Journal of Network and Systems Management (2023) 31:3
https://doi.org/10.1007/s10922-022-09691-3

Cyber Threat Intelligence Sharing Scheme Based on Federated Learning for Network Intrusion Detection

Mohanad Sarhan[1] · Siamak Layeghy[1] · Nour Moustafa[2] · Marius Portmann[1]

Received: 7 May 2022 / Revised: 29 July 2022 / Accepted: 20 September 2022 /
Published online: 7 October 2022

Abstract

The uses of machine learning (ML) technologies in the detection of network attacks have been proven to be effective when designed and evaluated using data samples originating from the same organisational network. However, it has been very challenging to design an ML-based detection system using heterogeneous network data samples originating from different sources and organisations. This is mainly due to privacy concerns and the lack of a universal format of datasets. In this paper, we propose a collaborative cyber threat intelligence sharing scheme to allow multiple organisations to join forces in the design, training, and evaluation of a robust ML-based network intrusion detection system. The threat intelligence sharing scheme utilises two critical aspects for its application; the availability of network data traffic in a common format to allow for the extraction of meaningful patterns across data sources and the adoption of a federated learning mechanism to avoid the necessity of sharing sensitive users' information between organisations. As a result, each organisation benefits from the intelligence of other organisations while maintaining the privacy of its data internally. In this paper, the framework has been designed and evaluated using two key datasets in a NetFlow format known as NF-UNSW-NB15-v2 and NF-BoT-IoT-v2. In addition, two other common scenarios are considered in the evaluation process; a centralised training method where local data samples are directly shared with other organisations and a localised training method where no threat intelligence is shared. The results demonstrate the efficiency and effectiveness of the proposed framework by designing a universal ML model effectively classifying various benign and intrusive traffic types originating from multiple organisations without the need for inter-organisational data exchange.

Keywords Cyber threat intelligence · Federated learning · Machine learning · NetFlow · Network intrusion detection

✉ Mohanad Sarhan
m.sarhan@uq.net.au

[1] University of Queensland, Brisbane, Australia

[2] University of New South Wales, Canberra, Australia

1 Introduction

Network Intrusion Detection Systems (NIDS) are tools used to detect intrusive network traffic as they penetrate a digital computer network [1]. They aim to preserve the three key principles of information security; confidentiality, integrity, and availability [2]. NIDSs scan and analyse the incoming traffic for malicious indicators that may present a threat or harm to the target network. There are two main types of NIDS; (1) signature-based NIDS, which operates by scanning for a set of previously known attack rules or Indicators Of Compromise (IOC) [3] such as source/destination IPs and ports, hash values or domain names in an incoming network feed. This traditional method works efficiently against known attack scenarios where the complete set of IOCs has been previously identified and registered within the NIDS. However, signature-based NIDSs have been vulnerable to zero-day attacks where there is a lack of knowledge of IOCs related to the occurrence of activity [4]. In addition, the detection of modern advanced and persistent threats such as Cobalt Strikes [5] requires a sophisticated depth of behavioural change monitoring [6], where the usage of traditional IOC is not sufficient in their detection. Therefore, the focus of NIDS development has shifted towards the modern type of NIDS with enhanced machine learning (ML) capabilities [7].

ML is a branch of Artificial Intelligence (AI) extensively used with great success to empower decision-making systems across various domains [8]. ML models operate by extracting and learning meaningful patterns from historical data during the training process. The models then apply the learnt semantics to classify or predict unseen data samples into their respective classes or values. The intelligence capability of ML has motivated its usage in many industries to provide a deeper level of analysis to automate and assist in complex decision-making tasks [9]. Overall, ML enhances the performance and efficiency of systems without being explicitly programmed [10], by learning complex patterns that are not trivial to recognize by domain experts. As such, ML has been welcomed in the development of NIDS to overcome the limitations faced by signature-based NIDS and to improve cyber attack detection using an intelligent defense layer [11]. ML-based NIDS capabilities have been widely adopted in the security of modern computer networks to detect zero-day and advanced cyber threats. ML models are capable of learning the distinguishing semantic patterns between intrusive and benign network traffic and using it to detect incoming traffic with malicious intent. Therefore, the focus on the network attacks' behavioural patterns and the lack of dependency on identified IOCs [12] has attracted attention towards the development of ML-based NIDS to detect network attacks.

In this paper, we propose a federated learning-based methodology to enable collaboration between multiple organisations to share Cyber Threat Intelligence (CTI). The collaborative sharing of valuable CTI in a secure manner will facilitate the design of an effective ML-based NIDS [13]. This will increase the exposure of the learning NIDS model to a multitude of network environments, including various benign traffic and malicious attack scenarios that occur in different organisational networks [14]. This is an important aspect considering a real-world implementation,

as each computer network often incorporates a unique statistical distribution as demonstrated in [15]. Therefore, the performance of the ML models might not generalise across different organisational networks or attack types. Although the proposed scheme has a great number of benefits, it also raises certain challenges, which we address in this paper. Unlike centralised learning approaches, federated learning enables collaboration between organisations while keeping training data samples secure and preserved internally within each organisation's perimeter. Decoupling the ability to learn from other organisations' network intelligence and attack experiences from the need for explicit exchange of sensitive data is important.

The outcome of the proposed method is a common and robust ML-based NIDS not limited to a single organisation's experience and available local training samples. The enhanced model is trained on heterogeneous data collected over a variety of heterogeneous networks, each of which presents its unique behaviour of benign and malicious traffic. Similarly to traditional federated learning approaches, a single global organisation is required to orchestrate the whole process by initiating a global ML model. Each participating organisation downloads a copy of the global model and trains it using its local data samples locally. The updated model parameters are uploaded back to the global organisation where they are aggregated to improve the global model before sending it back to each organisation for deployment. This presents a single federated learning round and can be repeated several times to reach a reliable state of performance.

The key contributions of this paper are the proposal of a novel privacy-preserving CTI scheme and the evaluation of its performance using two key and non-Independent and Identically Distributed (IID) [16] NIDS datasets. The results are analysed and compared to centralised and localised learning approaches to demonstrate the effectiveness of the proposed scheme. In Sect. 2, the differences between each ML training approach adopted in this paper are illustrated. Section 3 explores some of the key related works and highlights their limitations. The motivations and benefits of the proposed intelligence sharing scheme are discussed in Sect. 4. In Sect. 5, we perform an empirical evaluation and comparison of a collaboratively designed ML-based NIDS to demonstrate the robustness and benefits of the proposed framework. Finally, we conclude this paper in Sect. 6 and list some of the critical future works.

2 Background

ML technologies have been used widely across different domains and applications. As such, there are general guidelines and practises to be considered when designing a learning model. The choice of which process or technique to adopt depends on the available resources such as training data samples, data sensitivity, data heterogeneity, computing power, storage requirements, etc. Therefore, it is relatively easier to apply ML technologies in particular areas compared to the rest. In the application of ML-based NIDS, the privacy and security of data samples used in the training and testing stages are critical. Sharing user information with third parties and other entities could present a significant breach of data

privacy. Therefore, data scarcity is often faced when designing ML-based NIDSs using real-world datasets, due to the limited amount of data samples collected or insufficient data classes available.

Moreover, heterogeneity in network data samples often causes the problem of a lack of generalisation. Consequently, a trained high-achieving model in a certain network structure might not be effective in detecting intrusions in another network environment. This is due to the unique Standard Operating Environments (SOEs) [17] in each organisational network and different types of experienced threats, which is reflected in the statistical distribution of the utilised NIDS datasets. ML models are highly dependent on the extraction of meaningful patterns to distinguish between benign and intrusive traffic. As such, a wider variety of data samples are required in the training of an intrusion detection model. Taking into account the data scarcity and heterogeneity in the application of ML-based NIDS, we discuss each of the generally adopted common ML scenarios.

2.1 Localised Learning

A localised learning method involves local data samples collected from a single source, the learning and testing occur locally [18], where it is generally more effective with a larger amount of data. This method often provides a high detection accuracy over IID data samples with a similar probability distribution to the training data samples. However, since network traffic is often heterogeneous in nature [19], due to a multitude of safe applications/services and malicious threats/intrusions, localised learning approaches do not generalise or scale well with rapidly increasing and changing network traffic [20]. This is mainly due to the fact that the learning model is exposed to a limited variety of network traffic scenarios, hence it has a limited experience of other instances. As a result, modern research has adopted centralised learning methods to overcome some of the limitations faced by localised learning approaches.

2.2 Centralised Learning

Centralised learning is where local data samples are collected from various sources and transmitted to a central server [21]. The central entity holds all data samples, ideally reflecting an overall statistical representation of the organisational network structure. The learning and testing stages are carried out on the central server, where the learning models experience and extract useful patterns from heterogeneous network traffic. Therefore, NIDSs can effectively detect network intrusions in non-IID data samples [22]. However, centralised learning requires direct sharing of data samples between participants and a central entity [23]. This presents serious privacy and security concerns due to the nature of the transmitted data. Network data often contain sensitive information related to users' browsing sessions, applications, and services utilised, often revealing critical endpoint details.

2.3 Federated Learning

Federated learning is an advanced technique of ML designed to address certain limitations of centralised learning. A federated learning setup allows for the training of a model across multiple decentralised sources, each holding local data samples without exchanging them [23]. The key benefit of following a federated learning approach is to preserve and maintain the privacy and security of local data samples, as they are no longer shared with other entities [24]. In addition, due to a lack of a central entity storing all data samples, there is lower latency, power and storage requirements due to the reduced transmission of data [25]. This is often a motivation for usage in Internet of Things (IoT) networks where federated learning has been widely adopted [26]. In the context of NIDS, this enables the design of smarter ML models, as they are exposed to a large number of heterogeneous data samples generated using various sources, while ensuring the privacy of network users [27].

3 Related Works

A large number of research papers have aimed to adopt a federated learning approach in the design of ML-based NIDS. Although most of the papers focused on the structure and parameters of the adopted learning model, all training and evaluation stages were conducted using a single organisational network dataset divided over several local endpoints. Therefore, the data samples used in the learning model are not very different in nature as they all originate from the same network environment. To the best of our knowledge, no paper has considered the requirements of designing an ML-based NIDS using several heterogeneous data sources collected across multiple non-IID NIDS datasets.

In [28], Abdul Rahman et al. evaluated the detection performance of NIDS designed using centralised, on-device (localised), and federated learning approaches. The comparison was carried out using safe and malicious network data samples from the NSL-KDD dataset, which is an outdated dataset (20+ years) and does not represent modern network characteristics and threats [29]. As a single dataset is used, the federated learning approach splits the dataset amongst several endpoints. The results show that federated learning outperforms the on-device learning method and achieves similar detection performance in a centralised manner while maintaining the privacy of local data samples.

Mothukuri et al. [30] explored different parameters of a federated learning-based anomaly detection approach to detect IoT intrusions using decentralised data samples. The paper explored two deep learning models; Long Short Term Memory (LSTM) and Gated Recurrent Units (GRU) with various window sizes and an additional Random Forest ensemble component to combine the predictions from different layers. The evaluation was carried out on the Modbus-based dataset which consists of benign IoT telemetry traffic and four attack scenarios. The results show that their approach outperformed the centralised ML approach with an increased detection rate and reduced the number of false alarms. Similarly, this approach does not consider other attack scenarios or benign patterns in other network environments.

In paper [31], Popoola et al. proposed a Deep Neural Network (DNN) model to detect zero-day botnet traffic with a high classification performance. By following a federated learning approach, the method guarantees to preserve data privacy and security, in addition, it has a lower communication overhead, network latency, and memory space for storage of training data. The paper explored sixteen DNN models to determine the optimal neural architecture for efficient classification. The traditional *FedAvg* algorithm [32] is used for the aggregation of local model parameters. The performance of the federated learning methodology in the detection of zero-day botnet attacks is compared with centralised and localised methods where the federated learning achieves similar performance to the centralised method while preserving data privacy.

Zhao et al. [33], proposed an LSTM-based framework to detect host intrusions using the user's input of shell commands. The shell command block is fed into the network model to segment the word and convert it into a vector representation. The LSTM model maps the bidirectional semantic association between the words to improve the accuracy of predictions of malicious commands. The framework utilises a federated learning method to maintain the privacy of local datasets during training. The open-source SEA dataset is used to evaluate the proposed framework. The results are compared with standard LSTM and Convolutional Neural Network (CNN) models trained in a centralised method. The proposed method achieves a 99.21% accuracy compared to 99.51% and 95.48% by the LSTM and CNN modes, respectively.

In [34], a semi-supervised federated learning scheme (SSFL) via knowledge distillation for NIDSs is proposed. Unlabelled data samples are leveraged to enhance the classifier performance. A CNN model is built to extract deep features from network traffic packets. A discriminator module is added to the CNN model to avoid the failure of distillation training caused by non-IID data. A communication-efficient federated learning method that uses a combination of hard-label strategy and voting mechanisms is adopted. The evaluation of the proposed scheme on the N-BaIoT dataset shows that it can achieve better performance and lower communication costs compared to three state-of-the-art models.

Recent research has addressed aspects of the federated learning process, which is an active research area, such as communication cost, privacy, security, and resource allocation. However, no papers have considered the application of CTI sharing in ML-based NIDSs. Each of the above related works considers a single network environment for the federated training and evaluation, where multiple endpoints hold IID data samples similar to the overall data. In the real world, an organisation's network data is unique in its statistical distribution to its SOE and malicious threats experienced. Therefore, these approaches may neither generalise nor scale well with the rapid growth of network services and attacks available in other organisational networks. In this work, we investigate the applicability of collaborative CTI sharing based on federated learning for network intrusion detection. Several heterogeneous and non-IID datasets are used, each representing a unique network environment and attack classes.

4 Cyber Threat Intelligence Sharing

Data are considered the most valuable and powerful tool an organisation could have in the 21st century. A lot of organisations in many sectors depend on data to provide insights and extract meaningful patterns through data analytic engines. ML has provided organisations with intelligent algorithms, capable of extracting and learning semantic attributes from historical data [10] to provide insights for the prediction or classification of data. As such, ML capabilities have been adopted in the design of NIDSs to monitor and preserve the digital perimeters of organisations' networks. To achieve this goal, network data traffic has been captured from organisational networks to design an ML model. During the training process, the model learns the distinguishing patterns between benign and intrusive traffic, which can be used in future detection. ML-based NIDS has been proven to be reliable in the detection of zero-day and modern attacks by utilising the malicious behaviour and attack chains rather than a set of IOCs implemented in signature-based NIDS.

4.1 Motivation

A large amount of research work has been carried out to improve the overall performance of ML-based NIDS. Current traditional systems have generally been designed in a localised ML manner where models learn traffic patterns from a single network environment. This method provides the learning model with high visibility into a target organisational network's SOE activities and malicious threats encountered in the past. However, as an ML model only knows what it learns, traditional ML-based NIDS are limited to an organisation's experience independently and might be incapable to generalise across non-IID network sources. There is a high chance of varying distributions in different networks due to the unique SOEs and their associated threats implemented within organisations. This presents a significant risk to organisations due to the rapidly changing network environments caused by modern work practices, such as new services or an incoming advanced threat such as zero-day attacks.

Therefore, the current method of ML-based NIDS design does not scale with the rapid growth of network benign and attack variants as there is a requirement to collect the corresponding training data samples. We used the change of networks as a baseline in our experiments, that is, when an ML model is trained on one network source and evaluated in a different network environment. This measures how well a learning model generalises across other networks. Another key limitation of current approaches is the requirement to collect a large amount of training data samples to increase the performance and generalisation of the ML model and avoid overfitting over a few data samples [35]. Therefore, particularly in the design of ML-based NIDSs, following a supervised method adopted in this paper, a large number of benign and attack-labelled data samples are required. The lack of labelled training data is a major challenge for small organisations aiming to effectively design an intrusion detection model.

Due to the lack of shared intelligence, organisations can not benefit from the usage patterns of safe traffic or malicious intrusions occurring in other organisations. Therefore, a collaborative ML approach between organisations is necessary for the design of enhanced NIDS. Three ML scenarios are considered for this purpose. The localised learning method is inapplicable as it involves a single source of organisational data. This is used for comparison purposes in this paper as a non-collaborative scenario where an organisation does not share intelligence. The centralised learning scenario requires a direct sharing of data between organisations and a central entity to allow for the training of an ML model. This method enables the learning model to extract useful patterns from various data samples collected over the participating organisational networks to overcome the issues faced in the localised learning scenario.

However, network data often present sensitive information such as user browsing sessions, applications accessed, and critical endpoint details, e.g. domain controllers and firewalls. Therefore, following a centralised learning approach poses privacy, security, and transactional risks that organisations would generally avoid. Moreover, recent strict laws such as the General Data Protection Regulation (GDPR) [36], Health Insurance Portability and Accountability Act (HIPAA) [37], and Payment Services Directive Two (PSD2) [38] are enforced to protect consumer data privacy and address concerns related to unauthorised sharing of user-related information. The violation of privacy conserving regulations often presents serious legal concerns and hefty fines of up to $20 million [39] in the case of a GDPR breach. Unfortunately, centralised learning requires a central entity to collect, store, and analyse network data samples collected from participating organisations, which could make it unfeasible to conduct in the real world.

It is important to note that the sharing of CTI is not uncommon in the security field. In fact, many organisations using signature-based NIDS heavily rely on CTI platforms, such as Malware Information Sharing Platform (MISP) [40] a widely-used open-source platform. CTI platforms develop utilities and documentation for more effective threat intelligence by sharing IOCs related to external threat actors. Organisations generally integrate a threat intelligence feed with their traditional signature-based NIDS to provide high detection accuracy against associated attacks. However, in ML-based NIDS, there is a requirement to share both benign and malicious network data samples for the learning model to extract the distinguishing patterns. The sharing of network data samples often reveals information related to the targeted user, endpoint or application depending on the attributes provided.

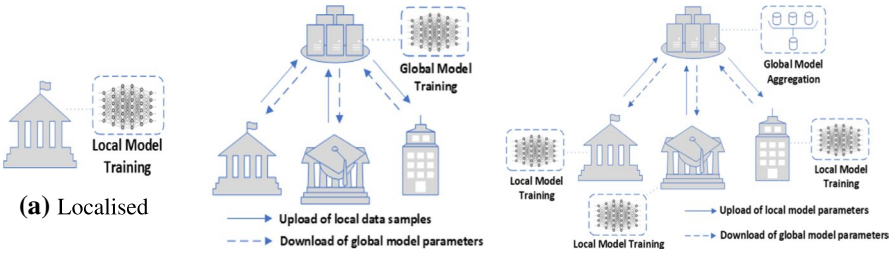
4.2 Collaborative Federated Learning

To overcome the limitations mentioned above, the sharing of CTI between organisations via a federated learning approach is required to increase the knowledge base of the learning models while maintaining the privacy of user information. The learning model is exposed to a wider range of benign and attack variants

to achieve reliable detection accuracy across previously unseen traffic in a given organisation. The proposed framework allows organisations to join forces by sharing their cyber intelligence and insights. In addition, organisations that do not collect and store a sufficient amount of network traffic required for the training of a learning model are now able to design an effective ML-based by collaborating with other organisations. As each participant contributing with a minimum amount of data samples would permit the design of a successful system, our approach tackles the data scarcity problem and makes it possible to design an ML-based NIDS without the need to collect a large amount of training data. The three learning scenarios considered in this paper are illustrated in Fig. 1.

Moreover, by adopting a federated learning approach, the local network data samples remain distributed across the organisations, hence persevering the privacy and integrity of sensitive users' network information. A federated learning setup includes a global server that coordinates and orchestrates the independent training of the local models. In this paper, the global server is hosted within a participant organisation, however, this framework enables it to be hosted externally within a trusted mediator such as cloud computing. One of the main requirements of this framework is for each participating organisation to hold its local network data traffic in a common logging format. The benefits of having a standard feature set are many and are explained here [41] and [42]. In this framework, a common feature set enables streamlined federated learning as the global model can extract meaningful patterns across a standard set of data features. The global model structure and parameters are designed to be compatible with the agreed network logging format.

The complete process is defined in Algorithm 1, where w is the set of initialised parameters, t is the federated learning round, K represents the participant organisations indexed by k, and m is the global learning rate. B is the size of the local training batch, E is the number of local epochs, $\mathcal{P}$ is the local training set, l is the prediction loss in example (x_i, y_i) and n is the local learning rate. Similar to standard federated learning approaches; Step 1: the process is triggered by a global server initiating an ML model with a pre-defined architecture and



(a) Localised **(b)** Centralised **(c)** Federated

Fig. 1 Machine learning scenarios

parameters. Step 2: the model is forwarded to each participant. Step 3: the model is trained and enhanced locally using the internal network data samples. Step 4: the updated weights are sent back to the global server. Step 5: the *FedAvg* technique [32] is followed, where the server aggregates the weights uploaded by each organisation to generate an enhanced intrusion detection model with an improved set of parameters designed over each participant's network. The *FedAvg* process is defined as

$$w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{m_k}{m} w_{t+1}^k \tag{1}$$

These five steps present a single federated learning round and can be repeated several times to achieve better detection performance in all network environments.

Algorithm 1: Federated Averaging for NIDS

Global Server Executes :
Initialize w_0
for *each federated learning round t = 1,2,...:* **do**
  $S_t \leftarrow$ (Set of K organisations)
  for *each organisation* $k \in S_t$ *:* **do**
    $w_{t+1}^k \leftarrow$ Local Organisation Update (k, w_t)
    $w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{m_k}{m} w_{t+1}^k$
  end
end
Local Organisation Update (k, w)**:**
Run on organisation k
$\mathscr{B} \leftarrow$ (split $\mathscr{P}_k$ into batches of size B)
for *each local epoch i from 1 to E:* **do**
  for *batch* $b \in \mathscr{B}$*:* **do**
    $w \leftarrow w - \mathrm{n}\nabla \mathrm{l}(w; b)$
    return w to server
  end
end

In this paper, we take the application of federated learning a step further, where each local client is observed as a single organisation with a unique network of heterogeneous data samples. The key outcome is the design of a robust ML-based NIDS obtained from a collaboration between organisations without the need to share data with other participants to preserve data privacy. The final model is capable of detecting a wider range of attacks originating from several sources, which are crucial in an organisational defence system. This provides a robust learning model with global intelligence and insights capable of distinguishing between benign and attack heterogeneous traffic. Such smart models would possibly lead to a lower false alarm rate in case of a variation of the benign traffic distribution caused by a modification of the SOE due to the learning from several networks' safe usage. Moreover, a higher detection rate of

advanced and zero-day attacks is promising due to the extraction of malicious patterns from a wider range of attacks targeting several organisational networks.

5 Experiments

To evaluate the feasibility and performance of our proposed collaborative CTI sharing scheme based on federated learning for NIDS, we use two widely used key NIDS datasets. Each dataset has been collected over a different network, each consisting of a different set of benign applications and malicious attack scenarios. Therefore, each dataset represents a certain organisational network with a unique SOE and malicious events encountered. The datasets also hold a very distinctive statistical distribution as presented here [15]. This matches the assumption of obtaining non-IID datasets collected over different real-world networks. Although the datasets are unique in their applications, protocols, and attack scenarios, they share a common set of features based on NetFlow v9 [43], a de facto standard protocol in the networking industry. In this paper, the NF-UNSW-NB15-v2 and NF-BoT-IoT-v2 datasets are used to simulate two organisations collaborating in the design of a universal ML-based NIDS. By following a federated learning-based technique, each dataset is preserved internally in the learning and testing stages. The datasets' structure and format are explained below and compared in Table 1;

- NF-UNSW-NB15-v2 [44]: A NetFlow-based dataset released in 2021 containing nine attack scenarios; Exploits, Fuzzers, Generic, Reconnaissance, DoS, Analysis, Backdoor, Shellcode, and Worms. The dataset is generated by converting the publicly available pcap files of the UNSW-NB15 dataset [45] to 43 NetFlow v9 features using the nprobe tool [46]. The total number of data flows is 2,390,275 out of which 95,053 (3.98%) are attack samples and 2,295,222 (96.02%) are benign. The source dataset (UNSW-NB15) is a widely used NIDS dataset in the research community. UNSW-NB15 was released in 2015 by the Cyber Lab of the Australian Center for Cyber Security (ACCS). The IXIA Perfect Storm tool was configured to simulate benign network traffic and synthetic attack scenarios.
- NF-BoT-IoT-v2 [44]: An IoT NetFlow-based dataset released in 2021 containing four attack scenarios; DDoS, DoS, Reconnaissance, and Theft. The dataset is generated by converting the publicly available pcap files of the BoT-IoT [47]

Table 1 Dataset comparison

Dataset	NF-UNSW-NB15-v2	NB-BoT-IoT-v2
Attack samples	95,053	37,628,460
Benign samples	2,295,222	135,037
Total samples	2,390,275	37,763,497
Attack classes	9	4
Tools	IXIA Perfect Storm	Ostinato and Node-red
Format	NetFlow v9	NetFlow v9

dataset to 43 NetFlow v9 features using the nprobe [46] tool. The total number of data flows is 37,763,497 network data flows, where the majority are attack samples; 37,628,460 (99.64%) and 135,037 (0.36%) are benign. The source dataset (BoT-IoT) is generated by an IoT-based network environment that consists of normal and botnet traffic. BoT-IoT was released in 2018 by the Cyber Range Lab of the ACCS. The non-IoT and IoT traffic was generated using the Ostinato and Node-red tools, respectively, and Tshark is used to capture network packets.

5.1 Experimental Methodology

Three different approaches are considered in the evaluation process; federated, centralised and localised learning scenarios, as shown in Fig. 1. In the federated learning approach, there are two participating clients (organisations), and a single global server. Each client holds a unique network traffic dataset collected from their respective environment. This represents a real-world scenario with two organisations are participating in the CTI operation. Client 1 represents the NF-UNSW-NB15-v2 dataset and client 2 represents the NF-BoT-IoT-v2 dataset. The traffic data distribution is illustrated in Table 1. Each organisation downloads an initialised ML model from a global server to be trained on its local data samples locally. The global server receives the updated parameter set from each organisation and averages the weights together into a global model. For the centralised learning scenario, each participating organisation sends their local data samples to a central server for the training and testing of the ML model on the complete set of aggregated data. In the localised learning scenario, there are no collaborations between organisations; therefore, the model is trained on each organisation's limited local data samples.

The evaluation metrics used to evaluate the performance of the ML models are defined in Table 2. The metrics are calculated in a binary format based on True Positive (TP) and True Negative (TN), representing the number of correctly classified attack and benign data samples, respectively. In addition to the False Positive (FP) and False Negative (FN) represent the numbers of incorrectly classified benign and

Table 2 Evaluation metrics

Metric	Definition	Equation
Accuracy	The percentage of correctly classified samples	$\frac{TP+TN}{TP+FP+TN+FN} \times 100$
Detection rate (DR)	The percentage of correctly classified total attack samples	$\frac{TP}{TP+FN} \times 100$
False alarm rate (FAR)	The percentage of incorrectly classified benign samples	$\frac{FP}{FP+TN} \times 100$
Area under the curve (AUC)	The area underneath the DR and FAR plot curve	N/A
F1 score	The harmonic mean of the model's precision and DR	$2 \times \frac{DR \times Precision}{DR + Precision}$
Time	The time required in seconds to complete the training of the ML model	N/A

Table 3 Training parameters

Parameter	Value
Local epochs	3
Batch size	2048
Local optimiser	Adam
Local learning rate	0.001
Loss function	Binary crossentropy
Federated learning rounds*	10
Server optimiser*	Adam
Server learning rate*	0.05

*Only applies to federated learning

Table 4 Hyperparameters for both DNN and LSTM

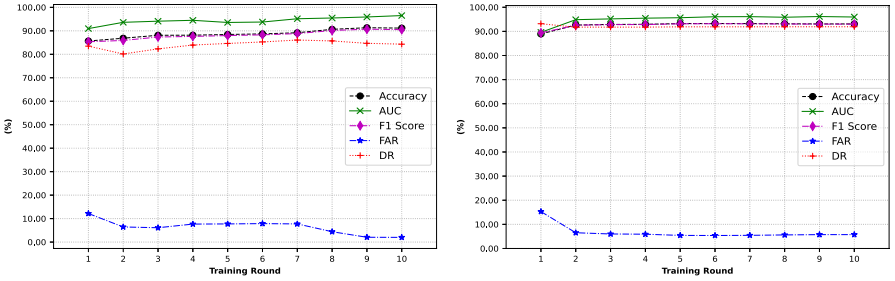
	Nodes	Activation function
Input layer	39 (number of input features)	N/A
Hidden layer 1	12	Relu
Hidden layer 2	6	Relu
Hidden layer 3	3	Relu
Output layer	1	Sigmoid

attack data samples, respectively. The experiments were conducted using Google's Tensorflow Federated (TFF) framework for the federated learning scenario and Tensorflow framework [48] for the centralised and localised scenarios. The datasets are pre-processed by dropping the flow identifiers, such as source/destination IP and port attributes, to avoid bias towards the attacking and victim end nodes. Undersampling has been used to address the extreme imbalance of the datasets. Each dataset has been divided into training and testing sets in a ratio of 70% to 30%, respectively. A Min-Max scaler has been applied to normalise each dataset's values, defined as

$$X_* = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{2}$$

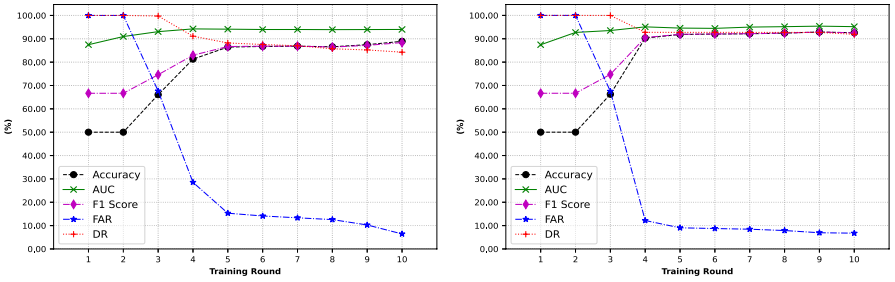
where X_* is the output value ranging from 0 to 1, X is the input value and $X_{\max}$ and $X_{\min}$ are the maximum and minimum values of the feature respectively. The parameters used in this paper to design the ML experiments are represented in Table 3.

It is important to note that, while the discovery stage was conducted by exploring a large number of hyperparameter sets to obtain reliable detection performance, the full exploration of the parameter space is not covered in this paper. The performance of the ML models and the overall proposed scheme can be further improved by optimising the set of parameters adopted. Two key ML models adopted in the ML-based NIDS have been designed to demonstrate the effectiveness of the



(a) NF-UNSW-NB15-v2 **(b)** NF-BoT-IoT-v2

Fig. 2 Federated learning using a DNN model



(a) NF-UNSW-NB15-v2 **(b)** NF-BoT-IoT-v2

Fig. 3 Federated learning using an LSTM model

proposed framework. The same parameters were used across the three scenarios for a fair comparison. A Deep Neural Network (DNN) and Long Short-Term Memory (LSTM) have been used with their parameters defined in Table 4. The hyperparameters were identically designed to provide a fair comparison of their performance. In both models, there is a dropout of 40% of the input units between each hidden layer to help prevent overfitting of the local client's data.

In the DNN model, the data is fed forward via an input layer through three hidden layers and the predictions are calculated in the output layer. Each dense layer consists of multiple nodes, each performing the Relu activation function, with randomly initialised weighted connections. During the training stage, the connections are optimised using the Adam algorithm to map the high-level features to the desired output through a process known as back-propagation. In the LSTM model, sequential information in the input data can be captured through an internal memory that stores a sequence of inputs. The input is converted to a 3-dimensional shape to be compatible with the requirements of the LSTM layer, and passed through three hidden layers made up of interconnected nodes, each performing the Relu function.

5.2 Results

The results in this section are collected over the test sets after the training has been conducted using the respective training scenario. We start with federated learning separately in Figs. 2 and 3, where the detection performance of the DNN and LSTM models, respectively, is evaluated in each dataset. The caption of each sub-figure identifies the test dataset used in the evaluation process. A set of results was collected after each federated learning round to analyse the improvement of the ML-based NIDS after each aggregation process. The results are plotted on line graphs, where the percentage value is presented on the y-axis, the number of federated learning rounds is listed on the x-axis, and each line presents a different evaluation metric.

In Fig. 2, the DNN model achieves a reliable performance across the two datasets, where it rapidly converges to its maximum performance after the second round and fairly stabilises thereafter. There is a slight drop in FAR in both datasets after the first federated learning, where the remaining metrics increase by around 5% in the NF-UNSW-NB15-v2 and NF-BoT-IoT-v2 datasets. In Fig. 3, the LSTM model requires a larger number of federated learning rounds to reach a reliable detection performance. During the first three rounds, the model was achieving a poor performance of 50% accuracy in both datasets. However, the performance increased rapidly between the fourth and seventh rounds until it converged to its maximum reliable performance. The FAR dropped from 100% to almost 8% during the 10 rounds of federated learning in both datasets.

Tables 5 and 6 compare the three training scenarios showing the complete set of evaluation metrics achieved in the NF-UNSW-NB15-v2 and NF-BoT-IoT-v2 test datasets, respectively. The results are grouped by the ML used and the scenario followed in the training process. In addition, the time required to complete the training stage is measured in seconds. In the federated learning scenario, the results achieved after the tenth round are presented in tables. It is important to note that for the federated learning scenario, the time is measured over ten rounds, which might not be required to achieve a reliable performance as demonstrated in Fig. 2.

Table 5 NF-UNSW-NB15-v2: binary-class detection

	ACC (%)	AUC (%)	F1 (%)	DR (%)	FAR (%)	Time (s)
DNN						
Federated	91.16	96.50	90.51	84.32	2.00	31.2
Centralised	99.38	99.47	99.38	99.42	0.67	5.83
Localised	51.34	59.89	7.89	4.17	1.48	3.77
LSTM						
Federated	88.92	94.00	88.38	84.27	6.43	51.92
Centralised	95.80	98.46	95.65	92.55	0.96	9.57
Localised	52.32	79.75	10.82	5.78	1.15	7.19

Table 6 NF-BoT-IoT-v2: binary-class detection

	ACC (%)	AUC (%)	F1 (%)	DR (%)	FAR (%)	Time (s)
DNN						
Federated	93.08	95.95	93.01	91.92	5.74	31.2
Centralised	93.83	96.74	93.84	93.99	6.32	5.83
Localised	86.21	86.89	86.92	91.66	19.25	3.10
LSTM						
Federated	92.57	95.18	92.52	91.90	6.75	51.92
Centralised	93.90	94.76	93.76	91.71	3.92	9.57
Localised	88.52	88.87	88.87	91.66	14.62	6.62

In Table 5, the binary class detection results achieved in the NF-UNSW-NB15-v2 dataset are presented, where the federated and centralised learning scenarios achieve a reliable performance of 91.16% and 99.38% accuracy using the DNN model and 88.92% and 95.80% using the LSTM model, respectively. The lower performance noted in the federated learning approach was mainly due to a higher number of FAR of 2.00% and 6.43% using the DNN and LSTM models compared to 0.67% and 0.96% in the centralised scenario. In the localised learning scenario, the lowest training time was achieved due to the smaller number of training samples by a single organisation. However, the model was unable to detect most of the attacks present in the NF-UNSW-NB15-v2 dataset after training in the NF-BoT-IoT-v2 dataset achieving an inadequate DR of 4.17% and 5.78% using the DNN and LSTM models, respectively.

In Table 6, the results of the detection of intrusion of the binary class collected on the NF-BoT-IoT-v2 test set are presented. A similar pattern is observed in the NF-UNSW-NB15-v2 dataset, where federated and centralised learning scenarios achieve reliable intrusion detection performance. The accuracy achieved by the federated and centralised learning methods is 93.08% and 93.83% using DNN and 92.57% and 93.90% using LSTM, respectively. The attack DR is slightly higher using both ML models in the federated learning method compared to the centralised learning method. Surprisingly, the localised learning approach achieved significantly better results on the NF-BoT-IoT-v2 test set when trained on the NF-UNSW-NB15-v2 dataset. This was not the same case the other way around. This could indicate the presence of meaningful patterns in NF-UNSW-NB15-v2 to help the model identify attacks in NF-BoT-IoT-v2. The accuracy achieved is 86.21% using the DNN model and 88.52% using the LSTM model, the performance drop is mainly caused by a high FAR of 19.25% and 14.62%, respectively.

In Tables 7 and 8, we deep dive into the results of the NF-UNSW-NB-v2 and NF-BoT-IoT-v2 datasets to measure each attack DR separately in a multi-class manner. The multi-class performances have been statistically calculated based on the binary classification tasks, where the detection rate of each attack class is measured. The results are grouped by the ML used and the scenario followed in the training process, and the federated learning results are measured after the tenth training round.

Table 7 NF-UNSW-NB15-v2: multi-class detection

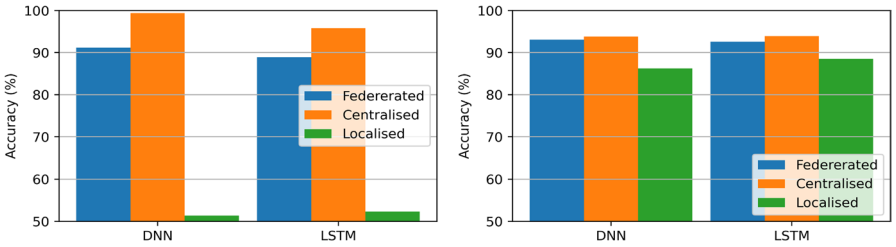
	Analysis	Backdoor	DoS	Exploits	Fuzzers	Generic	Recon	Shellcode	Worms	Average
DNN										
Federated	83.62	82.69	85.91	85.94	86.47	87.12	85.12	88.39	82.35	85.06
Centralised	100.00	99.23	98.22	99.15	99.51	99.84	99.82	100.00	100.00	99.41
Localised	3.62	3.23	4.37	5.09	4.12	1.17	6.21	2.10	6.12	4.00
LSTM										
Federated	83.62	82.99	84.78	85.09	85.69	85.78	84.28	89.73	85.29	85.25
Centralised	100.00	98.46	92.17	80.98	98.89	98.03	99.82	100.00	100.00	96.48
Localised	4.35	5.22	6.79	5.96	6.05	1.79	10.04	3.74	16.33	6.70

Table 8 NF-BoT-IoT-v2: multi-class detection

	DDoS	DoS	Recon	Theft	Average
DNN					
Federated	91.66	92.24	91.96	100.00	93.40
Centralised	99.98	95.31	44.46	100.00	84.94
Localised	98.04	92.97	36.33	100.00	81.84
LSTM					
Federated	92.40	93.16	92.88	100.00	94.61
Centralised	98.14	93.07	39.01	100.00	82.56
Localised	98.05	93.47	34.67	100.00	81.55

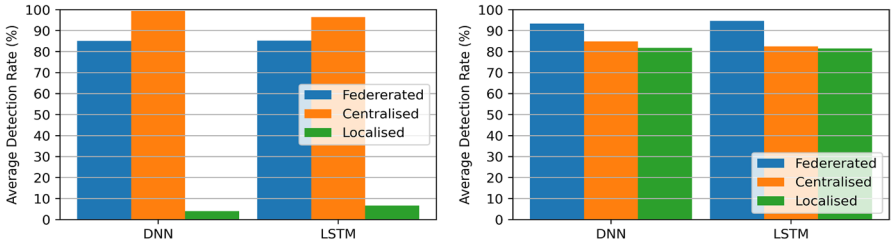
Furthermore, we calculate the average of the attack DR to compare the three scenarios based on the number of attack behaviours detected. In Table 7, the highest DR is achieved by the centralised method in the NF-UNSW-NB15-v2 with an almost perfect DR of 99.41% using the DNN model and 96.48% using the LSTM model. Analysis, shellcode, and worm attacks were fully detected using both models. The federated learning approach came in second with an average DR of around 85% using both models. As seen in previous results, the localised scenario is unreliable in the detection of any attacks in the NF-UNSW-NB15-v2 dataset with an average DR of 6.70%.

As demonstrated in Table 8, the federated learning approach is superior to other approaches in the detection of attacks available in the NF-BoT-IoT-v2 dataset with an average DR of 93.40% using the DNN model and 94.61% using the LSTM model. The centralised and localised learning approaches achieved 84.94% and 81.84% using the DNN model and 82.56% and 81.55% using the LSTM model, respectively. The reason for the average drop in DR is only due to the lack of recognition of reconnaissance attack samples, where the centralised and localised learning methods achieved 44.46% and 36.33%, respectively, compared to 91.96% detected by the federated learning method using the DNN model. Similarly, using the LSTM model, 39.01% and 34.67% reconnaissance attack samples were detected using centralised and localised learning methods, whereas the federated learning approach detected 92.88%.



(a) NF-UNSW-NB15-v2 **(b)** NF-BoT-IoT-v2

Fig. 4 Binary-class comparison



(a) NF-UNSW-NB15-v2 **(b)** NF-BoT-IoT-v2

Fig. 5 Multiclass comparison

In Figs. 4 and 5, a summary of the key results is presented in bar graphs to compare the binary- and multi-classes detection results following each ML scenario. In Fig. 4, the accuracy evaluation metric is used to compare the three methods, where the centralised learning method achieved the best performance using both ML models, followed by the federated learning method achieving a very similar overall detection performance. In a localised learning scenario, both models were able to transfer the information learnt from NF-UNSW-NB15-v2 to NF-BoT-IoT-v2. However, this was not the case in the reverse direction, where both models failed to achieve reliable detection performance. In Fig. 5, the average attack DR is displayed on the y axis, where centralised learning and federated learning approaches were the most effective in detecting attacks available in the NF-UNSW-NB15-v2 and NF-BoT-IoT-v2 datasets, respectively. The localised learning method did not detect most of the attacks available in the NF-UNSW-NB-v2 dataset.

The collected results demonstrate certain benefits and limitations in each of the three approaches adopted in this paper. In the federated and centralised learning approaches, both models achieved reliable detection performance on both datasets, which can be improved by tuning and optimising the hyperparameters. In the case of localised learning, the models were effective in transferring the information learnt from one dataset but not the other. Explainable AI [49] techniques could be used to provide insight into this behaviour. Furthermore, the proposed methodology could face certain limitations, such as that it may not be efficient with extremely heterogeneous data and certain domain adaptation techniques [50] may be required to deal with statistical variations. Additional verification steps can be performed, such as t-tests to measure the similarity between test and training sets prior to the training stage, although that would increase training resources, cost, and time.

Overall, a large number of experiments were conducted to evaluate and compare the performance of three ML scenarios, i.e., federated learning, centralised and localised learning. For a fair evaluation, two different ML models were used in the training and testing stages. The results demonstrate that the best performances were often achieved by following the centralised learning approach. However, this is not possible without breaching network users' privacy and sharing sensitive data with third parties. In the

real world, this might make centralised learning approaches unfeasible and costly for organisations. Therefore, the proposed scenario of a collaborative federated learning approach, which achieves similar performance to the centralised learning approach, makes it superior in terms of feasibility and preserving user privacy.

6 Conclusion

In this paper, a collaborative federated learning scheme is proposed to allow the sharing of CTI between organisations to design a more effective ML-based NIDS. The collaboration between organisations attracts many benefits including the design of a robust learning model capable of detecting intrusions effectively across various organisational networks. The heterogeneity of the network data samples exposes the model to a wider variety of SOEs and attack scenarios. This reflects the real-world behaviour where each network accounts for a unique statistical distribution that ML model performance might not generalise across. The detection performance of the models is compared to centralised and localised learning scenarios. The results demonstrate that the performance of federated learning is superior to the localised learning approach and similar to the centralised learning approach. However, the centralised method can not be used without breaching data privacy and security which renders it unfeasible in the real world. Therefore, we sacrifice a relatively small amount of classification performance for privacy and hence enable practical inter-organisational information sharing for collaborative ML-based NIDS. Future work involves improving the detection performance against lateral movement and persistent attacks using the temporal aspect of the network data features. In addition, the issue of maintaining the privacy in the context of Federated Learning represent another important direction for future work. For example techniques such as Differential Privacy or homomorphic encryption present promising solutions.

Author Contributions M.S. and S.L. wrote the main manuscript text. N.M. and M.P. designed the framework and evaluation methodology. All authors reviewed the manuscript.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions. No funding was received to assist with the preparation of this manuscript.

Data Availability All data generated or analysed during this study are included in this published article Sarhan, M., Layeghy, S. & Portmann, M. Towards a Standard Feature Set for Network Intrusion Detection System Datasets. Mobile Netw Appl (2021).

Declarations

Conflict of interest The authors have no competing interests to declare that are relevant to the content of this article.

Ethical Approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. Javaid, A., Niyaz, Q., Sun, W., Alam, M.: A deep learning approach for network intrusion detection system. EAI Endorsed Trans. Secur. Saf. **3**(9), e2 (2016)
2. Whitman, M.E., Mattord, H.J.: Principles of Information Security. Cengage Learning, Boston (2011)
3. Ashoor, A.S., Gore, S.: Importance of intrusion detection system (ids). Int. J. Sci. Eng. Res. **2**(1), 1–4 (2011)
4. Garcia-Teodoro, P., Diaz-Verdejo, J., Maciá-Fernández, G., Vázquez, E.: Anomaly-based network intrusion detection: techniques, systems and challenges. Comput. Secur. **28**(1–2), 18–28 (2009)
5. van der Eijk, V., Schuijt, C.: Detecting cobalt strike beacons in netflow data
6. Bhatt, P., Yano, E.T., Gustavsson, P.: Towards a framework to detect multi-stage advanced persistent threats attacks. In: 2014 IEEE 8th International Symposium on Service Oriented System Engineering, pp. 390–395, IEEE (2014)
7. Sarhan, M., Layeghy, S., Portmann, M.: Feature analysis for ML-based IIoT intrusion detection. arXiv:2108.12732 (2021)
8. Goodfellow, I., Bengio, Y., Courville, A.: Machine learning basics. Deep Learn. **1**(7), 98–164 (2016)
9. Jordan, M.I., Mitchell, T.M.: Machine learning: trends, perspectives, and prospects. Science **349**(6245), 255–260 (2015)
10. Mahesh, B.: Machine learning algorithms-a review. IJSR **9**, 381–386 (2020)
11. Tsai, C.-F., Hsu, Y.-F., Lin, C.-Y., Lin, W.-Y.: Intrusion detection by machine learning: a review. Expert Syst. Appl. **36**(10), 11994–12000 (2009)
12. Bhuyan, M.H., Bhattacharyya, D.K., Kalita, J.K.: Network anomaly detection: methods, systems and tools. IEEE Commun. Surv. Tutor. **16**(1), 303–336 (2013)
13. Brown, R., Lee, R.M.: The evolution of cyber threat intelligence (CTI): 2019 sans CTI survey. SANS Institute. https://www.sans.org/white-papers/38790/. Accessed 12 July 2021 (2019)
14. Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., Chandra, V.: Federated learning with non-IID data. arXiv:1806.00582 (2018)
15. Layeghy, S., Gallagher, M., Portmann, M.: Benchmarking the benchmark-analysis of synthetic NIDS datasets. arXiv:2104.09029 (2021)
16. Clauset, A.: A brief primer on probability distributions. In: Santa Fe Institute (2011)
17. Aupek, A. et al.: Architectural design of enterprise wide standard operating environments (2006)
18. Youssef, A., Aerts, J.-M., Vanrumste, B., Luca, S.: A localised learning approach applied to human activity recognition. IEEE Intell. Syst. (2020)
19. Kato, N., Fadlullah, Z.M., Mao, B., Tang, F., Akashi, O., Inoue, T., Mizutani, K.: The deep learning vision for heterogeneous network traffic control: Proposal, challenges, and future perspective. IEEE Wirel. Commun. **24**(3), 146–153 (2016)
20. Bhole, Y., Popescu, A.: Measurement and analysis of http traffic. J. Netw. Syst. Manage. **13**(4), 357–371 (2005)
21. Nardi, M., Valerio, L., Passarella, A.: Centralised vs decentralised anomaly detection: when local and imbalanced data are beneficial. In: Third International Workshop on Learning with Imbalanced Domains: Theory and Applications, pp. 7–20, PMLR (2021)

22. Abbasi, M., Shahraki, A., Taherkordi, A.: Deep learning for network traffic monitoring and analysis (NTMA): a survey. Comput. Commun. (2021)
23. Yang, Q., Liu, Y., Cheng, Y., Kang, Y., Chen, T., Yu, H.: Federated learning. Synth. Lect. Artif. Intell. Mach. Learn. **13**(3), 1–207 (2019)
24. Truex, S., Baracaldo, N., Anwar, A. , Steinke, T., Ludwig, H., Zhang, R., Zhou, Y.: A hybrid approach to privacy-preserving federated learning. In: Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security, pp. 1–11 (2019)
25. Yang, K., Jiang, T., Shi, Y., Ding, Z.: Federated learning via over-the-air computation. IEEE Trans. Wireless Commun. **19**(3), 2022–2035 (2020)
26. Imteaj, A., Thakker, U., Wang, S., Li, J., Amini, M.H.: A survey on federated learning for resource-constrained IoT devices. IEEE Internet Things J. (2021)
27. 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. **8**(12), 2663 (2018)
28. Rahman, S.A., Tout, H., Talhi, C., Mourad, A.: Internet of things intrusion detection: centralized, on-device, or federated learning? IEEE Netw. **34**(6), 310–317 (2020)
29. Siddique, K., Akhtar, Z., Aslam Khan, F., Kim, Y.: Kdd cup 99 data sets: a perspective on the role of data sets in network intrusion detection research. Computer **52**(2), 41–51 (2019)
30. Mothukuri, V., Khare, P., Parizi, R.M., Pouriyeh, S., Dehghantanha, A., Srivastava, G.: Federated learning-based anomaly detection for IoT security attacks. IEEE Internet Things J. (2021)
31. Popoola, S.I., Ande, R., Adebisi, B., Gui, G., Hammoudeh, M., Jogunola, O.: Federated deep learning for zero-day botnet attack detection in IoT edge devices. IEEE Internet Things J. (2021)
32. McMahan, B., Moore, E., Ramage, D., Hampson, S., Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial Intelligence and Statistics, pp. 1273–1282, PMLR (2017)
33. Zhao, R., Yin, Y., Shi, Y., Xue, Z.: Intelligent intrusion detection based on federated learning aided long short-term memory. Phys. Commun. **42**, 101157 (2020)
34. Zhao, R., Wang, Y., Xue, Z., Ohtsuki, T., Adebisi, B., Gui, G.: Semi-supervised federated learning based intrusion detection method for internet of things. IEEE Internet Things J. (2022)
35. Dietterich, T.: Overfitting and undercomputing in machine learning. ACM Comput. Surv. (CSUR) **27**(3), 326–327 (1995)
36. Truong, N., Sun, K., Wang, S., Guitton, F., Guo, Y.: Privacy preservation in federated learning: an insightful survey from the GDPR perspective. Comput. Secur. **110**, 102402 (2021)
37. Herold, R., Beaver, K.: The Practical Guide to HIPAA Privacy and Security Compliance. CRC Press, Boca Raton (2003)
38. Cortet, M., Rijks, T., Nijland, S.: Psd2: the digital transformation accelerator for banks. J. Paym. Strateg. Syst. **10**(1), 13–27 (2016)
39. Seo, J., Kim, K., Park, M., Park, M., Lee, K.: An analysis of economic impact on IoT under GDPR. In: 2017 International Conference on Information and Communication Technology Convergence (ICTC), pp. 879–881 (2017)
40. Wagner, C., Dulaunoy, A., Wagener, G., Iklody, A.: Misp: the design and implementation of a collaborative threat intelligence sharing platform. In: Proceedings of the 2016 ACM on Workshop on Information Sharing and Collaborative Security, pp. 49–56 (2016)
41. Sarhan, M., Layeghy, S., Portmann, M.: An explainable machine learning-based network intrusion detection system for enabling generalisability in securing IoT networks. arXiv:2104.07183 (2021)
42. Portmann, M.: Netflow datasets for machine learning-based network intrusion detection systems. In: Big Data Technologies and Applications: 10th EAI International Conference, BDTA 2020 and 13th EAI International Conference on Wireless Internet, WiCON 2020, Virtual Event, December 11, 2020: Proceedings, vol. 371, p. 117, Springer Nature (2021)
43. Claise, B., Sadasivan, G., Valluri, V., Djernaes, M.: Cisco systems netflow services export version 9 (2004)
44. Sarhan, M., Layeghy, S., Moustafa, N., Portmann, M.: Towards a standard feature set of NIDS datasets. arXiv:2101.11315 (2021)

45. Moustafa, N., Slay, J.: Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set). In: 2015 Military Communications and Information Systems Conference (MilCIS), pp 1–6, IEEE (2015)
46. Deri, L., SpA, N.: nprobe: an open source netflow probe for gigabit networks. In: TERENA Networking Conference, pp 1–4 (2003)
47. Koroniotis, N., Moustafa, N., Sitnikova, E., Turnbull, B.: Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset. Futur. Gener. Comput. Syst. **100**, 779–796 (2019)
48. Google, "Tensorflow." https://www.tensorflow.org
49. Samek, W., Montavon, G., Vedaldi, A., Hansen, L.K., Müller, K.-R.: Explainable AI: Interpreting, Explaining and Visualizing Deep Learning, vol. 11700. Springer, Berlin (2019)
50. Coulter, R., Zhang, J., Pan, L., Xiang, Y.: Domain adaptation for windows advanced persistent threat detection. Comput. Secur. **112**, 102496 (2022)

Mohanad Sarhan is a PhD student at the School of Information and Electrical Engineering (ITEE), at the University of Queensland, Australia. He obtained his Bachelor's degree in 2017 from the Queensland University of Technology, Australia. He is an experienced InfoSec Professional, working for a number of years for a managed cybersecurity service provider, deploying and managing enterprise security solutions. His areas of research interest include cybersecurity, intrusions detection systems and machine learning.

Siamak Layeghy received his Ph.D. in software-defined networking from the University of Queensland, Brisbane, Australia, where he is a Research Fellow with the School of Information Technology and Electrical Engineering. His research interests include software-defined networks, cybersecurity, AI, and machine learning.

Nour Moustafa is a Senior Lecturer, and a leader of Intelligent Security at SEIT, UNSW Canberra, Australia. He was a Post-doctoral Fellow at UNSW Canberra from June 2017 to December 2018. He received his Ph.D. degree in the field of Cyber Security from UNSW Canberra in 2017. He obtained his Bachelor's and Master's degree in Computer Science in 2009 and 2014, respectively, from the Faculty of Computer and Information, Helwan University, Egypt. His areas of interest include cyber security, network security, IoT security, intrusion detection systems, statistics, Deep learning and machine learning techniques. He has been awarded the 2020 prestigious Australian Spitfire Memorial Defence Fellowship award. He is also a Senior IEEE Member, ACM Distinguished Speaker, as well as CSCRC and Spitfire Fellow.

Marius Portmann received his Ph.D. degree in electrical engineering from the Swiss Federal Institute of Technology (ETH), Zurich, in 2002. He is currently an Associate Professor at the University of Queensland, Australia. His research interests include general networking, in particular SDN, wireless networks, pervasive computing, and cyber security.