Nanotechnology Perceptions
ISSN 1660-6795
www.nano-ntp.com

Secure Federated Learning for Collaborative Intrusion Detection Systems (IDS)

Utkarsh Parashar[1], Chetali Bandodkar[1], Aarav Rajput[1], Nitya Mittal[2], Shivi Khimesara[2]

[1]*VIT Bhopal University, Sehore, Madhya Pradesh, India*
[2]*Medi-Caps University, Indore, Madhya Pradesh, India*

Due to the increasing frequency and complexity of attacks, Intrusion Detection Systems (IDS) must be increasingly intelligent to identify malicious activity across network settings. Traditional IDS methods use central data for training, raising concerns regarding its privacy and security. We propose a Federated Learning (FL) strategy that lets companies train machine learning models without sharing data to tackle these challenges. FL allows users to train models locally and communicate model updates instead of sensitive data, enabling decentralized learning and privacy. Fully Homomorphic Encryption (FHE) or Secure Multi-Party Computation (SMPC) enhances training confidentiality and security. These methods protect model changes during transmission and enable safe aggregation without decryption. This strategy protects privacy and lets companies improve their IDS models without releasing personal information. The recommended approach uses FL-trained anomaly detection technologies like neural networks and decision trees. All model updates are merged privately via secure aggregation. If data is captured during transmission, privacy-preserving methods like SMPC or FHE will protect model parameters and gradients. This framework enhances intrusion detection systems (IDS) in recognizing and mitigating cyber threats by enabling secure communication and providing a scalable solution. It also increases collaborative cyber security efforts across various entities. This system's Federated Learning and secure encryption technologies allow organizations to share knowledge and enhance defenses against sophisticated attacks without revealing sensitive data, solving cyber security challenges.

Keywords: Federated learning, Collaborative intrusion detection system.

1. Introduction

The development of sophisticated attacks and the many new dangers that conventional intrusion detection systems (IDS) face are making it difficult for these systems to keep up with the pace of the situation. Concerns about security and privacy occur as a result of the fact that intrusion detection systems (IDS) often rely on centralized data processing, despite the fact that their primary purpose is to monitor network traffic for aberrant or destructive activities. Some of the reasons why companies can be reluctant or unable to share sensitive data with other parties include the risk of data breaches, the need to comply with regulatory requirements, and the desire to gain a competitive edge. As a consequence of this, it is of the utmost importance to build an IDS cooperation architecture that enables coordinated efforts to detect complex threats while yet maintaining a respect for privacy.

Federated Learning for Collaborative IDS

Federated learning (FL) is a relatively recent technique to machine learning that enables several users to collaborate on the training of a single model without sharing any of the data that is used to train the model. This suggests that in the context of intrusion detection systems (IDS), a number of companies may collaborate in order to train a model by making use of decentralized data sources, hence enhancing the detection capabilities of their respective systems. During the process of training the model, every participant makes adjustments to the shared model parameters by using their own data; nevertheless, each participant does not reveal any personally identifying information. Through the use of FL, model updates may be integrated and refined, resulting in a more robust intrusion detection system (IDS) that is able to identify detailed patterns of assaults across several networks.

Even if FL enables more secure data storage on local nodes, this does not guarantee that model updates are protected from malicious actors like as hackers and other cybercriminals. Both Fully Homomorphic Encryption (FHE) and Secure Multi-Party Computation (SMPC) are examples of approaches that are absolutely important at this moment in time for protecting individuals' privacy.

Privacy-Preserving Techniques: SMPC and FHE

By incorporating SMPC or FHE into the FL process, it is possible to alleviate concerns about the confidentiality of data during the training of employees. SMPC allows for the computation of a function over private inputs in a collaborative manner while maintaining the confidentiality of those respective inputs. Within the context of FL for IDS, SMPC ensures that participants are able to compute model updates while discretely concealing important local data from other individuals.

The Fully Homomorphic Encryption (FHE) algorithm, on the other hand, enables computations to be carried out on encrypted data. This ensures that the original data is concealed even while the model is being updated. By using these privacy-preserving cryptographic techniques, the intrusion detection system (IDS) is able to make advantage of collaborative learning without compromising the confidentiality of data or model updates.

In this paper, we investigate the potential advantages that secure federated learning might bring to interactive Intrusion Detection Systems (IDS) that are used in collaboration. The combination of federated learning with either SMPC or FHE is something that we wish to do in order to enhance the system's capabilities for breach detection and data protection.

2. OBJECTIVES

1. To make it possible for businesses to work together efficiently without exchanging data.

2. To use SMPC and FHE to safeguard sensitive data and model update confidentially.

3. RESEARCH METHODOLOGY

For the purpose of determining how well FL performs when it is applied to the detection of network threats, we employ Intrusion Detection as our benchmark. CIC-IDS2017, which stands for Evaluation Dataset2, has been made accessible to the public by the Canadian Institute of Cyber Security. These data sets are available to the scientific community so that they may do research on data mining for the purpose of achieving cyber goals. The usual configuration that was used for the purpose of acquiring network frames .The victim network is comprised of ten workstations that are configured with a variety of operating systems and are connected to two routers. The router that serves as the external interface is the target of an attack that is launched by a network of computers belonging to an outsider. Following the recording of the attack frames by the capture server, the final dataset is eventually collected.

Machine Learning dataset

Raw dataset: The capture server is responsible for gathering the system logs and network traffic of each individual computer. These data are subsequently included into the raw dataset. These figures are taken over a period of five days, commencing at nine o'clock in the morning on Monday, July 3, 2017, and finishing at five o'clock in the afternoon on Friday, July 7, 2017. It is possible for a feature extractor, namely CIC Flow Meter, to convert the raw data into tabular data that has been tagged; this research makes use of a processed dataset that is openly available to the general public.

Processed dataset for Machine Learning

The final collection is comprised of two,544,042 data points; each data point is described by 83 separate parameters, in addition to a label that defines the kind of traffic that it corresponds to. A total of fifteen separate types of traffic are identified by the labels, which are as follows: Benign, Bot, Distributed Denial of Service, Denial of Service Attack, Heartbleed, Infiltration, PortScan, SSH-Patator, Web Attack—Brute Force, Web Attack—Sql Injection, and Web Attack—XSS. A detailed explanation of the classification process for the data points is included in Tab 1. In order to classify them, we divide them into seven distinct categories: benign, bot, brute force, distributed denial of service, heartbleed, infiltration, and port scan. Eleven factors are not required since they do not help to explain the general variance in the dataset. These qualities are as follows: There are a number of factors that need to be taken into consideration, including the Forward Average Bulk Rate, Forward Average Packets/Bulk, Forward Average Bytes/Bulk, CWE Flag Count, Forward URG Flags, Backward PSH Flags, and Forward Average Bulk Rate. At the conclusion of the process, we remove the features that are responsible for providing the traffic ID and instead make use of the remaining 65 features in a cleaned dataset (CDS) to feed the algorithm described in Section 4.

Table 1: Class sample count and percentage.

Attack Type	Count	Percentage
Benign	599,945	58%
Bot	1,944	0.19%
D DoS	128,014	12%
DoS Golden Eye	10,286	1.0%

DoS Hulk	172,846	17%
DoS Slow http test	5,228	0.58%
DoS slow loris	5,385	0.52%
FTP-Patator	5,931	0.58%
Heart bleed	11	0.0011%
Infiltration	36	0.0035%
Port Scan	90,694	8.8%
SSH-Patator	3,219	0.31%
Web Attack-Brute Force	1,470	0.14%
Web Attack-Sql Injection	21	0.002%
Web Attack-XSS	652	0.063%

4. DATA ANALYSIS

Training set

The detection of network assaults continues to be a challenging task, even when using a centralized dataset, since there is a large imbalance between the many classes under consideration. For instance, the dataset includes more than sixty percent of traffic points that are not malicious, while the majority of attack types account for less than one percent of the total. In most cases, we may be able to solve this problem by resampling the classes that are considered to be in the minority. This would rectify the imbalance in the distribution of the data points. Validation of a federated optimization approach is required when it comes to FL. This technique must be verified against new significant qualities that are introduced by training set distributions. To begin, there is a possibility that the dataset belonging to the customers is not balanced. Furthermore, in contrast to a centralized dataset, which is often independent and uniformly distributed (non-IID), a decentralized dataset frequently includes skewed data distributions for each client.

This is because a centralized dataset is centralized. It is important to assess the resilience of a federated training system by taking into account the attributes highlighted above. Because of this, we decided to build two distinct decentralized training sets, which we referred to as DS1 and DS2. The procedure for acquiring these sets is described in Tab. 2. In the dataset that is presented in Section 1 of the CDS, these sets are derived from the 32 machines that have the highest population density. The machine level configuration is responsible for determining the training/test split, which is set at 70%. In addition, we generate two possible centralized datasets, which we refer to as B1 and B2. These datasets have the potential to be used in a conventional method of centralized training. We establish two hypothetical benchmarks in this simulation to investigate how FL impacts validation performance. This is done despite the fact that it is not possible to centralize remote private datasets in applications that are used in the real world.

Validation metrics and test set

A detailed examination of the performance of the federated training is carried out by making use of the two metrics that are shown in Tab. In classic Deep Learning models, the total number of epochs is a popular statistic that is used to evaluate the pace at which the model is going to converge. On account of the fact that a large number of local models are being trained concurrently in FL, it could be challenging to ascertain a comparable quantity. In the next step, we will employ one weight update per training cycle for both the benchmarks and the collaborative optimization. This will be done in the centralized architecture (batch) and the central server (round), respectively. When it comes to algorithms, this is referred to as the "s" unit step. It should be brought to your attention that we do not take into account any possible delays that may be caused by communication in apps that are used in the real world. It is not feasible to take into consideration these implications within the context of this work since it is a controlled research study that is conducted inside a simulated environment. Additionally, since there is only one GPU available, each round of federated training updates the decentralized models in a sequential manner.

This results in an increase in the amount of time that the user is required to spend on each cycle. The practical implication of this is that secondary models may be trained in parallel by making use of the computational resources that are made accessible by each decentralized model simultaneously. Therefore, if we utilized the actual user-time as the time variable, our simulations would be more likely to penalize federated trainings in comparison to conventional ones, and they would also fail to take into account the unavoidable latencies that occur in real networks. As functions of the training unit s, the F1 score (Eq. 1) and the percentage of false negative detections (fn) are shown as performance measurements. Both of these metrics are given in terms of the training unit s. The precision p and recall r are the metrics that are used in the calculation of these metrics. These metrics are obtained by adding up the total number of true positive detections (ntp), false positive detections (nfp), and false negative detections (nfn) across all of the data points that were seen while performing step s.

$$\mathrm{F}_1 = 2\frac{p\,r}{p+r}, \quad p = \frac{n_{\mathrm{tp}}}{n_{\mathrm{tp}} + n_{\mathrm{fp}}}, \; r = \frac{n_{\mathrm{tp}}}{n_{\mathrm{tp}} + n_{\mathrm{fn}}}. \qquad (1)$$

We may be able to get the set that is utilized to assess the model (the central server in a federated training scenario) if we consolidate the test sets from the separate machines without resampling them. Both centralized and federated trainings are evaluated against the same dataset, and that evaluation is carried out using the same metrics at each level.

Federated Learning And Neural Network Architecture

Studies that have been conducted recently on the topic make it abundantly evident that FL has garnered a great deal of interest, both in terms of study and practical applications.

Table 2: How to get training sets for our study's experiments. The decentralized dataset applies actions to a randomly picked portion of data for each client. The train/test ratio is 70%. The non-training data are centralized and used as a test set for both systems.

Decentralized datasets

Id.	Statistical properties	Description
DS1	Non-IID Balanced	Destination IP in CDS is used to assign data points to decentralized users. The 32 most populated users are kept. Classes in clients are balanced via re-sampling. Final dataset size is unchanged.
DS2	Non-IID Non-balanced	Destination IP in CDS is used to assign data points to decentralized users. The 32 most populated users are kept.

Benchmarks

Id	Statistical properties	Description
B1	Balanced	32 most populated machines in CDS are centralized. Classes are balanced via re-sampling
B2	Non-balanced	32 most populated machines in CDS are centralized.

This is due to the fact that it provides a framework for improving machine learning algorithms, which may help circumvent significant data-related constraints such as privacy concerns and the cost of aggregating information across several servers. Numerous optimization techniques have been proposed for use with neural networks. These ideas have been put forth. We make use of the well-known Federated Averaging technique in our research since it is included in the FL high-level APIs that are made available by the Tensor Flow Federated package.

Federated Averaging algorithm

The Federated Averaging method, which was first given in, is summarized in method 1, which offers a concise summary of the technique. The weights and biases ρ of the core model are represented on the clients, which is the fundamental concept. Whenever client k is available, it obtains the weights and biases of the central model during a training round t. Additionally, it updates its model instance ρk locally by using the nk-size training dataset Pk that is accessible locally. It is the client optimizer's responsibility to evaluate data batches of size B for E training epochs. Following the conclusion of the process of updating the decentralized models, the weights {ωk}k=1,...,Sk are sent to the central server. The central server then employs a weighted average that takes into account the quantity of local samples in order to aggregate the models. Taking into consideration this paradigm, Tab. 6 provides a list of the hyper-parameters that were retained for our experiences. These hyper-parameters are divided down according to the collaborative and centralized trainings that served as benchmarks. When it comes to the Tensor Flow Federated high-level APIs, we are not aware of any method that allows for the hyper-parameters of the clients to be fine-tuned. All along the course of our inquiry, we saw a change in the core.

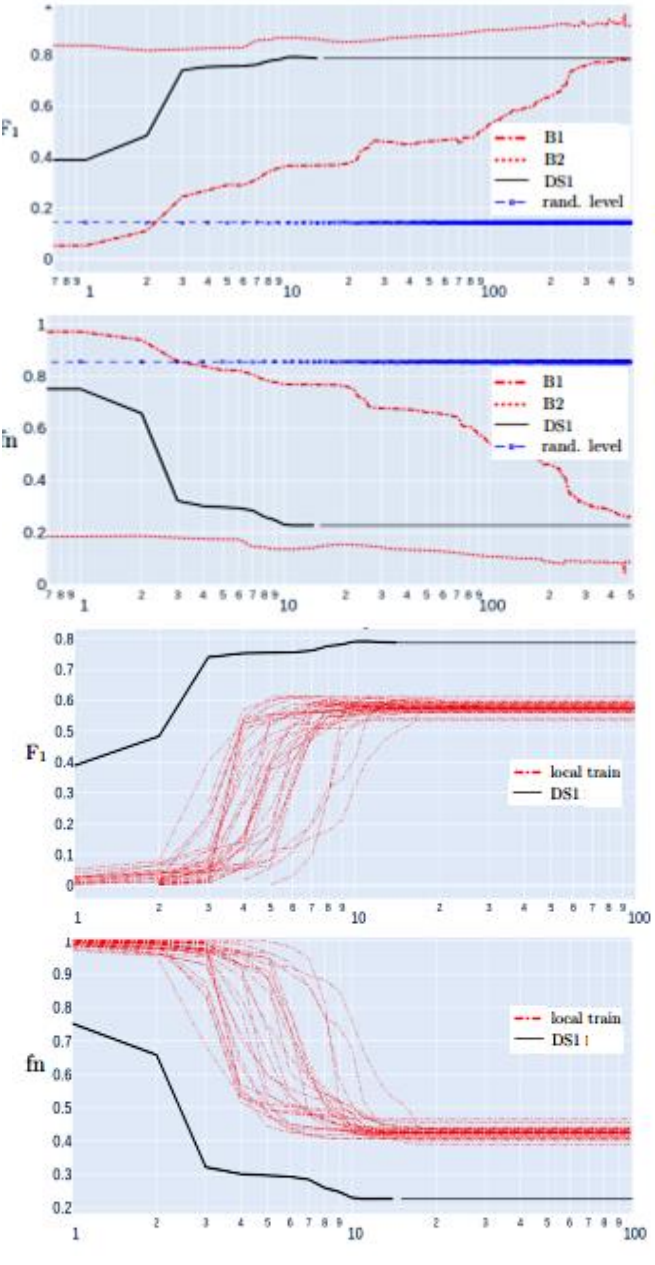

Fig. 1: Federated training performance on DS1. DS1's F1 score (black line) compared to centralized training for B1 (red dash-dotted line), B2 (red dotted line) and random prediction (dashed blue line with box shaped markers). The federated training was interrupted at the best performing point and extended for visualization; b) same analysis as in the first panel, but for the false negative detection ratio fn; c) F1 score obtained for DS1 (black line) against locally-trained machines in the 32 clients' databases retained for this study; d) same analysis as in the third panel, but for the

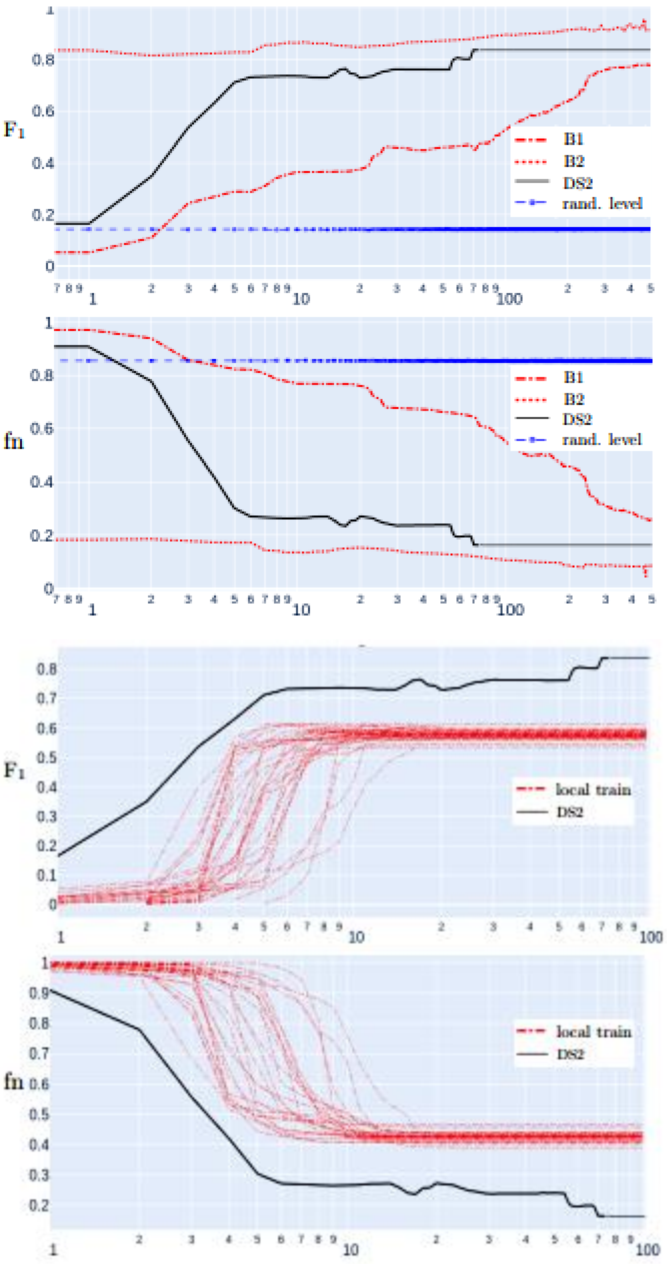

Fig. 2: DS2-based federated training performance analysis. From the top: a) DS2 F1 score (black line), centralized B1 and B2 training (red dash dotted lines) and random prediction (dashed blue line with box-shaped markers). The federated training was interrupted at the best performing point and extended for visualization; b) same analysis as in the first panel, but for the false negative detection ratio fn; c) F1 score obtained for DS2 (black line) against locally-trained machines in the 32 clients' databases retained for this study; d) same analysis as in the third panel.

Table 3: The proportion of clients accessible for server update determines FL's maximum performance across the two datasets. Minimum false-negative detection ratio defines maximum performance.

DS1

Frac. client	0.1	0.3	0.5	1.0
Min. fn	0.24	0.20	0.22	0.22
Round	4	4	3	15
F1	0.78	0.80	0.78	0.78

DS2

Frac. client	0.1	0.3	0.5	1.0
Min. fn	0.41	0.24	0.20	0.16
Round	24	72	12	80
F1	0.58	0.75	0.80	0.84

server's efficiency during the course of multiple training cycles. Through the use of early halting, we were able to circumvent this outcome. There is a great probability that this problem is brought about by consumers who over fit their own training set. This leads to the global model receiving significantly biased weights due to the fact that the data is not integrated into the data. Due to the fact that our focus is on the benefits of Federated Learning for network attack detection and the fact that this behavior does not have any impact on the results presented in Section 4, we do not go any further into this issue and instead leave it for future research.

5. RESULT

When it comes to apps that are used in the real world, not all users are available for contact throughout each cycle. In order to determine how effectively the optimization approach mentioned in Section 4 works, we make use of both the training set from Table 2 as well as a variable percentage of customers. The next step is to determine how reliable the results are by randomly picking fifty percent, thirty percent, and ten percent of the workstations that are available to connect with the primary server. Following the completion of an evaluation of a federated decentralized benchmark in which all 32 clients are always accessible, this step is carried out. In the event where each and every client is employed for server updates throughout each and every cycle, the results for the training set DS1 are summarized in the different panels of Figure 1. Table 2 presents the link between the algorithmic unit s and the F1 score and the false negative detection ratio for the two benchmarks and the federated training. This relationship is shown in descending order from the top to the bottom of the table.

Performing a random prediction on the attack classes is another component that is included in the system. In situations when there is no prior training set available, such as when a new client is being installed (also known as a cool start), this particular scenario may occur. It has been shown that the benchmark B2 performs better than the federated-trained neural networks that were terminated early in terms of the maximum number of epochs that may be used. It has been shown that the F1 score decreases by around 15% when a collaborative detection approach is used, while the false negative detection ratio increases by roughly 2.5 times. FL demonstrates virtually no performance degradation for the benchmark B1 when compared to the case in which the virtual centralization is considered. In addition to this, it delivers optimal performance with just a few minor adjustments to the model. If a new machine is brought online without any prior data (cold start), extracting the central model may ensure an F1 score that is 4.5 times higher and a false negative detection percentage that is about four times lower than a random detection in real-world applications. This is because the F1 score is an indicator of how well the machine is doing. If you would want to analyze the performance of the federated neural networks in comparison to a hypothetical conservative benchmark, please refer to Section 1 for the definitions of benchmarks B1 and B2. Flavor learning is the most effective method for machine learning in situations when data centralization is not feasible. Over the course of all 32 decentralized datasets, we evaluate the validation performance of both federated and locally trained models using the same architecture.

This evaluation is shown in the last two panels of Figure 1. There is a lot more significance to this exam. There have been no modifications made to the examination; it continues to contain all of the outcomes of population-based tests. Through the use of a consistent learning rate $\eta = 10{-}3$, the local models have been refined to perfection. When compared to the machine that performed the best, the data indicate that FL greatly enhances performance, which results in an increase of around 27% in the F1 score. On the contrary, there is a decrease of around 1.7 in the ratio of false negative detection. Figure 1 presents the findings of the same research project that was carried out on the DS2 dataset. Even though there have been some minor modifications made to the overall performance measures in comparison to the data shown in the previous paragraph, the significant conclusions have not been altered. A client who makes use of collaborative learning has the ability to increase their F1 score by about five times and lower the false negative detection ratio by approximately 5.3 times. This is in contrast to the situation in which the customer begins from scratch.

The federated training of a neural network outperforms even the most highly performing locally-trained machine by a margin of around 36% in terms of the F1 score and by a ratio of approximately 2.3 in terms of the fraction of false negative detection. As a last step in our analysis, we will determine whether or not the results are consistent when we gradually reduce the number of clients who are available at each round. Even when just a tiny number of customers take part in training rounds, as seen in Tab. 3, the performance of the dataset DS1 shows essentially no signs of deterioration. When just ten percent of the clients are used in each cycle, the performance of the dataset DS2 suffers as a consequence. Even in the most severe case detailed here, the federated trained neural network surpasses the highest performing locally-trained individual machine, proving that the benefits of collaborative training are still visible up to a fraction of clients of 0.3.

6. CONCLUSION

In conclusion, Secure Federated Learning for Collaborative Intrusion Detection Systems (IDS) is an optimistic step towards improved cyber security since it enables organizations to collaborate on the training of machine learning models without allowing their sensitive data to fall into the wrong hands. By using Federated Learning (FL) in combination with privacy-preserving technologies such as Secure Multi-Party Computation (SMPC) or Fully Homomorphic Encryption (FHE), it is possible to aggregate model updates while simultaneously ensuring that raw data is never taken out of the local context. This collaborative paradigm not only makes it easier to identify new threats across a wide range of networks, but it also makes it possible to react in real time to shifting attack strategies. The incorporation of these technologies that protect individuals' privacy results in an increase in the level of trust that exists between businesses. This, in turn, promotes a greater level of participation in systems that provide collective defense against cyber assaults. However, in order to fully exploit the potential of this strategy, further research will need to find solutions to difficulties such as regulating scalability, preserving computational efficiency, and dealing with any adversarial attacks. In the long term, this innovative design has the potential to pave the way for enhanced intrusion detection systems (IDS) that are capable of protecting vital infrastructure in a range of different environments.

References

1. E. Schiller, A. Aidoo, J. Fuhrer, J. Stahl, M. Ziorjen and B. Stiller, "Landscape of iot security", Computer Science Review, vol. 44, pp. 100467, 2022.

a. Imteaj, U. Thakker, S. Wang, J. Li and M. H. Amini, "A survey on federated learning for resource-constrained iot devices", IEEE Internet of Things Journal, vol. 9, no. 1, pp. 1-24, 2021.

2. Zakaria et al., "Cochain-sc: An intra- and inter-domain ddos mitigation scheme based on blockchain using sdn and smart contract", IEEE Access, vol. 7, pp. 98893-98907, 2019.

3. H. Moudoud, S. Cherkaoui and L. Khoukhi, "Towards a secure and reliable federated learning using blockchain", 2021 IEEE Global Communications Conference (GLOBECOM), pp. 01-06, 2021

4. Augenstein, S., McMahan, H. B., Ramage, D., Ramaswamy, S., Kairouz, P., Chen, M., Mathews, R., and Aguera y Arcas, B. Generative Models for Effective ML on Private, Decentralized Datasets. arXiv eprints (Nov. 2019), arXiv:1911.06679.

5. Brendan McMahan, H., Moore, E., Ramage, D., Hampson, S., and Aguera y Arcas, ¨ B. Communication-Efficient Learning of Deep Networks from Decentralized Data. arXiv e-prints (Feb. 2016), arXiv:1602.05629

6. Brisimi, T. S., Chen, R., Mela, T., Olshevsky, A., Paschalidis, I. C., and Shi, W. Federated learning of predictive models from federated Electronic Health Records International Journal of Medical Informatics 112 (2018), 59–67.

7. Dwork, C. A Firm Foundation for Private Data Analysis. Commun. ACM 54, 1 (jan 2011), 86–95.

8. Dwork, C., McSherry, F., Nissim, K., and Smith, A. Calibrating Noise to Sensitivity in Private Data Analysis. In Theory of Cryptography (01 2006), vol. Vol. 3876, pp. 265–284.

9. Konecnˇ y, J., Brendan McMahan, H., Ram- ´ age, D., and Richtarik, P. ´ Federated Optimization: Distributed Machine Learning for On-Device Intelligence. arXiv e-prints (Oct. 2016), arXiv:1610.02527.

10. Li, L., Fan, Y., Tse, M., and Lin, K.-Y. A review of applications in federated learning. Computers & Industrial Engineering 149 (2020), 106854.

11. Zakaria et al., "‘why should i trust your ids?’: An explainable deep learning framework for intrusion detection systems in internet of things networks", IEEE Open Journal of the Communications Society, vol. 3, pp. 1164-1176, 2022

12. Z. A. EI Houda, H. Moudoud, B. Brik and L. Khoukhi, "Securing federated learning through blockchain and explainable ai for robust intrusion detection in iot networks", IEEE INFOCOM 2023 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 1-6, 2023.
13. Zakaria et al., "A novel iot-based explainable deep learning framework for intrusion detection systems", IEEE Internet of Things Magazine, vol. 5, no. 2, pp. 20-23, 2022.
14. Zakaria et al., "When collaborative federated learning meets blockchain to preserve privacy in healthcare", IEEE Transactions on Network Science and Engineering, pp. 1-11, 2022.
15. H. Moudoud and S. Cherkaoui, "Toward secure and private federated learning for iot using blockchain", GLOBECOM 2022 – 2022 IEEE Global Communications Conference, pp. 4316-4321, 2022.
16. Zakaria et al., "A low-latency fog-based framework to secure iot applications using collaborative federated learning", 2022 IEEE 47th Conference on Local Computer Networks (LCN), pp. 343-346, 2022