People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
University of 8 Mai 1945Guelma

Faculty of Mathematics, Computer Science and Material Sciences
Department of Computer Science
Domiciliation laboratory of Information and Communication Sciences and Technologies

Thesis

Submitted in Candidacy for the Degree of *Doctorate in Third Cycle*

Field: Computer Science. Stream: Mathematics and Computer Science
Speciality: Information and Communication Sciences and Technologies

Presented by:
Hamouda Djallel

Title

New Technologies for Security and Privacy Issues in the Era of Industry 5.0

Defended on: 12/06/2024 Before the jury composed of:

Full name	Rank	University	
Mr Mohamed Nemissi	Professor	Univ. of 8 May 1945, Guelma	President
Ms Nadjette Benhamida	MCA	Univ. of 8 May 1945, Guelma	Supervisor
Ms Hassina Seridi	Professor	Univ. of Badji Mokhtar, Annaba	Examiner
Mr Brahim Farou	Professor	Univ. of 8 May 1945, Guelma	Examiner
Mr Ali Khebizi	MCA	Univ. of 8 May 1945, Guelma	Examiner
Mr Seridi Hamid	Professor	Univ. of 8 May 1945, Guelma	Invited

Academic year: 2023/2024

Acknowledgments

** مَنْ لمْ يشْكُر النَّاسَ لَمْ يشْكُر الله ** رواه الترمذي عن النبي محمد ﷺ

"*Whoever does not thank people does not thank Allah."*
*–**Prophet Muhammad** "Peace be upon him"*

*First and foremost, all praise is due to **Allah**, the Almighty, for granting me the courage, patience, and good health required to embark on this endeavour. I am deeply grateful for Allah's divine guidance, which made ways for me where there was no way.*

*My heartfelt appreciation goes my advisor, Dr.**Benhamida Nadjette**, for her extensive support, insightful feedback, and constant encouragement.*

*I am deeply grateful to Dr. **Mohamed Amine Ferrag**, who has consistently provided invaluable guidance, counsel, and support throughout this journey. I am grateful for the opportunities that have enhanced this work's quality and international standing.*

*I also extend my gratitude to Prof. **Seridai Hamid**, the head of the "Labstic" research lab, and the entire team for their excellent mentorship, all of which enriched this research.*

*I would like to express my sincere gratitude to the members of the jury to Pr **Seridi Hassina**, Pr **Farou Brahim** and Dr **Khebizi Ali** who dedicated their time, expertise, and insightful feedback to the evaluation of this thesis.*

*Finally, I want to convey my deepest and most sincere feelings to my beloved parents, Mr. **Hamouda Yazid** and Ms. **Mechtar Razika.** Your sacrifices for my success, prayers, and constant encouragement have been my pillars of strength throughout this journey.*

Djallel Hamouda

ABSTRACT

INDUSTRY 5.0, the latest industrial revolution, advances the Smart Factory concept by emphasizing Human-Machine collaboration and sustainability. It incorporates the human aspect into industrial processes, promoting critical thinking, personalization, and adaptability, while leveraging technologies like IoT and AI for increased efficiency and productivity. However, this era also introduces a complex landscape of cyber threats. As machines, systems, and humans become interconnected, ensuring cybersecurity in smart factories becomes crucial to balance innovation and efficiency with robust security and privacy preservation measures. In response to these challenges, this doctoral research contributes innovative solutions that address the security and privacy vulnerabilities inherent in the Industry 5.0 scenario.

The first contribution of this doctoral research revolves around federated learning methodology (FL) for malware detection based on network analysis. This contribution introduced a cost-effective and efficient approach to deep-learning-based malware detection using FL methodology. This methodology addresses computational overhead and privacy concerns by leveraging network traffic data balancing emerging technologies with security and privacy to mitigate large-scale malware attacks that could undermine Industry 5.0's core principles.

The second contribution puts forth a novel privacy-preserving secure framework called PPSS, integrating blockchain with energy-efficient Proof-of-Federated Deep Learning (PoFDL) consensus protocol to optimize the process of FL in terms of preserving data privacy, enhancing system reliability, and promoting transparency. PPSS adeptly tackles the challenges associated with cyber threat detection and data privacy, specifically within the context of resource-constrained and heterogeneous industrial systems.

The third contribution focuses on developing an efficient, robust, federated cyber threat detection framework for Industrial IoTs. The approach leverages federated learning and generative adversarial networks (GANs) to enhance IDS efficiency, privacy protection, and resilience against adversarial attacks. A federated generative model was employed for data augmentation to limit the attack surface, thereby improving cyber threat detection reliability in the face of zero-day and adversarial threats.

The performance evaluation of the proposed approaches was conducted using a new cyber security dataset named Edge-IIoTset. Specifically designed for cyber threat detection in Industrial IoTs. The results showcase the efficiency and reliability of cyber threat detection under various data distribution modes.

Combining the insights from these contributions, this thesis proposes a comprehensive approach to safeguard Industry 5.0 from cybersecurity threats. Federated deep learning techniques optimize the process of knowledge sharing among participants while protecting data privacy in a resource-efficient manner. Integrating blockchain-enabled intrusion detection systems ensures the integrity and security of data exchanged among IoT-based devices. Deploying generative adversarial networks fortifies the system's resilience against zero-day and adversarial attacks.

Keywords : *Cybersecurity, Industrial Internet of Things, Blockchain, Federated Learning, Privacy-Preserving, Intrusion Detection System, Cyber Threat Detection*

ملـــخص

يسلط العصر التحولي الذي نشهده في صناعة الجيل الخامس 5.0 الضوء على مفهوم المصانع الذكية مع التركيز على التعاون بين البشر والآلات من أجل الاستدامة. هذا التكامل يشجع على التفكير النقدي والتخصيص والقدرة على التكيف، مع الاستفادة من تكنولوجيا إنترنت الأشياء والذكاء الصناعي لتحقيق الكفاءة وزيادة الإنتاجية. ومع ذلك، تقدم هذه الفترة أيضًا منظرًا معقدًا للتهديدات السيبرانية. مع اتصال الآلات والأنظمة والبشر، يصبح ضمان الأمن السيبراني في المصانع الذكية ضروريًا لتحقيق توازن بين الابتكار والكفاءة مع تطبيق إجراءات أمان قوية وحفظ الخصوصية. تساهم هذه الأطروحة من خلال حلول مبتكرة تتعامل مع قضايا الأمان والخصوصية المترتبة على منظومة صناعة 5.0.

المساهمة الأولى تدور حول منهجية جديدة لأنظمة الكشف عن التسلل، باستخدام التعلم العميق المشترك (FDL)لاكتشاف البرمجيات الخبيثة. هذه المساهمة قدمت نهجًا فعالًا من حيث التكلفة والكفاءة لاكتشاف البرمجيات الخبيثة بناءً على التعلم العميق. تعالج هذه المنهجية العبء الحسابي ومخاوف الخصوصية، محققة توازنًا بين التقنيات الناشئة والأمان والخصوصية للحد من هجمات البرمجيات الخبيثة التي تهدد مبادئ صناعة 5.0.

المساهمة الثانية تقدم إطارًا آمنًا مبتكرًا للحفاظ على الخصوصية يُسمىPPSS ، حيث يتم دمج التكنولوجيا السلسلة Blockchain لتحسين عملية FDL من حيث الحفاظ على خصوصية البيانات وزيادة موثوقية النظام وتعزيز الشفافية. كما يعالج PPSS التحديات المتعلقة بالكشف عن تهديدات السيبراني والخصوصية للبيانات، وبشكل خاص ضمن سياق أنظمة الصناعة المحدودة الموارد والمتنوعة.

المساهمة الثالثة تركز على تطوير إطار حماية فعّال وقوي لاكتشاف تهديدات السيبراني لأنظمة الإنترنت الصناعية. تستفيد هذه الطريقة من عملية FDL وشبكات المولدة (GANs) لتعزيز كفاءة حماية الخصوصية، وزيادة التحمل ضد الهجمات الالكترونية. في هذا الإطار تم استخدام نموذج توليدي موحد لزيادة البيانات للحد من احتمالية الهجمات الغير معروفة، مما يعزز موثوقية الكشف عن التهديدات السيبرانية في وجه التهديدات الجديدة.

تم إجراء تقييم الأداء للطرق المقترحة باستخدام مجموعة بيانات أمنية جديدة والتي صممت خصيصًا لاكتشاف التهديدات السيبرانية في أنظمة الإنترنت الصناعية. تظهر النتائج كفاءة وموثوقية FDL في الكشف عن التهديدات السيبرانية تحت تحديات متنوعة. من خلال الأفكار المستفادة من هذه المساهمات، تقترح هذه الأطروحة نهجًا شاملًا لحماية الصناعة 5.0 من تهديدات الأمان السيبراني. تقنيات FDL تحسن عملية مشاركة المعرفة بين المستخدمين مع الحفاظ على خصوصية البيانات بطريقة فعّالة من حيث الموارد. دمج أنظمة الكشف عن التسلل التي تعتمد على التكنولوجيا السلسلة يضمن نزاهة وأمان البيانات المتبادلة بين أجهزة الإنترنت الصناعية. بالإضافة إلى ذلك، نشر الشبكات المولدة يعزز متانة النظام ضد التهديدات الجديدة والهجمات المعادية.

Résumé :

La dernière révolution industrielle 5.0 met l'accent sur le concept d'usines intelligentes, en mettant l'accent sur la collaboration entre les êtres humains et les machines pour la durabilité. Cette intégration favorise la réflexion critique, la personnalisation et l'adaptabilité, tout en tirant parti de technologies telles que l'IoT et l'IA pour l'efficacité et la productivité. Cependant, cette ère introduit également un paysage complexe de menaces cybernétiques. À mesure que les machines, les systèmes et les êtres humains se connectent, garantir la cybersécurité dans les usines intelligentes est crucial pour équilibrer l'innovation et l'efficacité avec des mesures de sécurité robustes et la préservation de la vie privée. Cette recherche doctorale propose des solutions novatrices pour aborder les problèmes de sécurité de la vie privée dans le paysage de l'Industrie 5.0.

Cette recherche doctorale se concentre sur trois principales contributions : la première concerne l'apprentissage profond (FL) pour la détection de logiciels malveillants basée sur l'analyse réseau, la deuxième porte sur un nouveau cadre sécurisé appelé PPSS, qui intègre la blockchain avec le protocole Proof-of-Federated Deep Learning (PoFDL) pour optimiser les processus FL tout en préservant la confidentialité des données, en améliorant la fiabilité du système et en favorisant la transparence. PPSS aborde les défis liés à la détection des menaces cybernétiques et à la confidentialité des données, en particulier dans les systèmes industriels hétérogènes et gourmands en ressources.

La troisième contribution se concentre sur le développement d'un cadre de détection des menaces cybernétiques fédéré, efficace et robuste, spécifiquement conçu pour les objets connectés industriels (IIoT). L'approche exploite l'apprentissage fédéré et les réseaux génératifs (GANs) pour améliorer l'efficacité des systèmes de détection des intrusions (IDS), la protection de la vie privée, et la résilience contre les attaques adverses. Un modèle génératif fédéré a été utilisé pour l'augmentation des données afin de limiter la surface d'attaque, améliorant ainsi la fiabilité de la détection des menaces cybernétiques face aux menaces zero-day et adverses.

Les performances de ces approches ont été évaluées à l'aide d'Edge-IIoTset, un nouvel ensemble de données conçu spécifiquement pour la détection de menaces IoT. Les résultats démontrent l'efficacité et la fiabilité de ces approches dans divers scénarios de distribution de données. En combinant ces enseignements, cette recherche propose une approche globale pour protéger l'Industrie 5.0 contre les menaces de cybersécurité.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

<AI> Artificial Intelligence

<ARP> Address Resolution Protocol

<CSV> Comma-Separated Values

<DL> Deep Learning

<DNN> Deep Neural Network

<DNS> Domain Name System

<DP-SGD> Differentially Private Stochastic Gradient Descent

<FDL> Federated Deep Learning

<FedID> Federated Intrusion Detection

<FL> Federated Learning

<FN> False Negative

<FP> False Positive

<GANs> Generative Adversarial Networks

<HTTP> Hypertext Transfer Protocol

<ICMP> Internet Control Message Protocol

<ICS> Industrial Control System Institute

<IID> Independent and Identically Distributed

<IIoT> Industrial Internet of Things

<IoT> Internet of Things

<IP> Internet Protocol

<ML> Machine Learning

<Modbus/TCP> Modbus over TCP/IP

<MQTT> Message Queuing Telemetry Transport

<NFV> Network Function Virtualization

<Non-IID> Non-Independent and Identically Distributed

<P2P> Peer-to-Peer

<PBFT> Practical Byzantine Fault Tolerance

<PoA> Proof of Authority

<PoFDL> Proof-of-Federated Deep-Learning

<PoL> Proof of Learning

<PoW> Proof of Work

<PPSS> Privacy-Preserving Secure Scheme

<PPSS> Privacy-Preserving Secure Scheme

<ReLU> Rectified Linear Unit

<SC> Smart Contracts

<t-SNE> t-distributed Stochastic Neighbour Embedding

<TCP> Transmission Control Protocol

<TEEs> Trusted Execution Environments

<TN> True Negative

<TP> True Positive

<UDP> User Datagram Protocol

<VMs> Virtual Machines

<WGAN> Wasserstein Generative Adversarial Network

CHAPTER 1
INTRODUCTION

"Security is a process, not a product. Products provide some protection, but the only way to effectively do business in an insecure world is to put processes in place that recognize the inherent insecurity in the products"

— Bruce Schneier

INDUSTRY 5.0, the latest phase of the industrial revolution, represents a significant shift in the manufacturing landscape that emphasizes Human-Machine collaboration and sustainability. It builds upon the Smart Factory concept, introducing technologies like Cloud computing, the Internet of Things (IoT), Artificial Intelligence (AI), and Big Data analytics, and further complements these advances by facilitating human intervention when necessary. It leverages critical thinking, personalization, and adaptability to enhance efficiency and productivity.

However, Industry 5.0 also brings about a complex landscape of security challenges. The interconnected nature of machines, systems, and humans, combined with extensive data exchange, opens the door to a range of cyber threats and privacy intrusions [1]. In response to these challenges, researchers are developing new cybersecurity strategies to protect privacy and secure industrial networks and control systems from large-scale cyber threats.

Federated learning (FL) has recently emerged as a decentralized and privacy-preserving computing paradigm, offering a viable solution to mitigate security and privacy risks in IIoT environments. FL facilitates the local training of machine-learning-based (ML) and deep-learning-based (DL) detection models on edge devices, wherein only model updates are shared for global optimization, sparing the transmission of raw sensitive data [2]. By adopting this approach, the privacy of sensitive information is upheld, fostering a secure environment for data processing.

Cross-silo federated learning, an extension of the FL paradigm, further advances the capabilities of IIoT systems by enabling different industrial organizations to exchange intrusion events, incident logs collaboratively, and reported alert data about cyber attacks. The participating entities share knowledge and insights through transfer learning without compromising data privacy, bolstering the collective defense against cyber threats.

Despite the promising advantages of FL and cross-silo FL, the security landscape remains dynamic and challenging. Adversarial attacks, including data poisoning and inference attacks, have demonstrated the potential to exploit vulnerabilities in IIoT systems, posing significant threats to the integrity and reliability of model updates. Additionally, trace information within model updates may inadvertently disclose private and sensitive data, necessitating robust audit gateways and enhanced security measures to thwart potential leaks.

This doctoral thesis aims to delve into the intricacies of the security and privacy challenges that impede the widespread adoption of IIoT technologies. By investigating the potential of federated learning and cross-silo federated learning, the research seeks to develop novel and practical strategies to enhance the security, privacy, and reliability of IIoT systems. Through empirical evaluations and rigorous experimentation, this research endeavors to contribute to the advancement of secure and privacy-preserving IIoT frameworks, fostering a resilient and trustworthy smart industry ecosystem.

In the subsequent chapters, we will explore our proposed approaches' theoretical foundations, methodology, and implementation details. We will subsequently present comprehensive analyses of experimental results, leading to valuable insights and practical recommendations for industry stakeholders, cybersecurity professionals, and researchers. By fortifying the foundations of IIoT with robust security measures, we strive to accelerate the realization of the full potential of the smart industry, heralding a new era of optimized productivity, reliability, and service quality.

1.1 Research Questions and Objectives:

Our research study concentrates on implementing an efficient and effective security monitoring mechanism, Intrusion Detection Systems (IDS), to safeguard Industry 5.0 against emerging cyber threats. To achieve this objective, we will comprehensively analyze the vulnerabilities and architectural characteristics of Industrial Internet of Things (IoT) networks.

This analysis will serve as the foundation for developing IDS solutions that are reliable, robust, and tailored to the constraints inherent in Industrial IoT environments. In particular, Table 1.1 outlines the specific research questions identified to address those goals. These research questions will guide our investigation and contribute to developing advanced IDS solutions for Industry 5.0, strengthening its cybersecurity posture and enhancing its resilience against evolving cyber threats.

1.2 Research Methodology

We have adopted the Systematic Literature Review approach (SLR) to identify relevant literature about our research interests. The primary objective is investigating cyber security solutions for IDS implementation in Industrial IoT. The research methodology involved identifying, selecting, and evaluating proposed and related studies. Specific research questions were formulated, and Scopus academic search

Research Questions	Objectives
RQ1: What are Industrial IoT networks' vulnerabilities and architectural characteristics in Industry 5.0?	To explore potential weaknesses and design flaws that may pose security risks in IIoT infrastructures deployed in smart industry settings. Understanding these vulnerabilities and architectural characteristics is crucial for developing effective security measures and intrusion detection strategies.
RQ2: What are the systems architecture and the technology type used by IDSs to secure IIoT networks and their various components?	To investigate and analyze the different systems architectures and technology types employed by IDSs specifically tailored for securing IIoT networks and their constituent components. Subsequently, we aim to provide insights into these systems' key design principles and implementation strategies.
RQ3: What are the used IDS detection methodologies for IIoT?	To identify and assess the various IDS detection methodologies specifically designed and applied for IIoT environments. By assessing these methodologies' efficacy, strengths, and limitations, we aim to gain a thorough understanding of their capabilities in detecting and mitigating cyber threats in IIoT networks.
RQ4: How are emerging technologies consolidated for effective and secure detection?	To explore integrating and consolidating emerging technologies, such as ML and DL, Cloud/fog services, Big data analytics, and Edge intelligence, to devise efficient and secure detection mechanisms tailored to IIoT systems. We enhance privacy preservation and the overall resilience and reliability of IIoT security
RQ5: What are the used IDS evaluation performance and the experimental datasets?	To investigate and evaluate the performance metrics commonly employed to assess the effectiveness of IDSs in IIoT settings. We aim to explore the experimental datasets utilized for comprehensive testing and validation of IDS functionalities in realistic IIoT scenarios.

TABLE 1.1: Research Questions and Objectives.

queries were conducted in the "Title," "Keywords," and "Abstract" fields of relevant publications. The search results were confined to publications from 2015 to 2021. The study selection process was refined by focusing on practical studies that align with the research questions described in Table 1.1. By leveraging the SLR methodology, we aim to contribute to the understanding and advancement of cyber security measures

within the Industrial Internet of Things.

Furthermore, we formulate our research strategy using the PICO framework. we ensure a structured and systematic approach to address the complex aspects of IDS implementation in IIoT. We identify the PICO research question as follows :

- Population (P): Our research focuses on IDS-based cyber threat detection in Industrial IoT.
- Intervention (I): We consider all the proposed works of IDS within the domain of IIoT.
- Comparison (C): Our investigation compares various IDS methods based on criteria and factors outlined in related studies and proposed solutions.
- Outcomes (O): The primary objectives of our study are to establish requirements, address challenges, and propose evaluation mechanisms for IDS-based solutions to enhance the security of IIoT. These findings will serve as valuable contributions to further research in this area.

This framework allows us to guide our investigation and enable valuable insights into developing and improving IDS solutions for securing IIoT environments.

1.3 Main Contributions

The main contributions of this thesis are summarised as follows:

1. A systematic review of IDS-based cyber threat detection for Industrial IoT was conducted, encompassing a comprehensive examination of deployment strategies, detection approaches, methodologies, and data sources employed for evaluation. The findings of this review highlight significant insights for the field. Furthermore, a critical analysis of well-selected literature reveals future directions and challenges that must be carefully navigated when designing robust IDS solutions to enhance the security of IoT-enabled critical infrastructure within

industrial sectors. This contribution was presented at the 2021 International Conference on Theoretical and Applicative Aspects of Computer Science (ICTAACS 2021) [1].

2. A federated learning methodology for Android malware detection based on network analysis was proposed. This contribution introduced the federated learning (FL) paradigm as a cost-effective deep-learning-based malware detection using network traffic data. The aim is to overcome the computational overhead and privacy concerns of conventional malware detection strategies while maintaining the efficiency of detecting large-scale malware attacks. The performance of our methodology was evaluated using the benchmark dataset AAGM-2017 across various FL settings, and its outcomes were compared against those of centralized training methods. The results demonstrate the efficiency and effectiveness of Android malware detection in terms of detection accuracy and computation cost while providing data privacy without any significant adverse effects on the classification performance compared to conventional centralized approaches. This contribution was published as a chapter in Springer's Cyber Malware [3].

3. A two-stage intrusion detection framework for IoT security was proposed. This contribution introduced a dual-detector approach. An adversarial training strategy was used as a robust optimization approach against the emergent adversarial threats in the initial stage that employs the first detector. Subsequently, a DL model was employed for the second detector, focused on intrusion identification. The performance evaluation of this framework is conducted using the recently published Edge-IIoTset dataset, we conducted evaluations in terms of detection accuracy and resilience against adversarial attacks. The experimental results underscore the proposed methodology's effectiveness in detecting intrusions and persistent adversarial examples. This contribution was presented at the 2023 International Conference [4].

4. A Privacy-Preserving Secure Framework (PPSS) using Blockchain-enabled Federated Deep Learning for Industrial IoT. The framework introduces a blockchain-based scheme designed to enhance the security of cross-organization Federated Learning (FL), ensuring the process remains secure while minimizing adverse effects on learning performance. A novel lightweight and energy-efficient proof of learning, PoFDL, is proposed for effective model validation and storage. Additionally, integrating differential privacy training enhances the privacy protection of model updates. The performance evaluation of the PPSS framework is conducted using the recently published Edge-IIoTset dataset, employing convolutional neural networks (CNNs) as deep networks across various FL settings. The experimental results demonstrate clear evidence of the framework's efficiency and effectiveness. Notably, the proposed framework's capabilities are shown in handling heterogeneous datasets and addressing non-IID data distribution. Moreover, the framework's robustness against common blockchain attacks, including Byzantine attacks, Sybil attacks, and honest-but-curious attacks, is thoroughly assessed to ensure security and reliability. This contribution was published in Elsevier's Pervasive and Mobile Computing [5].

5. A distributed learning paradigm has been proposed leveraging FL and generative adversarial networks (GANs). The aim is to improve privacy protection, facilitate effective training, and enable robust detection of large-scale cyber threats and emergent adversarial attacks. This contribution introduces a three-model framework incorporating Wasserstein-Conditional-GANs for data augmentation and a DL-classifier for cyber threat classification. First, a distributed deep generative model was trained on highly imbalanced and non-IID distributed data under the FL paradigm. This model generates qualified and diverse synthetic data. Subsequently, this augmented data undergoes validation using our proposed data curation method before being employed to train a federated learning classifier. This process enhances resilience and enables efficient detection of novel cyber threats not initially in the training data.

The performance evaluation of this framework is conducted using the recently published Edge-IIoTset dataset. The evaluations encompass detection efficiency against recent state-of-the-art adversarial attacks and zero-day cyber threats. Furthermore, we assess the effectiveness of incorporating differential privacy training as an additional technique for improved privacy preservation and its impact on model performance. The experimental results demonstrate the validity and diversity (multi-class) of the augmented data generated using the distributed generative model. Additionally, the results highlight enhanced cost-effectiveness when utilizing the proposed data augmentation approach in contrast to implementing DP training, particularly regarding privacy preservation. This contribution was published in Elsevier's Internet of Things [6].

1.4 List of Publications

Journal papers

- **Hamouda, D.**, Ferrag, M. A., Benhamida, N., & Seridi, H. (2022). PPSS: A privacy-preserving secure framework using blockchain-enabled federated deep learning for Industrial IoT. Pervasive and Mobile Computing, 88, 101738. `https://doi.org/10.1016/j.pmcj.2022.101738`

- **Hamouda, D.**, Ferrag, M. A., Nadjette, B., Hamid, S & Ghanem, M. C. (2024). Revolutionizing intrusion detection in industrial IoT with distributed learning and deep generative techniques. Internet of Things, 1-15. `https://doi.org/10.1016/j.iot.2024.101149`

Conference papers

- **Hamouda, D.**, Ferrag, M. A., Benhamida, N., & Seridi, H (2021, November), Android Malware detection based on network analysis and deep convolutional

neural network. The 4th International Hybrid conference on Informatics and Applied Mathematics (IAM'21).

- **Hamouda, D.**, Ferrag, M. A., Benhamida, N., & Seridi, H. (2021, December). Intrusion detection systems for industrial internet of things: a survey. In 2021 International Conference on Theoretical and Applicative Aspects of Computer Science (ICTAACS) (pp. 1-8). IEEE. https://doi.org/10.1109/ICTAACS53298.2021.9715177

- **Hamouda, D.**, Ferrag, M. A., Benhamida, N., & Seridi, H (2022, November), Network-based Intrusion Detection Using Generative Adversarial Networks. The 5th International Hybrid Conference on Informatics and Applied Mathematics (IAM'22).

- M. A. Ferrag, **D. Hamouda**, M. Debbah, L. Maglaras and A. Lakas, "Generative Adversarial Networks-Driven Cyber Threat Intelligence Detection Framework for Securing Internet of Things," 2023 19th International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT), Pafos, Cyprus, 2023, pp. 196-200, https://doi.org/10.1109/DCOSS-IoT58021.2023.00042.

Book Chapter

- **Hamouda, D**., Ferrag, M.A., Benhamida, N., Kouahla, Z.E., Seridi, H. (2024). Android Malware Detection Based on Network Analysis and Federated Learning. In: Almomani, I., Maglaras, L.A., Ferrag, M.A., Ayres, N. (eds) Cyber Malware. Security Informatics and Law Enforcement. Springer, Cham. https://doi.org/10.1007/978-3-031-34969-0_2

Co-authored papers

- Ferrag, M. A., Friha, O., **Hamouda, D**., Maglaras, L., & Janicke, H. (2022). Edge-IIoTset: A new comprehensive, realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning. IEEE Access, 10, 40281-40306. https://doi.org/10.1109/ACCESS.2022.3165809

- Ferrag, M. A., Shu, L., **Djallel, H**., & Choo, K. K. R. (2021). Deep learning-based intrusion detection for distributed denial of service attack in agriculture 4.0. Electronics, 10(11), 1257. https://doi.org/10.3390/electronics10111257

- Ferrag, M. A., Friha, O., Kantarci, B., Tihanyi, N., Cordeiro, L., Debbah, M., **Hamouda, D**.,... & Choo, K. K. R. (2023). Edge Learning for 6G-enabled Internet of Things: A Comprehensive Survey of Vulnerabilities, Datasets, and Defenses. in IEEE Communications Surveys & Tutorials. https://doi.org/doi:10.1109/COMST.2023.3317242.

1.5 Thesis Organisation

The remaining parts of this thesis are organized as follows: Chapter 2 on page 12, explores an IDS-oriented security solution for IoT-enabled critical industrial infrastructure. It examines its architecture, vulnerability to threat models, and security requirements. The chapter reviews adaptive IDS implementations, deployment strategies, machine learning techniques, and blockchain technologies for privacy-preserving and secure IDS. The study emphasizes the necessity for efficient and sophisticated privacy-preserving IDS systems comparable to centralized approaches while addressing challenges and security requirements.

In Chapter 3 on page 38, an innovative federated learning (FL) paradigm and network behavior analysis for malware detection are proposed. The focus is on preserving privacy, minimizing computation costs, and enhancing detection efficiency. The

chapter explores malware detection using network layer features. It presents an efficient detection methodology using FL with a CNN approach and compared with conventional centralized methods, highlighting advantages regarding computation cost and privacy protection.

In Chapter 4 on page 50, an innovative and privacy-preserving secure framework named PPSS is proposed. This chapter explores the development and experimental aspects of the PPSS framework. Topics covered include component interaction, blockchain-enabled federated learning, secure communication, key management, proof of federated deep learning, and blockchain security analysis. The chapter also discusses how PPSS enables cyber threat detection, considering various scenarios and experimental settings, including data distribution, global model accuracy, convergence time, differential privacy training, energy costs, and blockchain performance.

In chapter 5 on page 85, an improved federated generative framework named FedGen-ID is proposed. This framework addresses imbalanced and private data challenges by employing distributed data augmentation techniques. It aims to enhance efficiency and robustness against cyber threats. The chapter discussed using data augmentation methods to support a synthetically enhanced federated learning scheme, leading to improved detection efficiency and resilience against zero-day attacks. Three models are discussed: one refines local Critics to strengthen resilience, the second focuses on improving cybersecurity, and the third serves as a cyber threat classifier.

In conclusion, Chapter 6 on page 113 summarizes the key findings from this research and presents recommendations for future work.

CHAPTER 2
CONCEPTS AND LITERATURE REVIEW

"The art of war teaches us to rely not on the likelihood of the enemy's not coming, but on our own readiness to receive him; not on the chance of his not attacking, but rather on the fact that we have made our position unassailable"
— Sun Tzu, The art of war

Introduction

The above quotes resonate with the essence of our research endeavor, emphasizing the paramount importance of preparedness and resilience in the face of potential threats. In the realm of cybersecurity, this principle becomes ever more relevant as we navigate the dynamic landscape of Industry 5.0 and the integration of technologies such as the Internet of Things (IoT), cloud/fog computing, artificial intelligence (AI), and collaborative robotics to boost productivity and business. The industrial landscape has evolved into heightened interconnectivity and increased complexity. However, the evolution of this landscape has heightened its vulnerability to cyber intrusions, mainly due to the inherent security challenges embedded in the development of sophisticated technologies and their increased connectivity and exposure to public networks. Furthermore, the lack of worldwide-adopted technical standards

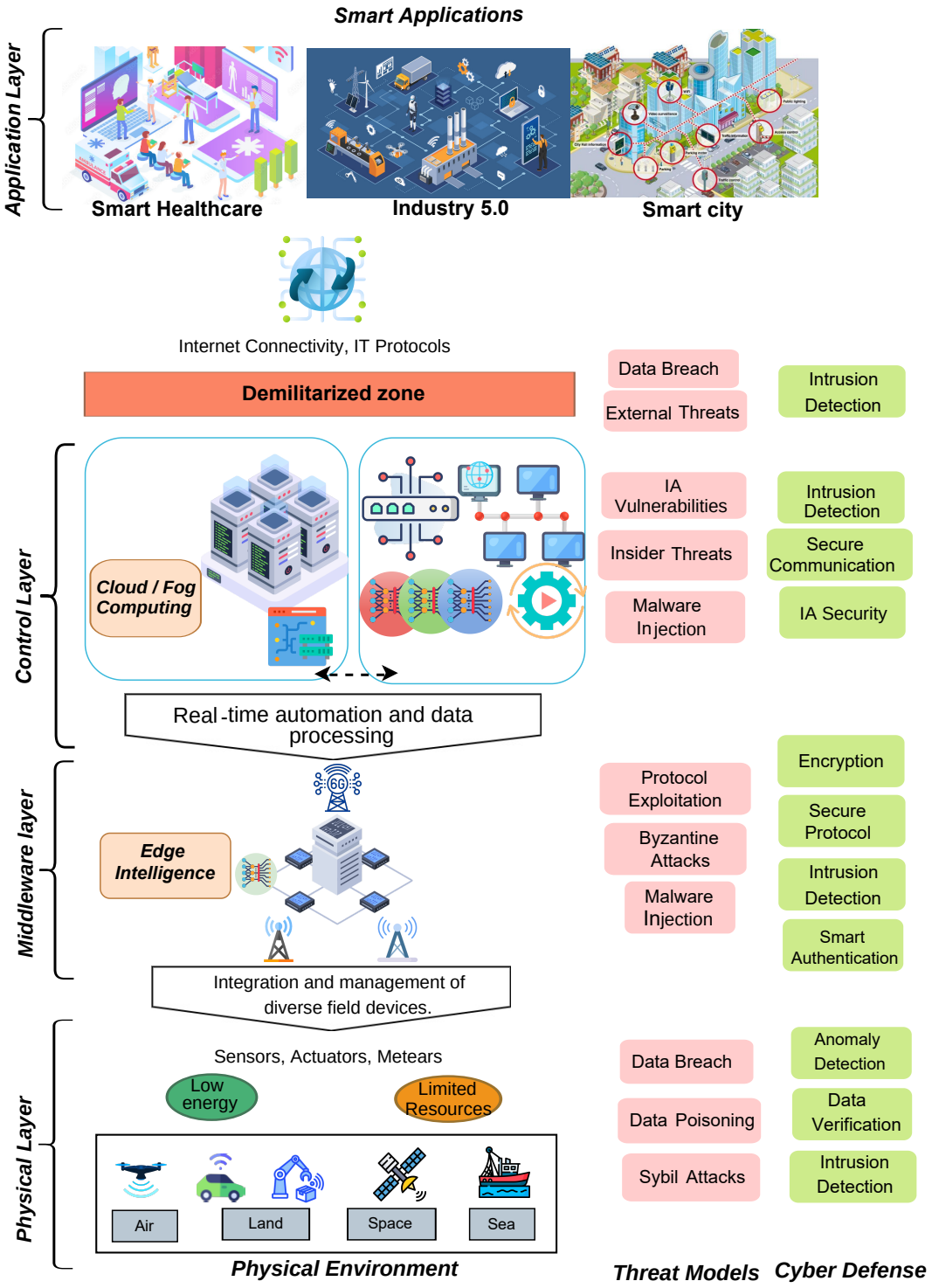


FIGURE 2.1: The Industrial IoT network model: Layers, Threats, and Defense Strategies.

for IIoT security and interoperability expands the sensitivity of this ecosystem to cyber security risks [7]. More than ever, cybersecurity breaches pose significant threats to enterprises across varying scales and sectors. The escalation of cybercrime has dramatically increased and has made businesses much more likely to suffer from financial and reputational damage due to cyber attacks, with damage related to cybercrime projected to hit $10 trillion annually by 2025 [8].

The role of cyber security in safeguarding Industry 5.0, including those in smart factory technologies and beyond, is crucial for ensuring confidentiality, integrity, and protection of shared information among interconnected components. To this end, security professionals and researchers recommend using and developing a proficient Intrusion Detection System (IDS) solution. This system serves as a robust security monitoring mechanism to identify ongoing potential security threats and safeguard industrial networks and control systems [1]. However, the prevalence of information and operational technologies in Industry 4.0 and 5.0 has changed the appearance of cyber threats and how we deal with them, as it also requires addressing challenges related to reliability, complexity, security, and data privacy [9].

In the following section, we explore the gap between the development of traditional IDSs and the design of adequate IDS schemes for the unique challenges of Industrial IoT ecosystems. Our study comprehensively addresses this deficiency, examines the most pertinent and effective detection strategies, and sheds light on the challenges and requirements of securing Industry 5.0 from emerging cyber threats. We intend to provide a comprehensive foundation for developing robust and future-proof intrusion detection solutions, fostering the continued growth and security of Industry 5.0.

2.1 Industrial IoT Architecture, Threat Models, and Security Requirements

In this section, we aim to explore potential weaknesses and design flaws that may pose security risks in Industrial IoT infrastructures deployed in the smart industry. Understanding these vulnerabilities and architectural characteristics is crucial for developing effective security measures and detection strategies.

The architectural framework of Industrial IoT architectures differs slightly from the conventional IoT and Cyber-Physical Systems (CPS) systems with additional critical control systems and security challenges. A typical IIoT architecture can be illustrated by hierarchical layers of various networking technologies and communication protocols that establish interconnections between hardware devices, control software, and end-users. Figure 2.1 illustrates the network model, including layers, threat models, and defensive mechanisms within the context of industrial IoT.

- **Physical layer:** This layer is designed to collect data about the physical environment or to act on it, using Sensors, actuators, and meters. Devices of these layers are usually resource-constrained. At this stage, communication protocols and technologies were designed to operate at limited bandwidth, constrained CPU and memory capacity, and low energy consumption [10].

- **Middle-ware layer:** This segment manages field devices to facilitate the integration and communication between physical objects and supervisory control systems. This layer includes applications like Programmable Logic Controller (PLC), Remote Terminal Unit (RTU), and Intelligent Electronic Device (IED). It also contains limited computation resources with heterogeneous communications infrastructures, including wired and wireless connections that interconnect objects with control systems.

- **Control layer:** Designed to manage automation and intelligent control of the industrial infrastructure. It enables real-time processing of data collected from substations system control of the previous layer. This layer includes control systems such as Supervisory Control and Data Acquisition (SCADA), Distributed Control System (DCS), HMI, and other applications such as data historian, Manufacturing Execution Systems (MES), and Enterprise Resource Planning (ERP).

- **DMZ Zone:** Contains critical devices that must be exposed to the outer network, such as an App server, web server, etc. At this stage, internet connectivity and standard IT protocols interconnect OT with IT and users across longer distances.

- **Application layer:** Includes processing and management tools that require costly computation and storage resources. Application at this layer is mainly based on cloud services and used to process the collected data to obtain valuable insights and information about the physical environment. Using AI approaches, these applications may make or reach decisions based on this information to control physical objects.

This architectural composition of IIoT, using diverse networking technologies and communication protocols, presents significant challenges for IDS and exacerbates the complexity of detection. Incorporating distinctive insecure-by-design protocols and various communication infrastructures, such as wireless networks, complicate cyber threat detection [11]. Furthermore, improving the security level while simultaneously ensuring the availability of IIoT systems faces essential challenges due to resource restrictions [12].

2.1.1 Industrial IoT Threat Models

Figure 2.1 and Table 2.1 highlight the various ways in which IIoT networks can be compromised, emphasizing the need for robust security measures and continuous monitoring to mitigate these risks.

Threat Model	Description
Data Poisoning	This involves injecting malicious or inaccurate data into the IoT system, leading to incorrect decisions or actions by connected devices [13].
Sybil Attacks	In this type of attack, an attacker creates multiple fake identities to overwhelm the system or gain unauthorized access to IoT devices [14].
Data Privacy Intrusion	IoT devices often collect and transmit sensitive data. Attackers may attempt to intercept or access this data, violating users' privacy [13].
Malware Injections	Attackers can inject malware into IoT devices, compromising their functionality and potentially using them for malicious purposes [3].
Byzantine Attacks	These attacks involve compromised or malicious nodes within a network that intentionally provide conflicting information, leading to system failures or incorrect decision-making [14].
Protocol Exploitation	IoT devices communicate using various protocols. Attackers can exploit vulnerabilities in these protocols to gain unauthorized access or manipulate device behavior [10].
AI Associated Threats	As AI is integrated into IoT devices, attackers could target vulnerabilities in AI algorithms to manipulate or disrupt device behavior and decision-making [15].
External Threats	External attackers can target IoT devices by exploiting vulnerabilities in the devices' software, firmware, or communication channels.

TABLE 2.1: Threat Models Against IoT Network Architecture.

In light of these multiple issues, it becomes more evident that a comprehensive strategy comprising improved security monitoring, proactive threat detection, and resource-efficient defensive mechanisms is important for adaptable IDS security solutions in IIoT environments.

2.2 Adaptive Intrusion Detection System (IDS) for Securing Industrial IoT

Cyber threat detection plays a pivotal role in addressing substantial security requisites within the context of complex technological landscapes of IIoT environments. An IDS is a crucial component in this endeavor by actively monitoring system operations and network activities, investigating data patterns, and identifying anomalous behaviors that could potentially signify malicious or unauthorized activities [16]. Recently, machine learning (ML) and deep learning (DL) have emerged as a recent advancement within the field of IDS, providing the means to identify novel effective, and continually evolving forms of cyber attacks [17].

However, given the heterogeneous and distributed nature of data sources in conjunction with the inherent resource limitations pertaining to storage, energy, and computational capabilities of end-point IIoT devices, it becomes imperative to meticulously incorporate considerations of resource utilization efficiency during the design and implementation of IDS security mechanisms [18]. Several studies have been conducted to tackle the deployment of IDS across diverse components within the IIoT. This includes the examination of communication protocols [19] and the inclusion of Infrastructure Control Systems (ICS) sectors [20]. Furthermore, the application of IDS has expanded to encompass pivotal sectors like transportation [21], critical infrastructures such as gas pipelines [22], and sophisticated domains like smart grids [23].

To gain a comprehensive insight into the distinctions between traditional IDS-based security systems implemented for information systems and the envisaged IDS systems tailored for Industrial IoT deployments, we present a comprehensive IDS taxonomy founded upon detailed categorizations 2.2.

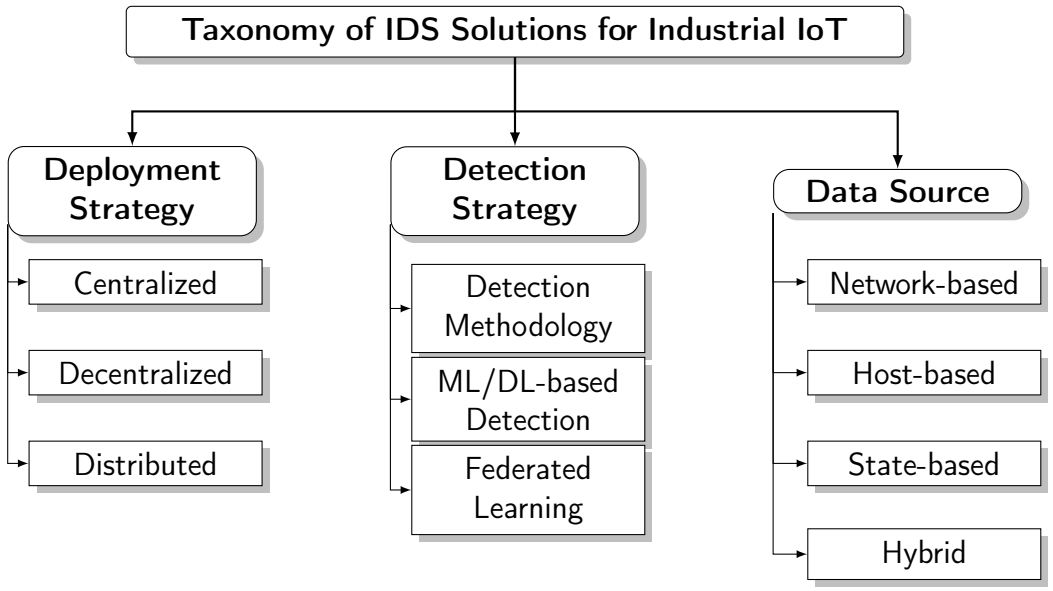


FIGURE 2.2: Taxonomy of IDS Solutions for Industrial IoT.

2.2.1 Taxonomy of IDS Deployment Strategies in IIoT Environments

Implementing an IDS necessitates considering several aspects, including system architecture, deployment strategy, monitoring methodologies, and detection strategies [24]. Within the realm of Industry5.0, the convergence of IoT solutions within the frameworks of both Cloud and fog paradigms highlights the potential deployment strategies for IDS architectures. Moreover, the success of ML and DL techniques has significantly contributed to enhancing the capabilities of IDS systems. Driven by these advancements, researchers have advanced proficient IDS models tailored for the IoT domain. Figure 2.2 presents a taxonomy of IDS implementation for Industrial IoT.

- **IDS Deployment Strategy :** It largely depends on locations and data flow dynamics within the IIoT framework. Depending on whether data aggregation is centralized or distributed, the IDS can be established within the same node responsible for data collection or be distributed across multiple nodes to cover a broader span of the network landscape effectively. This deployment facilitates the monitoring and analyzing IIoT components, thereby fortifying plant networks against diverse cyber threats [1].

  - **Centralized/Decentralized IDS** : Aydogan et al. [19], studied the effectiveness of a centralized IDS in promptly detecting attacks, owing to its comprehensive data aggregation, outperforming those of individual IDS agent nodes. Leveraging Cloud and Fog computing, a centralized approach could be adept and overcome the constraints posed by the limited resources of the IIoT. Within this context, both data aggregation and the execution of resource-intensive processing are hosted by Cloud services [25]. However, addressing concerns about data privacy in this environment is important. Another study proposed by Ioannou et al. [26], demonstrated that decentralized IDS offers the distinct advantage of operating as a fault-tolerant system. Moreover, deploying multiple detection models within the heterogeneous landscape of the IIoT holds the promise of effectively identifying large-scale attacks while mitigating data privacy concerns [26, 27].
  - **Distributed IDS**: integrating IDS across multiple agent nodes, collectively contributing to global decision-making based on gathered data. Leveraging Edge Computing, distributed IDS offers benefits such as reduced observed data volume and energy-efficient task execution. For instance, Zhang et al. [28] propose a multi-layer data-driven IDS approach, expanding attack detection coverage. Khan et al. [22] introduce a multi-level anomaly detection strategy with distinct detection methods at each level. Shu et al. [21] demonstrate the efficacy of distributed IDS using both Independent Identically Distributed (IID) and non-IID data sources.

- **IDS Detection Strategy:** The chosen detection strategy is vital for enhancing IDS performance and robustness. This involves a critical selection not only of the overall detection methodology—ranging from anomaly-based, signature-based, to hybrid systems—but also the effective inclusion of Machine Learning (ML) and Deep Learning (DL) approaches to ensure efficient and real-time detection capabilities. Moreover, the evolving landscape of IDS training paradigms

introduces the concept of federated learning in a new era of collaborative and decentralized model refinement over distributed IoT networks.

- **Detection Methodology :** This includes distinct approaches. Signature-based detection involves storing known attack signatures to quickly identify and validate future attacks, offering real-time and cost-effective detection. However, it falls short against unknown and polymorphic attacks, necessitating ongoing updates, human intervention, and secure connections. Anomaly-based detection relies on User Behavior Analysis (UBA) to create a dynamic detection model based on software, hardware, or human interactions. While efficient and self-adaptive for identifying unfamiliar attacks, it tends to produce more false alarms and requires increased computational resources [1].

  Specification-based detection establishes legitimate behavior models through protocol or system analysis, detecting deviations from specifications without a training phase [29]. Although effective in spotting attacks, it falters against attacks conforming to the specification model. Combining these methodologies yields higher accuracy, lower false alarms, and real-time detection. For instance, Otoum et al [30]. proposed a hybrid IDS framework using IoT gateways, integrating signature-based and anomaly-based approaches for enhanced effectiveness. Feng and Chana [31] presented a comparable method, augmenting their IDS with a baseline signature database for time-series anomaly detection, thereby bolstering system efficiency.

- **ML and DL-based detection:** ML algorithm techniques have proven effective in safeguarding IIoT networks and their physical entities [22, 25, 28]. The application of these algorithms is particularly well-suited to the context of IIoT due to its inherently task-oriented nature and the consistency of data distributions. These attributes serve to enhance both traffic predictability and the efficiency of intrusion detection [1]. In this context, DL

encompassed as a subset of ML, manifests as a collection of sophisticated ensembles of operations that proficiently acquire multi-layered representations [17]. However, the distinctive potential of DL-based IDS comes upfront when confronted with enormous quantities of training data. Various DL approaches are adeptly equipped to manage and counteract the diverse spectrum of intrusions and cyber-attacks. This encompasses varying degrees of intricacy, complexity, and distribution levels [32].

Although the application of ML-based and DL-based detection approaches has shown success in enhancing the security of IIoT through IDS, it is important to recognize certain limitations that require careful consideration. Both ML and DL models are sensitive to slight changes in data, which can significantly decrease detection and attack classification performance. The lack of interpretability and the limited transparency of decision-making processes present challenges in understanding the origins of attacks and conducting further forensic analyses. Additionally, the computational demands for data processing and learning exceed the capacities of available IIoT resources [1].

Equally important are adversarial attacks aimed at undermining the effectiveness of model learning, resulting in the evasion of the detection of malicious activities. This emerging challenge holds significance for both ML-based and DL-based IDS, [4]. Addressing these complexities is crucial for establishing strong and dependable security measures within IIoT environments.

– **Federated Learning-based IDS (FL-IDS)**: In light of the challenges above, FL is proposed as a promising training approach for ML and DL-based IDS for IIoT [27]. Its distributed and privacy-preserving approach aligns well with the characteristics of IIoT environments, offering a pathway to improved detection accuracy, data privacy, and resource efficiency [33]. More about this paradigm is in the following sections.

- **IDS Data Source :** Input data are essential characteristics to detect large-scale attacks effectively. Various dimensions offer insights into their sources, characteristics, and analytic potential.

  - **Data source**: Includes two principles; Network-Based Data and Host-Based Data. Industrial IoT also deploys physical information known as state-based IDS. Generally, a hybrid approach is often employed to achieve comprehensive detection results in time [34]. An example of hybrid-based IDS is proposed by Zhang et al. [28] to robustly detect intrusions that may not be detectable by monitoring network and host system data, such as command tampering and false data injection attacks by an insider in ICSs. Zhou et al. [34] proposed multiple data models to represent the general knowledge of Industrial Process Control Systems (PCS) to facilitate the implementation of hybrid anomaly-based IDS.
  - **Data Granularity :** Refers to the level of detail at which data is collected, processed, or stored in an information system. It could be raw packet-level data or aggregated flow-level data, or any equivalent level of data aggregation [35].

2.3 Privacy Preserving Intrusion Detection in Industrial IoT Network

An IDS security system aims to safeguard against security breaches by analyzing monitored data and detecting potential cyber threats. However, its implementation introduces a challenge to users' privacy, leading to the need for sophisticated IDS mechanisms that prioritize privacy preservation. This development has roots in earlier research, such as Park et al. [36], who employed cryptographic techniques to enhance the security of log files. In contemporary times, characterized by the deployment of ML and DL in various industries, major privacy concerns have been raised

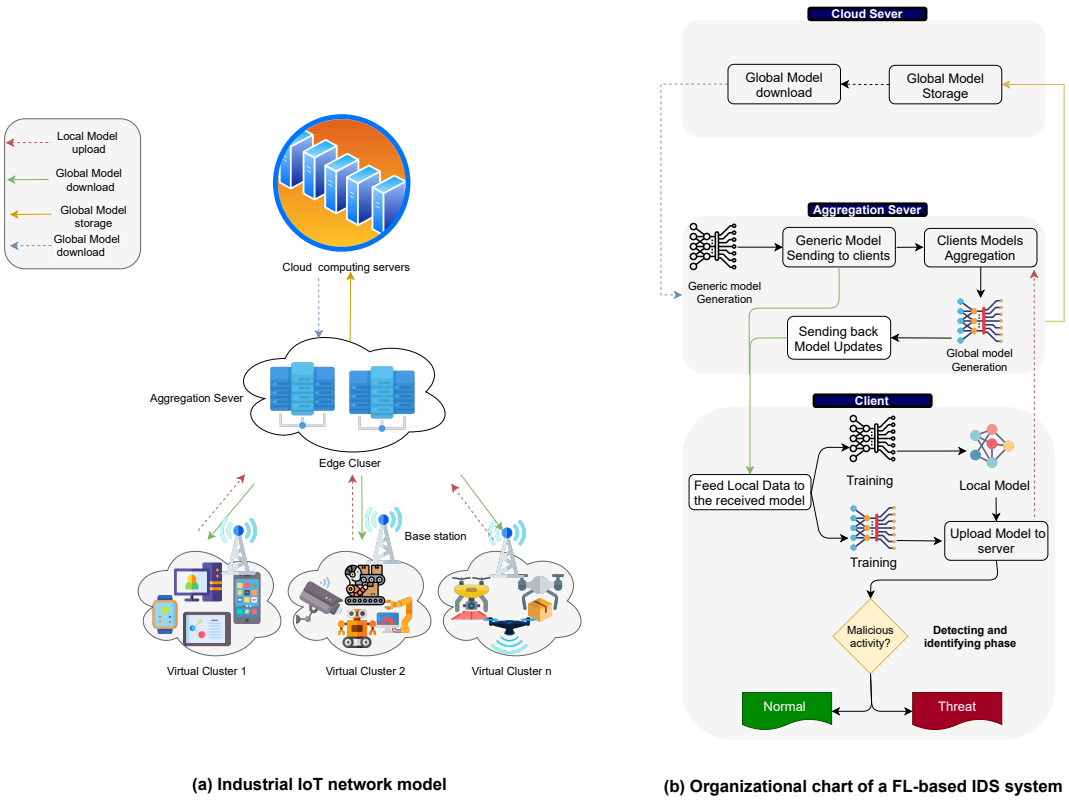
where the handling of sensitive proprietary data is a prominent issue. This concern extends to the domain of IDS development within the context of IIoT security, which consequently stimulated a demand for innovative conceptual frameworks that accommodate privacy preservation and security.

2.3.1 Federated Learning-based IDS

Federated Learning (FL) introduces an innovative way to collaboratively learn and distribute computations for applications based on ML and DL. Instead of sending client data to a central server, FL sends models from the server to specific clients. These clients then train the models using their local data and conventional ML methods. This approach ensures privacy and security by keeping data on the client's side [37].

In safeguarding Industrial IoT, an FL-based IDS emerges as a dependable security strategy. It addresses the security needs and challenges of IIoT by enabling decentralized decision-making for IDS across diverse IIoT setups. To visualize, Figure 2.3 depicts an overview of the Industrial IoT network model and the organizational structure of an FL-based IDS system. Algorithm 1 outlines the core process of the FL-based IDS for securing IIoT environments. This algorithm enables collaborative learning and distributed computations across client devices while maintaining data privacy. As demonstrated, the workflow started with the Server component initializing the model and orchestrating the FL process over several rounds. Each round randomly selects a subset of clients from the total client pool. These clients then participate in parallel computations to update their local models. The algorithm aggregates these client model updates to refine the global model at the server.

On the client side, represented by the Client (i.e., device) component, each client operates independently. They split their local dataset into batches and perform local training epochs on their data batches. These local model updates are communicated back to the server for aggregation. Thus ensuring data privacy preservation.



(a) Industrial IoT network model **(b) Organizational chart of a FL-based IDS system**

FIGURE 2.3: The Industrial IoT Network Model with an Organizational Chart of an FL-Based IDS System.

In a recent study [21], Shu et al. developed a collaborative IDS for Vehicular Adhoc Networks (VANETs) within a distributed SDN environment, utilizing multiple SDN controllers to train a single IDS model. The unique aspect is that they achieve this without directly sharing their sub-network data flows. Another approach, proposed by Fan et al. [38] revolves around an FL-based IDS framework and combining cloud and edge computing services to maintain privacy and coordinate the FL process. In a different study, Nguyen et al. [39] introduced an FL-based IDS, where a distinct detection model is developed for each IoT device, with security gateways building local models using unlabeled crowd-sourced traffic.

Although the FL-based IDS training paradigm ensures privacy-preserving and knowledge sharing and boosts efficient cyber threat detection that works well even with limited and dispersed IIoT data sources, specific challenges have emerged with its adoption. These include communication costs in extensive distribution, selecting

Algorithm 1: Federated Learning-based Intrusion Detection [37].

Server (K : *Number of Selected Clients*, C : *Total Clients*, R : *Total Rounds*)
  Initialize $model_1$
  for $t = 1, .., R$ **do**
    $S_t \leftarrow$ Randomly select K clients from C
    Parallel.for $k \in S_t$ **do**
      $model_{t+1}^{k} \leftarrow Client(model_t, k)$
    end
    $model_{t+1} \leftarrow \frac{1}{K} \sum_{k=1}^{K} model_{t+1}^{k}$
  end
Client (i.e., device) (m : *Model*, k : *Client ID*)
  Split the local dataset $\mathbb{D}$ into $\mathbb{B}$ local data batches
  $\mathcal{B} \leftarrow$ Split($\mathbb{D}$, $\mathbb{B}$)
  for $i = 1, .., E$: *Local epochs* **do**
    for $b \in \mathcal{B}$ **do**
      $m \leftarrow m - \eta \nabla f_c(m, b)$
    end
  end
  Send m to the Server

suitable equipment for federation, uneven distribution of data and resources, ensuring the security of FL audit gateways, and addressing issues related to adversarial attacks [1].

2.3.2 Blockchain based IDS

Adopting blockchain technology can make intrusion detection systems in IoT more secure. It offers a safe and decentralized way to store and share intrusion detection data, helping quickly identify and prevent manipulated data and injection attacks [40]. It brings several key features. Firstly, it establishes a decentralized network where data is stored across multiple nodes, eliminating central control and enhancing security against attacks. Secondly, information recorded on a blockchain is immutable, preventing alterations and safeguarding IDS data from manipulation. Thirdly, transparency is fostered, allowing all network nodes to access the same data,

and facilitating easier attack detection [41]. Additionally, using smart contracts automates the process of IDS, minimizing errors [42]. Furthermore, device authentication ensures that only authorized IoT devices access the network, reducing attack risks. Lastly, blockchain promotes collaboration among nodes, enabling shared data and cooperative defense strategies, thereby improving threat identification and response capabilities [43].

By combining blockchain and federated learning, a hybrid approach offers a secure and effective solution for adaptive IDS in IIoT. For instance, Kumar et al. [44] proposed an intelligent blockchain framework that integrates smart contracts for data authentication and FL-based IDS to mitigate data poisoning attacks. Similarly, Wang et al. [45] designed a blockchain-enabled decentralized FL to alleviate data falsification issues and reduce communication costs between cloud and edge devices. The PEFL framework uses two-level privacy-preserving modules: perturbation-based privacy, and DL-based intrusion detection.

Although this hybrid strategy establishes a decentralized network, ensuring privacy protection and secure FL automation, the integration of blockchain with different aspects and settings of federated learning, particularly in resource-constrained IIoT environments, remains a challenge.

2.3.3 Comprehensive Analysis Framework for Privacy-Preserving IDS

The interconnection of IoT-enabled industrial infrastructure using Cloud and Edge paradigms and ML and big data analytics illustrates how a security framework can be deployed to ensure secure data transmission and maintain privacy between Industry 5.0 components.

Several privacy-preserving and secure frameworks have recently been proposed for various Industry 5.0 applications. Table 2.2 provides an overview of these proposals to advance privacy-preserving IDS.

Main Idea	Challenges	Domain	Pros	Cons	Cite*
A combination of secure aggregation and differential privacy techniques ensures the protection of data privacy while preserving data utility	Sophisticated privacy threats and data utility	DL application	Enhanced privacy protection	Additional computational overhead, and decrease in accuracy	[46]
A blockchain scheme that uses smart contracts to securely aggregate participants' local model updates.	Privacy-preserving aggregation of model updates, contribution evaluation and reward mechanisms in FL	DL application	Secure communication and Secure multi-party computation	Additional computational overhead, and not suitable for all FL scenarios	[47]
Federated GAN training involves the integration of a least squares loss function to mitigate mode collapse issues	High-quality and diversified data augmentation	Renewable energy	Produce realistic and diverse data while preserving the privacy of the data.	Training stability and convergence issues, insecure communication	[48]
Protecting the confidentiality of sensitive data on active learning by using FL with homomorphic encryption property	Protecting data privacy while preserving data utility	Active learning application	Securely preventing gradient leakage during FL while preserving model accuracy.	computational overhead, scalability concerns, and privacy and data utility trade-offs.	[49]
GANs training within the FL framework	non-IID clients	Computer vision	Improved performance of FL	Training stability and convergence issues, insecure communication	[50]
Blockchain and FL integration involve clients uploading model updates, workers creating valid blocks, and a trusted committee verifying the aggregated model through an evolving verification contract over training iterations.	Ensuring the integrity and security of FL	Computer vision	Enhanced security and trustworthiness of FL	increased computational complexity, Potential security threats against the blockchain network	[51]
Generating synthetic data using FL and GAN while ensuring differential privacy.	Data storage and improved privacy protection	Computer vision	Efficient GAN training and enhanced privacy preservation	Assume that the data is i.i.d., but this assumption may not hold in real-world situations.	[52]
A decentralized approach using blockchain to store and share data among the Edge nodes and local FL training on these data.	Maintaining the accuracy and security of healthcare data	Healthcare	Secure and efficient data sharing	Computational overhead, potential sophisticated privacy threats against blockchain network	[53]
Blockchain and FL to improve the accuracy and precision of data mining while ensuring information privacy and security	Centralized server, security and privacy threats	Railway industry	Incentive mechanism for participating devices, reliability, and system robustness.	Potential sophisticated privacy threats against the blockchain network	[54]

TABLE 2.2: Overview of Privacy-Preserving and Secure Frameworks in Other Domains.

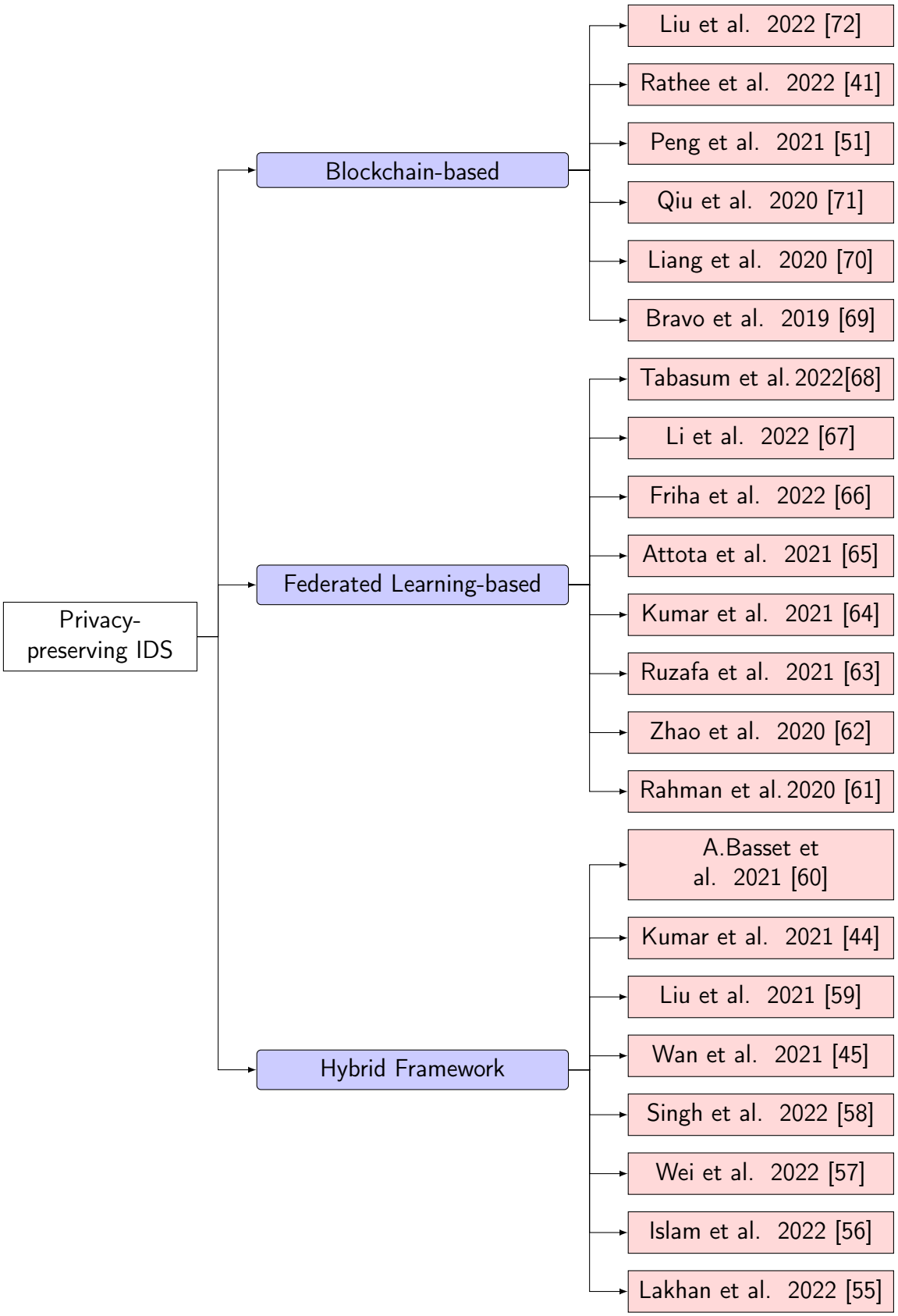


FIGURE 2.4: Privacy preserving IDS for Industrial IoT.

Main Idea	Challenges	Privacy Technique	Pros	Cons	Dataset	Cite*
FDL using federated averaging aggregation after several communication rounds for botnet attack detection	classification performance, and communication efficiency	FL	Outperform localized and Distributed DL methods in memory and communication efficiency	Computation overhead (long training time)	Bot-IoT, N-BaIoT	[73]
An ensemble multi-view FL approach that trains on multiple views of IoT network data, using a combination of local and global models	Data heterogeneity, resource constraints	FL	Improved accuracy, privacy-preserving distributed learning	Insecure communication and potential for model poisoning attacks	MQTT	[65]
Distributing a GAN network across IoT devices to function as a classifier and training it using locally augmented data	Data heterogeneity, non-IID data and privacy concerns	FL + GAN	Improved model convergence and accuracy	Data reliability, insecure communication, and communication overhead.	KDD99, NSL_KDD, UNSW-NB15	[68]
Robust FL using a GAN approach to monitor the global model aggregation for detecting Android malware applications in IIoT	Dynamic poisoning attacks against FL, Integrity and reliability of the FL	FL + GAN	Improved accuracy and data privacy	Data reliability, insecure communication, and communication overhead.	Drebin, Genome, Contagio (Android malware)	[74]
FDL by deploying a deep privacy-encoding mechanism that perturbs the data before sending it to the server.	Privacy threats, and data heterogeneity	FL + Data Perturbation	Enhanced privacy	Balancing between data utility and privacy	ToN-IoT	[64]
A new type of poisoning attack manipulates the global model using GAN and assesses its effectiveness against existing defense mechanisms	Poisoning attacks against FL	FL	Novel poisoning attacks that can be manipulated	Difficult to implement in practice	N/A	[75]
Homomorphic encryption encrypts the IDS alerts and performs clustering on the encrypted data without revealing the original data.	Inspect the content of encrypted traffic, Computational overhead	Homomorphic encryption	Perform clustering on encrypted data	Not be suitable for real-time detection	N/A	[76]
Deploy blockchain to enable secure and decentralized data sharing and management in smart transportation systems	data privacy, reliability, and security in a decentralized system. Communication and computation limitations	Blockchain + FL	Improved scalability and flexibility	Complexity of the system, potential vulnerabilities within blockchain participants	Ton-IoT	[60]
Combining FL and fraud-enabled blockchain, providing data provenance and permission control of the participants to enhance the security and privacy of parameters in FL	Security and scalability of the FL system	FL + Blockchain	Provides data provenance and permission control of FL participants	Require significant computational resources and may not be feasible for small-scale healthcare organizations	N/A	[55]
A decentralized and differentially private FL-based IDS	One point of failure, privacy threats	FL + Deferential privacy	Secure communication and improved privacy	Decreased accuracy with higher privacy regimes	EdgeIIoTset	[66]

TABLE 2.3: Privacy-Preserving IDS Overview.

This overview offers numerous advantages, including transferring privacy-preserving techniques, insights into distinct privacy concerns across domains, scalability solutions, enhanced robustness and security measures, and optimizing resources. By leveraging knowledge and techniques from diverse domains, we can develop more effective and comprehensive privacy-preserving IDS systems capable of addressing the specific privacy challenges inherent to intrusion detection.

Figure 2.4 illustrates a taxonomy of recent advancements in privacy-preserving IDS frameworks, categorized into three groups: blockchain-based IDS, federated learning-based IDS, and hybrid approaches. This taxonomy highlights the growing research on balancing threat detection with privacy protection.

Table 2.3 introduces a structured framework to explore privacy-preserving IDS within the IIoT context. It aims to provide a concise yet comprehensive overview of critical aspects of these systems, enabling scholars and researchers to systematically assess and compare various IDS implementations. The selection of columns covers essential dimensions of privacy-preserving IDS, clarifying core concepts, addressing challenges, enhancing privacy strategies, and assessing pros and cons.

2.4 Performance Analysis of Intrusion Detection System in Industrial IoT

Researcher commonly evaluate their proposed IDS solutions through validation strategies such as Hypothetical, Empirical, Simulation, or Theoretical methods, as detailed in [77]. A validation strategy ensures that the proposed IDS scheme suits its intended purpose and meets all requirements. This evaluation assesses whether the IDS detection strategy performs well according to predetermined objectives. The assessed works performed evaluation using Empirical and Simulation validation methods. In Empirical evaluation, real-world Industrial IoT (IIoT) data is used, while Simulation evaluation utilizes real IIoT network traces. Through the literature review, we synthesize the key performance metrics that illustrate the overall efficiency and effectiveness

of IDS for IIoT [1]. These metrics include [1]:

- **Accuracy**: This metric pertains to correctly identifying attacks and minimizing false alarms. Precision, Recall, True Positive Rate, False Positive Rate, and True Negative Rate are various Accuracy components.
- **Complexity**: This evaluates the resource expenditure (time, memory, energy, bandwidth) during IDS operations, including model learning and audit event processing. It measures real-time detection capabilities and the feasibility of implementation on resource-constrained devices. Complexity metrics are often omitted in proposed IIoT IDS solutions, hindering proper effectiveness and real-time potential assessment.
- **Completeness**: This indicator assesses the ability of an IDS to reliably and effectively detect known and unknown threats. In the context of IIoT, completeness is measured by an IDS's applicability to large-scale infrastructures and its capability to handle diverse data sources.
- **Scalability**: This indicates an IDS's ability to maintain detection effectiveness as the number of different behaviors grows due to IIoT advancements. Adaptive and self-learning IDSs autonomously generate and store information or profiles of previously encountered events, applying them to future detection scenarios.

2.4.1 Evaluation Datasets for IDS in Industrial IoT Networks

The imperative necessity of substantial online or offline voluminous datasets for the rigorous evaluation and credibility of AI-driven IDS remains indisputable. The scarcity of authentic, real-world data emanating from Industrial IoT contexts, primarily attributable to concerns regarding privacy, has notably catalyzed a proactive response from the research community. This proactive response materialized through endeavors and efforts to provide practically oriented industrial datasets that accurately capture the complexities of real industrial scenarios. To this end, specific datasets have

been simulated using various testbed setups incorporating relevant data from Industrial Internet of Things (IIoT) components. These simulations are conducted to assess and appraise ML-oriented IDS within IIoT. Examples of such datasets are N-BaIoT, SWaT, TON_IoT, and EdgeIIoTSet datasets.

However, it is noteworthy that conventional network traffic IDS datasets, including NSL-KDD, UNSW-NB15, and CICIDS2017, remain relevant within the IIoT landscape. In this context, these traditional datasets often serve as a collection that demonstrates data heterogeneity and the complex behaviors of various cyber-attacks or demonstrates the efficiency and effectiveness of particular ML-based detection approaches that can be used in resource-constrained IIoT. Table. 2.4 describes the commonly used datasets to validate IDS-based cyber security in IIoT.

2.5 Research Gaps

Drawing insights from a thorough review of relevant literature, designing a cost-effective yet efficient detection methodology, including factors such as detection rate and decision latency, stands as an open research issue within the domain of IDS-based security solutions for Industrial IoT. Table 2.5 lists research gaps related to IDS deployment in Industrial IoT, grouped by key qualities. These gaps are crucial for advancing IDS security in IIoT environments. Our thesis proposal has three key contributions to address the challenges above and open issues. The first contribution, detailed in Chapter 3 on page 38, presents a cost-effective and efficient IDS approach to detect malware network attacks targeting industrial Android systems. We have addressed computation efficiency and data privacy in this context by leveraging the FL training framework.

The second contribution, detailed in Chapter 4 on page 50, presents a novel privacy-preserving secure framework that incorporates blockchain-enabled federated learning. Within this context, we have addressed the detection of large-scale cyber-attacks

Dataset*	Year	Description	Limitations
NSL-KDD [78]	2009	An improved version of KDD 99. Used to evaluate detection efficiency against huge network data but not specific to IIoT	lacks real-world IoT traffic
UNSW-NB15 [79]	2015	Benchmark dataset. Provided by the Australian cybersecurity lab, contains real-world normal and attack traffic scenarios for NIDS evaluation	lacks real-world IoT traffic
SWaT [80]	2016	Collected from a water treatment testbed, contains time series traffic data	Small size, Limited scope
CICIDS2017 [81]	2017	Proposed by the Canadian Institute for Cybersecurity. Contains network traffic flow with the most common attacks	lacks real-world IoT traffic
TON_IoT [82]	2020	Collected from heterogeneous data sources: Telemetry datasets of IoT services, Windows and Linux Operating systems, Network traffic datasets	Limited features representation, Lacks Industrial IoT data.
N-BaIoT [83]	2018	Collected from a simulated IoT environment to capture several normal and botnet events	Limited threat model. Lacks Industrial IoT data.
Bot-IoT [84]	2019	Comprises legitimate and malicious traffic from IoT devices, including botnets on IoT networks.	Lacks Industrial IoT data.
MQTTset [85]	2020	Utilizes MQTT protocol traffic and various attack streams related to IoT devices.	Limited to only MQTT traffic.
X-IIoTID [86]	2021	Encompasses connectivity and device-agnostic data in the context of ML/DL-based IDS for both IoT and Industrial IoT.	Convenient for centralized learning
WUSTL-IIOT-2021 [87]	2021	Created using legitimate and malicious data generated by various IIoT and industrial devices to mimic an actual industrial application.	lacks real-world IoT traffic. Limited attack data.

TABLE 2.4: Datasets Used for IDS Cybersecurity Evaluation in IoT/IIoT.

key quality	**Related challenges and open issues**
Data Sources	Industrial data are large-scale and heterogeneous, stemming from diverse origins, networking technologies, and communication protocols, pose significant issues. Addressing imbalanced and Non-Identically Distributed Data (Non-IID) within this context remains unexplored research. Also, handling big data requires expensive processing methods impacting IDS performance dynamics. Conversely, some industrial scenarios lack the data volume needed for effective anomaly-based IDS. Lastly, the scarcity of authentic IIoT datasets and suitable testbeds casts doubt on the credibility of proposed IDS frameworks.
Detection Methodology	Employing behavioral analysis for threat detection in the context of IIoT pose a significant challenge. There are numerous cases where normal behavior occurs infrequently, and it's crucial to accurately differentiate between transient faults and potential threats or anomalies. Anomaly-based detection approaches frequently struggle with rising or diminished sensitivity, particularly when confronted with a growing range of diverse behavioral patterns.
System Deployment	The limited resources in IIoT environments constrain the availability of resources for implementing efficient IDS solutions.
Performance	The vulnerability stemming from inadequately secured IIoT communication protocols introduces an element of unpredictability to the spectrum of cyber threats while concurrently escalating the prevalence of false positive instances. Pursuing cost-effective IDS solutions inevitably impinges on the trade-off between accuracy and real-time detection, manifesting as a pivotal concern for IIoT security.
Security risks associated with IDS	Source data must be protected during acquisition and subsequent processing by IDS nodes. Privacy-preserving techniques such as differential privacy have adverse effects on detection performance. Adversarial attacks, such as data poisoning and evasion attacks, undermine IDS performance

TABLE 2.5: Research gaps for IDS Deployment in Industrial IoT.

in resource-constrained and heterogeneous industrial systems without exposing data to privacy issues. Furthermore, we have investigated differential privacy-enhanced FL and related security issues by deploying a novel blockchain design scheme. Empirical validation of our framework employs a novel industrial IoT dataset (Edge-IIoT dataset) to demonstrate the efficiency and effectiveness of our framework in terms of detection accuracy, computation overhead, and energy cost. The results demonstrate that our proposed secure system can efficiently detect and identify industrial IIoT attacks with high classification performance even when subjected to distinct data distribution modes (namely, Independent and Non-Independent Identically Distributed).

Our third contribution, detailed in Chapter 5 on page 85, presents a further investigation into the efficiency and robustness of IDS-based security in the IIoT. Specifically, we proposed a novel Distributed Learning and Deep Generative Model-Based Intrusion Detection Technique. Within this paradigm, we addressed robust optimization against zero-days and adversarial attacks and the challenges related to imbalanced and highly non-IID distributed data. Furthermore, we investigated differential privacy-enhanced distributed learning against model performance degradation. Our empirical validation on the same dataset demonstrated improved efficiency and reliability against zero-day cyber threats.

These contributions consequently contribute to the continued growth and advancement of the IIoT security landscape. The outcomes of our research have the potential to enhance the overall efficiency and reliability of IDS-oriented security within the domain of IIoT.of critical industrial systems.

2.6 Chapter Summary

This chapter aims to comprehend the fundamental concept of an IDS-oriented security solution for IoT-enabled critical industrial infrastructure. The discourse unfolds by investigating the Industrial IoT architecture, its vulnerability to threat models, and

the associated security requirements. Subsequently, an in-depth review of adaptive IDS implementation within the context of Industrial IoT is conducted.

CHAPTER 3

FEDERATED LEARNING FOR ANDROID MALWARE DETECTION

"If you think technology can solve your security problems, then you don't understand the problems and you don't understand the technology "

— Bruce Schneier

3.1 Introduction

Android is a popular open-source operating system with extensive traction in industrial IoT deployments due to its ability to enhance convenience and operational efficiency [88, 89]. However, this widespread adoption has inadvertently rendered these systems attractive targets for cybercriminals. This heightened appeal arises from the fact that industrial systems house valuable assets and store sensitive information essential for the seamless functioning of operational technology. Consequently, malicious actors are increasingly drawn to the potential of planting their malicious apps to exploit vulnerabilities in Android systems, spread through networks, and conduct devastating cyber attacks and privacy intrusions over a large network of connected devices.

This chapter introduces a novel, privacy-preserving, cost-effective, and efficient approach to deep-learning-based malware detection, employing the emerging Federated Learning (FL) paradigm and network analysis. Specifically, we propose Federated Convolutional Neural Networks (FedCNN) to detect several types of malware based on abnormal network behavior. By leveraging FL and network traffic data, this methodology addresses computational overhead and privacy considerations, mitigating large-scale and sophisticated malware attacks that could undermine Industry 5.0's core principles.

The remainder of this chapter is organized as follows: Section 3.2 provides an overview of malware detection strategies. Section 3.3 details the development and experimentation of the proposed FedCNN model for malware detection, including data processing, training methodologies, and performance evaluation. Finally, Section 3.4 presents the results and discussion on the proposed model's efficacy in detecting Android malware, compared with conventional centralized methods in terms of computation cost and privacy protection.

3.2 Malware Detection Strategies

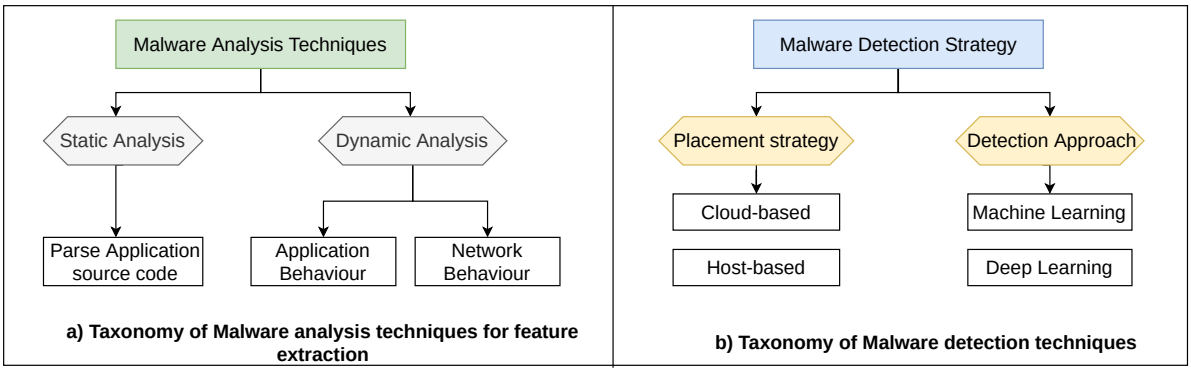


FIGURE 3.1: A Taxonomy of Malware Analysis Techniques and Detection Strategies.

Several studies have been conducted to detect malware, generally encompassing two key phases: malware analysis and detection, as demonstrated in Figure 3.1 [90].

The former entails techniques for analysis and processing to facilitate detection. Static analysis involves scrutinizing malware code without executing it, leveraging reverse engineering methods. However, these techniques have demonstrated efficacy against known established malware; they fall short against novel variants and can be easily evaded by obfuscation techniques. Dynamic analysis, on the other hand, entails observing and analyzing the runtime attributes of malware applications during code execution. This approach assesses behaviors to decipher malware functionality, including information flow tracking, function call monitoring, and instruction tracing [91]. Virtual environments and emulators are commonly employed for dynamic analysis and data collection. Although this methodology effectively identifies unknown malware, it is time-intensive and demands substantial computational resources. Dynamic analysis has also been extended to network traffic to identify malware that executes attacks via network pathways towards remote targets [92]. Network traffic traces can be detected by analyzing behavioral patterns in such cases.

Figure 3.1.b depicts malware detection strategies. This refers to the placement strategy and detection approach for detecting and identifying malware. The placement strategy determines whether the system is implemented on a host or in the cloud, thereby determining its efficiency against complex code variants while utilizing limited computational resources. Malware detection approaches describe the methods and algorithms employed to detect and identify malware. However, their efficiency relies on the availability of extensive and diverse datasets. Data privacy concerns and shortages pose significant challenges when deploying cloud-based and deep-learning-based security solutions.

Several studies on Android malware detection have been proposed and discussed [92, 93, 94, 95, 96], encompassing a range of ML and DL approaches and utilizing various malware analysis techniques and corresponding features. However, the discourse has not extensively explored the use of DL for malware detection, explicitly exploiting the predictability of network behavior. Moreover, several additional constraints have been identified but are not commonly addressed in these discussions,

such as limitations in computing resources, insufficient training data availability, and privacy concerns [3].

3.3 Model Development and Experiments

Our approach involves three main steps: selecting and processing relevant network data, training Federated Convolutional Neural Networks (FedCNN) to detect malware, and evaluating the results of this approach considering various performance metrics and settings.

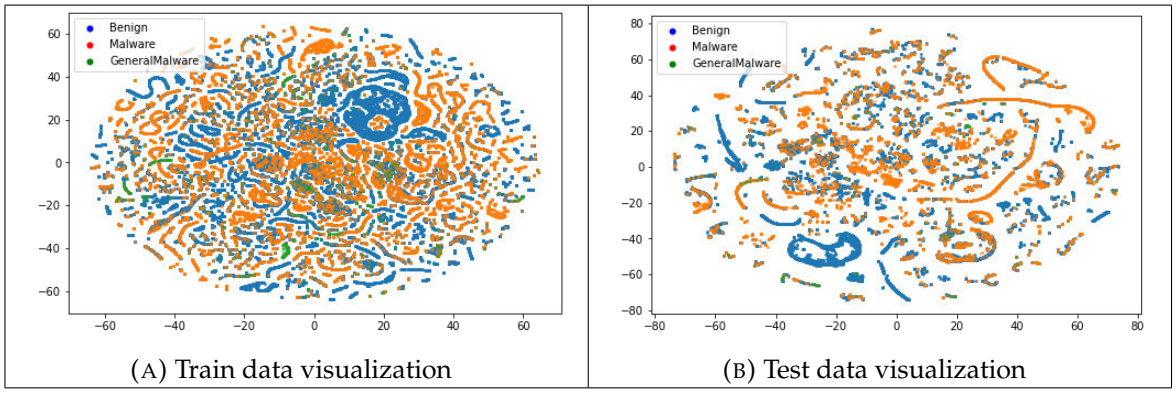
3.3.1 Dataset Selection and Processing :

Deep-learning-based malware detection relies significantly on the quantity and quality of training data. Increased availability of high-quality data leads to higher accuracy and improved results. For this study, we opted for the AAGM dataset (Android Adware and General Malware), renowned for its diverse collection of malware samples [97]. This dataset encompasses 1500 benign app samples and 400 malware samples categorized into 10 families, comprising 5 adware and 5 general malware families. To capture significant network traffic behavior, the authors deployed these samples on actual smartphones and executed user-interaction scenarios. The dataset includes 471,597 instances of benign behavior and 160,358 instances of malware behavior, accompanied by 80 network traffic features encompassing flow-based, time-based, and packet-based attributes. These features were employed to differentiate Android malware behavior from benign applications.

An essential step before training involves exploratory analysis and data preprocessing on the selected dataset to address various issues. Initially, we eliminated five null features, namely 'flow_urg', 'furg_cnt,' 'burg_cnt,' 'flow_ece,' 'flow_cwr,' and recognized their potential adverse impact on model performance. Additionally, four nearly null features were removed: 'bAvgBulkRate,' 'bAvgBytesPerBulk,' 'bAvgPacketsPerBulk,' and 'std_idle.' Subsequently, we pruned redundant instances and

those with missing values. Following this, the data underwent normalization. The dataset was then divided using the hold-out validation strategy; the dataset was then divided, allocating 80% for training and 20% for testing. In the Federated Learning (FDL) context, a noteworthy portion (80%) of the training data was further distributed to participating clients.

Figure 3.2 illustrates the dataset class distribution after the preprocessing step, utilizing the t-SNE technique [98]. The t-SNE technique is paramount in visualizing high-dimensional data in lower-dimensional space. It is particularly useful for understanding the underlying structure and relationships within complex datasets, as it aims to preserve the pairwise similarity between data points during the dimensionality reduction process. By applying t-SNE, we can gain insights into how the preprocessing steps have impacted the dataset's distribution and separability of classes.



(A) Train data visualization (B) Test data visualization

FIGURE 3.2: Exploring the High-Dimensional AAGM2017 Dataset Using the t-SNE Technique [98].

3.3.2 FedCNN for Malware Detection

Our detection methodology employed Convolutional Neural Networks (CNNs), a significant and specialized deep learning approach for data processing. These networks use a unique architecture that consists of multiple convolutional layers strategically crafted to extract essential spatial features that are crucial for accurate decision-making by the model. These features are pivotal in enabling the model to make well-informed decisions. A notable aspect of CNNs is their composition, which involves a sequence of convolutional layers using a mathematical operation called convolution. This operation allows the network to capture intricate patterns within the data effectively. Additionally, the network encompasses processing perceptron layers adept at effectively managing extensive-scale malware attacks. Table 5.2 demonstrates our CNN model architecture.

Our FedCNN approach harnesses the capabilities of CNNs in conjunction with the decentralized nature of Federated Learning (FL), enabling collaborative model training across distributed data sources while ensuring privacy protection. To address this, we formulate the FL optimization problem as follows:

- **Device Sampling :** This involves selecting participating Devices from a distributed network, each with its own private local dataset. In this study, these datasets were derived by sampling from the training set of the main dataset. We ensured these sampled datasets were identically distributed (IID), preserving the same feature vector. Typically, the selection process ensures client diversity and representation, considering factors like data distribution, device capabilities, and connectivity.

- **Local Training :** After device sampling, we proceed with local model training using the selected clients and their corresponding resources, including data and computational capabilities. The objective is to update the parameters of individual models to minimize the local loss function associated with their data.

Mathematically, for each client k, the local training seeks to find the optimal model parameters $\boldsymbol{\theta}_k$ that minimize the local loss $L_k(\boldsymbol{\theta}_k)$:

$$\boldsymbol{\theta}_k^* = \arg\min_{\boldsymbol{\theta}_k} \mathcal{L}_k(\boldsymbol{\theta}_k) \tag{3.1}$$

$$= \arg\min_{\boldsymbol{\theta}_k} -\frac{1}{N_k} \sum_{i=1}^{N_k} [y_i \log(f(x_i; \boldsymbol{\theta}_k)) + (1 - y_i) \log(1 - f(x_i; \boldsymbol{\theta}_k))] \tag{3.2}$$

where: x_i denotes the input data sample, y_i is the associated true label (0 as Benign or 1 as Malware), $f(x_i; \boldsymbol{\theta}_k)$ is the model output with parameters $\boldsymbol{\theta}_k$ for input x_i, and N_k signifies the count of data samples within client k's local dataset.

During this phase, clients utilize their local data to update their models, capturing domain-specific patterns and information.

- **Model Aggregation :** The central server initiates the model aggregation step following the local training phase. The objective is to combine the knowledge from individual clients' models to create a global model that benefits from collective intelligence while preserving data privacy. This involves weighted averaging of the model parameters from the selected clients. The aggregation process can be mathematically expressed as:

$$\boldsymbol{\theta}^{\text{global}} = \sum_{k=1}^{K} \frac{N_k}{N} \cdot \boldsymbol{\theta}_k^*$$

Where overall loss across all clients' datasets can be expressed as:

$$\min_{\boldsymbol{\theta}} \sum_{k=1}^{K} \frac{N_k}{N} \cdot L_k(\boldsymbol{\theta})$$

Here, N_k represents the size of the dataset at client k, and N is the total number of samples across all clients. The resulting global model, $\boldsymbol{\theta}^{\text{global}}$, represents a consensus reached by aggregating the insights from diverse sources.

Figure 3.3 illustrates the organizational chart of our FDL-based Android malware detection method, which comprises the following steps:

1. The Server takes the lead by initializing the global model's architecture along with essential global parameters, including the learning rate, the local training epochs, and the local batch size

2. The server transmits this comprehensive information to pre-selected clients. These clients are chosen based on their resource availability and the presence of sufficient training data. The interaction between the server and the clients operates asynchronously.

3. Each client independently engages in multiple local training epochs using the provided model. Subsequently, the client computes updates specific to its dataset and training progress. These computed updates, representing the new model parameters, are then returned to the server.

4. Having gathered these updates, the server proceeds to update the global model. Once this update is completed, the cycle repeats, encompassing steps 2, 3, and 4; as an iterative process, the global model converges to an optimal state.

5. The server evaluates and maintains the final version of the global model for future optimized models intended for future deployment in malware detection. Based on its individual performance, each participating client independently preserves any relevant global model states throughout the FedCNN training process.

3.4 Results and Discussion

The efficacy of the presented FedCNN approach for Android malware detection was systematically evaluated within the controlled environment of Google Colaboratory,

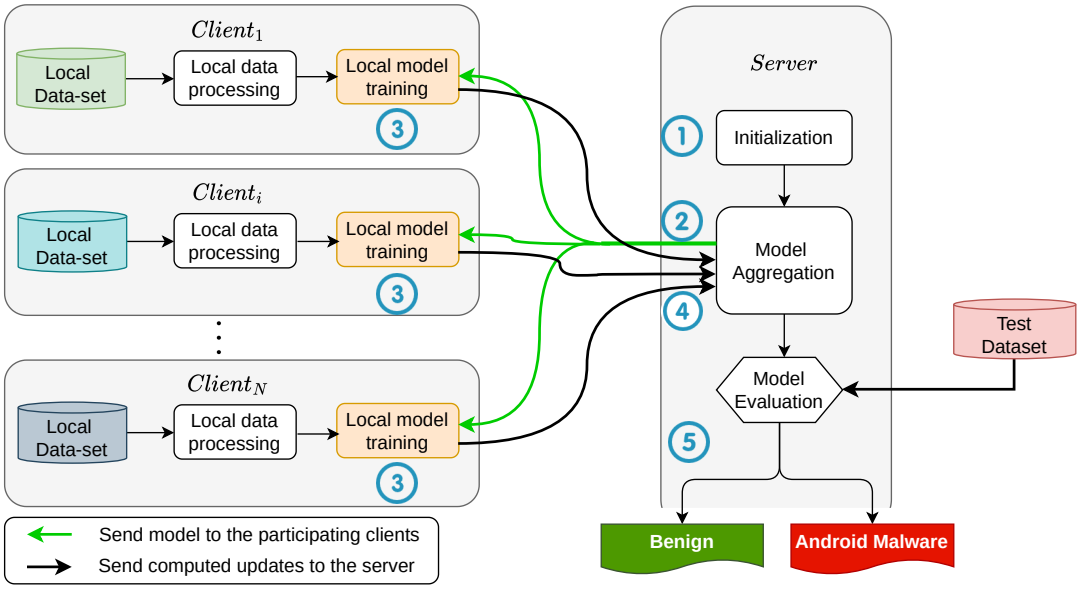


FIGURE 3.3: Flowchart: FDL-Based Detection of Android Malware.

using PyTorch library and GPU hardware accelerator for enhanced computational efficiency. A detailed overview of our experimental setup is outlined in Table 5.2.

To validate the performance of the proposed FedCNN approach, a comparative analysis was conducted against a centralized alternative. This alternative employed the same CNN model architecture and training configurations. A series of experiments were thoroughly conducted, during which hyper-parameters were fine-tuned to ensure a detection model characterized by precision and generalization.

In the presented comparative analysis (Table 3.2), we examine the performance of our proposed FedCNN for Android malware detection method in contrast to other relevant works. The assessment is carried out on the AAGM2017 dataset, employing key evaluation metrics such as Accuracy (Acc), Precision (Pr), Recall, F1-score, and Support. Notably, the evaluation settings in the related works differed markedly, encompassing distinct validation strategies and variations in the distribution of training and test samples (Support). Considering the centralized CNN model, the results demonstrate varying levels of performance. In this regard, the centralized CNN achieved an accuracy of 84% for malware detection, with precision and recall values

Subject	**Parameters**	**Values**
CNN()	Conv1d-1	[1, 64, 70]
	Conv1d-2	[1, 32, 70]
	Conv1d-3	[1, 16, 70]
	Linear-4	[1, 32]
	Linear-5	[1, 2]
	Learning rate η	0.001
	Loss function	*CrossEntropyLoss*
	Activation function	*ReLU*
	Batch size	126
	Classification function	*SoftMax*
FedL()	Clients Sets	[10, 20, 40]
	Data Distribution	IID
	Local epochs	[2, 3]
	Total rounds	30
	Local Batch size	32

TABLE 3.1: Experimental Settings for FedCNN.

Reference	Classes	Acc	Pr	Recall	F1-score	Support
lashkari et al. 2018 [97]	Benign + Mal	0.91	0.91	N/A	N/A	N/A
andresini et al. 2021 [99]	Benign	0.89	N/a	0.95	0.71	8000
	Malware			0.66		2000
acharya et al. 2022 [100]	Benign	N/A				
	Malware	N/A	0.97	0.96	0.97	1915
Centralized Cnn	Benign	0.84	0.87	0.89	0.88	41877
	Malware		0.78	0.76	0.77	22408
Proposed FedCNN approach	Benign	0.837	0.85	0.91	0.88	41877
	Malware		0.80	0.71	0.75	22408

Acc: Accuracy, **Pr**: Precision, **Support** : Number of test instances.

TABLE 3.2: Performance Comparison Between Our Proposed Detection Method (FedCNN) and Other Related Approaches Using the AAGM2017 Dataset.

Total clients	Round one			Round 10		
	Best client	Worst client	Global model	Best client	Worst client	Global model
K = 10	68.17	66.31	60.95	83.07	82.16	83.74
K = 20	69.01	66.45	68.27	82.14	81.74	82.27
K = 40	65.79	63.6	65.34	78.05	76.38	78.47

TABLE 3.3: Results of Accuracy Evaluation for the Proposed FedCNN Approach.

of 78% and 76%, respectively, resulting in an F1-score of 0.77%. Similarly, for the proposed FedCNN approach, the accuracy reached 83.7%, with a precision of 80% and

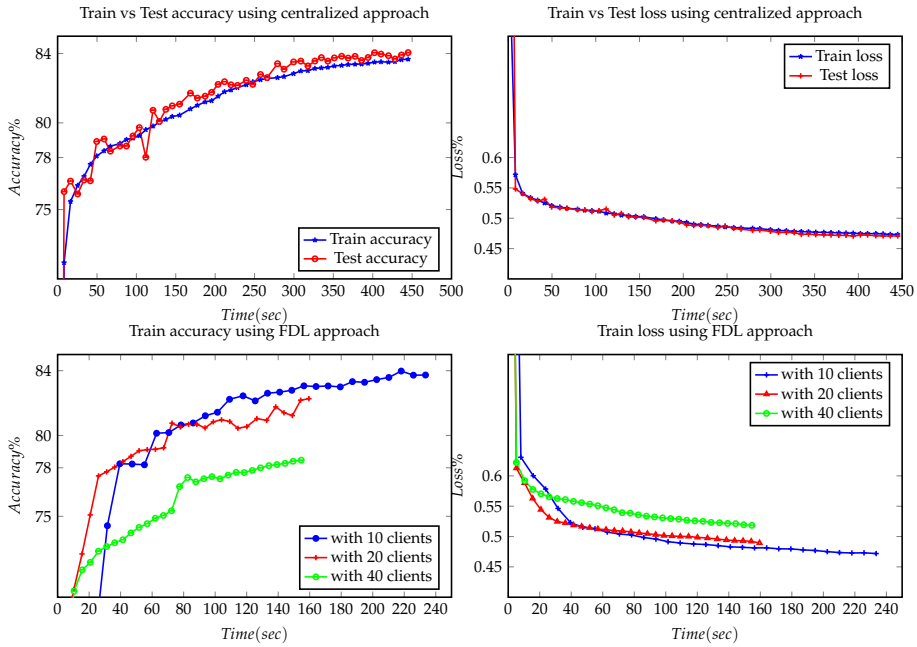


FIGURE 3.4: Analyzing Model Accuracy, Loss, and Time Complexity Across Various Training Approaches.

	Benign	Malware
Benign	37949	3928
Malware	6530	15878

(A) Using the centralized CNN

	Benign	Malware
Benign	37270	4607
Malware	5377	17031

(B) Using proposed FedCNN

FIGURE 3.5: Confusion Matrix: Insights and Outcomes.

a recall of 71%, yielding an F1-score of 75%. Furthermore, FedCNN effectively classified instances of the "Benign" class, corresponding to normal applications, with a recall rate of 92%. In contrast, the "Malware" class, including all 10 Android malware families, achieved a detection rate of 71%.

The results highlight the efficiency of the FedCNN approach, as it achieves a performance level nearly on par with the centralized approach. Nonetheless, it's worth noting that the outcomes of both detection methodologies fall short of meeting the requirements for real-world application. This is primarily attributed to the elevated

incidence of false positives and false negatives, as depicted in Figure 3.5.

Figure 3.4 compares model accuracy, loss, and time complexity across distinct training approaches. The time complexity analysis highlights the effectiveness of the proposed FedCNN approach. However, it's noteworthy that with an increased number of participating clients, the global model's accuracy declined from 83.74% to 78.47%, as illustrated in Table 3.3.

3.5 Chapter Summary

This chapter examines the effectiveness of an FL paradigm and network behavior analysis for malware detection, focusing on privacy preservation, computation cost, and detection efficiency. The analysis employed network layer features of malware samples to identify variations from their normal behavior. Experimental results demonstrated the efficiency and effectiveness of the proposed FL using a CNN approach compared with conventional centralized methods in terms of computation cost and privacy protection. However, the detection efficiency was inadequate when considering only network-based statistical features. Additionally, this analysis is confined to sets of malware that require network connectivity and exhibit abnormal network behavior.

Future research in malware detection could incorporate diverse sources of malware behavior data and ensemble learning to enhance detection capacity. In contrast, our primary focus centers on evaluating our proposed federated learning-based IDS methodology using recent Industrial IoT datasets while addressing other significant challenges and the associated security concerns, as discussed in Section 2.5.

Therefore, the next chapter will refine this approach by introducing an improved privacy-preserving FL, incorporating non-identically distributed data (non-iid), and establishing a secure IDS framework to counter the associated security risks.

CHAPTER 4

PPSS: PRIVACY-PRESERVING SECURE SYSTEM FOR INDUSTRIAL IOTS

"It is a capital mistake to theorize before one has data. Insensibly one begins to twist facts to suit theories, instead of theories to suit facts"

— Sherlock Holmes

4.1 Introduction

Drawing upon the insights acquired from the preceding chapter, FL-based IDS offers enhanced computation efficiency in deploying ML and DL approaches, outperforming centralized methods without compromising sensitive data to privacy issues. However, FL trains the models locally and transfers the updates to the centralized server for aggregation. Consequently, intruders or untrusted participants can compromise the quality of model updates and data privacy by exploiting inference attacks [101]. Furthermore, the centralized aggregation point presents a significant vulnerability as it functions as a single point of failure. This has given rise to new challenges, including establishing a reliable framework for secure aggregation and validation of

uploaded updates, addressing issues of system unreliability, and ensuring the safeguarding of privacy during the model uploading process.

To address these challenges, this chapter introduces an innovative and privacy-preserving secure framework named PPSS, which leverages the potential of blockchain technology by implementing a lightweight consensus protocol to optimize and secure the process of FL across untrusted participants. The effectiveness of PPSS is thoroughly assessed using a recent Industrial cyber security dataset (Edge-IIoT). A comprehensive set of key metrics, including detection rate, accuracy, computational efficiency, and energy consumption, is employed to evaluate the framework. Furthermore, this evaluation encompasses both non-IID and IID data distribution modes.

The remainder sections of this chapter are organized as follows: Section 4.2 provides an overview of the subject matter and the design objectives of PPSS. Section 4.3 discusses the development and experimental aspects of the PPSS framework, examining its components and algorithmic insights. It covers critical topics like component interaction, blockchain-enabled federated learning, secure communication, key management, proof of federated deep learning, and blockchain security analysis. Section 4.3.3 discusses PPSS-enabled cyber threat detection, including dataset selection, methods, and experimental settings. Section 4.4 analyzes PPSS performance across various scenarios, including class-specific, data distribution, global model accuracy, convergence time, differential privacy training, energy cost, and blockchain performance. Finally, Section 4.5 summarizes the chapter, providing a comprehensive overview of key findings and insights on the Privacy-Preserving Secure System for Industrial IoT and its multifaceted aspects.

4.2 Design Objectives of PPSS

The Industrial IoT brings numerous benefits to industries, such as increased efficiency, predictive maintenance, and real-time monitoring. However, it also introduces significant security risks and data privacy challenges. Industrial organizations must adopt

appropriate security mechanisms and strategies to effectively mitigate these potential cyber threats. Collaborating to implement a security monitoring mechanism like an IDS benefits industrial organizations by fostering shared threat intelligence, reducing costs, improving incident response, and collectively enhancing the security posture of the industry as a whole. In this context, cross-silo FL has emerged as a promising approach to address the unique challenges posed by the IIoT, including resource constraints and data privacy issues. However, the potential for malicious activity introduces concerns of model poisoning, privacy breaches, and intellectual property theft, while unintended privacy leakage can occur during aggregation. Secure aggregation protocols, model verification mechanisms, and trust-based systems must be employed to address these issues.

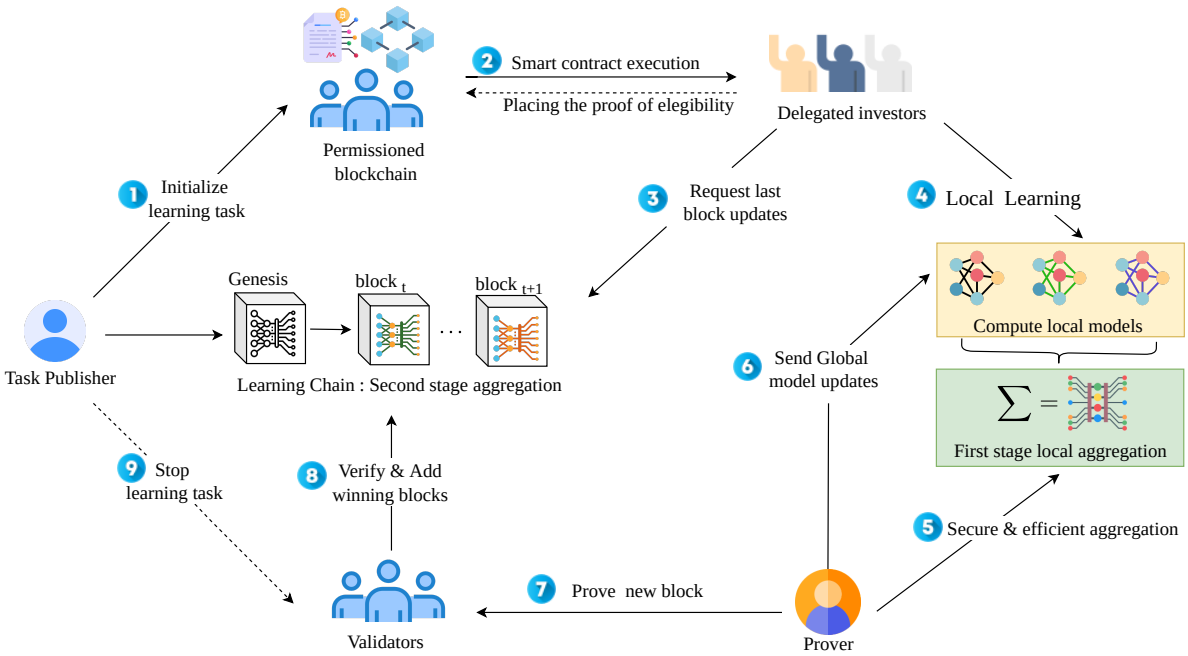


FIGURE 4.1: PPSS: An Overview of Our Proposed Blockchain and Federated Learning for Industrial IoT.

By navigating these challenges effectively, we designed a privacy-preserving secure system named PPSS for collaborative IDS across industrial organizations. Our PPSS security framework employs permissioned blockchain as a trust framework that verifies the identities of participating organizations to secure FL and multi-party

computation. Leveraging blockchain-distributed architecture, data is encrypted and transmitted using authenticated and private peer-to-peer (P2P) channels, allowing each organization to retain control over its data. This prevents unauthorized access, even if communication channels are compromised. Furthermore, the blockchain's design is customized to facilitate model sharing among participants, using cryptocurrency to reward and host qualified models and to encourage participant involvement and engagement in this collaborative environment.

PPSS incorporates two distinct federated stages for model aggregation to foster cross-silo FL-based IDS. The initial stage involves aggregating models across devices within an organization. Subsequently, the second stage occurs between participants, facilitated by the blockchain's utilization of model-containing blocks named the *Learning-Chain*. This enables the secure exchange of threat intelligence, enhancing the collective ability to detect and respond to emerging cyber threats.

At the core of PPSS, we incorporate a validation process for local training results, acting as a consensus mechanism within the blockchain. This mechanism, termed Proof-of-Federated deep learning (PoFDL), enhances privacy, reliability, and transparency. Figure 4.1 demonstrates the blockchain-based learning process of the proposed PPSS security framework, which enables secure communication and validation of the model updates. The chart illustrates how the organizations collaborate and contribute to the federated learning process while maintaining privacy and security :

1. **Initiating Learning Task**: Task publishers propose the learning process by creating a Smart Contract (SC) that defines the learning task (initial model, rewards, terms and conditions).

2. **Investor Applications**: Investors interested in participating submit applications to undertake specific tasks by providing proof of eligibility.

3. **Allocation of Terms**: Administrators review applications and assign predefined terms and conditions to eligible investors.

4. **Delegated Federated Learning (FL):** Selected investors become delegated participants and initiate Federated Learning (FL) tasks. Importantly, FL tasks are carried out without sharing raw data.

5. **Prover Supervision**: Each investor, acting as a *Prover*, moderates the FL task for their respective devices. They ensure the integrity of aggregation, transaction verification, and block generation processes.

6. **Global Update Transmission**: *Provers* share global updates with their devices, prompting subsequent rounds of federation. The focus here is on continuing the process, not yet on resolving the Proof-of-Federated deep learning (PoFDL).

7. **PoFDL Resolution and Block Generation**: The PoFDL challenge is resolved once conditions are met. The *Provers* create a new block containing the validated information and broadcast it to *Validators*, whose role is to validate the block's contents and reach a consensus.

8. **Learning-Chain Inclusion**: Upon consensus, the validated block becomes part of the Learning-Chain, ensuring a secure and tamper-proof record of the learning process. Both *Provers* and *Validators* are rewarded proportionally for their contributions. This mechanism encourages secure transfer learning among all participants.

It's worth noting that the task publishers and the *Provers* may either be intrinsic components of the blockchain system or external entities. In contrast, the *Validators* represent the trusted blockchain maintainers.

4.3 Framework Development and Experiments

This section presents the design scheme of our proposed PPSS security model. Figure 4.2 showcases the workflow of PPSS, illustrating its application in collaborative model training within Industrial IoT networks. The primary objective of this system

is to facilitate the training of a DL model using a federated approach. This process involves two distinct stages: local aggregation and global aggregation, referred to as off-chain and learning-chain aggregation. These stages are overseen by the Prover and Validator nodes, respectively.

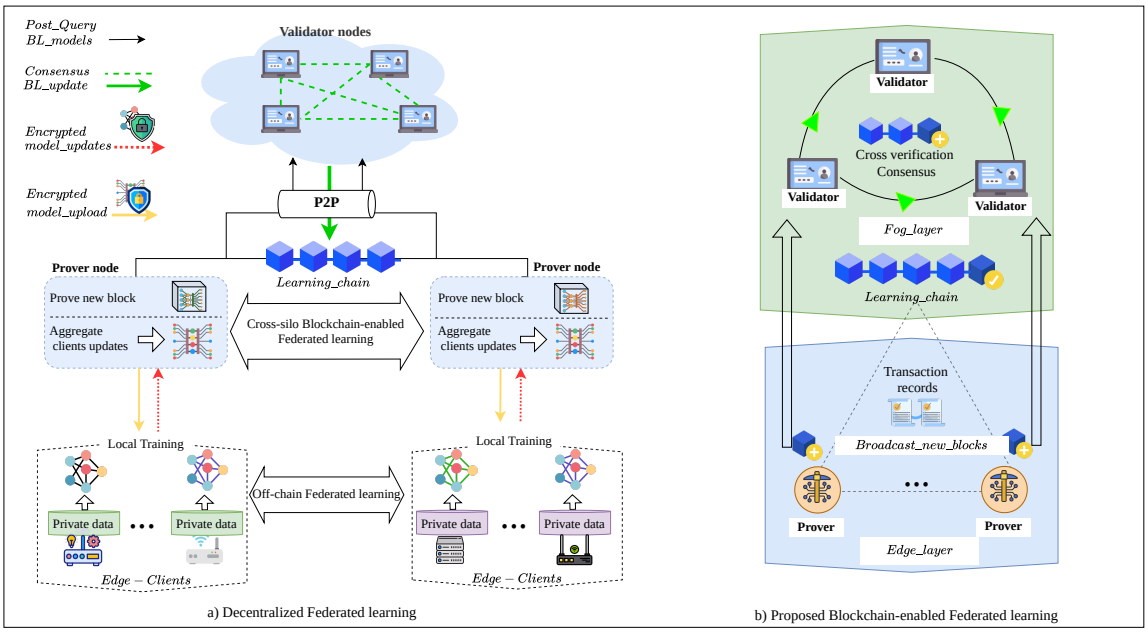


FIGURE 4.2: PPSS Security Model for Industrial IoT Networks: Overview of Architectural Framework and System Components.

Operating at the edge layer, Figure 4.2.a demonstrates the off-chain FL, which involves local model training within each participating organization, overseen by authorized representatives known as Provers ($\mathcal{P}$). The aggregated local model that results from this process is then sent back to the clients for more communication rounds if the model does not meet the criteria for global aggregation.

In contrast, the *Learning-Chain* operates within the fog layer and employs the permissioned-blockchain technology to facilitate the sharing of models and updates among organizations. The blockchain functions as a distributed ledger, recording model updates in blocks encompassing parameters and additional information like the origin organization and timestamp. In this paradigm, Validator entities ($\mathcal{V}$) play a pivotal role as trusted maintainers of the *Learning-Chain*, ensuring the integrity of the in-chain FL process as demonstrated in Figure 4.2.b. Furthermore, a consensus mechanism is

implemented among $\mathcal{V}$ nodes to validate block data, ensuring coherence across the blockchain's distributed ledger. A detailed discussion about blockchain integration in Section 4.3.2

This approach ensures the security and transparency of model updates, establishing a trustworthy and auditable account of the federated learning process.

4.3.1 Overview of Component Interaction and Algorithmic Insights

Notation	Description
C	Gradient norm bound
$\mathbf{I}$	Identity matrix
D	Local dataset
E	Local training epochs
k	Global security parameter
m	Message
W	Model weights
σ	Digital signature
SC	Smart Contract
sh_k	Ephemeral symmetric key
$\mathcal{P}sk$	Prover secret key
$\mathcal{P}pk$	Prover public key
$\mathcal{C}sk$	Client secret key
$\mathcal{C}pk$	Client public key
α	Learning rate
ϕ	Noise scale
(ϵ, δ)	Privacy cost
$g(x_i)$	Gradient computed on x_i
G_M	Global model
Txs	Transactions building the global model
$aggregate(.)$	Aggregate models by averaging (FedAvg)
$SC.aggregate(.)$	Aggregate models using smart contract
$Sign(.)$	Digital signature function
$Verify(.)$	Verify digital signature function
$Encrypt(.)$	Symmetric encryption function
$Decrypt(.)$	Symmetric decryption function
T_x	Transaction
Proof	Performance metric (e.g., Accuracy)
TestSet	Cross-validation dataset

TABLE 4.1: Notation for Algorithm Discussion.

Algorithm 2: Secure Aggregation in PPSS.

Validator Nodes : ***Function*** *validate_Blocks (Learning − Chain)*
  Validate submitted new model-containing blocks
  Achieve consensus and add model-containing blocks to *Learning-Chain* (refer to Algorithm 3)
Prover Nodes : ***Function*** *offChain_FL (Learning − Chain,* TestSet*)*
  Initialize model W from $Learning - Chain$
  for *each round* $t = 1$ *to* R **do**
    $S_t \leftarrow$ Random subset of clients k
    $m \leftarrow Encrypt(W, sh_k)$
    for $k \in S_t$ *in parallel* **do**
      $m_t, \sigma \leftarrow Edge\_client\ (m, Sign(m, \mathcal{P}sk))$
      if *Verify*(m_t, σ) **then**
        $W_{t+1}^k \leftarrow Decrypt(m_t.w_t, sh_k)$
      end
    end
    $W_{t+1} \leftarrow$ ***aggregate***$(W_{t+1}^1, W_{t+1}^2, \ldots, W_{t+1}^k)$
    $Proof \leftarrow$ predict$(W_{t+1},$ TestSet$)$
    if $Proof \geq Learning - Chain.Proof$ **then**
      $G_M, Txs \leftarrow$ ***SC.aggregate***$(m^1, m^2, \ldots, m^k)$
      Submit new_Block(G_M, Txs)
    end
  end
Edge-clients : ***Procedure*** *local_Training* $(m, \sigma, Params)$
  Input: α, E, ϕ, C
  Output: Privacy-preserved, signed, and encrypted model update
  if *Verify*(m, σ) **then**
    $W_k \leftarrow Decrypt(m.w, sh_k)$
  end
  for *batch of samples* B_j *in* D **do**
    Compute gradient $\mathbf{g}$
    for $i \in B_j$ **do**
      Compute $\mathbf{g}_j(x_i) \leftarrow \partial_{w_j} \beth(w_j, x_i)$
      Clip $\check{\mathbf{g}} \leftarrow \mathbf{g}$
      $\check{\mathbf{g}}_j(x_i) \leftarrow \frac{\mathbf{g}_j(x_i)}{\max(1, \frac{\|\mathbf{g}_j(x_i)\|_2}{C})}$
      Add noise
      $\check{\mathbf{g}}_j \leftarrow \frac{1}{|B|} \sum_i (\check{\mathbf{g}}_j(x_i) + \mathcal{N}(0, \phi^2, C^2\mathbf{I}))$
      Descent
      $w_{i+1} \leftarrow w_i - \alpha \check{\mathbf{g}}_j$
    end
  end
  $m \leftarrow Encrypt(W_E, sh_k)$
  Send $(m, Sign(m, \mathcal{C}sk))$ to corresponding Prover node

Figure 4.2.a illustrates the procedural dynamics of blockchain-enabled decentralized FL (DFL), demonstrating comprehensive interaction among components throughout process interaction. Table 4.1 provides a comprehensive notation guide for algorithmic discussions. Algorithm 2 demonstrates the operational framework for how various entities interact to achieve secure model aggregation within the context of DFL:

- **Validator Nodes ($\mathcal{V}$)**: As depicted in Function 1 of Algorithm 2, $\mathcal{V}$ nodes play a pivotal role as trusted authorities, dedicated to ensuring the integrity of the blockchain while facilitating the efficient storage of novel models within the *Learning-Chain*. They rely on the PoFDL algorithm (Section 4.3.2.2) to attain consensus regarding including new models. Once consensus is achieved, relevant participants are promptly notified, thereby initiating the transfer learning process by utilizing recently incorporated models. Moreover, the $\mathcal{V}$ nodes can also take on additional responsibilities as provers, leveraging their inherent data and computing resources to execute outstanding tasks.

- **Prover nodes ($\mathcal{P}$)** : Function 4 of Algorithm 2 illustrates the role of $\mathcal{P}$ nodes in orchestrating a decentralized FL (DFL) process and showcases $\mathcal{P}$ nodes' dual function of coordinating localized FL updates and underscoring their contribution to efficient collaborative learning and *Learning-Chain* integration. Through iterative rounds, $\mathcal{P}$ nodes engage a subset of clients, encrypting and processing their model updates. These updates are aggregated and used for prediction, with $\mathcal{P}$ nodes checking if the predictive Proof meets a preset threshold. Upon meeting this criterion, a smart contract aggregates the encrypted updates into a global model, which is subsequently packaged into a new block for submission.

- **Edge-clients** : Function 22 of Algorithm 2 illustrates the role of *Edge-clients* in contributing to secure, collaborative learning through sophisticated privacy-preserving and local_Training techniques. The procedure includes input and output specifications, verification and decryption, gradient computation and

clipping, privacy-preserving gradient manipulation, parameter updating, and secure encryption and transmission. Input parameters like α, E, ϕ, and C generate a privacy-ensured, signed, and encrypted model update. The procedure guarantees the authenticity of the received encrypted model and decrypts it using the shared key "sh_k," resulting in the localized model "W_k."

Gradient computation and clipping occur for each batch of training samples, yielding individual gradients for each data sample. The privacy-preserving technique of gradient clipping involves scaling the gradient by a factor determined by the gradient norm bound C. To enhance privacy, Gaussian noise is added to the clipped gradients, employing a zero-mean Gaussian distribution with a variance of ϕ^2 scaled by $C^2\mathbf{I}$. The aggregation of gradients is performed by calculating the average of locally adjusted gradients across all samples in the batch.

The process continues with the update of parameter weights using the aggregated gradient $\check{\mathbf{g}}_j$, effectively updating the model's weights as $w_{i+1} = w_i - \alpha\check{\mathbf{g}}_j$. Lastly, the model is encrypted using the encryption algorithm E and the shared key "sh_k", after which the encrypted model is sent to the corresponding $\mathcal{P}$ node, authenticated through the client's private key.

Our proposed PPSS secure system presents a sophisticated framework combining FL concepts and cryptographic principles to accomplish privacy-preserved, secure model aggregation. By outlining responsibilities, maintaining validation consensus, and prioritizing data privacy at every stage, PPSS serves as a promising strategy for enhancing collaborative machine learning within decentralized networks while safeguarding the sensitive nature of individual data points.

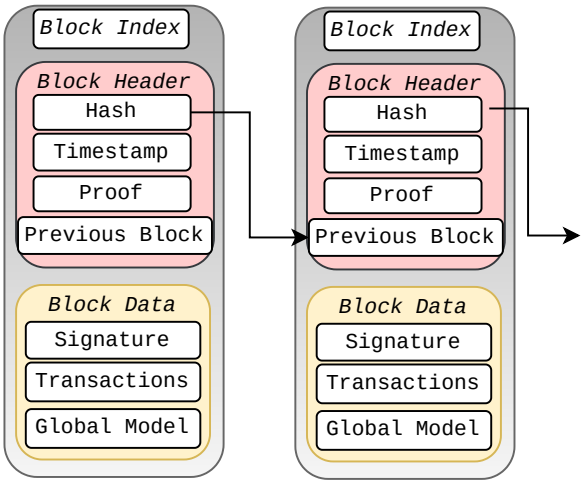


FIGURE 4.3: PPSS Learning-Chain Data Structure: Encapsulation of Information within Blocks.

4.3.2 Blockchain-enabled Federated Learning

Figure 4.3 illustrates the core architecture of the PPSS *Learning-Chain*. Each block contains essential components such as the Block Index and Block Header (Hash, Timestamp, Proof, Previous Block). The Block Data section houses a Signature with the client's ID and block data, Transactions timestamped and hashed using registered client IDs, and the Global Model. Building transactions for the global model are meticulously time-stamped and linked to registered client IDs, enhancing transparency. A cryptographic Signature fortifies data integrity. This data structure ensures secure, accountable information storage in the PPSS framework.

In the *Learning-Chain* network model, $\mathcal{P}$ nodes possess digital identities and access to the *Learning-Chain* for moderating local training and proposing new blocks to $\mathcal{V}$ nodes for inclusion in the *Learning-Chain*. To accomplish this, $\mathcal{P}$ nodes utilize confidential smart contracts to securely aggregate updates from their respective clients. As elaborated in [102], this approach leverages Trusted Execution Environments (TEEs) to ensure the aggregation process's security against unauthorized node manipulation. Then, $\mathcal{P}$ nodes generate blocks containing the approved model and necessary information resulting from successful smart contract execution. Subsequently, these

blocks are incorporated into the *Learning-Chain* during the subsequent global aggregation phase.

During the global aggregation phase overseen by $\mathcal{V}$ nodes, blocks containing models, as provided by $\mathcal{P}$ nodes, are incorporated into the *Learning-Chain*. This inclusion is accomplished through a consensus mechanism named Proof-of-Federated Deep Learning (PoFDL, Section 4.3.2.2). Once integrated, any authorized participant with access to the blockchain can request the latest model stored within the *Learning-Chain*. This model can then be utilized for deployment or further refinement purposes.

4.3.2.1 Secure communication and Key management To establish secure end-to-end communication between proposed PPSS system nodes, we propose a combination of asymmetric and symmetric encryption methods. AES is used for data encryption with a shared ephemeral key, while RSA handles authentication using private-public key pairs. A central entity, the "Trust Authority" ($\mathbb{A}$), manages the key generation, distributing public keys as identities and retaining private keys. In the second-stage aggregation, $\mathbb{A}$ establishes a secure AES key between the $\mathcal{P}$ and $\mathcal{V}$ nodes. In the first stage, $\mathcal{P}$ nodes secure communication among corresponding edge clients using AES-based data encryption. This comprehensive approach safeguards the Learning-Chain's integrity, protecting transmitted models from eavesdropping and cyber threats.

The establishment of *Learning-Chain* framework consists of the following algorithms:

- $PSetup(1^k)$: This algorithm takes 1^k where k is the security parameter of the system and returns the description of bilinear groups $\mathcal{E} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$.
- $KeyGen(\mathcal{E})$: This algorithm selects two generators $g_1 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$ with a random scalar $x \leftarrow \mathbb{Z}_p$. It produces a public/private key pair (pk_i, sk_i) for the party invoking it, where $pk_i = (g_1, g_2, \tilde{X}, z)$, $\tilde{X} \leftarrow g_1^x$, $z \leftarrow e(g_1, g_2)$, and $sk_i = x$.
- $KeyAgGen(\mathcal{E}, \mathcal{A})$: This algorithm selects a sequence of public keys $\mathcal{A}$ and then produces an aggregate public/private key pair $(\mathcal{A}pk_i, \mathcal{A}sk_i)$.

- $Sign(sk_i, m)$: This signature algorithm takes a message $m \in [WR, TS, AUX]$, a private key sk_i, and produces a signature $\sigma \leftarrow g_1^{1/(x+m)}$, where AUX includes the training information (e.g., hyper-parameters), TS is an optional collection of test data samples to evaluate model updates, and WR is a collection of intermediate model weights registered during the training process.

- $Verif(m, pub_i, \sigma)$: This verification algorithm takes a signature σ, a message $m \in [WR, TS, AUX]$, and a public key pub_i. The algorithm tests $e(\sigma, \tilde{X} \cdot g_2^m) = z$ and returns true or false.

Definition 1 (*Learning-Chain* correctness). Let O be a t-Learning-Chain scheme initialized with $\mathcal{E} \leftarrow PSetup(1^k)$, $KeyGen(\mathcal{E})$, and $KeyAgGen(\mathcal{E}, \mathcal{A})$, where $\mathcal{E} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$. Let $(pk_1, sk_1), \ldots, (pk_i, sk_i)$ be a sequence of keys generated via $KeyGen(\mathcal{E})$. Let $(\mathcal{A}pk_i, \mathcal{A}sk_i)$ be an aggregate public/private key pair generated via $KeyAgGen(\mathcal{E}, \mathcal{A})$. Let $m \in [WR, TS, AUX]$ be a message, and let $(pk_1, \sigma_1), \ldots, (pk_i, \sigma_i)$ be any sequence of key/signature pairs, where $\sigma \leftarrow g_1^{1/(x+m)}$. The O scheme is valid if, for every message and sequence, the following criteria are satisfied:

- When the $Verif(m, pub_i, \sigma_i)$ algorithm tests $e(\sigma_i, \tilde{X} \cdot g_2^m) = z$, the result is true for all i.

The integration of *Learning-Chain* with FL comprises four distinct phases: 1) Initialization phase, 2) agreement phase, 3) Model-Containing Block Generation Phase, and 4) PoFDL) Consensus Enabling Phase.

- **Initialization phase :** This involves a process where the *Trust authority* ($\mathbb{A}$) registers various types of nodes within the system: $\mathcal{V}$ nodes, which are trusted parties capable of validating and adding new blocks to the system; $\mathcal{P}$ nodes, which have limited capabilities and can only query the *Learning-Chain* and create new blocks; and *Edge-clients* owned by $\mathcal{P}$ nodes, serving as model trainers without querying capabilities. Each entity is provided a unique $\mathbb{ID}$ for authentication and a security parameter k for key generation. Additionally, $\mathcal{P}$ nodes

receive the initial model state W_0, learning parameters, and a data partition for constructing the Proof of Federated Learning (PoFDL) from the *"Learning-Chain*. Key generation operations, including $KeyGen(\mathcal{E}) \rightarrow (pk_i, sk_i)$ for $\mathcal{V}$ nodes and *Edge-clients* and $KeyAgGen(\mathcal{E}, \mathcal{A}) \rightarrow (\mathcal{A}pk_i, \mathcal{A}sk_i)$ for $\mathcal{P}$ nodes using *Edge-clients* public keys, are performed. Furthermore, it's important to note that $\mathcal{P}$ nodes risk losing their credentials and permissions if they fail to submit qualified models according to smart contract conditions.

- **Agreement phase:** When a participant publishes a learning task by providing the initial information and conditions (i.e., initial model state W_0, labeled test_dataset TestSet, parameters) for $\mathcal{P}$ nodes who want to join the federated learning task. $\mathcal{P}$ must provide proof of eligibility, such as training performance using its resources, to adhere to the smart contract terms to contribute to the *Learning-Chain*. The selected eligible $\mathcal{P}$ nodes must register all corresponding *Edge-clients* to ensure authentication.

- **Model-Containing Block Generation Phase:** A Prover $\mathcal{P}$ generates a block with corresponding *Edge-clients*' updates. These updates are verified using the tamper-proof ledger of the *Learning-Chain* as a reference to identify malicious clients. Only valid updates are encapsulated as transactions (T_x). Given two different hash functions: $H_1 : \Theta \times \{0,1\} \rightarrow \{0,1\}$ and $H_2 : \{0,1\} \times \{0,1\}^* \rightarrow \Omega$. Given a secret key $x_i, y_i \in \mathbb{Z}_p^*$ and a block $Bloc_i \in \{0,1\}^*$, $\mathcal{P}$ picks: $g_1 \in \mathbb{G}_1$, a random number $r_i \in \mathbb{Z}_p^*$ and computes $\sigma_i = H_1(g_1^{(x_i + Bloc_i + y_i + r_i)^{-1}}) \in \{0,1\}$. Then $\mathcal{P}$ computes $b_i = H_2(\sigma_i, Bloc_i, r_i) \in \Omega$ and sets the time intervals of a block generation as T. The signature of a block $Bloc_i$ is (σ_i, b_i, r_i). Finally, $\mathcal{P}$ broadcasts transaction data combined with the signature to the blockchain $\mathcal{V}$ nodes.

- **PoFDL Consensus Enabling Phase**: discussed in the following section 4.3.2.2

4.3.2.2 Proof of Federated Deep Learning for Consensus Establishment : Inspired by the Proof-of-Authority (PoA) consensus mechanism, we develop the PoFDL to

complete verification and add new blocks to the *Learning-Chain*. However, in contrast to relying solely on pre-selected reputable nodes, we empower each $\mathcal{P}$ to become a $\mathcal{V}$ node in the PoFDL by staking a deposit of cryptocurrency or staking their reputation. This approach enhances trust levels among participants and strengthens blockchain immutability. Algorithm 3 describes PoFDL consensus-driven procedures, whereby a requisite number of $\mathcal{V}$ nodes confirm the validity of added blocks.

Specifically, $\mathcal{V}$ nodes maintain the *Learning-Chain* by adding new blocks. After a $\mathcal{P}$ generates a new block, it submits it to the corresponding mining authority for verification. This authority operates as the "**Leader**" for the subsequent block in the *Learning-Chain*. To equitably distribute the responsibility of block creation among validators, PoFDL implements a time-based mining rotation scheme, ensuring the selection of a single elected **Leader** at each time-step, as specified by the smart contract [103]. If the current leader fails to transmit a block within the allotted time, they must submit an empty block to uphold their reputation.

Figure 4.4 illustrates the consensus process and message exchanges for block proposals based on PoFDL. The *Leader* broadcasts the received block to other validators for block acceptance. Each ($\mathcal{V}$) evaluates the received model-containing block, broadcasts the results, and compares them with those of other validators to decide on block acceptance. The block validation process and consensus mechanism are depicted in Algorithm 3. The block is added to the chain if: (1) the *Leader* is the one anticipated to be the current leader, and (2) at least $\frac{N}{2} + 1$ *Validators* received the same block and confirmed its acceptance.

In contrast to prior proof of learning concepts [104], our proposed PoFDL uses the inference phase to validate learned models, ensuring computational efficiency and data privacy. This approach involves:

(i) Each *Prover* $\mathcal{P}$ receives the same TestSet of labeled data to prove new models.

(ii) Each *Validator* $\mathcal{V}_i \in N$ is allocated a distinct partition of labeled data $D_i \in$ ValidationSet during the initialization phase.

The following relationships constrain these conditions:

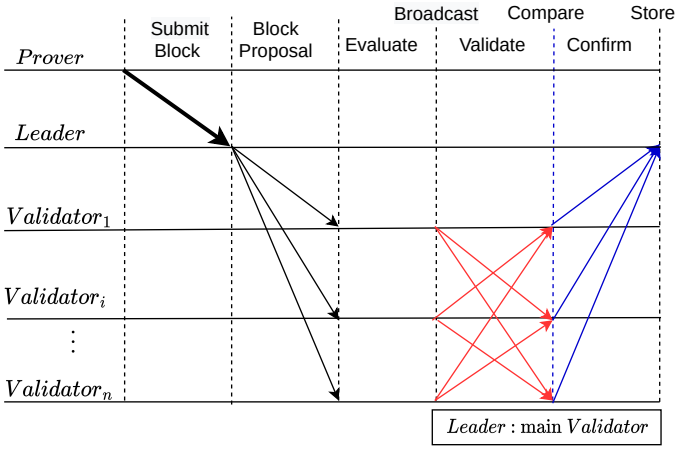


FIGURE 4.4: Consensus Process and Message Exchanges in PoFDL.

Algorithm 3: Validators: Validate and Add New Block.

Validators *Function* *ValidateAndAddBlock(Block, previous_hash,* $\mathbb{D}^*$*)*
  Input : Block, previous_hash, Dependent Validation data: $\mathbb{D}^*$
  Output: Add valid block: Success or Fail
  for *Validator* $\mathcal{V} \in N$ **do**
    if *Sender is current Leader* **then**
      if *Authenticated and Valid (Block.signatures)* **then**
        $Valid\_proof \leftarrow$ predict(Block.model, $\mathbb{D}^*$)
        if $Average(Valid\_proof, Block.Proof) \geq Block[previous\_hash].Proof$ **then**
          $\mathcal{V}$ Broadcast m(Block.hash, Success) to Validators
      else
        Broadcast to all (Block.hash, Fail)
    else
      Broadcast to all (Block.hash, Fail)
  if *Each* $\mathcal{V} \in N$ *upon receiving at least* $N/2+1$ m*(Block.hash, Success)* **then**
    Reaches consensus and confirms adding full Block data to Leader
  else
    Sends a penalty to Leader
  if *Leader upon receiving at least* $N/2+1$ *Confirm adding (Block.hash)* **then**
    It stores the current block

$$\mathsf{TestSet} \cap \mathsf{ValidationSet} = \emptyset \tag{4.1}$$

$$\mathsf{ValidationSet} = (D_1, D_2, \ldots, D_N) \tag{4.2}$$

$$\forall (D_i)_{i \in I} \subset \mathsf{ValidationSet};\ \text{with Card}\ I = \frac{N}{2};\ \bigcap_{i \in I} D_i = \emptyset \tag{4.3}$$

To maintain integrity, at least $N/2$ validation data partitions are required to be mutually exclusive. This requirement is to avoid fraudulent actions by Provers and Byzantine nodes. As depicted in Figure 4.4, $\mathcal{P}$ functions as the current *Leader* node, and if qualified, as $\mathcal{V}$ in subsequent rounds. The *Leader* proposes model-containing blocks to $\mathcal{V}$ nodes. After evaluating and broadcasting results, the system reaches consensus with at least $\frac{N}{2} + 1$ agreeing $\mathcal{V}$. Validated blocks are stored in the *Learning-Chain*, and both *Leader* and $\mathcal{P}$ receive cryptocurrency rewards.

This innovative approach consensus safeguards against malicious activities, enhances efficiency, and guarantees the systematic incorporation of validated model updates into the collaborative learning process.

4.3.2.3 Blockchain Security Analysis

1. **Sybil attack:** This attack undermines the decentralized nature of the network by creating a large number of pseudonymous identities to gain influence. Then, the attacker can manipulate consensus mechanisms or execute malicious actions by leveraging their extensive control over these fake identities. To oppose these nodes, our PPSS design only admits Prover nodes with positive reputations earned by contributing positively to the learning environment. This ensures that only trusted and reliable participants can participate in the federated learning process.

2. **Byzantine attacks:** Referred to as Byzantine nodes, these entities intentionally deviate from the established protocol to disrupt consensus mechanisms and compromise the integrity of the blockchain. To prevent attacks, our PPSS design allows trusted Validator nodes to collaboratively detect malicious nodes during their leadership using a voting mechanism. A leader node can be voted as malicious and subsequently removed based on the following scenarios:
   (i) failing to propose any blocks; (ii) overstepping the expected number of proposed blocks (Denial of Service attacks); or (iii) presenting varying blocks to different authorities.

3. **Model inversion and membership inference attacks**: To mitigate these attacks, especially in honest-but-curious scenarios, our PPSS framework incorporates a differential privacy-enhanced training mechanism. By differentiating parameter gradients of client models during training, our PPSS framework ensures that sensitive information about individual data records cannot be inferred from the model parameters. Furthermore, our framework employs encryption and authentication mechanisms to protect model sharing from the public. This further enhances the privacy and security of the FL, preventing unauthorized access to the models and reducing the risk of inference attacks.

4. **Model theft attacks**: This pertains to the scenario where a consensus node pilfers a trained model upon receipt for validation from other Provers, subsequently asserting ownership by re-broadcasting it to other consensus nodes, like with a replay attack. To prevent these attacks, we impose two security measures. Firstly, we require that a Prover node incorporate updates from intermediate clients into building transactions for the global model within the block data (Figure 4.3). These transactions are provided with timestamps and hashed using matching registered client IDs on the blockchain. Secondly, we require that a Prover include a signature in the block data containing their ID and the block data itself. This makes it difficult for an adversary to falsify block data and rebroadcast it. Moreover, this approach alleviates the communication overhead from messages between Prover nodes and Validator nodes during investigations.

By integrating these measures, the PPSS framework fortifies the security and integrity of the model-sharing process, rendering it resilient against various attack scenarios.

4.3.3 PPSS-enabled Cyber Threat Detection

To safeguard industrial networks against large-scale and emergent cyber threats, such as cloud security weaknesses exposing sensitive data, ransomware attacks encrypting critical data, DDoS attacks causing operational disruptions, IoT device vulnerabilities leading to unauthorized access, insider threats, and Advanced Persistent Threats (APTs) maintaining stealthy network access and privacy breaches, we propose the implementation of a two-tiered security approach. This approach combines the PPSS framework with an anomaly and deep learning-based IDS, bolstering the network's overall security posture.

The PPSS framework serves not only to enhance privacy and security but also facilitates a decentralized deployment of the IDS system. In this arrangement, detection nodes receive frequent updates of efficient and reliable detection models. These models are trained across extensive networks with minimal cost. This improves the scalability and efficiency of the IDS.

4.3.3.1 DataSet Selection and Processing: Our study employs the recently proposed EdgeIIoTSet dataset for evaluation [33]. This dataset comprises a realistic representation of Industrial IoT environments, a comprehensive feature set, diverse attack scenarios, and suitability for FL-based IDS evaluation.

The following considerations illustrate why this dataset is an appropriate candidate for assessment of our proposed PPSS-enabled IDS :

1. **Realistic Environment Representation :** The dataset is specifically tailored for IoT and IIoT security research. it was created by modeling and emulating actual industrial systems in real-world IIoT environments, imparting a realistic representation.

2. **Comprehensive Feature Set :** The dataset encompasses extensive features from diverse sources, including alerts, system resources, logs, and network traffic. These features provide a rich source of information for training and enhancing

an FL-based IDS. The dataset contains over 10 million normal records, 9 million malicious, and 67 features. These records are collected from device and alert logs across a network of seven interconnected layers, which include the cloud/-fog computing layer, Blockchain layer, SDN layer, edge layer, and IoT/IIoT perception layer. The dataset also covers a range of related protocols, including industrial protocols like Modbus and MQTT.

3. **Variety of Attacks :** The dataset encompasses attacks relevant to IIoT connectivity protocols, systematically categorized into five threat categories as depicted in Table 4.2. These threats encompass a wide range of 15 class attack types, comprehensively representing the cybersecurity challenges in IoT and IIoT applications.

Attack Category	**Description**
Malware Attacks	These attacks involve the installation of backdoors or malicious programs on IoT devices or edge servers. This category covers attacks like Ransomware attacks and Backdoor attacks.
DoS/DDoS Attacks	These attacks are intended to render the victim's IoT edge server inaccessible to legitimate requests. This category encompasses attacks such as TCP SYN Flood DDoS attack, UDP flood DDoS attack, HTTP flood DDoS attack, and ICMP flood DDoS attack.
Information Gathering	These attacks involve the analysis of IoT data packets to identify vulnerabilities in IoT devices and edge servers. This category encompasses attacks like Port Scanning, OS Fingerprinting, and Vulnerability Scanning Attacks.
Man-in-the-Middle	These attacks involve the interception of communications between IoT devices and edge servers. This category includes attacks such as ARP Spoofing attacks and DNS Spoofing attacks.
Injection Attacks	These attacks involve sending malicious scripts to unsuspecting users, allowing the attacker to gain access to sensitive information. This category encompasses Cross-site Scripting (XSS) attacks and SQL Injection.

TABLE 4.2: EdgeIIoTSet: Attack Categories and Descriptions.

These alignments are with the focus of our research on an FL-based IDS for IIoT as a reliable and representative dataset when evaluating IDS within the complex and dynamic settings of IIoT applications.

The processing of the EdgeIIoTSet dataset involves detecting and rectifying corrupt or inaccurate records by eliminating duplicates and missing values. Main flow features, such as IP addresses, ports, timestamps, and payload information, are excluded. Furthermore, categorical variables undergo conversion into one-hot encoded feature variables [33].

4.3.3.2 PPSS Detection Method : In our IDS detection methodology, we've chosen anomaly-based detection. This method allows IDS to continuously monitor and categorize various behaviors, enabling timely identification of potential cyber threats. Moreover, this method has proven effective in identifying unknown attacks, including Zero-day attacks.

We leverage this method by employing convolutional neural networks (CNNs) as the foundational detection module [105]. Within the domain of DL, CNNs occupy a significant position as a distinctive model. Figure 4.5 depicts our proposed CNN model adopted within the Privacy-Preserving Secure System (PPSS) framework. The architecture of CNNs comprises interconnected convolutional layers that serve as information extraction modules. These layers employ learnable filters denoted as parameters W; these layers employ learnable filters denoted as W, applied to input data X, resulting in feature maps F through convolution represented as $F = W * X$. Subsequently, pooling operations reduce feature map dimensionality. For instance, the max pooling is defined by :

$$P_{\max}(F)_{i,j} = \max_{(m,n)\in R_{i,j}} F_{m,n} \tag{4.4}$$

. After a convolutional layer with pooling and activation, denoted as :

$$F_{\text{out}} = \sigma(P(W * F_{\text{in}})) \tag{4.5}$$

CNNs effectively identify localized patterns, capturing intricate details often overlooked by conventional neural networks. Additionally, CNNs incorporate fully connected layers for flattened feature maps, making them well-suited to identify several types of cyber attacks.

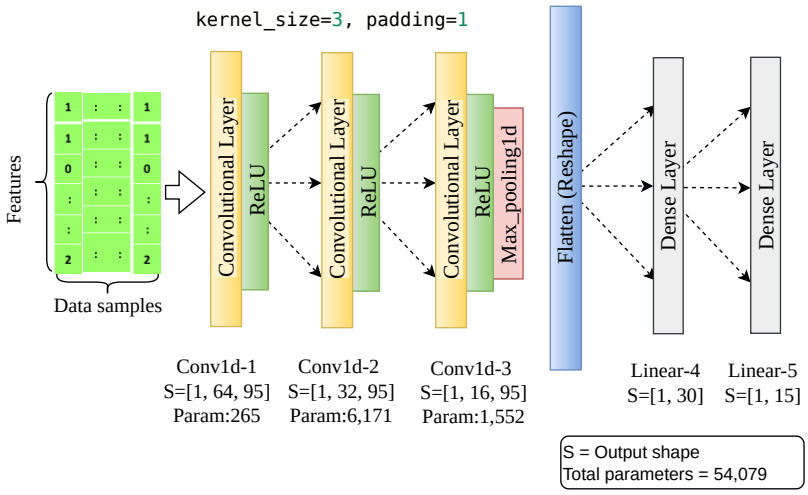


FIGURE 4.5: Structure of the CNN Model Adopted by the PPSS Framework.

4.3.3.3 Experimental Settings : The performance evaluation of the presented PPSS-enabled cyber threat detection was systematically evaluated within the controlled environment of Google Colaboratory, using the PyTorch library and Tesla-T4 GPU hardware accelerator for enhanced computational efficiency. The experiments were conducted in two distinct aggregation stages.

In the first stage, we instantiated localized Federated Learning (CFL) using multiple edge Provers, often called FL servers, each equipped with dedicated data resources and clients. Several scenarios were explored within this stage, including variations in the number of participating clients per Prover and the integration of differential privacy training via DP-SGD. These explorations assessed the initial aggregation phase's performance when dealing with limited data resources and privacy preservation concerns.

Subsequently, in the second aggregation stage, we introduced the PPSS-enabled decentralized FL (DFL) mechanism, leveraging the proposed PoFDL consensus mechanism (Section 4.3.2.2). This stage facilitated knowledge transfer among all participating Provers. During this phase, DFL executed a restricted federated aggregation exclusively incorporating validated models stored within the *Learning-Chain*. The evaluation encompassed assessing the global model's performance, accounting for different data distribution characteristics (namely IID/Non-IID) and exploring data augmentation concerning the number of participating Provers.

Within the differential privacy settings, we employed the Opacus library [106] for implementation, introducing a noise multiplier parameterized by (ϵ, δ) and imposing a maximum gradient norm value of $C = 1.2$. Table 4.3 shows the experimental configurations and learning parameters adopted in this study, while Tables 4.5 and 4.6 provide summaries of the evaluation outcomes for both localized and decentralized (CFL, DFL) across diverse settings.

Various metrics were employed to evaluate the efficiency and effectiveness of IDS (Intrusion Detection System) detection within CFL and DFL to assess the impact of security constraints on the learning process. These metrics encompassed Accuracy, Precision, Detection Rate, Time Complexity, and Energy Cost.

- **Time Complexity**: This metric represents the time complexity associated with the convergence of the global model. It encompasses several key factors, including the individual client's training time, the computational overhead of the model aggregation, and the time required for consensus inference under the PoFDL. Notably, this calculation does not incorporate the computational costs of secure communication and data transmission.

- **Energy Cost**: The term 'energy cost' pertains to the energy consumption incurred during training the global model. It is quantified by the following expression [107]:

$$\Theta(e, \mathcal{N}, \mathcal{R}) = \sum_{r=1}^{\mathcal{R}} \sum_{i=1}^{\mathcal{N}} \mathbf{1}_{\{n_{i,r}\}}(t_{n_i}, e_{n_i}) \quad Kwh \tag{4.6}$$

Here, e signifies the energy power consumption across N devices, R corresponds to the total federated rounds, t_{n_i} represents the wall clock time of a device n_i during round r, e_{n_i} signifies the energy consumption of device n_i during round r, and $1_{\{n_{i,r}\}}$ serves as an indicator function assessing whether a device n_i is chosen for FL training during round r. It is important to note that n_i can denote either a client or a server. The energy cost is expressed in Kilowatt per hour (*Kwh*) and is estimated using the *Carbontracker* library [108].

- **Heterogeneity in data distribution :** Experiments were conducted between Independent and Identically Distributed (IID) and Non-Independent and Non-Identically Distributed (Non-IID) data sets to evaluate training performance in diverse data distributions. In the IID scenario, the training dataset was partitioned into independent subsets with identical distributions, allocated to client groups, and overseen by a Prover node. This strategy maintained data homogeneity among clients, ensuring they had a comparable dataset. In contrast, a label partitioning approach was implemented in the Non-IID scenario, assigning each client group a randomly selected subset of labels associated with the same feature vectors in the training data. This setup assumed each Prover node had partial knowledge of the entire set of classes within the problem. The Non-IID configuration introduced data heterogeneity among clients, deviating from the uniformity observed in the IID scenario.

4.4 Results and Discussion

Numerous experiments have been systematically executed to assess the efficacy of the proposed PPSS-enabled Decentralized Federated Learning (DFL) framework. These

	Parameter	Values
Federated Learning	Number of Clients	[20, 40, 80]
	Global Rounds	15
Differential Privacy	Epsilon (ϵ)	[0.1, 1, 2]
	Delta (δ)	1.5e-5
	Gradient Norm Bound (C)	1.2
*	Model Architecture	CNN() (Refer to Figure 4.5)
	Learning Rate	0.01-0.001
	Optimizer	Adam
	Local Batch Size	100
	Local Epochs	1
	Loss Function	CrossEntropyLoss()
	Learning Rate	0.01

TABLE 4.3: Experimental Configurations for PPSS-enabled IDS.

experiments were designed to investigate the influence of security constraints on the FL learning process.

4.4.1 Class-Specific Performance Across Different Scenarios :

The examination of per-class performance using various models within the PPSS framework reveals several key findings, Table 4.4.

Both CFL and PPSS-enabled Decentralized FL exhibit exceptional precision and detection rates when identifying normal network traffic, underscoring their effectiveness in benign traffic detection.

In the context of attack identification, both Non-IID and IID data training approaches yield similar results in terms of precision and detection rates for both CFL and PPSS, demonstrating the efficient transfer learning enabled by federated aggregation in Non-IID data scenarios. However, utilizing Differential Privacy training via DP-SGD demonstrates a trade-off between privacy preservation and model performance, negatively impacting attack detection, especially for less-represented classes. PPSS excels over CFL in attack identification due to an additional aggregation phase using exclusively qualified models stored in the *Learning-Chain*, reducing training iterations. Nevertheless, certain attack classes, such as Fingerprinting and Ransomware,

		IID				**Non-IID**				
Classes	Metrics	Precision %		Detection rate%		Precision%		Detection rate%		**Support**
	Settings	CFL	PPSS	CFL	PPSS	CFL	PPSS	CFL	PPSS	
Normal	No-DP	100	100	100	100	100	100	100	100	323129
	DP	100	100	100	100	100	100	100	100	
Backdoor	No-DP	72	74	90	89	68	76	95	88	4972
	DP	72	72	89	89	40	00	92	00	
Vulnerability_scan	No-DP	93	94	85	85	94	94	85	85	10022
	DP	93	93	85	77	93	93	84	84	
DDoS_ICMP	No-DP	100	100	100	100	100	100	97	100	23287
	DP	100	94	97	100	100	90	69	100	
Password	No-DP	43	100	83	07	100	10	07	07	10031
	DP	14	36	01	100	00	00	00	00	
Port_Scanning	No-DP	65	65	09	09	29	65	92	09	4513
	DP	00	00	00	00	32	00	10	00	
DDoS_UDP	No-DP	98	98	100	100	98	98	100	100	22007
	DP	93	100	100	99	58	98	100	100	
Uploading	No-DP	59	57	39	37	31	100	83	15	7527
	DP	00	100	00	00	00	00	00	00	
DDoS_HTTP	No-DP	71	70	99	99	71	75	97	94	9982
	DP	70	67	99	99	70	70	98	99	
SQL_injection	No-DP	54	41	17	90	42	39	28	100	10241
	DP	37	00	98	00	37	37	100	100	
Ransomware	No-DP	00	00	00	00	00	77	00	14	2185
	DP	00	00	00	00	00	00	00	00	
DDoS_TCP	No-DP	69	68	100	100	00	69	00	100	10012
	DP	37	68	100	100	00	53	00	100	
XSS	No-DP	92	100	05	02	65	52	10	28	3183
	DP	00	100	00	00	33	00	00	00	
MITM	No-DP	100	100	100	93	100	100	100	93	80
	DP	00	00	00	00	00	00	00	00	
Fingerprinting	No-DP	00	00	00	00	13	00	57	00	200
	DP	00	00	00	00	00	00	00	00	

CFL : Centralized FL IDS; **PPSS** : PPSS-enabled decentralized FL IDS;
No-DP : No differentially private training; **DP** : with differentially private training;
Support : number of test samples; **IID, Non-IID :** data distribution

TABLE 4.4: Per-class performance using different models.

are prone to misclassification, emphasizing the limitations of transfer learning in cases of data insufficiency, notably within the FL framework.

These results underscore the PPSS framework's efficacy in normal traffic detection and attack identification while shedding light on the nuanced influence of differential privacy and the constraints of transfer learning in specific scenarios.

4.4.2 Evaluation Results of PPSS under IID Data Distribution :

Table 4.5 presents a comparative analysis of global model accuracies within the PPSS framework across various configurations, including client distribution and differential privacy settings. Test accuracy was used to assess Prover's performance.

In IID data distribution, the best Prover accuracy reaches 93.71% Expanding the Prover count to six, the best Prover accuracy attains 93.83% in the IID mode, and the global model accuracy reaches 93.98% with $K = 20$. In differential privacy settings, the best Prover accuracy is 93.46% with $K = 40$, while the global model accuracy achieves 93.72% at $K = 80$.

Finally, with eight Provers contributing to global model updates, the best Prover accuracy reaches 93.86% at $K = 20$, and the global model accuracy attains 94.01% with $K = 40$ in the IID scenario. However, under differential privacy constraints, the best Prover accuracy stands at 93%, with the worst at 90.92%.

These findings offer valuable insights into the performance of the PPSS framework across various settings, highlighting the influence of Prover count, differential privacy, and data distribution on model accuracy.

4.4.3 Evaluation Results of PPSS under NonIID Data Distribution :

Similarly, Table 4.6 compares global model accuracies within the proposed PPSS framework under NonIID Data Distribution. The best Prover accuracy attains 92.45% at a hyperparameter value of $K = 80$, while the worst Prover accuracy registers at 81.36% with $K = 40$. The global model achieves an accuracy of 92.52% at $K = 80$ when subjected to differential privacy constraints.

Expanding the Prover count to six, the best Prover accuracy achieves 93.27% in the Non-IID mode, with the worst accuracy declining to 83.22% at $K = 20$. The global model exhibits an accuracy of 93.75% with $K = 40$. In differential privacy, the best Prover accuracy is 92.11%

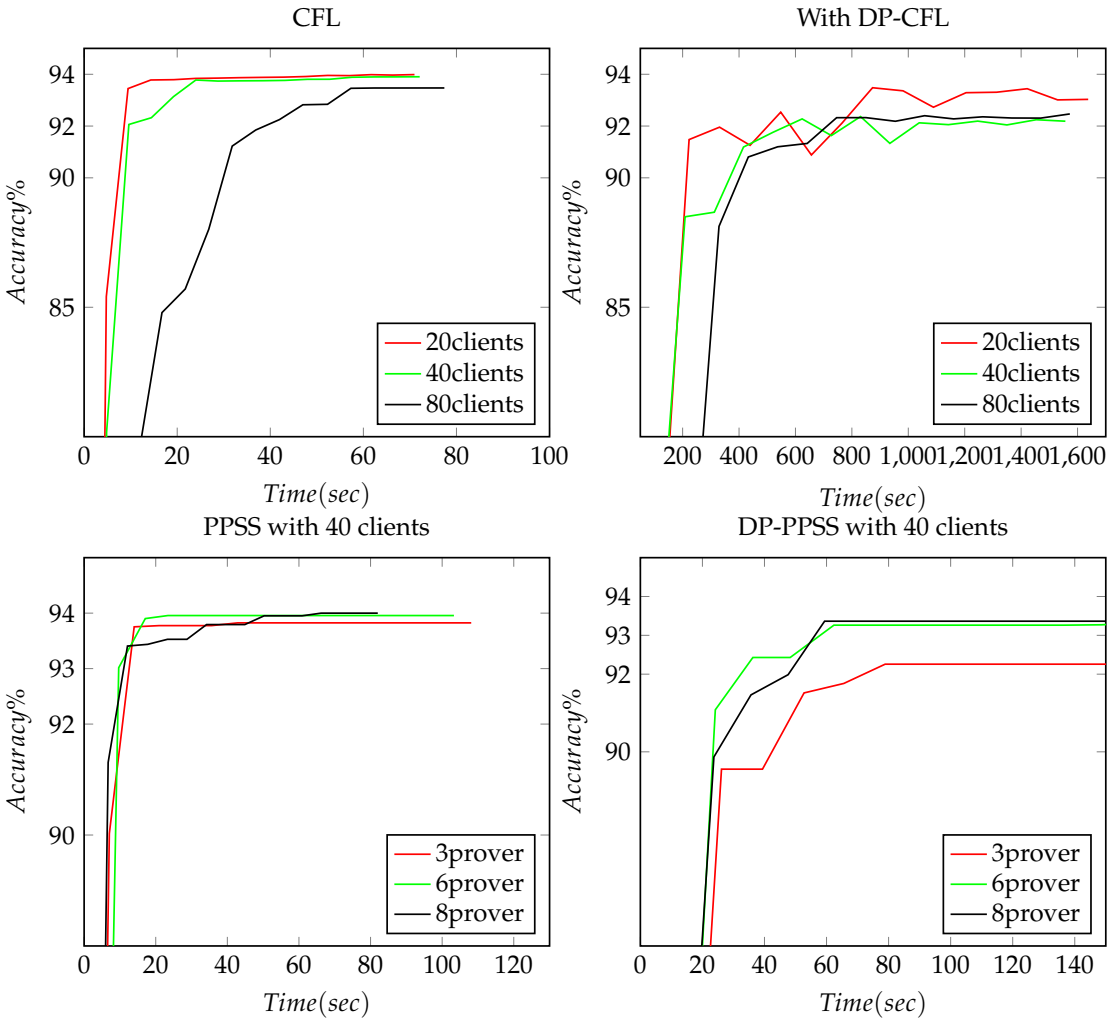


FIGURE 4.6: Temporal Evolution of Global Model Accuracy with Varying Numbers of Provers and Clients in DP-CFL and DP-PPSS.

With eight Provers contributing to global model updates, the best Prover accuracy attains 93.40% at $K = 80$, while the global model's accuracy reaches 93.74% at $K = 20$ within the Non-IID scenario. In differential privacy settings, the best Prover accuracy is 91.71%, with the worst at 80.84%. The global model's performance in this scenario reaches an accuracy 92.63% at $K = 80$.

Overall, we demonstrate that the number of participating clients had a minimal impact on the accuracy of the global model. Moreover, we can demonstrate that incorporating a certain number of Provers can alleviate the adverse effects of differential privacy on model accuracy.

Provers	Clients	1^{st} **round**						15^{th} **round**					
		Without DP			With DP			Without DP			With DP		
		B	W	G	B	W	G	B	W	G	B	W	G
P = 3	K = 20	90.12	86.43	90.12	75.44	73.21	75.44	93.44	92.80	93.99	92.65	91.71	93.40
	K = 40	90.03	89.71	90.03	73.21	73.21	73.21	93.71	93.35	93.85	91.66	90.97	92.26
	K = 80	87.89	85.49	87.89	73.21	73.21	73.21	93.70	93.67	93.90	92.38	90.31	92.09
P = 6	K = 20	92.81	78.92	92.81	78.57	36.75	78.57	93.83	93.55	93,98	89.30	82.76	91.47
	K = 40	91.02	83.12	91.02	73.21	73.21	73.21	93.79	93.42	93.97	93.46	91.30	93.46
	K = 80	90.94	88.13	90.94	73.21	73.21	73.21	93.80	93.67	93.97	93.14	92.01	93.72
P = 8	K = 20	93.21	55.39	93.21	86.98	73.21	86.98	93.86	75.26	93.92	90.92	82.46	92.23
	K = 40	91.30	75.47	91.30	75.18	73.21	75.18	93.84	93.20	94.01	93.00	88.33	93.36
	K = 80	93.25	81.93	93.25	73.21	73.21	73.21	93.84	93.33	93.90	91.98	90.04	92.90

(**W**): **W**orst prover ; (**G**): **G**lobal model ; (**B**): **B**est prover;

TABLE 4.5: Accuracy results of PPSS under IID Data Distribution.

Provers	Clients	1^{st} **round**						15^{th} **round**					
		No-DP			DP			No-DP			DP		
		B	W	G	B	W	G	B	W	G	B	W	G
P = 3	K = 20	87.46	78.56	87.46	73.21	73.21	73.21	92.41	81.39	92.44	89.55	80.00	91.74
	K = 40	87.78	79.52	87.78	73.21	73.21	73.21	92.32	81.36	92.48	89.15	80.06	89.80
	K = 80	88.87	80.91	88.87	73.21	73.21	73.21	92.45	81.42	92.52	87.89	80.06	88.75
P = 6	K = 20	90.69	73.86	90.69	84.60	71.29	84.60	90.79	83.22	93.28	89.59	78.52	91.83
	K = 40	91.32	78.94	91.32	78.20	73.21	78.20	93.27	81.50	93.75	90.18	80.06	92.27
	K = 80	86.64	80.07	86.64	80.33	73.21	80.33	93.19	81.36	93.55	92.11	80.29	93.18
P = 8	K = 20	92.61	73.65	92.61	73.21	73.21	73.21	91.92	73.90	93.74	90.68	80.16	92.45
	K = 40	91.06	81.25	91.06	75.03	73.21	75.03	93.35	81.51	93.69	90.94	80.05	92.38
	K = 80	92.58	78.46	92.58	73.41	73.21	73.41	93.40	81.10	93.69	91.71	80.84	92.63

(**W**): **W**orst prover ; (**G**): **G**lobal model ; (**B**): **B**est prover;

TABLE 4.6: Accuracy results of PPSS under Non-IID Data Distribution.

The findings of this study underscore the potency of the proposed PPSS framework as a viable and privacy-preserving solution for Intrusion Detection Systems in the realm of Industry 5.0. By skillfully merging blockchain and federated deep learning technologies, PPSS contributes to the fortification of cyber security in the context of Industrial IoT, laying the groundwork for enhanced protection against cyber threats while maintaining the integrity of sensitive data and critical industrial operations.

4.4.4 Global Model Accuracy and Convergence Time

Figure 4.6 visually represents the global model's accuracy evolution over time, considering varying numbers of Provers and clients. Notably, the PPSS approach demonstrates superior global model convergence time performance, particularly when employing DP-SGD for training. This advantage stems from PPSS's utilization of fewer training iterations and more intensive model aggregation operations than CFL. For instance, when deploying N clients in CFL, PPSS allocates N/P clients per Prover (where P is the number of Provers) for training, ensuring knowledge transfer for all N clients through aggregation. We can demonstrate that the computational overhead associated with model aggregation is significantly lower than individual client training. Additionally, DP training within PPSS can be improved by introducing additional Provers, thereby mitigating the adverse effects of Differential Privacy and enhancing overall training efficiency.

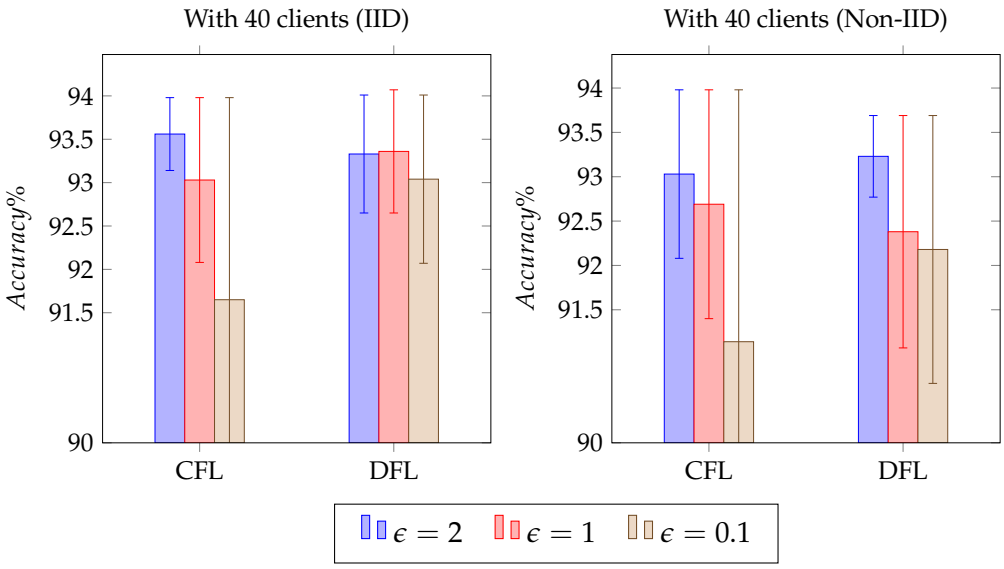


FIGURE 4.7: Comparative Analysis of Global Model Performance under High Privacy Regimes Employing DP-SGD.

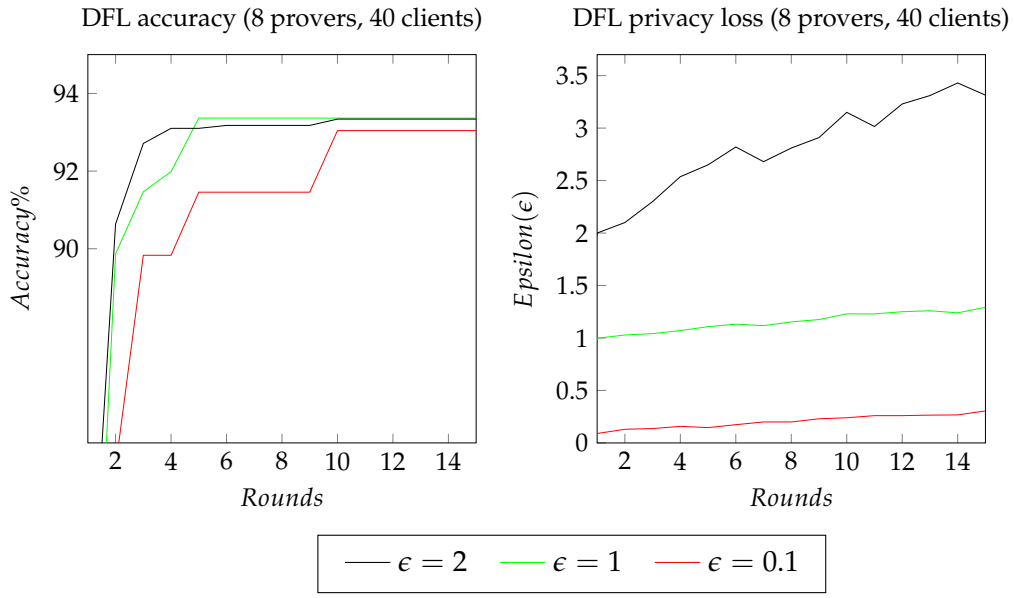


FIGURE 4.8: Comparative Analysis of Global Model Training and Privacy Loss Across Varied Noise Levels.

4.4.5 The Impact of Differential Privacy Training via DP-SGD on Global Model Accuracy:

Training using Differential Privacy Stochastic Gradient Descent (DP-SGD) to protect data privacy aligns with the principles of the strong composition theorem. This theorem asserts that the degree of privacy breach, quantified by standard (ϵ, δ)-differential privacy, tends to grow at an approximate rate of $\sqrt{K}$ under conditions of stringent privacy requirements. Here, K represents the number of training iterations in the learning process. Figure 4.8 and 4.7 provides an insight into the influence of differential privacy parameters (ϵ, δ) on global model performance across distinct data distribution modes IID and Non-IID. We conducted experiments using varying epsilon values ($\epsilon = 0.1, \epsilon = 1, \epsilon = 2$), representing the introduced noise level while maintaining a fixed δ value of $1.5e - 5$ for both CFL and PPSS training methodologies. This was done to illustrate the trade-off between data privacy and model performance and establish the practicality of applying DP in non-IID settings. Notably, we fixed δ due to the observation that as per [109], both ϵ and δ have similar effects on the introduced

noise, with ϵ having the more pronounced impact on training performance.

The results indicate a notable decrease in CFL's performance under the influence of DP. For instance, with $\epsilon = 1$, CFL's accuracy decreased from 93.98% to 92.69%, and further to 91.14% for $\epsilon = 0.1$. Conversely, PPSS also experienced a reduction in accuracy, decreasing from 94.01% to 92.38% for $\epsilon = 1$, and to 92.18% for $\epsilon = 0.1$. This decline in accuracy is a well-recognized consequence of the introduced noise and underscores the inherent trade-off between model performance and data privacy preservation. However, our proposed PPSS exhibits greater privacy preservation than CFL, leveraging transfer learning and reducing the number of training iterations. Furthermore, our experiments demonstrate the practicality of employing PPSS in non-IID settings, showcasing its efficacy in preserving data privacy across varying scenarios.

4.4.6 PPSS Energy Cost:

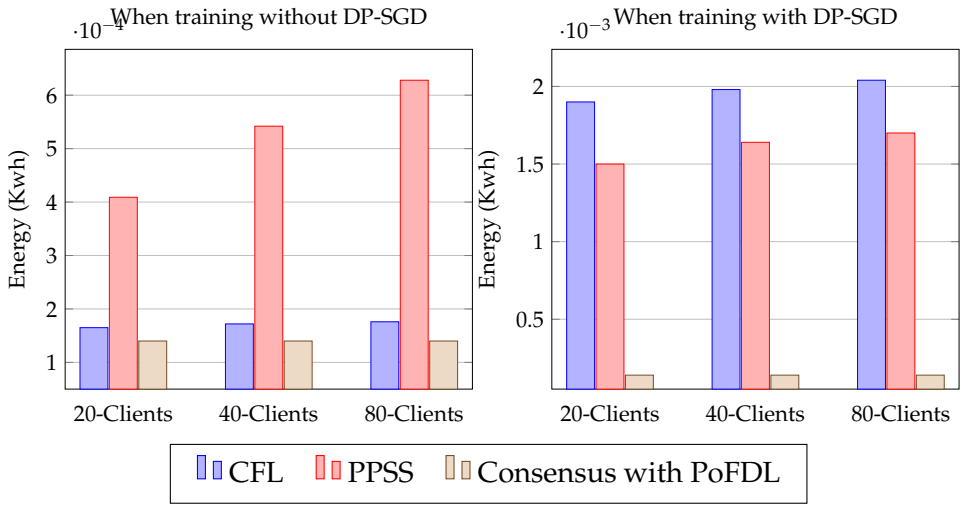


FIGURE 4.9: Average Energy Consumption of PPSS-Enabled Decentralized Federated Learning (DFL) on Tesla-T4 GPU Devices.

Figure 4.9 illustrates the average energy consumption of the proposed PPSS-enabled DFL framework over a single round of Federated Learning (FL). Both CFL and DFL, with and without the PoFDL consensus mechanism, are evaluated, considering DP-training (DP-SGD).

In the absence of DP-SGD, CFL demonstrates higher energy efficiency than PPSS due to additional model evaluation operations in PPSS. However, when DP-SGD was introduced, both CFL and PPSS experienced substantial increases in energy consumption. CFL's energy costs surge nearly tenfold, while PPSS's increase by around threefold. Consequently, PPSS proves more energy-efficient, particularly in multi-round FL scenarios with DP-training, thanks to cost-effective model aggregation, which reduces the number of training rounds compared to CFL. Notably, the energy costs of model aggregation and PoFDL consensus remain significantly lower compared to training.

Overall, we can demonstrate that PPSS excels in energy efficiency, especially in multi-round FL with DP-training, due to its efficient model aggregation and reduced training rounds compared to CFL.

4.4.7 Blockchain performance and storage overhead

Metric	3 Provers	6 Provers	8 Provers
Nb_Blocks	6	9	5
Nb_transactions	195	226	159
Storage (MB)	43.48	50.84	35.48

TABLE 4.7: The Average Data Generation Rate and Storage Overhead of the Learning-Chain.

We comprehensively evaluated our blockchain scheme, focusing on throughput, latency, and storage overhead. Latency was quantified as the between transaction tx submission and block confirmation. Throughput, on the other hand, which measures the rate at which transactions tx are confirmed, was assessed in terms of computational load and message exchanges during block confirmation. Computationally, our proposed PoFDL consensus algorithm was employed for validation, utilizing model inference with inference costs dependent on model size and the specified ValidationSet. In terms of message exchanges, our PoFDL orchestrates $2(\frac{N}{2}+1)$ message rounds, where N represents the number of *Validator* nodes.

Furthermore, based on empirical data from our experiments, we examined the storage overhead of the resulting *Learning-Chain*. Table 4.7 presents the average rate of block data generation across various *Prover* nodes. We calculated the storage capacity of each block based on the data it contained, as illustrated in Figure 4.3, without factoring in the storage overhead from validation, which is contingent on the size of the employed ValidationSet. For context, the size of the trained model used in our experiments was approximately 0.21 *MB*, while the average block size stood at 6.46 *MB*. Although the maximum storage overhead of the *Learning-Chain* for the presented learning task amounted to 50.84 MB, we anticipate that this figure may escalate with additional tasks.

Nonetheless, these findings hold relevance for real-world permissioned blockchain applications operating within the fog computing layer.

4.5 Chapter Summary

In this chapter, we proposed a novel security framework, PPSS, designed to fortify Industry 4.0/5.0 against privacy breaches and emerging cyber threats. PPSS encompasses two core components: a blockchain-enabled FL system and a privacy-preserving cyber threat detection mechanism. Within the blockchain networked model, PPSS facilitates cross-silo FL through the involvement of specific roles: *Validator* nodes ($\mathcal{V}$) serve as trusted blockchain maintainers, *Prover* nodes ($\mathcal{P}$) moderate localized FL processes and provide efficient and precise models added to the *Learning-Chain*, and *Edge-clients*, which are connected to multiple $\mathcal{P}$ nodes, engage in differential privacy-enhanced model training. On the other hand, the cyber threat detection mechanism capitalizes on PPSS's secure features to enhance the effectiveness, reliability, and efficiency of the Intrusion Detection System (IDS) in industrial IoT networks.

We comprehensively evaluated PPSS's reliability and efficiency under various scenarios and experimental settings. The findings demonstrate that our proposed framework fortifies the security and integrity of the model-sharing process, rendering it

resilient against multiple attack scenarios. Moreover, the results notably confirm that the PPSS framework exhibits adept classification skills across a wide range of attacks, considering the unique challenges posed by industrial IoT and the influence of security constraints on the FL learning process.

CHAPTER 5

FEDGEN-ID: FEDERATED DEEP GENERATIVE MODEL FOR INTRUSION DETECTION

"The five most efficient cyber defenders are Anticipation, Education, Detection, Reaction, and Resilience."

— Stephane Nappo

5.1 Introduction

Drawing upon the insights acquired from the preceding chapter, blockchain technology offers a robust framework for facilitating secure federated learning (FL) processes and enhancing the validation approach through proof of learning. However, the challenges of differential privacy training and non-IID data in cyber threat detection limit the effectiveness of FL-based models. Striking a balance between preserving privacy and maintaining model accuracy is a delicate task, especially when dealing with heterogeneous data sources with distinct threat profiles. Addressing these challenges requires innovative techniques and strategies to adapt FL to the unique characteristics of the cybersecurity domain.

Furthermore, recent studies have shown that ML and DL models are vulnerable to zero-day attacks, which exploit unknown vulnerabilities in software or hardware. These attacks create unique behaviors and attack patterns, posing a challenge in detection and identification, especially in situations with limited training data [110].

In addition, federated ML and DL models exhibit a distinct vulnerability to adversarial attacks. These attacks, which compromise model integrity and privacy, can exploit vulnerabilities during the training and inference stages. They are primarily attributed to the inaccessibility of data, which further compounds the challenges faced in securing these models [111]. During training, adversaries employ poisoning attacks to manipulate the model's learning process and compromise performance. During inference, adversaries employ evasion attacks to deceive trained models, leading to incorrect cyber threat detection.

In the preceding chapter (4.2), our proposed Privacy-Preserving Secure System (PPSS) addressed vulnerabilities within the training stages. It offered secure aggregation and authentication schemes to ensure the reliability of the aggregated model. This chapter focuses on the inference stage and aims to develop a highly efficient federated cyber threat detection framework that identifies zero-day cyber attacks while preserving data privacy and enhancing adversarial robustness against evasion attacks.

In this chapter, we introduce an innovative security framework named "Federated Generative Intrusion Detection" or "FedGen-ID," which addresses challenges related to privacy-preserving training and non-IID data distribution, contributing significantly to the robustness of cyber threat detection. Specifically, FedGen-ID is designed to enhance the security of Industrial IoT networks, which are known for their complexity and require specialized intrusion detection solutions. This framework employs FL and generative AI capabilities to address privacy concerns during model training by facilitating collaborative model development without sharing sensitive raw data. Additionally, FedGen-ID recognizes the challenges posed by non-IID data

distribution, where individual devices may have unique threat patterns. By utilizing generative techniques, the framework adapts to the specific data characteristics of each device, promoting a more consistent and effective threat detection process across the entire network. Moreover, in response to the evolving threat landscape, FedGen-ID contributes to cyber threat detection resilience by generating synthetic data that covers a broader range of potential attack scenarios, thereby improving its ability to detect previously unseen zero-day attacks, a significant concern in cybersecurity.

The remainder sections of this chapter are organized as follows: Section 5.2 discusses the design objectives of FedGen-ID, outlining its development goals and purposes. Section 5.3 delves into framework development, covering training procedures, learning objectives, and the quality of generated IDS data. Section 5.4 presents results and discussions, including the evaluation of augmented IDS data, the effectiveness of FedGen-ID in detecting adversarial attacks, and its overall performance in cyber threat detection. Finally, Section 5.5 summarizes the key takeaways and findings of FedGen-ID and its practical application in intrusion detection.

5.2 Design Objectives of FedGen-ID

Recently, Generative Adversarial Networks (GANs) have emerged as a promising approach for enhancing the robustness of optimization techniques in DL-based IDS. This advancement enables IDS to effectively detect and counter adversarial attacks without making predetermined assumptions about the capabilities of potential adversaries. Moreover, GANs can serve as a valuable tool for data augmentation, particularly in addressing the challenges associated with imbalanced and private datasets. However, the practicality and effectiveness of deploying federated GANs for threat detection and bolstering resilience against adversarial attacks are still in their early stages. Additionally, assessing the generated data's consistency, reliability, and suitability, particularly in handling IDS source data, necessitates further exploration.

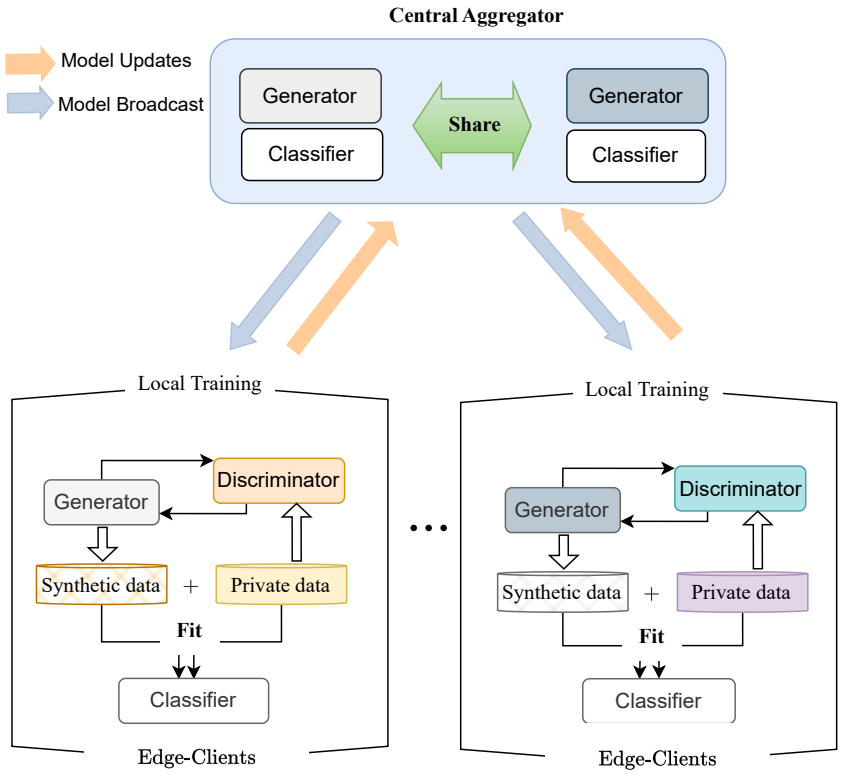


FIGURE 5.1: FedGen-ID Design Scheme: The Proposed Federated Conditional Wasserstein Generative Adversarial Network.

To address these challenges, we introduce an innovative security framework named "FedGen-ID" to enhance the efficiency of IDS, ensuring privacy protection and fortifying resilience against adversarial attacks. Simultaneously, FedGen-ID seeks to optimize the sharing of security knowledge among participating entities, contributing to a more robust and secure cyber landscape.

Figure 5.1 illustrates the workflow of our framework. Specifically, we employ FL to address privacy concerns and the computation efficiency of industrial IoT, allowing models to train on distributed data locally on user devices while only exchanging model updates. In addition, we propose a generative framework to overcome limited data, imbalanced, and non-IID data challenges and enhance adversarial resilience, allowing robust and efficient cyber threat detection. We have designed a three-model

system that includes a federated generative model (i.e., cGAN-Generator), a Discriminator model (i.e., cGAN-Critic), and a Classifier model. The federated generative model creates a variety of artificial samples, the Discriminator (D) is trained to differentiate between artificially generated and real samples, and the Classifier (C) is trained on both original and artificially generated data to efficiently and robustly identify cyber threats.

In the federation process of FedGen-ID, we suggest that both the cGAN Generator and Classifier models be shared among clients, while the cGAN Discriminator is kept on the client side. This setup is driven by the need to improve the stability and privacy protection of distributed GAN training, which is also vulnerable to adversarial attacks. By using the cGAN Discriminator locally, clients can identify and flag potential adversarial attacks for further investigation. This also enhances the communication efficiency and privacy aspects of federated learning. By sharing the Generator, clients can locally produce a variety of artificial samples and enhance their local datasets, aiding in the detection of zero-day and sophisticated adversarial attacks.

Alternatively, the global Classifier, which is shared among clients, updates that are also influenced by the synthetic samples generated using the global Generator instead of only relying on local updates contributed by individual participants. This enables the classifier to be trained on extensive and diverse datasets. As a result, the classifier can generalize well and excel in identifying a variety of attacks based on their features, offering crucial insights for threat analysis and response. Furthermore, this approach is designed to enhance the model's overall resilience and reduce the potential risks associated with learning patterns induced by attackers from poisoned updates.

5.3 Framework Development and experiments

5.3.1 Training Objectives and Algorithmic Insights

Our training goal is to reach a balance where the generator creates a variety of realistic samples. Concurrently, the critic accurately differentiates between real and generated data, offering valuable feedback to the generator to produce samples that meet the specified condition (that is, the target class label). Our implementation of the Conditional-GAN leverages the power of deep convolutional neural networks (CNNs) to efficiently extract significant features from the conditioning input samples. This approach ensures that our model is well-equipped to handle a variety of scenarios and challenges:

- **The Discriminator model (*D*)**: As shown in Figure 5.2, this model consists of four convolutional layers, each with a rectified linear unit (ReLU) activation function. It accepts both generated and real data samples and calculates the estimated Wasserstein distance between the fake and real data distributions. This serves as a loss function for training objectives, offering enhanced feedback to the generator and guiding it to produce samples that closely mimic the real data distribution while aligning with the specified conditions on target classes. Additionally, D undergoes fine-tuning for predicting adversarial attacks in the post-GAN training phase. To facilitate this, we integrate a Dense layer that applies a binary cross-entropy loss with a Sigmoid function to its outputs. This quantifies the discrepancy between the predicted and actual values of real and generated data samples. By adopting this approach, we aim to bolster the Critic's capacity to effectively discern and classify adversarial attacks.

- **The Generator model (*G*)**: As shown in Figure 5.2, this model consists of four transposed convolutional layers, each with batch normalization and a ReLU activation function. G accepts random samples drawn from a uniform latent space, denoted as $z \in \mathbb{R}^d$ where d represents the dimension of the feature. It also

Symbol	Explanation
K	The set of clients involved
I	The number of local iterations
E	The number of global epochs
m	The size of the local batch
α_G	The learning rate for the Generator
α_D	The learning rate for the Critic
λ	The penalty factor
$\mathcal{G}$	The Global Generator
$\mathcal{D}$	The Global Critic
P_z	The distribution of noise
$D(\cdot\|\cdot)$	The function of the Critic
$G(\cdot\|\cdot)$	The function of the Generator
P_r	The distribution of real data
x	A sample of real data
z	A vector of noise
y	A randomly selected label
$\tilde{x}$	An interpolated sample
$\nabla_{\tilde{x}} D(\tilde{x}\|y)$	The gradient of the output of the Critic with respect to $\tilde{x}$
$\mathcal{L}_{\text{gen}}$	The loss of the Generator
$\mathcal{L}_{\text{disc}}$	The loss of the Critic

TABLE 5.1: Notation for Algorithm Discussion.

takes a condition vector of class labels, denoted as y. The goal is to generate the necessary labeled examples. In accordance with the distribution of real data, the output of the generator is passed through a Sigmoid activation function. This maps the generated features into normalized values ranging between 0 and 1.

- **The Classifier Model (*C*)**: This is a standalone CNN model that is specifically engineered for tasks involving multi-class classification. By using augmented data during the training phase, C is able to effectively grasp the complex variations and intricacies inherent in real-world data. As a result, C exhibits a high degree of skill in identifying a diverse array of attack classes, thereby demonstrating its robustness and resilience, even in the face of adversarial attempts.

  Consequently, C demonstrates proficiency in identifying a wide range of attack classes, showcasing its robustness and resilience when manipulated with adversarial attempts.

- **Federated Learning Objective**: The aim of Federated Learning (FL) is to update

the global generator model $\mathcal{G}$, and the global Classifier $\mathcal{C}$, using K local models from respective clients. To accomplish this, we implement an averaging algorithm, which can be expressed as follows:

$$\mathcal{G} \leftarrow \frac{1}{K}\sum_{k=1}^{K} G_k, \quad \mathcal{C} \leftarrow \frac{1}{K}\sum_{k=1}^{K} C_k \tag{5.1}$$

Averaging allows for consolidating knowledge from multiple clients. This fosters collaborative learning in a distributed environment, which in turn boosts the performance of the model and its ability to generalize.

Algorithm 4 outlines the federated training procedure of FedGen-ID. This process involves several clients concurrently training their local generators and critics. Following this, they collectively update a global generator. The use of a convergence threshold can potentially decrease the training time for each client, especially if a certain level of convergence is reached early on.

Algorithm 4: FedGen-ID cGAN Training.

Input : Set of clients $\mathcal{K}$, Local iterations I, global epochs E, local batch size m, Critic's learning rate α_D, Generator's learning rate α_G, gradient penalty factor λ
Output: Trained Critic $\mathcal{D}$ and Generator $\mathcal{G}$

Initialize Generator $\mathcal{G}$ with random weights
for $r = 1$ *to* R **do**
  Parallel. For each *client* $k \in \mathcal{K}$ **for** $t = 1$ *to* E **do**
    Train Local Critic $\mathcal{D}_n$ on client n using Algorithm 6
    Train Local Generator $\mathcal{G}_n$ on client n using Algorithm 5
    if *distance between fake and real predictions* ≤ 0.1 **then**
      break // Convergence threshold
  end
  Update Global Generator $\mathcal{G}$ by averaging local generators:
  $\mathcal{G} \leftarrow \frac{1}{|\mathcal{K}|}\sum_{n=1}^{|\mathcal{K}|} \mathcal{G}_n$
  return Trained Generator $\mathcal{G}$ to Clients

- **Local Training Objective**: The training goal for the client-side cGAN involves a process of alternating updates between the critic and generator networks. We incorporated the Wasserstein loss function into the objectives of both models [112], which serves as an approximation function that quantifies the similarity between the distributions of real and generated data, based on the amount of movement required to transform one distribution into the other. The aim is to stop the generator from falling into a single mode and to ensure that the samples it generates are realistic. The definition of the Wasserstein loss is as follows:

$$\min_{G} \max_{D} \left(\mathbb{E}_{x \sim P_r}[D(x|y)] - \mathbb{E}_{z \sim P_z}[D(G(z|y))] \right) \tag{5.2}$$

where P_z represents the noise distribution and generates synthetic data samples. $D(\cdot|\cdot)$ the critic function, also known as the critic, which evaluates and distinguishes between real data samples x drawn from the real data distribution P_r and the generated samples produced by the generator function $G(\cdot|\cdot)$.
In simple terms, the critic's goal is to differentiate between a variety of real data and fake data, given the labels. Concurrently, the generator aims to deceive the critic by generating data that is as realistic as possible, based on the target labels. To enhance the stability of cGAN, we incorporated the gradient penalty (GP) into the previous loss equation. This serves as an approximation for enforcing the 1-Lipschitz continuity, ensuring that the critic's gradient norm is almost always one. The implementation of GP is as follows:

$$\min_{G} \max_{D} \left(5.2 + \lambda \cdot \frac{1}{n} \sum_{i=1}^{n} \left[\left\| \nabla_{\tilde{x}_i} D(\tilde{x}_i|y_i) \right\|_2 - 1 \right]^2 \right) \tag{5.3}$$

Where, λ is the hyper-parameter controlling the strength of the gradient penalty, $\tilde{x}_i$ is a sample randomly interpolated between real data x_i and generated data $G(z_i|y_i)$, and $\nabla_{\tilde{x}_i} D(\tilde{x}_i|y_i)$. It represents the gradient of the critic's output concerning $\tilde{x}_i$.
Furthermore, for better performance, the critic independently applies the binary

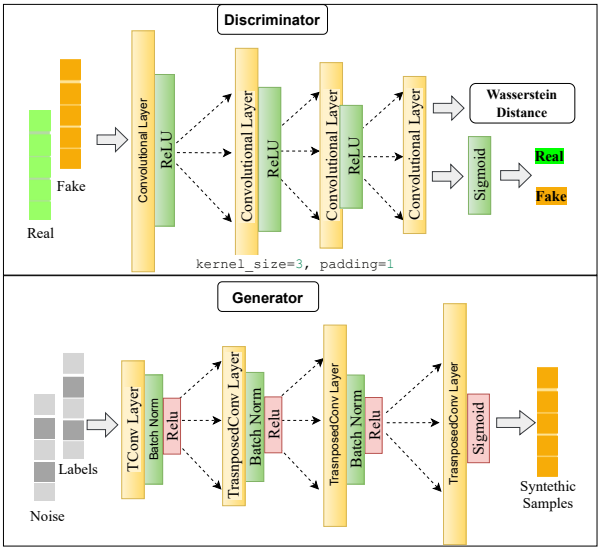


FIGURE 5.2: The Proposed Three-Model Approach for Efficient and Robust Cyber Threat Detection.

cross-entropy loss expressed as:

$$\min_{D} \left(\frac{1}{n} \sum_{i=1}^{n} [\log(D(x_i|1)) + \log(1 - D(G(z_i|y_i)|0))] \right) \tag{5.4}$$

Where $D(.|1)$ and $D(.|0)$ represent the D's prediction for the input data sample as real or fake, respectively, compared to the ground truth values (0,1).
On the other hand, for updating the classifier for multi-class classification, the objective can be formulated as:

$$\min_{C} \left(-\frac{1}{n} \sum_{i=1}^{n} \sum_{c=1}^{C} y_{i,c} \log(C(x_i)) \right) \tag{5.5}$$

where C is the classifier, x_i is the augmented data sample, $y_{i,c}$ is the ground truth label for class c, and $C(x_i)$ is the predicted probability distribution over the classes.

Algorithm 5: Training of FedGen-ID Generator.

Input : Number of local iterations I, size of local batch m, Generator's learning rate α_G, penalty factor λ
Output: The trained Generator $\mathcal{G}$

Download Generator $\mathcal{G}$ from the Server
for $i = 1$ *to* I **do**
  Draw m noise vectors $\{z_1, z_2, \ldots, z_m\}$ from the noise distribution P_z
  Obtain m random labels $\{y_1, y_2, \ldots, y_m\}$ from the clients
  Create synthetic samples:
    $\{G(z_1|y_1), G(z_2|y_2), \ldots, G(z_m|y_m)\}$
  Determine the generator loss using the Wasserstein loss:
    $\mathcal{L}_{\text{gen}} = \frac{1}{m} \sum_{i=1}^{m} D(G(z_i|y_i))$
  Adjust the weights of the Generator using gradient descent:
    $\mathcal{G} \leftarrow \mathcal{G} - \alpha_G \cdot \nabla_{\mathcal{G}} \mathcal{L}_{\text{gen}}$
return The trained Generator $\mathcal{G}$

Algorithm 6: Training of FedGen-ID Critic.

Input : Number of local iterations I, size of local batch m, Critic's learning rate α_D, penalty factor λ
Output: The trained Critic $\mathcal{D}$

Start by initializing the Critic $\mathcal{D}$ with weights chosen randomly
for $i = 1$ *to* I **do**
  Collect m real data samples $\{x_1, x_2, \ldots, x_m\}$ from the clients
  Draw m noise vectors $\{z_1, z_2, \ldots, z_m\}$ from a uniform distribution P_z
  Obtain m random labels $\{y_1, y_2, \ldots, y_m\}$ from the clients
  Create synthetic samples: $\{G(z_1|y_1), G(z_2|y_2), \ldots, G(z_m|y_m)\}$
  Draw m random interpolation factors $\{\alpha_1, \alpha_2, \ldots, \alpha_m\}$ from a uniform distribution
  Calculate interpolated samples:
    $\{\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_m\} = \alpha_i x_i + (1 - \alpha_i) G(z_i|y_i)$
  Determine the critic loss using the Wasserstein loss with gradient penalty:
    $\mathcal{L}_{\text{disc}} = \frac{1}{m} \sum_{i=1}^{m} \left[D(x_i|y_i) - D(G(z_i|y_i)) + \lambda \cdot \left(\|\nabla_{\tilde{x}_i} D(\tilde{x}_i|y_i)\|_2 - 1 \right)^2 \right]$
  Adjust the weights of the Critic using gradient descent:
    $\mathcal{D} \leftarrow \mathcal{D} - \alpha_D \cdot \nabla_{\mathcal{D}} \mathcal{L}_{\text{disc}}$

5.3.2 FedGen-ID: Quality of Generated IDS Data

The data produced by the conditional GAN necessitates further processing and validation to comply with the constraints and traffic feature boundaries of the original

Algorithm 7: Refinement of Generated Data.

Input : Original dataset O with n instances and d attributes, synthetic dataset S with the same dimensions as O and d attributes, indices R of attributes that need correction for out-of-range values, indices B of binary attributes that need correction for incorrect values, indices C of one-hot encoded attributes that need correction for incorrect values

Output: Refined synthetic dataset S' with the same dimensions as S

Procedure *RefineData* (O, S, R, B, C)

$S' \leftarrow S$ // Make a copy of synthetic data

for $i \in R$ **do**

// Correct out-of-range values

$v_{\min,i} \leftarrow \min(O_{:,i})$

$v_{\max,i} \leftarrow \max(O_{:,i})$

$S'_{:,i} \leftarrow \max(\min(S_{:,i}, v_{\max,i}), v_{\min,i})$

for $i \in B$ **do**

// Correct binary values

$S'_{\text{incorrect},i} \leftarrow (S_{:,i} \neq 0) \wedge (S_{:,i} \neq 1)$ // Identify incorrect values

$S'_{\text{corrected},i} \leftarrow \lfloor S_{\text{incorrect},i} \rceil$ // Round incorrect values to closest integer

$S'_{\text{corrected},i} \leftarrow S_{\text{corrected},i} \cdot S'_{\text{incorrect},i} + S_{:,i} \cdot (\neg S'_{\text{incorrect},i})$ // Substitute incorrect values with corrected values

for $i \in C$ **do**

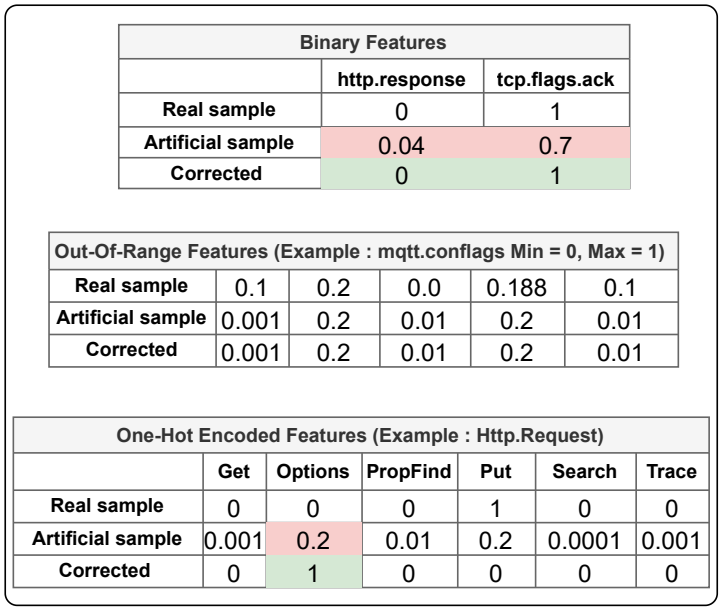
// Correct one-hot encoded values

$h_i \leftarrow \text{argmax}(S_{:,i})$ // Determine index of maximum value

$S'_{:,i} \leftarrow \boldsymbol{e}_{h_i}$ // Set all values except the maximum to 0

return S' // Return refined synthetic data

data. Algorithm 7 is designed to verify the accuracy of generated data that might contain errors or inconsistencies, especially in certain traffic feature categories. We take into account features that have out-of-range values, incorrect values for binary features, and incorrect values for one-hot encoded features. For features that are out-of-range, we identify instances where the synthetic data exceeds the valid range defined by the original data and adjust their values to fit within this range. For binary features, we correct these values by rounding them to the nearest integer. Lastly, for one-hot encoded features, the algorithm identifies the index of the highest value in the one-hot encoded feature vector and sets all other values to 0. This strategy effectively guides researchers in addressing errors and inconsistencies in synthetic data



Binary Features		
	http.response	tcp.flags.ack
Real sample	0	1
Artificial sample	0.04	0.7
Corrected	0	1

Out-Of-Range Features (Example : mqtt.conflags Min = 0, Max = 1)					
Real sample	0.1	0.2	0.0	0.188	0.1
Artificial sample	0.001	0.2	0.01	0.2	0.01
Corrected	0.001	0.2	0.01	0.2	0.01

One-Hot Encoded Features (Example : Http.Request)						
	Get	Options	PropFind	Put	Search	Trace
Real sample	0	0	0	1	0	0
Artificial sample	0.001	0.2	0.01	0.2	0.0001	0.001
Corrected	0	1	0	0	0	0

FIGURE 5.3: Example of Data Refinement for Generated Network Traffic Data.

produced by GANs for network traffic data, facilitating the creation of more consistent and reliable synthetic datasets for network-based cyber threat detection. Figure 5.3 provides a visual representation of how artificial samples are refined based on chosen features. For example, a feature such as 'mqtt.conflags' is expected to only take on the values of 0 or 1. Similarly, a feature like 'Http.Request' should fall into one of six categories. It's also important to note that features that are within the range of actual examples are kept as they are.

5.3.3 Experimental Settings

We carried out the experiments of our proposed FedGen-ID security framework on Google Collaboratory, utilizing PyTorch and Tesla-T4 GPU accelerators. The participating clients were provided with non-iid datasets, as depicted in Figure. 5.5. We initially established the federated cGAN training. Subsequently, we leverage the federated generative model to augment the training of the federated classifier model by supplying it with augmented data. Following this, we implemented DP training for the federated classifier model and evaluated the extent to which the augmented data

could alleviate the negative effects of DP. This approach ensures privacy while effectively detecting and identifying zero-day cyber threats.

It is important to note that each client retains its own critic model, which serves as a discriminator for identifying adversarial examples. The specifics of the experimental settings and learning parameters used in this study are detailed in Table 5.2. To evaluate the impact of security constraints on the learning process, we have employed a variety of metrics to evaluate both detection efficiency and effectiveness and gain insights into the performance and robustness capabilities of our proposed framework for detecting zero-day cyber threats.

In addition, we aim to explore the effects of security constraints, including distributed learning and differential privacy training, on the effectiveness of our framework.

	Parameter	**Values**
Federated cGAN	cGAN Generator	Refer to 5.2
	cGAN Critic	Refer to 5.2
	Local cGAN epochs	10
	Critic repeats for one epoch	2
	Learning rate	0.0002
	Local Batch_size	32
	Global rounds	5
Federated Classifier	Classifier	CNN 15-class
	Local Batch_size	64
	Global rounds	15
	Learning rate	0.001
Differential privacy	Epsilon (ϵ)	1
	Delta (δ)	1.5e-5
	Gradient norm bound (C)	1.2
*	Optimizer	Adam

TABLE 5.2: Experimental settings for FedGen-ID.

5.3.3.1 Dataset Processing: This framework is also the new Edge-IIoTset [33] previously discussed in Section 4.3.3.1, which exhibits characteristics of both imbalanced and non-IID. This dataset comprises fourteen labeled network attacks. The initial distribution of the dataset after the holdout split is depicted in Table 5.3. To emulate data

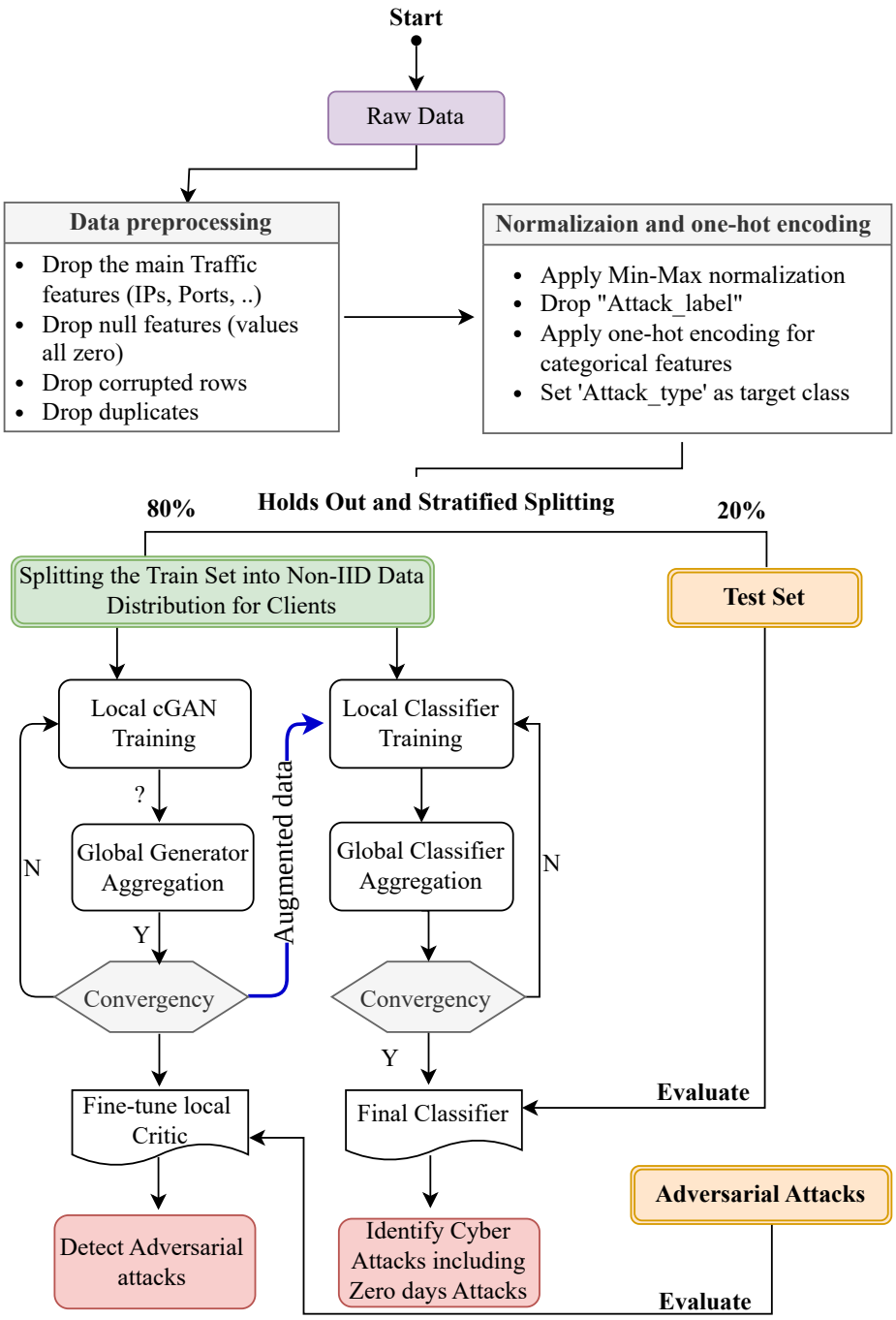


FIGURE 5.4: Flowchart of FedGen-ID Framework Training, Aggregation, and Evaluation.

heterogeneity, we partitioned the training set into non-IID partitions and distributed

Classes	Original Train Count	Original Test Count
Normal	1046926	323129
Backdoor	19890	4972
Vulnerability_scanner	40088	10022
DDoS_ICMP	93149	23287
Password	40122	10031
Port_Scanning	18051	4513
DDoS_UDP	88027	22007
Uploading	30107	7527
DDoS_HTTP	39929	9982
SQL_injection	40962	10241
Ransomware	8740	2185
DDoS_TCP	40050	10012
XSS	12732	3183
MITM	320	80
Fingerprinting	801	200

TABLE 5.3: Edge-IIoTset Data Distribution.

them among ten clients. We employed a label partition method for this purpose, ensuring that each client receives a random subset of labels with the identical feature vector of the training data. This approach is based on the assumption that each client possesses partial knowledge of the total classes involved in the problem, as shown in Figure 5.5.

5.4 Results and Discussion

5.4.1 Evaluating Convergence of Federated cGAN Training

A comprehensive set of experiments were carried out to determine the optimal hyperparameter configuration for the stability of the training process in our proposed federated cGAN scheme. Our results indicate that the stability improves when multiple local epochs are used with a fewer number of federated rounds. Figures 5.9 showcases the local training loss of the Federated cGAN, which uses the Wasserstein distance with gradient penalty (Wass-GP), reported at specific training steps. The Critic loss, which is directly related to the Wasserstein distance in both cGAN models, represents an approximation of the negative of the Wasserstein distance. As shown, the

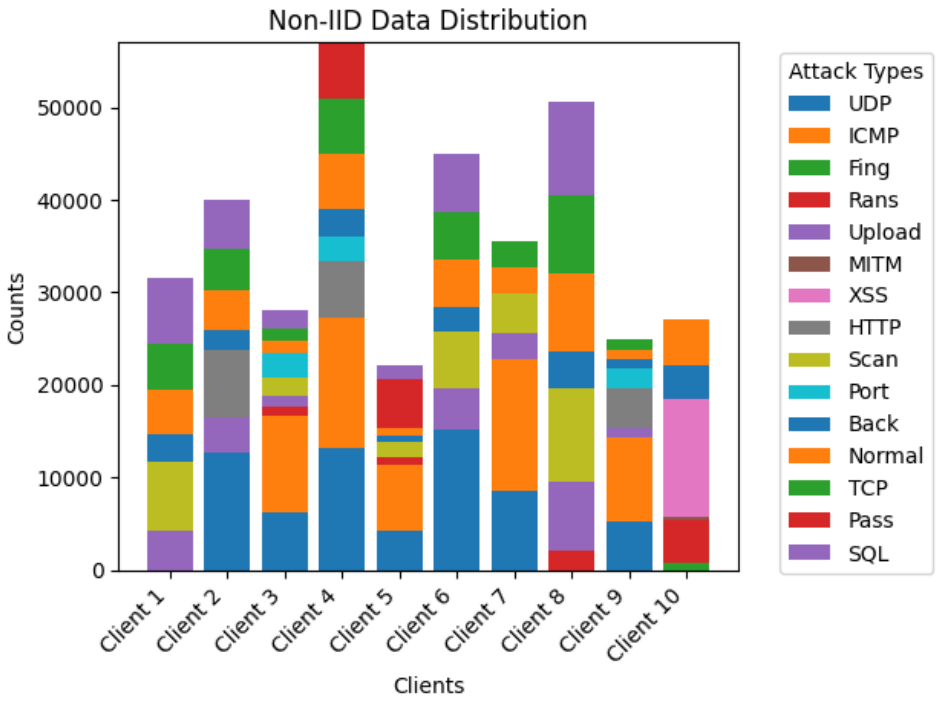


FIGURE 5.5: Non-IID Data Distribution.

Wass-GP, unlike standard loss functions, is unbounded and can produce any value. This characteristic enhances the critic without encountering the vanishing gradient issue.

Interestingly, we observe that the critic's loss begins at a relatively high value and gradually diminishes over time, indicating an enhancement in the Critic's capability to differentiate between real and generated samples. On the other hand, the generator loss starts at a lower value and slightly escalates over time. This can be attributed to the improved performance of the Critic, which sets a more challenging adversarial goal for the generator. Importantly, as the training advances, a pattern of convergence becomes apparent, where the losses associated with both the generator and Critic tend to converge towards each other.

5.4.2 Evaluating FedGen-ID for Adversarial Attack Detection

In order to reinforce the robustness and adaptability of our framework against the constantly evolving adversarial attacks, we have enhanced the ability of local critics to detect adversarial examples by adjusting their decision threshold through the application of the Sigmoid function.

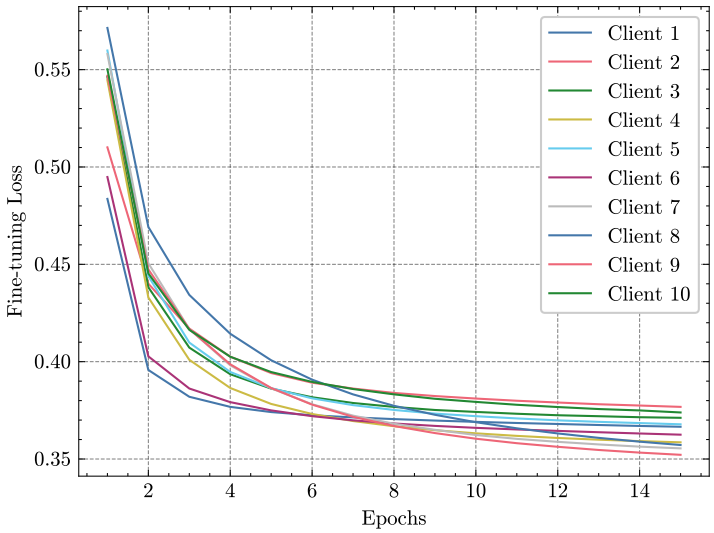


FIGURE 5.6: Loss history of fine-tuning client critics for adversarial attack detection.

It's worth noting that the Critic models were trained using the Wasserstein loss, which maximizes the distance between real and fake inputs. Therefore, if we applied an activation function, we could predict adversarial examples and evaluate them against a corresponding ground truth value. However, our cGAN-Critic models showed limitations in generalizing to other attack techniques, thereby reducing the practicality of the defense mechanism in real-world scenarios. To address this, we further fine-tuned the Critic models using data from the global generator and data from other advanced attack techniques to enhance adversarial variety.

More specifically, we added a linear layer and trained it using authentic data from the clients' datasets, combined with data from the global generator and more sophisticated attack methods, including FGSM adversarial attacks. Figure 5.6 depicts the

fine-tuning history of the Clients Critic over 15 epochs. The results are reported in Table 5.4.

5.4.3 Evaluating FedGen-ID for Data Augmentation

Notably, our proposed federated generative model (FGM) approach incorporates class-conditioned labels, which, although not immune to ensuring label accuracy, significantly contributes to enhancing data diversity. Our investigation produced a dataset comprising 50,000 instances for each distinct attack class. However, following the application of our data refinement methodology, which introduces marginal modifications to feature values, a mismatch was detected between the initially specified target classes and the resulting predicted labels upon employing the FedID classifier.

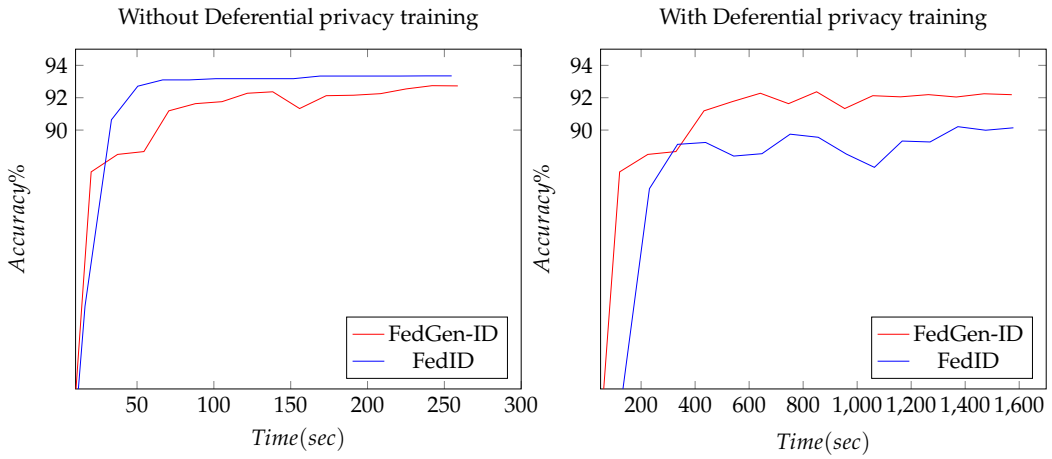


FIGURE 5.7: An Examination of Validation Accuracy in FedGen-ID Compared to Standalone FedID with and without Differential Privacy Training on the Original Test Data.

Figure 5.8 demonstrates the class distribution of generated Dataset. The results demonstrate that the approach successfully captures the underlying patterns and features of classes such as Normal, Password, Fingerprinting, XSS, and Portscanning as indicated by their relatively high sample counts.

It is worth mentioning that our approach utilizes conditioning during the generation process to label the generated samples by employing specific class targets;

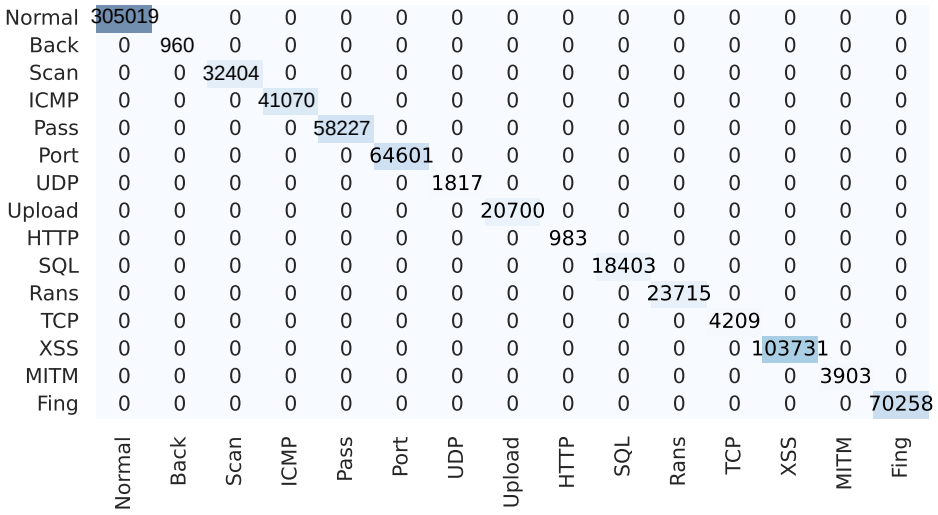


FIGURE 5.8: Confusion Matrix Depicting the Class Distribution of Generated Traffic, Labeled Using the FedID Classifier.

we observed numerous misalignments between the generated data and the intended ground truth target class. This analysis is conducted using a pre-trained classifier. In the scope of our research, we proceed with this labeling technique using the FedID classifier with 96% accuracy on the original train data to rectify the labeling discrepancies. However, it is worth noting that techniques such as self-supervised learning could be investigated in prospective studies.

These results highlight the Wasserstein conditional GAN's ability to generate synthetic data that faithfully exhibits the distinct characteristics associated with each class. However, it is worth noting that certain classes, including Backdoor, HTTP, and DDoS_UDP, exhibit relatively low counts, suggesting the presence of fewer distinctive patterns or features, posing challenges for an accurate generation. Nevertheless, by integrating these generated samples into the local training process of participating clients, we aim to enhance robustness and classification efficiency against adversarial and zero-day cyber attacks.

Table 5.4 showcases the effectiveness of our proposed individual detector against three different adversarial attacks. The table reveals that the performance of individual critics varies against different adversarial attacks. Some clients have shown high

Attacks	**Clients**	**Accuracy %**	**DR %**	**FPR %**
FGSM	Worst client : 2	92.05	98.64	15.76
	Best client 8	96.74	97.29	04.57
BIM	Worst client : 10	92.97	98.96	18.52
	Best client : 9	98.04	98.92	04.01
DeepFool	Worst client : 2	92.12	98.82	15.76
	Best client: 9	98.79	100	04.01

TABLE 5.4: Assessing the effectiveness of our proposed individual detector compared to three different adversarial attacks.

accuracy and detection rates while maintaining relatively low false positive rates. For example, under the FGSM attack, the most effective client (Client 8) achieved a detection rate of 97.29% and a false positive rate of only 4.57%. On the other hand, the least effective client (Client 2) had a false positive rate of 15.76%. Across all evaluated attacks, Client 9 consistently performed the best, demonstrating high detection rates and accuracy.

These results highlight the capability of our proposed method of refining individual critics to effectively identify complex adversarial attacks, rather than depending on a single model for defense against all attacks. Our approach stands out from traditional methods as it employs an additional classifier detection model to examine adversarial inputs that went undetected, thereby improving the overall robustness of the system. This approach underscores the importance of diversity and adaptability in building resilient defense mechanisms against adversarial attacks..

In evaluating the computational efficiency of our proposed federated generative framework (FedGen-ID), Figure 5.7 offers a comparative analysis of the training accuracy between FedGen-ID and FedID over time. This comparison considers both scenarios - with and without Differential Privacy (DP), and uses the Original Real-TestSet for the evaluation. The figure clearly shows that FedGen-ID achieves performance levels nearly equivalent to FedID without DP. Interestingly, FedGen-ID outperforms FedID in terms of performance when DP training conditions are implemented. This underscores the effectiveness and efficiency of our proposed FedGen-ID framework.

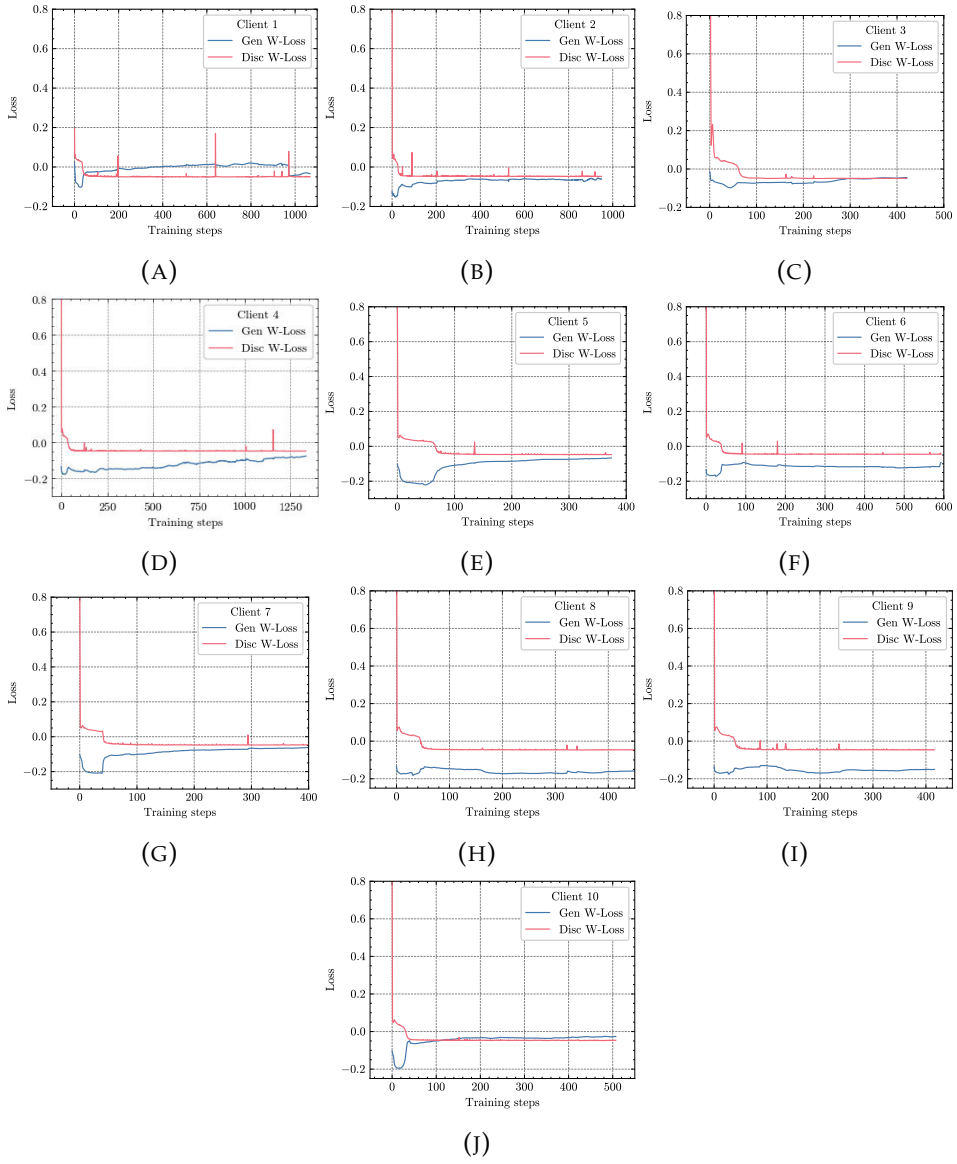


FIGURE 5.9: Training Local cGAN: Loss versus Training Steps.

Additionally, our analysis reveals that incorporating DP incurs a training overhead for both frameworks, with FedGen-ID displaying a comparatively lower increase in computational time attributable to its data augmentation approach. We

can demonstrate that our proposed FedGen-ID exhibits significantly enhanced cost-effectiveness in contrast to the implementation of DP training when considering privacy preservation. Even when both strategies boost privacy protection, FedGen-ID continues to exhibit efficacy. This further substantiates the robustness and efficiency of our proposed framework, emphasizing its practical applicability in privacy-sensitive settings.

Moreover, our examination indicates that the inclusion of DP results in a training overhead for both frameworks. However, FedGen-ID shows a relatively smaller increase in computational time, attributable to its data augmentation strategy. It's evident that our proposed FedGen-ID demonstrates considerably improved cost-effectiveness compared to the implementation of DP training, particularly in terms of privacy preservation. Even when both methods enhance privacy protection, FedGen-ID maintains its effectiveness. This further validates the robustness and efficiency of our proposed framework, underlining its practical use in settings where privacy is a priority.

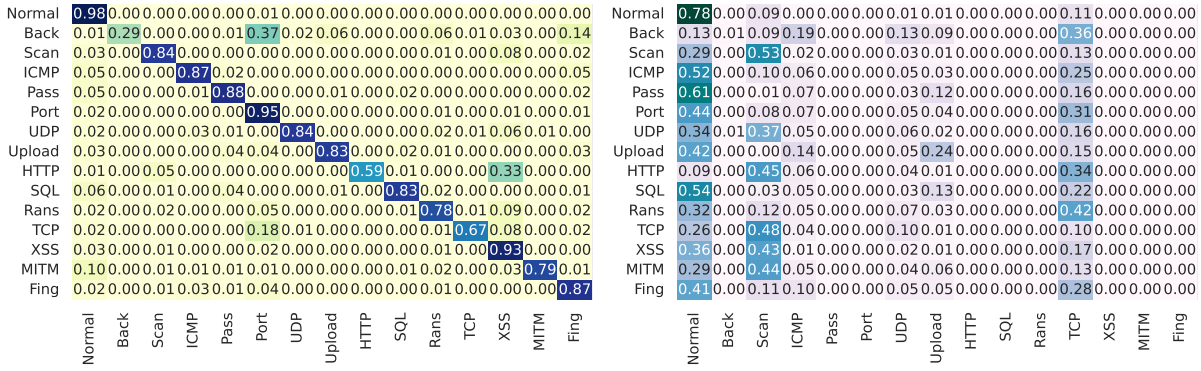
These results underscore the potential of FedGen-ID as a valuable tool for privacy-preserving FL in security-sensitive contexts. However, while FedGen-ID demonstrates promising results in accuracy, resilience, and generalization, there are specific classes where further refinement may enhance precision and recall.

Indeed, these findings highlight the potential of FedGen-ID as a valuable asset for privacy-preserving FL in contexts that are sensitive to security. However, while FedGen-ID shows encouraging outcomes in terms of accuracy, resilience, and generalization, there are certain areas where additional refinement could potentially improve precision and recall. This suggests that while FedGen-ID is a robust and efficient tool, there is always room for further enhancement to optimize its performance in various scenarios.

5.4.4 Evaluating FedGen-ID for Zero-day Attack Detection

Similarly, to determine the detection accuracy and resilience against zero-day threats. Our non-IID setup mimics the unpredictability of these threats by omitting certain

attack classes from the datasets of specific clients. Furthermore, we simulate these attacks by augmenting the TestSet to incorporate variations and new instances, utilizing the global generator. We maintained the integrity of test by ensuring that there were no duplicate records. The generated samples of zero-day attacks were labeled with their corresponding known attack labels. Figure 5.10 showcases the performance results in detecting and identifying these Zero-day attacks. Furthermore, we evaluated



(A) FedGen-ID (B) FedID

FIGURE 5.10: Classification Analysis: Visualizing Zero-Day attack detection.

the combined original TestSet and the generated samples of zero-day attacks, which were referred to as the "Augmented TestSet". Figure 5.11 presents a comparative analysis between FedGen-ID and FedID, taking into account the impact of DP training on the classification accuracy of both frameworks across both test sets. The findings underscore the potential of our proposed FedGen-ID and its capacity to uphold competitive accuracy levels across various TestSets. Specifically, FedGen-ID achieves an accuracy of 92.72% without DP-training and 92.47% with DP-training on the original TestSet. Even with a minor decrease in accuracy with DP-training, FedGen-ID maintains impressive performance. Notably, it surpasses FedID in the "Augmented-TestSet" by 14% without DP training and by 10% with DP training. These results further highlight the robustness and efficiency of FedGen-ID, particularly in privacy-sensitive settings. These findins advocate for our FedGen-ID as a robust and adaptable privacy-preserving IDS capable of tackling the ever-evolving challenges of cyber

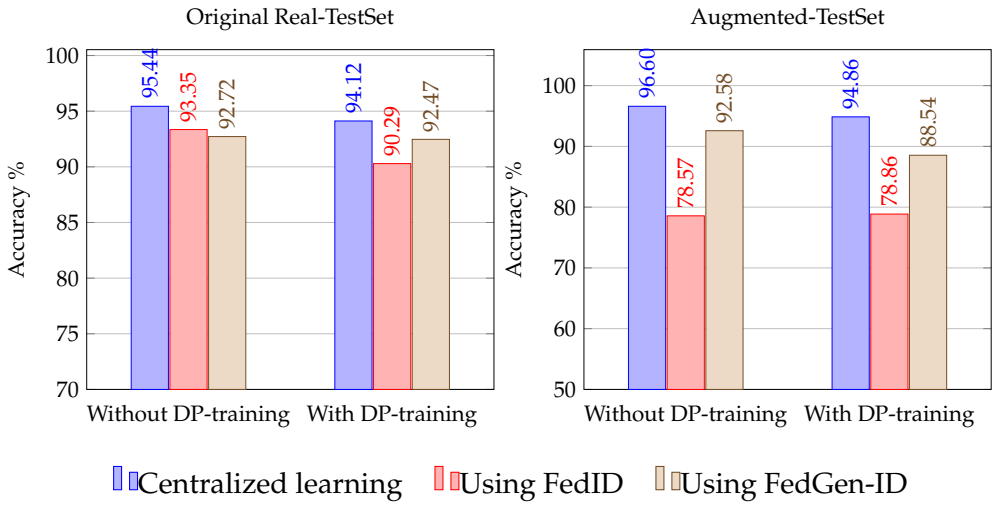


FIGURE 5.11: Comparative Analysis of Cyber Threat Detection Performance and Robustness using our proposed Federated Generative Intrusion Detection (FedGen-ID) and Standalone Federated Intrusion Detection (FedID).

threat detection in privacy-sensitive IoT environments.

5.4.5 Overall Evaluation of FedGen-ID for Cyber Attack Detection

Table 5.5 showcases the performance results for each class to evaluate the effectiveness of FedGen-ID in improving the precision and recall of detecting and identifying various cyber threats, as well as its robustness against zero-day attacks. Both FedID and FedGen-ID achieve high precision and recall without DP in detecting 'Normal' traffic for threat detection. With DP, there is a slight decrease in precision, while recall remains competitive across all experiments.

When it comes to specific attack categories, the precision and recall scores of FedGen-ID and FedID show significant differences across various privacy settings. In scenarios such as 'MITM,' 'DDoS_UDP,' 'DDoS_ICMP,' and 'Password,' FedGen-ID achieves performance levels that are nearly equivalent to or better than FedID without DP. When both privacy-enhancing strategies are combined, FedGen-ID exhibits strong performance, particularly in scenarios involving zero-day attacks. This further emphasizes the robustness and adaptability of our proposed framework.

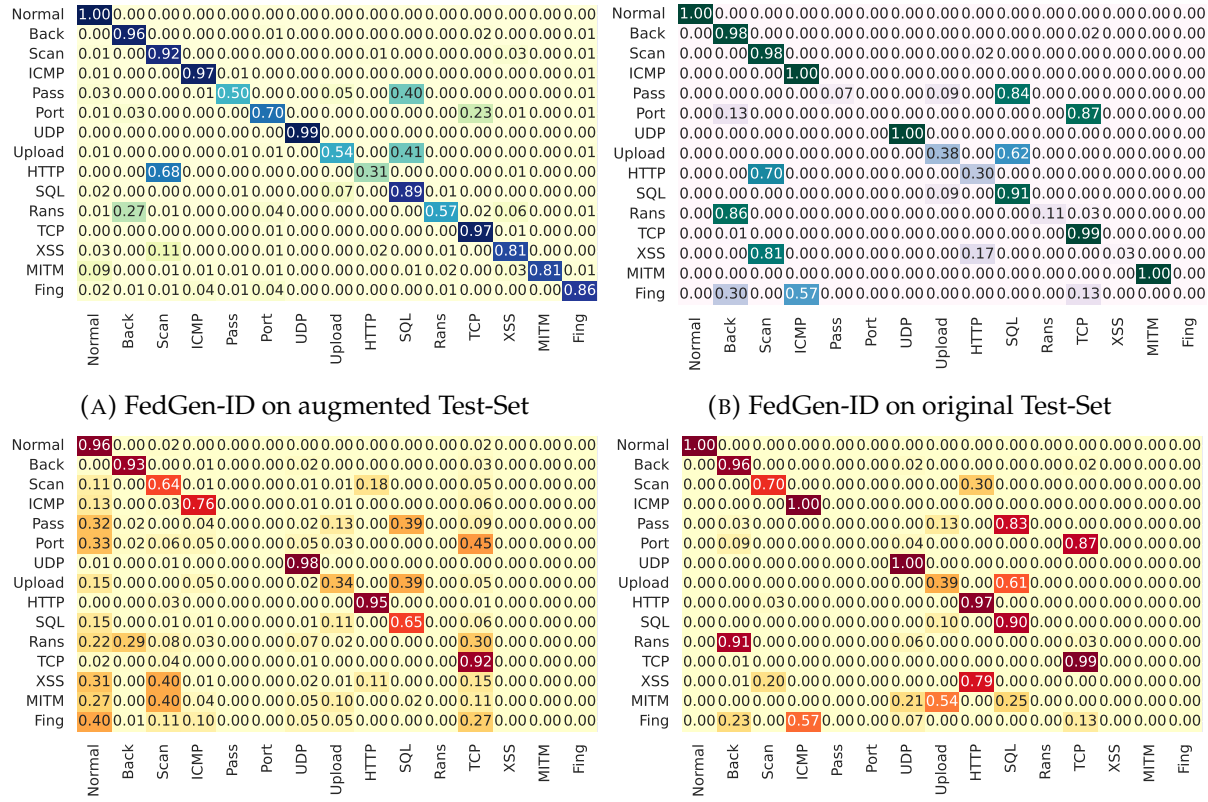
		Original TestSet				**Augmented TestSet**			
Classes	Metrics	Precision %		Detection rate%		Precision%		Detection rate%	
	Settings	FedID	FedGen-ID	FedID	FedGen-ID	FedID	FedGen-ID	FedID	FedGen-ID
Normal	No-DP	1.00	0.99	1.00	1.00	0.91	0.99	0.96	1.00
	DP	1.00	1.00	1.00	1.00	0.86	1.00	0.99	1.00
Backdoor	No-DP	0.63	0.65	0.96	0.98	0.62	0.65	0.93	0.96
	DP	0.72	0.73	0.89	0.89	0.72	0.73	0.86	0.89
Vulnerability_scan	No-DP	0.89	0.60	0.70	0.98	0.33	0.60	0.64	0.92
	DP	0.58	0.49	0.94	0.99	0.57	0.49	0.58	0.99
DDoS_ICMP	No-DP	1.00	0.97	1.00	1.00	0.83	0.97	0.76	0.97
	DP	0.97	1.00	0.98	0.99	0.90	1.00	0.73	0.99
Password	No-DP	0.00	0.94	0.00	0.07	0.00	0.94	0.00	0.50
	DP	0.00	1.00	0.00	0.07	0.00	1.00	0.00	0.07
Port_Scanning	No-DP	0.00	0.86	0.00	0.00	0.00	0.86	0.00	0.70
	DP	0.00	0.58	0.00	0.03	0.14	0.58	0.00	0.03
DDoS_UDP	No-DP	0.98	1.00	1.00	1.00	0.83	1.00	0.98	0.99
	DP	0.96	0.97	1.00	1.00	0.96	0.97	0.98	1.00
Uploading	No-DP	0.54	0.76	0.39	0.38	0.36	0.76	0.34	0.54
	DP	0.45	0.57	0.42	0.37	0.17	0.57	0.41	0.37
DDoS_HTTP	No-DP	0.64	0.79	0.97	0.30	0.64	0.79	0.95	0.31
	DP	0.75	0.87	0.52	0.25	0.49	0.87	0.51	0.25
SQL_injection	No-DP	0.41	0.48	0.90	0.91	0.41	0.48	0.65	0.89
	DP	0.40	0.41	0.82	0.90	0.40	0.41	0.59	0.90
Ransomware	No-DP	0.00	0.85	0.00	0.11	0.00	0.85	0.00	0.57
	DP	0.00	0.30	0.00	0.06	0.00	0.30	0.00	0.06
DDoS_TCP	No-DP	0.71	0.71	0.99	0.99	0.25	0.71	0.92	0.97
	DP	0.69	0.69	1.00	1.00	0.57	0.69	0.92	1.00
XSS	No-DP	0.00	0.92	0.00	0.03	0.00	0.92	0.00	0.81
	DP	0.99	1.00	0.02	0.02	0.56	1.00	0.01	0.02
MITM	No-DP	0.00	0.92	0.00	1.00	0.00	0.92	0.00	0.81
	DP	0.00	1.00	0.00	1.00	0.00	1.00	0.00	1.00
Fingerprinting	No-DP	0.00	0.90	0.00	0.00	0.00	0.90	0.00	0.86
	DP	0.00	0.00	0.00	0.00	0.12	0.00	0.00	0.00

FedID: Federated Intrusion detection; **FedGen-ID** : Federated Generative Intrusion detection; **No-DP** : No differentially private training; **DP** : with differentially private training.

TABLE 5.5: Evaluating performance across individual classes using various assessment criteria.

Figure 5.12 visually presents the confusion matrices for various settings, showcasing the performance of the FedGen-ID framework on the augmented-TestSet. These results offer valuable insights into the model's ability to classify different types of attacks and normal traffic instances.

Overall, our proposed FedGen-ID framework presented a novel contribution to federated generative intrusion detection. We demonstrated its effectiveness in tackling the challenges associated with preserving privacy, defending against zero-day and adversarial attacks, and confronting emerging cyber threats in industrial IoT applications. This underscores the potential of FedGen-ID as a robust and efficient tool



(A) FedGen-ID on augmented Test-Set

(B) FedGen-ID on original Test-Set

(C) FedID on augmented Test-Set

(D) FedID on original Test-Set

FIGURE 5.12: Classification Analysis: Visualizing Confusion Matrices.

for enhancing security in the rapidly evolving field of IoT. Its adaptability and resilience make it particularly suited for real-world applications where privacy and security are paramount.

5.5 Chapter Summary

This chapter introduces a three-model paradigm (FedGen-ID) to enhance privacy preservation and resilience against evolving cyber threats. The Federated Generative Model's primary model employs the GAN approach for data augmentation. Only generator model updates are exchanged among clients, and we introduce a novel loss function to diversify generated samples, addressing challenges posed by imbalanced and distributed data.

We also implement a data refinement method to align generated data with predefined constraints. The second model refines local Critics to enhance resilience, while the third model is a cyber threat classifier. We evaluate our FedGen-ID framework using an industrial cybersecurity dataset, demonstrating its efficiency and robustness in detection accuracy while maintaining data privacy. The results indicate that our proposed data augmentation method supports a synthetically enhanced federated learning scheme, improving detection efficiency and resilience against zero-day attacks.

CHAPTER 6
CONCLUSION AND FUTURE WORK

This chapter introduces the principal findings derived from the thesis and offers recommendations for prospective research endeavors as inspiration for researchers to embark upon novel academic investigations.

6.1 Conclusion

In this thesis, a privacy-preserving security framework was proposed for cyber threat detection in the Industrial IoT infrastructure. This framework addresses the security requirements and challenges posed by Industrial IoT ecosystems and proposes new, effective, and robust detection strategies to secure Industry 5.0 from emerging cyber threats.

First, we introduced a cost-effective and efficient federated learning methodology for malware detection targeting, with a primary focus on privacy preservation, computation cost, and detection efficiency. The results demonstrated the efficiency and effectiveness of this methodology using a CNN approach in comparison with conventional centralized methods in terms of computation cost and privacy protection. However, the detection efficiency proved inadequate when considering only

network-based statistical features. Furthermore, the inherent insecurity of the FL process, which encompasses challenges such as establishing a reliable framework for secure aggregation and validation of uploaded updates, addressing issues related to system unreliability, and ensuring the safeguarding of privacy during the model uploading process, has been a critical area of concern.

In the second part of our study, we introduced a privacy-preserving secure framework, PPSS, which seamlessly integrates blockchain technology and the energy-efficient Proof-of-Learning consensus protocol. This framework is designed to enhance the security and reliability of the FL process while promoting transparency, especially in resource-constrained industrial systems. We thoroughly evaluated the effectiveness of PPSS using a recent dataset focused on industrial cybersecurity (Edge-IIoT). Our evaluation encompassed key metrics such as detection rate, accuracy, computational efficiency, and energy consumption. The results highlight that PPSS substantially enhances the security and integrity of the model-sharing process, rendering it resilient to vulnerabilities and potential exploit scenarios. Furthermore, PPSS demonstrated impressive identification capabilities against a variety of attacks while effectively managing security constraints within the FL learning process.

Lastly, the third contribution extended the scope of FL by employing federated generative adversarial networks, FedGen-ID, and data augmentation techniques to develop a robust cyber threat detection framework for Industrial IoTs. FedGen-ID employs two approaches: the FL-based GAN approach and the FDL approach. It uses the GAN approach with a Wasserstein loss function to produce high-quality and diversified IDS data, addressing challenges posed by imbalanced and distributed data. FedGen also refines local GAN Critics to enhance resilience against adversarial attacks. In the second approach, FedGen-ID uses GAN-based augmented data to support FDL, improving detection efficiency and resilience against zero-day attacks. The results demonstrate that our proposed data augmentation method supports a synthetically enhanced federated learning scheme, improving detection efficiency and resilience against zero-day attacks.

In summary, we effectively balanced the demands of emerging technologies with the security and privacy concerns of IoT-enabled industrial infrastructure. These collective efforts underscore the importance of innovative detection strategies for countering large-scale malware attacks and ensuring the resilience of critical industrial systems against evolving cyber threats. The findings offer valuable insights to industry stakeholders, cybersecurity professionals, and researchers, enabling them to maintain the stability and security of Industry 5.0 operations.

6.2 Future work

6.2.1 Deployment of Privacy-Preserving Secure System

Our future research agenda includes the implementation of the proposed Privacy-Preserving secure framework (PPSS), broadening the scope of applicability and robustness testing on tangible IoT devices such as Raspberry Pi and other open-source platforms. Moreover, we aim to explore alternative privacy protection measures, such as homomorphic encryption, alongside other unsupervised learning methodologies. This diversified exploration promises to enrich our understanding of privacy-preserving collaborative learning and strengthen the framework's versatility in accommodating various privacy paradigms.

6.2.2 Empowering Federated Learning with Generative-AI

Future studies will focus on improving our federated generative framework using more promising approaches, such as ensemble learning for collective decision-making, and self-supervised learning methodologies for enhancing generative model capabilities.

BIBLIOGRAPHY

[1] Djallel Hamouda et al. “Intrusion Detection Systems for Industrial Internet of Things: A Survey”. In: *2021 International Conference on Theoretical and Applicative Aspects of Computer Science (ICTAACS)*. IEEE. 2021, pp. 1–8.

[2] Dinh C Nguyen et al. “Federated learning for internet of things: A comprehensive survey”. In: *IEEE Communications Surveys & Tutorials* (2021).

[3] Djallel Hamouda et al. “Android Malware Detection Based on Network Analysis and Federated Learning”. In: *Cyber Malware: Offensive and Defensive Systems*. Ed. by Iman Almomani et al. Cham: Springer International Publishing, 2024, pp. 23–39. ISBN: 978-3-031-34969-0. DOI: 10.1007/978-3-031-34969-0_2. URL: https://doi.org/10.1007/978-3-031-34969-0_2.

[4] Mohamed Amine Ferrag et al. “Generative Adversarial Networks-Driven Cyber Threat Intelligence Detection Framework for Securing Internet of Things”. In: *2023 19th International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT)*. 2023, pp. 196–200. DOI: 10.1109/DCOSS-IoT58021.2023.00042.

[5] Djallel Hamouda et al. “PPSS: A privacy-preserving secure framework using blockchain-enabled federated deep learning for Industrial IoTs”. In: *Pervasive and Mobile Computing* (2022), p. 101738.

[6] Djallel Hamouda et al. "Revolutionizing intrusion detection in industrial IoT with distributed learning and deep generative techniques". In: *Internet of Things* (2024), p. 101149. ISSN: 2542-6605. DOI: https://doi.org/10.1016/j.iot.2024.101149. URL: https://www.sciencedirect.com/science/article/pii/S254266052400091X.

[7] Pai Zheng et al. "Smart manufacturing systems for Industry 4.0: Conceptual framework, scenarios, and future perspectives". In: *Frontiers of Mechanical Engineering* 13.2 (2018), pp. 137–150.

[8] Security Boulevard. *Cybercrime to Cost Over $10 Trillion by 2025*. 2021. URL: https://securityboulevard.com/2021/03/cybercrime-to-cost-over-10-trillion-by-2025/ (visited on 08/19/2023).

[9] Abid Haleem et al. "Perspectives of cybersecurity for ameliorative Industry 4.0 era: a review-based framework". In: *Industrial Robot: the international journal of robotics research and application* 49.3 (2022), pp. 582–597.

[10] Apostolos Gerodimos et al. "IoT: Communication protocols and security threats". In: *Internet of Things and Cyber-Physical Systems* (2023).

[11] Konstantinos Tsiknas et al. "Cyber threats to industrial IoT: a survey on attacks and countermeasures". In: *IoT* 2.1 (2021), pp. 163–186.

[12] Akseer Ali Mirani et al. "Key Challenges and Emerging Technologies in Industrial IoT Architectures: A Review". In: *Sensors* 22.15 (2022), p. 5836.

[13] Mohamed Amine Ferrag et al. "Federated deep learning for cyber security in the internet of things: Concepts, applications, and experimental analysis". In: *IEEE Access* 9 (2021), pp. 138509–138542.

[14] Anjana Rajan, J Jithish, and Sriram Sankaran. "Sybil attack in IOT: Modelling and defenses". In: *2017 International conference on advances in computing, communications and informatics (ICACCI)*. IEEE. 2017, pp. 2323–2327.

[15] Mohamed Amine Ferrag et al. "Edge Learning for 6G-enabled Internet of Things: A Comprehensive Survey of Vulnerabilities, Datasets, and Defenses". In: *arXiv preprint arXiv:2306.10309* (2023).

[16] Djallel Hamouda. "Un système de détection d'intrusion pour la cybersécurité". University 8 Mai 1945, Guelma, Algeria, 2020.

[17] Mohamed Amine Ferrag et al. "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study". In: *Journal of Information Security and Applications* 50 (2020), p. 102419.

[18] Haipeng Yao et al. "Hybrid intrusion detection system for edge-based IIoT relying on machine-learning-aided detection". In: *IEEE Network* 33.5 (2019), pp. 75–81.

[19] Emre Aydogan et al. "A Central Intrusion Detection System for RPL-Based Industrial Internet of Things". In: *2019 15th IEEE International Workshop on Factory Communication Systems (WFCS)*. 2019, pp. 1–5. DOI: 10.1109/WFCS.2019.8758024.

[20] Chao Wang et al. "Anomaly detection for industrial control system based on autoencoder neural network". In: *Wireless Communications and Mobile Computing* 2020 (2020).

[21] Jiangang Shu et al. "Collaborative intrusion detection for VANETs: a deep learning-based distributed SDN approach". In: *IEEE Transactions on Intelligent Transportation Systems* (2020).

[22] Izhar Ahmed Khan et al. "HML-IDS: A hybrid-multilevel anomaly prediction approach for intrusion detection in SCADA systems". In: *IEEE Access* 7 (2019), pp. 89507–89521.

[23] Jie Chen et al. "Cyber-attack detection and countermeasure for distributed electric springs for smart grid applications". In: *IEEE Access* 10 (2022), pp. 13182–13192.

[24] Hung-Jen Liao et al. "Intrusion detection system: A comprehensive review". In: *Journal of Network and Computer Applications* 36.1 (2013), pp. 16–24.

[25] Dukka Karun Kumar Reddy et al. "Exact greedy algorithm based split finding approach for intrusion detection in fog-enabled IoT environment". In: *Journal of Information Security and Applications* 60 (2021), p. 102866.

[26] Christiana Ioannou, Andronikos Charalambus, and Vasos Vassiliou. "Decentralized Dedicated Intrusion Detection Security Agents for IoT Networks". In: *2021 17th International Conference on Distributed Computing in Sensor Systems (DCOSS)*. IEEE. 2021, pp. 414–419.

[27] Othmane Friha et al. "FELIDS: Federated learning-based intrusion detection system for agricultural Internet of Things". In: *Journal of Parallel and Distributed Computing* 165 (2022), pp. 17–31.

[28] Fan Zhang et al. "Multilayer data-driven cyber-attack detection system for industrial control systems based on network, system, and process data". In: *IEEE Transactions on Industrial Informatics* 15.7 (2019), pp. 4362–4369.

[29] Chinyang Henry Tseng et al. "A specification-based intrusion detection model for OLSR". In: *Recent Advances in Intrusion Detection: 8th International Symposium, RAID 2005, Seattle, WA, USA, September 7-9, 2005. Revised Papers 8*. Springer. 2006, pp. 330–350.

[30] Yazan Otoum and Amiya Nayak. "AS-IDS: Anomaly and Signature Based IDS for the Internet of Things". In: *Journal of Network and Systems Management* 29.3 (2021), pp. 1–26.

[31] Cheng Feng, Tingting Li, and Deeph Chana. "Multi-level anomaly detection in industrial control systems via package signatures and LSTM networks". In: *2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE. 2017, pp. 261–272.

[32] Xiaofei Wang et al. "Convergence of edge computing and deep learning: A comprehensive survey". In: *IEEE Communications Surveys & Tutorials* 22.2 (2020), pp. 869–904.

[33] Mohamed Amine Ferrag et al. "Edge-IIoTset: A New Comprehensive Realistic Cyber Security Dataset of IoT and IIoT Applications for Centralized and Federated Learning". In: *IEEE Access* 10 (2022), pp. 40281–40306.

[34] Chunjie Zhou et al. "Design and Analysis of Multimodel-Based Anomaly Intrusion Detection Systems in Industrial Process Automation". In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 45.10 (2015), pp. 1345–1360. DOI: 10.1109/TSMC.2015.2415763.

[35] VR Saraswathy, N Kasthuri, and IP Ramyadevi. "Multi-granularity approach for enhancing the performance of network intrusion detection with supervised learning". In: *2016 10th International Conference on Intelligent Systems and Control (ISCO)*. IEEE. 2016, pp. 1–7.

[36] Hyun-A Park et al. "PPIDS: privacy preserving intrusion detection system". In: *Pacific-Asia Workshop on Intelligence and Security Informatics*. Springer. 2007, pp. 269–274.

[37] Brendan McMahan et al. "Communication-efficient learning of deep networks from decentralized data". In: *Artificial intelligence and statistics*. PMLR. 2017, pp. 1273–1282.

[38] Yulin Fan et al. "Iotdefender: A federated transfer learning intrusion detection framework for 5g iot". In: *2020 IEEE 14th International Conference on Big Data Science and Engineering (BigDataSE)*. IEEE. 2020, pp. 88–95.

[39] Thien Duc Nguyen et al. "DÏoT: A federated self-learning anomaly detection system for IoT". In: *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*. IEEE. 2019, pp. 756–767.

[40] Jaspreet Kaur and Gagandeep Singh. "A Blockchain-Based Machine Learning Intrusion Detection System for Internet of Things". In: *Principles and Practice of Blockchains*. Springer, 2022, pp. 119–134.

[41] Geetanjali Rathee, Chaker Abdelaziz Kerrache, and Mohamed Amine Ferrag. "A blockchain-based intrusion detection system using viterbi algorithm and indirect trust for iiot systems". In: *Journal of Sensor and Actuator Networks* 11.4 (2022), p. 71.

[42] Hafsa Benaddi and Khalil Ibrahimi. "A review: Collaborative intrusion detection for IoT integrating the blockchain technologies". In: *2020 8th International Conference on Wireless Networks and Mobile Communications (WINCOM)*. IEEE. 2020, pp. 1–6.

[43] Mohamed Amine Ferrag and Leandros Maglaras. "DeepCoin: A novel deep learning and blockchain-based energy exchange framework for smart grids". In: *IEEE Transactions on Engineering Management* 67.4 (2019), pp. 1285–1297.

[44] Randhir Kumar et al. "Sp2f: a secured privacy-preserving framework for smart agricultural unmanned aerial vehicles". In: *Computer Networks* 187 (2021), p. 107819.

[45] Yichen Wan et al. "Privacy-preserving blockchain-enabled federated learning for B5G-Driven edge computing". In: *Computer Networks* (2021), p. 108671.

[46] Meng Hao et al. "Towards efficient and privacy-preserving federated deep learning". In: *ICC 2019-2019 IEEE international conference on communications (ICC)*. IEEE. 2019, pp. 1–6.

[47] Xudong Zhu and Hui Li. "Privacy-preserving decentralized federated deep learning". In: *Proceedings of the ACM Turing Award Celebration Conference-China*. 2021, pp. 33–38.

[48] Yang Li, Jiazheng Li, and Yi Wang. "Privacy-preserving spatiotemporal scenario generation of renewable energies: A federated deep generative learning approach". In: *IEEE Transactions on Industrial Informatics* 18.4 (2021), pp. 2310–2320.

[49] Hendra Kurniawan and Masahiro Mambo. "Homomorphic Encryption-Based Federated Privacy Preservation for Deep Active Learning". In: *Entropy* 24.11 (2022), p. 1545.

[50] Mohammad Rasouli, Tao Sun, and Ram Rajagopal. "Fedgan: Federated generative adversarial networks for distributed data". In: *arXiv preprint arXiv:2006.07228* (2020).

[51] Zhe Peng et al. "VFChain: enabling verifiable and auditable federated learning via blockchain systems". In: *IEEE Transactions on Network Science and Engineering* 9.1 (2021), pp. 173–186.

[52] Bangzhou Xin et al. "Private fl-gan: Differential privacy synthetic data generation based on federated learning". In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2020, pp. 2927–2931.

[53] Yuxia Chang, Chen Fang, Wenzhuo Sun, et al. "A blockchain-based federated learning method for smart healthcare". In: *Computational Intelligence and Neuroscience* 2021 (2021).

[54] Gaofeng Hua et al. "Blockchain-based federated learning for intelligent control in heavy haul railway". In: *IEEE Access* 8 (2020), pp. 176830–176839.

[55] Abdullah Lakhan et al. "Federated-Learning Based Privacy Preservation and Fraud-Enabled Blockchain IoMT System for Healthcare". In: *IEEE Journal of Biomedical and Health Informatics* (2022).

[56] Anik Islam, Ahmed Al Amin, and Soo Young Shin. "FBI: A federated learning-based blockchain-embedded data accumulation scheme using drones for Internet of Things". In: *IEEE Wireless Communications Letters* 11.5 (2022), pp. 972–976.

[57] Jiannan Wei et al. "A Redactable Blockchain Framework for Secure Federated Learning in Industrial Internet-of-Things". In: *IEEE Internet of Things Journal* (2022).

[58] Saurabh Singh et al. "A framework for privacy-preservation of IoT healthcare data using Federated Learning and blockchain technology". In: *Future Generation Computer Systems* 129 (2022), pp. 380–388.

[59] Hong Liu et al. "Blockchain and federated learning for collaborative intrusion detection in vehicular edge computing". In: *IEEE Transactions on Vehicular Technology* 70.6 (2021), pp. 6073–6084.

[60] Mohamed Abdel-Basset et al. "Federated Intrusion Detection in Blockchain-Based Smart Transportation Systems". In: *IEEE Transactions on Intelligent Transportation Systems* (2022).

[61] Sawsan Abdul Rahman et al. "Internet of Things intrusion Detection: Centralized, On-Device, or Federated Learning?" In: *IEEE Network* 34.6 (2020), pp. 310–317.

[62] Ruijie Zhao et al. "Intelligent intrusion detection based on federated learning aided long short-term memory". In: *Physical Communication* 42 (2020), p. 101157.

[63] Pedro Ruzafa-Alcázar et al. "Intrusion detection based on privacy-preserving federated learning for the industrial IoT". In: *IEEE Transactions on Industrial Informatics* 19.2 (2021), pp. 1145–1154.

[64] Prabhat Kumar, Govind P Gupta, and Rakesh Tripathi. "PEFL: Deep Privacy-Encoding based Federated Learning Framework for Smart Agriculture". In: *IEEE Micro* (2021).

[65] Dinesh Chowdary Attota et al. "An ensemble multi-view federated learning intrusion detection for iot". In: *IEEE Access* 9 (2021), pp. 117734–117745.

[66] Othmane Friha et al. "2DF-IDS: Decentralized and differentially private federated learning-based intrusion detection system for industrial IoT". In: *Computers & Security* 127 (2023), p. 103097.

[67] Beibei Li et al. "Federated Anomaly Detection on System Logs for the Internet of Things: A Customizable and Communication-Efficient Approach". In: *IEEE Transactions on Network and Service Management* (2022).

[68] Aliya Tabassum et al. "FEDGAN-IDS: Privacy-preserving IDS using GAN and Federated Learning". In: *Computer Communications* (2022).

[69] Felipe Bravo-Marquez, Steve Reeves, and Martin Ugarte. "Proof-of-learning: a blockchain consensus mechanism based on machine learning competitions". In: *2019 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPCON)*. IEEE. 2019, pp. 119–124.

[70] Chao Liang et al. "Intrusion detection system for the internet of things based on blockchain and multi-agent systems". In: *Electronics* 9.7 (2020), p. 1120.

[71] Chao Qiu et al. "Networking Integrated Cloud–Edge–End in IoT: A Blockchain-Assisted Collective Q-Learning Approach". In: *IEEE Internet of Things Journal* 8.16 (2020), pp. 12694–12704.

[72] Yuan Liu et al. "A Blockchain-empowered Federated Learning in Healthcare-based Cyber Physical Systems". In: *IEEE Transactions on Network Science and Engineering* (2022).

[73] Segun I Popoola et al. "Federated Deep Learning for Zero-Day Botnet Attack Detection in IoT Edge Devices". In: *IEEE Internet of Things Journal* (2021).

[74] Rahim Taheri et al. "FED-IIoT: A Robust Federated Malware Detection Architecture in Industrial IoT". In: *IEEE Transactions on Industrial Informatics* (2020). ISSN: 19410050.

[75] Jiale Zhang et al. "Poisongan: Generative poisoning attacks against federated learning in edge computing systems". In: *IEEE Internet of Things Journal* 8.5 (2020), pp. 3310–3322.

[76] G. Spathoulas, G. Theodoridis, and Georgios-Paraskevas Damiris. "Using homomorphic encryption for privacy-preserving clustering of intrusion detection alerts". In: *International Journal of Information Security* 20 (2020), pp. 347–370. DOI: 10.1007/s10207-020-00506-7.

[77] Bruno Bogaz Zarpelão et al. "A survey of intrusion detection in Internet of Things". In: *Journal of Network and Computer Applications* 84 (2017), pp. 25–37.

[78] Mahbod Tavallaee et al. "A detailed analysis of the KDD CUP 99 data set". In: *2009 IEEE symposium on computational intelligence for security and defense applications*. IEEE. 2009, pp. 1–6.

[79] Nour Moustafa and Jill Slay. "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)". In: *2015 military communications and information systems conference (MilCIS)*. IEEE. 2015, pp. 1–6.

[80] Jonathan Goh et al. "A dataset to support research in the design of secure water treatment systems". In: *International conference on critical information infrastructures security*. Springer. 2016, pp. 88–99.

[81] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A Ghorbani. "Toward generating a new intrusion detection dataset and intrusion traffic characterization." In: *ICISSp* 1 (2018), pp. 108–116.

[82] Nour Moustafa. "A new distributed architecture for evaluating AI-based security systems at the edge: Network TON_IoT datasets". In: *Sustainable Cities and Society* 72 (2021), p. 102994.

[83] Yair Meidan et al. "N-baiot—network-based detection of iot botnet attacks using deep autoencoders". In: *IEEE Pervasive Computing* 17.3 (2018), pp. 12–22.

[84] Nickolaos Koroniotis et al. "Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset". In: *Future Generation Computer Systems* 100 (2019), pp. 779–796.

[85] Ivan Vaccari et al. "MQTTset, a new dataset for machine learning techniques on MQTT". In: *Sensors* 20.22 (2020), p. 6578.

[86] Muna Al-Hawawreh, Elena Sitnikova, and Neda Aboutorab. "X-IIoTID: A connectivity- and device-agnostic intrusion dataset for industrial Internet of Things". In: *IEEE Internet of Things Journal* (2021).

[87] Maede Zolanvari et al. "Machine learning-based network vulnerability analysis of industrial Internet of Things". In: *IEEE Internet of Things Journal* 6.4 (2019), pp. 6822–6834.

[88] Ala Al-Fuqaha et al. "Internet of things: A survey on enabling technologies, protocols, and applications". In: *IEEE communications surveys & tutorials* 17.4 (2015), pp. 2347–2376.

[89] Shams Forruque Ahmed et al. "Industrial Internet of Things enabled technologies, challenges, and future directions". In: *Computers and Electrical Engineering* 110 (2023), p. 108847.

[90] Alireza Souri and Rahil Hosseini. "A state-of-the-art survey of malware detection approaches using data mining techniques". In: *Human-centric Computing and Information Sciences* 8.1 (2018), pp. 1–22.

[91] Attia Qamar, Ahmad Karim, and Victor Chang. "Mobile malware attacks: Review, taxonomy & future directions". In: *Future Generation Computer Systems* 97 (2019), pp. 887–909.

[92] Anshul Arora, Shree Garg, and Sateesh K Peddoju. "Malware detection using network traffic analysis in android based mobile devices". In: *2014 Eighth International Conference on Next Generation Mobile Apps, Services and Technologies*. IEEE. 2014, pp. 66–71.

[93] Dali Zhu et al. "DeepFlow: Deep learning-based malware detection by mining Android application for abnormal usage of sensitive data". In: *2017 IEEE symposium on computers and communications (ISCC)*. IEEE. 2017, pp. 438–443.

[94] ElMouatez Billah Karbab et al. "MalDozer: Automatic framework for android malware detection using deep learning". In: *Digital Investigation* 24 (2018), S48–S59.

[95] Giacomo Iadarola et al. "Towards an interpretable deep learning model for mobile malware detection and family identification". In: *Computers & Security* 105 (2021), p. 102198.

[96] Arash Habibi Lashkari et al. "Toward developing a systematic approach to generate benchmark android malware datasets and classification". In: *2018 International Carnahan Conference on Security Technology (ICCST)*. IEEE. 2018, pp. 1–7.

[97] Arash Habibi Lashkari et al. "Towards a network-based framework for android malware detection and characterization". In: *2017 15th Annual conference on privacy, security and trust (PST)*. IEEE. 2017, pp. 233–23309.

[98] Martin Wattenberg, Fernanda Viégas, and Ian Johnson. "How to use t-SNE effectively". In: *Distill* 1.10 (2016), e2.

[99] Giuseppina Andresini, Annalisa Appice, and Donato Malerba. "Autoencoder-based deep metric learning for network intrusion detection". In: *Information Sciences* 569 (2021), pp. 706–727.

[100] Saket Acharya, Umashankar Rawat, and Roheet Bhatnagar. "A Low Computational Cost Method for Mobile Malware Detection Using Transfer Learning and Familial Classification Using Topic Modelling". In: *Applied Computational Intelligence and Soft Computing* 2022 (2022).

[101] Hanxiao Chen et al. "Practical membership inference attack against collaborative inference in industrial IoT". In: *IEEE Transactions on Industrial Informatics* 18.1 (2020), pp. 477–487.

[102] Raymond Cheng et al. "Ekiden: A platform for confidentiality-preserving, trustworthy, and performant smart contracts". In: *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE. 2019, pp. 185–200.

[103] Stefano De Angelis et al. "PBFT vs proof-of-authority: Applying the CAP theorem to permissioned blockchain". In: (2018).

[104] Hengrui Jia et al. "Proof-of-learning: Definitions and practice". In: *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2021, pp. 1039–1056.

[105] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *nature* 521.7553 (2015), pp. 436–444.

[106] Ashkan Yousefpour et al. "Opacus: User-friendly differential privacy library in PyTorch". In: *arXiv preprint arXiv:2109.12298* (2021).

[107] Xinchi Qiu et al. "A first look into the carbon footprint of federated learning". In: *arXiv preprint arXiv:2102.07627* (2021).

[108] Lasse F Wolff Anthony, Benjamin Kanding, and Raghavendra Selvan. "Carbontracker: Tracking and predicting the carbon footprint of training deep learning models". In: *arXiv preprint arXiv:2007.03051* (2020).

[109] Martin Abadi et al. "Deep Learning with Differential Privacy". In: New York, NY, USA: Association for Computing Machinery, 2016, 308–318. ISBN: 9781450341394. DOI: 10.1145/2976749.2978318. URL: https://doi.org/10.1145/2976749.2978318.

[110] Sarah Bin Hulayyil, Shancang Li, and Lida Xu. "Machine-learning-based vulnerability detection and classification in Internet of Things device security". In: *Electronics* 12.18 (2023), p. 3927.

[111] Nuria Rodríguez-Barroso et al. "Survey on federated learning threats: Concepts, taxonomy on attacks and defences, experimental study and challenges". In: *Information Fusion* 90 (2023), pp. 148–173.

[112] Ishaan Gulrajani et al. "Improved training of wasserstein gans". In: *Advances in neural information processing systems* 30 (2017).