AI4FIDS: Multimodal Federated Intrusion Detection

Panagiotis Radoglou-Grammatikis[†‡], Pavlos S. Bouzinis[§], Ioannis Makris[§], Thomas Lagkas[¶], Vasileios Argyriou[‖], Georgios Th. Papadopoulos[**], Panagiotis Fouliras[††], George Seritan[‡‡] and Panagiotis Sarigiannidis[†]

***Abstract*—The rapid progression of smart technologies creates several advantages like enhanced connectivity, personalisation solutions and environmental sustainability. However, this revolution creates also several cyber risks. In particular, the attackers have the ability to synthesise and automate advanced attack scenarios over time, while it is evident that Artificial Intelligence (AI) allows the composition of intelligent attack vectors that can adapt in real-time to conventional countermeasures. Despite the fact that AI can also benefit defensive mechanisms, there are still functional and privacy issues that need to be resolved. First, AI requires appropriate datasets that can differ from environment to environment. In addition, these datasets usually are not available due to privacy issues. Finally, adversarial attacks have the ability to target and affect the AI-based decision-making process. Therefore, in light of the previous remarks, we provide AI4FIDS, a multimodal Intrusion Detection System (IDS) for critical infrastructures. AI4FIDS leverages Federated learning (FL) and combines multiple data sources, thus allowing cooperative intelligence across multiple domains in a private manner and minimising the impact of potential adversarial attacks. In this paper, we present in detail the architectural design and specifications of AI4FIDS, while the evaluation results demonstrate their detection performance, taking into account several datasets and aggregation strategies. Finally, based on the evaluation results, we discuss how the overall reliability and detection capabilities (in terms of detecting multi-step attack scenarios) of AI4FIDS can be improved by combining the detection outcomes of the components behind AI4FIDS.**

***Index Terms*—Artificial Intelligence, Cybersecurity, Data Heterogeneity, Federated Learning, Intrusion Detection**

[*] *This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101070450 (AI4CYBER). Disclaimer: Funded by the European Union. Views and opinions expressed are, however, those of the author(s) only and do not necessarily reflect those of the European Union or European Commission. Neither the European Union nor the European Commission can be held responsible for them.*

[†] P. Radoglou-Grammatikis and P. Sarigiannidis are with the Department of Electrical and Computer Engineering, University of Western Macedonia, Campus ZEP Kozani, 50100, Kozani, Greece - E-Mail: pradoglou@uowm.gr; psarigiannidis@uowm.gr

[‡] P. Radoglou-Grammatikis is also with K3Y Ltd, William Gladstone 31, 1000, Sofia, Bulgaria - E-Mail: pradoglou@k3y.bg

[§] P. S. Bouzinis and I. Makris are with MetaMind Innovation P.C., Kila Kozani, 50100, Kozani, Greece - E-Mail: pbouzinis@metamind.gr; makris@metamind.gr

[¶] T. Lagkas is with the Department of Computer Science, Democritus University of Thrace, Kavala Campus, 65404, Kavala, Greece - E-Mail: tlagkas@cs.duth.gr

[‖] V. Argyriou is with the Department of Networks and Digital Media, Kingston University London, Penrhyn Road, Kingston upon Thames, Surrey KT1 2EE, London, UK - E-Mail: vasileios.argyriou@kingston.ac.uk

[**] G. Th. Papadopoulos is with the Department of Informatics and Telematics, Harokopio University of Athens, Omirou 9, Tavros, GR17778, Athens, Greece - E-Mail: g.th.papadopoulos@hua.gr

[††] P. Fouliras is with the Department of Applied Informatics, University of Macedonia, 156 Egnatia Street, GR54-636, Thessaloniki, Greece - E-Mails: pfoul@uom.edu.gr

[‡‡] G. Seritan is with Electrical Engineering Faculty, Politehnica University of Bucharest, Bucharest, Romania - E-Mail: george.seritan@upb.ro

I. Introduction

Despite the fact that smart technologies, such as the Internet of Things (IoT) [1], Artificial Intelligence (AI) [2] and Beyond 5G (B5G) [3] networks offer multiple benefits, they also raise significant cybersecurity risks. In particular, the attackers have the ability to create multi-step attack scenarios that may result in disastrous consequences or even fatal accidents. According to MITRE ATT&CK, characteristic examples are C0034 - 2022 Ukraine Electric Power Attack, C0004 - CostaRicto and C0029 - Cutting Edge. These scenarios integrate a sequence of carefully designed steps that may exploit potential vulnerabilities or weaknesses against various levels of an organisation's infrastructure or system architecture. Furthermore, the attackers leverage the power of generative AI in order to synthesise sophisticated cyberattacks that may adapt to potential mitigation actions and countermeasures. Usually, these attacks try to evade or mislead conventional security mechanisms. Finally, although AI can play a significant role in the arsenal of cybersecurity, adversarial attacks can impact the decision-making processes of AI models [4].

Therefore, considering the aforementioned challenges, the presence of reliable and effective Intrusion Detection Systems (IDS) is necessary. Typically, IDS rely on predefined rules, referred to as signatures, that represent particular patterns of known cyberattacks. These signatures are compared to network and system activities in order to identify potential malicious activities and generate relevant security alerts. However, these signatures require continuous updates in order to take into account the evolving threat landscape. On the other hand, AI can play a significant role in the detection of potential cyberattacks. Both Machine Learning (ML) and Deep Learning (DL) models have already demonstrated their efficiency regarding the detection of potential cyberattacks and operational anomalies [5]. Nevertheless, both ML and DL rely on labeled datasets that vary from environment to environment. Moreover, these datasets are not usually available publicly due to privacy issues. Finally, as noted, adversarial attacks have the ability to affect their prediction performance. To this end, Federated Learning (FL) promises one of the new big steps in the era of AI, allowing knowledge sharing in a private manner. In particular, FL introduces a federated paradigm of collective intelligence where multiple FL clients train their ML/DL models locally while a federated server is responsible for aggregating the training parameters of the local models and generating a global federated model.

Driven by the previous discussion, in this paper, we introduce AI4FIDS, a multimodal FL-driven IDS that combines

four federated detection systems, namely (a) Network-based Federated Intrusion Detection System (N-FIDS), (b) Log-based Federated Intrusion Detection System (L-FIDS), (c) Operational-based Federated Intrusion Detection System (O-FIDS) and (d) Vissual-based Federated Intrusion Detection System (V-FIDS). N-FIDS is responsible for detecting potential cyberattacks, using network flow statistics. On the other hand, L-FIDS and O-FIDS rely on system logs and operational data, respectively. Finally, V-FIDS leverages visual representations for the detection process. Finally, AI4FIDS integrates the Training for Federated Intrusion Detection (T4FIDS) module, which is responsible for the federated training of the detection systems mentioned above.

- **Multimodal FL-driven intrusion detection**: A multimodal IDS is provided, leveraging FL for collective intelligence across collaborative clients and detection systems. The multimodality of the proposed IDS lies in the utilization of various types of data within the scope of intrusion detection, including network flow statistics, system logs, operational data and visual representations. Finally, the IDS design specifications and architectural structure are presented with granular abstraction layers, as specified by the C4 model [6] to ensure ease of adoption and potential for future extensions by leveraging this modular design.
- **Comparison study of FL aggregation strategies**: A comparison study of various FL aggregation strategies is conducted for different data types in the context of intrusion detection systems. Specifically, well-establised strategies such as FedAvg, FedProx, FedAdam, FedYogi, FedAdagrad are tested to assess their effectiveness within intrusion detection systems.
- **Improved Detection Capabilities and Reliability**: Based on the detection outcomes of AI4FIDS, we discuss how the overall detection reliability and capabilities (in terms of detecting multistep attack scenarios) of AI4FIDS can be enhanced by employing majority voting, weighted majority voting and time window analysis mechanisms. Majority voting can enhance detection accuracy and reduce false alarms, while temporal correlation helps identify multi-step attacks and attack vectors. The analysis includes a mathematical formulation and algorithmic description of those methods, towards leveraging and combining the detection outcomes of heterogeneous IDS, whose decisions rely on various type of data.

Therefore, the rest of this paper is organised as follows. Section II discusses similar works in this field, thus drawing the motivation behind our work and highlighting our contributions. Section III provides preliminary information regarding FL. Next, section IV describes the architecture and specifications of AI4FIDS. Finally, section V focuses on the evaluation analysis of AI4FIDS, while section VI presents how the overall reliability and detection capabilities (in terms of detecting multi-step attack scenarios) of AI4FIDS can be improved. Finally, section VII concludes this paper.

II. Related Work, Motivation and Contributions

Several works have already investigated the impact of FL in the cybersecurity sector and, more specifically, in the context of intrusion detection and prevention. Some survey papers in this field are listed in [7]–[11]. In particular, M. Alazab et al. [7] discuss the impact and role of FL in cybersecurity, discussing particular use cases, applications and challenges. Similarly, in [8], B. Ghimire and B. Rawat present a review paper regarding the advancements of FL and cybersecurity in a complementary manner. On the one hand, they discuss the role of FL in cybersecurity applications (including intrusion detection), paying special attention to IoT and Cyber-physical systems (CPS). Second, they investigate the role of cybersecurity in FL. In [9], E. M. Campos et al. provide a comprehensive survey regarding the use of FL for the IoT, discussing the role of FL-based intrusion detection and how FL can further evolve existing ML/DL-driven approaches. Finally, the authors highlight potential open issues and research directions for future work. In [10], L. Lavour et al. present a systematic literature review regarding how FL-driven IDS can be further evolved. After providing the methodological framework, the authors analyse existing works based on several criteria, such as detection mechanisms, mitigation strategies, data sources, types of federated learning, local models, aggregation methods, datasets, and communication details (such as overhead reduction and encryption measures). Based on these criteria, a relevant taxonomy of FL-driven IDS is introduced, and a comparison of existing research works is carried out. Lastly, the authors discuss open issues and research directions. In [11], S. Arisdakessian et al. introduce a survey for intrusion detection in the context of IoT, combining and discussing several technological and research areas, such as FL, game theory, social psychology and Explainable Artificial Intelligence (XAI). Based on 19 criteria, they study and analyse several works, thereby identifying research gaps regarding the aforementioned technological and research areas. On the other hand, A. Sqib et al. focus their attention on combining blockchain and FL in order to enhance federated intrusion detection strategies. After providing the necessary background, they analyse existing IDS and IPS that combine both FL and blockchain. According to this analysis, they summarise research challenges and future steps. To complete our analysis, subsequently, we focus our attention on some technical works providing remarkable FL-driven IDS.

In [12], S. I. Popoola et al. introduce an FL-driven detection system, allowing individual nodes to train Deep Neural Networks (DNNs) with their respective local network traffic data. A dedicated server receives the resulting parameters from each model, aggregates them using the Fed+ aggregation strategy, and broadcasts the aggregated parameters back to all nodes. The architecture of the DNNs consists of an input layer, two fully connected hidden layers, and an output layer. The simulation results of this proposed system demonstrate an impressive accuracy of 99.27%, precision of 97.03%, TPR of 98.06%, and an F1-score of 97.50%. These findings demonstrate the efficiency of the FL models compared to local DNNs. To determine the optimal aggregation strategy, the

authors conducted several experiments, evaluating Federated Averaging (FedAvg), Fed+, and Coordinate Median (CM). According to the evaluation results, it seems that Fed+ exceeds the other state-of-the-art aggregation strategies.

In [13], O. Friha et al. proposed FELIDS. FELIDS is an FL-driven IDS that preserves data privacy and security by training models locally while increasing the detection rate by aggregating the knowledge which was produced by training the local models of all participating devices, resulting in a global model with improved detection. In terms of architecture, the proposed system relies on a Convolutional Neural Network (CNN), which consists of pooling and fully connected layers for the pre-processing of the data, and a Recurrent Neural Network (RNN), like LSTM, for processing input sequences. Regarding the evaluation of FELIDS, CSE-CIC-IDS2018, MQTTset and InSDN datasets are used, while the experimental results demonstrate the efficiency of FELIDS over centralised approaches.

In [14], R. Zhao et al. introduce an FL-driven IDS, relying on Long Short-Term Memory (LSTM) networks. The primary goal of the authors is to identify high-risk malicious behaviour, including activities such as directory traversal attacks, bulk reading and deletion of files, and bulk software uninstallation. In terms of implementation, a Bidirectional LSTM (BiLSTM) – a two-way LSTM network is deployed to all clients. The SEA dataset is utilised for the federated training process. Each local model receives user commands as input, undergoes tokenisation during preprocessing, and feeds the results into the forward and backward LSTMs. Subsequently, a dropout layer is introduced to randomly deactivate a fraction of neurons during training, preventing overfitting. The clients send their training parameters to the server, which aggregates them using a weighted average method to update the parameters of the global model. This updated global model is sent back to the users. The initial comparison between the BiLSTM and a Convolutional Neural Network (CNN) shows that the former presents higher accuracy and lower loss. Furthermore, the comparison analysis between the FL Bi-LSTM (FL-LSTM) and a Centralised Bi-LSTM (CL-LSTM) demonstrates that the FL-LSTM model achieves better performance.

In [15], the authors presented Fed-ANIDS for the detection of network intrusions, which utilizes various type of autoencoders and leverages the reconstruction error to classify network traffic as malicious or benign. Regarding the experimental setup of FL, all clients consist of a local discriminator, a local decoder, and a local encoder, while a server consists of a global encoder, decoder, and discriminator. Fed-ANIDS was tested on USTC-TFC2016, CIC-IDS2017, and CSE-CS3-CIC-IDS2018 datasets, while different aggregation strategies were also evaluated (FedAvg, FedProx). The results showed that FedProx achieved better results compared to FedAvg in the majority of the datasets and metrics.

The authors in [16] evaluate the utilization of FL in the context of IDS, by testing an artificial neural network for the identification of network attacks. The datasets that the proposed method used was ToN IoT and CICIDS2017, while the performance metrics were accuracy, precision, recall, and F1-score. Regarding the experimental results, FedAvg, FedAdam, FedAdagrad, and FedAvgM were tested, and it showed that FedAvg and FedAvgM performed better than the two adaptive algorithms, with an exception on CICIDS2017, where FedAdahtad achieved 90% in all evaluation metrics.

In [17], the authors examined FL in the domain of IoT for cyber threats identification. In particular, they presented an experimental evaluation that relied on real-world settings and utilized a distributed FL-based IDS. Regarding the experimental setup, the TON-Iot dataset was used. Two different AI models were evaluated, namely a deep belief network and a DNN, and three different aggregation strategies were tested (FedAvg, FedProx, FedYogi). The results showcased that the FL-based IDS, has no significant performance gap from the centralized IDS performance.

In [18], G. Shingi et al. highlight that due to the different nature of each network's data, a single model cannot fit all cases. For this reason, they propose a Segmented FL (Segmented-FL) framework, in which similar networks are grouped (segmentation) by periodically evaluating local models. The global model aggregates the local models' parameters, utilising a weighted average algorithm based on the size of the dataset in each network. Regarding the evaluation of Segmented-FL, it seems that the proposed solution outperforms centralised and traditional approaches, utilising the CIDDS-001 and CIDDS-002 datasets.

Undoubtedly, the previous research endeavors provide invaluable insights, practical solutions, and methodological frameworks for integrating the collective intelligence of FL into IDS. However, it is worth noting that the majority of current solutions mainly focus on network traffic data without considering other data types and sources. Furthermore, it is important to highlight that many of these works rely on outdated datasets, which may not be suitable for smart environments such as industrial energy settings, the finance sector, and healthcare ecosystems. In addition, the current solutions typically adopt FedAvg as their aggregation method without exploring other or custom aggregation strategies that may result in better detection efficiency. Finally, it is worth mentioning that the current implementations do not take into account the effects of potential adversarial attacks. Therefore, considering the previous points, in this paper we provide the first release of AI4FIDS, a multimodal IDS, integrating multiple FL-driven IDS that process data types from multiple sources. emphasis is given to the modular design of AI4FIDS, where the role of each module and the interconnections between them are outlined to provide insight into the system's architectural specifications. It is noted that this approach, which emphasizes system design, is generally absent from the existing literature.Regarding the evaluation process, adequate security datasets are used, while several aggregation methods are taken into consideration with the goal to tackle data heterogeneity issues. Additionally, extensions for integrating the outputs of different types of IDS are proposed, utilizing majority voting and temporal correlation. On the one hand, majority voting can improve detection accuracy and minimize false positives, since it relies on multiple heterogeneous detections. On the other hand, temporal correlation aids in identifying multi-step attacks and attack vectors. Finally, these

methods facilitate the generation of a unified decision, based on the outputs of different IDS.

III. Overview of Federated Learning

We consider an FL environment consisting of N clients, indexed as $i \in \mathcal{N} = \{1, 2, ..., N\}$ and a server. Each client owns a dataset $\mathcal{D}_i = \{(\boldsymbol{x}_i^j, y_i^j) \in \mathbb{R}^S \times \mathbb{C}\}_{j=1}^{D_i}$, where $\boldsymbol{x}_i^j$ is the j-th input sample, $D_i = |\mathcal{D}_i|$ is the number of samples and S denotes the number of features. Additionally, we denote $\mathbb{C}$ as the set to which the label y_i^j belongs, e.g., it could be a subset of the real numbers, a set of categorical values for classification tasks, etc. In this paper, $\mathbb{C}$ contains the labels of cyberattacks and will be described below in this work, along with the description of the datasets used in the evaluation experiments.

The overall dataset across all clients is denoted as $\mathcal{D} = \bigcup_{i \in \mathcal{N}} \mathcal{D}_i$ and the size of all training data is $D = \sum_{n=i}^{N} D_i$. The loss function of client i, is defined as:

$$F_i(\boldsymbol{w}) \triangleq \frac{1}{D_i} \sum_{j=1}^{D_i} \phi\left(\boldsymbol{w}, \boldsymbol{x}_i^j, y_i^j\right), \quad \forall i \in \mathcal{N}, \tag{1}$$

where $\phi(\boldsymbol{w}, \boldsymbol{x}_i^j, y_i^j)$ captures the error of model parameter $\boldsymbol{w}$ for the input-output pair $(\boldsymbol{x}_i^j, y_i^j)$. The ultimate goal of the FL process is to obtain the global parameter $\boldsymbol{w}$, which minimises the loss function on the whole dataset.

$$F(\boldsymbol{w}) = \sum_{n=1}^{N} n_i F_i(\boldsymbol{w}), \tag{2}$$

where $n_i = \frac{D_i}{D}$ is the proportion of data samples owned by client i relative to the entire dataset. In a nutshell, the FL process is executed for a specified number of communication rounds. At the t-th round, the server firstly broadcasts the global model $\boldsymbol{w}^{(t)}$ to all clients. Each client i updates its local model $\boldsymbol{w}_i^{(t)}$ via a gradient-based method on the loss function F_i and uploads it to the server. Finally, the server generates the global model $\boldsymbol{w}^{(t+1)}$ by using an aggregation strategy of its preference. The aforementioned process is repeated for the selected number of rounds until the convergence of the global model is achieved. As depicted in Fig. 1, the federated training procedure is conducted as follows.

Step #1 - Federated Model Initialisation: A federated training starts with the Federated Server initialising the initial federated model that will be distributed to the participating Federated Clients that will train these local data. It is important to note that this step can take place in two different ways. On the one hand, the Federated Server may know the architecture and the number of parameters of the federated model. In this case, the Federated Server does not require any information from the Federated Clients. On the other hand, the Federated Server may not have prior knowledge of the architecture and the parameters of the federated model. Therefore, in this case, the Federated Server has to ask the Federated Clients about the architectural schema and parameters of the federated model.

Step #2 - Local Training: The participating clients receive the federated model and start training a local model with their own local data. It is worth mentioning that each Federated Client has the ability to adjust training parameters like batch size, epochs and optimiser.

Step #3 - Parameter Update: After completing the local training, the parameters of the local models are transmitted to the Federated Server. Encryption and anonymisation techniques can be used in order to further protect the identity and characteristics of the Federated Clients. To further enhance privacy, differential privacy techniques may be employed for the transmission of the local models.

Step #4 - Aggregation: After receiving the parameters from the Federated Clients, the Federated Server is responsible for aggregating them, thus creating a global model. For this purpose, various aggregation methods can be applied, with the default method being FedAvg, which considers clients' contributions based on the proportion of their datasets.

Step #5 - Federated Model Update: After the parameter aggregation process, the global federated model is broadcast to the Federated Clients, who then proceed with the Local Training step.

IV. AI4FIDS Architecture and Specifications

To define the architecture and specifications of AI4FIDS, we leverage the abstraction layers of the C4 model [6]. This framework is usually adopted in software engineering for visualising and documenting the architecture of software systems. It was created by Simon Brown and stands for (a) Context, (b) Containers, (c) Components, and (d) Code, that represent the different levels of abstraction in the model. In this paper, we focus on the context, containers and components of AI4FIDS, while an open version of the code will be provided through the AI4CYBER project. Therefore, Fig. 2 illustrates the Context level of AI4FIDS, providing the relationship of AI4FIDS with external and internal entities. First, as a multi-datasource-based IDS, AI4FIDS retrieves various kinds of data from a Critical System, such as network traffic, system logs, and operational data. This data is generated through the interaction of the Critical System (which is under inspection by AI4FIDS) with external End Users and External Networks/Systems (such as the Internet). Next, AI4FIDS is responsible for analysing this data and detecting potential cyberattacks and anomalies. Based on the detection outcomes, AI4FIDS then sends the corresponding security events to another external system called Security Information and Event Management (SIEM). The primary purpose of a SIEM system is to enhance an organisation's security posture by providing real-time visibility into security incidents and threats, facilitating incident detection and response, and helping organisations comply with security regulations and policies. More specifically, a SIEM system is responsible for normalising, correlating, and prioritising the AI4FIDS security events. Finally, the System Security Operator can monitor, analyse and assess the AI4FIDS security events through the SIEM analysis.

Next, while the Context level provides a high-level overview of the system's interactions with external entities, the Container level delves deeper into the architecture of AI4FIDS. In particular, the Container level structures the architecture of the system (i.e., AI4FIDS) into logical entities that may

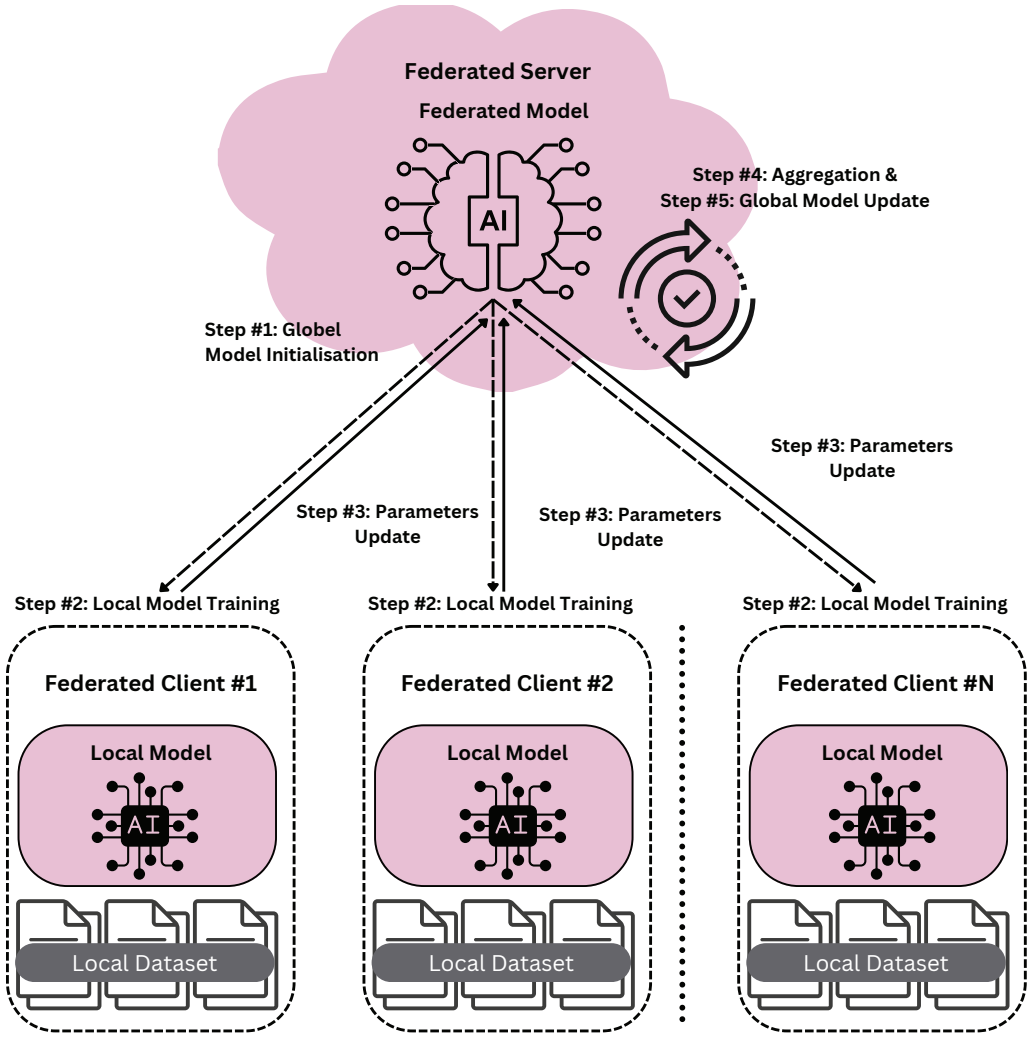

Fig. 1. Federated Learning Workflow

communicate with each other and with external entities. Therefore, as illustrated in Fig. 3, AI4FIDS is composed of five main containers: (a) L-FIDS, (b) O-FIDS, (c) N-FIDS, (d) V-FIDS and T4FIDS. First, L-FIDS receives system logs through an integration bus and is responsible for analysing them and detecting potential cyberattacks. Similarly, O-FIDS receives operational data from the integration bus and recognises potential cyberattacks and operational anomalies. On the other hand, both N-FIDS and V-FIDS capture the network traffic data of the underlying Critical System and detect potential cyberattacks through network flow statistics and binary representations, respectively. In the first case, L-FIDS and O-FIDS use an integration bus that may rely on asynchronous technologies like Apache Kafka. In contrast, both N-FIDS and V-FIDS use `tcpdump` to capture the network traffic data. Depending on the nature, the functional characteristics and the available data of the Critical System, one or more of the aforementioned containers could be used, respectively. More technical information regarding the technologies of each container is presented in Fig. 4 regarding the Component level of AI4FIDS. It is worth mentioning that the detection process of the previous containers relies on pre-trained AI models that are generated in an offline and federated manner by T4FIDS. More specifically, in each case, a federated model is generated by T4FIDS, which is responsible for orchestrating and carrying out the federated training procedure in a decentralised way across multiple data sources. Python and Flower framework are used for this purpose. Finally, based on the detection results, L-FIDS, O-FIDS, N-FIDS and V-FIDS use the integration bus to send their security events to SIEM.

Building upon the Context and Container levels, the Component level structures further each container, providing their architectural components and communications. Additionally, the core technologies for each component are provided, while L-FIDS, O-FIDS, N-FIDS, and V-FIDS follow a similar architectural design.

First, L-FIDS is composed of four components, namely: (a) Log Collection Module, (b) Data Preprocessing Module, (c) INF-Detection Engine and (d) Security Event Generation and Notification Module. The Log Collection Module is responsible for retrieving the system logs from the integration bus. These logs may include, among others, memory usage, disk and process-scheduling activities, as well as CPU-related information. Next, the Data Preprocessing Module receives and

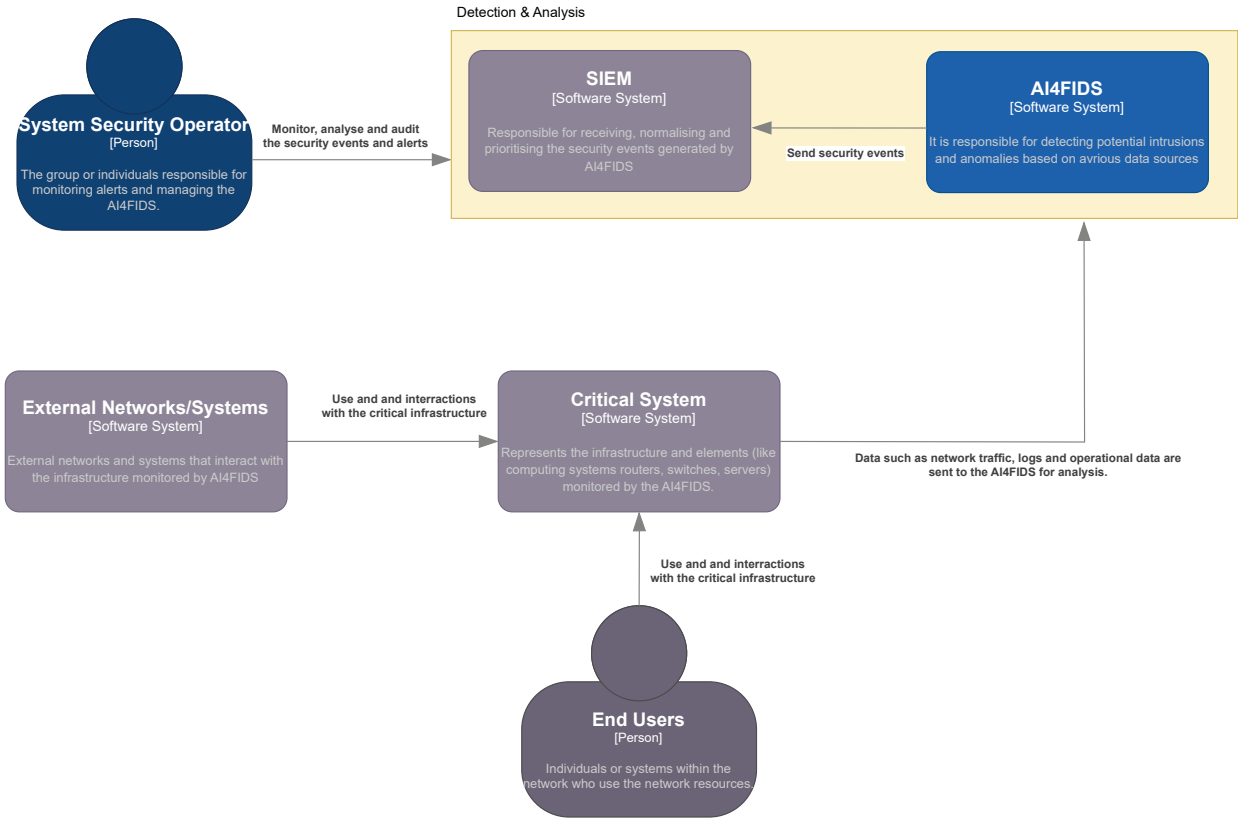

Fig. 2. AI4FIDS Context Level

preprocesses this data in terms of data cleaning, normalisation, and label encoding utilising Python packages, such as Numpy and Pandas. This preprocessing step is essential for ensuring that the data is in a consistent form and ready for further analysis. Then, the INF-Detection Engine is responsible for the online inference task, receiving the preprocessed data with their formatting complying to that of the Data Preprocessing Module, loading the pre-trained federated model and identifying potential cyberattacks. Finally, based on the detection events, the Security Event Generation and Normalisation Module creates and publishes the corresponding security events to the integration bus. The structure of O-FIDS is identical to L-FIDS. However, in this case, the Data Collection Module of O-FIDS retrieves operational data instead of system logs.

In what follows, N-FIDS comprises the following components: (a) Network Traffic Capturing Module, (b) Flow Statistics Generation Module, (c) INF-Detection Engine, and (d) Security Event Generation and Notification Module. It is noted that the components (c) and (d) serve a role similar to that of L-FIDS and O-FIDS. In contrast to the previous container setups, the Network Traffic Capturing Module is responsible for capturing inbound and outbound traffic of the network and storing it in pcap files. Subsequently, the Flow Statistics Generation Module processes the network traffic data and generates flow statistics (e.g., mean packet inter-arrival time, total backward and forward packets, etc.) of the bi-directional traffic captured by the previous module. The statistics serve as input features for the local FL training phase, carried out by T4FIDS.

Finally, V-FIDS comprises five components: (a) Network Traffic Capturing Module, (b) Network Flow Extraction Module, (c) Visualisation Module, (d) INF-Detection Engine, and (e) Security Event Generation and Notification Module. The roles of the Network Traffic Capturing Module and Security Event Generation and Notification Module remain identical to those in the previous containers. However, the Network Flow Extraction Module is responsible for receiving network traffic data (i.e., pcap files) from the previous component and organising them into flows, resulting in multiple pcap files, which is achieved using the pcap-splitter tool. The Visualisation Module then takes these pcap files and transforms them into visual representations. Specifically, each byte from the pcap files is translated into a pixel, following a colour scheme: (a) Black: 00, (b) White: FF, (c) Blue: representing printable characters, and (d) Red: everything else. Consequently, each pixel is placed on a two-dimensional visual representation, taking into consideration the proximity of binary elements. Binary elements that are close within the pcap files are positioned as closely as possible on the two-dimensional representation, with the Hilbert Curve employed to arrange the pixels within the image. The Hilbert Curve is part of the family of recursive Space-Filling Curves (SFCs), which divide a space into multiple segments and visit those segments in a specific order. SFCs, also known as Peano curves, transform data from one-dimensional space into an n-dimensional space while preserving the properties of the original data. The scope of

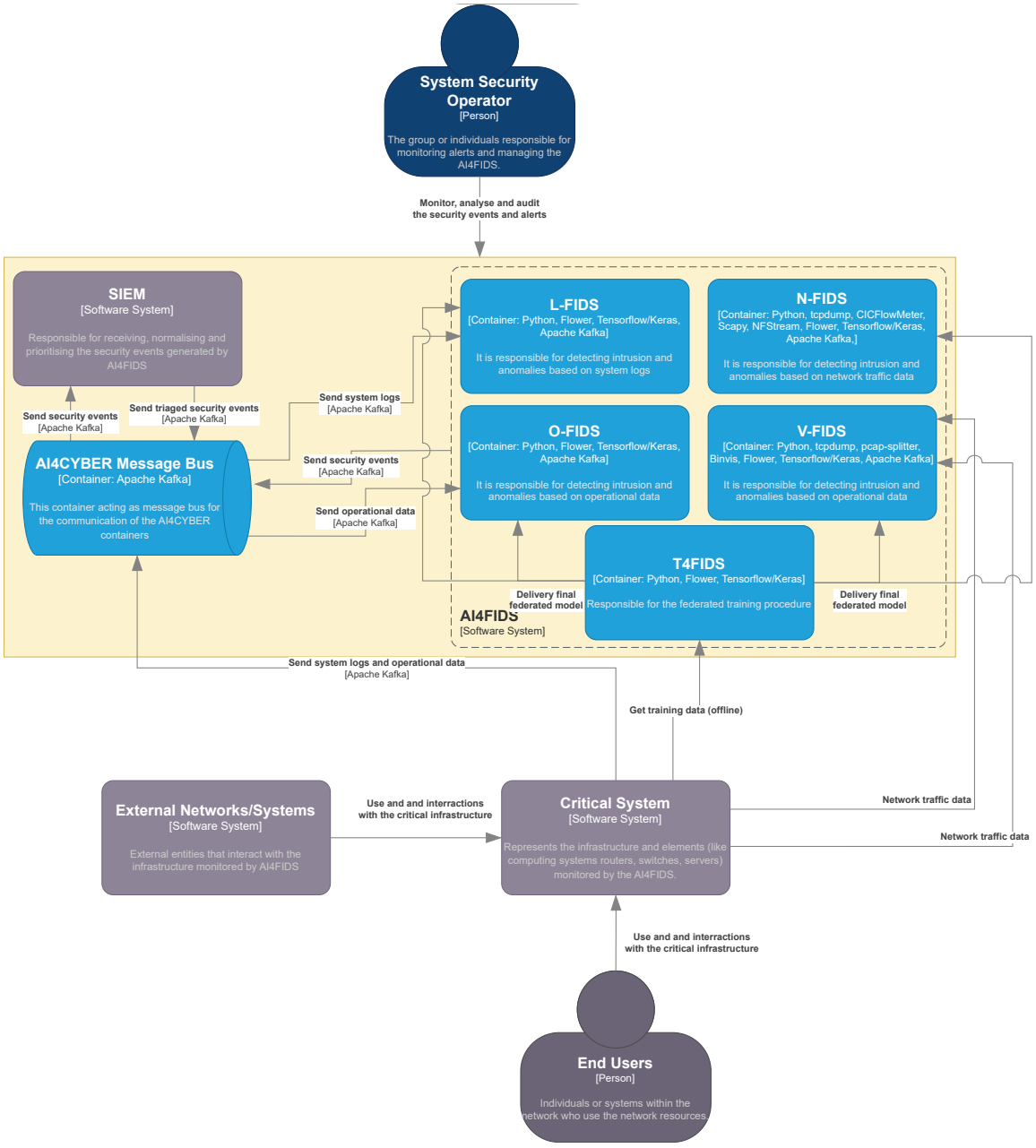

Fig. 3. AI4FIDS Container Level

SFC encompasses the two-dimensional unit square and, more generally, an n-dimensional unit hypercube. Therefore, a two-dimensional unit square corresponds to a visual representation of n × n pixels, and the Hilbert curve represents a continuous curve for each unit square (i.e., pixel of the image). The Hilbert Curve is a space-filling fractal curve that was introduced by the German mathematician David Hilbert in 1891. It is one of the earliest examples of a continuous, self-replicating curve that can fill a two-dimensional space. The Hilbert Curve is constructed by recursively subdividing a square into smaller squares and then connecting their corners with a single continuous curve. The process starts with a single square, and in each iteration, that square is divided into four smaller squares. The curve then traverses the smaller squares in a specific order, creating a path that fills the entire space within the original square. One of the most remarkable properties of the Hilbert Curve is that it can completely cover any 2D area, making it a space-filling curve. In this process, each byte of the binary pcap file is mapped to a specific colour according to a colour scheme as described above. Subsequently, the Hilbert curve is employed to convert the one-dimensional data into a two-dimensional visual representation. Next, the INF-Detection Engine takes the responsibility of AI inference, receiving the visual representations, loading the appropriate pre-trained federated model, and detecting potential cyberattacks. Finally, through the integration bus, the Security Event Generation and Notification Module generates and publishes the corresponding security events.

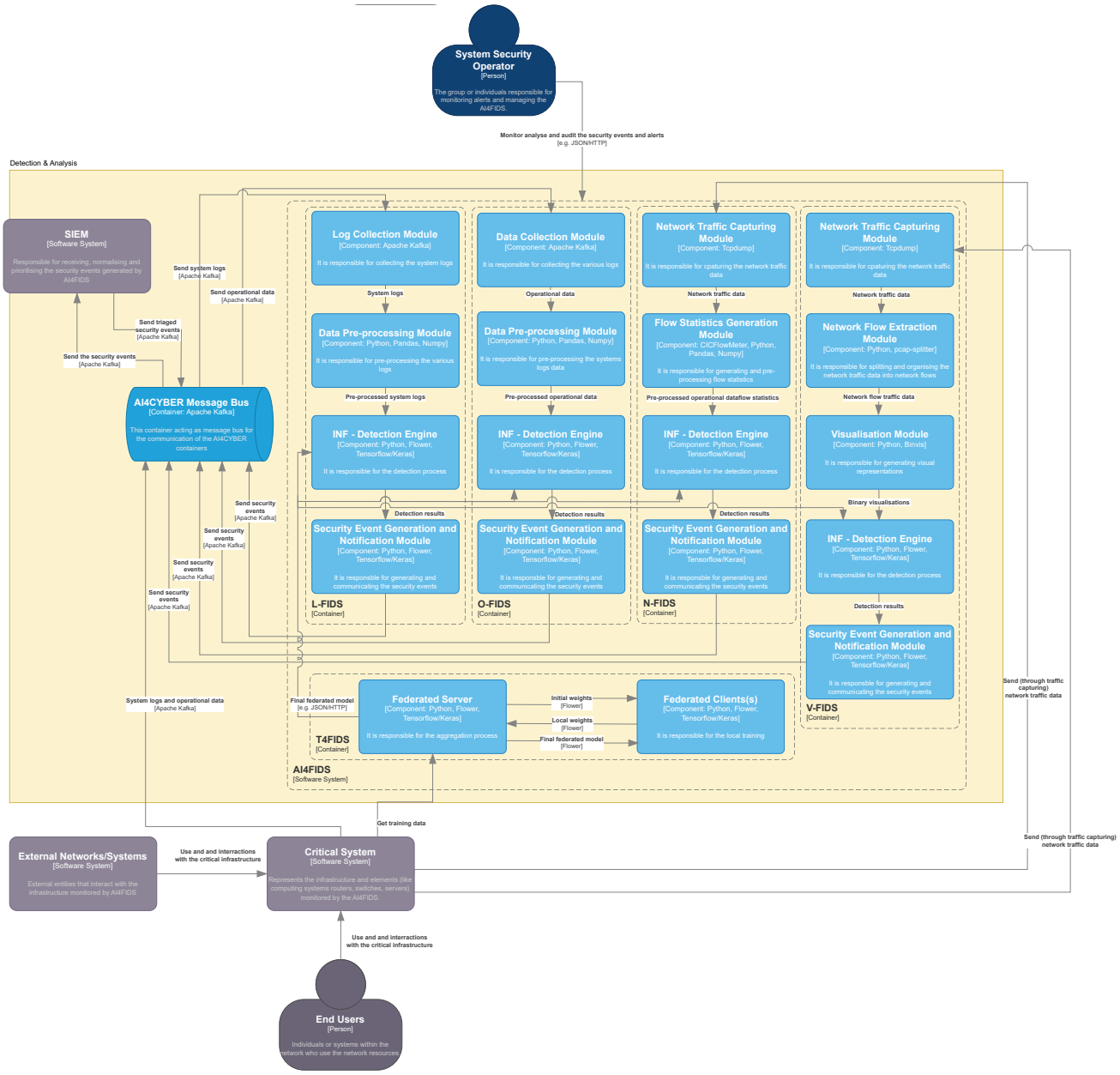

Fig. 4. AI4FIDS Component Level

Finally, T4FIDS is composed of two components: (a) Federated Server and (b) Federated Client(s) that together carry out the federated training procedure as illustrated in Fig. 1 and described in section III. It is worth mentioning that T4FIDS generates the global FL model, which is utilised by the INF-Detection Engine to detect intrusions in real-time during inference phase. Finally, it is noted that the model architecture, which depends on the number of features and the type of attacks, is also specified in T4FIDS.

V. Evaluation Analysis

This section aims to showcase and evaluate the detection efficiency of AI4FIDS. For this purpose, the following subsections summarise the datasets, metrics and the aggregation strategies used within the scope of the evaluation analysis. Finally, the evaluation results are presented, including the respective corresponding confusion matrices.

A. Evaluation Datasets and Metrics

Four security datasets are employed: (a) TON IoT Dataset [19], (b) CSE CIC IDS 2018 Dataset [20], (c) CIC IoT Dataset 2022 [21] and (d) UOWM Modbus/TCP Intrusion Detection Dataset. In particular, the TON IoT Dataset is used to evaluate the detection efficiency of L-FIDS and O-FIDS. Next, the CSE CIC IDS 2018 Dataset and CIC IoT Dataset 2022 are utilised for the evaluation process of N-FIDS. Finally, the UOWM Modbus/TCP Intrusion Detection Dataset (which is under publication by the authors) is used to validate the effectiveness of V-FIDS. On the other hand, five evaluation metrics are used to calculate the detection performance of AI4FIDS. These

metrics are provided through the Equations 3-7. True Positive (TP) indicates the number of malicious samples classified correctly. Similarly, True Negative (TN) denotes the number of benign instances that are categorised correctly. On the other hand, False Negative (FN) implies the number of malicious data samples that are identified mistakenly as normal cases. Finally, False Positive (FP) denotes the number of benign instances classified as cyberattacks or operational anomalies. Next, the metrics used within our evaluation analysis are discussed.

Accuracy calculates the ratio between the data samples classified correctly and the total instances.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

TPR indicates the ratio of the malicious samples detected successfully as cyberattacks or anomalies.

$$TPR = \frac{TP}{TP + FN} \tag{4}$$

FPR expresses the ratio of the normal instances recognised mistakenly as cyberattacks or operational anomalies.

$$FPR = \frac{FP}{FP + TN} \tag{5}$$

The F1 score calculates the golden ratio between TPR and Precision. Precision is measured by dividing TP by the sum of TP and TN.

$$F1 = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{6}$$

$$AUC = \int_a^b TPR(FPR^{-1}(x))dx = P(X_1 > X_0) \tag{7}$$

Where X_1 is the score for a positive instance (i.e., malicious instances) and X_0 is the score for a negative instance (i.e., normal cases). The AUC represents the probability that AI4FIDS will rank a randomly chosen positive instance higher than a randomly chosen negative one.

B. Aggregation Strategies

Five aggregation strategies are employed within the scope of a comparison study. These aggregation strategies are summarised below.

FedAvg [22] represents a generalisation of Federated Stochastic Gradient Descent (FedSGD), which, in turn, is a federated adaptation of the conventional Stochastic Gradient Descent (SGD). The primary distinctions between these two fundamental fusion techniques lie in the number of locally performed SGD steps on each client and the nature of the data collected on the aggregation server. In FedSGD, each participating client executes a single SGD step during each federated training round. Conversely, with FedAvg, each participating worker conducts one or more SGD steps in each federated training round. Once all the SGD steps are completed, each client transmits the updated parameters (weights and biases) of its model to the federated server. To elaborate, the federated server initiates the process by sending its initial model parameters to the participating federated clients. These clients then undertake several steps of SGD to update their local model parameters. Upon the completion of local model training and the transmission of resulting parameters to the federated server, the aggregation of updated models occurs server-side. The federated server calculates the updated global parameters by employing a weighted average of the collected parameters. Finally, the resulting aggregated global model parameters are transmitted back to the clients. This entire procedure constitutes a round of training within a FL environment. FedAvg is influenced by three key parameters: the first being the fraction c, which represents the proportion of available clients participating in a federated training round. Notably, if c equals 1, all clients partake in the federated training round. The second parameter concerns the number of local epochs, determining how many epochs each client will perform for updating its local model parameters. The third parameter is the batch size (B) utilized for client model updates.

FedProx [23] serves as an extension of FedAvg, aiming to utilize all available clients—where FedAvg opts for a subset—while also ensuring convergence, a guarantee not provided by FedAvg. Clients often exhibit diverse constraints, such as limited resources in terms of hardware capabilities, network connection reliability, and battery status. FedProx accommodates varying degrees of local workload across devices based on their system resources and averages the solutions received from each client. To prevent divergence, FedProx introduces a proximal term $h_i(\boldsymbol{w}; \boldsymbol{w}^{(t)}) = F_i(\boldsymbol{w}) + \frac{\mu}{2}\|\boldsymbol{w} - \boldsymbol{w}^{(t)}\|^2$, effectively curbing the impact of variable local updates. Rather than seeking the minimum of the local function F_i, client i locally employs its chosen solver to approximate the minimum of $h_i(\boldsymbol{w}; \boldsymbol{w}^{(t)})$. The constant penalty μ influences convergence, with FedProx exhibiting behaviour akin to FedAvg when $\mu = 0$. By selecting an appropriate μ, $h_i(\boldsymbol{w}; \boldsymbol{w}^{(t)})$ becomes convex if F_i is non-convex; additionally, when F_i is convex, h_i becomes μ-strongly convex.

FedAdam [24] takes advantage of the strengths of Adam and AdaGrad, the famous adaptive optimizers. Similar to Adam, FedAdam employs a moving average of squared gradients to dynamically modify the learning rate, preventing it from reaching extremes that might destabilize the training process. Nevertheless, in contrast to Adam, FedAdam incorporates a second-moving average of gradients to monitor the training process's progression. This feature enables FedAdam to make more assertive adjustments to the learning rate, facilitating accelerated convergence. Through two decay parameters, controls the importance that the algorithm will give to historical updates and the importance that will be given to current model updates.

FedAdagrad [24] represents a tailored iteration of the AdaGrad optimizer expressly crafted for FL. It incorporates a per-parameter learning rate that dynamically adapts in response to the gradients' magnitudes. This adaptive adjustment mitigates the risk of the learning rate becoming excessively large or small, thereby averting potential instability in the training process. This strategy performs the aggregation based on the difference between each client model and the server's global

model.

FedYogi [24] facilitates the famous optimisation algorithm, Yogi, which is focused on non-convex optimization problems. FedYogi is a strategy that aggregates the clients' models using the distance they have from the server's model, the direction of this difference (sign), and a decay parameter.

C. Experimental Setup & Evaluation Results

This subsection describes the evaluation results of AI4FIDS. In particular, Table I summarises the experimental results of L-FIDS with the TON IoT Dataset, focusing on the detection of five cyberattacks, namely (a) Denial of Service (DoS), (b) Distributed DoS (DDoS), (c) injection, (d) password brute-forcing and (e) Man-In-The-Middle (MITM). Before the federated training procedure, typical preprocessing steps like data cleaning, label encoding, and features' normalisation with StandardScaler took place. The aggregation strategies mentioned above are evaluated in the context of a comparison study, while it is worth mentioning that for each aggregation strategy, 20 training rounds with three epochs took place. Based on the evaluation results, the best detection performance is achieved through FedProx with $ACC = 82.29\%$, $TPR = 69.75\%$, $FPR = 12.74\%$, $F1 = 83.79\%$ and $AUC = 95.82\%$.

TABLE I
EVALUATION RESULTS OF L-FIDS WITH TON IOT DATASET - COMPARISON OF AGGREGATION STRATEGIES

Strategy	ACC	TPR	FPR	F1	AUC
FedAvg	36.65%	32.30%	12.74%	44.72%	71.41%
FedProx	82.29%	69.75%	3.29%	83.79%	95.82%
FedAdam	45.31%	43.90%	10.64%	52.23%	77.75%
FedAdagrad	57.50%	53.13%	8.36%	63.97%	81.12%
FedYogi	43.47%	41.29%	11.75%	52.86%	75.85%

Similarly, Table II summarises the evaluation results of O-FIDS with the TON IoT Dataset. In this case, seven cyberattacks are considered: (a) backdoor, (b) injection, (c) password brute-forcing, (d) DDoS, (e) ransomware, (f) Cross-Site Scripting (XSS) and (e) scanning. Typically, data cleaning, label encoding, one-hot encoding, and feature normalisation took place before starting the federated training procedure through T4FIDS. All the aggregation strategies mentioned above are evaluated in the scope of a comparison study, while it is worth noting in this case that after several experiments, the best results for all the aggregation strategies are achieved after ten federated training rounds with five local epochs. However, the best performance is accomplished by FedAdam with $ACC = 67.22\%$, $TPR = 24.77\%$, $FPR = 4.5\%$, $F1 = 62.40\%$ and $AUC = 83.59\%$, while the confusion matrix of this federated model is depicted in Fig. II.

Table III shows the evaluation results of N-FIDS with the TON IoT Dataset. Nine cyberattacks are considered by this dataset: (a) DoS, (b) DDoS, (c) injection, (d) password brute-forcing, (e) scanning, (f) XSS, (g) MITM, (h) ransomware and (i) backdoor. Data cleaning, one-hot encoding, label encoding, StandardScale and Synthetic Minority Over-sampling Technique (SMOTE) are applied before the federated training process. Ten federated training rounds with five epochs were carried out for all the aggregation strategies discussed previously. In this case, the best detection effectiveness is performed by FevAvg with $ACC = 76.53\%$, $TPR = 57.56\%$, $FPR = 3.78\%$, $F1 = 75.55\%$, $AUC = 95.69\%$, while the confusion matrix of the respective federated model is illustrated in Fig. 6.

TABLE II
EVALUATION RESULTS OF O-FIDS WITH TON IOT DATASET - COMPARISON OF AGGREGATION STRATEGIES

Strategy	ACC	TPR	FPR	F1	AUC
FedAvg	60.47%	24.33%	5.03%	58.92%	83.50%
FedProx	59.87%	23.81%	5.51%	57.02%	81.51%
FedAdam	67.22%	24.77%	4.50%	62.40%	83.59%
FedAdagrad	59.62%	25.13%	4.50%	58.76%	83.50%
FedYogi	67.13%	25.00%	4.40%	61.58%	83.69%

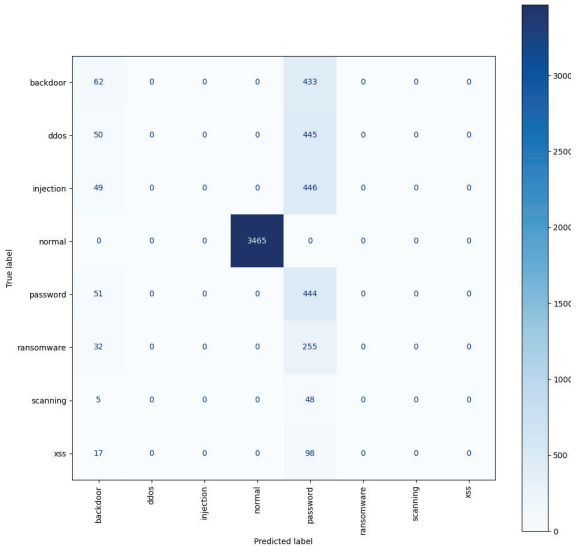

Fig. 5. Confusion matrix of the FL model (trained with TON IoT Dataset and FedYogi) behind O-FIDS

TABLE III
EVALUATION RESULTS OF N-FIDS WITH TON IOT DATASET - COMPARISON OF AGGREGATION STRATEGIES

Strategy	ACC	TPR	FPR	F1	AUC
FedAvg	76.53%	57.56%	3.78%	75.55%	95.69%
FedProx	75.93%	58.12%	4.08%	75.35%	94.89%
FedAdam	31.20%	25.44%	7.47%	37.13%	68.69%
FedAdagrad	32.16%	33.33%	6.98%	40.04%	85.66%
FedYogi	19.95%	16.47%	8.08%	29.98%	79.24%

N-FIDS was also evaluated with the CSE CIC IDS 2018 dataset as summarised in Table IV. This dataset includes 13 cyberattacks: (a) DDoS with HOIC, (b) DoS with Hulk, (c) Bot, (d) File Transport Protocol (FTP) bruteforce, (c) Secure Shell (SSH) bruteforce, (d) infiltration, (e) DoS with SlowHTTPTest, (d) DoS with GoldenEyem (e) DoS with Slowloris, (f) DDoS with LOIC, (g) Web bruteforce, (h) bruteforce-XSS and (i) SQL injection. Similarly, data cleaning,

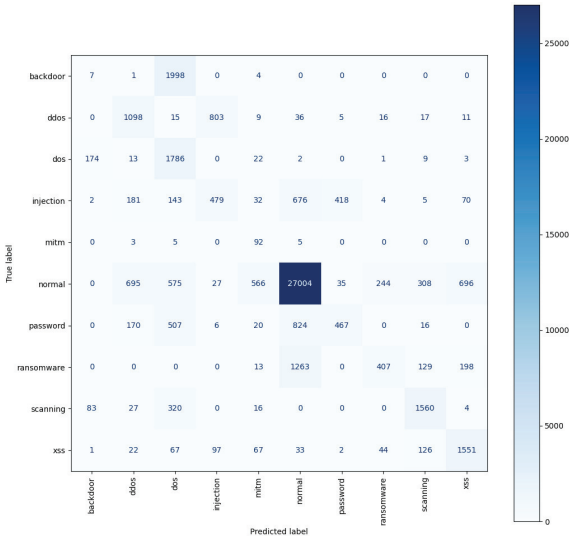

Fig. 6. Confusion matrix of the FL model (trained with TON IoT Dataset and FedProx) behind N-FIDS

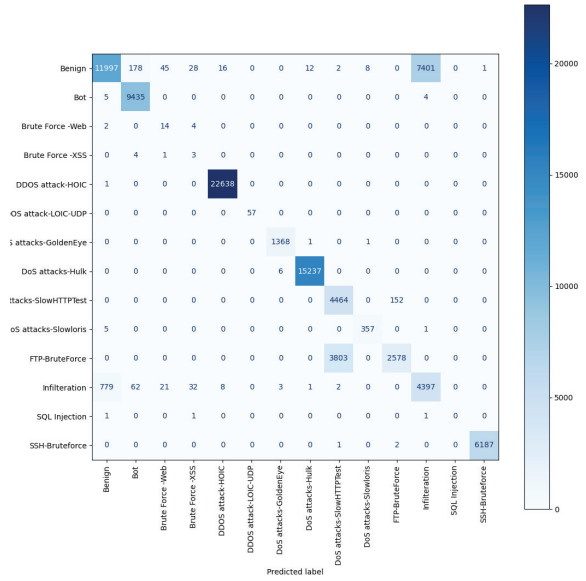

Fig. 7. Confusion matrix of the FL model (trained with CSE CIC IDS 2018 and FedAvg) behind N-FIDS

StandardScaler, SMOTE and label encoding are used before starting the federated training procedure that includes, in this case, 30 federated training rounds and two local epochs. According to the evaluation results, the best performance is carried out through FedProx with 86.73%, 78.68%, 1.01%, 87.42%, 98.18%, while the confusion matrix of this federated model is provided by Fig. 7.

TABLE IV
EVALUATION RESULTS OF N-FIDS WITH CSE CIC IDS 2018 DATASET - COMPARISON OF AGGREGATION STRATEGIES

Strategy	ACC	TPR	FPR	F1	AUC
FedAvg	84.97%	80.96%	1.13%	85.80%	98.60%
FedProx	86.73%	78.68%	1.01%	87.42%	98.17%
FedAdam	27.80%	35.66%	5.52%	28.11%	77.83%
FedAdagrad	85.66%	74.28%	1.10%	86.19%	97.86%
FedYogi	74.93%	71.01%	1.86%	77.22%	95.41%

As summarised in Table V, N-FIDS is also evaluated with the CIC IoT 2022 dataset comprising two cyberattacks: (a) flood attacks and (b) brute force attacks. Data cleaning and StandardScaler are employed prior to the federated training process, which is composed of five training rounds with two local epochs. Based on the evaluation results, in this case, FedAvg achieves the best detection performance with $ACC = 97.60\%$, $TPR = 97.60\%$, $FPR = 1.19\%$, $F1 = 97.51\%$, $AUC = 99.47\%$. The confusion matrix of the respective federated model is illustrated in Fig. 8.

Finally, Table V provides the evaluation results of V-FIDS with the UOWM Modbus/TCP Intrusion Detection Dataset. This dataset involves 14 cyberattacks: (a) modbus/dos/writeSingleCoils, (b) modbus/dos/writeSingleRegister, (c) modbus/function/readCoils, (d) modbus/function/readCoils (DoS), (e) modbus/function/readDiscreteInput, (f) modbus/function/readDiscreteInputs (DoS), (g) modbus/function/readHoldingRegister, (h) modbus/function/readHoldingRegister (DoS), (i) modbus/function/readInputRegister, (j) modbus/function/readInputRegister (DoS), (k) modbus/function/writeSingleCoils, (l) modbus/function/writeSingleRegister, (m) modbus/scanner/getfunc and (n) modbus/scanner/uid. The federated training procedure, in this case, includes five training rounds with three local epochs, while the best detection efficiency is achieved by FedAvg with $ACC = 0.984$, $TPR = 0.885$, $FPR = 0.008$, $F1 = 0.885$ and $AUC = 0.923$.

TABLE V
EVALUATION RESULTS OF N-FIDS WITH CIC IOT DATASET 2022 - COMPARISON OF AGGREGATION STRATEGIES

Strategy	ACC	TPR	FPR	F1	AUC
FedAvg	97.60%	97.60%	1.19%	97.51%	99.47%
FedProx	97.57%	97.57%	1.20%	97.56%	99.46%
FedAdam	69.83%	69.83%	15.98%	69.22%	77.28%
FedAdagrad	58.34%	58.34%	20.79%	54.53%	87.02%
FedYogi	94.07%	94.07%	2.95%	94.04%	98.90%

TABLE VI
EVALUATION RESULTS OF V-FIDS WITH UOWM MODBUS INTRUSION DETECTION DATASET - COMPARISON OF AGGREGATION STRATEGIES

Strategy	ACC	TPR	FPR	F1	AUC
FedAvg	0.984	0.885	0.008	0.885	0.923
FedProx ($\mu = 0.01$)	0.962	0.697	0.018	0.713	0.779
FedAdam	0.980	0.854	0.010	0.854	0.899
FedAdagrad	0.982	0.865	0.009	0.864	0.910
FedYogi	0.977	0.822	0.011	0.829	0.886

A general observation concerning the evaluation results across all datasets, is that different aggregation strategies exhibit varying levels of performance, with no single method consistently outperforming all others. This phenomenon is quite common in FL settings, as comprehensively examined in the experimental study in [25]. Similar observations were

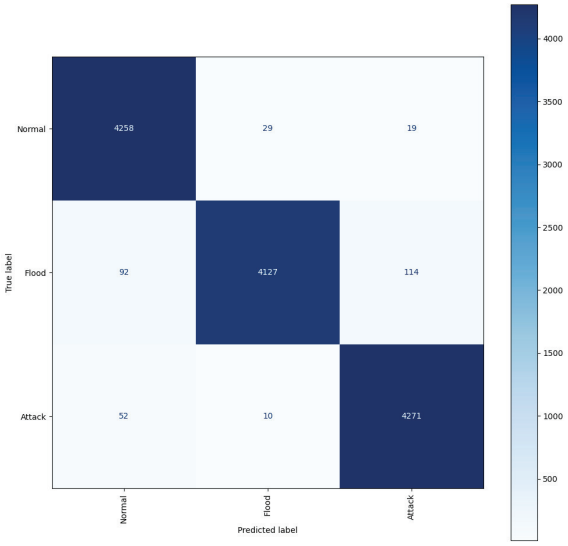

Fig. 8. Confusion matrix of the FL model (trained with CIC IoT 2022 Dataset and FedYogi) behind N-FIDS

made in [16]. For instance, vanilla FedAvg may outperform other sophisticated aggregation techniques, e.g., in Table III, since the latter often introduce instabilities during FL training. However, it is evident that FedProx demonstrates enhanced performance in most of the experiments, owing this to its design aimed at mitigating local model drift in heterogeneous and non-iid data between clients, i.e., label or feature distribution skew in the local datasets.

VI. Discussion

Based on the evaluation analysis, it is evident that AI4FIDS has the ability to recognise a wide range of cyberattacks. However, the reliability and the entire potential of AI4FIDS can be further enhanced by combining the detection outcomes of each federated detection system (i.e, N-FIDS, L-FIDS, O-FIDS and V-FIDS). On the one hand, considering that each federated detection system of AI4FIDS can detect the same attack classes, then their outcomes can be combined in order to enhance the overall reliability of AI4FIDS, utilising statistic methods such as majority voting and weighted majority voting. Therefore, the overall detection performance of AI4FIDS can be improved in terms of the aforementioned evaluation metrics. On the other hand, the detection outcomes of AI4FIDS can be associated with each other over time in order to detect multi-step attack scenarios. To this end, time window analysis techniques can be utilised. Subsequently, we further elaborate on these methods; however, due to the unavailability of the necessary datasets, detailed experimental results with respect to these methods will be presented in a future work.

A. AI4FIDS Majority Voting and Weighted Majority Voting

Let $S = \{S_1, S_2, \ldots, S_k\}$ represent k federated intrusion detection systems, such as S_1: N-FIDS, S_2: L-FIDS, S_3: V-FIDS and S_4: O-FIDS. In a simplified manner, for an instance i (based on the nature of each federated detection system), the prediction can be expressed as:

$$P_{j,i} = \begin{cases} 1 & \text{if system } S_j \text{ detects an attack on instance } i, \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

where $P_{j,i} \in \{0, 1\}$ for $j = 1, 2, \ldots, k$.
The combined prediction C_i for instance i is calculated by aggregating the individual predictions $P_{j,i}$. Therefore, the vote sum V_i, for instance i, is given by:

$$V_i = \sum_{j=1}^{k} P_{j,i}. \tag{9}$$

A threshold T is also defined to determine the final combined prediction. In particular, this threshold represents the minimum number of federated detection systems that have to agree in order to classify an instance as an attack. Thus, the final decision C_i, for instance i, is provided by:

$$C_i = \begin{cases} 1 & \text{if } V_i \geq T, \\ 0 & \text{if } V_i < T. \end{cases} \tag{10}$$

where $T = \lceil k/2 \rceil$ for simple majority voting. It is worth mentioning that the threshold T can be customised based on sensitivity requirements.
Finally, if the federated intrusion detection systems have different levels of reliability, weights w_j can be assigned to each system S_j. The weighted vote sum W_i is provided by:

$$W_i = \sum_{j=1}^{k} w_j \cdot P_{j,i}. \tag{11}$$

Then, the decision rule is given as follows.

$$\begin{cases} 1 & \text{if } W_i \geq T_w, \\ 0 & \text{if } W_i < T_w. \end{cases} \tag{12}$$

where T_w is the weighted threshold.

Following the previous analysis, the majority voting and weighted majority voting algorithms for AI4FIDS are summarized as follows.

Algorithm 1 AI4FIDS Majority Voting

Require: $\{P_{j,i}\}$: Predictions from k systems for n instances
T: Threshold for majority voting (e.g., $T = \lceil k/2 \rceil$)
Ensure: C: Combined predictions for n instances
$C \leftarrow [\,]$ {Initialize combined predictions}
for $i = 1$ to n **do**
  $V_i \leftarrow \sum_{j=1}^{k} P_{j,i}$ {Aggregate votes for instance i}
  if $V_i \geq T$ **then**
    $C_i \leftarrow 1$ {Attack detected}
  else
    $C_i \leftarrow 0$ {No attack}
  end if
  Append C_i to C
end for
return C =0

Algorithm 2 AI4FIDS Weighted Majority Voting

Require: $\{P_{j,i}\}$: Predictions from k systems for n instances
w_j: Weights assigned to each system S_j
T_w: Weighted threshold for decision
Ensure: C_i: Combined predictions for n instances
$C \leftarrow [\,]$ {Initialize combined predictions}
for $i = 1$ to n **do**
  $W_i \leftarrow \sum_{j=1}^{k} w_j \cdot P_{j,i}$ {Compute weighted vote sum for instance i}
  if $W_i \geq T_w$ **then**
    $C_i \leftarrow 1$ {Attack detected}
  else
    $C_i \leftarrow 0$ {No attack}
  end if
  Append C_i to C
end for
return C =0

The majority voting algorithm for AI4FIDS aims to aggregate the predictions from multiple federated intrusion detection systems to produce a single, combined prediction for each instance. For n instances and k federated detection systems, the algorithm iterates through all instances. For each instance i, it calculates the total votes V_i by summing the binary predictions (0 or 1) from all k systems. It is noteworthy that a predefined threshold T set to $\lceil k/2 \rceil$ for a simple majority, determines the final decision. If $V_i \geq T$, the combined prediction C_i for the instance is 1 (attack detected); otherwise, it is 0 (no attack). This approach ensures that the final prediction reflects the consensus of the federated intrusion detection systems, minimising the impact of individual errors. On the other hand, the weighted majority voting algorithm extends the majority voting approach by assigning reliability-based weights to each federated detection system. For n instances and k systems, this algorithm calculates a weighted vote sum W_i for each instance i by multiplying the binary predictions of each system by its corresponding weight w_j and summing these weighted values. A weighted threshold T_w is predefined to decide the final prediction. If $W_i \geq T_w$, the combined prediction C_i for the instance is 1 (attack detected); otherwise, it is 0 (no attack).

B. AI4FIDS Correlation Over Time Window Analysis

Let's define:

- t_i is the timestamp of a security event detected by AI4FIDS.
- $a_i \in A$ is the attack type, and A is the set of possible attack types.

Supposing that T denotes the time window for correlation (e.g., $T = 5$ minutes). The set of the security events detected by AI4FIDS is $\mathcal{E}_S = \{E_1, E_2, \ldots, E_n\}$.

$$\mathcal{C}(E_i, E_j) = \begin{cases} 1 & \text{if } |t_i - t_j| \leq T, \\ 0 & \text{otherwise.} \end{cases} \tag{13}$$

Equation (13) ensures that only events within the time window T are considered related.

Let's define a valid attack sequence $S = (a_N, a_L, a_O, a_V)$ as a tuple of attack types, where $a_N \in \mathcal{E}_N$, $a_L \in \mathcal{E}_L$, $a_O \in \mathcal{E}_O$, and $a_V \in \mathcal{E}_V$. A valid sequence must satisfy: $|S \in \mathcal{R}$ where $\mathcal{R}$ is the set of predefined rules for multi-step attacks. Given the sets of events $\mathcal{E}_N$, $\mathcal{E}_L$, $\mathcal{E}_O$, and $\mathcal{E}_V$, the detection algorithm of multi-step attacks is given as follows:

Algorithm 3 Multi-Step Attack Detection with AI4FIDS

Require: $\mathcal{E}_N$, $\mathcal{E}_L$, $\mathcal{E}_O$, $\mathcal{E}_V$: Event sets from N-FIDS, L-FIDS, O-FIDS, and V-FIDS
T: Maximum time window for correlation
$\mathcal{R}$: Set of predefined valid attack sequences
Ensure: $\mathcal{A}$: Detected multi-step attacks
$\mathcal{A} \leftarrow [\,]$ {Initialize list to store detected attacks}
for all $E_N \in \mathcal{E}_N$ **do**
  $t_N \leftarrow E_N.t$, $a_N \leftarrow E_N.a$
  for all $E_L \in \mathcal{E}_L$ **do**
    $t_L \leftarrow E_L.t$, $a_L \leftarrow E_L.a$
    if $|t_N - t_L| \leq T$ **then**
      for all $E_O \in \mathcal{E}_O$ **do**
        $t_O \leftarrow E_O.t$, $a_O \leftarrow E_O.a$
        if $|t_L - t_O| \leq T$ **then**
          for all $E_V \in \mathcal{E}_V$ **do**
            $t_V \leftarrow E_V.t$, $a_V \leftarrow E_V.a$
            if $|t_O - t_V| \leq T$ **then**
              if $(a_N, a_L, a_O, a_V) \in \mathcal{R}$ **then**
                $\mathcal{A} \leftarrow \mathcal{A} \cup \{(E_N, E_L, E_O, E_V)\}$
              end if
            end if
          end for
        end if
      end for
    end if
  end for
end for
return $\mathcal{A}$
=0

The above algorithm has the ability to detect multi-step attacks by correlating events from N-FIDS, L-FIDS, O-FIDS and V-FIDS based on temporal proximity and predefined attack rules. More specifically, it iterates through all combinations of security events from the previous federated detection systems, ensuring that their timestamps fall within a specified time window T. For each combination, the algorithm validates whether the sequence of attack types matches a predefined multi-step attack pattern in the rule set R. These rules define valid attack sequences, such as a workflow starting with a reconnaissance attack and followed by exploitation, privilege escalation and data exfiltration. If a valid sequence is available, the corresponding events are recognised as a detected multi-step attack. The algorithm ensures comprehensive detection of complex attacks while maintaining logical consistency, with the computational complexity proportional to the number of events in each federated detection system. The output of the algorithm is a list of detected attacks, providing information on how various attack stages correlate across modalities.

VII. Conclusions

Given the evolving threat landscape, in this paper, we provide AI4FIDS, a multimodal, FL-driven IDS for critical domains. AI4FIDS combines four detection systems, namely L-FIDS, O-FIDS, N-FIDS and V-FIDS, allowing federated detection through four different data types: system logs, operational data, network flow statistics and visual representations. On the other hand, T4FIDS orchestrates and automates the federated training procedure across different domains, taking into account multiple FL aggregation strategies. The evaluation results demonstrate the detection effectiveness of the proposed IDS. In our future plans, we aim to test and enhance AI4FIDS in order to improve its overall reliability and how

References

[1] P. Radoglou-Grammatikis, "Securecyber: An sdn-enabled siem for enhanced cybersecurity in the industrial internet of things," *MMTC Communications-Frontiers*, vol. 18, no. 2, pp. 16–21, 2023.

[2] M. Siganos, P. Radoglou-Grammatikis, I. Kotsiuba, E. Markakis, I. Moscholios, S. Goudos, and P. Sarigiannidis, "Explainable ai-based intrusion detection in the internet of things," in *Proceedings of the 18th International Conference on Availability, Reliability and Security*. Benevento, Italy: Association for Computing Machinery, 2023.

[3] P. Radoglou-Grammatikis, P. Sarigiannidis, P. Diamantoulakis, T. Lagkas, T. Saoulidis, E. Fountoukidis, and G. Karagiannidis, "Strategic honeypot deployment in ultra-dense beyond 5g networks: A reinforcement learning approach," *IEEE Transactions on Emerging Topics in Computing*, pp. 1–12, 2022.

[4] D. C. Asimopoulos, P. Radoglou-Grammatikis, I. Makris, V. Mladenov, K. E. Psannis, S. Goudos, and P. Sarigiannidis, "Breaching the defense: Investigating fgsm and ctgan adversarial attacks on iec 60870-5-104 ai-enabled intrusion detection systems," in *Proceedings of the 18th International Conference on Availability, Reliability and Security*. Benevento, Italy: Association for Computing Machinery, 2023, pp. 1–8.

[5] P. Radoglou-Grammatikis, P. Sarigiannidis, G. Efstathopoulos, T. Lagkas, G. Fragulis, and A. Sarigiannidis, "A self-learning approach for detecting intrusions in healthcare systems," in *ICC 2021-IEEE International Conference on Communications*. Montreal, QC, Canada: IEEE, 2021, pp. 1–6.

[6] A. Vázquez-Ingelmo, A. García-Holgado, and F. J. García-Peñalvo, "C4 model in a software engineering subject to ease the comprehension of uml and the software," in *2020 IEEE Global Engineering Education Conference (EDUCON)*. IEEE, 2020, pp. 919–924.

[7] M. Alazab, S. P. RM, M. Parimala, P. K. R. Maddikunta, T. R. Gadekallu, and Q.-V. Pham, "Federated learning for cybersecurity: Concepts, challenges, and future directions," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 5, pp. 3501–3509, 2021.

[8] B. Ghimire and D. B. Rawat, "Recent advances on federated learning for cybersecurity and cybersecurity for federated learning for internet of things," *IEEE Internet of Things Journal*, vol. 9, no. 11, pp. 8229–8249, 2022.

[9] E. M. Campos, P. F. Saura, A. González-Vidal, J. L. Hernández-Ramos, J. B. Bernabe, G. Baldini, and A. Skarmeta, "Evaluating federated learning for intrusion detection in internet of things: Review and challenges," *Computer Networks*, vol. 203, p. 108661, 2022.

[10] L. Lavaur, M.-O. Pahl, Y. Busnel, and F. Autrel, "The evolution of federated learning-based intrusion detection and mitigation: a survey," *IEEE Transactions on Network and Service Management*, vol. 19, no. 3, pp. 2309–2332, 2022.

[11] S. Arisdakessian, O. A. Wahab, A. Mourad, H. Otrok, and M. Guizani, "A survey on iot intrusion detection: Federated learning, game theory, social psychology, and explainable ai as future directions," *IEEE Internet of Things Journal*, vol. 10, no. 5, pp. 4059–4092, 2022.

[12] S. I. Popoola, G. Gui, B. Adebisi, M. Hammoudeh, and H. Gacanin, "Federated deep learning for collaborative intrusion detection in heterogeneous networks," in *2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall)*. Norman, OK, USA: IEEE, 2021, pp. 1–6.

[13] O. Friha, M. A. Ferrag, L. Shu, L. Maglaras, K.-K. R. Choo, and M. Nafaa, "Felids: Federated learning-based intrusion detection system for agricultural internet of things," *Journal of Parallel and Distributed Computing*, vol. 165, pp. 17–31, 2022.

[14] R. Zhao, Y. Wang, Z. Xue, T. Ohtsuki, B. Adebisi, and G. Gui, "Semi-supervised federated learning based intrusion detection method for internet of things," *IEEE Internet of Things Journal*, 2022.

[15] M. J. Idrissi, H. Alami, A. El Mahdaouy, A. El Mekki, S. Oualil, Z. Yartaoui, and I. Berrada, "Fed-anids: Federated learning for anomaly-based network intrusion detection systems," *Expert Systems with Applications*, vol. 234, p. 121000, 2023.

[16] R. Lazzarini, H. Tianfield, and V. Charissis, "Federated learning for iot intrusion detection," *Ai*, vol. 4, no. 3, pp. 509–530, 2023.

[17] O. Belarbi, T. Spyridopoulos, E. Anthi, I. Mavromatis, P. Carnelli, and A. Khan, "Federated deep learning for intrusion detection in iot networks," in *GLOBECOM 2023-2023 IEEE Global Communications Conference*. IEEE, 2023, pp. 237–242.

[18] G. Shingi, H. Saglani, and P. Jain, "Segmented federated learning for adaptive intrusion detection system," *arXiv preprint arXiv:2107.00881*, 2021.

[19] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and A. Anwar, "Ton_iot telemetry dataset: A new generation dataset of iot and iiot for data-driven intrusion detection systems," *Ieee Access*, vol. 8, pp. 165 130–165 150, 2020.

[20] I. Sharafaldin, A. H. Lashkari, A. A. Ghorbani *et al.*, "Toward generating a new intrusion detection dataset and intrusion traffic characterization." *ICISSp*, vol. 1, pp. 108–116, 2018.

[21] S. Dadkhah, H. Mahdikhani, P. K. Danso, A. Zohourian, K. A. Truong, and A. A. Ghorbani, "Towards the development of a realistic multidimensional iot profiling dataset," in *2022 19th Annual International Conference on Privacy, Security & Trust (PST)*. Fredericton, NB, Canada: IEEE, 2022, pp. 1–11.

[22] L. Collins, H. Hassani, A. Mokhtari, and S. Shakkottai, "Fedavg with fine tuning: Local updates lead to representation learning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 10 572–10 586, 2022.

[23] X. Yuan and P. Li, "On convergence of fedprox: Local dissimilarity invariant bounds, non-smoothness and beyond," *Advances in Neural Information Processing Systems*, vol. 35, pp. 10 752–10 765, 2022.

[24] S. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečnỳ, S. Kumar, and H. B. McMahan, "Adaptive federated optimization," *arXiv preprint arXiv:2003.00295*, 2020.

[25] Q. Li, Y. Diao, Q. Chen, and B. He, "Federated learning on non-iid data silos: An experimental study," in *2022 IEEE 38th international conference on data engineering (ICDE)*. IEEE, 2022, pp. 965–978.

Dr. Panagiotis Radoglou-Grammatikis received Diploma (five years) and PhD from the Dept. of Electrical and Computer Engineering, University of Western Macedonia, Greece, in 2016 and 2023, respectively. His main research interests focus on AI-driven cybersecurity, intrusion detection and security games. He has published more than 50 research papers in international scientific journals, conferences and book chapters, while he has received five best paper awards. He was included in Stanford University's list (shared by Elsevier) of the Top 2% of Scientists in the World for 2021 and 2022. Currently, he is working as a research director at K3Y Ltd, while he is also a postdoc researcher at the ITHACA Lab of the University of Western Macedonia and co-founder of MetaMind Innovations P.C. He is involved in several national and international projects. Finally, he is a member of IEEE, ACM and the Technical Chamber of Greece.

Dr. Pavlos Bouzinis received the Diploma (five years) and Ph.D. degrees in electrical and computer engineering from the Aristotle University of Thessaloniki, Greece, in 2019 and 2023, respectively, where he was a member of the Wireless Communications and Information Processing Group. Currently, he works as a researcher at MetaMind Innovations P.C. His main research interests include machine learning, optimization, and intrusion detection systems. He has served as a reviewer for several scientific journals and was an exemplary reviewer of IEEE WIRELESS COMMUNICATIONS LETTERS, in 2021 (top 3% of reviewers).

Ioannis Makris received his BSc in Computer Science with specialization in Artificial Intelligence and Software Engineering from the Aristotle University of Thessaloniki (AUTh) and his MSc in Business Analytics from the University of Edinburgh. Furthermore, he is a Certified Associate in Project Management (CAPM) by the Project Management Institute (PMI). His interests include privacy-preserving AI techniques, interpretable machine learning, and security. He is currently employed by MetaMind Innovations, working as an AI Engineer/Researcher in several European-funded projects on cybersecurity, telecommunications, and energy efficiency.

Dr. Thomas Lagkas is Assistant Professor at the Department of Computer Science of the Democritus University of Thrace and Director of the Laboratory of Industrial and Educational Embedded Systems. He graduated with honours from the Department of Informatics, Aristotle University of Thessaloniki and awarded PhD on Wireless Networks. He also completed MBA studies at the Hellenic Open University and received a postgraduate certificate on Teaching and Learning from The University of Sheffield. He has been scholar of the Aristotle University Research Committee and postdoctoral scholar of the National Scholarships Institute of Greece. His research interests are in the areas of IoT communications with numerous highly cited publications. Dr. Lagkas is an IEEE Senior Member, Fellow of the Higher Education Academy in the UK, and member of the Editorial Board of reputable scientific journals. Moreover, he actively participates in several EU-funded research projects.

Prof. Vasileios Argyriou received the B.Sc. degree in computer science from the Aristotle University of Thessaloniki, Greece, in 2001, and the M.Sc. and Ph.D. degrees in electrical engineering working on registration from the University of Surrey, in 2003 and 2006, respectively. From 2001 to 2002, he held a research position with Aristotle University, with a focus on image and video watermarking. He joined the Communications and Signal Processing Department, Imperial College London, London, in 2007, where he was a Research Fellow working on 3D object reconstruction. He is currently a Professor with Kingston University, London, working on computer vision and AI for crowd and human behavior analysis, computer games, entertainment, and medical applications. Also, research is conducted on educational games and on HCI for augmented and virtual reality (AR/VR) systems.

Dr. Georgios Th. Papadopoulos is an Assistant Professor in the area of Computer Graphics and Computational Vision at the Department of Informatics and Telematics of the Harokopio University of Athens in Greece. He received the Diploma and Ph.D. degrees in electrical and computer engineering from the Aristotle University of Thessaloniki (AUTH), Thessaloniki, Greece. He has worked as a Post-doctoral Researcher at the Foundation For Research And Technology Hellas / Institute of Computer Science (FORTH/ICS) and the Centre for Research and Technology Hellas / Information Technologies Institute (CERTH/ITI). He has published over 70 peer-reviewed research articles in international journals and conference proceedings. His research interests include computer vision, artificial intelligence, machine/deep learning, human action recognition, human-computer interaction and explainable artificial intelligence. Dr. Papadopoulos is a member of the IEEE and the Technical Chamber of Greece.

Dr. Panagiotis Fouliras received the B.Sc. degree in physics from the Aristotle University of Thessaloniki, Greece, and the M.Sc. and Ph.D. degrees in computer science from the University of London, U.K. (QMW). He is currently a permanent Assistant Professor with the University of Macedonia, Thessaloniki, Greece. He has participated in several national and European-funded (H2020) research projects and published articles in many international journals. His research interests include computer networks and network security, blockchain, and system evaluation methods.

Prof. George Seritan has graduated from the "Politehnica" Institute Bucharest in 1997. He is Director of Laboratory Electrical Energy Quality, His research experience is in the fields: digital signal processing, power quality, power systems, automatic measurements systems and metrology.

Prof. Panagiotis Sarigiannidis is the Director of ITHACA Lab, Co-Founder of MetaMind Innovations P.C. and Full Professor at the Department of Electrical and Computer Engineering, University of Western Macedonia, Kozani, Greece. He received his B.Sc. and Ph.D. in computer science from the Aristotle University of Thessaloniki, Thessaloniki, Greece, in 2001 and 2007, respectively. His research interests include telecommunication networks, Internet of Things and cybersecurity. He has published over 270 papers in international journals, conferences and book chapters, while he has also received five best paper awards. He is involved in several national and international projects. He served as the project coordinator of three H2020 projects, namely SPEAR, EVIDENT and TERMINET. Moreover, he has coordinated national and Erasmus+ KA2 projects, while he served as a principal investigator in SDN-microSENSE and three Erasmus+ KA2: ARRANGE-ICT, JAUNTY and STRONG. Finally, he participates in several editorial boards of various journals.