algorithms

MDPI

Article

Federated Learning for Intrusion Detection in the Critical Infrastructures: Vertically Partitioned Data Use Case

Evgenia Novikova [1,2,†], Elena Doynikova [2,*,†] and Sergey Golubev [2]

[1] Department of Computer Science and Engineering, St. Petersburg Electrotechnical University "LETI", 197022 St. Petersburg, Russia; esnovikova@etu.ru
[2] Computer Security Problems Laboratory, St. Petersburg Federal Research Center of the Russian Academy of Sciences, 199178 Saint-Petersburg, Russia; ser9800@gmail.com
* Correspondence: doynikova@comsec.spb.ru
† These authors contributed equally to this work.

Abstract: One of the challenges in the Internet of Things systems is the security of the critical data, for example, data used for intrusion detection. The paper research construction of an intrusion detection system that ensures the confidentiality of critical data at a given level of intrusion detection accuracy. For this goal, federated learning is used to train an intrusion detection model. Federated learning is a computational model for distributed machine learning that allows different collaborating entities to train one global model without sharing data. This paper considers the case when entities have data that are different in attributes. Authors believe that it is a common situation for the critical systems constructed using Internet of Things (IoT) technology, when industrial objects are monitored by different sets of sensors. To evaluate the applicability of the federated learning for this case, the authors developed an approach and an architecture of the intrusion detection system for vertically partitioned data that consider the principles of federated learning and conducted the series of experiments. To model vertically partitioned data, the authors used the Secure Water Treatment (SWaT) data set that describes the functioning of the water treatment facility. The conducted experiments demonstrate that the accuracy of the intrusion detection model trained using federated learning is compared with the accuracy of the intrusion detection model trained using the centralized machine learning model. However, the computational efficiency of the learning and inference process is currently extremely low. It is explained by the application of homomorphic encryption for input data protection from different data owners or data sources. This defines the necessity to elaborate techniques for generating attributes that could model horizontally partitioned data even for the cases when the collaborating entities share datasets that differ in their attributes.

Keywords: intrusion detection; critical infrastructures; confidential data; federated learning; vertically partitioned data; gradient boosting decision trees; homomorphic encryption

Citation: Novikova, E.; Doynikova, E.; Golubev, S. Federated Learning for Intrusion Detection in the Critical Infrastructures: Vertically Partitioned Data Use Case. *Algorithms* **2022**, *15*, 104. https://doi.org/10.3390/a15040104

Academic Editor: Quan Qian

Received: 15 February 2022
Accepted: 19 March 2022
Published: 23 March 2022

1. Introduction

The Internet of Things Technology allows the construction of various intelligent information systems that can be used in critical infrastructures. Application of such systems can increase the efficiency of management of technology and business processes via solving the challenges related to automation of tasks of monitoring and managing the life cycle of equipment, energy consumption, and building optimal links between companies and service consumers [1]. This leads to the active implementation of services, applications, and information systems using this technology in different areas from smart cities and smart homes to smart grids and transport systems. However, the development of the Internet of Things is directly related to the emergence of new threats and information security risks. The landscape of such threats is extremely wide because of the variety of devices, and they can affect the security and privacy of citizens [2].

Considering [3], the risks related to the security of any Internet of Things system can be divided into the four following groups: the risks related to authentication of the devices and users; the risks related to physical impact on the system elements; the risks related to violation of confidentiality, integrity, and availability; and the risks related to the processing of personal data or other sensitive data. Security of personal data or other sensitive data in the Internet of Things is essential nowadays [4] as soon as its different components can collect and use various users' data that allows identifying them. These data incorporate biometric data, health data, etc. Moreover, according to the General Data Protection Regulation of the European Union, unique device identifiers, IP addresses, unique identifiers of mobile operators, and wireless access points are also personal data, since they help to establish the position of their owner [5]. These data are usually used in information security management systems. Thus, solving the tasks of information security management in the systems based on Internet of Things technology leads to an increase in the risks associated with the processing of personal data [6] and any unauthorized access to the information security management system threatens the confidentiality of information of IoT devices users, as well as other essential information [3].

Federated learning is a relatively novel approach to distributed machine learning. A key feature of this approach is that learning is performed locally, i.e., directly on devices that implement data collection. As the result, local models are trained and then aggregated to one global model. This feature of federated learning allows for constructing the systems that provide privacy of personal data and other sensitive data.

This paper researches the applicability of federated learning for intrusion detection systems in critical infrastructures to preserve the confidentiality of data describing technological processes and equipment settings. Analysis of such data can essentially increase the efficiency of detection of complex multi-step attacks affecting the both physical level and network level of the automated process control system [7,8]. The analysis of the research devoted to the application of the federated learning based approaches for designing intrusion detection systems showed that the researchers mainly focus on the case when all clients have datasets with a similar set of attributes. Such data distribution across clients in the federated learning settings is known as horizontally partitioned data. However, the collaborating entities could have datasets that describe the same objects but in different attributes. This case corresponds to the situation when the industrial facility consists of several collaborating entities that are responsible for different stages or parts of one technological process, and the monitoring and analysis of the state of such process require data from all entities simultaneously. Such data partition in federated learning settings is known as vertically partitioned data. To the best authors' knowledge, this research is the first one that addresses the challenge of designing intrusion detection systems based on the federated learning principles and applicable for the vertically partitioned data. Another challenge that is considered in the paper is the lack of datasets that could be used to model vertically partitioned data. The authors propose a possible solution to construct an appropriate dataset on the basis of the dataset that describes the functioning of the secure water treatment facility.

Thus, the novelty of the paper lies in the research of the applicability of federated learning for intrusion detection in the case of vertical data partition, i.e., when collaborating entities have information about one object but are presented by a different set of attributes.

The main contribution of the paper is as follows:

- the paper analyses existing intrusion detection systems based on the federated learning, and shows that the proposed solutions mainly focus on the specific type of data distribution across collaborating clients, namely horizontal partition when all clients share a similar set of attributes about different samples;
- the paper proposes an approach for the construction of intrusion detection system in case of vertically partitioned data and an architecture of the intrusion detection system based on federated learning;

- the paper evaluates the efficiency of the proposed system on the experiments in terms of accuracy, training and inference time parameters and outlines that, although existing federated learning frameworks allow constructing efficient intrusion detection systems in case of vertically partitioned data in terms of accuracy, these solutions are not practically applicable due to extremely large values of temporal parameters.

The experiments demonstrated that the accuracy of the intrusion detection model trained using federated learning is compared with the accuracy of the intrusion detection model trained using centralized machine learning. However, the computational efficiency of the learning and inference process is currently extremely low. It is explained by the application of homomorphic encryption for input data protection from different data owners or data sources. Thus, to make the application of federated learning in the case of vertically partitioned data practically useful, it is required to either reconsider the privacy preserving techniques used to secure input data or an elaborate approach that is able to transform vertically partitioned data into horizontally partitioned data.

The paper is structured as follows: Section 2 describes the distinctive features of federated learning. Section 3 considers existing solutions for intrusion detection based on federated learning. Section 4 provides an approach and an architecture of the intrusion detection system that allows for training intrusion detection models. Section 5 describes the results of the experiments. The paper ends with conclusions and future work prospects.

2. Federated Learning Paradigm

In 2016, McMahan et al. proposed a novel computational paradigm of distributed machine learning [9]. According to it, data are trained directly on the nodes where the data are stored. Then, the results of local training are transferred to the node that performs the aggregation of the parameters of local models and the calculation of the global analysis model. Thus, the global model considers the data belonging to different nodes, without transferring them over the network and collecting them in a single storage.

The key difference between federated learning and distributed machine learning is the usage of data sources and performing calculations directly on them, thus bringing computations as close to data as possible. On the contrary, in distributed machine learning, the nodes are used as sources of the computing resources to scale parallel processing of large amounts of data. Thus, the application of federated learning allows for decreasing the risk of unauthorized access to the data as soon as they are not transferred over the network. In addition, it allows for decreasing the volumes of the transferred traffic as soon as the training results are usually much lower in volume than the data itself.

The scheme of federated learning is represented in Figure 1. It includes three following components:

- the clients, i.e., data owners or data sources;
- the server that aggregates the local models and organises the communications between the clients;
- communication and computing infrastructure.

Two types of the federation are outlined depending on the computational resources of client nodes and their availability during the training process: a federation of organisations (cross-silo) and a federation of the devices (cross-device). For cross-silo federated learning settings, the small number of clients is typical. These clients are usually represented by organisations or/and data centers that are characterized by high computational resources, wide network bandwidth, and a high availability level during the training process. On the contrary, the cross-device federated learning settings are characterised by a large number of clients with limited computational, energy resources. These devices can drop out from the training process at any moment and appear again.

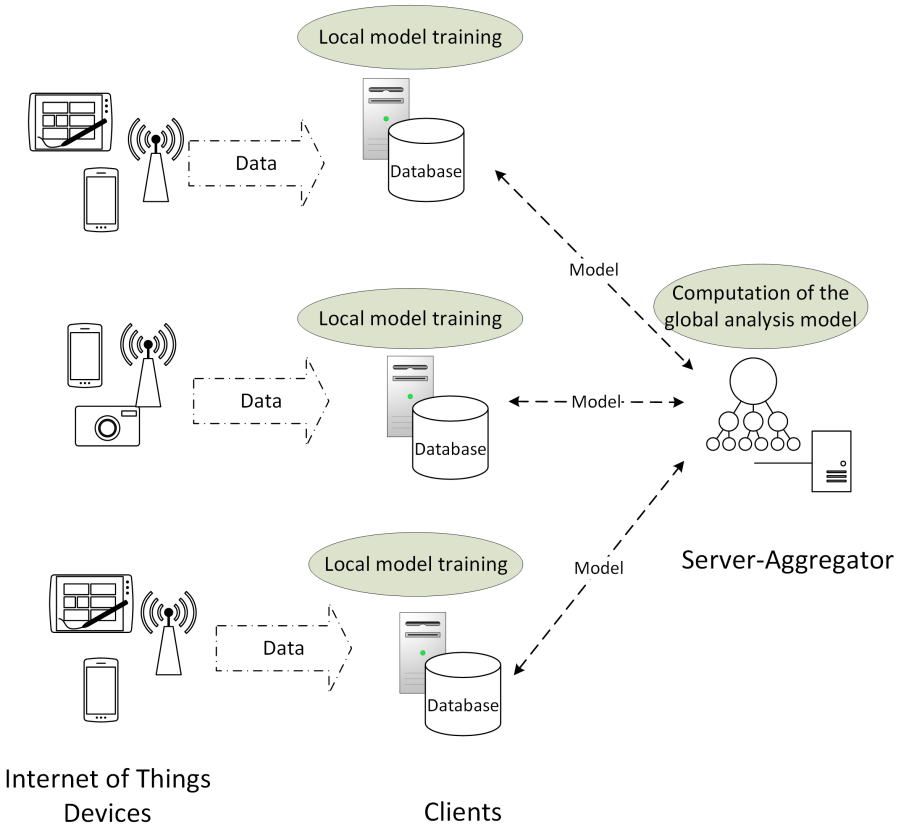


Figure 1. The scheme of federated learning.

Federated learning systems can implement two communication schemes between their components: centralised and decentralised [10,11]. The centralised scheme includes the main server used to organise different stages of the federated learning process and to coordinate the clients. This scheme is typical for the federation of the devices of the Internet of Things. In the decentralised scheme, the clients can coordinate their actions to obtain the global model, and the role of the aggregating server could be assigned to any client at any round of aggregation. Such a training scheme is also known as swarm learning [12]. It is more common for the federation of organisations where the clients have sufficiently high computing resources.

Another important feature of federated learning is the way data are distributed across the clients. In federated learning, this characteristic is referred to as data partition. There are two main types of data partition between the clients: horizontal and vertical ones. Let us consider them in detail. The data set usually contains information about some set of samples (or objects) that is presented by a set of attributes (or features). In the case of the horizontally partitioned data, the clients have datasets that describe different objects using the same set of attributes. In the case of the vertically partitioned data, the clients store the data that describe the same set of objects but in different attributes. Figure 2 shows the difference between horizontally partitioned data and vertically partitioned data. In real world cases, there is also a hybrid data partition when part of the data are partitioned partly horizontally and partly vertically.

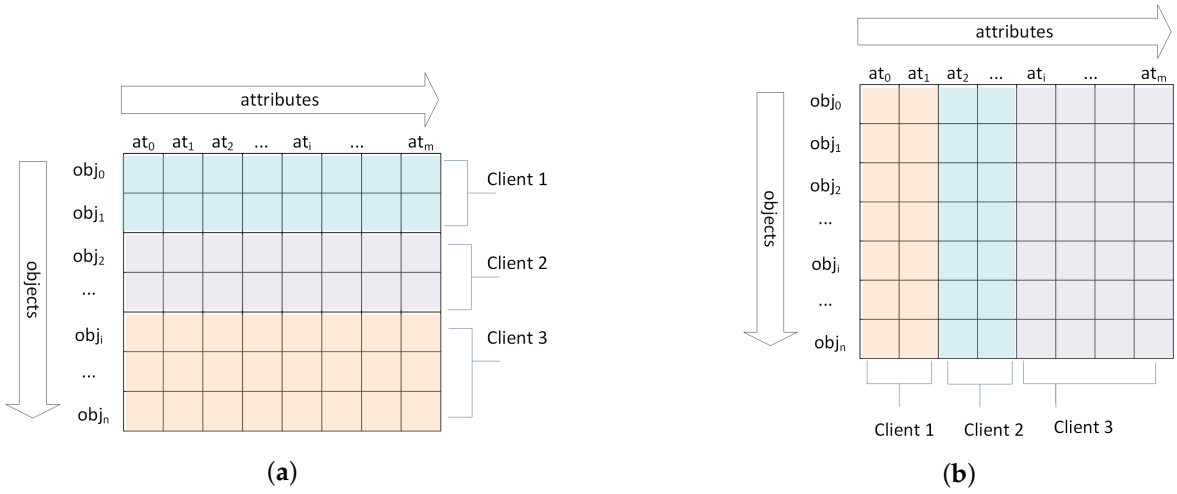


Figure 2. Types of data partition: horizontal (**a**), vertical (**b**).

3. Intrusion Detection Systems Based on the Federated Learning

Currently, machine learning methods are widely used in intrusion detection. By now, the researchers proposed various approaches based on different machine learning methods [13] including artificial neural networks such as adaptive resonance theory [14], self-organization maps [15,16], radial basis function [16], multi-layer perceptron [17]; Bayesian networks [18], genetic algorithms [19], support vector machine [20], k-nearest neighbors [21], decision trees [22], clustering [23], fuzzy logic [24], deep networks such as convolutional neural networks [25,26], recurrent neural networks [27], deep belief networks [28], deep auto-encoders [29], and Boltzmann Machine [30] et al.

Intrusion detection approaches can be classified by data sources, used machine learning algorithms, and used machine learning schemes. In terms of machine learning schemes, it is possible to outline two schemes, namely, centralized learning schemes when all data are sent to the central server that implements analysis model training, and federated learning schemes when machine learning models are trained locally on the devices, and only their parameters are sent to the central server. Nowadays, centralized machine learning is the most common approach. However, it leads to privacy and security risks. A federated learning scheme allows for avoiding these risks. In this paper, the authors focus on learning schemes and compare these schemes on the example of the GBDT training model in terms of the accuracy and training time parameters. The goal of this comparison is to evaluate if the advantages of the federated scheme are not eliminated by the losses in accuracy and training time.

One of the first research works proposing the idea of an intrusion detection system based on federated learning is the paper [31]. It describes the autonomous self-learning distributed system DÏoT for detection of the compromised devices of the Internet of Things. The system generates the device profiles based on their network behavior. Federated learning is used to train the models that detect anomalies in the device's behavior.

In [32], the authors also proposed the intrusion detection system for the devices of the Internet of Things. It is based on the federated learning of the convolutional neural network with the FedACNN memory mechanism. The conducted experiments show that this model can provide intrusion detection with high accuracy (up to 99.76%) with relatively good inference performance. The open dataset [33] was used as a training test dataset. However, the authors do not describe how the experimental environment with several clients has been simulated.

The first intrusion detection system for the industrial cyber-physical systems is provided in [34]. The authors conducted a series of experiments with different numbers of clients and showed that the proposed intrusion detection model outperforms many other modern approaches in machine learning quality metrics, and all analysis models developed

for different attack scenarios converge after a sufficient number of rounds of local model parameter aggregation.

Shingi et al. proposed to apply segmented federated learning (Segmented-FL) to construct a more efficient intrusion detection system [35]. A key difference of the proposed Segmented-FL is splitting clients or data owners into groups (segments). Each group of nodes operates with a specific global model for adaptive learning. This scheme is used both to exchange the parameters between the clients and to group the clients automatically. The automated grouping is used to increase the adaptability of the system to various parameters of the network infrastructure. In particular, it implements a periodic local assessment of the model used for clients segmentation: clients with similar network infrastructure parameters are combined into one group. Besides in the Segmented-FL system, the weighted aggregation function of parameters of local models is implemented. It considers the number of training set samples on each client. Such modifications of the federated learning allowed authors to construct an adaptive system of intrusion detection. The developed system is resistant to possible differences in network infrastructures. The open datasets CIDDS-001 and CIDDS-002 [36] were used as training test datasets.

In [37], the authors researched how differences in the data distribution by clients influence the intrusion detection accuracy. They created three different attack scenarios based on the CIC-ToN-IoT dataset [38]. The dataset was distributed among the clients to simulate different data distributions with strongly skewed distributions in classes. The conducted research demonstrated that data distribution essentially influences the accuracy of the intrusion detection models. To solve this challenge, the authors proposed the algorithm for selecting local sample instances based on Shannon's entropy estimate. The proposed algorithm allows for increasing the total accuracy and obtaining similar results compared to the scenario where datasets are balanced across clients.

Thus, the related research analysis demonstrates that the development of intrusion detection approaches based on federated learning is still in its initial state. The researchers mainly test the applicability of federated learning to train analysis models in distributed systems. For this goal, the known datasets are used. They are applied to simulate horizontal data distribution. The absence of datasets simulating the actual distribution of data between different clients leads to the fact that only some works evaluate the impact of unbalanced datasets on the quality of intrusion detection. Challenges of building intrusion detection systems for systems in which data are partitioned vertically, i.e., each system node stores only part of some object attributes, are not considered in the research papers. This paper addresses these challenges and proposes the approach for designing the intrusion detection system based on federated learning and applicable for the case of vertically partitioned data. The authors also propose a solution for modelling vertically partitioned data.

4. Approach for Designing of the Intrusion Detection System in the Case of Vertically Partitioned Data

In the case of vertically partitioned data, the clients store information on the same objects, but the set of data stored by each client is different in terms of attributes. The authors decided to use the model of the SWaT water treatment facility [39] as a prototype of a cyber-physical system with vertically partitioned data. The SWaT dataset was developed at the University of Singapore to simulate the attacks against cyber-physical systems. This testbed is a fully operational scaled-down water treatment plant. It models large modern water treatment systems that are used in the cities. Its main goal is to provide an opportunity for experimentally validated research on the development of secure cyber-physical systems. The SWaT facility models consist of the six basic processes corresponding to the physical and control components of a water treatment plant.

The prototype facility incorporates a multilevel communication network, programmable logic controllers, human–machine interfaces, supervisory control and data acquisition workstation (SCADA), and the repository. The data from sensors are available in the repository of the SCADA system. They are recorded by the server for further analysis. Thus, the system

is developed to implement the centralised data analysis. However, it can be converted to the system with vertically partitioned data. Each technological process of the facility could be considered as a separate element of the water treatment network. The parameters of all technological processes characterise the state of the system as a whole. Thus, the following architecture of the intrusion detection system for the cyber-physical system with vertically partitioned data can be proposed.

In Figure 3, each technological process is implemented by process facility PF*i* that is supplemented with a special device—an intelligent agent-hub. It registers the data generated by the sensors and control devices of the process's facility. This hub also collects network data of the facility. In Zone B, the additional server is located together with the central server of the SCADA system. The additional server is responsible for training the global model in the system and coordinating the whole federated learning process as well as the inference process. It is referred to as an aggregation server. This server is also responsible for monitoring the model performance and initiation of the re-training of the global model if it is required.

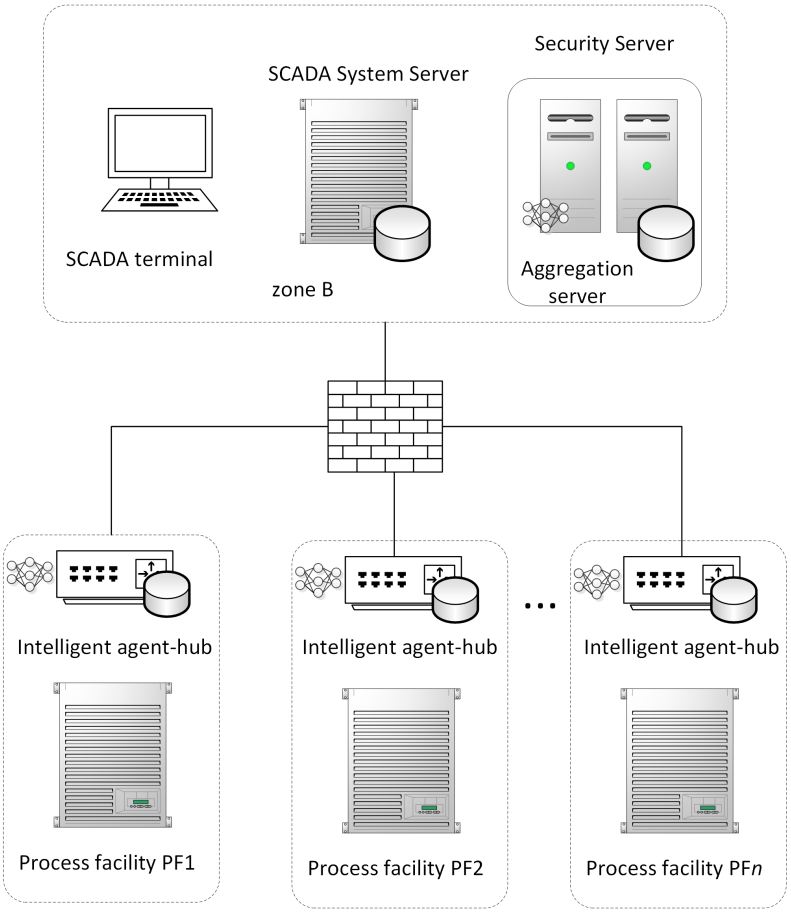


Figure 3. The intrusion detection system for the cyber-physical system with vertically partitioned data.

The training process looks as follows. The hub of each process facility collects both technological and network data. As each facility supports a different process, the collected technological data are different for each process facility, and the technological data are vertically partitioned. In the case of vertically partitioned data, the aggregation server initiates the training process. All agent–hubs actively participate in the training by calculating local models and then they send obtained results to the aggregation server. It constructs the global analysis model by selecting appropriate local models and sends updates to each intelligent agent–hub after each aggregation round.

When constructed in such a way, the intrusion detection system does not have direct access to data, only to the model parameters, and each process facility is isolated from others and does not share data with other participating facilities.

It should be noticed that the process of training and logical inference in the case of vertically partitioned data differs from the training process in the case of horizontally partitioned data. In the case of the horizontally partitioned data, as all clients have the same set of attributes, the global model is delivered to each client, and they could analyze new data samples independently from each other. In the case of the vertically partitioned data, when clients have different sets of attributes, the global model is distributed across the clients, and each client has a part of the model that relates to the data attributes the given client has. Thus, in order to infer about a new input sample, all clients need to cooperate and be available during the inference process. This fact means that, in order to preserve privacy of the input and training data, additional privacy preserving mechanisms are needed, and existing solutions of the federated learning for vertically partitioned data assume application of the privacy enhancing techniques based either on encryption [40,41] or differential privacy [42,43]. In the case of intrusion detection systems, these peculiarities of federated learning require that the hubs of all process facilities are to be available when making inference about a sample. To detect the anomalies, the agent–hub calculates the local results when a new sample is received and sends the results to the aggregation server to make the final decision.

5. The Experiments and Discussion

In [44], a detailed review of the frameworks with open source code is provided. It is demonstrated that currently two frameworks have a rather high level of technological readiness. They support both different analysis models and different communication schemes and data partitioning types. These frameworks are FATE and PaddleFL [45].

The PaddleFL framework supports training neural networks with dense layers only and linear regression models on vertically partitioned data, and the FATE framework supports training neural networks, gradient boosting decision trees (GBDT), and linear regression models. In both frameworks, all algorithms are secured using encryption-based techniques. For example, in PaddleFL, the multi-party computation protocol ABY^3 is used, and its application limits the number of the collaborating entities to the three ones, as it allows only three computational parties by the design [46]. To analyse vertically partitioned data, FATE implements two algorithms: HeteroNN and SecureBoost. The algorithm HeteroNN is a neural network training algorithm. It supports only two clients simultaneously because of the implementation features of the cryptographic transformations used to protect data belonging to different clients. The algorithm SecureBoost implements gradient boosting decision trees (GBDT), which is also supplemented by an additional privacy preserving mechanism to protect input data. In both cases, these mechanisms use Paillier's homomorphic encryption scheme and its two lightweight versions [47]. The SecureBoost algorithm does not have limitations on the number of clients. The results of the open-source frameworks evaluation in [44] showed that PaddleFL framework shows unstable behavior in the model training process, which could be explained by the fact that it uses a proprietary Paddle library as a back end for machine learning. The FATE framework uses PyTorch and TensorFlow libraries as the back end for machine learning. Thus, to deploy the test stand to simulate the intrusion detection system for water treatment system, the framework of federated learning FATE v. 1.5.0. [48] was selected in this research. The SecureBoost algorithm was selected as an aggregation strategy to produce the global model.

One of the serious challenges in evaluating the efficiency of federated learning is a lack of suitable data sets that model different types of data partition [10]. The most commonly used for the intrusion detection datasets such as CICIDS 2017 or Bot-IoT contain only network data either in packet-based or flow-based format [49]. To model horizontal data partition, it is possible to split the source data set into several subsets preserving or changing the distribution of attack classes if required. The authors assume that is one of

the reasons why existing intrusion detection approaches based on federated learning focus on horizontally partitioned data. Modelling real world vertical data partition is a more complicated task. Data sets that contain only network data could not be used to model vertically partitioned data because splitting them into subsets by the attributes violates their semantic meaning. As a result, the authors focused on the search of data sets with data from sensors; as such, data could be grouped based on the sensor type, location, etc. There are not so many publicly available such data sets, and the SWaT dataset is one of the most suitable ones for these purposes [39]. It contains data from physical sensors and network traffic with 36 attack scenarios that are implemented against different technological processes. The total number of entries with data from physical sensors is 946,722. Each entry contains 51 attributes.

To model the vertical data partition across process facilities, the following data preprocessing were implemented:

- grouping data by the technological processes based on the sensors' names because initial data do not contain information on the technological processes;
- data normalization;
- removing timestamps due to the type of the analysis model;
- data partitioning on the test and training samples followed by stratification i.e., preserving the class structure of labels in datasets.

Table 1 shows that the partition of the data across the nodes as well as a description of sensor types designated to different process facilities. The labels are stored on the aggregation server.

Table 1. The partition of SWaT dataset parameters among process facilities.

Client	No. of Parameters	List of Parameters	Description of Parameters
Process facility PF1	5	FIT101 LIT101 MV101 P101, P102	FIT—Flow Meter LIT—Level Transmitter MV—Motorized Valve P—Pomp AIT—Analyzer PIT—Pressure Meter DPIT—Differential Pressure Indicating Transmitter UV—Dechlorinator
Process facility PF2	11	AIT201, AIT202, AIT203, MV201 P201, P202, P203, P204, P205, P206	
Process facility PF3	9	DPIT301 FIT301 LIT301 MV301, MV302, MV303, MV304 P301, P302	
Process facility PF4	9	AIT401, AIT402, FIT401, LIT401 P401, P402, P403, P404 UV401	
Process facility PF5	13	AIT501, AIT502, AIT503, AIT504 FIT501, FIT502, FIT503, FIT504 P501, P20 PIT501, PIT502, PIT503	
Process facility PF6	4	FIT601, P601, P602, P603	
Aggregation server	1	label (normal/attack)	

Tables 2 and 3 provide the results of the conducted experiments. Table 2 shows the results of training the GBDT model in federated mode. During this series of experiments, the authors changed the settings of the SecureBoost algorithm by varying the number of constructed trees and types of encryption algorithms. When training in a federated mode, we also measured the time of the training and inference process. The authors also performed the experiments with a centralized machine learning model to compare the accuracy of the obtained models (Table 3). The settings of the GBDT model were similar; to train and test the model, the initial data set was split in a ratio of 0.7 to 0.3. The training was performed using the scikit-learn library [50]. The experiments demonstrated that the obtained accuracy is comparable with the results of GBDT in the case of centralized learning: as the number of trees increases, the accuracy increases. However, the training time is measured in hours, while training decision trees in centralized manner takes less than 1 min. It is also clearly seen that the training time of the model also depends linearly on the number of trees. The use of a fast implementation of the homomorphic encryption algorithm can significantly speed up the training process (up to three times). The gain in training time increases with an increase in the number of trees, but the accuracy of training decreases in this case.

Table 2. The parameters of the intrusion detection model training when training in federated mode.

No.	SecureBoost Encryption Type	Trees Number	Analysis Model Accuracy	Processing Time
1	Normal	3	91.9%	02:45:08
2	Fast	3	90.4%	01:45:37
3	Normal	15	97%	14:34:29
4	Fast	15	96%	04:30:48
5	Normal	30	99%	27:05:37
6	Fast	30	97%	09:52:40

Table 3. The parameters of the intrusion detection model training when training in a centralized manner.

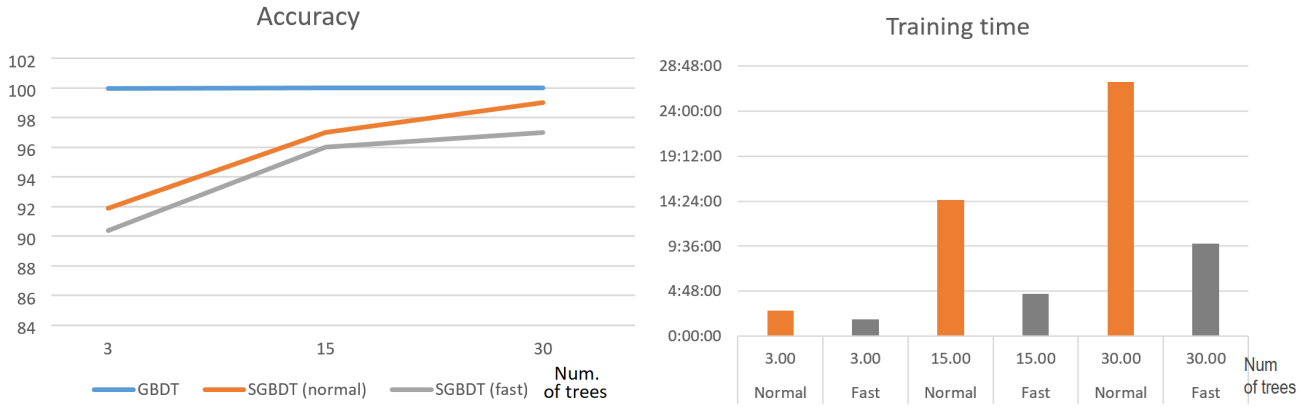
No.	Trees Number	Analysis Model Accuracy
1	3	99.98%
2	15	99.99%
3	30	99.99%

Figure 4 summarizes obtained results and shows the comparison of the accuracy of the gradient boosting decision tree constructed in a centralized and decentralized manner, and shows the duration of training process in the federated mode for different encryption settings. The comparison of temporal characteristics of training in FL mode and centralized one is omitted as training time is less than 1 min for 30 trees.

The inference time turned out to also be significant, reaching 40 min to make a decision on the input sample. Such a significant duration of training and inference is explained by the use of homomorphic encryption. To use homomorphic encryption, it is necessary to convert floating-point numbers to an integer representation. Machine learning algorithms require switching between arithmetic operations such as multiplication and addition and non-arithmetic operations such as approximate activation functions (such as the logistic function) and piecewise polynomial functions (such as RELU). The implementation of both types of transformations is computationally expensive [41].

The duration of the training process is not critical when building an intrusion detection model, since it can be implemented in the daemon mode. The achieved accuracy of the model is quite high, which allows for speaking about the applicability of federated learning for intrusion detection in scenarios when the transmission of initial data is critical. However, extremely long inference time, as well as the requirement of the availability of the agent–hubs make intrusion detection systems based on the current implementation of

federated learning frameworks practically inapplicable. The response time of the intrusion detection system is one of the critical parameters of the evaluating efficiency of such systems. Thus, it is the procedure of logical inference, which is the critical point when considering federated learning for intrusion detection in the case of the vertical data partition. Therefore, the authors can conclude that, to apply this technology in practice, it is necessary to study the algorithms used to infer an already trained model distributed among many clients. The authors assume that a possible solution may be the use of differential privacy techniques to protect the intermediate results of inference. These techniques do not use cryptographic transformations. The anonymization of the transmitted data is carried out by introducing random noise into them. Another promising solution is to develop techniques that allow transforming vertical data partitions into horizontal ones, by mapping different sets of features in one feature space.



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

Figure 4. The comparison of the accuracy of gradient boosting decision tree model trained in centralized and federated mode with different encryption parameters (**a**), and duration of the training process for different settings of the encryption protocol (**b**).

6. Conclusions

One of the open challenges related to the security of systems built based on the Internet of Things technology is to ensure the security of critical data, such as data required for intrusion detection. A possible solution to this challenge is to use the principles of federated learning while generating the analysis model. The key idea of federated learning is a training of the analysis model using training data partitioned among different clients without data transferring. Research papers provide several solutions based on federated learning for intrusion detection. However, most of the research considers the case when partitioned data are described using the same attributes.

In this paper, the case when the collaborating entities participating in the training process have data that are different in attributes is considered. The architecture of the intrusion detection system for vertically partitioned data that considers the principles of federated learning is introduced. It can be deployed in the cyber-physical system, for example, in the water treatment system.

To evaluate the applicability of federated learning for this case, the authors conducted a series of experiments. To overcome the problem relating to the lack of the appropriate data sets that could be used to model vertically partitioned data in the intrusion detection task, the authors chose the SWaT dataset as the training dataset. It describes the operation of the water treatment system for 11 days, and contains both network data and data from sensors. To model vertical data partition, the authors suggested splitting the data set based on the technological processes they describe. The conducted experiments showed that the intrusion detection model trained in the federated mode demonstrates high accuracy with a high privacy guarantee that is achieved by using encryption of the input parameters. However, the computational efficiency of the learning and inference process is currently

extremely low. According to the conducted experiments, the learning process can take hours and the inference process can take up to 40 min, while training decision trees in a centralized manner takes less than 1 min. It is explained by the use of homomorphic encryption to protect the transmitted model parameters. These indicators are the limiting factors to use this approach in practice for intrusion detection. To overcome this challenge, differential privacy techniques to protect the intermediate results of inference can be used or techniques that allow for transforming vertical data partition to a horizontal one, by mapping different sets of features in one feature space, should be developed.

The scope of the future works includes several tasks. The first one relates to the analysis of the datasets that could be used to model vertically partitioned data for the network security tasks as this is an essential problem when evaluating the efficiency of the approaches based on federated learning. This task is closely related to another direction of the future works, and the authors are planning to investigate the approaches that allow for mapping different sets of features in one feature space.

Author Contributions: Conceptualization, E.N.; methodology, E.N.; software, E.N. and S.G.; validation, E.N.; formal analysis, E.N. and E.D.; investigation, E.N. and E.D.; resources, E.N. and S.G.; data curation, E.N.; writing—original draft preparation, E.N.; writing—review and editing, E.N. and E.D.; supervision, E.N.; project administration, E.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research is being supported by the grant of RSF #22-21-00724 in SPC RAS.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

Abbreviations

The following abbreviations are used in this manuscript:

IoT	Internet of Things
SCADA	Supervisory Control and Data Acquisition
SWaT	Secure Water Treatment

References

1. Elrawy, M.; Awad, A.; Hamed, H. Intrusion detection systems for IoT-based smart environments: A survey. *J. Cloud Comput.* **2018**, *7*, 21. [CrossRef]
2. Baseline Security Recommendations for IoT. ENISA Report. Available online: https://www.enisa.europa.eu/publications/baseline-security-recommendations-for-iot (accessed on 15 February 2022).
3. Liu, X.; Zhao, M.; Li, S.; Zhang, F.; Trappe, W. A security framework for the internet of things in the future internet architecture. *Future Internet* **2017**, *9*, 27. [CrossRef]
4. Hassan, A.M.; Awad, A.I. Urban transition in the era of the internet of things: Social implications and privacy challenges. *IEEE Access* **2017**, *6*, 36428–36440. [CrossRef]
5. General Data Protection Regulation (GDPR). Available online: https://gdpr-info.eu/ (accessed on 15 February 2022).
6. Danda, J.M.R.; Hota, C. Attack Identification Framework for IoT Devices. In *Information Systems Design and Intelligent Applications. Advances in Intelligent Systems and Computing*; Satapathy, S., Mandal, J., Udgata, S., Bhateja, V., Eds.; Springer: New Delhi, India, 2016; pp. 32–58.
7. Ciholas, P.; Lennie, A.; Sadigova, P.; Such, J.M. The Security of Smart Buildings: A Systematic Literature Review. *arXiv* **2019**, arXiv:1901.05837v1. Available online: https://www.researchgate.net/publication/330466072_The_Security_of_Smart_Buildings_a_Systematic_Literature_Review (accessed on 15 February 2022).
8. Giechaskiel, I.; Zhang, Y.; Rasmussen, K.B. A Framework for Evaluating Security in the Presence of Signal Injection Attacks. In Proceedings of the European Symposium on Research in Computer Security, Luxembourg, 23–27 September 2019.
9. McMahan, H.B.; Moore, E.; Ramage, D.; Hampson, S.; Agüera y Arcas, B. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Proceedings of the AISTATS, Fort Lauderdale, FL, USA, 20–22 April 2017.
10. Kairouz, P.; McMahan, H.B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A.N.; d'Oliveira, R.G. Advances and open problems in federated learning. *arXiv* **2019**, arXiv:1912.04977.

11. Li, Q.; Wen, Z.; Wu, Z.; Hu, S.; Wang, N.; He, B. A Survey on Federated Learning Systems: Vision, Hype and Reality for Data Privacy and Protection. *arXiv* **2019**, arXiv:1907.09693.
12. Swarm Learning: Driving Advances Both Practical and Profound. Available online: https://www.hpe.com/us/en/insights/articles/swarm-learning-driving-advances-both-practical-and-profound-2111.html (accessed on 15 February 2022).
13. Branitskiy, A.; Kotenko, I. Analysis and Classification of Methods for Network Attack Detection. *SPIIRAS Proc.* **2016**, *2*, 207–244. [CrossRef]
14. Bukhanov, G.; Polyakov, V.M. Detection of network attacks based on adaptive resonance theory. *J. Phys. Conf. Ser.* **2018**, *1015*, 042007. [CrossRef]
15. Hoglund, A.J.; Hatonen, K.; Sorvari, A.S. A Computer Host-Based User Anomaly Detection System Using The Self-Organizing Map. In Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks, Como, Italy, 24–27 July 2000; Volume 5, pp. 411–416.
16. Horeis, T. Intrusion Detection with Neural Networks—Combination of Self-Organizing Maps and Radial Basis Function Networks for Human Expert Integration. Available online: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.106.191&rep=rep1&type=pdf (accessed on 5 March 2022).
17. Dilipkumar, S.; Durairaj, M. Detection of Attacks Using Multilayer Perceptron Algorithm. In *Inventive Communication and Computational Technologies. Lecture Notes in Networks and Systems*; Ranganathan, G., Fernando, X., Shi, F., Eds.; Springer: Singapore, 2022; Volume 311. [CrossRef]
18. Heckerman, D. A Tutorial on Learning with Bayesian Networks. *Innov. Bayesian Netw. Theory Appl.* **2008**, *156*, 33–82.
19. Dave, M.H.; Sharma, S.D. Improved Algorithm for Intrusion Detection Using Genetic Algorithm and SNORT. *Int. J. Emerg. Technol. Adv. Eng.* **2014**, *4*, 273–276.
20. Chen, W.H.; Hsu, S.H.; Shen, H.P. Application of SVM and ANN for intrusion detection. *Comput. Oper. Res.* **2005**, *32*, 2617–2634. [CrossRef]
21. Liao, Y.; Vemuri, V.R. Use of k-nearest neighbor classifier for intrusion detection. *Comput. Secur.* **2002**, *21*, 439–448. [CrossRef]
22. Kruegel, C.; Toth, T. Using Decision Trees to Improve Signature-Based Intrusion Detection. In Proceedings of the International Workshop on Recent Advances in Intrusion Detection, Pittsburgh, PA, USA, 8–10 September 2003; pp. 173–191.
23. Ranjan, R.; Sahoo, G. A new clustering approach for anomaly intrusion detection. *Int. J. Data Min. Knowl. Manag. Process (IJDKP)* **2014**, *4*, 29–38. [CrossRef]
24. Ireland, E. Intrusion Detection with Genetic Algorithms and Fuzzy Logic. In Proceedings of the UMM CSci Senior Seminar Conference, Morris, MN, USA, 7 December 2013; pp. 1–6.
25. Hu, J.; Liu, C.; Cui, Y. An Improved CNN Approach for Network Intrusion Detection System. *Int. J. Netw. Secur.* **2021**, *23*, 569–575. [CrossRef]
26. Li, Z.; Qin, Z.; Huang, K.; Yang, X.; Ye, S. Intrusion Detection Using Convolutional Neural Networks for Representation Learning. In *Neural Information Processing. ICONIP 2017. Lecture Notes in Computer Science*; Liu, D., Xie, S., Li, Y., Zhao, D., El-Alfy, E.S., Eds.; Springer: Cham, Switzerland, 2017; Volume 10638. [CrossRef]
27. Vinayakumar, R.; Soman, K.P.; Poornachandran, P. Evaluation of Recurrent Neural Network and its Variants for Intrusion Detection System (IDS). *Int. J. Inf. Syst. Model. Des.* **2017**, *8*, 43–63. [CrossRef]
28. Alom, M.Z.; Bontupalli, V.; Taha, T.M. Intrusion detection using deep belief networks. In Proceedings of the National Aerospace and Electronics Conference (NAECON), Dayton, OH, USA, 15–19 June 2015; pp. 339–344. [CrossRef]
29. Song, Y.; Hyun, S.; Cheong, Y.-G. Analysis of Autoencoders for Network Intrusion Detection. *Sensors* **2021**, *21*, 4294. [CrossRef]
30. MohanaPriya, P.; Shalinie, S.M. Restricted Boltzmann Machine based detection system for DDoS attack in Software Defined Networks. In Proceedings of the Fourth International Conference on Signal Processing, Communication and Networking (ICSCN), Chennai, India, 16–18 March 2017; pp. 1–6. [CrossRef]
31. Nguyen, T.D.; Marchal, S.; Miettinen, M.; Fereidooni, H.; Asokan, N.; Sadeghi, A. DÏoT: A Federated Self-learning Anomaly Detection System for IoT. In Proceedings of the IEEE 39th International Conference on Distributed Computing Systems (ICDCS), Dallas, TX, USA, 7–9 July 2019; pp. 756–767.
32. Zhao, R.; Yin, Y.; Shi, Y.; Xue, Z. Intelligent intrusion detection based on federated learning aided long short-term memory. *Phys. Commun.* **2020**, *42*, 101157. [CrossRef]
33. NSL-KDD Dataset. Available online: https://www.unb.ca/cic/datasets/nsl.html (accessed on 15 February 2022).
34. Li, B.; Wu, Y.; Song, J.; Lu, R.; Li, T.; Zhao, L. DeepFed: Federated Deep Learning for Intrusion Detection in Industrial Cyber–Physical Systems. *IEEE Trans. Ind. Inform.* **2021**, *17*, 5615–5624. [CrossRef]
35. Shingi, G.; Saglani, H.; Jain, P. Segmented Federated Learning for Adaptive Intrusion Detection System. *arXiv* **2021**, arXiv:2107.00881.
36. CIDDS Dataset. Available online: https://www.hs-coburg.de/forschung/forschungsprojekte-oeffentlich/informationstechnologie/cidds-coburg-intrusion-detection-data-sets.html (accessed on 15 February 2022).
37. Campos, E.M.; Saura, P.F.; González-Vidal, A.; Ramos, J.L.; Bernabé, J.B.; Baldini, G.; Gómez-Skarmeta, A.F. Evaluating Federated Learning for Intrusion Detection in Internet of Things: Review and Challenges. *arXiv* **2021**, arXiv:2108.00974.
38. CIC-ToN-IoT Dataset. Available online: https://staff.itee.uq.edu.au/marius/NIDS_datasets/#RA13 (accessed on 15 February 2022).

39. Secure Water Treatment (SWaT). Available online: https://itrust.sutd.edu.sg/itrust-labs_datasets/dataset_info/ (accessed on 15 February 2022).
40. Rachuri, R.; Suresh, A. Trident: Efficient 4pc framework for privacy preserving machine learning. *arXiv* **2019**, arXiv:1912.02631.
41. Zhang, C.; Li, S.; Xia, J.; Wang, W.; Yan, F.; Liu, Y. BatchCrypt: Efficient Homomorphic Encryption for Cross-Silo Federated Learning. In Proceedings of the 2020 USENIX Annual Technical Conference, Boston, MA, USA, 15–17 July 2020.
42. Tang, P.; Cheng, X.; Su, S.; Chen, R.; Shao, H. Differentially private publication of vertically partitioned data. *IEEE Trans. Dependable Secur. Comput.* **2021**, *18*, 780–795. [CrossRef]
43. Xu, D.; Yuan, S.; Wu, X. Achieving differential privacy in vertically partitioned multiparty learning. *arXiv* **2019**, arXiv:1911.04587.
44. Kholod, I.; Yanaki, E.; Fomichev, D.; Shalugin, E.; Novikova, E.; Filippov, E.; Nordlund, M. Open-Source Federated Learning Frameworks for IoT: A Comparative Review and Analysis. *Sensors* **2021**, *21*, 167. [CrossRef]
45. Baidu PaddlePaddle Releases 21 New Capabilities to Accelerate Industry-Grade Model Development. Available online: http://research.baidu.com/Blog/index-view?id=126 (accessed on 15 February 2022).
46. aby3 Library. Available online: https://github.com/ladnir/aby3 (accessed on 15 February 2022).
47. Cheng, K.; Fan, T.; Jin, Y.; Liu, Y.; Chen, T.; Papadopoulos, D.; Yang, Q. SecureBoost: A Lossless Federated Learning Framework. *IEEE Intell. Syst.* **2021**, *36*, 87–98. [CrossRef]
48. An Industrial Grade Federated Learning Framework. Available online: https://fate.fedai.org/ (accessed on 15 February 2022).
49. Ring, M.; Wunderlich, S.; Scheuring, D.; Landes, D.; Hotho, A. A survey of network-based intrusion detection data sets. *Sens. Comput. Secur.* **2018**, *86*, 147–167. [CrossRef]
50. Scikit-Learn Library. Available online: https://scikit-learn.org/stable/ (accessed on 15 February 2022).