big data and cognitive computing

MDPI

Article

Anomaly Detection of IoT Cyberattacks in Smart Cities Using Federated Learning and Split Learning

Ishaani Priyadarshini

School of Information, University of California, Berkeley 94720, CA, USA; ishaani@berkeley.edu

Abstract: The swift proliferation of the Internet of Things (IoT) devices in smart city infrastructures has created an urgent demand for robust cybersecurity measures. These devices are susceptible to various cyberattacks that can jeopardize the security and functionality of urban systems. This research presents an innovative approach to identifying anomalies caused by IoT cyberattacks in smart cities. The proposed method harnesses federated and split learning and addresses the dual challenge of enhancing IoT network security while preserving data privacy. This study conducts extensive experiments using authentic datasets from smart cities. To compare the performance of classical machine learning algorithms and deep learning models for detecting anomalies, model effectiveness is assessed using precision, recall, F-1 score, accuracy, and training/deployment time. The findings demonstrate that federated learning and split learning have the potential to balance data privacy concerns with competitive performance, providing robust solutions for detecting IoT cyberattacks. This study contributes to the ongoing discussion about securing IoT deployments in urban settings. It lays the groundwork for scalable and privacy-conscious cybersecurity strategies. The results underscore the vital role of these techniques in fortifying smart cities and promoting the development of adaptable and resilient cybersecurity measures in the IoT era.

Keywords: IoT; cyberattacks; smart city; split learning; federated learning

Citation: Priyadarshini, I. Anomaly Detection of IoT Cyberattacks in Smart Cities Using Federated Learning and Split Learning. *Big Data Cogn. Comput.* **2024**, *8*, 21. https://doi.org/10.3390/bdcc8030021

Academic Editors: Guarino Alfonso, Rocco Zaccagnino, Emiliano Del Gobbo and Giuseppe Maria Luigi Sarnè

Received: 7 December 2023
Revised: 10 February 2024
Accepted: 21 February 2024
Published: 22 February 2024

1. Introduction

In an era of digital transformation and urbanization, the emergence of smart cities represents a paradigm shift in how we design, build, and experience urban environments. Smart cities are a response to the ever-growing urban population and the challenges it presents, such as congestion, pollution, and resource scarcity. These cities harness the power of advanced technologies, data analytics, and interconnected systems to create efficient, sustainable, and livable urban spaces. At the heart of a smart city lies the Internet of Things (IoT), an ecosystem of interconnected devices and sensors that collect and exchange data. These devices, embedded in infrastructure, transportation systems, and everyday objects, enable cities to monitor and manage various functions in real time. From optimizing traffic flow and conserving energy to enhancing public safety and providing citizens with personalized services, the potential of smart cities is boundless. However, the transition to smart cities has challenges, and cybersecurity is one of the most pressing concerns [1]. As cities increasingly rely on interconnected IoT devices, they become susceptible to new cyber threats. Cyberattacks targeting smart cities can disrupt critical infrastructure, compromise public safety, and undermine the advantages that make these cities smart. Ensuring the cybersecurity of smart cities is, therefore, a matter of paramount importance. The cyber risks associated with smart cities are multifaceted. These risks encompass a spectrum of threats, from data breaches and unauthorized access to manipulating critical infrastructure [2]. A smart city's security breach can have far-reaching consequences, impacting public services, citizen trust, and economic stability. Common cyber threats in smart cities include data breaches, denial of service (DoS) attacks, ransomware, phishing, and manipulation of IoT devices [3]. Given the magnitude of these threats, research

and development efforts have been directed toward enhancing the cybersecurity posture of smart cities. Existing solutions include intrusion detection systems, firewalls, and encryption protocols. However, these traditional approaches often fail to address smart cities' unique challenges.

The motivation for this research lies in the critical need to bolster the cybersecurity defenses of smart cities. While progress has been made in securing individual components and systems, a holistic and privacy-preserving approach is essential. Moreover, decentralized, real-time threat detection is paramount to safeguarding smart city environments effectively. This paper introduces an approach to IoT cyberattack detection in smart cities, utilizing federated and split-learning methodologies. These techniques effectively address the challenge of enhancing IoT network security while maintaining data privacy [4]. Federated learning facilitates collaborative model training across decentralized devices, enabling anomaly detection without the need to share raw data among smart cities. Split learning, an extension of federated learning, further enhances privacy preservation and model efficiency. The research highlights several key elements and contributions:

a. Application of federated learning to smart cities enables decentralized anomaly detection while safeguarding data privacy.
b. Extension of privacy preservation capabilities with split learning enhances both privacy and efficiency.
c. Comprehensive evaluation using real-world smart city datasets provides valuable insights into effectiveness.
d. Performance comparison with existing algorithms offers a thorough analysis of efficiency.
e. Utilizing a suite of performance metrics, including precision, recall, F-1 score, accuracy, and training/deployment time, facilitates a holistic assessment of the approach.

By harnessing the capabilities of federated learning and split learning, the aim is to provide a robust solution to the cybersecurity challenges that smart cities face. The work represents a significant step toward ensuring the integrity and security of future smart cities. The rest of the paper is organized as follows: Section 2 incorporates materials and methodology highlighting recent research works and the proposed methods for detecting anomalies in the smart city environment. This section incorporates the data flow, federated learning architecture, and split-learning architecture. Section 3 presents results and observations based on the datasets and performance metrics for evaluation and results. This is followed by a comparative analysis depicting the analysis results against similar research works. Section 4 concludes the study and highlights future works.

2. Materials and Methods

This section comprises Related Work and Methodology. The literature survey is discussed in the Related Work section, and the limitations of some of the recent work on anomaly detection in smart cities are stated. The Methodology section proposes the infrastructure for anomaly detection of cyberattacks in smart cities. This section also details the federal learning architecture and split-learning architecture applied to the study.

2.1. Related Works

Ajao and Apeh, 2023 [5] introduce a Petri net-genetic algorithm-based reinforcement learning (GARL) technique to address security issues in smart cities' Industrial Internet of Things (IIoT) networks. The framework includes a trust model and distributed authorization for information control, achieving high anomaly detection rates. The model has been implemented on a secure framework with trust, privacy, and authentication. The results look satisfactory; however, the main challenges of the study may be scalability, implementation, and robustness assessment against adversarial attacks. Rashid et al., 2023 [6] depict the vulnerability of IoT devices by addressing cybersecurity concerns. The study presents a machine learning-based approach, using various algorithms and ensemble methods like support vector machines, decision trees, random forests, bagging, boosting, and stacking to detect attacks and anomalies. Unique to this work is the integration of feature selection,

cross-validation, and multi-class classification. Experimentation with recent attack datasets shows that the stacking ensemble model outperforms others, offering promise for robust cybersecurity in smart cities. The study relies on traditional machine learning algorithms. It does not investigate advanced machine learning models or delve into deep learning approaches, potentially overlooking intricate and sophisticated cybersecurity threats within smart city contexts. Mukherjee, 2023 [7] present a study on deploying deep learning for detecting cyberattacks in smart grids, specifically false data injection attacks (FDIAs). The study is based on a novel deep learning model that detects FDIAs and accurately locates intrusions in real time. The architecture incorporates conventional bad data detectors, offering a cost-effective solution. In addition to a multilabel classification strategy, the architecture can also capture attack co-occurrence dependencies within raw measurements, operating without prior statistical knowledge of the grid. The study was conducted using standard test bench evaluations. However, scalability and resource requirements in smart city infrastructure could be a concern. Almuqren et al., 2023 [8] suggest a novel white shark equilibrium optimizer and hybrid deep learning methods for detecting cyberattacks in smart cities. The study aims at optimizing electricity management to improve the quality of life and resource efficiency. The study deploys a White Shark Equilibrium Optimizer with a Hybrid Deep-Learning-based Cybersecurity Solution (WSEO-HDLCS) to address the issue of disrupting essential services due to DDoS attacks. The architecture incorporates feature selection and a stacked deep autoencoder (SDAE) model, further optimized through the gravitational search algorithm (GSA). While the results seem satisfactory and the study acknowledges practical implementations and scalability concerns, the primary focus of the study is limited to DDoS attacks. However, a smart city is vulnerable to several advanced cyberattacks. Alsaade and Al-Adhaile, 2023 [9] suggest deep autoencoder algorithms for detecting cyberattacks in self-driving vehicles. The study analyzes the behavior of electronic control units (ECUs) within connected and autonomous vehicles (CAVs), interconnected through in-vehicle networks (IVNs), for facilitating data exchange and optimizing vehicle operation. The analysis incorporates machine learning and deep learning methods for identifying cyberattacks that detect erroneous data on vehicle data buses. Some algorithms deployed for the study are gradient boosting, k-nearest neighbor (KNN), decision trees, and long short-term memory (LSTM). The study uses the car-hacking and UNSE-NB15 datasets, and the performance is evaluated using statistical parameters. While the accuracy is shown to be the best for decision trees and autoencoders, the study may face challenges concerning scalability training time. Ding et al., 2023 [10] present a deep learning model incorporating three key components, i.e., the residual-based spatial representation (RSR) block, the temporal representation block (TRB), and the detection block (DB) for detecting cyberattacks. The model performance is 90.57%, 94.96%, and 98.41% in terms of accuracy concerning the TON-IoT, Edge-IIoTset, and UNSW-NB15 datasets, respectively. The complex architecture addresses issues like vanishing gradient and deploys techniques like feature extraction. The model shows generalization, although the study considers specific IoT environments and cyberattacks. Sharma and Babbar, 2023 [11] highlight the rising cybersecurity concerns in the IoT environment, specifically concerning smart transportation. The study is based on deploying a BoT-IoT dataset comprising several attack categories and subcategories. The dataset is deployed to analyze the system's dependability for training and evaluation purposes. The study trains and evaluates different ML techniques, such as random forest (RF), naive Bayes (NB), and decision tree (DT). It is observed that RF and DT achieved the highest accuracy of 91% on the BoT-IoT dataset. The limitation of the study is that the analysis is based on an existing dataset, which may not cover all IoT security scenarios or provide a realistic assessment. Ajao and Apeh, 2023 [12] suggest an integrated architecture comprising of blockchain and machine learning for securing the fog computing layer vulnerability in smart city infrastructure. The study also discusses combining smart city-based Industrial Internet of Things (IIoT) with IPv6 addressing and 5G networks to enhance the Quality of Experience (QoE). However, it is prone to risks concerning IPv4 wireless sensor networks. The blockchain ensures privacy and confidentiality for packet

traffic to the public, while machine learning is deployed for intrusion detection between the edge and fog layers. The proposed architecture exhibits high performance and a low processing time but relies on specific network architecture, which may not be generalizable for a smart city's diverse infrastructure. Alrayes et al., 2023 [13] present a novel deep learning architecture with chaotic poor and rich optimization for intrusion detection in smart city environments. The proposed model aims to achieve improved execution, sustainability, and security and encompasses data preprocessing, feature selection using a chaotic optimization algorithm (CPROA-FS), and intrusion detection employing the butterfly optimization algorithm (BOA) and deep sparse autoencoder (DSAE). Simulation analysis on the CICIDS dataset reveals superior performance, with IDCPRO-DLM achieving a maximum accuracy of 98.53%. While the results seem satisfactory, deploying the proposed architecture on a specific dataset may not exhibit model generalizability. Taleb and Saqib, 2023 [14] propose a hybrid deep learning model based on a convolutional neural network (CNN) and quasi-recurrent neural network (QRNN) for identifying cyber threats in smart city environments. The study was carried out on two independent datasets: BoT-IoT and TON_IoT. While the proposed model exhibited better performance than benchmark models, the parameter considered in this case was only accuracy. Due to the involvement of neural networks, training time could be another challenge for this proposed study. Bilakanti et al. [15] performed a study to analyze anomalies and faults in sensors deployed in the IoT landscape due to tampering using machine learning techniques. The study deployed methods like isolation forest and local outlier factors to compare against supervised techniques like naive Bayes, support vector Machines, and extreme gradient boosting for the analysis. Lin et al. [16] propose a self-adaptive thresholding method due to the high false anomaly ratio. A deep learning-based hierarchical context representation learning has been proposed to transform time series patterns into images such that they can be used for gathering spatial features. The study was conducted on multiple datasets, and the proposed method outperformed the baseline models used in the study. Mitropoulou et al. [17] highlight using knowledge graph embedding to detect anomalies in the cloud computing environment. This method was deployed using machine learning algorithms like isolation forest and cluster-based local outlier factor (CBLOF). The study focuses on optimizing the performance in a simulated environment. Jithish et al. [18] suggest a federated learning technique for detecting distributed anomalies in smart grids due to the challenges involved in server-based model training. The study highlights that the models are trained locally in smart meters such that data are not shared with the central server, and regular parameter updates maintain the model's privacy. The study was conducted on multiple datasets and highlights the feasibility of deploying federated learning techniques in such environments. Dang et al. [19] suggest a split-and-conquer approach for detecting anomalies in sensory data. The technique analyzes the spatial and temporal correlation between sensors to detect anomalies and creates trend-based profiles for detection. The proposed technique obtains 8% higher accuracy and a 5% lower false-positive rate than existing methods. Nassif et al. [20] reviewed 290 research articles on machine learning techniques applied for anomaly detection. The overall study analyzes 29 distinct algorithms and 22 datasets that exhibit experimental analysis on the same. Takiguchi and Shiono [21] deployed a split-training method for anomaly detection in gas turbine engines. A clustering algorithm was deployed to extract input data and generate a simulation model. A regression model classifies the data points and enhances the model performance by optimizing the overall process. Nixon et al. [22] propose a split active learning algorithm combined with unsupervised methods for anomaly detection. The study used autoencoders with active learning to reduce labeling costs. The proposed method reduces training time and improves performance by 20%, outperforming traditional learning methods. Dragoi et al. [23] propose a protocol for splitting data in independently and identically distributed (IID) testing splits. The performance was analyzed using diverse algorithms and validated using the receiver operating characteristic (ROC) curve. Zhang et al. [24] deploy a combination of semi-supervised learning and adaptive multiclass balancing for network anomaly detection. The study used a multiclass split

balancing technique and adaptive confidence threshold function to handle imbalance in the data. The proposed technique enhances anomaly detection performance and outperforms other baseline models.

2.2. Methodology

This section discusses the smart city infrastructure for detecting cyberattacks followed by federated learning and split learning

2.2.1. Smart City Infrastructure for Anomaly Detection

Smart city infrastructure is prone to attacks or anomalies due to the heavy volume of data, diverse data sources, real-time monitoring, and seamless data integration. IoT devices and sensors dispersed throughout the city continually observe various parameters, supplying detailed data for analysis. Predictive analytics can be used for analyzing and managing such anomalies. Moreover, robust network security and resource optimization can improve safety, security, and operational efficiency, making smart cities more adaptable and responsive to evolving challenges. Figure 1 depicts the smart city infrastructure for anomaly detection. The work is highlighted as follows:

- IoT devices at the edge layer of a smart city IoT network incorporate various devices, including sensors, cameras, and connected vehicles. These devices generate vast data, capturing traffic, air quality, energy usage, and more information. While enhancing city operations, they are also susceptible to cyber threats.
- Fog Layer (the cybersecurity hub) bridges IoT devices and cloud services. It consists of a distributed computing resource network, including servers and gateways. Positioned strategically within the city, its primary role is data processing and analysis and ensuring security. Within the fog layer, machine learning plays a pivotal role in securing the smart city's IoT infrastructure. Data generated by IoT devices are continuously collected, cleaned, normalized, and structured for analysis. Once data are prepared, relevant features are captured for meaningful information and pattern analysis. Machine learning algorithms are then fed these data to train the models and to understand device behaviors and network patterns. Potential threats or anomalies, such as distributed denial of service (DDoS) attacks or unusual data traffic, are caused by deviations in network traffic and device behavior. These anomalies are detected in real time, and as these are detected, the system can raise alerts and take necessary actions to mitigate the threat. As many models are adaptive, they can adapt to new data patterns.
- Reducing latency and bandwidth usage. Machine learning in the fog layer minimizes latency and conserves bandwidth. By processing and filtering data locally, only important information is sent to the cloud, which enhances security and optimizes resource utilization in the smart city infrastructure. Hence, the fog layer, powered by machine learning for real-time anomaly detection, is an integral component of a smart city's cybersecurity hub. The fog layer protects the interconnected systems configuring smart cities by continuously monitoring and mitigating anomalies.

2.2.2. Federated Learning

Federated learning operates under the assumption that data residing on various edge devices, such as IoT sensors, cameras, and smart infrastructure within a smart city, need not be centralized for machine learning purposes. Instead, machine learning models are distributed to these edge devices, where model training occurs locally on the device without transferring sensitive data to a central server. This decentralized approach offers several key advantages:

- In smart cities, data privacy is paramount. Federated learning ensures that data remain on the edge devices, eliminating the need to transfer raw data over the network and minimizing privacy risks.

- Federated learning allows real-time model updates at the edge, minimizing latency. This is essential for timely cyberattack detection and response, especially in traffic management or public safety scenarios.
- By conducting frequent model updates, federated learning significantly reduces network bandwidth usage. This is crucial in smart cities where data transmission can strain network resources.
- The local models on edge devices learn from local data and collaborate to improve a global model. This collective learning enhances the model's accuracy and adaptability.

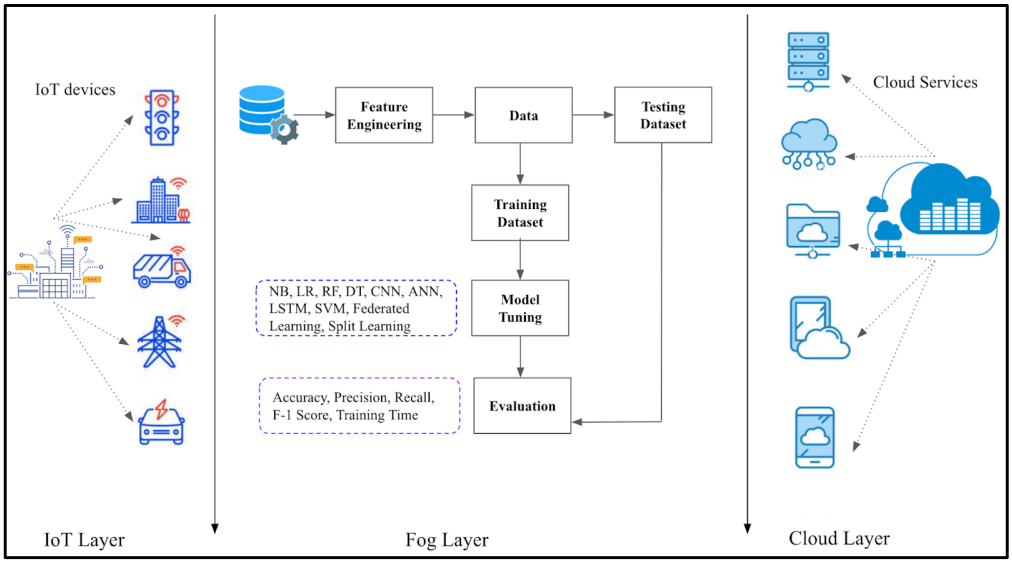

Figure 1. Smart city infrastructure for anomaly detection.

Smart city infrastructure incorporates diverse IoT devices and interconnected systems. This makes it vulnerable to various cyberattacks. Figure 2 illustrates the federated learning process in a smart city environment, where edge devices contribute to model training without centralizing sensitive data, ensuring privacy and security. Federated learning can be deployed to detect cyberattacks by performing the following steps:

Step 1: A central authority or an agency is responsible for distributing a machine learning model specifically for cyberattack detection to edge devices throughout the smart city.

Step 2: On each edge device, the model conducts training using locally generated data, such as network traffic patterns, device behavior, and security logs. The model learns to identify anomalies and potential cyber threats within its specific domain.

Step 3: Periodically, the local models communicate with the agency or each other to share model updates without sharing raw data. Collaborative model updates aggregate knowledge gained from diverse edge devices, enhancing the overall model's accuracy and threat detection capabilities.

Step 4: The updated global model is deployed on edge devices to perform real-time threat detection. As new data are generated across the smart city, the model identifies anomalies, suspicious patterns, or known attack signatures from data streams, such as traffic cameras, environmental sensors, and network traffic.

Step 5: Upon detecting potential cyber threats, smart city systems can initiate rapid responses, such as isolating compromised devices, alerting security personnel, or reconfiguring network traffic to mitigate attacks. In the diagram, data generated by IoT devices flow through a federated learning process. Models are distributed to edge devices, which conduct local training.

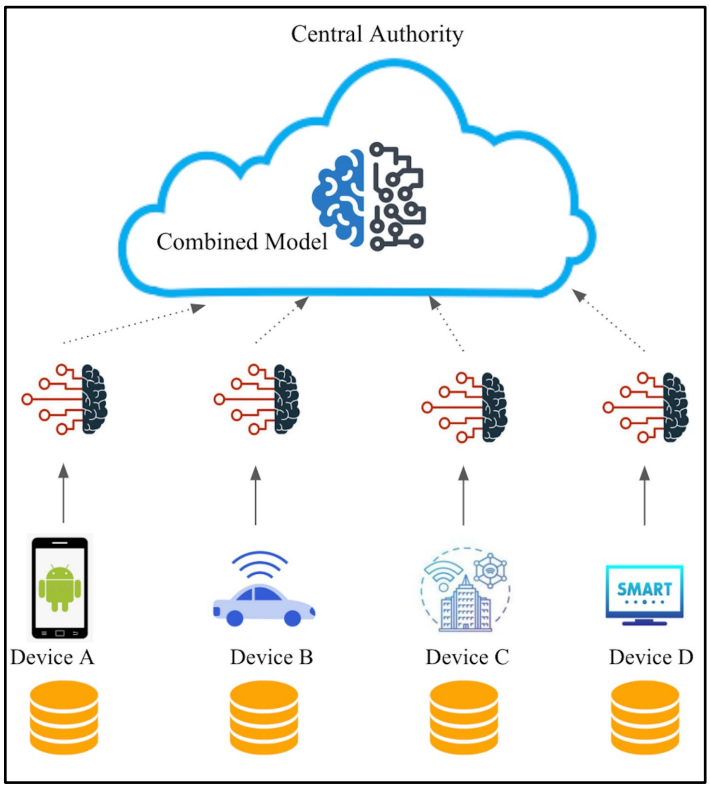

Figure 2. Federated learning architecture.

2.2.3. Split Learning

Split learning represents a decentralized machine learning approach aimed at tackling data privacy and security issues while harnessing the combined capabilities of edge devices. It fundamentally transforms the conventional centralized machine learning model by dispersing the training process across multiple entities, commonly known as "parties". This distribution facilitates model training without disclosing raw data, safeguarding individual data privacy. While split and federated learning are decentralized machine learning techniques designed to address data privacy concerns, they differ in their approaches and applications. In federated learning, data remain decentralized, but the model is sent to edge devices. Each device performs local model training using its data without sharing them with a central server. On the other hand, in split learning, both data and models are partitioned. Data are divided into segments, and the model is divided into parts. The model segments on different devices collaborate without directly sharing raw data. Split learning works like

Step 1: In a split-learning framework, the data are partitioned into multiple segments, each residing on a different edge device or party. These parties can be IoT devices, cameras, sensors, or any device within the smart city's network.

Step 2: A machine learning model is also divided into segments, with each part residing on a separate edge device. Each local model processes the data segment on its respective device and extracts relevant information.

Step 3: The intermediate output from one party's model is shared with another party's model, enabling the exchange of information without revealing the underlying data. This process is iterative, with models continually refining their understanding of the data.

Step 4: The intermediate outputs, containing model updates, are sent to a central server, where they are aggregated to improve the global model. The central server cannot access the raw data, ensuring privacy.

Step 5: The refined global model is deployed on edge devices to perform various tasks, such as cyberattack detection, without exposing sensitive information.

Figure 3 depicts how split learning may be used to detect anomalies in the smart city environment.

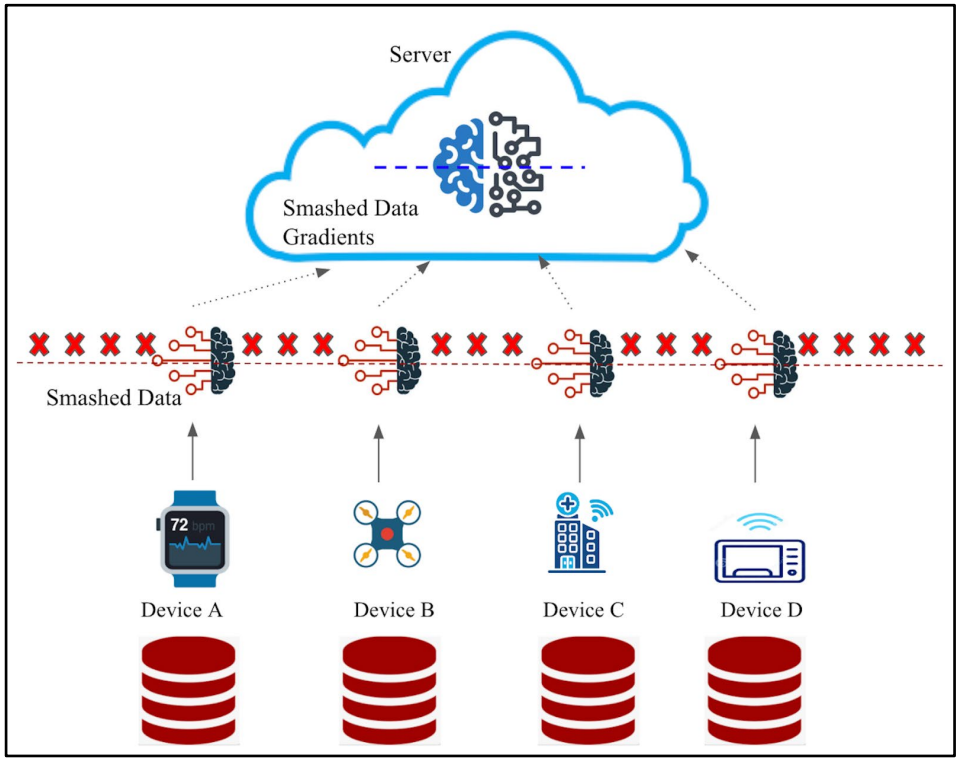

Figure 3. Split-learning architecture.

Split learning ensures that raw data from IoT devices remain on the edge, eliminating the need to centralize or share sensitive information. This privacy-focused approach aligns with the necessity to protect data and maintain regulatory compliance in smart cities. Moreover, the edge devices equipped with split-learning models can analyze data streams, such as network traffic, sensor data, and security logs, in real time. This real-time analysis enables swift identification of anomalies and potential cyber threats, critical for a timely response and mitigation. This method is highly scalable, as it can handle smart cities' vast and diverse ecosystems. Edge devices with split-learning models possess enhanced intelligence, allowing them to recognize emerging attack patterns and adapt to evolving threats. By processing and analyzing data locally, split learning minimizes latency and conserves network bandwidth. This efficiency ensures rapid cyberattack detection and response in time-sensitive scenarios. The distributed nature of this technique enhances resilience against network failures or localized attacks. Even if certain devices are compromised, the privacy-preserving design minimizes the potential impact on the overall system. The technique aligns with data privacy regulations, ensuring smart cities maintain compliance while detecting anomalies. It also facilitates accountability by allowing the auditability of model updates without exposing raw data.

3. Results and Observations

This section discusses the datasets used for the study and the performance metrics for evaluation. This is followed by the results obtained for the methods discussed above and a comparative analysis of the work with some similar previous works.

3.1. Datasets

For this study, the following datasets were considered:

- Network-Based Intrusion Detection Dataset (NSL-KDD) [25] contains a diverse set of network traffic data, including normal network activities and various types of network intrusions or attacks. It has 125,973 data points for the training dataset and 22,544 for the test dataset. The network intrusions are categorized as normal (non-intrusion) traffic), denial of service (DoS) attacks, probe attacks, user-to-root (U2R) attacks (e.g., unauthorized access attempts), and remote-to-local (R2L) attacks (e.g., unauthorized remote access). The primary attributes or features of the dataset are connection record, protocol type, service, flag, etc. There are 41 features in total.
- UNSW-NB15—University of New South Wales—Network Behavior 2015 dataset [26] incorporates 175,341 data points in the training set and 82,332 in the test set. The dataset includes diverse network traffic data comprising benign and malicious network activities and various network intrusions or cyberattacks. The dataset incorporates nine different types of attacks, some of which are exploits, shellcode, worms, generic, etc. Forty-nine features describe the dataset, including attributes like protocol type, service, source and destination IP addresses, and source and destination ports.

3.2. Performance Metrics for Evaluation

The study deployed five parameters for evaluating the performance of the models. The parameters are as follows:

- Accuracy: Accuracy measures the overall correctness of predictions
- Accuracy = Number of Correct Prediction/Total Number of Predictions;
- Precision: Precision focuses on the quality of positive predictions
- Precision = True Positives/(True Positives + False Positives);
- Recall: Recall measures the ability to identify all actual positive instances.
- Recall = True Positives/(True Positives + False Negatives);
- F1 Score: The F1 score is the harmonic mean of precision and recall, providing a balanced assessment;
- F1 Score = $2 * \text{Precision} * \text{Recall}/(\text{Precision} + \text{Recall})$
- Time-to-Run: Time-to-run represents the computational efficiency of the intrusion detection system. It measures the time taken to process the dataset, perform feature extraction, and make predictions. Minimizing running time is crucial for real-time or near-real-time intrusion detection systems in smart cities, where timely responses to threats are essential.

3.3. Results

For the analysis, the study considered the NSL-KDD and UNSW-NB15 datasets. Both datasets have multiple attributes and multiple target variables. Figures 4 and 5 depict the distribution of the attacks for both the datasets, respectively.

It is observed that most attacks are normal and denial of service (DoS) attacks. The number of attacks belonging to the probe, remote-to-local, and user-to-root categories are relatively fewer.

It is observed that most attacks for the UNSW-NB15 dataset are generic and exploits. The number of attacks belonging to the fuzzers, shellcode, exploits, etc. categories are relatively fewer. For analyzing anomaly detection in the smart city environment, the study deployed several machine learning algorithms in addition to federated learning and split learning. The analysis also considered some classical machine learning algorithms, ensembles, and deep learning algorithms as benchmarks for evaluation purposes. Algorithms like naive Bayes (NB), logistic regression (LR), decision tree (DT), random forest (RF), extreme gradient boosting (XGB), artificial neural networks (ANNs), convolutional neural networks (CNNs), long short-term memory (LSTM), and support vector machines (SVM) were deployed for comparing the performances across the two datasets. The primary metrics for evaluating the performance are accuracy, precision, recall, F-1 score, and the model training time. Tables 1 and 2 depict the model performance for the NSL-KDD dataset and the UNSW-NB15 dataset, respectively.

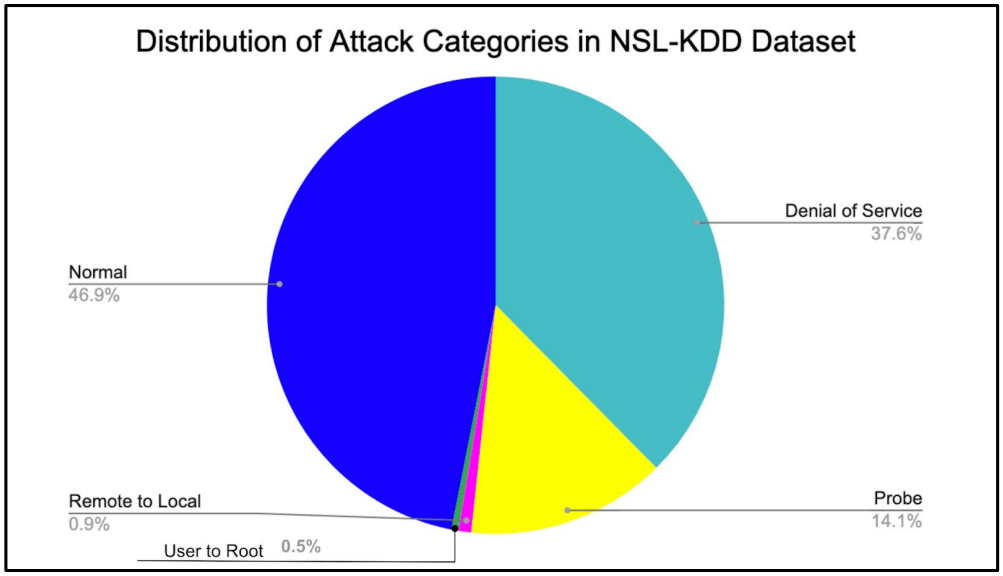

Figure 4. Distribution of attack categories for the NSL-KDD dataset.

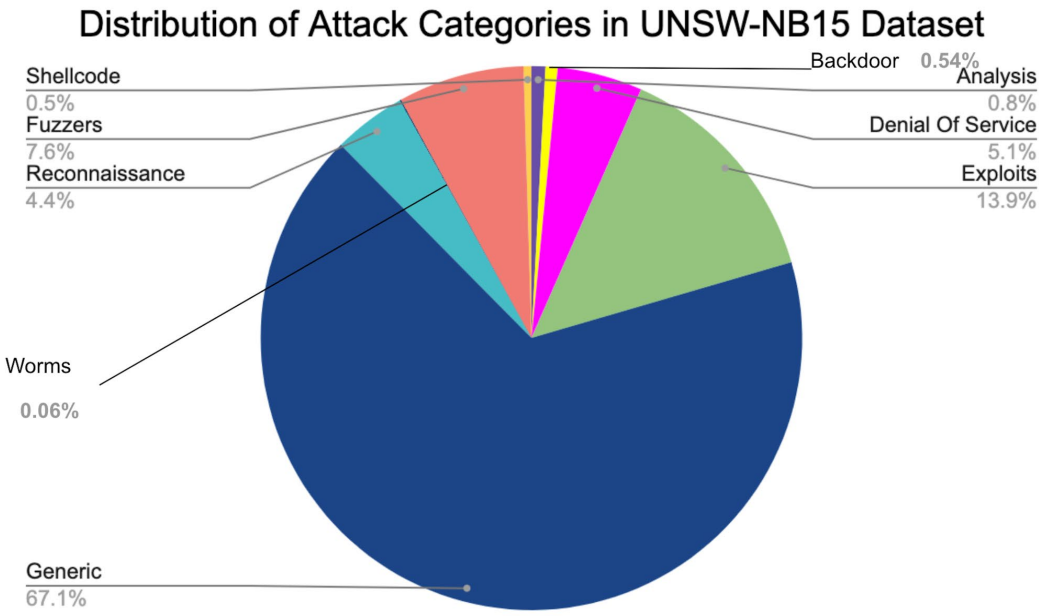

Figure 5. Distribution of attack categories for the UNSW-NB15 dataset.

Table 1. Performance evaluation of ML models on NSL-KDD dataset.

Models	Accuracy	Precision	Recall	F-1 Score	Time-to-Run (s)
NB	50.14	56.87	49.46	50.42	Less than 5
LR	68.97	75.07	62.76	67.86	Less than 5
DT	96.66	96.91	95.65	96.64	Less than 5
RF	97.84	98.32	93.96	96.99	7.88
XGB	97.06	97.55	94.04	96.28	6.31
ANN	86.54	89.89	80.77	87.66	12.88
CNN	99.09	99.23	93.28	99.12	72.66
LSTM	99.04	99.87	91.72	99.10	57.34
SVM	98.77	98.89	93.84	97.68	49.77
FL	98.99	99.32	93.22	98.24	225.46
SL	99.23	99.64	98.68	99.29	172.34

Table 2. Performance evaluation of ML models on UNSW-NB15 dataset.

Models	Accuracy	Precision	Recall	F-1 Score	Time-to-Run (s)
NB	88.98	88.87	88.80	88.83	Less than 5
LR	92.80	92.97	92.80	92.09	Less than 5
DT	96.38	96.38	96.32	96.44	Less than 5
RF	97.68	97.69	97.68	97.68	5.68
XGB	95.85	95.86	95.85	95.85	46.95
ANN	96.25	96.66	96.32	96.03	43.94
CNN	95.09	95.03	95.01	95.03	178.77
LSTM	96.48	96.32	96.44	96.46	127.56
SVM	93.08	92.77	93.55	92.79	88.76
FL	97.78	97.64	97.56	97.89	232.56
SL	98.02	98.12	98.02	98.11	222.33

Based on the evaluation metrics, it is observed that naive Bayes and logistic regression exhibit the most unsatisfactory performance, while CNN, LSTM, federated learning and split learning exhibit the best performance in terms of accuracy, precision, recall and F-1 scores. The accuracy scores for CNN, LSTM, federated learning, and split learning are 99.09, 99.04, 98.99, and 99.32, respectively. It is also observed that the training time for federated learning (225.46 s) is higher compared to split learning (172.34), which is higher compared to CNN (72.66) and LSTM (57.34). Federated learning and traditional neural networks have different purposes and execution times, making direct comparisons challenging. However, the execution time of federated learning can be influenced by several factors like communication overhead, number of participating devices, model complexity, and local device capabilities. Therefore, although the time-to-train is slightly high for federated learning and split learning, given the number of devices, it may be worth considering the models for anomaly detection in the IoT environment. Regarding training time and accuracy, random forest, and extreme gradient boosting show similar performance; in terms of accuracy, split learning performs best.

Based on the evaluation metrics, it is observed that naive Bayes, logistic regression, and support vector machine exhibit the most unsatisfactory performance, while random forests, LSTM, federated learning, and split learning exhibit the best performance in terms of accuracy, precision, recall, and F-1 scores. The accuracy scores for random forest (RF), LSTM, federated learning, and split learning are 97.68, 96.48, 97.78, and 98.02, respectively. It is also observed that the training time for federated learning (232.56) is higher compared to split learning (222.33), which is higher compared to random forest (5.68) and LSTM (127.56). Regarding training time, RF performs the best, and in terms of accuracy, split learning performs the best. Figures 6 and 7 show the performance evaluation of both the datasets concerning accuracy using the Ml models and federated and split learning.

Figure 6. Performance evaluation of ML models for both the datasets.

Figure 7. The training time of ML models for both the datasets.

The "time to run" values in Tables 1 and 2 represent an average over multiple runs and indicate a commitment to capture the algorithm's consistency in terms of performance and stability. Encouraging multiple runs provides a more robust assessment, accounting for potential variations in execution times due to factors like dataset nuances or algorithmic randomness. This approach also enhances the reliability of the results and ensures that the time metrics reflect a representative measure of the algorithm's efficiency in real-world scenarios. This practice also aligns with best practices in machine learning research, fostering transparency and enabling a more nuanced understanding of the algorithm's computational behavior.

It is observed that federated learning and split learning fare well compared to many other baseline models considered for the study in terms of accuracy. CNN and LSTM also show good performance.

Federated learning and split learning have the maximum training time compared to the other baseline models considered for the study. In federated learning, model updates are typically sent between edge devices (e.g., smartphones and IoT devices) and a central server or federated learning coordinator. The frequency and volume of these communications can impact execution time, especially in scenarios with limited network bandwidth or high latency. The more devices participate in federated learning, the longer it may take to aggregate and synchronize model updates. Moreover, large-scale federated learning deployments with numerous devices can introduce delays. The complexity of the trained machine learning model can affect execution time as more complex models may require more iterations or epochs to converge, extending the training time. Edge devices participating in federated learning may have varying computational capabilities. Some devices may process updates quickly, while others may be slower, impacting the overall execution time. Federated learning can be slower than split learning due to the need to coordinate and synchronize updates from distributed devices. In split learning, each device or party processes a part of the model, and the speed of split learning can be influenced by the computational capabilities of individual devices and the ability to process model updates in parallel. Hence, split learning can be faster than federated learning due to no involvement of communication overhead.

3.4. Comparative Analysis

This section presents a comparative analysis of the proposed work with some similar works performed in the past. Table 3 presents the overall summary of the comparative analysis.

Table 3. Comparative analysis of proposed work with similar works.

Author and Year	Research	Methodologies/Parameters	Results
Ding et al., 2023 [10]	Detecting cyberattacks using Deep learning	RSR, TRB, and DB on TON-IoT, Edge-IIoTset, and UNSW-NB15 datasets	Accuracy~90.57%, 94.96%, and 98.41%
Sharma and Babbar, 2023 [11]	Detecting cyberattacks in smart transportation	Rf, NB, and DT on BoT-IoT dataset	RF shows the highest accuracy of 91%

Table 3. *Cont.*

Author and Year	Research	Methodologies/Parameters	Results
Alrayes et al., 2023 [13]	Deep learning with chaotic poor and rich optimization for intrusion detection in smart city environment	Butterfly optimization algorithm (BOA) and deep sparse autoencoder (DSAE) on CICIDS dataset	Accuracy is 98.53%
Lin et al., 2024 [16]	Multivariate anomaly detection framework	Hierarchical context representation learning with deep learning methods	Proposed method shows precision values 0.90, 0.89, and 0.90 on three different datasets.
Mitropoulou et al., 2024 [17]	Anomaly detection in cloud computing	Knowledge graph embedding with isolation forest and CBLOF	Precision values for the methods proposed are 0.79 and 0.62.
Jithish et al., 2023 [18]	Distributed anomaly detection in smart grids	Federated learning compared against other algorithms	Federated learning with convolutional neural networks achieve highest accuracy of 0.989
Dang et al., 2021 [19]	Anomaly detection in IoT sensory data	Monotone split and conquer (MSC) technique	Increased accuracy by 8%, reduced false-positive rate by 5%
Nixon et al., 2021 [22]	Anomaly detection in network data streams	Split active learning with autoencoders	Highest accuracy achieved is 98.78%
Dragoi et al., 2022 [23]	Network intrusion detection with data shifting	Protocol splitting technique, AnoShift	ROC curve shows best performance for local outlier factor, 91.50
Zhang et al., 2023 [24]	Detecting anomalous network traffic	Semi-supervised learning and adaptive multiclass balancing	Precision and recall enhanced up to 5.7%,
Kasongo, 2023 [27]	Intrusion detection in IoT	Recurrent neural network, LSTM, gated recurrent units on NSL-KDD and UNSW-NB15	XGB-LSTM achieved highest accuracy~88.13% on NSL-KDD XGB-RNN achieved highest accuracy~86.93% on UNSW-NB15
Jahromi et al., 2023 [28]	Cyber-threat hunting model for industrial internet of things	Ensemble-based deep federated learning	Accuracy~94% to 99%
Alazab et al., 2023 [29]	Privacy-preserving intrusion detection	Federated learning on NSL-KDD dataset	97.77% Accuracy
Proposed Work	**Anomaly detection in IoT devices in smart city**	**Federated learning, split learning, classical ML models, ensembles, deep learning models on NSL-KDD and UNSW-NB15**	**Federated learning achieves accuracy of 98.99% and 97.78%. Split learning achieves an accuracy of 99.23% and 98.02%, respectively.**

Based on the experimental and comparative analysis, a few observations can be made.

- Performance of Anomaly Detection Methods—The study comprehensively evaluated various anomaly detection methods, including classical machine learning algorithms (e.g., naive Bayes, logistic regression, and decision trees), ensemble models (e.g., random forests and XGBoost), and deep learning models (e.g., ANN and CNN). It observed that the performance varied significantly across these methods.
- Deploying Federated Learning—Federated learning demonstrated promising results concerning IoT anomaly detection. It exhibited strong potential for maintaining data privacy while achieving satisfactory accuracy despite longer training times.
- Deploying Split Learning—Split learning showcased the ability to safeguard sensitive information within IoT environments while delivering significantly high accuracy.
- Training Time—It is observed that the raining times associated with privacy-preserving techniques like federated learning and split learning are relatively high compared to the other models. This may be attributed to several factors ranging from model complexity to the number of devices.
- Multiple Datasets—Two different datasets were considered: NSL-KDD and UNSW-NB15, and the models exhibited consistent performance across both datasets.
- Evaluation Parameters—The analysis adopted accuracy, precision, recall, and F1 scores to evaluate the model performance. The models were also compared based on training times.

4. Conclusions

In this paper, anomaly detection was performed within IoT devices operating in smart city environments, primarily focusing on safeguarding against cybersecurity threats. Various machine learning techniques were comprehensively analyzed, ranging from traditional algorithms like naive Bayes, random forests, support vector machines, and neural networks to cutting-edge techniques like federated learning and split learning. Valuable insights into the strengths and weaknesses of these approaches have been gained through rigorous experimentation using the NSL-KDD and UNSW-NB15 datasets. Notably, federated learning and split learning have emerged as promising strategies for balancing data privacy and detection accuracy. While these methods did entail longer training times, their potential for safeguarding sensitive information in IoT environments cannot be underestimated. The focus will be on optimizing federated and split-learning models to reduce training times and computational demands in real time. Furthermore, there is a potential possibility of exploring hybrid models that combine the strengths of federated and split learning. These hybrid models promise to achieve even higher detection performance while preserving privacy. This study lays the foundation for future research and practical implementations dedicated to fortifying IoT device security within smart cities. The resilience of smart cities can be enhanced against cyberattacks by ensuring the safety and reliability of the devices connecting these environments.

Funding: This research received no external funding.

Data Availability Statement: The data were downloaded from www.kaggle.com (accessed on 6 December 2023). The study deploys two datasets: the NSL-KDD dataset and the UNW-NB15 dataset.

Conflicts of Interest: The author declares no conflicts of interest.

References

1. Rashid, M.M.; Kamruzzaman, J.; Hassan, M.M.; Imam, T.; Wibowo, S.; Gordon, S.; Fortino, G. Adversarial training for deep learning-based cyberattack detection in IoT-based smart city applications. *Comput. Secur.* **2022**, *120*, 102783. [CrossRef]
2. Priyadarshini, I.; Alkhayyat, A.; Gehlot, A.; Kumar, R. Time series analysis and anomaly detection for trustworthy smart homes. *Comput. Electr. Eng.* **2022**, *102*, 108193. [CrossRef]
3. Priyadarshini, I.; Mohanty, P.; Alkhayyat, A.; Sharma, R.; Kumar, S. SDN and application layer DDoS attacks detection in IoT devices by attention-based Bi-LSTM-CNN. *Trans. Emerg. Telecommun. Technol.* **2023**, *34*, e4758. [CrossRef]
4. Thapa, C.; Chamikara MA, P.; Camtepe, S.A. Advancements of federated learning towards privacy preservation: From federated learning to split learning. *Fed. Learn. Syst. Towards Next Gener. AI* **2021**, 79–109.
5. Ajao, L.A.; Apeh, S.T. Secure edge computing vulnerabilities in smart cities sustainability using petri net and genetic algorithm-based reinforcement learning. *Intell. Syst. Appl.* **2023**, *18*, 200216. [CrossRef]
6. Rashid, M.M.; Kamruzzaman, J.; Hassan, M.M.; Imam, T.; Gordon, S. Cyberattacks detection in iot-based smart city applications using machine learning techniques. *Int. J. Environ. Res. Public Health* **2020**, *17*, 9347. [CrossRef]
7. Mukherjee, D. Detection of data-driven blind cyber-attacks on smart grid: A deep learning approach. *Sustain. Cities Soc.* **2023**, *92*, 104475. [CrossRef]
8. Almuqren, L.; Aljameel, S.S.; Alqahtani, H.; Alotaibi, S.S.; Hamza, M.A.; Salama, A.S. A White Shark Equilibrium Optimizer with a Hybrid Deep-Learning-Based Cybersecurity Solution for a Smart City Environment. *Sensors* **2023**, *23*, 7370. [CrossRef]
9. Alsaade, F.W.; Al-Adhaileh, M.H. Cyber attack detection for self-driving vehicle networks using deep autoencoder algorithms. *Sensors* **2023**, *23*, 4086. [CrossRef]
10. Ding, W.; Abdel-Basset, M.; Mohamed, R. DeepAK-IoT: An effective deep learning model for cyberattack detection in IoT networks. *Inf. Sci.* **2023**, *634*, 157–171. [CrossRef]
11. Sharma, A.; Babbar, H. BoT-IoT: Detection of Attacks in IoT-Cybersecurity for Smart Transportation. In Proceedings of the 2023 8th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 1–3 June 2023; pp. 522–527.
12. Ajao, L.A.; Apeh, S.T. Blockchain Integration with Machine Learning for Securing Fog Computing Vulnerability in Smart City Sustainability. In Proceedings of the 2023 1st International Conference on Advanced Innovations in Smart Cities (ICAISC), Jeddah, Saudi Arabia, 23–25 January2023; pp. 1–6.
13. Alrayes, F.S.; Asiri, M.M.; Maashi, M.; Salama, A.S.; Hamza, M.A.; Ibrahim, S.S.; Zamani, A.S.; Alsaid, M.I. Intrusion Detection Using Chaotic Poor and Rich Optimization with Deep Learning Model for Smart City Environment. *Sustainability* **2023**, *15*, 6902. [CrossRef]
14. Al-Taleb, N.; Saqib, N.A. Towards a hybrid machine learning model for intelligent cyber threat identification in smart city environments. *Appl. Sci.* **2022**, *12*, 1863. [CrossRef]

15. Bilakanti, H.; Pasam, S.; Palakollu, V.; Utukuru, S. Anomaly detection in IoT environment using machine learning. *Secur. Priv.* **2024**, e366. [CrossRef]
16. Lin, C.; Du, B.; Sun, L.; Li, L. Hierarchical Context Representation and Self-adaptive Thresholding for Multivariate Anomaly Detection. *IEEE Trans. Knowl. Data Eng.* **2024**, 1–12. [CrossRef]
17. Mitropoulou, K.; Kokkinos, P.; Soumplis, P.; Varvarigos, E. Anomaly Detection in Cloud Computing using Knowledge Graph Embedding and Machine Learning Mechanisms. *J. Grid Comput.* **2024**, *22*, 6. [CrossRef]
18. Jithish, J.; Alangot, B.; Mahalingam, N.; Yeo, K.S. Distributed Anomaly Detection in Smart Grids: A Federated Learning-Based Approach. *IEEE Access* **2023**, *11*, 7157–7179. [CrossRef]
19. Dang, T.B.; Le, D.T.; Nguyen, T.D.; Kim, M.; Choo, H. Monotone split and conquer for anomaly detection in IoT sensory data. *IEEE Internet Things J.* **2021**, *8*, 15468–15485. [CrossRef]
20. Nassif, A.B.; Talib, M.A.; Nasir, Q.; Dakalbab, F.M. Machine learning for anomaly detection: A systematic review. *IEEE Access* **2021**, *9*, 78658–78700. [CrossRef]
21. Takiguchi, Y.; Shiono, S. Split Training Method to Generate Data Driven Model for Gas Turbine Engine Anomaly Detection. In Turbo Expo: Power for Land, Sea, and Air. *Am. Soc. Mech. Eng.* **2020**, *84140*, V005T05A027.
22. Nixon, C.; Sedky, M.; Hassan, M. SALAD: An Exploration of Split Active Learning based Unsupervised Network Data Stream Anomaly Detection using Autoencoders. *TechRxiv* **2021**.
23. Dragoi, M.; Burceanu, E.; Haller, E.; Manolache, A.; Brad, F. AnoShift: A distribution shift benchmark for unsupervised anomaly detection. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 32854–32867.
24. Zhang, H.; Xiao, Z.; Gu, J.; Liu, Y. A network anomaly detection algorithm based on semi-supervised learning and adaptive multiclass balancing. *J. Supercomput.* **2023**, *79*, 20445–20480. [CrossRef]
25. NSL-KDD Dataset. Available online: http://nsl.cs.unb.ca/nsl-kdd/ (accessed on 30 November 2023).
26. Moustafa, N.; Slay, J. UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In Proceedings of the 2015 Military Communications and Information Systems Conference (MilCIS), Canberra, ACT, Australia, 10–12 November 2015; pp. 1–6.
27. Kasongo, S.M. A deep learning technique for intrusion detection system using a Recurrent Neural Networks based framework. *Comput. Commun.* **2023**, *199*, 113–125. [CrossRef]
28. Jahromi, A.N.; Karimipour, H.; Dehghantanha, A. An ensemble deep federated learning cyber-threat hunting model for Industrial Internet of Things. *Comput. Commun.* **2023**, *198*, 108–116. [CrossRef]
29. Alazab, A.; Khraisat, A.; Singh, S.; Jan, T. Enhancing Privacy-Preserving Intrusion Detection through Federated Learning. *Electronics* **2023**, *12*, 3382. [CrossRef]