Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 1522179, 7 pages
https://doi.org/10.1155/2022/1522179

Research Article

Federated Deep Learning Approaches for the Privacy and Security of IoT Systems

Malik Bader Alazzam,[1] **Fawaz Alassery**,[2] **and Ahmed Almulihi**[3]

[1]*Faculty of Computer Science and Informatics, Amman Arab University, Jordan*
[2]*Department of Computer Engineering, College of Computers and Information Technology, Taif University, P.O. Box 11099, Taif 21944, Saudi Arabia*
[3]*Department of Computer Science, College of Computers and Information Technology, Taif University, P.O. Box 11099, Taif 21944, Saudi Arabia*

Correspondence should be addressed to Malik Bader Alazzam; m.alazzam@aau.edu.jo

Received 27 October 2021; Revised 6 November 2021; Accepted 18 March 2022; Published 1 April 2022

Academic Editor: Shalli Rani

Using federated learning, which is a distributed machine learning approach, a machine learning model can train on a distributed data set without having to transfer any data between computers. Instead of using a centralised server for training, the model uses data stored locally on the device itself. After that, the server uses this model to create a jointly trained model. Federated learning asserts that privacy is preserved because no data is sent. Botnet attacks are detected using on-device decentralised traffic statistics and a deep autoencoder. This proposed federated learning approach addresses privacy and security concerns about data privacy and security rather than allowing data to be transferred or relocated off the network edge. In order to get the intended results of a previously centralised machine learning technique while also increasing data security, computation will be shifted to the edge layer. Up to 98% accuracy is achieved in anomaly detection with our proposed model using features like MAC IP and source/destination/IP for training. Our solution outperforms a standard centrally managed system in terms of attack detection accuracy, according to our comparative performance analysis.

1. Introduction

While [1] is credited with coining the term "federated learning," the first description of its implementation can be found in [2]. Multiple devices work together to train a shared model in federated learning. Multiple clients' parametric improvements are combined over numerous training rounds to achieve this. Several customers compete in each round to improve a globally available model using data that they have access to only locally. Figures 1 and 2 show the steps in such a round.

Because the models are assumed to be smaller in size than the data set, federated learning reduces data transfer while also addressing privacy concerns associated with sending personal information to a server [3]. Another advantage is that all computation can be performed on the clients' devices. Maintaining server farms, calculating new models, and dealing with large amounts of data are all made easier as a result.

While federated learning's round-based nature means models are smaller than the amount of data that can be exchanged, it is possible that a significant amount of bandwidth will be needed. Federated learning's communication costs should be reduced, especially for mobile users with limited data access. As a result, a number of communication cost-cutting techniques have been developed.

In some cases, Figure 3 federated learning outperforms existing models. Mobile device implementations for next word prediction and emoticon prediction [4] were demonstrated by Google researchers. These use cases show how effective federated learning can be in a variety of situations.

In Minneapolis, when one of these coupons is delivered to the home of a high school student, her father calls the

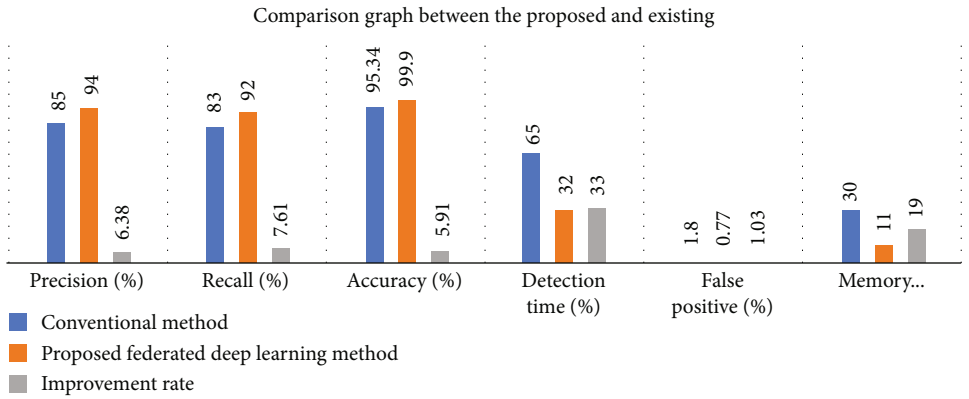

Figure 1: Comparison graph between the existing and proposed methods.

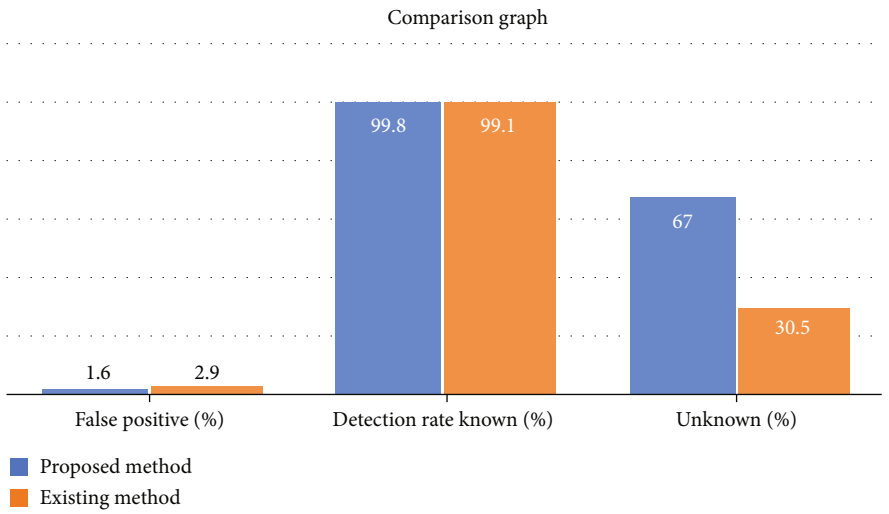

Figure 2: Comparison graph.

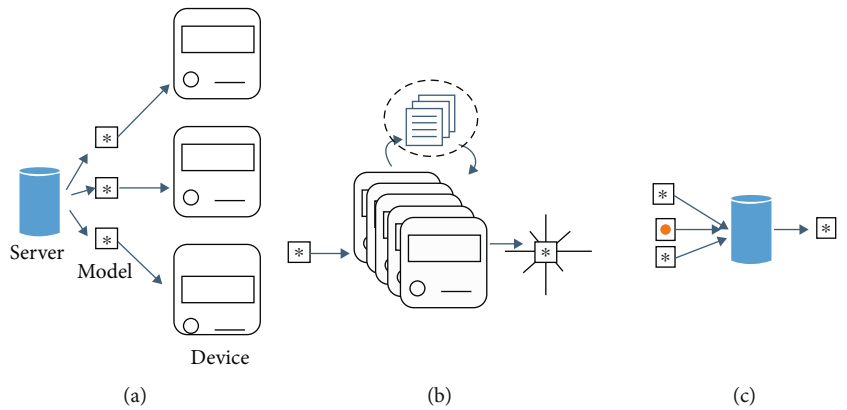

Figure 3: Working diagram of federated learning.

store manager and inquiries about the coupon's contents. It was only after this incident that Target's management began paying attention to customer complaints [5]. These stories can help us see how critical it is to safeguard personal information stored digitally.

2. Related Work

A federated learning approach to developing WID models is proposed by [6]. As shown in Figure 4, edge devices can first train their local models using local data. Local models are

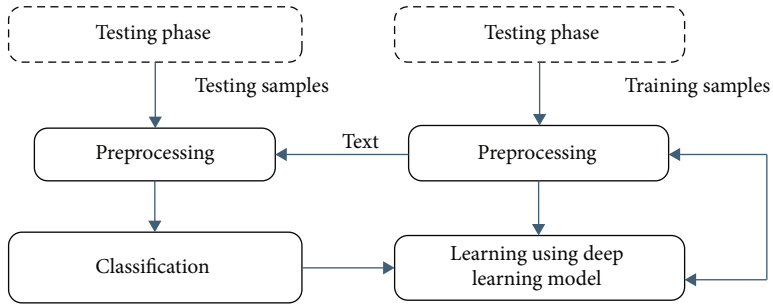

Figure 4: Deep learning-based classification process.

then averaged to create a global model. Edge devices do not have to share raw training data this way. These devices train a local model and send only model parameters to the server instead of raw training data.

2.1. Privacy and Security. Due to the fact that privacy and security are often used in the same context, it is critical to know the difference. Transferring data securely does not guarantee privacy, and keeping it private does not guarantee security from intruders. A malicious adversary is one whose primary objective is to obtain victim-specific personal data [7]. This particular victim is either a preselected candidate or was selected at random. This section also gives some background on privacy preservation from a legislative standpoint.

Privacy can be loosely defined as "the control to determine to whom personal information is revealed." As such, for a system to be privacy-preserving would, in theory, mean that it:

(i) Reveals no personal information to anyone other than those with consent

(ii) Reveals no nonconsensual personal information

In practice, however, what is regarded as personal information is only that which can be appointed specifically to the person. Security, on the other hand, can be defined as "the state of being free from danger or threat." In the current context, these are primarily unintended distribution of sensitive data (or data leakage). Whatever is done with the leaked data is in this context irrelevant but could be as malicious [8–11]. There is a notion in encryption and cybersecurity that it is impossible to have an entirely secure system. This stems from the fact that security systems must have a key or password of sorts. The space in which to define such a key is finite by design. A password for instance commonly has a maximum amount of characters and a finite set of characters to pick from. Therefore, a program can be written that checks all combinations of keys. This will crack the system given enough computational resources. This is called a brute force attack. In practice, keys of lengths higher than a certain length are commonly regarded as being unbreakable because the number of combinations to check would take far too much time and resources [12]. The costs of trying this would far exceed any value an adversary gets from breaking the security.

There is however a balancing act. Because performing the encryption and decryption also has an associated computational cost, which makes it a challenge to minimize security costs while maximizing the security of the system. Another notion of security is security by obscurity, meaning that you can minimize security risks by not disclosing the details of your protective measures [13]. This has been widely criticized and shall be cast aside as a viable security measure for the remainder of the thesis. Any adversary is considered to have complete knowledge of how the system works which includes complete knowledge of the implemented security measures.

3. Background

For security researchers and industry professionals, DL has recently become a hot topic. DNNs, also known as deep learning (DL), are a subset of AI that are inspired by how the brain works. Architectures based on deep learning (DL) can understand the meaning of large amounts of data and automatically update derived meaning without domain expert knowledge.

An important part of feature engineering is feature extraction, and doing so requires some familiarity with the subject matter [14]. The classifier's performance is dependent on feature extraction. The 1950s saw the introduction of the NN ML technique. It has the ability to automatically extract and classify features without the involvement of a human. To some extent, the classical NN performs admirably. However, using advanced NN, also known as deep learning, it is possible to completely avoid the phase of feature engineering (DL) [15–17]. Figure 5 shows the training and testing processes involved in traditional ML algorithms and DL architectures. Because of this, the DL was able to outperform other long-standing AI applications in a variety of fields.

Figure 4 depicts DL architecture classifications. The terms neural networks (NNs), machine learning (ML), and deep learning (DL) are all intertwined in AI discourse. All of these fields are frequently misunderstood. DL is a branch of machine learning that developed from neural networks (NNs). By processing data and generating patterns, this simulates the workings of the human brain. When it comes to DL, the most important part is the NNs, and the term "many NNs" usually means just that many NNs [18]. Vanishing and exploding gradients and, most importantly, the lack of

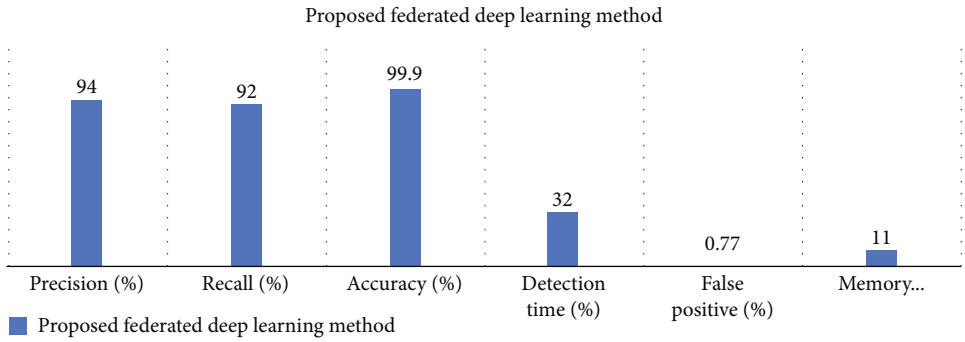

Figure 5: Parameter analysis graph for the proposed method.

high-performance computing systems arise when NNs are deep.

As computing systems have improved, new kinds of DL architectures have been introduced, and improvements have been made in optimizers, activation functions, loss functions, and the disappearing and exploding gradient issues. DL is now being used to solve a variety of cyber security problems, and it outperforms that classical ML in every case depicts two types of DL architecture: generative and discriminative [19]. Deep Boltzmann machine (DBM), deep autoencoder (DAE), deep belief network (DBN), and recurrent structures are used to generate new ideas. Recurrent structures and convolutional neural networks are used to discriminate between these new ideas and the old ones (CNN).

4. Proposed Architecture

Our process architecture is depicted in Figure 5. Botnet assaults are detected with the use of a decentralised FL-based deep anomaly detection engine. As depicted in the diagram, IoT security gateways are responsible for operating and monitoring traffic to and from IoT devices. FL and anomaly detection are two examples of forensics-based IoT security gateways. This is due to the fact that port mirroring keeps track of network traffic. The network traffic entering and departing the IoT security gateway is monitored since botnets might masquerade as regular traffic. Infected Internet of Things devices frequently transmit signals to unexpected locations [20]. Once connected to their FL server, which would host device models, the IoT security gateways will be able to communicate with one another. The security gateway will communicate with the FL server in order to determine the deep autoencoder model to use.

Based on data from the global FL server, which is connected to the same network as the proposed IoT device, the FL model is only applicable to computer learning. As discussed previously, the security gateway hardware can be customised to work with a wide range of devices and hardware. We claim that a security gateway can use port mirroring to record all network traffic and process it afterwards. We can exchange information with our virtual worker by taking a snapshot of it. These gateways can even host multiple virtual employees simultaneously. The truth is that each security gateway can host an unlimited number of virtual workers.

It is up to employees to obtain the company a specific gadget. Although a gateway can handle several virtual employees, only one is required per gateway. In this thesis, we will regard security gateways and virtual employees as one-to-one interactions.

4.1. Deep Autoencoder. Figure 5 shows a special deep learning algorithm that uses two symmetrical deep belief networks with four or five shallow layers. Half of the network encodes and decodes. Autoencoders are a subset of neural networks. PCA and PCA are closely related, but PCA is much more flexible. Unlike PCA, which can only perform linear transformations, autoencoders can encode data in nonlinear ways [21]. Using autoencoders can maximize data utilization by reducing reconstruction error. Each layer has the same number of neurons using autoencoders (input and output).

(i) This is done in the first step of the deep autoencoder, which uses PyTorch linear layers for all steps of the ML process, encoding and decoding continuously as each layer is added and subtracted. Data from the first layer represents the source IP, destination IP, and UDP/TCP socket details and is encoded to 75% of its original size before being sent to the second layer for decompression

(ii) The input from the previous layer will be passed on to the next sup sequential layer for encoding. Half of the input size will be encoded in the next encoding layer, reducing the size by 50%. The input is reduced by 33% in size in the third layer, which continues the encoding process. The input will be encoded down to 25% of the previous step again in the final encoding layer. The compression level is the lowest at this point

(iii) The effects of the encoding stage will be undone during the decoding stage. It decodes the input and then adds on to the size of it for the next layer, using the same encoding and decoding values and the same decompression aids as input features. Using the decoder's opposing direction helps to produce a decompressed data set that is not 1-to-1 identical to the input, as well as expanding and

zeroing out some data points to help produce the threshold

(iv) The output layer will recreate the encoding and decoding process. After encoding and decoding the network traffic's behaviours, a threshold is generated and used for testing by comparing the input and output

4.2. Dataset. Massive amounts of data are now available due to the proliferation of data collectors like smartphones [22]. In terms of building machine learning models, these data are priceless! New approaches, tailored for decentralised settings, are needed to make use of all this data. More data helps machine learning models because it allows them to be trained on a broader set of features rather than having to remember the details of each individual training example. Overfitting occurs when a neural network memorises training samples rather than looking for correlations in the general characteristics of an input. Machine learning models frequently overfit data. When data is gathered for training purposes from a variety of dispersed and possibly infrequently used devices, three common characteristics emerge.

4.3. Massively Distributed. Because data is stored across a large number of clients, the amount of data available to each client may be significantly smaller than the average amount of data available to each client.

4.4. Non-IID. When compared to other clients, the data provided for a particular client may be taken from dramatically different distributions. This means that the data that is readily available in the local area does not accurately reflect the broader data dispersion.

4.5. Example. The photographs stored on a cat enthusiast's mobile phone may be radically different from those kept on a vehicle enthusiast's mobile phone.

4.6. Unbalanced. The amount of data that is available for a single customer can vary significantly from one client to the next.

The centralised model is the most widely used machine learning technique for decentralised data since it is the most conventional. Because it is explained, it is possible to see how this model differs from that of collaborative techniques in practise.

4.7. Matrix. The following metrics are used to determine overall performance of the IDS model:

Detection accuracy: how many samples were correct out of the total sample population.

$$\text{Accuracy} = \frac{(\text{TP} + \text{TN})}{(\sum \text{P} + \sum \text{N})}. \tag{1}$$

Recall: fraction of relevant instances over the total amount of relevant instances.

Table 1: Parameters.

Parameters	Conventional methods	Proposed federated deep learning method	Improvement rate
Precision (%)	85	94	6.38
Recall (%)	83	92	7.61
Accuracy (%)	95.34	99.9	5.91
Detection time (s)	65	32	33
False positive (%)	1.8	0.77	1.03
Memory utilization (mb)	30	11	19

$$\text{Recall} = \frac{\text{TP}}{(\text{TP} + \text{FN})}. \tag{2}$$

F1 score: weighted average of the precision and recall.

$$\text{F1} = \frac{2 \times (\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}}. \tag{3}$$

False positive rate: the rate at which alerts are generated for normal samples.

$$\text{FPR} = \frac{(\text{FP})}{(\text{FP} + \text{TN})} = 1 - \text{precision}. \tag{4}$$

False negative rate: the rate at which attacks are missed.

$$\text{FNR} = \frac{(\text{FN})}{(\text{FN} + \text{TP})} = 1 - \text{Recall}. \tag{5}$$

5. Results and Discussion

Existing traditional procedures are contrasted with the federated deep learning method that is being proposed. The results are encouraging. The data set is being utilized to determine the effectiveness of the proposed method, which is being evaluated. First, the new approach is compared to the old one in terms of detection performance, and the results are compared. The results of the tests, which were carried out, are presented in Tables 1 and 2 as well as Figure 1. Table 3 has been demonstrated that the proposed method has a greater detection rate than current methods. The proposed detection approach is evaluated on the basis of criteria such as precision value, recall value, accuracy, detection time, false positives, and memory utilization, among others. Based on Table 1, it is clear that the new method outperforms the current one.

Figure 5 shows the TN, TP, FP, and FN rates for studies with input dimensions ranging from 15 to 115. These matrices represent the non-FL baseline and the proposed FL techniques lowest and highest tested input features.

Figure 1 displays false positives that resulted in results up to 43954 on the non-FL figure, contrast this with which shows the same parameters but using a multiworker

Table 2: Matrix comparison table between the existing and proposed method.

Metric	Proposed method	Existing method
False positive (%)	1.6	2.9
Detection rate known (%)	99.8	99.1
Unknown (%)	67	30.5

Table 3: Parameter analysis table for the proposed federated deep learning method.

Parameters	Proposed federated deep learning method
Precision (%)	94
Recall (%)	92
Accuracy (%)	99.9
Detection time (s)	32
False positive (%)	0.77
Memory utilization (mb)	11

technique. This is a positive reflection on the model, which maintains performance even when the number of workers increases. This applies to all input dimensions, including those with larger dimensions than the default. In Table 2, the non-FL model, for example, produced 31 false positives; however, the multiworker model, in Figure 1, produced 36 false positives, proving that the model's performance can be maintained across several workers.

6. Conclusion

Then, we demonstrate how to make use of these federated learning datasets in a simulated learning environment. If we compare federated deep learning to server-trained deep learning in the context of wireless intrusion detection, the results are similar. In contrast to the conventional deep learning approach, the suggested model does not transmit data to a central server, thereby safeguarding the privacy of the user. Because of this, they rush to repair and patch equipment, leaving new and existing networks exposed. For proactive threat detection, we demonstrated a viable proof-of-concept model. FL is a reliable performer in enterprise networks. This technique secures Internet of Things devices and allows for the creation of complicated machine learning models. On edge networks, gateways provide self-updating attack detection thanks to their self-learning capabilities. In the simulation, it is demonstrated that accuracy and scores are maintained when there are sufficient features to train a model. IoT devices connected to a corporate network can be protected through the use of a large variety of security gateways and devices.

Data Availability

The data underlying the results presented in the study are available within the manuscript.

Disclosure

The study was performed as a part of the Employment of Institutions.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

Acknowledgments

We deeply acknowledge Taif University for supporting this study through Taif University Researchers Supporting Project Number (TURSP-2020/344), Taif University, Taif, Saudi Arabia.

References

[1] S. Badotra, D. Nagpal, S. N. Panda, S. Tanwar, and S. Bajaj, "IoT-enabled healthcare network with SDN," in *In 2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, pp. 38–42, Noida, India, 2020.

[2] I. Udrea, I. Gheorghe Viorel, A. Cartal Laurentiu et al., "IoT solution for monitoring indoor climate parameters in open space offices," in *E3S Web of Conferences*, Romania, 2020.

[3] T. Alladi, V. Chamola, B. Sikdar, and K. R. Choo, "Consumer IoT: security vulnerability case studies and solutions," *IEEE Consumer Electronics Magazine*, vol. 9, no. 2, pp. 17–25, 2020.

[4] S. Almutairi, S. Mahfoudh, S. Almutairi, and J. S. Alowibdi, "Hybrid botnet detection based on host and network analysis," *Journal of Computer Networks and Communications*, vol. 2020, Article ID 9024726, 2020.

[5] H. Haddad Pajouh, R. Khayami, A. Dehghantanha, K.-K. R. Choo, and R. M. Parizi, "AI4SAFE-IoT: an AI-powered secure architecture for edge layer of Internet of Things," *Neural Computing and Applications*, vol. 32, no. 20, pp. 16119–16133, 2020.

[6] H. U. Rahman, M. A. Habib, S. Sarwar, N. Mahmood, M. Ahmad, and H. Ahmad, "Fundamental issues of future Internet of Things," in *In: 2020 International Conference on Engineering and Emerging Technologies (ICEET)*, pp. 1–6, Lahore, Pakistan, 2020.

[7] H. Haddad Pajouh, A. Azmoodeh, A. Dehghantanha, and R. M. Parizi, "MVFCC: a multi-view fuzzy consensus clustering model for malware threat attribution," *IEEE Access*, vol. 8, pp. 139188–139198, 2020.

[8] A. A. Hamad and L. M. Thivagar, "Conforming dynamics in the metric spaces," *Journal Of Information Science And Engineering*, vol. 36, no. 2, pp. 229–279, 2020.

[9] N. A. Noori and A. A. Mohammad, "Dynamical approach in studying GJR-GARCH (Q, P) models with application," *Tikrit Journal of Pure Science*, vol. 26, no. 2, pp. 145–156, 2021.

[10] W. A. Saeed and A. J. Salim, "Convergence solution for some harmonic stochastic differential equations with application," *Tikrit Journal of Pure Science*, vol. 25, no. 5, pp. 119–123, 2020.

[11] R. N. Salih and M. A. Al-jawaherry, "Finding minimum and maximum values of variables in mathematical equations by applying firefly and PSO algorithm," *Tikrit Journal of Pure Science*, vol. 25, no. 5, pp. 99–109, 2020.

[12] M. J. Sheller, G. A. Reina, B. Edwards, J. Martin, and S. Bakas, "Multi institutional deep learning modeling without sharing patient data: a feasibility study on brain tumor segmentation. Brain lesion: glioma, multiple sclerosis, stroke and traumatic brain injuries," *BrainLes (Workshop)*, vol. 11383, pp. 92–104, 2019.

[13] S. A. Salih and G. A. Zarraq, "Applying a mathematical model to simulate the ground water reservoir in Al-Alam area/North-east Tikrit city/Iraq," *Tikrit Journal of Pure Science*, vol. 26, no. 3, pp. 60–66, 2021.

[14] A. A. Hamad, G. N. Nguyen, and D.-N. Le, "Efficient dual-cooperative bait detection scheme for collaborative attackers on mobile ad-hoc networks," *IEEE Access*, vol. 8, pp. 227962–227969, 2020.

[15] M. K. Shahoodh, "The adjacency matrix of the compatible action graph for finite cyclic groups of p-power order," *Tikrit Journal of Pure Science*, vol. 26, no. 1, pp. 123–127, 2021.

[16] F. J. Suhae and A. I. Hussain, "Suitability evaluation of mud-stone of Injana formation for dam filling materials in Taq Taq area/Erbil/Iraq," *Tikrit Journal of Pure Science*, vol. 25, no. 3, pp. 49–56, 2020.

[17] M. L. Thivagar and A. A. Hamad, "A theoretical implementation for a proposed hyper-complex chaotic system," *Journal of Intelligent & Fuzzy Systems*, vol. 38, no. 3, pp. 2585–2590, 2020.

[18] S. R. Thanoon, "A comparison between Bayes estimation and the estimation of the minimal unbiased quadratic standard of the bi-division variance analysis model in the presence of interaction," *Tikrit Journal of Pure Science*, vol. 25, no. 2, pp. 116–123, 2020.

[19] S. A. Wuhaib and N. F. Abd, "Control of prey disease in stage structure model," *Tikrit Journal of Pure Science*, vol. 25, no. 2, pp. 129–135, 2020.

[20] S. Ramaswamy, R. Mathews, K. Rao, and F. Beaufays, "Federated learning for emoji prediction in a mobile keyboard," *Computation and Language*, vol. 22, 2019.

[21] L. Huang, A. L. Shea, H. Qian, A. Masurkar, H. Deng, and D. Liu, "Patient clustering improves efficiency of federated machine learning to predict mortality and hospital stay time using distributed electronic medical records," *Journal of Biomedical Informatics*, vol. 99, p. 103291, 2019.

[22] S. S. Noon, "Estimation the variogram function indicator which represent the transmissivity coefficient in the groundwater," *Tikrit Journal of Pure Science*, vol. 25, no. 5, pp. 110–118, 2020.