*electronics*

MDPI

*Article*

# Collaborative Federated Learning-Based Model for Alert Correlation and Attack Scenario Recognition

**Hadeel K. Alkhpor [1] and Faeiz M. Alserhani [2,*]**

[1] Department of Computer Science and Technology, Al-Jouf University, Al-Jouf 72311, Saudi Arabia; 431205901@students.ju.edu.sa
[2] Department of Computer Engineering and Networks, Al-Jouf University, Al-Jouf 72311, Saudi Arabia
* Correspondence: fmserhani@ju.edu.sa

**Abstract:** Planned and targeted attacks, such as the advanced persistent threat (APT), are highly sophisticated forms of attack. They involve numerous steps and are intended to remain within a system for an extended length of period before progressing to the next stage of action. Anticipating the next behaviors of attackers is a challenging and crucial task due to the stealthy nature of advanced attack scenarios, in addition to the possible high volumes of false positive alerts generated by different security tools such as intrusion detection systems (IDSs). Intelligent models that are capable of establishing a correlation individual between individual security alerts in order to reconstruct attack scenarios and to extract a holistic view of intrusion activities are required to exploit hidden links between different attack stages. Federated learning models performed in distributed settings have achieved successful and reliable implementations. Alerts from distributed security devices can be utilized in a collaborative manner based on several learning models to construct a federated model. Therefore, we propose an intelligent detection system that employs federated learning models to identify advanced attack scenarios such as APT. Features extracted from alerts are preprocessed and engineered to produce a model with high accuracy and fewer false positives. We conducted training on four machine learning models in a centralized learning; these models are XGBoost, Random Forest, CatBoost, and an ensemble learning model. To maintain privacy and ensure the integrity of the global model, the proposed model has been implemented using conventional neural network federated learning (CNN_FL) across several clients during the process of updating weights. The experimental findings indicate that ensemble learning achieved the highest accuracy of 88.15% in the context of centralized learning. CNN_FL has demonstrated an accuracy of 90.18% in detecting various attacks of APTs while maintaining a low false alarm rate.

**Keywords:** IDS; APT; alert correlation; centralized learning; federated learning



**Citation:** Alkhpor, H.K.; Alserhani, F.M. Collaborative Federated Learning-Based Model for Alert Correlation and Attack Scenario Recognition. *Electronics* **2023**, *12*, 4509. https://doi.org/10.3390/electronics12214509

Academic Editors: Aryya Gangopadhyay and Andrei Kelarev

Received: 25 September 2023
Revised: 23 October 2023
Accepted: 31 October 2023
Published: 2 November 2023



## 1. Introduction

An intrusion detection system (IDS) refers to a hardware or software entity that monitors a network to detect and identify unauthorized activities or violations of established policies. In the event that an intrusion is detected, IDS captures pertinent data regarding the event, disseminates alerts, and executes appropriate remedial or preventive measures as deemed required [1]. The pervasive utilization of the internet by individuals has led to a significant escalation in network attacks, resulting in substantial detrimental consequences for both institutions and individuals alike. As a consequence of the heightened occurrence and enhanced complexity of cyberattacks, contemporary networked enterprises are compelled to adopt rigorous security protocols in order to ensure the integrity of their data transmissions. Consequently, IDS has become an indispensable component of any security framework [2,3].

IDS is a form of passive monitoring system utilized to detect potential threats, supply a comprehensive account of an ongoing or attempted theft, and initiate alerts that

may be reviewed by analysts stationed in a security operations center (SOC) or incident responders. The majority of attacks are executed through the examination of network communications, wherein packets traversing the network are intercepted and subjected to analysis. This process involves the identification of heuristics and patterns, commonly referred to as signatures, which serve as means to detect and classify common attacks. Upon detection, operators are promptly alerted, thereby enabling appropriate action to be taken [4]. Network attacks often involve a series of concerted attempts to attack various parts of the system. Defending against these multi-stage attacks requires knowledge of the current attack state and the expected future attack phase. The sub-components in a web network are interlinked, and analyzing the logs of one device may reveal events triggered in another [5]. Advanced attack scenarios are a contemporary form of attack employed by adversaries, characterized by the utilization of sophisticated and elusive exploits and payloads. The primary objective of APTs is to establish prolonged persistence within a targeted system and subsequently move laterally to accomplish various goals, including but not limited to extensive data collection, disruption of operations, or denial of services [6]. In addition, it should be noted that anomaly-based detection methods have been found to produce an extensive amount of false positive results [7]. The concept of alert correlation (AC), also referred to as IDS post-processing, has been suggested as a means to address these constraints. The system that is able to predict full or part of attack scenarios can provide early and preventive measures to reduce the severity and threat caused by network attacks, hence adopting a proactive approach [8]. In the context of IDS models based on learning, one additional challenge is the insufficiency of datasets that adequately describe various advanced attacks and APT patterns, which is crucial for effectively training a robust detection model [9]. So, to protect enterprise networks from cyberattacks, IDS is continuously developed to solve various limitations, one of which is the production of a substantial quantity of alerts that are of low quality. Furthermore, a significant proportion of the alerts generated by IDSs are classified as false positives. It is essential to have a robust level of security to ensure transparent and reliable communication between parties to find malicious trends and help administrators fine-tune, organize, and deploy efficient controls.

Machine learning (ML) is a form of artificial intelligence methodology that has the capability to autonomously extract valuable insights from extensive datasets [10]. ML-based IDS can attain reasonable levels of detection performance given the availability of ample training data. Additionally, these IDS employ machine learning models that possess enough generalization capabilities to identify attack variations and novel attacks [11]. Deep learning (DL) is a subfield within the broader domain of machine learning that has demonstrated remarkable capabilities in achieving exceptional levels of performance. DL approaches have demonstrated superior performance in handling large datasets as compared to traditional machine learning techniques. Furthermore, DL techniques possess the capability to autonomously acquire feature representations from unprocessed data, afterwards generating outcomes encompassing multiple hidden learning layers. DL trains several neural nodes instead of one like linear regression in statistical learning [12].

In addition, our study examined the concepts of centralized and federated learning in order to assess the accuracy of detection while taking into account concerns regarding privacy, heterogeneity, and data availability. Federated learning (FL) is a collaborative ML learning approach that is implemented across various clients, ensuring that clients' personal data are not transferred to a central server provider and instead remain stored on the local client device. The conventional approach to neural network (NN) training involves the utilization of local datasets by all clients, which are then shared with a central server. On the other hand, distributed convolutional neural network training employs parallel training methods. The training of federated learning is operated under the assumption that local client datasets are separate and cannot be accessed or shared by others. The study is to provide a comparative analysis between an ensemble centralized model that combines three machine learning algorithms and a CNN_FL federated learning model. The

evaluation process is to examine the detection rate in addition to the metrics typically used in such a design. In the FL approach, several local models using heterogeneous algorithms are trained in each device (clients), and the outcomes of these models are used to build and update a global model (server). The inputs of the global model are the results of the effective collaboration of data across multiple clients. This collection of various trained models contributes to the building of an intelligent IDS able to perform detection and prevention facilities with higher performance. Multistage attacks are modeled based on typical attack frameworks such as cyber kill chains [13]. The primary contributions of this study are as follows:

- Perform an extensive analysis of alert data from various datasets to model multi-stage and sophisticated attacks in order to reconstruct intrusion scenarios.
- Design a security system utilizing collaborative federated learning models to detect the cross-correlation between alerts generated from different sources and with various formats. Zero-day and novel attacks can be predicated by building high-level abstraction of attacker actions against protected systems.
- Evaluate the proposed system using a benchmark dataset based on different metrics of performance and accuracy.

The rest of the paper is organized as follows: Related work is described in Section 2. In Section 3, the suggested method is described, and in Section 4, the results are discussed. Conclusion and future work are described in Section 5.

## 2. Related Work

The potential damage of multi-stage and advanced attacks necessitates the establishment of a comprehensive analysis to identify the steps conducted by intruders to perform such attacks. Detection of the links between different activities based on triggered alerts is crucial to create a global view of the attack in progress. This knowledge, whether it is complete or partial, can provide network administrators with valuable information in order to counter these attacks and to apply a mitigation procedure.

In their study, Rahman et al. [14] devised models for detecting advanced attack scenarios by employing a centralized approach. The NSL-KDD dataset is employed for evaluating the efficacy of federated architecture in IDS [7]. The study considers a range of practical scenarios and instances of intrusion attacks. Based on the empirical findings, a comprehensive evaluation is conducted to compare the FL, centralized, and self-learning methodologies. FL consistently demonstrated superior performance compared to the alternative methodologies in nearly all training iterations. The concept of "virtual reality" pertains to the procedure of developing programs that simulate a virtual reality experience. The intrusion detection capabilities of a federated network increase proportionally with its scale. In [15], an IDS in Wireless Edge Networks (WENs), combining GRU and SVM models under a custom FL algorithm, was proposed. They used Attention Mechanism to determine the significance of the uploaded model parameters. This is conducted with the goal of both measuring the global model's performance improvement and sorting the clients according to their importance. The authors [16] provided a strategy to improve the training effect by sharing a limited sample of data globally. This offers a scheme for the case of non-IID data in federated learning, which is an important consideration.

To detect distributed anomaly intrusion on an IoT-based industrial control system, a hybrid model that includes federated learning, autoencoder, transformer, and Fourier mixing sublayer was developed [17]. It delivered a high detection performance for time-series data while simultaneously solving the problem of anomaly detection on a minute-by-minute time scale with rapid learning. F. Wilkens et al. [18] used a kill chain state machine (KCSM) to detect complex attacks like advanced persistent threats (APTs) without having to spend more time analyzing large volumes of alerts. Their method generated scenario graphs from state machines by deriving potential attack stages from single and meta-alerts and modeling the resulting attack scenarios. The algorithm generates APT scenario graphs,

which are graphical representations of the attack, with nodes representing involved hosts and edges representing infection activity.

The researchers in [17] proposed a model called MLAPT [19], which generates a correlation between alerts and the corresponding APT scenarios. Subsequently, ML models were employed to forecast APT events at their first stages, achieving a prediction accuracy of 84.8%. The authors [20] suggested a machine learning-based intrusion detection system that uses DT, RF, SVM, KNN, and DNN in both models of centralized and federated learning. A used Edge-IIoT set that had more than 10 different kinds of IoT devices was used to collect a dataset that categorized 15 attacks. These attacks were grouped into five threats, and some of their features were identified with high correlations. In centralized learning, the best DDN accuracy was found to be 94.67% for 15-class and 96.01% for 6-class. On the other hand, RF obtained 99.9% in binary classification. However, the federated learning results achieved were better; the accuracy results of the global model training with 10 clients and 10 iterations were 93.37%, 95.99%, and 100%, respectively, for 15-class, 6-class, and binary classification. M. Khosravi et al. [21] proposed a method for detecting (APTs) that relies on causal analysis and correlation between alerts. Multiple sensors' alarms are monitored over a lengthy period to determine which ones are most likely to be part of the APT attack's well-known IKC. Finally, it acceptably calculated the host infection score over all APT phases using a semi-real-world dataset and simulation.

A system for detecting APT attacks based on federated learning was proposed in [22] to differentiate various APT attack patterns. The global model is updated across multiple clients with various iterations. The malicious events collected as alerts are then fed to the correlation module to determine which alerts are most relevant to APT attack steps. Based on alert type-determined APT stages, an APT scenario indicated the probability of the change of the attack's step. With 400 iterations, their model applied to UNSW-NB15 datasets and synthetic datasets from five clients obtained 96.7% accuracy, which is higher than local models. The authors [23] proposed models for anomaly detection on two datasets, namely Contagio and CICIDS2017, using an unsupervised learning approach. Subsequently, the study will explore various known malware attacks targeting networks.

The authors [24] presented federated learning and CNN to detect abnormal IoT traffic without alert correlation. Mayfly optimization was used to minimize feature dimension, and the FL framework was then trained for each CNN local model for collaborative training without sharing private data. The Aposemat IoT-23 dataset detected anomalous IoT traffic with 97.73% accuracy using multi-class detection. The researchers [25] presented a federated learning framework to identify APT attacks in an SDN environment. They employed ML and DL techniques to categorize harmful indications. The researchers ran an experiment using the NF-UQ-NIDS dataset and models to showcase the viability of FL in addressing cyber threats while preserving privacy for data holders within the SDN environment. W. Giura et al. [26] proposed a model for APT detection that can be applied to general occurrences, expanding beyond the scope of IDS alerts. The attack stages are structured in a hierarchical pyramid, wherein the ultimate objective occupies the apex, while the preceding steps are organized into several strata. HTTP-based connections are considered more advantageous compared to alternative options for several reasons. Firstly, HTTP-based command and control (C&C) traffic is generally recognized as permissible within the majority of enterprise environments. Secondly, alternative C&C protocols like peer-to-peer (P2P) and Internet Relay Chat (IRC) exhibit distinctive network characteristics, such as specific ports and package content, which can be readily detected and obstructed [27]. The propagation of malware occurs through the utilization of custom encrypted partitions on removable media, as well as the exploitation of vulnerabilities within authentication protocols [28,29]. Kasongo et al. [30] suggested an ensemble model incorporating feature selection, specifically targeting the 19 most significant features out of the total 42 features available in the UNSW-NB15 dataset. The performance accuracy yielded a result of 75%. The study [31] presented an improved CNN architecture for the purpose of identifying malicious attack traffic, with a particular focus on zero-day attacks that have not been

previously reported within the network. The binary classification findings of the study indicate that the model exhibited superior performance in detecting previously undetected instances of intrusion, as compared to the standard CNN model. The authors [32] conducted a study on intrusion anomaly detection, specifically examining different kernel functions within Support Vector Machines. They also utilized the Principal Component Analysis feature selection technique in their investigation. The datasets provided are the UNSW-NB15 datasets. The Gaussian kernel achieved the highest level of accuracy, measuring at 93.94% when applied to the UNSW-NB15 datasets.

Multi-step attack detection in IDSs using analysis of correlated logs or alerts is still not sufficiently studied from all angles. The majority of research efforts focus on analysis of network traffic features as shown in Table 1. This is the research gap addressed in our proposal to extract causal relationships between events in device logs and IDS alerts by training distributed models. This can provide insight into the operations of attack behaviors. FL is a technique for decentralized learning that protects users' privacy by not transferring data but instead training models locally and sending the parameters to a centralized server, making FL an appropriate choice for improving the results of IDS applications. In this work, attack phases are characterized by alerts triggered from various IDSs and possibly server logs. This methodology is based on the assumption that alerts and logs have valuable information if we extract the logical links between these alerts. We propose a security analysis system consisting of a centralized global model in addition to a number of federated models to effectively identify and classify various attack scenarios.

**Table 1.** Related works and the proposed models for the detection of multi-stages attacks.

| Work | Year | Dataset | Approaches | Moel | Weakness |
|---|---|---|---|---|---|
| M. A. Ferrag [20] | 2022 | Edge-IIoTset | Modeling attacks traffic and process used DT, RF, SVM, KNN, and DNN. | Centralized and federated learning | Identified 61 features with high correlations for traffic, did not include scenario for alerts, and generated meta-alerts. |
| M. Khosravi et al. [21] | 2020 | Semi real-world dataset | Modeling attacks process, generating meta-alerts with APT steps and host score for all risk levels. | Finding IKCs using Causal Relation Analysis | No centralized and federated learning to classify APT attacks. |
| Z. Li et al. [22] | 2020 | UNSW-NB15 and synthetic datasets | Modeling attacks process, correlating alerts to APT stages and identifying the probability of APT stage change. | Federated | Gain correlation method determines association but cannot predict causation. |
| I. Ghafir et al. [19] | 2018 | Simulation dataset | Created a correlation framework to link the alerts to the APT attacks and use ML models to predict network events. | ML | Only considered network events. |
| H. Neuschmied et al. [23] | 2022 | Contagio and CICIDS2017 | Detection of abnormal behavior based on network traffic analysis. | Several autoencoders | Lack of generality and only identified network events. |
| Q. Xia et al. [24] | 2022 | Aposemat IoT-23 | Detection of abnormal behavior based on network traffic. | FL CNN | Did not study the causal relation of alerts. |
| H. T. Thi et al. [25] | 2022 | UNSW-NB15 | Detection of APT attacks based on network traffic in SDN. | FL | Considered network events. |
| Yin, Y., Jang-Jaccard [29] | 2023 | UNSW-NB15 | Filter methods were employed to assess the impact of less significant features in relation to high-frequency values. | MLP | Only classification network attacks based on features filtering without studying the causal relation of alerts. |
| Kasongo et al. [30] | 2021 | UNSW-NB15 | Detection of abnormal behavior based on network traffic analysis. | Ensemble models | Did not conduct the causal relation of alerts or stages. |

**Table 1.** *Cont.*

| Work | Year | Dataset | Approaches | Moel | Weakness |
|---|---|---|---|---|---|
| B. I. Hairab [31] | 2022 | Bot-IoT dataset | Focus on zero-day attacks that have not been previously reported within the network. | ML and DL methods | Only considered DoS and DDoS scenarios for traffic attacks. |
| M. A. Almaiah et al. [32] | 2022 | UNSW-NB15 | Detection of abnormal behavior based on network traffic. | PCA and kernals of SVM | Only classification network attacks based on features filtering without studying the causal relation of alerts. |

## 3. Materials and Methods

The proposed system comprises several components, some components as shown in Figure 1. Security alerts are collected from different security devices configured in the network or hosts. In this research, alerts are generated from the UNSW-NB15 dataset [33] using Snort [34,35] and Zeek [36]. First, the resulting alerts are normalized and aggregated, encompassing hyper-alerts to remove redundancy. Then, a preprocessing stage is applied to the collected alerts to create a reliable and accurate training dataset. Null and extreme values are removed, and some values are normalized to make the training process faster and more stable. The rows of the dataset are assigned a label representing the attack stage. Dataset features are extracted and engineered to handle imbalanced data. Synthetic minority over-sampling technique (SMOTE) [37] is used to balance the label distribution. The next stage is to divide the whole dataset randomly into groups that are used to train models configured in each client. Local models configured in clients' machines are trained using the assigned sub-datasets in order to build the detection model. The global model is built and optimized using the resulting trained models from the distributed client models. In other words, the global model is designed based on cooperation between the detection of attack data supplied by individual client models. This federated learning paradigm maintains the data privacy of involved clients, more accurate detection and prediction functions, and higher performance of elaborate systems.

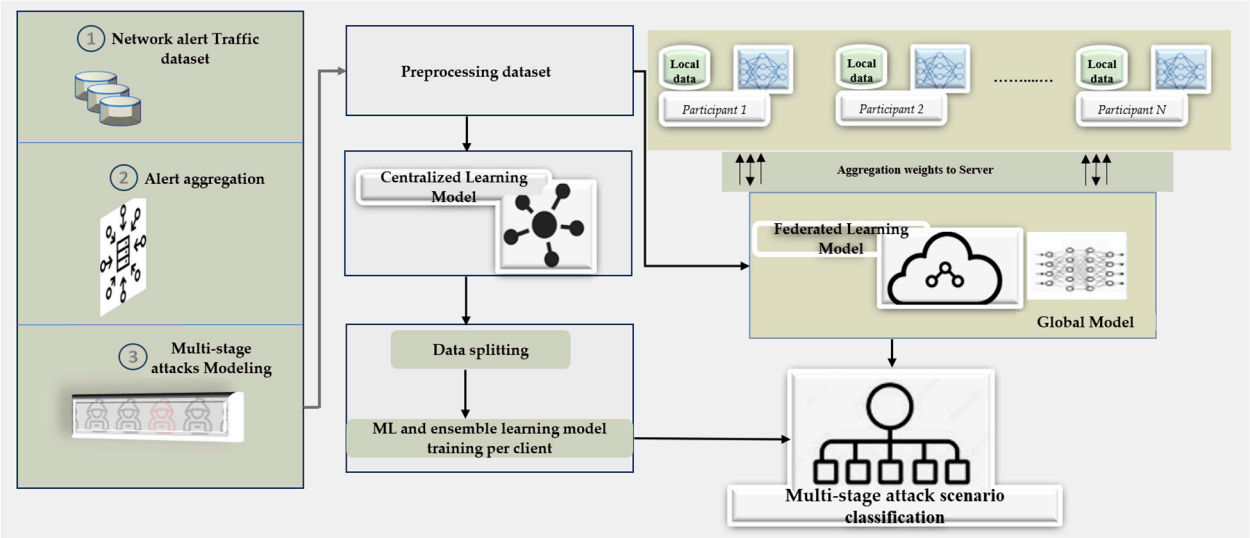



**Figure 1.** The proposed federated learning-based model for alert detection of multi-stage attack scenarios.

### *3.1. Modeling Multi-Stage Attack Scenario*

The cyber kill chain [13] models security attacks as a sequence of seven stages, including: (1) Reconnaissance, (2) Weaponization, (3) Delivery, (4) Exploit, (5) Installation, (6) Command and Control, and (7) Actions on Objectives. Moreover, the MITRE ATT&CK

framework [38] is to map alerts to various attack stages. This framework is an integration of tactics and approaches based on summaries of actual attacks [22]. It is possible to summarize the attack stage by looking at the techniques that can be employed for specific tactical reasons, which are listed in a matrix format in ATT&CK. Each alert's advanced attack scenario stage can be identified using the mapping.

We assumed a similar modelling paradigm, but the stages are not hardcoded; instead, we anticipate a flexible model to consider full or partial sequential stages and satisfy the dynamic nature of the attack security. Initially, the attacker passively fingerprints the target system to collect relevant information; this stage is known as passive reconnaissance. This stage can be skipped because it does not involve any contact with the target system and does not cause any alerts to be produced. Subsequently, the target system is scanned to determine the status of the connection and identify running services and protocols. Then, vulnerability scanning is performed to discover any weaknesses or a point to obtain access. The actual attack is executed during the exploit stage, and that is based on the scanning obtained from previous stages. Malicious code is delivered and installed, and at this point, the target machine becomes compromised. After that, latent activities are performed during the latent movement stage, such as attacking other machines and being a part of denial access attacks. Thus, in this research, we consider five stages as they are a part of the UNSW-NB15 dataset [22,33]. It is worth noting that missing one or more stages from the sequential stages will not affect the construction of the scenario; for instance, the attack path can jump directly to any later stages. This is considered partial knowledge of the attack in progress. In Table 2, we outline the five phases of an attack and assign each alert type to the appropriate phase.

**Table 2.** Advanced attack scenario stages and alerts.

| Stages of APT | Type of Alert | No of Records | Encoding Label |
|---|---|---|---|
| 1st: Reconnaissance | Gathering information | 13,987 | 0 |
| 2nd: Initial Access | Fuzzer, Analysis | 26,923 | 1 |
| 3rd: Exploitation | Exploits | 44,525 | 2 |
| 4th: Persistent | Backdoor, Shellcode | 3840 | 3 |
| 5th: Lateral Movement | Worms | 174 | 4 |

### *3.2. Alert Correlation Model*

The main objective of the design is to reconstruct the attack scenario inferred from alert information from multiple sources. Once the alert set is acquired, similar alerts generated by the same event are aggregated to provide hyper-alerts and avoid duplicated data. Then, these hyper-alerts are mapped to a specific attack phase. Mapping functionality is determined based on model learning, which reflects the causality relationship between any two or more alerts. Similarity score methods have been widely used to perform alert correlation [22,39]. However, these methods require several probabilities in advance collected from comprehensive experimental efforts. The outcome of the correlation component yields multiple alert groupings vertically and alert correlation horizontally. The concept of alert correlation relies on the utilization of a similarity score, which serves as a metric for assessing the causal relationship between two alerts, as described in [22]. However, the model parameters to achieve a reliable correlation task are generated by the learning stage. Alerts that are both relevant and part of multi-stage attack activities are inferred from the correlation model. Hence, the correlated alerts are used to reconstruct attack phases based on the strategies and tactics employed by the adversary at that phase. Accordingly, hyper-alerts are mapped to one of the attack stages, considering temporal and spatial attributes of each alert. The attack scenario is constructed based on the attack scenario model described in Section 3.1. Figure 2 illustrates the attack stages ordered from the initial phase to represent the attack sequential stages.

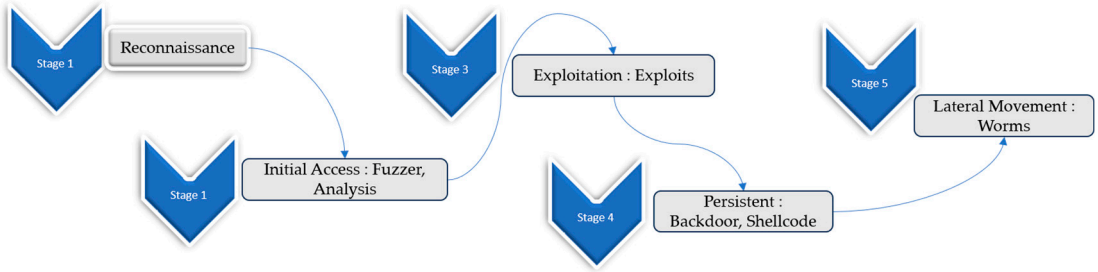



**Figure 2.** The Sequence Graph of Attack.

### *3.3. Dataset Description*

In this research, we have analyzed the UNSW-NB15 intrusion detection dataset, which contains attacks classified as Fuzzers, Analyzers, Backdoors, Denial-of-Service Attacks, Exploits, Generic Attacks, Reconnaissance Attacks, Shellcode, and Worms. It has 49 features, as well as the class label and 2,540,044 records. The majority of the data are regular traffic; however, we have analyzed the records of attacks that belong to the advanced attack scenario, as indicated in Table 3.

**Table 3.** The types of attacks utilized in this study.

| Alert Type | Records |
|---|---|
| Exploits | 44,525 |
| Fuzzers | 24,246 |
| Reconnaissance | 13,987 |
| Analysis | 2677 |
| Backdoor | 2329 |
| Shellcode | 1511 |
| Worms | 174 |

### *3.4. Implementation*

Federated learning can be implemented using two settings: centralized FL, which requires a central server to coordinate multiple clients and contains the global dataset, and decentralized FL, which allows client devices to train on their own data and share the model. We have implemented the proposed federated system in the centralized architecture for comparison of its results of individual ML and ensemble algorithms with decentralized CNN_FL. The implementation stages are explained in the following sections.

#### 3.4.1. Preprocessing

During the preprocessing stage, we first determined all features for null label values to assign each record by the 'Normal' label. Some features have extreme values, such as ct_flw_http_mthd and is_ftp_login; thus, we replaced those values with the median value. In addition, using the OrdinalEncoder library encodes the categorical features ('proto', 'state', 'service', 'ct_ftp_cmd', and 'attack_cat') to generate numeric values. In the final step, we normalized and scaled the data by employing the MinMaxScaler.

#### 3.4.2. Feature Selection

Analyzing the dataset's features to ensure that they are normalized and in a stable state is crucial for obtaining a more accurate model. Feature importance is calculated using the Forest algorithm, as shown in Table 4 and Figure 3. We have excluded any feature less than the threshold value (0.012). Features with high correlation strength are excluded using a threshold of 95%, and that is due to their lack of involvement in the generalization process. We added two features to represent the total number of bytes directed from source to destination and vice versa.

**Table 4.** The order of features according to values of importance.

| Feature Name | Importance Value | Feature Name | Importance Value |
|---|---|---|---|
| bytes | 0.129649 | loss | 0.023135 |
| means | 0.106072 | Sujit | 0.023083 |
| ct_srv_dst | 0.072535 | dur | 0.022949 |
| state | 0.052549 | Spkts | 0.022914 |
| bytes | 0.052046 | snack | 0.022460 |
| ct_srv_src | 0.044081 | Dintpkt | 0.021106 |
| ct_dst_src_ltm | 0.042994 | Dpkts | 0.020949 |
| means | 0.039951 | Sloss | 0.018946 |
| Sload | 0.039715 | tcprtt | 0.017903 |
| proto | 0.034719 | Djit | 0.013530 |
| Dload | 0.034529 | Ackdat | 0.013002 |
| service | 0.031950 | | |
| Sintpkt | | | |

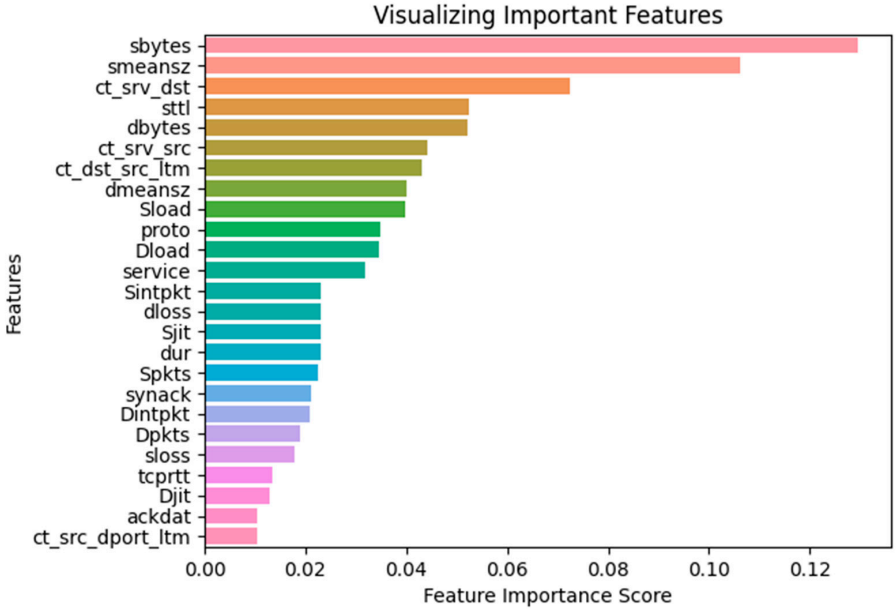



**Figure 3.** The importance values of selected features using the Random Forest algorithm.

#### 3.4.3. Data Splitting, Training, and Testing

The data from each class are randomly divided into training and testing sets, with 80% allocated to the training set and 20% allocated to the testing set. However, there is an imbalance in the labeling of APT scenario attacks. To address this issue, we employed the SMOTE technique to oversample the data, specifically focusing on the majority class of Exploitation.

#### 3.4.4. Centralized Learning Models

We selected some classical machine learning models, such as XGBoost, RF, CatBoost, and ensemble model to be applied to the whole dataset in the centralized context. The implementation of the centralized learning method requires the utilization of a central server for the purpose of aggregating the data and conducting training on the integrated model. The process of centrally training machine learning and DL models requires huge amounts of data and robust computational resources [40,41]. In this experimental study, the dataset is fed within the centralized models and distributed for the federated model for all training purposes. The voting classifier technique in ensemble learning is used, merging multiple ML models to generate an optimal result. In this approach, we conducted an ensemble voting model by fusing the state-of-the-art ML models mentioned above to enhance the

performance of attack detection. We also conducted a CNN model that comprises an input layer with dimensions corresponding to the number of features, followed by four hidden dense layers, ReLU activation, and a SoftMax output layer.

#### 3.4.5. Federated Learning Model

FL is referred to as a distributed approach in machine learning where global models are trained through the collaborative efforts of multiple clients. Data privacy is preserved through the utilization of local models contributed by participants rather than the sharing of private local data. Nevertheless, the efficacy of federated learning is heavily contingent upon the number of individuals involved and the extent of their contributions [42]. The participants updated their local models to serve in specified rounds. In federated learning, the global model is trained among a large number of clients in a distributed manner. In order to provide data privacy, computational efficacy, and a broader detection range, clients only train the model locally and share the model parameters to update the global model. This technique has great advantages for many distributed learning scenarios. Assuming that there are K clients with the same goal in federated learning to jointly train a model, at each iteration, the global model is distributed $\mathcal{M}$g to the clients, and the clients train the model individually through local data. After the local training is completed, each client sends the model parameters back to the central machine, and the global model is generated by aggregating the model parameters of each client. The update process of the global model is shown in Formula (1).

$$\mathcal{M}_g^{t+1} = \mathcal{M}_g^t + \alpha \frac{1}{K} \sum_{1}^{K} \omega_t^K \tag{1}$$

where $\mathcal{M}_g^t$ is the global model, $\omega_t^K$ is the model parameters, and $\alpha$ is a weight for the update process. The federated averaging algorithm can be solved by several methods, such as gradient computation. However, to perform this task efficiently, all clients perform local training on available data. Then, the updated model version is transmitted to the server to update the global model that is distributed to the clients. The averaging process is described in Algorithm 1.

**Algorithm 1:** CNN_FL: K Clients, Model parameter $\omega_t$, Global Model $\boldsymbol{\mathcal{M}}$g

**Input**: Dataset UNSW-NB15, $\mathcal{D}_\mathrm{k}$: local dataset, k: number of Clients $\mathcal{C}$, i: number of rounds
**Output**: Model parameter $\omega_t$, Global Model $\boldsymbol{\mathcal{M}}_\mathrm{g}$

**Server**
# Initialize the global model parameters $\omega_t$
Initialize $\boldsymbol{\mathcal{M}}_\mathrm{g}$
**for** each round i **do** # Repeat until $\boldsymbol{\mathcal{M}}_\mathrm{g}$ converges
  $\mathcal{C}\leftarrow$ set of $k$ clients # Repeat optimization of global model $\boldsymbol{\mathcal{M}}_\mathrm{g}$
**for** each client **do**
    $\omega_{t+1}^k \leftarrow$ ***updatClient***$(\omega_t)$
  $\mathcal{M}_g^{t+1} = \mathcal{M}_g^t + \alpha \frac{1}{K} \sum_{1}^{K} \omega_t^K$ #aggregate local models $\boldsymbol{\mathcal{M}}_\mathrm{k}$
  **distribute** $\mathcal{M}_g^{t+1}$ on k clients
**UpdateClient**
    For each local round
      #train local models $\boldsymbol{\mathcal{M}}_\mathrm{k}$
        Train ($\boldsymbol{\mathcal{M}}_\mathrm{k}$, $\mathcal{D}_\mathrm{k,}$ $\omega_t^k$) # train local models

This study leverages federated learning, employing the tensorflow federated library to aggregate local models through the process of averaging. The data should be read on a per-client basis, with each client's model being trained until all clients have completed the training process. Algorithm 2 illustrates the implementation process using the UNSW-NB15 dataset. The function learning.build_federated_averaging_process is responsible for computing the global model by aggregating all local models and afterwards transferring

the updated weights to the clients based on the specified number of rounds. Algorithm 2 shows the pseudo-code for implementation procedures.

**Algorithm 2:** Experimental implementation of the proposed CNN_FL model

```
START
import tensorflow federated liberary as tff
Declare No_of_Clients, No_of_rounds,lr,labels
    for c in range (No_Of_Clients):
        Read data(c)
        Preprocessing(c)
        Splitting(data)
        Call function createModel()
Trainer=tff.learning.build_federated_averaging_process(createModel,optimizer)
    build CNN model(trainlist[0])
    for j in range (No_of_rounds):
        state, metrics = trainer.next(state, train):
        print(metrics['train']['loss']}, Accuracy={metrics['train']['accuracy']
END
```

### *3.5. Experimental Setup*

For the centralized approaches, the three models, XGBoost, RF, and CatBoost, are designed with n_estimators = 150, max_depth = 10, base_score = 0.5, booster = 'gbtree', learning_rate = 0.001, max_bin = 256, and random_state = 45. The CNN_FL model is designed to construct the APT detection classifier using the Tensorflow Federated library. The model consists of an input layer, four hidden layers with ReLU activation, and a SoftMax layer, Dropout (0.2), after each hidden layer. Table 5 shows the rest parameters of the CNN_FL model [20].

**Table 5.** The parameters of the CNN_FL model.

| Parameter | Description |
|---|---|
| N_CLIENTS = 4 | The total number of clients |
| TEST_FRAC = 0.2 | The fraction of the complete dataset that will be taken for the test set |
| N_CLASSES = 5 | APT scenario attacks |
| LEARNING_RATE = 0.0001 | Learning rate |
| BATCH_SIZE = 32 | Batch size |
| N_EPOCHS = 50 | The number of epochs (times the dataset will be repeated) |
| N_ROUNDS = 20 | Rounds between clients and server to update weights |

### *3.6. Evaluation*

The models that were obtained were afterwards examined by employing the test data and taking into account the following criteria for detection:

1. Accuracy is a metric employed to assess the ratio of accurate classifications of the overall number of entries, as expressed by the following formula:

$$Accurac\ (ACC) = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

2. Precision refers to the ratio of accurately expected attack classes to the total number of predicted attack results. This can be calculated using the formula:

$$Precision\ (PRE) = \frac{TN}{TN + FP} \tag{3}$$

3. Recall refers to the ratio of correctly classified attack occurrences to the total number of samples that should have been identified as attacks. It is mathematically represented as:

$$Recal\ (REC) = \frac{TP}{TP + FN} \tag{4}$$

4. The F1-score is a metric that measures the ratio between Precision and Recall by calculating their Harmonic Mean.

$$\mathrm{F1-Score} = \frac{2\mathrm{Precision} \times \mathrm{Sensitivity}}{\mathrm{Precision} + \mathrm{Sensitivity}} \quad (5)$$

## 4. Results and Discussion

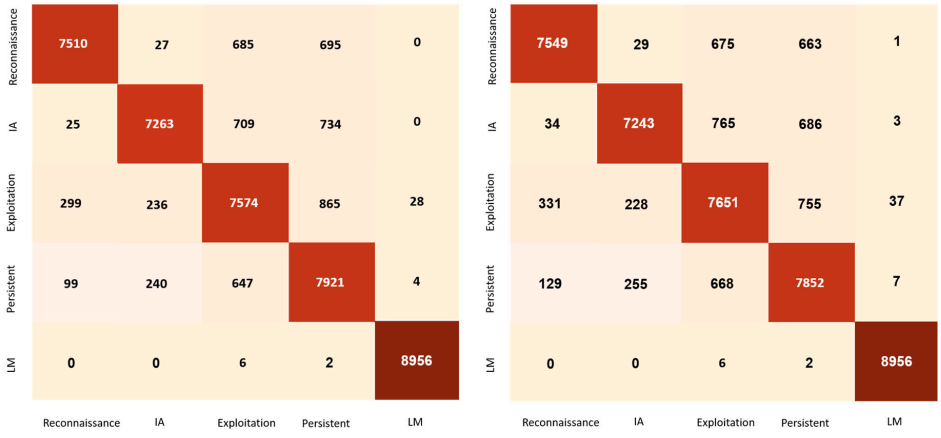
The findings of the detection process were showcased through the utilization of centralized learning as well as federated learning approaches. The experimental findings indicate that the ensemble model attained the highest accuracy of 88.15% among the centralized models. In contrast, the federated learning model CNN_FL is outperformed in terms of accuracy, with a rate of 90.01%, while also ensuring privacy, data balance, and integrity.

### *4.1. Centralized Model Results*

The implementation of the stand-alone models has been performed using machine learning techniques, including XGBoost, RF, and CatBoost, in separate experiments. Subsequently, an ensemble voting model is employed to combine these algorithms. Each classifier has been trained to classify APT scenario attacks, as specified in the previous section, using the following parameters: n_estimators = 150, max_depth = 10, and learning_rate = 0.001. The XGBoost algorithm has been identified as the most effective model for classification among many individual machine learning models. However, the ensemble learning approach produced greater classification performance, with an accuracy of 88.15%. This outperforms the XGBoost model, which achieved an accuracy of 88.00%, as shown in Table 6, and the confusion matrix of XGBoost and ensemble learning in Figure 4.

**Table 6.** Classification evaluation results of the individual ML models and ensemble model.

| Models | ACC | REC | PRE | F1-Score | AUC |
|---|---|---|---|---|---|
| XGBoost | 0.8809 | 0.8809 | 0.8879 | 0.8823 | 0.925 |
| RF | 0.8795 | 0.8795 | 0.8833 | 0.8803 | 0.8803 |
| CatBoost | 0.8529 | 0.8529 | 0.8629 | 0.8547 | 0.8547 |
| Ensemble model | 0.8815 | 0.8815 | 0.8876 | 0.8827 | 0.9259 |
| CNN | 0.8457 | 0.8457 | 0.8452 | 0.8393 | 0.8639 |




**(a)** **(b)**

**Figure 4.** The confusion matrix of machine learning models: (**a**) XGBoost and (**b**) ensemble learning models.

### *4.2. FL Model Results*

The proposed model uses the CNN algorithm for all clients. The CNN_FL model is trained using the local models of four participants. These models are trained independently without any contact between them. The training process involves 10 rounds of weight updates to the server due to four clients, as shown in Table 7. The model has an input

layer, four hidden layers with ReLU activation, and a SoftMax layer. A dropout rate of 0.2 is applied after each hidden layer, and the model is trained for 30 epochs per participant. In this study, we conducted a comparison between the outcomes of a CNN implemented using centralized learning and federated learning approaches.

**Table 7.** Distribution of dataset per clients in training the federated model.

| Stages of APT | Records | Client 1 | Client 2 | Client 3 | Client 4 |
|---|---|---|---|---|---|
| 1st: Reconnaissance | 13,987 | 3570 | 3496 | 3514 | 3407 |
| 2nd: Initial Access | 26,923 | 6817 | 6718 | 6688 | 6700 |
| 3rd: Exploitation | 44,525 | 11,005 | 11,120 | 11,098 | 11,302 |
| 4th: Persistent | 3840 | 931 | 984 | 1015 | 910 |
| 5th: Lateral Movement | 174 | 49 | 44 | 47 | 43 |
| Total | 89,449 | 22,372 | 22,362 | 22,362 | 22,362 |

The analysis revealed that the centralized learning approach achieved an accuracy rate of 84.57%, while the federated learning approach yielded an accuracy rate of 90.18%, as shown in Table 8 and Figure 5. The FL approach addresses the issue of privacy and the imbalanced distribution of APT data. Additionally, it improves the classification accuracy of models with a large increase in the number of repetition rounds, as shown in Figure 6.

**Table 8.** Classification evaluation results of the proposed CNN_FL model for multi-stage attack classification.

| **Strategy** | **ACC** | **SEN** | **SPE** | **F1-Score** | **AUC** |
|---|---|---|---|---|---|
| The proposed CNN_FL Model | 0.9018 | 0. 9018 | 0.9011 | 0.9009 | 0.9322 |

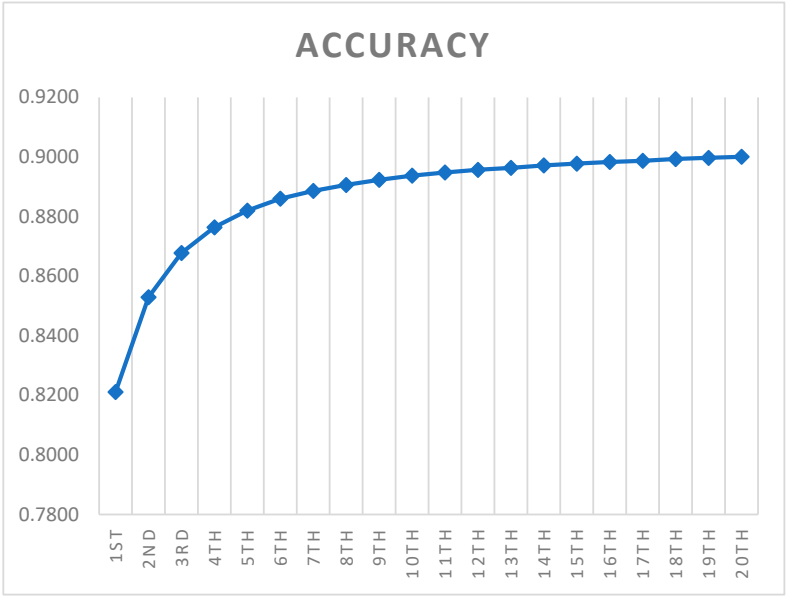



**Figure 5.** The proposed CNN_FL model result of accuracy per round with updated weights to the server.

Figure 7 shows the performance comparison findings for all ML models, including the ensemble model and the proposed CNN_FL model. For ML models, the XGBoost algorithm demonstrates superior classification performance across all evaluation measures, particularly in terms of total accuracy and F1-score, achieving an 88.09% accuracy. The RF model demonstrates predictive performance, achieving an overall accuracy of 87.95%. In contrast, the CatBoost and CNN models showed comparatively inferior performance when compared to other ML models.

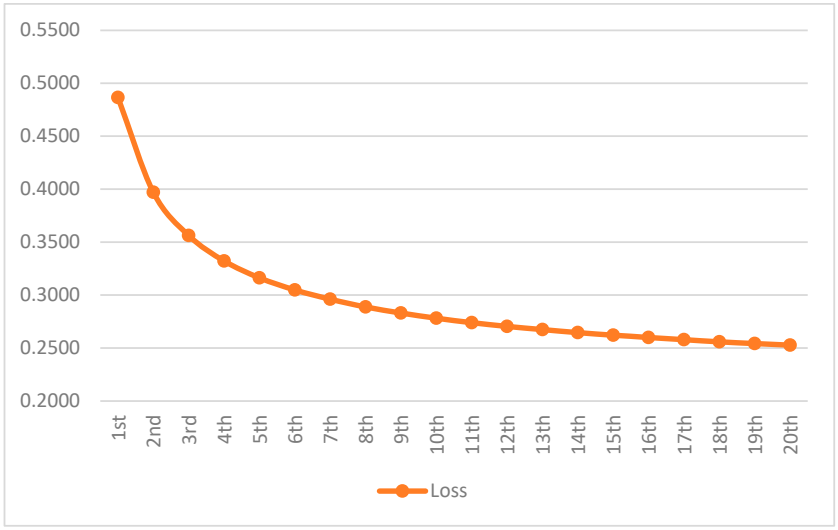



**Figure 6.** The proposed CNN_FL model result of loss per round with updated weights to the server.

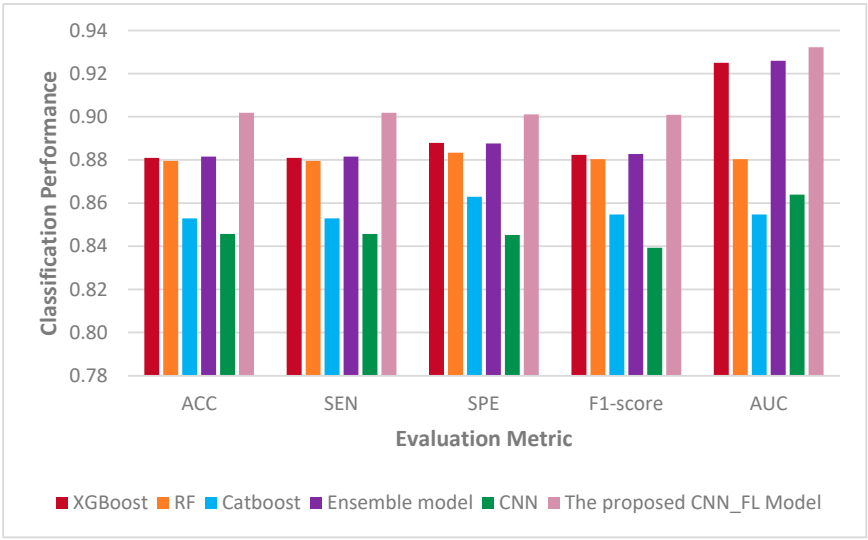



**Figure 7.** The comparison results for the proposed CNN_FL model against the ensemble learning and ML models.

The ensemble learning model is formed by combining the previous three machine learning models. The accuracy of the multi-stage attacks employing ensemble learning surpassed that of ML models, as evidenced by the evaluation performance results. In contrast, the CNN_FL model exhibited superior performance, achieving an accuracy rate of 90.18%.

### *4.3. Comparison of the Results of Work with Related Works*

Quantitatively comparing the detection performance of the proposed system with other similar efforts is not a direct task due to variations in the evaluation process and the absence of standardized datasets for advanced attack scenarios. Additionally, certain studies were carried out to examine the association between alerts and the process of determining the procedures of the advanced attack scenario for host scoring. In contrast, other studies focused on the application of FL for classifying the threats discovered from network traffic. In our research, we consider alerts generated from security devices. Then, we train the models to extract the correlation between individual alerts. Table 9 shows a comparative analysis of our suggested methodology and other relevant studies in terms of the reported performance metrics for alerts or classification.

**Table 9.** A comparative analysis of the outcomes obtained from the proposed study and those of other related research endeavors.

| Approach | ACC |
|---|---|
| M. A. Ferrag [20] | Identified 61 features with high correlations for traffic but did not include scenarios for alerts and generating APT. |
| M. Khosravi et al. [21] | Modeling attacks process, generating meta-alerts with APT steps and host score for all risk levels, not AI classification |
| W. Giura, P., and Wang [26] | 81.80% |
| X. Wang and K. Zheng [27] | 83.30% |
| Lajevardi et al. [28] | 84.21% |
| Yin, Y. [29] | 84.26% |
| M. Khosravi and B. T. Ladani [21] | 87.10% |
| The proposed work | 90.01% |

## 5. Conclusions and Further Work

Alerts reported by various security devices can assist in the recognition of advanced multi-staged attack scenarios using an intelligent analysis of the attack data. Moreover, novel and unknown attacks can be detected using the prediction facility of trained models. For intrusion prevention systems to operate as intended, it is necessary to identify the next phase of any attack with a sequence of steps, which can be derived after monitoring several previous steps of the attack in progress. This study proposed a centralized and federated learning architecture consisting of multi-components in order to recognize multi-stage attacks and further predict the expected next stage/stages. These functions have been performed through alert correlation, which is constructed using a training task resulting from distributed components. The experimental results showed excellent accuracy in the detection of attack stages and the prediction of the next attack stage. Phases of preparation of data were conducted, including handling null values, encoding categorical values, selecting features, oversampling, and, finally, training models. The evaluation of the FL system based on the CNN algorithm demonstrates higher performance. The findings from the federated learning experiment demonstrate that the CNN_FL model outperforms in terms of accuracy in detecting advanced attack scenarios. The ensemble mode combines three machine learning models, namely XGBoost, RF, and CatBoost. In addition, FL can be used to satisfy other design requirements, such as privacy, data diversity, real-time model updates, and improved prediction. In future work, it is recommended to explore the utilization of various ensemble DL models [43], hybrid models of CNN [44], XAI [45], and transformer models [46] in combination with federated learning. In addition, we will broaden our analysis to include real-world network traffic. Attacks targeting the model itself, such as poisoning attacks, require a robust security framework. Available datasets in the IDS field that simulate multi-stage attacks are generally limited, and as a result, developing more datasets is a significant requirement.

**Author Contributions:** Conceptualization, F.M.A.; Methodology, H.K.A. and F.M.A.; Software, H.K.A. and F.M.A.; Validation, H.K.A. and F.M.A.; Investigation, F.M.A.; Resources, H.K.A. and F.M.A.; Data curation, H.K.A.; Writing—original draft, H.K.A.; Writing—review & editing, F.M.A.; Supervision, F.M.A.; Project administration, F.M.A. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** The datasets used in this paper are publicly available at: https://research.unsw.edu.au/projects/unsw-nb15-dataset (accessed on 6 September 2023).

**Acknowledgments:** The authors would like to thank the Deanship of Graduate Studies at Al-Jouf University, Saudi Arabia for funding and supporting this research through the initiative of DGS, Graduate Students Research Support (GSR) at Al-Jouf University, Saudi Arabia.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References


1. Bhattacharya, S.; Maddikunta, P.K.; Kaluri, R.; Singh, S.; Gadekallu, T.R.; Alazab, M.; Tariq, U. A Novel PCA-Firefly Based XGBoost Classification Model for Intrusion Detection in Networks Using GPU. *Electronics* **2020**, *9*, 219. [CrossRef]
2. Preuveneers, D.; Rimmer, V.; Tsingenopoulos, I.; Spooren, J.; Joosen, W.; Ilie-Zudor, E. Chained Anomaly Detection Models for Federated Learning: An Intrusion Detection Case Study. *Appl. Sci.* **2018**, *8*, 2663. [CrossRef]
3. Bhatti, D.G.; Virparia, P.V. Soft Computing-Based Intrusion Detection System With Reduced False Positive Rate. In *Design and Analysis of Security Protocol for Communication*; Wiley: Hoboken, NJ, USA, 2020; pp. 109–139. [CrossRef]
4. Anwar, S.; Mohamad Zain, J.; Zolkipli, M.F.; Inayat, Z.; Khan, S.; Anthony, B.; Chang, V. From Intrusion Detection to an Intrusion Response System: Fundamentals, Requirements, and Future Directions. *Algorithms* **2017**, *10*, 39. [CrossRef]
5. Jadidi, Z.; Hagemann, J.; Quevedo, D. Multi-step attack detection in industrial control systems using causal analysis. *Comput. Ind.* **2022**, *142*, 103741. [CrossRef]
6. Sharma, A.; Gupta, B.B.; Singh, A.K.; Saraswat, V.K. A novel approach for detection of APT malware using multi-dimensional hybrid Bayesian belief network. *Int. J. Inf. Secur.* **2023**, *22*, 119–135. [CrossRef]
7. Manzoor, E.; Milajerdi, S.M.; Akoglu, L. Fast Memory-efficient Anomaly Detection in Streaming Heterogeneous Graphs. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 1035–1044. [CrossRef]
8. Ansari, M.S.; Bartos, V.; Lee, B. Shallow and Deep Learning Approaches for Network Intrusion Alert Prediction. *Procedia Comput. Sci.* **2020**, *171*, 644–653. [CrossRef]
9. Zhang, J.; Zhao, Y.; Wang, J.; Chen, B. FedMEC: Improving Efficiency of Differentially Private Federated Learning via Mobile Edge Computing. *Mob. Netw. Appl.* **2020**, *25*, 2421–2433. [CrossRef]
10. Michie, D.; Spiegelhalter, D.J.; Taylor, C.C. *Machine Learning, Neurall and Statistical Classification*; Ellis Horwood Series in Artificial Intelligence: New York, NY, USA, 1994; Volume 13.
11. Liu, H.; Lang, B. Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey. *Appl. Sci.* **2019**, *9*, 4396. [CrossRef]
12. Dong, S.; Wang, P.; Abbas, K. A survey on deep learning and its applications. *Comput. Sci. Rev.* **2021**, *40*, 100379. [CrossRef]
13. Martin, L. Cyber Kill Chain. 2014. Available online: http://cyber.lockheedmartin.com/ (accessed on 27 July 2023).
14. Rahman, S.A.; Tout, H.; Talhi, C.; Mourad, A. Internet of Things Intrusion Detection: Centralized, On-Device, or Federated Learning? *IEEE Netw.* **2020**, *34*, 310–317. [CrossRef]
15. Chen, Z.; Lv, N.; Liu, P.; Fang, Y.; Chen, K.; Pan, W. Intrusion Detection for Wireless Edge Networks Based on Federated Learning. *IEEE Access* **2020**, *8*, 217463–217472. [CrossRef]
16. Zhao, Y.; Li, M.; Lai, L.; Suda, N.; Civin, D.; Chandra, V. Federated learning with non-IID data. *arXiv* **2018**, arXiv:1806.00582. [CrossRef]
17. Truong, H.T.; Ta, B.P.; Le, Q.A.; Nguyen, D.M.; Le, C.T.; Nguyen, H.X.; Do, H.T.; Nguyen, H.T.; Tran, K.P. Light-weight federated learning-based anomaly detection for time-series data in industrial control systems. *Comput. Ind.* **2022**, *140*, 103692. [CrossRef]
18. Wilkens, F.; Ortmann, F.; Haas, S.; Vallentin, M.; Fischer, M. Multi-Stage Attack Detection via Kill Chain State Machines. In Proceedings of the 3rd Workshop on Cyber-Security Arms Race, Virtual, 15 November 2021; pp. 13–24. [CrossRef]
19. Ghafir, I.; Hammoudeh, M.; Prenosil, V.; Han, L.; Hegarty, R.; Rabie, K.; Aparicio-Navarro, F.J. Detection of advanced persistent threat using machine-learning correlation analysis. *Future Gener. Comput. Syst.* **2018**, *89*, 349–359. [CrossRef]
20. Ferrag, M.A.; Friha, O.; Hamouda, D.; Maglaras, L.; Janicke, H. Edge-IIoTset: A New Comprehensive Realistic Cyber Security Dataset of IoT and IIoT Applications for Centralized and Federated Learning. *IEEE Access* **2022**, *10*, 40281–40306. [CrossRef]
21. Khosravi, M.; Ladani, B.T. Alerts Correlation and Causal Analysis for APT Based Cyber Attack Detection. *IEEE Access* **2020**, *8*, 162642–162656. [CrossRef]
22. Li, Z.; Chen, J.; Zhang, J.; Cheng, X.; Chen, B. Detecting Advanced Persistent Threat in Edge Computing via Federated Learning. In Proceedings of the Security and Privacy in Digital Economy: First International Conference, SPDE 2020, Quzhou, China, 30 October–1 November 2020; Springer: Singapore, 2020; pp. 518–532. [CrossRef]
23. Neuschmied, H.; Winter, M.; Stojanović, B.; Hofer-Schmitz, K.; Božić, J.; Kleb, U. APT-Attack Detection Based on Multi-Stage Autoencoders. *Appl. Sci.* **2022**, *12*, 6816. [CrossRef]
24. Xia, Q.; Dong, S.; Peng, T. An Abnormal Traffic Detection Method for IoT Devices Based on Federated Learning and Depthwise Separable Convolutional Neural Networks. In Proceedings of the 2022 IEEE International Performance, Computing, and Communications Conference (IPCCC), Austin, TX, USA, 11–13 November 2022; pp. 352–359. [CrossRef]
25. Thi, H.T.; Son, N.D.H.; Duy, P.T.; Pham, V.-H. Federated Learning-Based Cyber Threat Hunting for APT Attack Detection in SDN-Enabled Networks. In Proceedings of the 2022 21st International Symposium on Communications and Information Technologies (ISCIT), Xi'an, China, 27–30 September 2022; pp. 1–6. [CrossRef]
26. Giura, P.; Wang, W. Using large scale distributed computing to unveil advanced persistent threats. *Sci. J.* **2012**, *1*, 93–105.
27. Wang, X.; Zheng, K.; Niu, X.; Wu, B.; Wu, C. Detection of command and control in advanced persistent threat based on independent access. In Proceedings of the 2016 IEEE International Conference on Communications (ICC), Kuala Lumpur, Malaysia, 22–27 May 2016; pp. 1–6. [CrossRef]
28. Lajevardi, A.M.; Amini, M. A semantic-based correlation approach for detecting hybrid and low-level APTs. *Future Gener. Comput. Syst.* **2019**, *96*, 64–88. [CrossRef]

29. Yin, Y.; Jang-Jaccard, J.; Xu, W.; Singh, A.; Zhu, J.; Sabrina, F.; Kwak, J. IGRF-RFE: A hybrid feature selection method for MLP-based network intrusion detection on UNSW-NB15 dataset. *J. Big Data* **2023**, *10*, 15. [CrossRef]
30. Kasongo, S.M.; Sun, Y. Performance Analysis of Intrusion Detection Systems Using a Feature Selection Method on the UNSW-NB15 Dataset. *J. Big Data* **2020**, *7*, 105. [CrossRef]
31. Hairab, B.I.; Elsayed, M.S.; Jurcut, A.D.; Azer, M.A. Anomaly Detection Based on CNN and Regularization Techniques Against Zero-Day Attacks in IoT Networks. *IEEE Access* **2022**, *10*, 98427–98440. [CrossRef]
32. Almaiah, M.A.; Almomani, O.; Alsaaidah, A.; Al-Otaibi, S.; Bani-Hani, N.; Hwaitat, A.K.; Al-Zahrani, A.; Lutfi, A.; Awad, A.B.; Aldhyani, T.H. Performance Investigation of Principal Component Analysis for Intrusion Detection System Using Different Support Vector Machine Kernels. *Electronics* **2022**, *11*, 3571. [CrossRef]
33. Moustafa, N.; Slay, J. UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In Proceedings of the 2015 Military Communications and Information Systems Conference (MilCIS), Canberra, Australia, 10–12 November 2015; pp. 1–6. [CrossRef]
34. Cox, K.J.; Gerg, C. *Managing Security with Snort & IDS Tools: Intrusion Detection with Open Source Tools*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2004.
35. Roesch, M. Snort: Lightweight intrusion detection for networks. In Proceedings of the LISA '99: 13th Systems Administration Conference, Seattle, WA, USA, 7–12 November 1999; Volume 99, pp. 229–238.
36. Waleed, A.; Jamali, A.F.; Masood, A. Which open-source IDS? Snort, Suricata or Zeek. *Comput. Netw.* **2022**, *213*, 109116. [CrossRef]
37. Wang, J.; Xu, M.; Wang, H.; Zhang, J. Classification of Imbalanced Data by Using the SMOTE Algorithm and Locally Linear Embedding. In Proceedings of the 2006 8th International Conference on Signal Processing, Guilin, China, 16–20 November 2006. [CrossRef]
38. Strom, B.E.; Applebaum, A.; Miller, D.P.; Nickels, K.C.; Pennington, A.G.; Thomas, C.B. Mitre Att&ck: Design and Philosophy. 2018. Available online: https://www.mitre.org/news-insights/publication/mitre-attck-design-and-philosophy (accessed on 30 October 2023).
39. Alhaj, T.A.; Siraj, M.M.; Zainal, A.; Elshoush, H.T.; Elhaj, F. Feature Selection Using Information Gain for Improved Structural-Based Alert Correlation. *PLoS ONE* **2016**, *11*, e0166017. [CrossRef]
40. Fauzi, M.A.; Yang, B.; Blobel, B. Comparative Analysis between Individual, Centralized, and Federated Learning for Smartwatch Based Stress Detection. *J. Pers. Med.* **2022**, *12*, 1584. [CrossRef] [PubMed]
41. Khan, M.; Glavin, F.G.; Nickles, M. Federated Learning as a Privacy Solution—An Overview. *Procedia Comput. Sci.* **2023**, *217*, 316–325. [CrossRef]
42. Ma, X.; Liao, L.; Li, Z.; Lai, R.X.; Zhang, M. Applying Federated Learning in Software-Defined Networks: A Survey. *Symmetry* **2022**, *14*, 195. [CrossRef]
43. Al-Hejri, A.M.; Al-Tam, R.M.; Fazea, M.; Sable, A.H.; Lee, S.; Al-antari, M.A. ETECADx: Ensemble Self-Attention Transformer Encoder for Breast Cancer Diagnosis Using Full-Field Digital X-ray Breast Images. *Diagnostics* **2022**, *13*, 89. [CrossRef]
44. Houssein, E.H.; Mohamed, O.; Abdel Samee, N.; Mahmoud, N.F.; Talaat, R.; Al-Hejri, A.M.; Al-Tam, R.M. Using deep DenseNet with cyclical learning rate to classify leukocytes for leukemia identification. *Front. Oncol.* **2023**, *13*, 1230434. [CrossRef]
45. Nwakanma, C.I.; Ahakonye, L.A.; Njoku, J.N.; Odirichukwu, J.C.; Okolie, S.A.; Uzondu, C.; Ndubuisi Nweke, C.C.; Kim, D.S. Explainable Artificial Intelligence (XAI) for Intrusion Detection and Mitigation in Intelligent Connected Vehicles: A Review. *Appl. Sci.* **2023**, *13*, 1252. [CrossRef]
46. Al-Tam, R.M.; Al-Hejri, A.M.; Narangale, S.M.; Samee, N.A.; Mahmoud, N.F.; Al-Masni, M.A.; Al-Antari, M.A. A Hybrid Workflow of Residual Convolutional Transformer Encoder for Breast Cancer Classification Using Digital X-ray Mammograms. *Biomedicines* **2022**, *10*, 2971. [CrossRef] [PubMed]