2017.
[14] H. Moayedi and A. Mosavi, “Synthesizing multi-layer perceptron network with ant lion biogeography-based dragonfly algorithm evolutionary strategy invasive weed and league champion optimization hybrid algorithms in predicting heating load in residential buildings,” *Sustainability*, vol. 13, no. 6, Mar. 2021, doi: 10.3390/su13063198.
[15] G. Ren, Y. Cao, S. Wen, T. Huang, and Z. Zeng, “A modified Elman neural network with a new learning rate scheme,” *Neurocomputing*, vol. 286, pp. 11–18, Apr. 2018, doi: 10.1016/j.neucom.2018.01.046.
[16] N. Loizou, S. Vaswani, I. Laradji, and S. Lacoste-Julien, “Stochastic Polyak step-size for SGD: an adaptive learning rate for fast convergence,” *International Conference on Artificial Intelligence and Statistics*, pp. 1306–1314, Feb. 2020.
[17] K. Pasupa and W. Sunhem, “A comparison between shallow and deep architecture classifiers on small dataset,” in *2016 8th International Conference on Information Technology and Electrical Engineering (ICITEE)*, 2016, pp. 1–6, doi: 10.1109/ICITEED.2016.7863293.
[18] L. Luo, Y. Xiong, Y. Liu, and X. Sun, “Adaptive gradient methods with dynamic bound of learning rate,” in *ICLR 2021 Conference*, Feb. 2019, pp. 1–21, *arXiv:1902.09843*.
[19] H. Jindal, N. Sardana, and R. Mehta, “Analyzing performance of deep learning techniques for web navigation prediction,” *Procedia Computer Science*, vol. 167, no. 2019, pp. 1739–1748, 2020, doi: 10.1016/j.procs.2020.03.384.
[20] V. M. A. Souza, D. M. dos Reis, A. G. Maletzke, and G. E. A. P. A. Batista, “Challenges in benchmarking stream learning algorithms with real-world data,” *Data Mining and Knowledge Discovery*, vol. 34, no. 6, pp. 1805–1858, Nov. 2020, doi: 10.1007/s10618-020-00698-5.
[21] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, “Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study,” *Journal of Information Security and Applications*, vol. 50, Feb. 2020, doi: 10.1016/j.jisa.2019.102419.
[22] H. Al-Khamees, “Data stream clustering using fuzzy-based evolving Cauchy algorithm,” *International Journal of Intelligent Engineering and Systems*, vol. 14, no. 5, pp. 348–358, Oct. 2021, doi: 10.22266/ijies2021.1031.31.
[23] R. Chereshnev and A. Kertész-Farkas, “HuGaDB: Human gait database for activity recognition from wearable inertial sensor networks,” in *in International Conference on Analysis of Images, Social Networks and Texts, Moscow, Russia*, 2018, pp. 131–141.
[24] A. Abbasi, A. R. Javed, C. Chakraborty, J. Nebhen, W. Zehra, and Z. Jalil, “ElStream: An ensemble learning approach for concept drift detection in dynamic social big data stream learning,” *IEEE Access*, vol. 9, pp. 66408–66419, 2021, doi: 10.1109/ACCESS.2021.3076264.
[25] A. S. Kyatham, M. A. Nichal, and B. S. Deore, “A novel approach for network intrusion detection using probability parameter to ensemble machine learning models,” in *2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC)*, Mar. 2020, pp. 608–613, doi: 10.1109/ICCMC48092.2020.ICCMC-000113.
[26] A. S. Ahanger, S. M. Khan, and F. Masoodi, “An effective intrusion detection system using supervised machine learning techniques,” in *5th International Conference on Computing Methodologies and Communication (ICCMC)*, Apr. 2021, pp. 1639–1644, doi: 10.1109/ICCMC51019.2021.9418291.
[27] H. A. A. Al-Khamees, N. Al-A’araji, and E. S. Al-Shamery, “Classifying the human activities of sensor data using deep neural network,” in *Intelligent Systems and Pattern Recognition*, 2022, pp. 107–118.
[28] B. Heuju and D. S. Baral, “Anomaly detection in computer networks using multilayer perceptron,” in *Proceedings of 10th IOE Graduate Conference*, 2021, vol. 10, pp. 1367–1374.

BIOGRAPHIES OF AUTHORS

Hussein Abdul Ameer Abbas Al-Khamees received a B.Sc. degree in computer science from the University of Babylon, Iraq in 1999. He received an M.Sc. degree in information technology from the University of Turkish Aeronautical Association, Institute of Science and Technology, Ankara, Turkey in 2017. He is currently preparing for his Ph.D. degree in Software Department, Babylon University. His main research interests are data mining, data stream analysis, deep learning, and intelligent systems. He can be contacted at email: hussein.alkhamees7@gmail.com.



Nabeel Al-A'araji received his B.Sc. degree in mathematics from Al-Mustansiryah University, Iraq, in 1976. And he received an M.Sc. degree in mathematics from University of Baghdad, Iraq, in 1978. And he received a Ph.D. degree in mathematics from the University of Wales, Aberystwyth, UK, in 1988. He is currently a professor at the Software Department, University of Babylon. His research interests include artificial intelligence, geographic information system (GIS), machine learning, neural networks, deep learning, and data mining. He can be contacted at email: nhkaghed@itnet.uobabylon.edu.iq.

Eman Salih Al-Shamery received the B.Sc. and M.Sc. degrees in computer science from the University of Babylon, Iraq, in 1998 and 2001, respectively. After completing her M.Sc., she worked as an assistant lecturer at the Department of Computer Science, University of Babylon. In 2013, she received her Ph.D. in computer science from the University of Babylon. Currently, she holds a professor position at Software Department, University of Babylon. Her current research interests include artificial intelligence, bioinformatics, machine learning, neural networks, deep learning, and data mining. She can be contacted at email: emanalshamery@itnet.uobabylon.edu.iq.