Approaches to Federated Computing for the Protection of Patient Privacy and Security Using Medical Applications

Osman Sirajeldeen Ahmed,1 Emad Eldin Omer,2 Samar Zuhair Alshawwa,3 Malik Bader Alazzam,4 and Reefat Arefin Khan5

1Ajman University, College of Humanities and Sciences, UAE
2Ajman University, College of Mass Communication, UAE
3Department of Pharmaceutical Sciences, College of Pharmacy, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia
4Information Technology College, Ajloun National University, Jordan
5IUBAT-International University of Business Agriculture and Technology, Dhaka, Bangladesh

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

Received 23 December 2021; Revised 13 January 2022; Accepted 27 January 2022; Published 11 February 2022

Academic Editor: Fahd Abd Algalil

Copyright © 2022 Osman Sirajeldeen Ahmed et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Computing model may train on a distributed dataset using Medical Applications, which is a distributed computing technique. Instead of a centralised server, the model trains on device data. The server then utilizes this model to train a joint model. The aim of this study is that Medical Applications claims no data is transferred, thereby protecting privacy. Botnet assaults are identified through deep autoencoding and decentralised traffic analytics. Rather than enabling data to be transmitted or relocated off the network edge, the problem of the study is in privacy and security in Medical Applications strategies. Computation will be moved to the edge layer to achieve previously centralised outcomes while boosting data security. Study results in our suggested model detects anomalies with up to 98 percent accuracy utilizing MAC IP and source/destination/IP for training. Our method beats a traditional centrally controlled system in terms of attack detection accuracy.

1. Introduction

While Badotra et al. [1] are credited with coining the term “Medical Applications”, the first description of its implementation can be found in [2]. Multiple devices work together to train a shared model in Medical Applications. 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.

Because the models are supposed to be smaller than the dataset, Medical Applications lowers data transmission while simultaneously addressing privacy issues [3]. Also, all calculations may be done on the customers’ devices. Maintaining server farms, developing new models, and handling enormous datasets become simpler.

While the round-based structure of Medical Applications implies models are smaller than data transmitted, it is feasible that large bandwidth may be required. In particular, mobile customers with restricted data access should have lower communication expenses. Many communication cost-saving strategies have emerged as a consequence.

To address privacy issues, Medical Applications has several customers that compete in each round to construct a globally available model using data that they only have access to locally. While the round-based structure of Medical Applications implies models are smaller than data supplied,
a large quantity of bandwidth may be required. Customers with restricted data access should save money.

In Figure 1, Medical Applications outperforms ex Google researchers that showed next word prediction and emoticon prediction [4] on mobile devices. These examples demonstrate the versatility of Medical Applications.

2. Related Work

A Medical Applications approach to developing WIDS models is proposed by Rahman et al. [5]; edge devices can first train their local models using local data. Local models are 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. It is critical to differentiate between privacy and security. Even encrypted data may be accessed by intruders. An adversary’s principal purpose is to get victim-specific personal data [6].

Privacy is “the control over who receives personal information”. In principle, a privacy-preserving technology would not reveal personal information to anybody other than those who had given authorization.

In reality, personal information is only that which can be assigned to a particular individual. Security, on the other hand, is “the absence of danger or threat”. In the contemporary context, this is mostly accidental data dispersion (or data leakage). What happens to the released data is unimportant but potentially dangerous [7–10]. In the world of encryption and cyber security, it is said that no system can be 100% safe. This is because security systems need a key or password. Define a key in limited space. A password, for example, usually has a maximum character count and a limited character set. So, software that verifies all key combinations may be built.

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 [11]. The costs of trying this would far exceed any value an adversary gets from breaking the security.

There is however a balancing acts because performing the encryption and decryption also has an associated computational cost, which makes it a challenge in minimizing 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 [12]. 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 computing (DL), are a subset of AI that are inspired by how the brain works. DL-based architectures may automatically update derived meaning without domain expert knowledge.

Feature extraction is a crucial aspect of feature engineering [13]. Feature extraction affects classification performance. The NN ML approach appeared in the 1950s. It can automatically extract and categorise characteristics without human intervention. The classical NN performs beautifully. Using sophisticated NN (deep learning), however, the feature engineering step may be totally avoided [14–16]. Figure 2 shows the training and testing procedures in classic ML and DL algorithms. As a result, the DL outperformed other long-standing AI applications in several sectors.

Figure 2 depicts DL architecture classifications. The terms neural networks (NNs), computing (ML), and computing (DL) are all intertwined in AI discourse. All of these fields are frequently misunderstood. DL is a branch of computing 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 [17]. Vanishing and exploding gradients and, most importantly, the lack of high-performance computing systems arise when NNs are deep.

New DL architectures have emerged, as have advancements in optimizers, activation functions, loss functions, and the vanishing and ballooning gradient
concerns. There are two kinds of DL architecture: generative and discriminative [18]. Ideas are generated using deep Boltzmann machines, deep autoencoders, deep belief networks, and recurrent structures. This is done using recurrent structures and convolutional neural networks (CNN).

3.1. Proposed Architecture. A decentralised FL-based deep anomaly detection engine detects botnet attacks. These security gateways operate and monitor traffic to and from Medical Applications equipment. Forensic-based Medical Applications security gateways include FL and anomaly detection. Port mirroring maintains track of network activity. The network traffic entering and leaving the Medical Applications security gateway is monitored for botnets. Infected computing devices typically send messages in unexpected directions [19]. The Medical Applications security gates may communicate after linked to its FL server, which hosts device models. The security gateway will ask the FL server for the deep autoencoder model.

Based on data from the worldwide FL server, which shares the same network as the proposed Medical Applications device, the FL model is exclusive for computers. As previously stated, the security gateway hardware may be configured to operate with a variety of devices. We claim a security gateway may employ port mirroring to capture and process network traffic. We may provide information to our virtual worker by snapping a picture. These portals may also accommodate many virtual workers. Unlimited virtual employees may be hosted by each security gateway.

It is up to employees to obtain the company a specific gadget. The many to one link between virtual works and security gates. 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.

3.2. Deep Autoencoder. 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 [20]. 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 subsequential 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, the same decompression aids as input features. Using the decoder’s opposing direction helps to produce a decompressed dataset 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.

![Figure 2: Computing-based classification process.](image-url)
3.2.1. 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.

3.2.2. 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.

3.2.3. 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.

3.2.4. 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 computing 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 practice.

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

(i) Detection accuracy: How many samples were correct out of the total sample population?

\[
\text{Accuracy} = \frac{TP + TN}{P+ N}. \tag{1}
\]

(ii) Recall: Fraction of relevant instances over the total amount of relevant instances.

\[
\text{Recall} = \frac{TP}{TP + FN}. \tag{2}
\]

(iii) F1 score: Weighted average of the precision and recall.

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

(iv) False positive rate: The rate at which alerts are generated for normal samples.

\[
FPR = \frac{FP}{FP + TN} = 1 - \text{precision}. \tag{4}
\]

(v) False negative rate: The rate at which attacks are missed.

\[
FNR = \frac{FN}{FN + TP} = 1 - \text{Recall}. \tag{5}
\]

4. Results and Discussion

Traditional processes are compared to the suggested federated computing method. Inspiring outcome of the data collection is being used to assess the method’s efficacy. First, the new approach’s detection performance is compared to the old ones. The results of the tests are shown in Tables 1, 2, and 3 and Figures 3 and 4. The suggested strategy outperforms existing methods in terms of detection rate. The suggested detection method is assessed for precision, recall, accuracy, detection time, false positives, and memory utilization. Table 2 clearly shows that the new strategy outperforms the old.

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 3 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 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 3, the non-FL model, for example, produced 31 false positives; however, the multiworker model, in Figures 3 and 4, produced 36 false positives, proving that the model’s performance can be maintained across several workers.

5. Conclusion

In this study, we explore how to use Medical Applications datasets in a virtual classroom. Federated and server-trained computing both perform well in terms of wireless intrusion detection. It does not communicate data to a central server, therefore ensuring user privacy. Then, they rush to patch and repair equipment, leaving new and old networks exposed. We built a model for proactive threat...
Detection. FL thrives in business networks, protects computer hardware, and allows complicated computing models. Self-learning gateways provide for edge network threat detection. As shown in the simulation, accuracy and scores are maintained when enough features are available. These devices may protect Medical Applications equipment connected to a corporate network.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

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

Acknowledgments

The authors extend their appreciation to Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2022R165), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

References

[1] S. Badotra, D. Naggal, S. N. Panda, S. Tanwar, and S. Bajaj, “IoT-enabled healthcare network with SDN,” 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, V. I. Gheorghe, L. A. Cartal et al., “IoT solution for monitoring indoor climate parameters in open space offices,” in 9th International Conference on Thermal Equipments, Renewable Energy and Rural Development (TE-RE-RD 2020), vol. 180, p. 02012, Constanta, Romania, 2020.

[3] T. Alladi, V. Chamola, B. Sikdar, and K. R. Choo, “Consumer IoT: security vulnerability case studies and solutions,” in 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, 16 pages, 2020.

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

[6] H. Haddad Pajohou, 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.

[7] A. A. Hamad and L. M. Thivagar, “Conforming dynamics in the metric spaces,” Journal of Information Science and Engineering, vol. 36, no. 2, pp. 279–291, 2020.

[8] 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.

[9] 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.

[10] 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.

[11] M. J. Sheller, G. Anthony Reina, B. Edwards, J. Martin, and S. Bakas, “Multi institutional computing modeling without sharing patient data: a feasibility study on brain tumor segmentation. Brain lesion: glioma, multiple sclerosis, stroke and traumatic brain injuries,” Brain Les (Workshop), vol. 11383, pp. 92–104, 2019.

[12] S. A. Salih and G. A. Zarraq, “Applying a mathematical model to simulate the ground water reservoir in Al-Alam area/Northeast Tikrit City/Iraq,” Tikrit Journal of Pure Science, vol. 26, no. 3, pp. 60–66, 2021.

[13] O. I. Khalaf, F. Ajesh, 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.

[14] 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.

[15] F. J. Suhae and A. I. Hussain, “Suitability evaluation of mudstone of Injana Formation for dam filling materials in TaqTaq area/Erbil/Iraq,” Tikrit Journal of Pure Science, vol. 25, no. 3, pp. 49–56, 2020.

[16] 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.

[17] 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.

[18] 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.

[19] S. Ramaswamy, R. Mathews, K. Rao, and F. Beaufays, “Medical Applications for emoji prediction in a mobile keyboard,-CoRR,” 2019, https://arxiv.org/abs/1906.04329.

[20] 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, article 103291, 2019.