IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 9 July 2024, accepted 23 July 2024, date of publication 30 July 2024, date of current version 9 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3435920

RESEARCH ARTICLE

FBMP-IDS: FL-Based Blockchain-Powered Lightweight MPC-Secured IDS for 6G Networks

SABRINA SAKRAOUI[1], AHMED AHMIM[2], MAKHLOUF DERDOUR[3], MARWA AHMIM[1], SARRA NAMANE[1], AND IMED BEN DHAOU[4,5,6], (Senior Member, IEEE)

[1]Networks and Systems Laboratory, Department of Computer Science, Badji Mokhtar–Annaba University, Annaba 23000, Algeria
[2]Department of Computer Science, Mohamed Cherif Messaadia University-Souk Ahras, Souk Ahras 41000, Algeria
[3]LIAOA Laboratory, University of Oum El Bouaghi, Oum El Bouaghi 4000, Algeria
[4]Department of Computer Science, Hekma School of Engineering, Computing and Design, Dar Al-Hekma University, Jeddah 22246, Saudi Arabia
[5]Department of Computing, University of Turku, 20014 Turku, Finland
[6]Department of Technology, Higher Institute of Computer Sciences and Mathematics, University of Monastir, Monastir 5000, Tunisia

Corresponding author: Ahmed Ahmim (a.ahmim@univ-soukahras.dz)

ABSTRACT The coming 6G wireless network is poised to achieve unprecedented data rates, latency, and integration with newer technologies like AI and IoE. On the other hand, along with this kind of growth in the AI domain and the large-scale connectivity in 6G. It is also going to raise many security concerns at the level of intrusion detection and prevention. For intrusion detection, centralized approaches won't be able to work effectively, therefore there is an utmost need to design decentralized and privacy-preserving solutions. In this work, we propose a novel secure gradients exchange algorithm for distributed intrusion detection in 6G networks. Our method is designed to take into account the use of Federated Learning with secure multi-party computation and blockchain technology. This way ensures that the collaborating parties are able to conduct the training of intrusion detection models in a secure and collaborative manner by retaining privacy in the data. Gradient compression and adaptive secure aggregation strategies are used to further optimize communication overhead and computational complexity. Therefore, our design works in a robust and efficient manner with the high data rates and huge connectivity that 6G networks will provide. To achieve our goal, experiments using the CICIoT2023 dataset were performed, and results showed that our federated learning-based hybrid model composed of CNN1D and a multi-head attention mechanism outperformed other well-known deep learning models in terms of performance. It achieved the highest average accuracy with 79.92%, the highest average detection rate with 77.41%, and a low false alarm rate with 2.55%.

INDEX TERMS 6G, computer security, network security, intrusion detection, IDS, blockchain, IoT, machine learning, deep learning, multi-head attention CNN, LSTM, hybrid model, anomaly detection, federated learning.

I. INTRODUCTION

Wireless communication has been one of the catalysts of societal progress from audible primitive vibrations to data transmission using radio waves. The wireless communication industry has seen rapid proliferation and significant inventions in the past few decades, starting with the seminal invention of the Advanced Mobile Phone System (AMPS), or 1G, by Bell Labs. The feature sets have improved considerably in subsequent generations, including 2G, 3G, 4G, and the latest 5G networks [1]. The 5G technology was supposed to bring a lot of advanced features, including the IoE and enhanced broadband to MTC. Realization of some of the hyped goals, such as wireless interconnectivity of machines without human intervention, seems far-fetched with the current 5G networks [2]. This realization makes it important to critically reflect on whether 5G would be able to satisfy the originally stated goals for conceived

The associate editor coordinating the review of this manuscript and approving it for publication was Bilal Khawaja.

applications like IoE and whether the upcoming generation of wireless network 6G shall demonstrate singular adaptability and efficiency to cope with the large number of sophisticated demands projected by 2030 [1].

Figure 1 summarizes an overview of connected intelligence for future 6G networks. When the next generation of wireless communication networks, the sixth-generation (6G) network, is compared to its predecessors, a quantum leap in communication is supposed to be experienced. These networks are claimed to offer faster data speeds, lower latency, and the ability to connect a much larger number of devices [1]. Looking toward the realization of these advances, 6G will probably have a network architecture divided into layers, each catering to different functionalities. The perception layer concerns transmitting signals and their modulation techniques, which could possibly be done through even higher frequency bands than the millimeter wave bands for even higher capacity [3]. The Edge layer encompasses intelligent devices capable of smart, efficient, and tailored routing and processing capabilities [4], such as cloud task offloading and edge caching. The core network includes more processing and storage capabilities tailored for more power-demanding tasks and operates as the brain of the entire network, that does all the high-lifting tasks, such as data mining and analytics. With such configurations, the application layer will support the deployment of futuristic applications, including real-time critical-mission tasks, such as intelligent transportation and smart healthcare services.

The integration of artificial intelligence into 6G network frameworks is heralded to have great potential in fostering innovation and facilitating real-time intelligent decision-making. However, this is also likely to introduce some challenges in terms of security risks and potential attacks. As AI becomes more deeply ingrained in network functionalities, it is increasingly necessary to address vulnerabilities and establish solid security measures to protect the integrity and privacy of these advanced systems [2]. An intrusion detection system is either software or hardware that analyzes the traffic in a network or host logs to detect any security policy violations [5]. Utilizing AI, especially machine learning and deep learning, is essential to enhance the capability of intrusion detection systems in the IoT environment [6]. The strategic deployment of IDS is a guarantee to ensure network and system security and integrity. IDSs, long recognized as stalwart guardians of network integrity, are faced with new challenges in the dynamic and complex environment of 6G—borne out of the upsurge of data rates, growth of IoT devices, and integration of AI [7]. Novel intrusion detection approaches and secure collaborative learning are required to meet these emerging challenges. In such a context, federated learning has emerged as a promising paradigm that allows collaborative machine learning to ensure data privacy [8]. Through model training on distributed data sources without having to share the data, FL bypasses most of the privacy concerns associated with the traditional, centralized machine learning approach. However, the integration of FL in 6G networks introduces new security challenges, such as unprotected exchange of model gradients during the training phase.

This paper proposes a new secure gradients exchange-based IDS that uses the synergy of FL, secure MPC, and blockchain technology, specially designed for 6G wireless networks; the algorithm provides the enabling of decentralized, secure gradient aggregation; embedding techniques of gradient compression and quantization; and adaptive secure aggregation strategies for the optimization of communication overhead and computation complexity. On top of that, the algorithm is designed to be integrated with 6G network slicing and virtualization, ensuring that efficient resource allocation and Quality of Service (QoS) are guaranteed for the process of secure gradient exchange. The architecture proposed for the FBMP-IDS leverages a hybrid deep learning model that adopts the powerful feature extractors of CNNs combined with multi-head attention to detect intrusions in the 6G networks. This approach offers several benefits including long-range dependencies and contextual analysis. Multi-head attention component helps in uncovering long-range dependencies inherent in the data [9]. This has the effect of making the model scrutinize not just the data points but also their relationships in a way that understanding the network behavior is taken to a holistic level. In light of such contextual information, the model will easily contrast normal network traffic patterns from probable intrusions.

The rest of this paper is organized as follows. Section II reviews existing works on intrusion detection systems (IDS) and especially points out the relevant state-of-the-art in deep learning and federated learning techniques applied to 6G network security. Section III explains the FBMP-IDS architecture and focuses on its component design and functionalities. Section IV deals with the computational complexity analysis of the proposed architecture FBMP-IDS with respect to communication complexity, compression overhead, and communication overhead of federated learning. Section V explains the experimental setup, including the intrusion detection dataset, evaluation metrics, used models, and implementation details, and presents the results of the evaluations in terms of key performance metrics such as TPRs for different types of intrusions and AUC-ROC scores. Analyze the performance comparison of our hybrid model compared to the individual deep learning models inside the FBMP-IDS framework. Section VI concludes this paper.

II. RELATED WORKS

The quest to harden the security infrastructure of 6G networks has seen many remarkable studies, which discuss novel methods of intrusion detection and prevention. Each one of them targets a different type of network and aims at different problems. This section details the most important ones on the topic of 6G intrusion detection, looking at methodologies applied to vehicular networks, IoT settings, and more general network scenarios, their key strengths and limitations, and

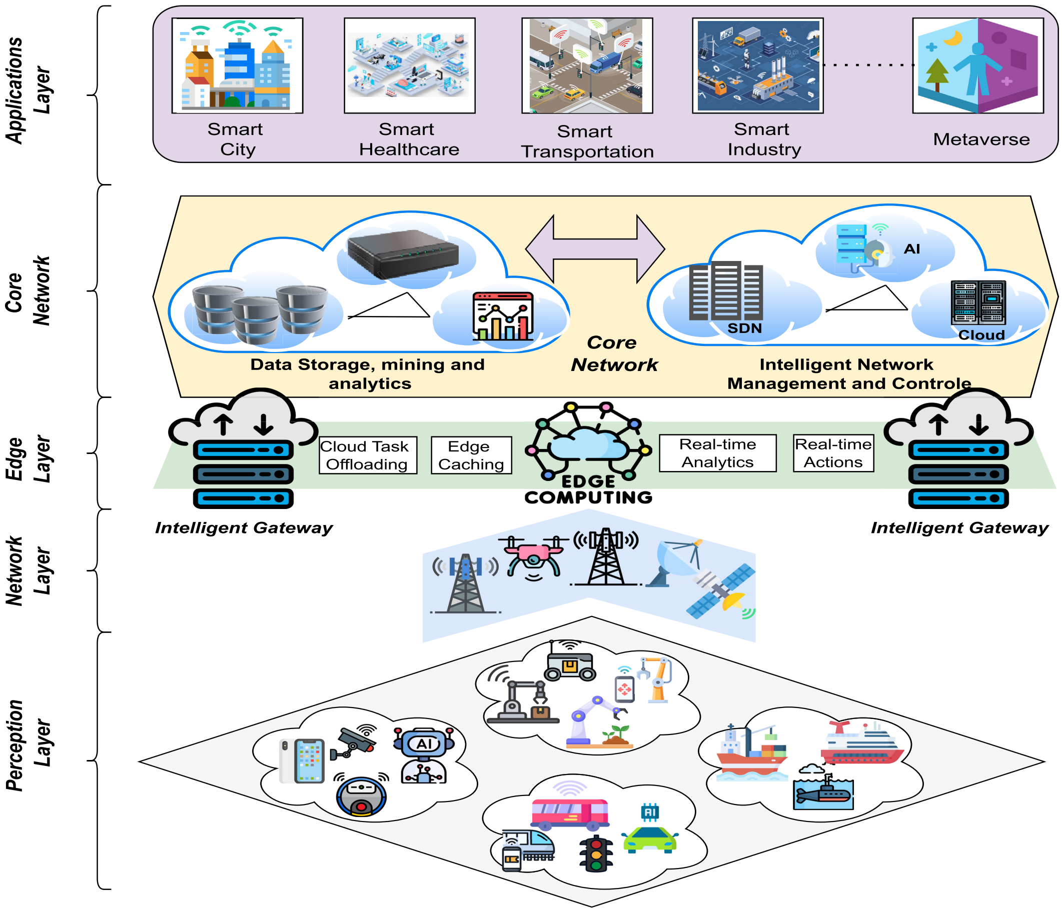


FIGURE 1. Vizualization of connected intelligence for future 6G Networks.

what exactly they bring new to the ever-changing landscape of network security.

For the purpose of detecting intrusion or attacks in 5G and IoT networks, a deep auto-encoded dense neural network technique has been developed by Rezvy et al. [10]. The benchmark Aegean Wi-Fi Intrusion dataset served as the basis for the authors' method evaluation. The system only detects three sorts of attacks, which are Flooding (or DoS), Injection, and impersonation type of attacks, making it infeasible for additional types and unable to protect against them. Nevertheless, the findings demonstrated excellent performance with an overall detection accuracy of 99.9%. Moudoud et al. [12] presented a detection and prediction stochastic Markov-based model to prevent false data injection and DDoS attacks and secure the 5G-enabled IoT. The proposed multi-layer IDPS system by Abdulqadder et al. [14] secures entities such as switches, domain controllers/smart controllers, NFV infrastructure, and other networking and data acquisition devices. In addition, many levels of security are employed. Zhang et al. [15] proposed a novel ensemble machine-learning algorithm based on weight techniques. Their purpose was to handle the challenges of dynamic and heterogeneous vehicular networks and meet the requirements of the sixth-generation network, particularly in terms of highly reliable and robust security measures. The experimental evaluations are conducted by taking into account 5 different CAN bus ID data packets and show good results with an increase in AUC of 13% and 15% in F-measure. However, the proposed system is only considered for centralized deployments.

Almiani et al. [17] proposed the use of the Kalman-backpropagation neural network in constructing an IDS model for detecting DDoS attacks in 5G-enabled IoT networks. The system was benchmarked using the CICDDoS2019 dataset and achieved an average detection accuracy of 94%. However, only one attack was considered. Chen et al. [19] recommended using fuzzing to evaluate NIDS's (Network Intrusion Detection Systems) rules. When compared to the speed of anomaly-based IDSs, the processing time variability is deemed inefficient.

Alotaibi et al. [20] proposed a software-driven FL-based IDS, integrated into the network architecture and making use of the advantages that 6G technologies have to offer. Within IDSoft, they created the hierarchical FL framework for intrusion detection. This framework features both synchronous and asynchronous aggregation techniques, as well

TABLE 1. Comparison of related works.

Study	Year	Approach	Used dataset	Results	Strengths	Limitations
Rezvy et al. [10]	2019	Deep auto-encoded dense neural network for intrusion detection in 5G and IoT networks	Aegean Wi-Fi Intrusion dataset [11]	Accuracy (99.9%)	High detection rate	Detects only 3 types of attacks
Moudoud et al. [12]	2020	Stochastic Markov-based model for detecting and predicting false data injection and DDoS attacks in 5G-enabled IoT	Real log activity from [13]	Detection rate >95% for DDoS	Lightweight; No complex computing	Considers only two types of attacks
Abdulqadder et al. [14]	2020	Multi-layer IDPS system for 6G networks with game theory and Four-Q-Curve technique	Own Dataset	High Detection Rate (96.08%)	Detects and mitigates various attacks	Centralized learning; Privacy issues
Zhang et al. [15]	2021	Ensemble machine learning algorithm for intrusion detection in 6G vehicular CAN bus networks	CAN intrusion dataset [16]	AUC(77.83%), Fmeasure (75.33%)	Addresses dynamic and heterogeneous vehicular networks; Improves AUC and F-measure	Centralized deployment; - High False Positive Rate (FPR) (10.46%)
Almiani et al. [17]	2021	Kalman backpropagation neural network IDS model for detecting DDoS attacks in 5G-enabled IoT	CICDDoS 2019 [18]	Acc. (94%), FPR (0.09%), DR (97.49%)	Good detection performances	Considers only one type of attack (DDoS)
Chen et al. [19]	2022	Fuzzing-based evaluation of Network Intrusion Detection Systems (NIDS) rules	Own Dataset	Analyzing phase time cost from 1ms to 1min	Evaluates effectiveness of rule implementations	Inefficient processing time variability; Signature-based detection shortcomings
Alotaibi et al. [20]	2023	Software-driven FL-based IDS (IDSoft) with hierarchical FL framework	MNIST [21]	Accuracy (94.17%)	Reduces communication rounds; Leverages 6G technologies	Lack of security against malicious nodes; -Dataset may not be suitable for the task
Vinita et al. [22]	2023	FL-based IDS for 6G Internet of Vehicles (IoV) with three-tier model weight aggregation	Sybil attack dataset [23]	Accuracy (87%)	High detection rate; Sybil attack detection	Does not consider malicious clients
Prasad et al. [24]	2023	Fuzzy Logic System (FLS) and ML-based IDS for securing mobile ad-hoc networks	Own Dataset	BRS: TPR (100%) and Accuracy (99.7%) on Wormhole	Evaluates performance reliability; Potential for network security solutions	Centralized learning; Privacy concerns; Only two attacks (Blackhole and Wormhole)

as a further offloading mechanism, for an overall performance increase of the system. The MNIST dataset was used [21]. When using three clusters, the number of communication rounds was reduced by 30%, and when using four clusters, the number of communication rounds was reduced by 60%. However, the system's security and the jeopardizing of the entire system in the presence of malicious nodes were not considered.

Vinita et al. [22] presented an FL-based IDS designed for the IoV, compatible with 6G networks. The improvement of security within cars, where training and detection are made, was their main priority. Their research showed that using a small number of global aggregations might hit a high detection accuracy rate of 87%. Their system also encompasses a Sybil attack detection where the data instances are differentiated between normal and Sybil attacks. However, it doesn't take into consideration that normal clients can be targeted and rendered malicious nodes. Prasad et al. [24] proposed an approach using the Fuzzy Logic System for performance reliability evaluation and introduced an ML-based IDS to effectively secure MANETs. The experiments are run in a virtual network environment with benign and hostile nodes to replicate black-hole and wormhole attacks.

Recent research investigated new paradigms, such as over-the-air computation technology [25], which efficiently encodes and decodes information using superimposed wave-forms to mitigate issues such as interference and data integrity. Other lines of research highlighted the problems that arise while dealing with the management and exchange of information within blockchain-based intrusion detection systems secured by Multi-Party Computation (MPC) in real-world scenarios. On the other hand, it has been shown that Software-Defined Networking (SDN) can effectively manage network resources. Meanwhile, SDN has proven effective in the efficient allocation of network resources. Integration of these technologies is increasingly recognized as a pivotal area of investigation to overcome practical hurdles [26].

In Table 1, we summarize the related approaches adopted for each study, strengths, and limitations. Clearly, most of the current approaches are based on deep learning and machine learning techniques that provide high detection rates but incur some major limitations. The current approaches face various limitations. 1. *Limited attack scope*: several works target a particular set of attacks (DDoS, FDI) or specific network environments (Vehicle CAN bus). 2. *Centralized learning limitation*: the centralized learning architecture is often criticized because it raises privacy concerns and might not be scalable for large-scale 6G networks. 3. *High False Positive Rates:* approaches that produce high false positives will lead to unnecessary disruptions in the network. 4. *Limited dataset representation:* datasets used in some works might not characterize the complex attack landscape that could appear in 6G networks.

The major contributions of the proposed system are:

1) DECENTRALIZED AND SECURE GRADIENT AGGREGATION

The system removes the requirement for a centralized aggregation server by distributing the gradient aggregation process across a blockchain network and client devices. Gradients are aggregated securely by using MPC protocols, ensuring that individual gradients remain private while correctly computing global model updates.

2) GRADIENT COMPRESSION

This system, to meet the challenges of high data rates and massive connectivity issues in 6G networks, uses gradient compression and quantization. In this way, we reduce the communication overhead associated with the transmission of gradients, thereby improving efficiency without significantly compromising model accuracy.

3) ADAPTIVE SECURE AGGREGATION STRATEGIES

The system has a set of MPC protocols with varying complexity, communication overhead, and security guarantees. Given the real-time network conditions and participant characteristics, the system will dynamically choose the most appropriate MPC protocol in each round of FL to ensure secure aggregation, optimally trading off between security, communication overhead, and computational complexity.

4) BLOCKCHAIN-ENABLED DECENTRALIZATION AND TRANSPARENCY

This system includes blockchain technology to make the FL process decentralized, transparent, and immutable. The blockchain network collectively manages the global state of the model, verifies the gradients' authenticity and integrity, and broadcasts the updated model weights to participating clients.

5) HYBRID MODEL BASED ON CNN1D AND MULTI-HEAD ATTENTION

With the aim of reaching the best performance regarding the detection rate and false alarm rate, we devised a hybrid model that combines lightweight design, high performance, and efficient training. For this purpose, we incorporate both CNNs with multi-head attention in the architecture of the FBMP-IDS, which are very promising techniques for the conception of performing intrusion detection on the complex 6G network environment. Such techniques are designed to guarantee better feature extraction, better contextual analysis of network traffic data, and possibly even superior learning efficiency as compared to models based on a single learning paradigm.

III. THE FBMP-IDS: AN OVERVIEW

In this paper, we have proposed a secure gradients exchange-based IDS for FL in a 6G wireless network environment, as provided in Alg. 1. The algorithm uses the synergy of FL, secure MPC, and blockchain to enable privacy-preserving and secure collaborative model training in a decentralized manner.

A. MOTIVATION

The motivation for the present work, of course, lies in the foreseen challenges and requirements of 6G wireless networks, such as massive connectivity, stringent latency and reliability constraints, and efficient resource utilization and network slicing issues [1], [2], and not to forget the issues related to privacy and security in the backdrop of collaborative machine learning and data sharing [7]. Combining FL, MPC, and blockchain technology into our proposed system, we are able to respond to these challenges in a holistic and innovative manner. It allows privacy-preserving and secure collaborative model training while harnessing the unique characteristics of the 6G networks, such as network slicing and virtualization, high data rates, and the potential integration of blockchain-enabled infrastructure. The resilience is better enhanced through the inherent distributed nature of the algorithm hence avoiding single points of failure. The adaptive secure aggregation strategies and gradient compression techniques further optimize the trade-offs among security, communication overhead, and computational complexity [14]. Overall, the present work is a contribution in view of realizing privacy-preserving and secure FL-based IDS for next-generation wireless networks to realize collaborative and distributed machine learning for a wide range of applications and services in the 6G era.

B. THREAT MODEL

To model the network for distributed intrusion detection in 6G networks, a number of potential threats and security vulnerabilities would have to be taken seriously that could compromise the system's integrity and effectiveness. With this, it would help to develop a more comprehensive threat model that appreciates the adversarial landscape with regard to the designing of effective countermeasures.

1) MALICIOUS INTRUSION DETECTION AGENTS

Intrusion Detection Agents are responsible for providing local network traffic data and computing resources to the collaborative training process. On the other hand, some of these agents might be compromised or controlled by adversaries, which will cause the following threats:

- Poisoning attacks: Malicious agents could inject corrupted or manipulated data into the training process to degrade the performance of the intrusion detection model [7].
- Model extraction attacks: Adversaries could attempt to extract or reconstruct the global intrusion detection model by exploiting the gradients or model updates exchanged during the training process [27].
- Privacy violations: Malicious agents could try to infer sensitive information about other participants' network traffic data from the exchanged gradients or model updates [28].

2) INSECURE COMMUNICATION CHANNELS

The secure exchange of gradients and model updates relies on communication channels between the Intrusion Detection Agents, Security Edge Nodes, and the Blockchain Network. Some of the potential threats include:

- Eavesdropping attacks: Adversaries could intercept and monitor the communication channels, making unauthorized access to sensitive information, for example, gradients or model updates [27].
- Man-in-the-middle attacks: Adversaries can intercept and tamper with the data being exchanged, possibly being injected with malicious payloads or corrupting the gradients or model updates [29].

C. NETWORK MODEL

Network architectures have to be designed for the distributed intrusion detection problem arising from integrating FL and AI in 6G wireless networks. This architecture corresponds to our new architecture featuring a network architecture tuned for distributed intrusion detection and designed in a way to take advantage of network slicing, virtualization, and edge computing for effective resource allocation, low latency, and high security for collaborative training of intrusion detection models. In this network model, we consider the 6G network infrastructure that offers network slicing and thereby makes possible the creation of multiple logical network instances dedicated to specific security services or applications [30]. Each network slice can further be divided into multiple sub-slices that can be dynamically allocated and orchestrated based on the needs of the distributed intrusion detection system [30]. The network consists of three main parts: the Intrusion Detection Agents, the Security Edge Nodes, and the Blockchain Network, as can be seen in Figure 2.

1) INTRUSION DETECTION AGENTS

The contributing devices or nodes are called Intrusion Detection Agents in this case, and they contribute their local network traffic data and computing resources to a collaborative training of the intrusion detection models. Therefore, the agents can represent any IoT device, network router, firewall, or any other kind of connected device that can capture network traffic and undertake local model training and computing of gradients [27]. Each agent will be assigned a specific sub-slice in the slice of the network reserved for the application of intrusion detection. This sub-slice will ensure isolated and secure communication channels for gradient exchange and resource allocation, while it provides QoS guarantees.

2) SECURITY EDGE NODES

The Security Edge Nodes are distributed computing resources placed on the network edge and closest to the Intrusion Detection Agents. These nodes become secure transmission points for the gradients in the collaborative training process, using the basic tenets of edge computing to reduce the overhead of latency and communication [27]. Security Edge Nodes are tasked with coordinating the secure gradients exchange algorithm, thereby helping the secure aggregation of gradients from the agents in their respective sub-slices. They also handle the integration with the Blockchain Network, broadcasting the aggregated gradients for verification and global model updates [28].

3) BLOCKCHAIN NETWORK

The Blockchain Network is the decentralized and immutable platform for secure gradient aggregation and management of the global intrusion detection model. Each node in the distributed network maintains and validates the blockchain ledger, ensuring transparency, traceability, and resilience against single points of failure [31]. Within the Blockchain Network, smart contracts are employed to encode the secure gradients exchange algorithm, enabling the verification of gradients, the computation of global model updates, and the distribution of the updated global intrusion detection model to the participating agents [31]. The Blockchain Network is integrated with the Security Edge Nodes and Intrusion Detection Agents to ensure a secure and decentralized collaborative training process. It makes use of blockchain technology to provide this functionality while keeping the integrity and privacy of the participating entities' network traffic data [27].

With network slicing, virtualization, edge computing, and blockchain techniques, our proposed network model provides a secure and strong frame for the deployment of distributed intrusion detection in 6G wireless networks. It will meet the challenges of resource allocation, latency, and security in a

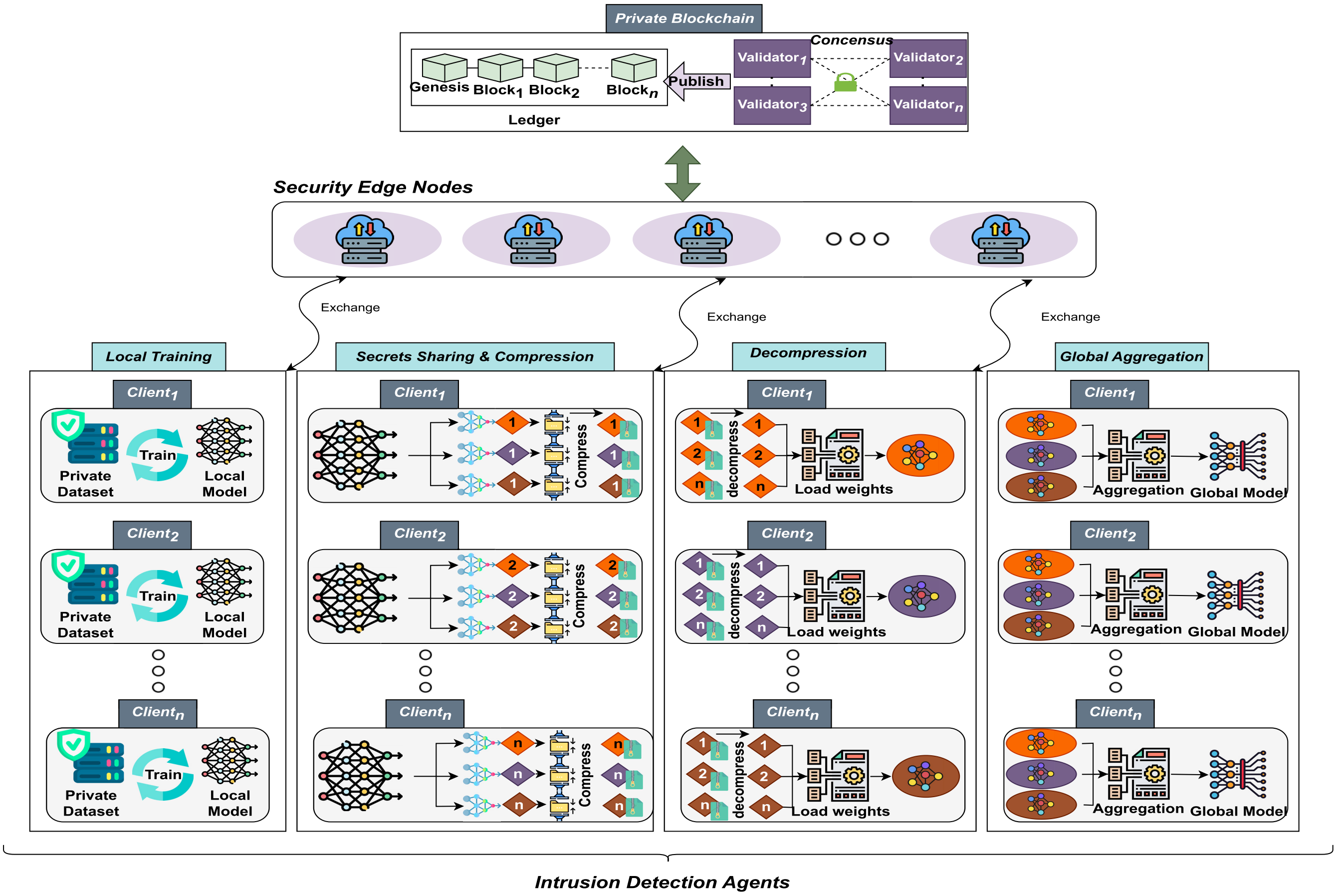


FIGURE 2. Network model for the FBMP-IDS system.

framework that allows for efficient and privacy-preserving collaborative training of intrusion detection models for the enhanced security of 6G networks.

D. SYSTEM MODEL

1) INITIALIZATION

- **a.** The global model M_G is initialized with random weights.
- **b.** Participants $P_1, P_2, \ldots, P_n$ engage in an MPC setup phase, establishing secure communication channels and generating shared secrets or cryptography keys necessary for MPC computations.
- **c.** A set of MPC protocols $\mathcal{P}$ with varying levels of complexity, communication overhead, and security guarantees are defined.

2) ROUND PREPARATION

For each round r of FL, an MPC protocol $\mathcal{P}_r \in \mathcal{P}$ is selected based on the current network conditions and participant characteristics.

3) CLIENT TRAINING

- **a.** Each participant P_i initializes a local model M_i with the same architecture as the global model M_G.
- **b.** P_i copies the current weights of M_G to M_i.
- **c.** P_i trains M_i on their local data using a suitable optimization algorithm and loss function.
- **d.** P_i computes the gradients ∇_i of M_i with respect to the initial weights copied from M_G.
- **e.** P_i compresses the gradients ∇_i using the compression function $Q(\cdot)$ to obtain compressed gradients $Q(\nabla_i)$.
- **f.** P_i secret-shares the compressed gradients $Q(\nabla_i)$ using the selected MPC protocol $\mathcal{P}_r$, obtaining shares $[Q(\nabla_i)]_1, [Q(\nabla_i)]_2, \ldots, [Q(\nabla_i)]_n$.

4) SECURE AGGREGATION USING MPC

- **a.** Participants execute the secure aggregation protocol defined by $\mathcal{P}_r$, where each P_i holds shares $[Q(\nabla_j)]_i$ of every other participant's compressed gradients $Q(\nabla_j)$.
- **b.** Through the secure aggregation protocol, participants jointly compute the sum of their shared compressed gradients without revealing individual values, yielding shares $[\overline{Q(\nabla)}]_i = \frac{1}{n}\sum_{j=1}^{n}[Q(\nabla_j)]_i$ of the average compressed gradients for each P_i.

5) RECONSTRUCTION AND BLOCKCHAIN INTEGRATION

- **a.** Participants reconstruct the final average compressed gradients $\overline{Q(\nabla)}$ from the shares $[\overline{Q(\nabla)}]_1, [\overline{Q(\nabla)}]_2, \ldots,$

Algorithm 1 FL-Based Blockchain-Powered Lightweight MPC-Secured IDS for 6G Networks

```
Require: Participants $P_1, P_2, \ldots, P_n$, Blockchain $BC$,
    Global Model $M_G$, Compression Function $Q(\cdot)$, Set of
    MPC Protocols $\mathcal{P}$
Initialize $M_G$ with random weights
Perform MPC setup: establish secure channels and
    generate shared secrets
for each round $r$ do
    Select MPC protocol $\mathcal{P}_r \in \mathcal{P}$ based on network
        conditions and participant characteristics
    for each participant $P_i$ do
        Initialize local model $M_i$ with same architecture
            as $M_G$
        Copy weights of $M_G$ to $M_i$
        Train $M_i$ on local data of $P_i$ using optimization
            algorithm and loss function
        Compute gradients $\nabla_i$ of $M_i$ with respect to initial
            weights from $M_G$
        Compress gradients: $Q(\nabla_i)$
        Secret-share $Q(\nabla_i)$ using $\mathcal{P}_r$ to obtain shares
            $[Q(\nabla_i)]_1, [Q(\nabla_i)]_2, \ldots, [Q(\nabla_i)]_n$
    end for
    Execute secure aggregation protocol in $\mathcal{P}_r$ to
        compute $[\overline{Q(\nabla)}]_i = \frac{1}{n}\sum_{j=1}^{n}[Q(\nabla_j)]_i$ for each $P_i$
    Reconstruct average compressed gradients $\overline{Q(\nabla)}$
        from shares $[\overline{Q(\nabla)}]_1, [\overline{Q(\nabla)}]_2, \ldots, [\overline{Q(\nabla)}]_n$
    Broadcast $\overline{Q(\nabla)}$ to $BC$ and verify using participants'
        signatures
    Update $M_G$ weights using $\overline{Q(\nabla)}$: $W_{M_G}^{(t+1)} = W_{M_G}^{(t)} -$
        $\eta\overline{Q(\nabla)}$
end for
```

$[\overline{Q(\nabla)}]_n$ using the MPC reconstruction protocol defined by $\mathcal{P}_r$.

- **b.** $\overline{Q(\nabla)}$ is broadcasted to the blockchain network BC.
- **c.** The authenticity and integrity of $\overline{Q(\nabla)}$ are verified using the participants' digital signatures.

6) GLOBAL MODEL UPDATE

- **a.** Once the average compressed gradients $\overline{Q(\nabla)}$ are verified on the blockchain, the global model M_G weights are updated using the verified gradients:

$$W_{M_G}^{(t+1)} = W_{M_G}^{(t)} - \eta\overline{Q(\nabla)}$$

where η is the learning rate, and t represents the current round of FL.

E. GRADIENT COMPRESSION APPROACH

DNN quantization has emerged as a pivotal technique in optimizing the deployment of neural networks on resource-constrained devices [32]. DNN quantization is defined as the process of reducing the precision in weights and activations of neural networks from higher bit-widths (e.g., 32-bit floating point) to lower bit-widths (e.g., 8-bit integers) [33], [34]. This reduction is strongly driven by the quest for improvements in computational efficiency and memory footprint, with an additional inference speed boost and no large losses in terms of model accuracy [35]. In our framework we implemented this stage using the QKeras quantization library from Google [35], [36], [37]. The library is built upon the work of [38], in creating Quantized Neural Networks (QNNs), where during training, all activations and weights are quantized to Q bits in a fixed point representation. The quantization function in the forward pass can be formulated by [38]:

$$q = clip(\frac{round(2^{Q-1} \times W)}{2^{Q-1}}, -1, 1 - 2^{-(Q-1)}) \tag{1}$$

The quantization function is a mathematical transformation applied to the weights of the neural network during the training process. The equation provided describes how a weight W is quantized to Q bits using a fixed-point representation.

- **Rounding and Scaling**: The expression $round(2^{Q-1} \times W)$ scales the weight W by 2^{Q-1} and then rounds it to the nearest integer.
- **Normalization**: The result is then divided by 2^{Q-1} to normalize it back to the range of the original weight values.
- **Clipping**: The *clip* function ensures that the quantized value q stays within the range $[-1, 1 - 2^{-(Q-1)}]$. This prevents the values from exceeding the re-presentable range for the given bit-width Q.

By quantizing the weights and activations to lower bit-widths, significant improvements in computational efficiency and memory usage are achieved, making it feasible to deploy complex neural networks on resource-constrained devices without substantial loss in accuracy. This has been proven by several works including [33], [37].

F. DYNAMIC SELECTION AND EXECUTION OF MPC PROTOCOLS

The secure aggregation protocol within our system is crucial for ensuring that the gradients from participants are aggregated securely without revealing individual contributions. The dynamic selection of MPC protocols is based on current network conditions and participant characteristics, as presented in Alg. 2.

The scoring function $S(\mathcal{P}_r, NC, PC)$ can be defined based on various factors such as communication overhead, computational complexity, and security guarantees. The scoring function can be defined as:

$$\begin{aligned} S(\mathcal{P}_r, NC, PC) = \alpha \cdot \mathrm{s}(\mathcal{P}_r) - \beta \cdot \mathrm{o}(\mathcal{P}_r, NC) \\ - \gamma \cdot \mathrm{c}(\mathcal{P}_r, PC) \end{aligned} \tag{2}$$

where s is the security guarantees, o is the communication overhead, c is the computational complexity, and α, β, γ are weights assigned to each factor based on their importance.

Algorithm 2 Dynamic Selection and Execution of MPC Protocols

```
Require: Set of MPC Protocols $\mathcal{P}$, Network Conditions $NC$,
    Participant Characteristics $PC$
function SelectMPCProtocol($\mathcal{P}, NC, PC$)
    Define a scoring function $S(\mathcal{P}_r, NC, PC)$ that
        evaluates the suitability of each protocol $\mathcal{P}_r \in$
        $\mathcal{P}$ based on current network conditions $NC$ and
        participant characteristics $PC$
    Initialize an empty list scores
    for each protocol $\mathcal{P}_r \in \mathcal{P}$ do
        Compute the score $s_r = S(\mathcal{P}_r, NC, PC)$
        Append $(\mathcal{P}_r, s_r)$ to scores
    end for
    Sort scores based on $s_r$ in descending order
    Select the protocol with the highest score: $\mathcal{P}_{best} =$
        scores[0], [0]
    return $\mathcal{P}_{best}$
end function
```

G. ADVANTAGES

By incorporating secure multi-party computation techniques, this system achieves an additional layer of security and privacy protection for the gradients exchange process [39]. Individual gradients remain private throughout the computation, and the aggregated gradients are computed correctly, even in the presence of compromised or malicious participants, thanks to the properties of MPC protocols [40]. Moreover, the system uses gradient compression to reduce communication overhead and adaptive secure aggregation strategies that dynamically choose the most appropriate MPC protocol according to the network conditions and the characteristics of the participants. This makes the algorithm more suited to the challenges and requirements of the 6G wireless network while preserving security and privacy.

H. OUR SYSTEM VS. TRADITIONAL FL-BASED IDSS

The system is a decentralized secure gradients exchange algorithm tailored for 6G networks, there is no centralized aggregation server [8]. The role traditionally played by a central server is distributed across the blockchain network and the participating clients [31]. The key aspects of the server's functionality are handled as follows:

1) GLOBAL MODEL INITIALIZATION AND UPDATE

A global model M_G is randomly initialized; its weights are further updated with the aggregated gradients from the set of clients participating in a given round. Yet, all these activities are performed not by a central server but are collectively managed by the blockchain network through consensus mechanisms and smart contracts.

2) GRADIENT AGGREGATION

The process of aggregating gradients from participating clients is performed in a decentralized manner by the blockchain network and the secure multi-party computation (MPC) protocols. Each client secret-shares their compressed gradients using the selected MPC protocol [40], and the participants jointly compute the average compressed gradients without revealing their individual values. This secure aggregation is facilitated by the MPC protocols and the blockchain network's consensus mechanisms.

3) GLOBAL MODEL DISTRIBUTION

After updating the global model weights, the new global model state needs to be distributed to the participating clients for the next round of training. In this approach, the updated global model weights can be broadcasted as transactions on the blockchain, which achieves transparency and immutability [31]. The updated model can be stored in a distributed file system for access by all participants [27]. Without a centralized aggregation server, the proposed algorithm may avoid potential single points of failure and enhance the security, resilience, and decentralization of the FL process. The blockchain network provides a decentralized and transparent basis, enjoying the properties of decentralization, immutability, and consensus mechanisms, for secure gradient aggregation and global model management.

IV. COMPUTATIONAL COMPLEXITY ANALYSIS

The computational complexity of the algorithm proposed can be decomposed into communication overhead, compression overhead, and the number of training rounds.

A. COMMUNICATION COMPLEXITY

The choice of MPC protocol in each round ($\mathcal{P}_r$) is highly influential in the choice of the time complexity. Popular MPC protocols typically have communication complexity scaling with $\mathcal{O}(n^d.K)$, where n is the number of participants and d and K are variable factors depending on the exact protocol used. [41], [42].

B. COMPRESSION OVERHEAD

Employment of the compression function $Q(\cdot)$ comes with some computational overhead compared to uncompressed gradients. However, the compression ratio attained will directly impact communication overhead. An efficient compression function with a higher ratio can hugely reduce the amount of data transmitted during secure aggregation. This, therefore, calls for care in the choice of the compression function, since there is a trade-off between the overhead incurred in compression and the reduction of communication.

C. TRAINING ROUNDS

The global model is then iterated with many rounds of training to achieve convergence. Total communication cost scales with the number of rounds, linearly (R).

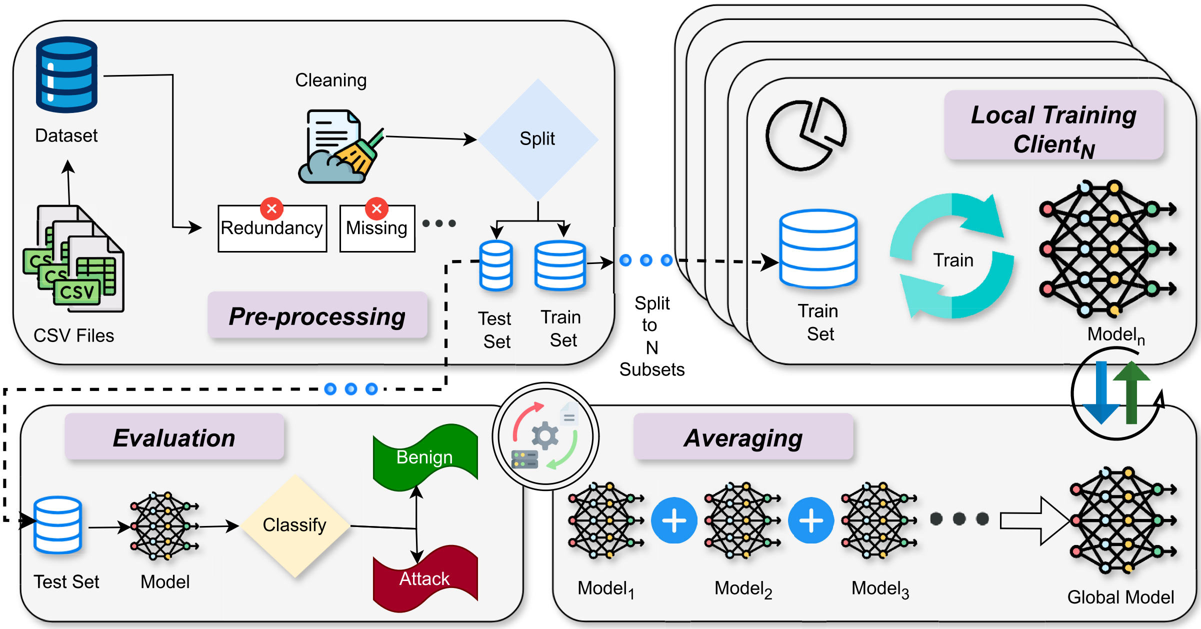


FIGURE 3. Simplified illustration of dataset pre-processing, model training, and evaluation.

1) TIME COMPLEXITY

Now, this complexity can be written in Big O notation as: $\mathcal{T} = O(R \cdot C_{MPC} + C_{compression})$ where:

- R is the number of rounds of training.
- C_{MPC} denotes the communication complexity of the MPC protocol chosen for each round.
- $C_{compression}$ denotes the computational cost associated with the application of the compression function $Q(\cdot)$.

V. EXPERIMENTATION

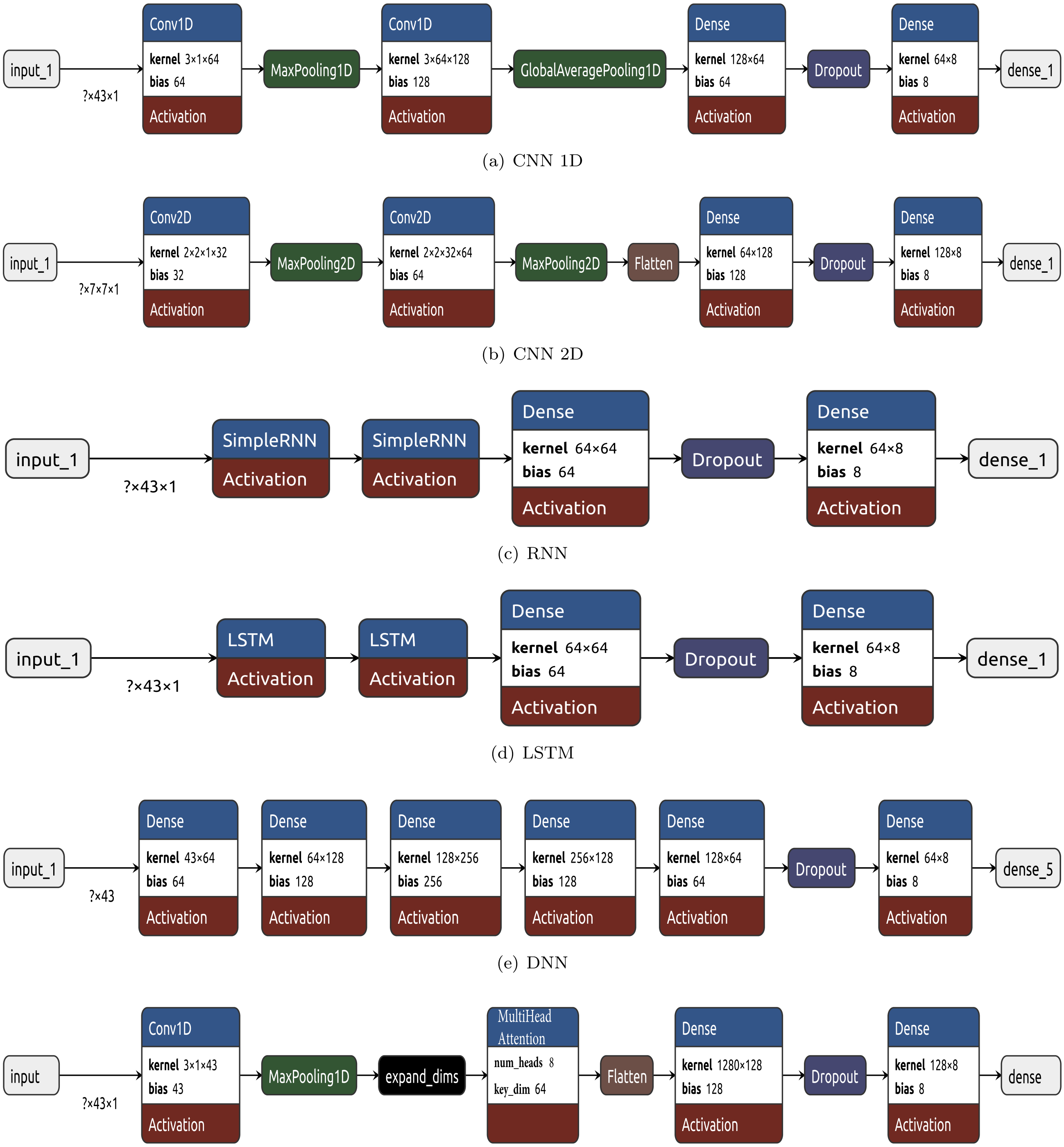
To conduct our experiments, we follow the steps illustrated in Figure 3. We start with the pre-processing steps. Then, we use the training dataset for the training phase, splitting it into five subsets. Each subset is used to train only one federated client. After completing the various rounds of federated learning, we use a model from any client to evaluate the performance using the test dataset. Various experiments were done using different deep neural network models: simple deep neural network, DNN; convolutional neural networks, CNNs; long short-term memory, LSTM, networks; and a hybrid model that combines 1D CNN with Multi-Head Attention. We use the Flower framework for simulation using TensorFlow in training such models with Federated Learning [43]. The CICIoT2023 dataset [44] is used to train the different clients and to test one of them. This choice of dataset is due to its realistic gathered traffic and accurate representation of modern IoT network traffic.

1) THE CICIoT2023 DATASET [44]

The CICIoT2023 dataset is one of the most recent datasets, aiming to assist in the design of security analytics for the IoT environment. It offers full data generated through different IoT attack scenarios. The authors developed a complex IoT network topology with more than 100 devices and subjected these devices to 33 different attacks. These attacks were part of a large set of categories, including Distributed Denial-of-Service (DDoS), Denial-of-Service (DoS), reconnaissance (Recon), web-based attacks, brute-force attacks, spoofing attacks, and Mirai botnet-related attacks. Worth mentioning, that all attacks were initiated from compromised IoT devices to target another device in the network. The process of data collection was three-fold: generating, extracting, and labeling the data. In the first step, the attacks were initiated by simulating various attack scenarios against the prepared IoT network. Next, the generated network traffic through these attacks was captured in pcap format using the Wireshark network protocol analyzer. Finally, the traffic data was captured and labeled against the attack scenario that each segment represents. The captured pcap files formed a substantial amount of data, exceeding approximately 548 GB. To make the data easy to analyze in the next steps, the authors applied a chunking process to the data, dividing it into 10-MB chunks. This chunking allowed for parallel conversion from its pcap format to CSV format, which is more readily analyzable. After chunking the data, the authors used the DPKT library to extract a rich set of features from the traffic data. Table 2 provides a summary of the benign traffic and different attack types, along with their sub-types and the number of instances per type.

2) PRE-PROCESSING

As illustrated in Figure 3, to conduct our experiments, we initially concatenated all CSV files from the CICIoT2023 dataset into a single data frame. Subsequently, we removed



(a) CNN 1D

(b) CNN 2D

(c) RNN

(d) LSTM

(e) DNN

(f) The hybrid model

FIGURE 4. Models used for the experimental evaluation.

redundancy, rows with missing values, and columns with the same value for all rows. Following this, we selected a fixed number of rows for each type of attack and benign activity, as specified in Table 3. Then, we applied normalization to every value within each column using Equation 3. Subsequently, we divided the selected dataset into the training set and the test set. Finally, we split the training set into different training sub-sets, where each sub-set is used to train the model deployed in one node.

$$\overline{x_i(j)} = \frac{x_i(j) - min(x(j))}{max(x(j)) - min(x(j))} \tag{3}$$

A. USED MECTRICS

To evaluate the performance of the proposed models, we employed the True Positive Rate (TPR) for each class, Global Accuracy, Average Detection Rate, False Alarm

TABLE 2. The CICIoT2023 dataset classes distribution.

Category	Attack	Number of Instances
DDoS (72.7%)	DDoS-ICMP_Flood	7,200,504
	DDoS-UDP_Flood	5,412,287
	DDoS-TCP_Flood	4,497,667
	DDoS-PSHACK_Flood	4,094,755
	DDoS-SYN_Flood	4,059,190
	DDoS-RSTFINFlood	4,045,285
	DDoS-SynonymousIP_Flood	3,598,138
	DDoS-HTTP_Flood	28,790
	DDoS-UDP_Fragmentation	286,925
	DDoS-ACK_Fragmentation	285,104
	DDoS-SlowLoris	23,426
	DDoS-ICMP_Fragmentation	452,489
DoS (17.3%)	DoS-UDP_Flood	3,318,595
	DoS-HTTP_Flood	71,864
	DoS-TCP_Flood	2,671,445
	DoS-SYN_Flood	2,028,834
Recon (0.75%)	Recon-PingSweep	2262
	Recon-HostDiscovery	134,378
	Recon-OSScan	98,259
	Recon-PortScan	82,284
Mirai (5.64%)	Mirai-greeth_flood	991,866
	Mirai-udpplain	890,576
	Mirai-greip_flood	751,682
Spoofing (1.04)	DNS_Spoofing	178,911
	MITM-ArpSpoofing	307,593
Web (0.05%)	Uploading_Attack	1252
	DictionaryBruteForce	13,064
	BrowserHijacking	5859
	CommandInjection	5409
	SqlInjection	5245
	XSS	3846
	Backdoor_Malware	3218
BruteForce (0.02%)	DictionaryBruteForce	13,064
Benign (2.3%)	BenignTraffic	1,098,195

Rate, and Average Accuracy, which are respectively detailed in Equation 4, Equation 5, Equation 6, Equation 7, and Equation 8. These metrics are based on the confusion matrix illustrated in Table 4. Besides, we further employ the Receiver Operating Characteristic curve and Area Under the Curve as metrics for performance evaluation. Also, the communication overhead that is presented in Equation 9, where R represents training rounds, N represents the total number of clients, and CMS or the Compressed Model Size, which represents the final model size, after introducing the 8bit quantization using the TensorFlow Model Optimization Toolkit [45] and making it resource constrained friendly using the TensorFlow Lite Framework [46].

$$TPR_{classX} = \frac{TP_{classX}}{TP_{classX} + FN_{classX}} \tag{4}$$

$$Accuracy_{Global} = \frac{\sum^{NBclass} \mathrm{TP}}{\sum^{NBclass} (\mathrm{TP} + \mathrm{FP})} \tag{5}$$

$$DR_{Average} = \frac{\sum TPR_{AttackX}}{NB_{OfAttackX}} \tag{6}$$

$$\mathrm{FAR} = 1 - \frac{TP_{Benign}}{TP_{Benign} + FN_{Benign}} \tag{7}$$

$$ACC_{Average} = \frac{1}{NBClass} \sum TPR_X \tag{8}$$

$$\mathrm{Overhead} = R \cdot N \cdot (N - 1) \cdot \mathrm{CMS} \tag{9}$$

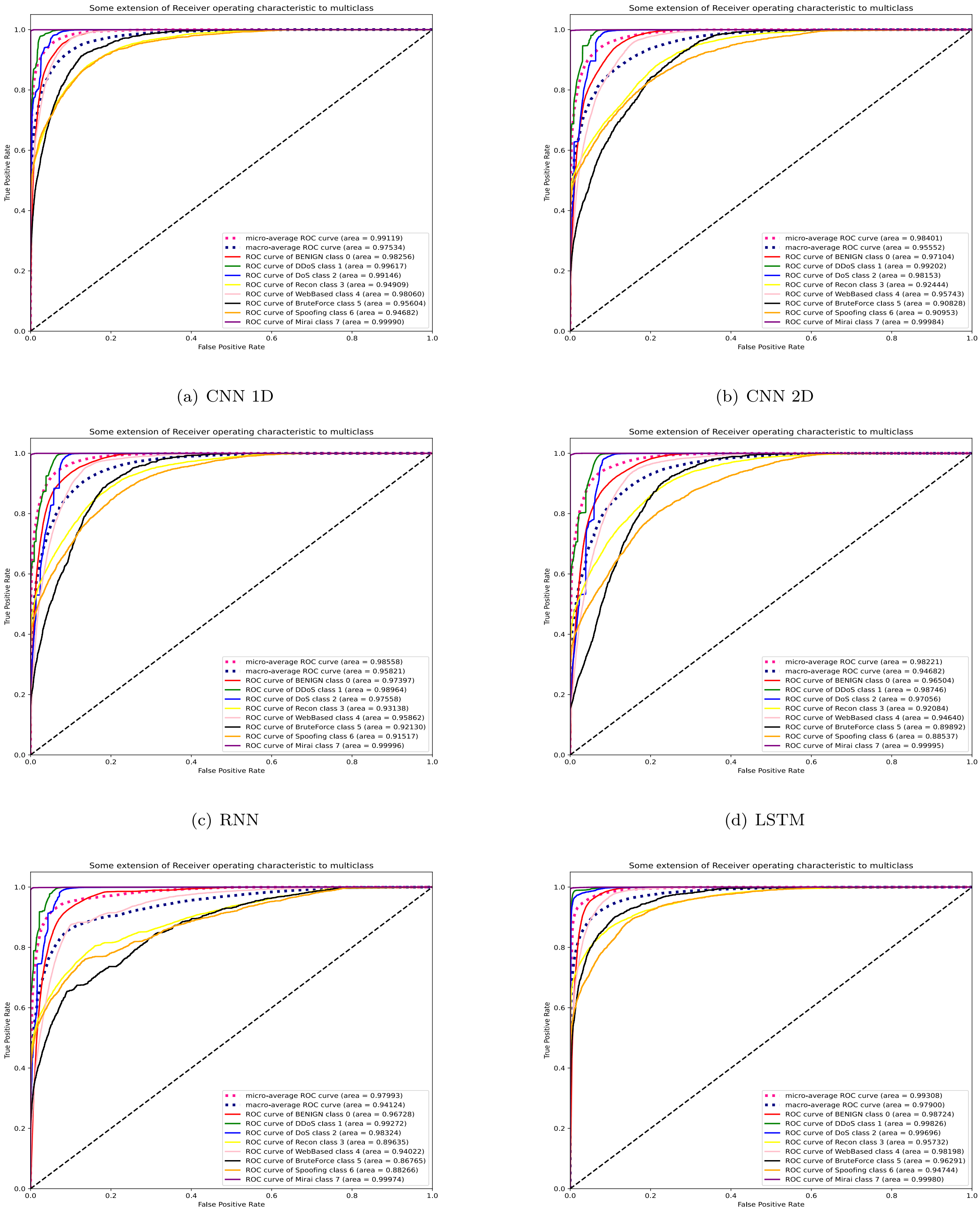
B. MODELS USED FOR FEDERATED LEARNING

Figure 4 summarized the different models used as initial models for federated learning. These models are broadcast to all the participating clients. The models employed are as follows:

- *CNN1D model:* This is a sequential model architecture. Two convolutional layers (Conv1D) are employed: the first with kernel size 3 × 1×64 and a bias of 64. Following the first convolutional layer is a max pooling layer to reduce the dimensionality of the data. The second convolutional layer is applied, with 128 biases and a kernel size of 3 × 64×128. A non-linear activation function is applied. Following this, a global average pooling is applied to reduce the data along the spatial dimension. A dropout layer is used to prevent over-fitting. Finally, there exists a fully connected layer.
- *CNN2D model:* The first layer is a convolutional 2D layer with a kernel size of 2 × 2×1 × 32 and applies a bias of 32. After that, a max pooling 2D layer is applied to reduce the data dimensionality. A second convolutional layer 2D is applied with a kernel size of 2×2×32×64 and applies a bias of 64, with an activation layer. Then a max pooling 2D is applied to further reduce the data. A flattening layer into a 1D vector is applied before a fully connected layer. A dropout layer is applied to prevent over-fitting. Finally, we find a fully connected layer.
- *RNN model:* The structure is sequential with an input layer, 2 RNN layers—SimpleRNN, and then there are the dense layers. The input layer gets a vector corresponding to the first element in the sequence. Then an activation function adds non-linearity to the output coming from the RNN layer. A dense layer finally transforms the sequence data that is processed into the desired output vector.
- *LSTM:* The same network architecture is used for the RNN model, except we will replace the Simple RNN layers with the LSTM layer.
- *Hybrid model:* It will amalgamate CNN with a Transformer block. A CNN—1D tries to extract features from the networking traffic data, while a multi-head attention analyzes the relationships between different data points in sequences and could therefore show long-range patterns of attacks.

1) USED HYPERPARAMATERS

To reach our purpose and achieve high performance of the FBMP-IDS system, different values of various hyper-parameters were tried. After several attempts, we identified the optimal hyper-parameters and their corresponding models, which are detailed in Table 5.



(a) CNN 1D

(b) CNN 2D

(c) RNN

(d) LSTM

(e) DNN

(f) The hybrid model

FIGURE 5. **ROC curve and AUC ROC of the different models using federated learning.**

C. RESULTS

The effectiveness of the FBMP-IDS architecture is assessed in this part. In the process of evaluation, we used the different training sub-datasets for the training steps of the different nodes, and the test sub-set for the test step. Table 6 and Figure 6 summarize the results obtained for the various deep learning models.

TABLE 3. Distribution of attacks and benign classes on training and test subsets extracted from CICIoT2023 dataset.

Type of connection			Training	Test	All
Bening			105025	44972	149997
Attack	DDoS	ACK Fragmentation	3545	1455	5000
		UDP Flood	3562	1438	5000
		SlowLoris	3526	1474	5000
		ICMP Flood	3462	1538	5000
		RSTFIN Flood	3470	1530	5000
		PSHACK Flood	3520	1480	5000
		HTTP Flood	3518	1482	5000
		UDP Fragmentation	3482	1518	5000
		ICMP Fragmentation	3515	1485	5000
		TCP Flood	3531	1469	5000
		SYN Flood	3460	1540	5000
		SynonymousIP Flood	3427	1573	5000
		Total DDoS	42018	17982	60000
	DoS	TCP Flood	3498	1502	5000
		HTTP Flood	3546	1453	4999
		SYN Flood	3502	1498	5000
		UDP Flood	3467	1533	5000
		Total DoS	14013	5986	19999
	Recon	Ping Sweep	1554	708	2262
		OS Scan	3453	1547	5000
		Vulnerability Scan	3521	1479	5000
		Port Scan	3485	1515	5000
		Host Discovery	3511	1489	5000
		Total Recon	15524	6738	22262
	Web-Based	Sql Injection	3436	1564	5000
		Command Injection	3549	1451	5000
		Backdoor Malware	2194	1024	3218
		Uploading Attack	892	360	1252
		XSS	2718	1128	3846
		Browser Hijacking	3517	1483	5000
		Total Web-Based	16306	7010	23316
	Brute Force	Dictionary Brute Force	3497	1503	5000
	Spoofing	Arp Spoofing	3536	1464	5000
		DNS Spoofing	3515	1485	5000
		Total Spoofing	7051	2949	10000
	Mirai	GREIP Flood	3484	1516	5000
		Greeth Flood	3491	1509	5000
		UDPPlain	3491	1509	5000
		Total Miria	10466	4534	15000
		Total Attack	108875	46702	155577
Total			213900	91674	305574

TABLE 4. Confusion matrix.

		Predicted class	
		Negative (Benign)	Positive (Attack)
Effective class	Negative (Benign)	True negative	False positive
	Positive (Attack)	False negative	True positive

1) SPECIFIC METRICS

The table groups the True Positive Rates (TPRs) that each model has achieved in detecting various intrusion categories. Results can be seen in the FBMP-IDS with the Hybrid Model, attaining the highest overall performance and highest detection rate with 99.43% in Mirai attacks, 70.39% in Recon-based attacks, and 96.76% in DDoS-based attacks while being powerful in the identification of other attack types. As depicted in Figure 6, the following are the comparisons of the models based on the TPR obtained:

- *Benign Traffic:* The majority of models obtained high TPR values, demonstrating how these models are capable of distinguishing between normal and malicious traffic. The best of these models was CNN1D with a TPR of 98.45% and the second one is the Hybrid Model with a TPR of 97.45%.
- *Distributed Denial-of-Service Attacks:* Again, the Hybrid Model was better than the rest with a TPR of 96.76%. The DNN and RNN models performed very well in detecting DDoS attacks, with TPR values of 96.25% and 96.06% respectively.
- *Denial-of-Service (DoS) Attack:* As in the DDoS attack, the Hybrid Model was the best of all models, with a TPR of 94.65%. While CNN2D, RNN, and others had lower rates.
- *Reconnaissance Attacks:* The Hybrid Model achieved the highest TPR (70.39%) in the case of reconnaissance attack detection. A good TPR value was received by DNN at 57.15%. All other models showed lower performance.
- *Web-based Attack:* Again, the Hybrid Model showed strong performance at 74.39% in the case of web-based attack detection. A notable TPR (80.63%) is achieved by CNN2D, too, while others showed lower detection rates.

TABLE 5. Hyperparamaters of the different models used for the experimental evaluation.

Hyperparameters		CNN1D	CNN2D	RNN	LSTM	DNN	Hybrid Model
Client parameters	Learning rate	0.001	0.001	0.001	0.001	0.001	0.001
	Batch size	64	64	64	64	64	256
	Number of epochs	70	70	20	15	70	70
	Optimizer	adam	adam	adam	adam	adam	adam
	Loss function	sparse categorical crossentropy	sparse categorical crossentropy	sparse categorical crossentropy	sparse categorical crossentropy	sparse categorical crossentropy	sparse categorical crossentropy
Server Parameters	Number of Round	40	40	10	10	40	70
	Number of clients	5	5	5	5	5	5
	Fraction of clients per round	100%	100%	100%	100%	100%	100%
	Aggregation method	FedAvg	FedAvg	FedAvg	FedAvg	FedAvg	FedAvg
	Client selection strategy	Both	Both	Both	Both	Both	Both

TABLE 6. Obtained results for the different used models.

		CNN1D	CNN2D	RNN	LSTM	DNN	Hybrid Model
Specific Metrics	TPR BENIGN	98,45%	91,09%	96,30%	95,22%	94,20%	97,45 %
	TPR DDoS	93,68%	95,34%	96,06%	90,49%	96,25%	96,76 %
	TPR DoS	78,67%	45,42%	36,54%	46,58%	47,58%	94,65 %
	TPR Recon	53,64%	51,07%	45,52%	40,52%	57,15%	70,39%
	TPR WebBased	71,40%	80,63%	67,90%	64,96%	57,82%	74,39%
	TPR BruteForce	25,68%	16,10%	15,17%	14,70%	29,67%	51,56%
	TPR Spoofing	45,41%	43,30%	38,12%	26,45%	46,42%	54,70 %
	TPR Mirai	99,29%	99,40%	99,25%	99,23%	99,14%	99,43%
Global Metrics	Accuracy	88,00%	82,85%	83,39%	81,45%	83,70%	91,35 %
	FAR	1,55%	8,91%	3,70%	4,78%	5,80%	2,55 %
	Average DR	66,82%	61,61%	56,94%	54,70%	62,01%	77,41 %
	Average Accuracy	70,78%	65,30%	61,86%	59,77%	66,03%	79,92%

- Brute-Force Attack: Hybrid Models showed significant improvement in the case of brute-force attack detection over the other models and achieved a TPR of 51.56%. While all others showed much lower TPR.
- Spoofing Attack: Again, the Hybrid model showed the best performance at a TPR of 54.70% in the case of spoofing attack detection. Also, moderate TPRs are shown by CNN1D and DNN models, while others showed lower detection rates.
- Mirai Botnet Attack: All models achieved exceptionally high TPRs for detecting Mirai botnet attacks and obtained values over 99%, demonstrating how these are very capable of finding that threat. And the best among them was the hybrid model at 99.43%.

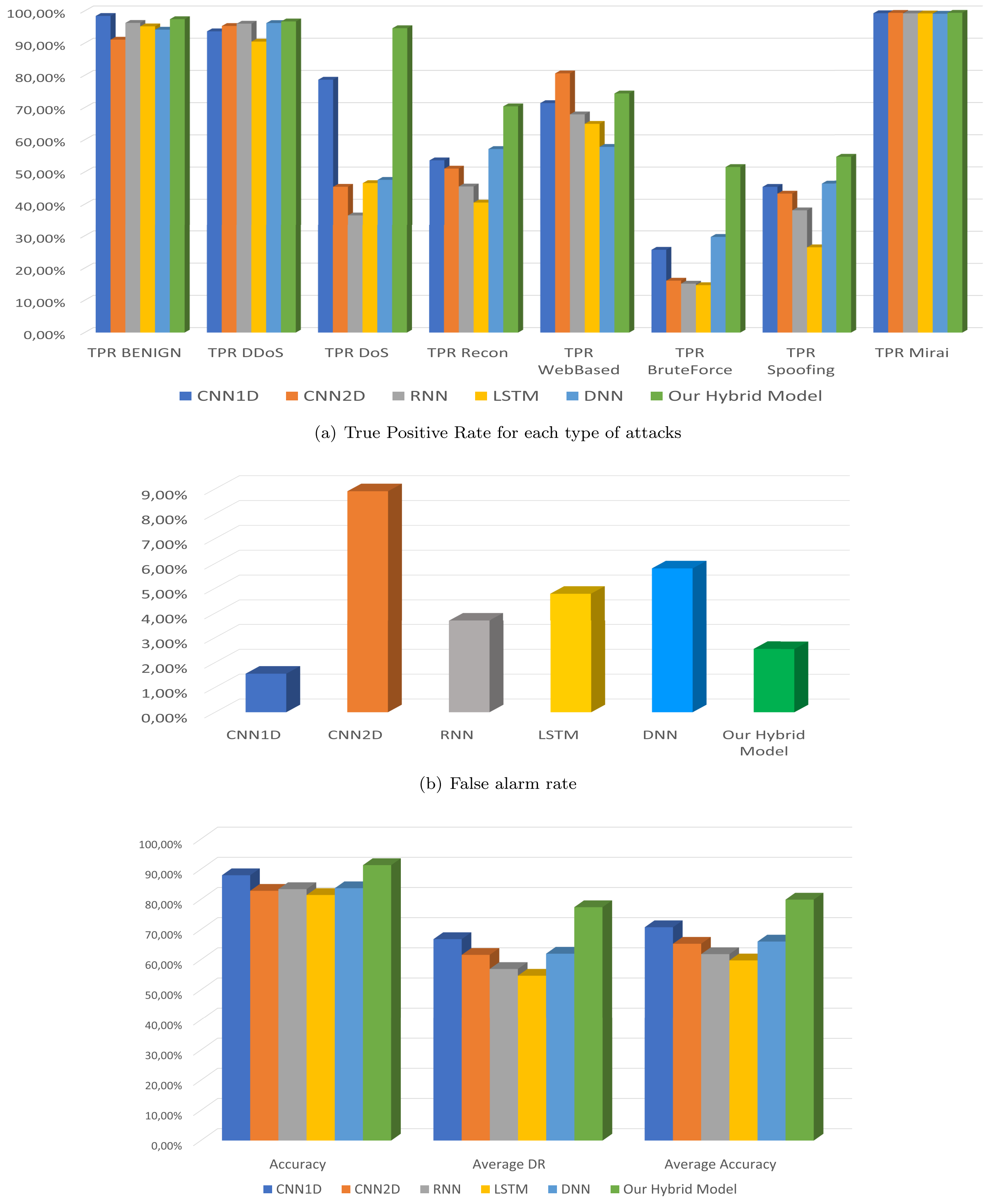
2) GLOBAL PERFORMANCE METRICS

As shown in Table 6, the Global performance metrics include Accuracy, False Alarm Rate (FAR), Average Detection Rate (DR), and Average Accuracy. Figure 6 shows that the Hybrid Model has the highest overall Accuracy of 91.35% and Average Accuracy of 79.92%, along with a very low FAR of 2.55%. Hence, the results indicate the performance of the Hybrid Model in terms of an adequate balance between the accurate detection of intrusions and low false alarms.

3) ROC ANALYSIS FOR THE DIFFERENT MODELS

The performance evaluation of the FBMP-IDS was not limited only to True Positive Rates of each intrusion class. For a better description of the performance of models, Receiver Operating Characteristic curves were also constructed for each deep learning model that can be used in the FBMP-IDS framework, as illustrated in Figure 5. The ROC curve plots the trade-off between the True Positive Rate (TPR) and the False Positive Rate (FPR) at different classification thresholds.

In this part, we go through the strengths and weaknesses of each model using micro-averaged ROC curves and individual class ROC curves. Analysis of the ROC curves indicated distinct performance levels among the compared classification models. Importantly, for a model's capability to separate between the positive and negative instances among all classes, the Hybrid model (Figure 5 (f)) scored the highest micro-averaged ROC score of 0.993%, which definitely denoted superior overall classification performance. This means that the model has satisfactory robustness to deal with potential class imbalances within the dataset. The micro-averaged ROC curve, however, does not show any possible weaknesses at the level of individual classes. To further delve into the matter, we analyze the class-specific ROC curves. Herein, we note that several models, including



(a) True Positive Rate for each type of attacks

(b) False alarm rate

(c) Global metrics

FIGURE 6. **TPR and global metrics for the different used models.**

CNN2D (Figure 5 (b)), LSTM (Figure 5 (d)), and DNN (Figure 5 (e)), perform poorly in certain classes; this is evidenced by lower AUC values and less defined curves in their class-specific ROC plots. In contrast, the Hybrid model

TABLE 7. Models sizes and the network overhead.

	CNN1D	CNN2D	RNN	LSTM	DNN	Our Hybrid Model
Original model Total params	33,736 (131.78 KB)	17,768 (69.41 KB)	17,160 (67.03 KB)	54,600 (213.28 KB)	85,832 (335.28 KB)	206,602 (807.04 KB)
Original model Trainable params	33,736 (131.78 KB)	17,768 (69.41 KB)	17,160 (67.03 KB)	54,600 (213.28 KB)	85,832 (335.28 KB)	206,602 (807.04 KB)
Compressed model size	42.29 KB	22.21 KB	32.66 KB	77.96 KB	101 KB	251 KB
Network Overhead	33.04 MB	17.35 MB	6.38 MB	15.23 MB	78.91 MB	343.16 MB

shows a consistently good performance across most classes from its individual class ROC curves.

While the micro-averaged ROCs provided a very good initial idea regarding the performance of the models, the single-class ROCs proved to be very interesting. For instance, while the CNN1D model (Figure 5 (a)) managed a good micro-averaged ROC score of 0.991%, it showed its limits in handling the "Recon" class. On the other hand, the CNN2D (Figure 5 (b)) model was poor in both the "Recon" and "Spoofing" classes. Surprisingly, both the RNN and LSTM models performed similarly but were marked with certain inconsistencies at the class level. For instance, the Hybrid model performed better than the LSTM model on the "Brute-force" class with an AUC of 0.96% against 0.89%. However, the DNN models performed worst of all, with AUCs less than 0.9 for three classes: "Recon", "Bruteforce", and "Spoofing". These observations underline the importance of looking both at micro-averaged and single-class ROC curves for a proper evaluation. Though a model can have strong performance in general, it may still have problems with some classes due to some inherent limitations in the architecture or some potential biases in the training data. It's very important to be aware of weaknesses in class performance for real-world applications since class importance may vary. For instance, it may be imperative to correctly classify the "DoS" class, whereas the misclassification of the "Recon" class may not be very serious.

D. COMPARISONS

Comparisons presented in Figure 6, show that across various assessment metrics, the effectiveness of the FBMP-IDS architecture, particularly the Hybrid Model, is very effective in detecting a wide range of intrusion types. The Hybrid Model consistently achieved superior performance across most intrusion categories, which shows how the model is flexible and robust. Furthermore, the Hybrid Model showed the best AUC-ROC across different intrusion categories. This finding strengthens the conclusion that the Hybrid Model shows superior performance compared to other models of individual deep learning models within the FBMP-IDS framework. Table 7 presents the communication overhead for the different models. While the analysis shows that the hybrid model has the highest communication overhead due to its large number of parameters, this apparent drawback does need to be weighed against possible benefits. With this, the hybrid architecture can leverage the strengths of the CNNs and Multi-Head Attention synergistically to come up with superior performance on the task in question. This becomes significantly interesting for tasks that involve complex relationships in the data or that require high accuracy. Therefore, if optimal performance is the objective, then probably the marginally increased overhead of the hybrid model can be justified as a concession in light of the possible gains.

VI. CONCLUSION

Coupled with the realization of 6G wireless networks are unprecedented opportunities and challenges in network security and privacy. Leveraging the power of AI in network connectivity foreseen in 6G networks demands network intrusion detection and prevention mechanisms that scale and dynamically adapt to the constantly changing network topology and distributed nature of these networks. In this work, we propose a new secure gradients exchange algorithm for distributed intrusion detection in 6G networks, synergistically combining the power of federated learning with secure multi-party computation and blockchain. Our proposed system allows collaborative training of intrusion detection models, preserving data privacy and secure gradient aggregation through MPC protocols, ensuring adaptivity in secure aggregation to optimize communication overhead and computation complexity in real time. Blockchain technology is used to offer a decentralized, transparent, and tamper-proof FL process. Finally, a hybrid model architecture is presented, where Convolutional Neural Networks are used for feature extraction and Multi-Head Attention for better contextual analysis in order to enhance detection rates and reduce the occurrence of false alarm rates. We have demonstrated the feasibility of the proposed approach through extensive experimental evaluations and comparisons against several baseline models.

REFERENCES

[1] S. Dang, O. Amin, B. Shihada, and M.-S. Alouini, "What should 6G be?" *Nature Electron.*, vol. 3, no. 1, pp. 20–29, Jan. 2020.

[2] T. Nguyen, N. Tran, L. Loven, J. Partala, M.-T. Kechadi, and S. Pirttikangas, "Privacy-aware blockchain innovation for 6G: Challenges and opportunities," in *Proc. 2nd 6G Wireless Summit (6G SUMMIT)*, Mar. 2020, pp. 1–5.

[3] M. Matthaiou, O. Yurduseven, H. Q. Ngo, D. Morales-Jimenez, S. L. Cotton, and V. F. Fusco, "The road to 6G: Ten physical layer challenges for communications engineers," *IEEE Commun. Mag.*, vol. 59, no. 1, pp. 64–69, Jan. 2021.

[4] K. B. Letaief, Y. Shi, J. Lu, and J. Lu, ''Edge artificial intelligence for 6G: Vision, enabling technologies, and applications,'' *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 5–36, Jan. 2022.

[5] A. Ahmim and N. Ghoualmi-Zine, ''A new fast and high performance intrusion detection system,'' *Int. J. Secur. Appl.*, vol. 7, no. 5, pp. 67–80, Sep. 2013.

[6] A. Ahmim, F. Maazouzi, M. Ahmim, S. Namane, and I. B. Dhaou, ''Distributed denial of service attack detection for the Internet of Things using hybrid deep learning model,'' *IEEE Access*, vol. 11, pp. 119862–119875, 2023.

[7] S. Sakraoui, M. Derdour, and A. Ahmim, ''6G-SECUREIDS: Blockchain-enhanced secure knowledge transfer for distributed intrusion detection systems in advanced networks,'' in *Proc. Int. Conf. Netw. Adv. Syst. (ICNAS)*, Oct. 2023, pp. 1–6.

[8] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, ''Communication-efficient learning of deep networks from decentralized data,'' in *Proc. 20th Int. Conf. Artif. Intell. Statist.*, vol. 54, 2017, pp. 1273–1282.

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, ''Attention is all you need,'' in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

[10] S. Rezvy, Y. Luo, M. Petridis, A. Lasebae, and T. Zebin, ''An efficient deep learning model for intrusion classification and prediction in 5G and IoT networks,'' in *Proc. 53rd Annu. Conf. Inf. Sci. Syst. (CISS)*, Mar. 2019, pp. 1–6.

[11] C. Kolias, G. Kambourakis, A. Stavrou, and S. Gritzalis, ''Intrusion detection in 802.11 networks: Empirical evaluation of threats and a public dataset,'' *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 184–208, 1st Quart., 2016.

[12] H. Moudoud, L. Khoukhi, and S. Cherkaoui, ''Prediction and detection of FDIA and DDoS attacks in 5G enabled IoT,'' *IEEE Netw.*, vol. 35, no. 2, pp. 194–201, Mar. 2021.

[13] J. Zhu, S. He, J. Liu, P. He, Q. Xie, Z. Zheng, and M. R. Lyu, ''Tools and benchmarks for automated log parsing,'' in *Proc. IEEE/ACM 41st Int. Conf. Softw. Eng. Softw. Eng. Pract. (ICSE-SEIP)*, May 2019, pp. 121–130.

[14] I. H. Abdulqadder, S. Zhou, D. Zou, I. T. Aziz, and S. M. A. Akber, ''Multi-layered intrusion detection and prevention in the SDN/NFV enabled cloud of 5G networks using AI-based defense mechanisms,'' *Comput. Netw.*, vol. 179, Oct. 2020, Art. no. 107364.

[15] Z. Zhang, Y. Cao, Z. Cui, W. Zhang, and J. Chen, ''A many-objective optimization based intelligent intrusion detection algorithm for enhancing security of vehicular networks in 6G,'' *IEEE Trans. Veh. Technol.*, vol. 70, no. 6, pp. 5234–5243, Jun. 2021.

[16] E. Seo, H. M. Song, and H. K. Kim, ''GIDS: GAN based intrusion detection system for in-vehicle network,'' in *Proc. 16th Annu. Conf. Privacy, Secur. Trust (PST)*, Aug. 2018, pp. 1–6.

[17] M. Almiani, A. AbuGhazleh, Y. Jararweh, and A. Razaque, ''DDoS detection in 5G-enabled IoT networks using deep Kalman backpropagation neural network,'' *Int. J. Mach. Learn. Cybern.*, vol. 12, no. 11, pp. 3337–3349, Nov. 2021.

[18] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, ''Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy,'' in *Proc. Int. Carnahan Conf. Secur. Technol. (ICCST)*, Oct. 2019, pp. 1–8.

[19] H. Chen, B. Yuan, D. Zou, and H. Jin, ''A fuzzing-based method for testing rules in intrusion detection systems in 6G networks,'' *IEEE Netw.*, vol. 36, no. 4, pp. 150–158, Jul. 2022.

[20] A. Alotaibi and A. Barnawi, ''IDSoft: A federated and softwarized intrusion detection framework for massive Internet of Things in 6G network,'' *J. King Saud Univ. Comput. Inf. Sci.*, vol. 35, no. 6, Jun. 2023, Art. no. 101575.

[21] Y. LeCun. (1998). *The MNIST Database of Handwritten Digits*. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[22] L. Jai Vinita and V. Vetriselvi, ''Federated learning-based misbehaviour detection on an emergency message dissemination scenario for the 6G-enabled Internet of Vehicles,'' *Ad Hoc Netw.*, vol. 144, May 2023, Art. no. 103153.

[23] L. J. Vinita and V. Vetriselvi, ''Impact of Sybil attack on software-defined vehicular fog computing (SDVF) for an emergency vehicle scenario,'' in *Proc. Inventive Commun. Comput. Technol. (ICICCT)*. Cham, Switzerland: Springer, 2022, pp. 809–825.

[24] M. Prasad, S. Tripathi, and K. Dahal, ''An intelligent intrusion detection and performance reliability evaluation mechanism in mobile ad-hoc networks,'' *Eng. Appl. Artif. Intell.*, vol. 119, Mar. 2023, Art. no. 105760.

[25] J. Du, T. Lin, C. Jiang, Q. Yang, C. F. Bader, and Z. Han, ''Distributed foundation models for multi-modal learning in 6G wireless networks,'' *IEEE Wireless Commun.*, vol. 31, no. 3, pp. 20–30, Jun. 2024.

[26] J. Du, C. Jiang, A. Benslimane, S. Guo, and Y. Ren, ''SDN-based resource allocation in edge and cloud computing systems: An evolutionary Stackelberg differential game approach,'' *IEEE/ACM Trans. Netw.*, vol. 30, no. 4, pp. 1613–1628, Aug. 2022.

[27] M. A. Ferrag, O. Friha, L. Maglaras, H. Janicke, and L. Shu, ''Federated deep learning for cyber security in the Internet of Things: Concepts, applications, and experimental analysis,'' *IEEE Access*, vol. 9, pp. 138509–138542, 2021.

[28] E. Ashraf, N. F. F. Areed, H. Salem, E. H. Abdelhay, and A. Farouk, ''FIDChain: Federated intrusion detection system for blockchain-enabled IoT healthcare applications,'' *Healthcare*, vol. 10, no. 6, p. 1110, Jun. 2022.

[29] H. Sedjelmaci and N. Ansari, ''Zero trust architecture empowered attack detection framework to secure 6G edge computing,'' *IEEE Netw.*, vol. 38, no. 1, pp. 196–202, Jan. 2024.

[30] W. Wu, C. Zhou, M. Li, H. Wu, H. Zhou, N. Zhang, X. S. Shen, and W. Zhuang, ''AI-native network slicing for 6G networks,'' *IEEE Wireless Commun.*, vol. 29, no. 1, pp. 96–103, Feb. 2022.

[31] W. Issa, N. Moustafa, B. Turnbull, N. Sohrabi, and Z. Tari, ''Blockchain-based federated learning for securing Internet of Things: A comprehensive survey,'' *ACM Comput. Surv.*, vol. 55, no. 9, pp. 1–43, Sep. 2023.

[32] J. Wu, C. Leng, Y. Wang, Q. Hu, and J. Cheng, ''Quantized convolutional neural networks for mobile devices,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4820–4828.

[33] R. Krishnamoorthi, ''Quantizing deep convolutional networks for efficient inference: A whitepaper,'' 2018, *arXiv:1806.08342*.

[34] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. Kalenichenko, ''Quantization and training of neural networks for efficient integer-arithmetic-only inference,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2704–2713.

[35] E. Wang, J. J. Davis, D. Moro, P. Zielinski, J. J. Lim, C. Coelho, S. Chatterjee, P. Y. K. Cheung, and G. A. Constantinides, ''Enabling binary neural network training on the edge,'' in *Proc. 5th Int. Workshop Embedded Mobile Deep Learn.*, Jun. 2021, pp. 37–38.

[36] C. N. Coelho Jr., A. Kuusela, S. Li, H. Zhuang, T. Aarrestad, V. Loncar, J. Ngadiuba, M. Pierini, A. Alan Pol, and S. Summers, ''Automatic heterogeneous quantization of deep neural networks for low-latency inference on the edge for particle detectors,'' 2020, *arXiv:2006.10159*.

[37] C. N. Coelho, A. Kuusela, S. Li, H. Zhuang, J. Ngadiuba, T. K. Aarrestad, V. Loncar, M. Pierini, A. A. Pol, and S. Summers, ''Automatic heterogeneous quantization of deep neural networks for low-latency inference on the edge for particle detectors,'' *Nature Mach. Intell.*, vol. 3, no. 8, pp. 675–686, Jun. 2021.

[38] B. Moons, K. Goetschalckx, N. Van Berckelaer, and M. Verhelst, ''Minimum energy quantized neural networks,'' in *Proc. 51st Asilomar Conf. Signals, Syst., Comput.*, Oct. 2017, pp. 1921–1925.

[39] D. Byrd and A. Polychroniadou, ''Differentially private secure multi-party computation for federated learning in financial applications,'' in *Proc. 1st ACM Int. Conf. AI Finance*, Oct. 2020, pp. 1–9.

[40] V. Mugunthan, A. Polychroniadou, D. Byrd, and T. H. Balch, ''SMPAI: Secure multi-party computation for federated learning,'' in *Proc. NeurIPS Workshop Robust AI Financial Services*, vol. 21. Cambridge, MA, USA: MIT Press, 2019.

[41] E. Sotthiwat, L. Zhen, Z. Li, and C. Zhang, ''Partially encrypted multi-party computation for federated learning,'' in *Proc. IEEE/ACM 21st Int. Symp. Cluster, Cloud Internet Comput. (CCGrid)*, May 2021, pp. 828–835.

[42] Z. Beerliová-Trubìniová and M. Hirt, ''Perfectly-secure MPC with linear communication complexity,'' in *Proc. Theory Cryptography Conf.* Cham, Switzerland: Springer, 2008, pp. 213–230.

[43] Flower. (2024). *Flower: A Friendly Federated Learning Framework*. Accessed: Jan. 11, 2024. [Online]. Available: https://flower.ai/

[44] E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, and A. A. Ghorbani, ''CICIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment,'' *Sensors*, vol. 23, no. 13, p. 5941, 2023.

[45] (2024). *TensorFlow Model Optimization*. Accessed: Jan. 11, 2024. [Online]. Available: https://www.tensorflow.org/model_optimization

[46] (2024). *TensorFlow Lite*. Accessed: Jan. 11, 2024. [Online]. Available: https://www.tensorflow.org/lite

SABRINA SAKRAOUI received the bachelor's and master's degrees in computer science from Badji Mokhtar–Annaba University, Algeria, in 2016 and 2018, respectively, where she is currently pursuing the Ph.D. degree in networks and cybersecurity with the Networks and Systems Laboratory (LRS). Her research interests include 5G/beyond 5G networks, blockchain technology, and the application of advanced machine learning techniques for cybersecurity purposes.

AHMED AHMIM received the bachelor's, master's, and Ph.D. degrees in computer science from Badji Mokhtar–Annaba University, Algeria, in 2007, 2009, and 2014, respectively. From May 2015 to September 2019, he was an Assistant Professor with the Department of Mathematics and Computer Science, University of Larbi Tebessi—Tebessa, Algeria. Since October 2019, he has been a Senior Lecturer with the Department of Mathematics and Computer Science, Mohamed Cherif Messaadia University-Souk Ahras, Algeria. His research interests include the IoT, computer security, network security, machine learning, deep learning, federated learning, and intrusion detection systems. He has served as a reviewer for various journals, including Elsevier, IEEE, Springer, and Wiley. In addition, he has been actively involved in organizing international conferences, serving as an organizing committee member in various capacities, such as the track chair, the co-chair, the publicity chair, and the proceedings editor.

MAKHLOUF DERDOUR received the Engineering degree in computer sciences from the University of Constantine, Algeria, in 2004, the Magister degree in computer sciences from the University of Tebessa, and the Ph.D. degree in computer networks from the University of Pau and Pays de l'Adour (UPPA), France, in 2012. He has been a Full Professor with the Department of Computer Science, University of Oum El Bouaghi, Algeria, and the Director of the Artificial Intelligence and Autonomous Things Laboratory, since 2023. His research interests include software architecture, multimedia applications, adaptation and self-adaptation of applications, design and modeling of systems, and systems security. He is the Founder and the General Chair of the International Conference on Pattern Recognition and Intelligent Systems (PAIS).

MARWA AHMIM received the Ph.D. degree in computer science from Badji Mokhtar–Annaba University, Algeria, in 2016. She is currently an Associate Professor with the Department of Computer Science, Badji Mokhtar–Annaba University. Her current research interests include network security, security metrics, blockchain, cryptography, and the Internet of Things security.

SARRA NAMANE received the Ph.D. degree in computer science from Badji Mokhtar-Annaba University, in 2018. She is currently an Associate Professor with the Department of Computer Science, Badji Mokhtar–Annaba University. Her research interests include network security, grid computing security, cloud computing security, access control techniques, blockchain, and the Internet of Things security.

IMED BEN DHAOU (Senior Member, IEEE) received the Ph.D. degree from the Royal Institute of Technology, Sweden.

He is currently an Associate Professor and a Docent in embedded systems for IoT. Since 2021, he has been with the Department of Computer Science, Dar Al-Hekma University. He has authored and co-authored more than 110 journals and conference papers, book chapters, and technical reports.

Dr. Ben Dhaou received numerous awards, including the Best Paper Award from the 1997 Finnish Symposium on Signal Processing, a travel grant from the Ph.D. forum at DAC (Los Angeles, in 2000), a Publication Award from Qassim University, and Dr. Hussein Mohammed Al-Sayyed Award for Research. In recognition of his commitment to scientific inquiry and innovation, he is a valued member of Sigma Xi, The Scientific Research Honor Society. Since September 2014, he has been an Editor of *Microelectronics Journal* (Elsevier). He served as a Guest Editor for four special issues of the ISI journals (*Electronics*, *Journal of Cloud Computing*, *Analogue Integrated Circuits and Signal Processing*, and *Microprocessors and Microsystems*). He has chaired or served on the TPC for various conferences in his core areas of expertise.