applied sciences

MDPI

Article

Protocol-Specific and Sensor Network-Inherited Attack Detection in IoT Using Machine Learning

F. Zahra [1], N. Z. Jhanjhi [1,*], N. A. Khan [1], Sarfraz Nawaz Brohi [2], Mehedi Masud [3] and Sultan Aljahdali [3]

[1] School of Computer Science (SCS), Taylor's University, Subang Jaya 47500, Malaysia
[2] Computer Science and Creative Technologies, University of the West of England, Bristol BS16 1QY, UK
[3] Department of Computer Science, College of Computers and Information Technology, Taif University, P.O. Box 11099, Taif 21944, Saudi Arabia
* Correspondence: noorzaman.jhanjhi@taylors.edu.my

Abstract: For networks with limited resources, such as IoT-enabled smart homes, smart industrial equipment, and urban infrastructures, the Routing Protocol for Low-power and Lossy Networks (RPL) was developed. Additionally, a number of optimizations have been suggested for its application in other contexts, such as smart hospitals, etc. Although these networks offer efficient routing, the lack of active security features in RPL makes them vulnerable to attacks. The types of attacks include protocol-specific ones and those inherited by wireless sensor networks. They have been addressed by a number of different proposals, many of which have achieved substantial prominence. However, concurrent handling of both types of attacks is not considered while developing a machine-learning-based attack detection model. Therefore, the ProSenAD model is proposed for addressing the identified gap. Multiclass classification has been used to optimize the light gradient boosting machine model for the detection of protocol-specific rank attacks and sensor network-inherited wormhole attacks. The proposed model is evaluated in two different scenarios considering the number of attacks and the benchmarks for comparison in each scenario. The evaluation results demonstrate that the proposed model outperforms with respect to the metrics including accuracy, precision, recall, Cohen's Kappa, cross entropy, and the Matthews correlation coefficient.

Keywords: RPL protocol; secure IoT; protocol-specific attacks; sensor network-inherited attacks; attack detection; machine learning

Citation: Zahra, F.; Jhanjhi, N.Z.; Khan, N.A.; Brohi, S.N.; Masud, M.; Aljahdali, S. Protocol-Specific and Sensor Network-Inherited Attack Detection in IoT Using Machine Learning. *Appl. Sci.* **2022**, *12*, 11598. https://doi.org/10.3390/app122211598

Academic Editors: Kwangjo Kim and Kwok-Yan Lam

Received: 18 October 2022
Accepted: 9 November 2022
Published: 15 November 2022

1. Introduction

The Internet of Things (IoT) technology has become an important paradigm for building smart infrastructures such as smart healthcare systems, smart homes, smart cities, and IoT-enabled smart industrial systems [1]. It is an effort to progress toward ever-connected architectures. The devices in these infrastructures, including sensors, actuators, and systems of interconnected things, can perform machine-to-machine communication [2] and participate in decision-making processes. Therefore, IoT can be described as an interconnection of a multitude of information-sensing and actuating equipment embedded within everyday objects. IoT-enabled wearables, security alarms, smartphones, etc., are some examples of these devices that may communicate and share information across various application domains. They connect through the internet using advanced Internet Protocols (IP) such as Internet Protocol version 6 (IPv6) and communicate with each other via IPv6 over Low-power Wireless Personal Area Networks (6LoWPAN). Features such as location, light, heat, and heart rate are sensed, and relevant data is forwarded to border devices and cloud systems for further processing [3,4]. According to the global statistical studies performed by Statista Research and Analysis department, approximately twenty-nine billion IoT-enabled devices will be interconnected throughout the globe by 2030 [5].

At the base of IoT architectures, Wireless Sensor Networks (WSNs) are situated in which the nodes are resource constrained. Conventionally, they depend on power-scavenging technologies to communicate and function [6]. Therefore, these domains are under continuous research for enhancements concerning the development of compatible components for deployment in resource-constrained environments [7]. It is typical for such systems to have inadequate storage capacity as well as low processing power. These characteristics require befitting communication protocols and network standards for data transmission. Figure 1 illustrates a use case of an IoT-enabled smart healthcare system.

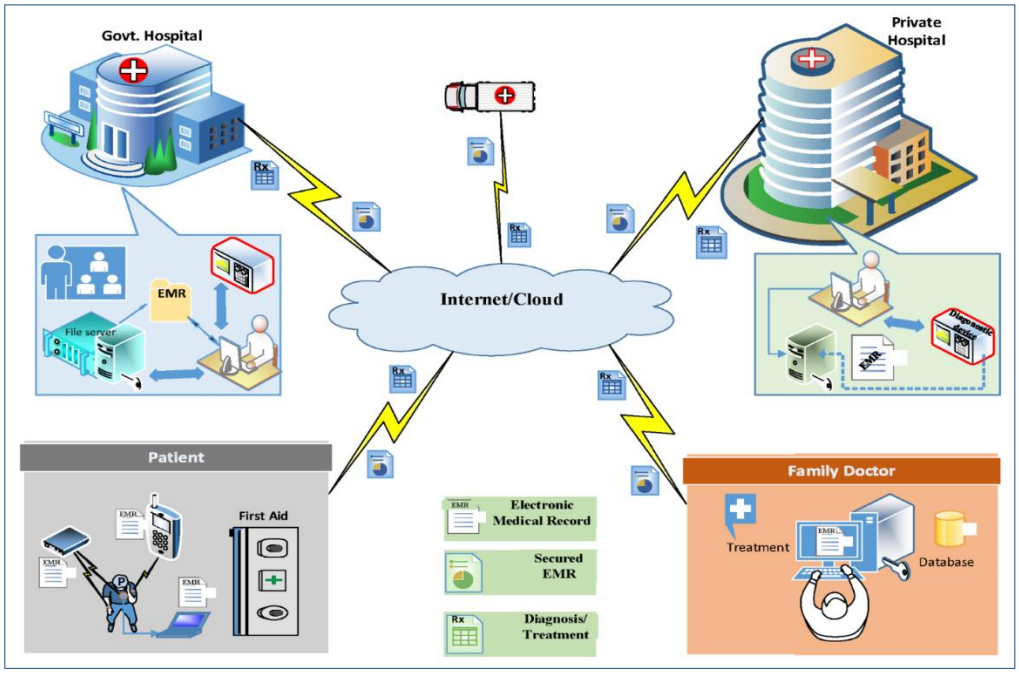

Figure 1. IoT-enabled use case of smart healthcare system [8].

IoT systems use RPL for routing and run on 6LoWPAN [9]. However, the low-power and lossy characteristics, heterogeneity, and resource-constrained nature make them highly vulnerable to security attacks. These attacks are classified into two main categories: (1) protocol-specific and (2) Sensor Network (SN)-inherited attacks [10]. It is necessary to investigate these attacks and propose adequate solutions for a secure transition toward IoT technology on a large scale.

In the literature, various approaches for addressing the routing and network security in IoT are proposed, including machine-learning-based (ML-based), intrusion-detection-based, and protocol-optimization-based strategies [11]. Because of the various advantages such as automation, feature extraction, and pattern recognition in the network traffic data, the ML-based approach has the potential to fulfill the routing security needs. Furthermore, because IoT produces big data, it highly benefits the ML-based model building procedures because of their dependence on the datasets. Several ML-based solutions have been proposed by researchers to address the security attacks in IoT. However, in the RPL-based IoT domain, the concurrent detection of protocol-specific and SN-inherited attacks is insufficiently addressed, particularly for protocol-specific rank (PS-R) attacks and SN-inherited wormhole (SN-W) attacks. These attacks are among the most detrimental and damaging attacks for the routing mechanisms and network resources.

This research work focuses on the security of IoT networks that implement RPL for routing. An ML-based model is proposed to improve the security against PS-R attacks as well as SN-W attacks. The approaches to build the model are carefully selected depending on the nature of the dataset, and the objectives to be achieved. The model parameters are carefully selected through critical analysis, while considering their value and impact

on optimization. The proposed model, named Protocol-specific and SN-inherited Attack Detection (ProSenAD), is evaluated considering the type of learning paradigm used for building the model, which is classification, and the type of classification considered, which is multiclass classification.

The performance of the ProSenAD model is evaluated using the testing set of the self-generated LIoTN-RPL dataset. The results demonstrate that the proposed model outperforms when compared with the benchmarks in terms of: (1) number of attacks detected when compared with ML-LGBM [12] and GRU-DL [13], (2) accuracy, precision, recall in comparison with GAN-C [14], (3) Cross Entropy (CE), Cohen's Kappa (CK), and the Matthews Correlation Coefficient (MCC), when compared with gradient boosting (GB), extreme GB (XGBoost), and light gradient boosting machine (LGBM). Following is the summary of contributions of this research work:

- The ML-based approaches are analyzed for attack detection in IoT and RPL-based networks to secure the routing mechanisms as well as the network resources.
- We intend to improve the ML models for concurrent detection of protocol-specific and SN-inherited attacks.
- The proposed model is evaluated using ML-based metrics, such as accuracy, Positive Predictive Values (PPV), sensitivity, CE, inter-rater reliability and agreement through CK, and MCC to determine its performance in comparison with the existing models.
- The lack of updated dataset is addressed by generating novel LIoTN-RPL dataset using different network simulation scenarios that is then employed for developing attack detection model using ML approaches.

The aim of this research is to contribute to the enhancement of IoT routing and network security for its widespread secure adoption in different domains. The remaining paper is structured as follows: Section 2 describes the protocol-specific and SN-inherited attacks focusing on PS-R and SN-W attacks, followed by the review of existing methods for attack detection. Section 3 presents the proposed ProSenAD model in complete detail along with the process model adopted for different model building phases. The novel LIoTN-RPL dataset generated for this study is also explained in addition to the parameters selected for model development and optimization. Section 4 discusses the results and comparative performance analysis. Section 5 provides the conclusion and future research direction.

2. Related Work

In this section, the protocol-specific and SN-inherited attacks in RPL-based IoT networks are discussed with a focus on PS-R and SN-W attacks. Their workings and impact on the IoT environments are presented in the next subsections.

2.1. Protocol-Specific Attacks in RPL-Based IoT

Protocol-specific attacks are named based on an RPL feature or mechanism that an attacker aims to target. For example, in a PS-R attack, the attacker manipulates the rank value, rank type, and its routing operation in an IoT network. Since this attack is capable of modifying or altering RPL features, impacting the network performance, forming routing loops, and disrupting the topology [15], it is considered one of the highly damaging attacks. A malicious node manipulates the rank value and objective function through control messages, severely affecting the routing topology [16]. To increase the attack impact, it broadcasts a fake, lower rank value (decreased rank attack) and shortest distance toward the root node in the network. Consequently, the child IoT nodes add the attacker node to its preferred parent list and, in other cases, directly select it as a parent. This causes the RPL to rebuild the directed acyclic graph (DAG). Moreover, the resources of victim nodes are wasted in the attack process. The rank attack causes loop formation, which results in node isolation at the individual or cluster level where a group of victim nodes are isolated and obstructed from communicating in the network. If the rank rule in RPL is compromised, it can cause control overhead, delay, and packet collisions [17]. When this type of attack is

conceptualized in a critical IoT-enabled scenario, such as healthcare, the impact is severely detrimental and may cause irreversible damage to the system and associated bodies.

There are three popular variants of rank attacks: (1) decreased rank attack, (2) increased rank attack, and (3) worst parent attack [10]. In the decreased rank attack, the malicious nodes illegitimately broadcast lower rank to other nodes and attract network traffic. As a result, most of the normal nodes choose the illegitimate node as their preferred parent [10]. This attack is similar to the SN-based sinkhole attack in some regards. In contrast, adversaries in an increased rank attack advertise a higher rank value illegitimately to deteriorate the nearest parent selection process. Consequently, the routing topology is disrupted, node communications are delayed, and latency is introduced in the overall RPL network by forcing the nodes to select other nodes as parents that might be farther away from the root node [18]. In the worst parent selection, the victims simply select a parent with poor communication or routing capabilities because of rank manipulation by the attacker. Overall, the important consequences of rank attacks are communication delay resulting from disruption in packet transmission, end-to-end delay, latency, weak or worsened routing path, loop formation, and decreased packet delivery ratio.

2.2. SN-Inherited Attacks in RPL-Based IoT

SN-inherited attacks are named so because they inherit the attacking mechanisms from the WSNs and optimize them to target the IoT networks. Another factor is the presence of foundational base of sensor networks in an IoT infrastructure which contributes towards the instigation of SN-inherited attacks in IoT. In an SN-W attack, two malicious nodes cooperate to create a tunnel [19] between each other and entirely or selectively transmit the network traffic maliciously through a poor routing path, rather than sending it through the original route. The attacking nodes establish a fake link between two nodes, which is apparently fast with low latency [20], to target attack the network. The goal is to disrupt the routing mechanism by misguiding the victim nodes and exhausting the network resources. An SN-W attack is capable of instigating other attacks in the network, such as selective forwarding, packet dropping, black hole, grey hole, denial of service [21], and Sybil attacks [22]. The attacker(s) sniffs, eavesdrops, and replays the data packets, which consequently impacts the overall network performance. Therefore, in this paper, this attack from the SN-inherited category is selected for detection along with the PS-R attack from the protocol-specific category of RPL attacks.

There are three methods of wormhole creation, which include (1) packet relay, (2) packet encapsulation, and (3) out-of-bound link. In the packet encapsulation method, malicious nodes use a path, which is originally meant for sending regular data, and create a logical tunnel by encapsulating the data packets [10]. This is done to hide the hop counts from other nodes present on the tunnel's route. In the packet relay method, one or more illegitimate nodes send packets or control messages between two legitimate nodes, which are located far from each other, to mislead them into being close neighbors [23]. Usually, this is done by transmitting packets without updating the hop count [10,24]. In the out-of-bound strategy, a wormhole is created by utilizing a wired or wireless link that is out of the network boundary to create a tunnel between the external attacker and an internal malicious node.

2.3. Machine Learning for Securing IoT—An Overview

In this section, the use of ML approaches in IoT is explored with an emphasis on security. Different Learning Paradigms (LP) are identified that are exploited for designing and developing security solutions in IoT infrastructures by the research community.

Supervised Learning (SL) approaches have been extensively used for the development of solutions to address the security issues in IoT-enabled systems. For example, Ref. [25] have proposed supervised classifiers for the detection of security attacks including DoS, DDoS, cross-site scripting (XSS), injection and scanning attacks, backdoor malware, and password cracking attacks in Vehicular Ad hoc Networks (VANETs). ToN-IoT dataset,

which contains the aforementioned attacks, has been used to train eight SL models in two contexts: (1) binary classification, and (2) chi-square and synthetic minority oversampling technique-included classification. The models are then evaluated against performance evaluators such as accuracy, precision, recall, F1-score, and confusion matrix. XGBoost has performed well overall with the highest accuracy and confusion matrix parameters such as False Positive Rate (FPR), while in the second context, k-NN has performed well among other models.

In [26], the authors have proposed a DL-based attack detection framework leveraging fog technology to train supervised DL models for attack detection in IoT networks. Six models are trained on five datasets and Long Short-Term Memory (LSTM) has outperformed other SL models. The architecture of the LSTM model is based on artificial recurrent neural networks, and they are well-suited for classification as well as prediction-related tasks. The authors of [27] have used the distributed LSTM as part of their proposal for energy efficient calculations in mobile edge computing systems. The authors of [28] have proposed a DL-based anomaly detection technique using binary and multiclass classification techniques in six datasets. They have trained convolutional neural networks (CNN) for classifying the normal and attack data as well as further classification of attacks using the transfer learning approach. The authors of [29] have proposed a novel feature selection approach to improve the accuracy of ML models for anomaly detection. Furthermore, four models are trained for attack detection in bot-related traffic. It is observed that the researchers have proposed IDSs, classification-based models, and improved feature selection methods for increasing the classification accuracy of the security models for IoT systems.

Unsupervised Learning (UL) approaches have been used to develop models for anomaly detection, attack detection, node clustering, and pattern recognition. For example, the authors of [30] have proposed a DL-based unsupervised learning method to detect botnets in IoT. Balanced and unbalanced datasets are used to evaluate the model efficiency by detecting the threats. The false-positive rate has been used to evaluate the detection capability and performance of the model. The authors of [31] have employed unsupervised and supervised learning methods to detect intrusions in IoT using a multistage approach and employing a feed-forward neural network with a single hidden layer. SVM is used with a synthetic minority oversampling approach for clustering and data reduction. This approach has achieved good results in comparison with other classifiers used as benchmarks. The authors of [32] have incorporated unsupervised-learning-based dimensionality reduction using one-class SVM, autoencoder, and isolation forest-based techniques. In [33], the authors have used autoencoders as UL models for detecting network intrusions in IoT.

In [34], the authors have proposed semi-supervised learning (SSL)-based IDS for handling intrusions in IoT networks. Deep learning is used to develop the system and train it on two datasets. The IDS is evaluated for seven security attacks namely: DDoS, bots, infiltration, PortScan, web attacks, and brute force attacks. The IDS is deployed using Python, Keras, and TensorFlow, and evaluated using accuracy precision, recall, and F1-score. Similarly, Ref. [35] have leveraged supervised DL and unsupervised clustering techniques to develop an SSL approach to address the attacks present in the NSL-KDD dataset. The proposed method is tested on the IoT-fog testbed and 99.78% accuracy is achieved. The authors of [36] have also leveraged a DL-based SSL approach to address security attacks in IoT. Transfer learning methodology is used to develop the solution by training it on nine IoT datasets. The technique is evaluated using comparative effectiveness of information transfer, analysis of processing time, and performance comparison of the models on labeled and unlabeled datasets using AUC scores. The authors of [37] have implemented semi-supervised and federated learning approaches in an industrial IoT use case. Federated learning is used to locally train the model, and an active learning-based SSL technique is used to globally adjust the model. A 7.1% accuracy increment is observed through the proposed method in ten active learning queries.

Numerous reviews, investigative studies, and surveys have been conducted by the researchers in the domain of IoT networks, security challenges, and routing standards used

by LLNs such as IoT and WSNs. For instance, in [38], the authors have studied ML and DL methods for securing IoT ecosystems. They have explored the potential role of ML and DL in improving security aspects of these systems and developed a thematic taxonomy in terms of IoT security threats, learning methods to counter them, layer-wise security approaches using ML and DL, and other dimensions of IoT security. Both ML and DL approaches for IoT security are categorized into (1) supervised, (2) unsupervised, (3) semi-supervised, and (4) reinforcement learning. An elaboration of respective techniques and possibilities of their application in IoT systems is also presented. They are discussed in terms of working principles, advantages, disadvantages, and potential for successful application in an IoT environment. Similarly, Ref. [39] have surveyed ML and DL algorithms categorically for IoT application security. Moreover, security problems and threat models are systematically surveyed, and complexity analysis of ML algorithms is performed along with the discussion on the limitations of their application in IoT networks. The authors of [40] have performed a detailed survey on DL and big data technologies for security in IoT. DL-based architectures and frameworks are explained along with performance evaluation metrics. Additionally, layer-wise security attacks and datasets used for experimental analysis are discussed in detail.

In [41], a comprehensive review is conducted on ML-based security solutions for power systems. Various aspects of these systems are covered, including Power Quality Disturbances (PQD), Voltage Stability Assessment (VSA), Transient Stability Assessment (TSA), and Supervisory Control and Data Acquisition (SCADA). ML classifiers, IDSs, and other ML-based security approaches are explored, compared, and evaluated in terms of accuracy for the four power system facets mentioned earlier. The authors of [42] have presented a brief review of ML techniques for improvement in the security of smart grids, detection of power quality events, estimation of transformer life loss, making energy dispatch decisions, and operations of the electricity market.

ML has been used in IoT and WSNs for security enhancements in recent years. This solution development domain has gained the acclaim of researchers due to its prospective potential for developing robust security models. For instance, in [43], the authors have used ML as part of their proposal of a robust architecture for adversarial attack detection to improve the identification and classification of High Spatial Resolution Remote Sensing (HSRRS) images. They have used adversarial detection models based on SVM with single or fused two-level features to improve detection accuracy. The proposed model has achieved an overall accuracy of 94.5%, detection probability of 0.933, and false alarm probability of 0.040. The performance evaluation has indicated that the proposed model obtains better results as compared to other methods used in previous relevant studies. In another study, [44], have used ML as part of analytical data algorithm proposed for the identification of two False Data Injection (FDI) attacks in Industrial Control Systems (ICSs), i.e., measurement injection attack and control variable tampering attack. In [45], the authors have used a random-forest-based ML model to classify four different types of botnet attacks in an IoT-enabled smart factory environment. An average accuracy of 96.67%, 0.241 FPR, and model reliability of 94.6% is achieved which is calculated using kappa coefficient. In [46], the authors have proposed an ameliorated ANN-based model that employs dimensionality reduction technique to improve the DDoS attack detection process in IoT. The authors of [47] have proposed a deep reinforcement-learning-based strategy for securing the mobile edge computing systems against interference and jamming attacks. Similarly, Ref. [48] have used reinforcement learning for the development of a secure data collection strategy in the domain of IoT. The reviewed literature demonstrates the potential of ML approaches for developing security solutions. Table 1 presents the summary of recent literature related to the IoT security and attack detection.

Table 1. Summary of recent literature.

Ref.	Attack	Countermeasure	Dataset for Model Development	Limitations/Gaps
[12]	Protocol-specific attack	LGBM is leveraged to address protocol-specific attacks in RPL-based IoT	Self-generated	SN-inherited attacks are not considered
[14]	Protocol-specific and SN-inherited attacks	GAN-C is used to address rank, version, and hello flood attacks	IRAD	High processing and computational power required
[13]	SN-inherited attack	GRU-DL is used to counter hello flood attacks in RPL-based IoT	Self-generated	Protocol-specific attacks are not considered, DL methods require high processing and computation power
[49]	SN attacks	Generative DL using adversarial autoencoder and bidirectional generative adversarial network models	IoT-23 dataset	Protocol-specific attacks are not considered, SN-inherited attacks in RPL-based IoT setting are not considered
[50]	SN attacks	IoT device classification (binary and multiclass) using the logistic decision tree model which is based on logistic regression and decision tree methods	IoT device dataset primarily generated by authors and secondary dataset obtained from [51]	SN-inherited attacks in RPL-based IoT are not considered, protocol-specific attacks are not considered
[52]	SN attacks	Long short-term memory and gated recurrent unit-based DL framework for intrusion detection	CICIDS2017 dataset [53]	Protocol-specific attacks are not considered, SN-inherited attacks in RPL-based IoT setting are not considered
[54]	SN attacks	IDS for attack detection based on binary and multiclass DL-based classification using long short-term memory model	UNSW-NB15 and Bot-IoT datasets	SN-inherited attacks in RPL-based IoT are not considered, protocol-specific attacks are not considered
[55]	SN attacks	Graph-based botnet detection system for classification of attacks using Naïve Bayes, decision and extra trees, random forest, AdaBoost, and k-Nearest Neighbors (k-NNs)	CTU-13 and IoT-23 datasets	Protocol-specific attacks are not considered, SN-inherited attacks in RPL-based IoT setting are not considered
[56]	SN attacks	DL recurrent-neural-network-based IDS and SL classifiers for attack classification	NSL-KDD dataset	SN-inherited attacks in RPL-based IoT are not considered, protocol-specific attacks are not considered, relatively old dataset is used

From the review of the literature, it is identified that the PS-R and SN-W attacks are active and internal routing attacks, while an SN-W attack can also be launched using resources external to the network. Each of them belongs to one of the two attack classes that the RPL-based IoT is vulnerable to. Both attacks cause severe security issues and are capable of instigating other attacks. They may collaborate with each other to increase the attack impact and occur concurrently in an RPL-based IoT network causing twofold or multifold deterioration of the IoT network, RPL mechanism, and routing resources which are naturally constrained. Table 2 presents the routing attacks considered in this paper along with their impact on Confidentiality, Integrity, Availability (CIA) security triad and effects on the IoT network.

Table 2. PS-R and SN-W attacks, their impact on CIA triad and network performance.

Attack	Description	CIA Triad Impact	Impact on Network Performance
PS-R attack	The attacker node broadcasts fake credentials for attracting traffic, such as fake rank value, and shortest path to the sink node	Confidentiality, integrity	Resource consumption, network destabilization, end-to-end delay,
SN-W attack	Malicious nodes form a connection and attract network traffic toward themselves by illegitimately displaying increased child and neighbor nodes using their path to communicate with the sink node. They do this by replaying the same messages at short intervals. Secondly, the malicious nodes selectively or completely drop the incoming victim packets	Confidentiality, availability	Increase in the illegitimate routing of packets, network, partial or complete packet loss

Since PS-R attack is capable of altering the RPL components, while also affecting the network performance, forming routing loops, and disrupting the topology, it is considered as one of the gravest attacks. Therefore, PS-R attack is addressed in this research. Similarly, SN-W is also a highly damaging and detrimental attack which has a high potential to instigate other security attacks as well on an IoT network. Therefore, both of these attacks are addressed using ML approaches. The proposed ProSenAD model to handle these attacks is discussed in the next section, followed by results, comparison with benchmarks, and conclusions in the subsequent sections.

3. Methodology

The Sample, Explore, Modify, Model, Assess (SEMMA) process model is used to explain the ProSenAD model development phases. These phases are discussed in detail in the forthcoming subsections.

3.1. SEMMA Process Model for ProSenAD Model Development

SEMMA process model is adopted to define and explain the design and development stages of the ProSenAD model. Figure 2 illustrates the process model. The Sample stage in SEMMA includes data collection methods, functions, tools, and techniques used for gathering the required data from specific sources. The Explore stage of SEMMA is responsible for exploring the collected data and performing preliminary analysis to discover trends and patterns. In a labeled dataset, this stage helps in identifying the data points, instances, features, variables or attributes, and target classes. Descriptive statistical methods are used to find the percentage of data instances from different angles depending on the requirements analysis. In the Modify stage of SEMMA, the raw dataset is prepared, features are engineered, and selected. The data is cleaned by addressing missing values and duplicate features. Data points with incomprehensible values are replaced with suitable ones. For example, in this case, categorical features are converted to numerical using one hot encoding method. Next, the Model stage of SEMMA involves the baseline model development, parameter and hyperparameter tuning, ProSenAD optimization, and model training. This step is followed by the Assessment phase of SEMMA where the model performance is assessed using performance evaluation metrics. The model is then compared with the selected benchmarks.

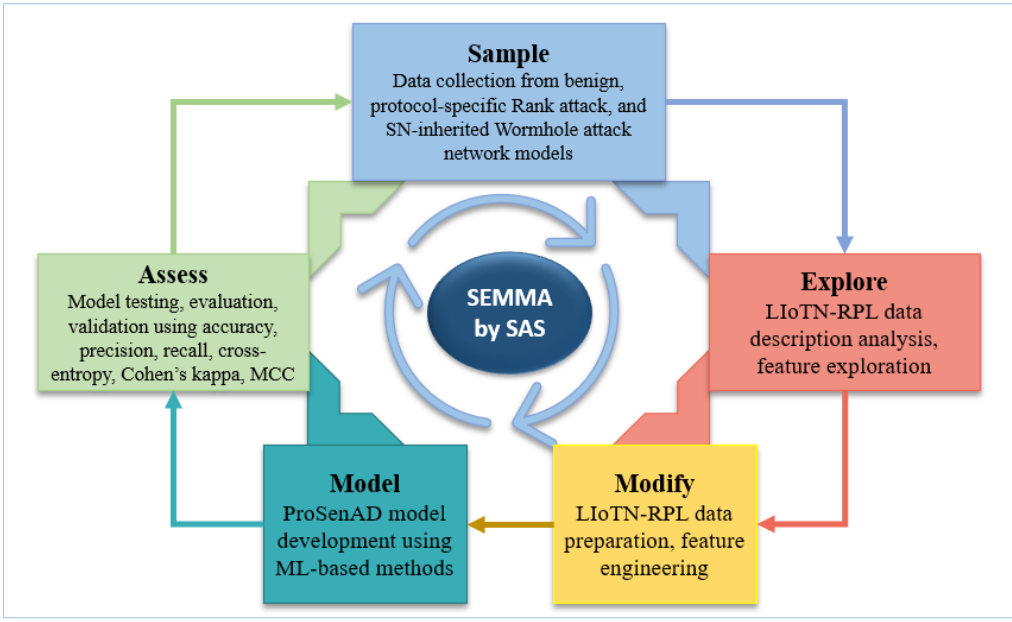

Figure 2. SEMMA process model adopted for ProSenAD development.

3.2. Sampling Phase in SEMMA

In the Sample stage of the SEMMA process model, the data collection procedure is performed. Various network models are designed and implemented in the Contiki Cooja simulator, which is designed for simulating the resource-constrained IoT networks with RPL enabled for data communication and routing. The next subsections explain the data generating models developed in this study which are broadly classified into the Normal Traffic Network Model, PS-R Attack Model, and SN-W Attack Model.

Due to the scarcity of publicly available protocol-specific and SN-inherited data, researchers generate the datasets using simulations and capture the packets using the 6LoWPAN analyzer. It is a frequent practice observed in literature. For example, Ref. [12] have created network models depending on the requirement of the attack dataset and simulated the network scenarios for data collection. Therefore, similar practice is followed for data collection in this paper using the Cooja simulator. The network models of interest are designed and implemented in Cooja, which include, normal traffic network model, PS-R attack network model, and SN-W attack network model. The network size, distance coverage, and node positioning parameters considered in this paper are supported through relevant literature, a summary of which is given in Table 3.

The first attack model is designed to simulate the PS-R attack, where the attacker nodes relay false rank values in addition to ideal conditions as a prospective parent using DODAG Information Object (DIO) messages to attract the child nodes. The attacker nodes are deliberately implemented in close proximity to the sink to speed up the parent selection process and for recording the attack effects. The malicious node initiates with a delay of 60 s after the network is stabilized. The victim nodes consider the attacker as a parent and eventually joins it. In the second attack model, i.e., SN-W attack, two nodes create a tunneling effect logically by exploiting the control messages. They probe the neighbor nodes using the DODAG Information Solicitation (DIS) control message to join a node and initialize the traffic transmission process. Other control messages including DODAG Information Object (DIO), DODAG Advertisement Object (DAO), and DAO acknowledgment are used to illegitimately present the shortest path towards the root node to attract the victim nodes. A similar strategy to the previous attack is used in this attack as well, in terms of attacker node placement, where they are placed near the sink and victim nodes for speeding up the attack process. The closeness of malicious nodes to the sink appears favorable to the victim nodes and, as a result, they join the attacker parent(s). The

effects of the two attacks on the network resources and routing mechanism are presented in the previous section.

Table 3. Network model simulation parameters.

Ref.	Simulator	No. of Nodes	Range	Node Positioning	Sink Node Placement	Simulation Time (s)
Our work	Contiki Cooja	1st case: 1 sink node, 19 sender nodes 2nd case: 1 sink node, 49 sender nodes	Transmission range: 50 m Interference: 100 m	Random and grid	central	3600 s
[57]	Cooja	1 sink node, 10 sender nodes	Transmission range: 50 m Interference: 100 m	Random	Central	3600 s
[58]	Cooja	1 sink node, 20–100 sender nodes	Transmission range: 100 m	Random and grid	Central	900 s
[59]	Cooja	1 root node, 50 sender nodes	Transmission range: 50 m Interference: 60 m	Random	Central	1500 s
[12]	Cooja	1st case: one sink node, 11 sender nodes 2nd case: one sink node, 12 sender nodes 3rd case: 1 root node, 23 sender nodes	Transmission range: 50 m	Random	Central	600 s
[60]	Cooja	1 root node, 60 sender nodes	Transmission range: 50 m	Bidimensional grid positioning with a uniform distance of 30 m	-	3600 s

3.3. Exploration Phase in SEMMA

Data generation, feature segmentation, feature extraction, and data collection steps are performed in this phase. The data is generated using Cooja, which is sent to the Wireshark through 6LoWPAN analyzer. The network traffic is analyzed in the Wireshark and feature segmentation is also performed. The data is collected for one benign and two attack scenarios. Figure 3 shows the raw data sample collected through Wireshark. The benign and attack datasets are consolidated in the LIoTN-RPL data pool and analyzed to understand the features and their relationships with the target columns.

The feature extracting unit manages the extraction of broadly related features from the packets and IoT nodes after fragmentation in the Wireshark. To assure the coverage of all the associated attributes from the network models in RPL-based IoT, the data from each use case is collected, which has contributed towards a novel and diverse LIoTN-RPL dataset. The features are extracted based on the main protocols operating in the network, which are ICMPv6, UDP, and IEEE 802.15.4. The associated features are presented in Figures 4–6.

The pivot table analysis shows that 3259 data points belong to the benign network scenario, 6213 to the PS-R attack scenario, and 4408 to the SN-W attack scenario. The descriptive statistics of the protocol-based feature set, data points belonging to each protocol, and their percentages are shown in Figures 7 and 8.

The figures show that "ICMPv6" filter retrieves five primary feature classes with thirty-nine features in total, "UDP" filter retrieves six feature classes with a total of twenty-three features inclusive of the features from other protocols, while "IEEE 802.15.4" has two feature classes with a total of nineteen features. Figure 8 indicates that the percentage of total data points is 41.2, 36.7, and 22.1, respectively.

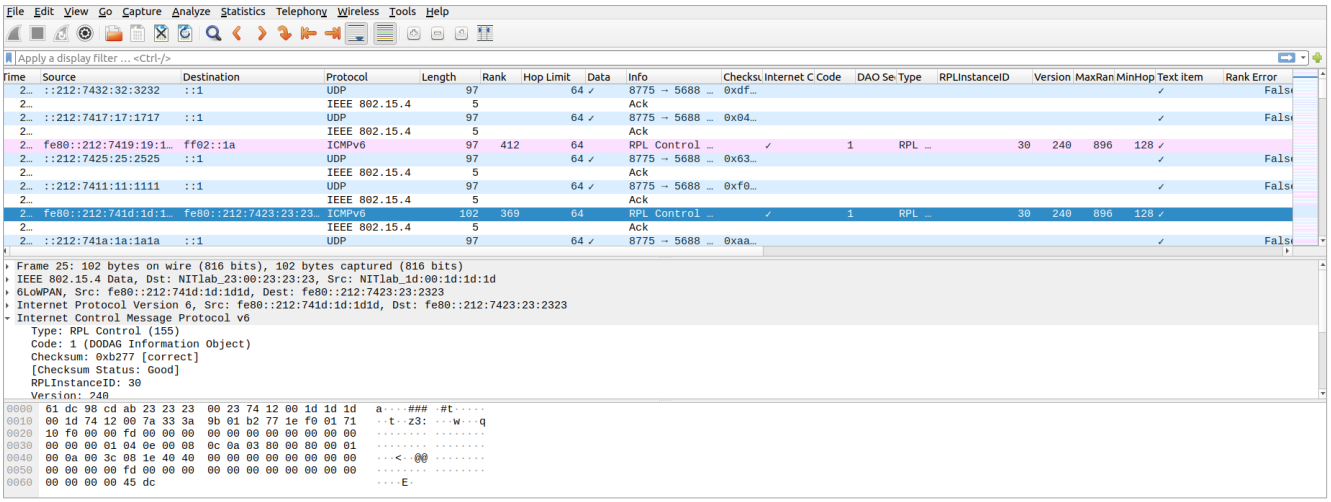

Figure 3. Network traffic data collection.

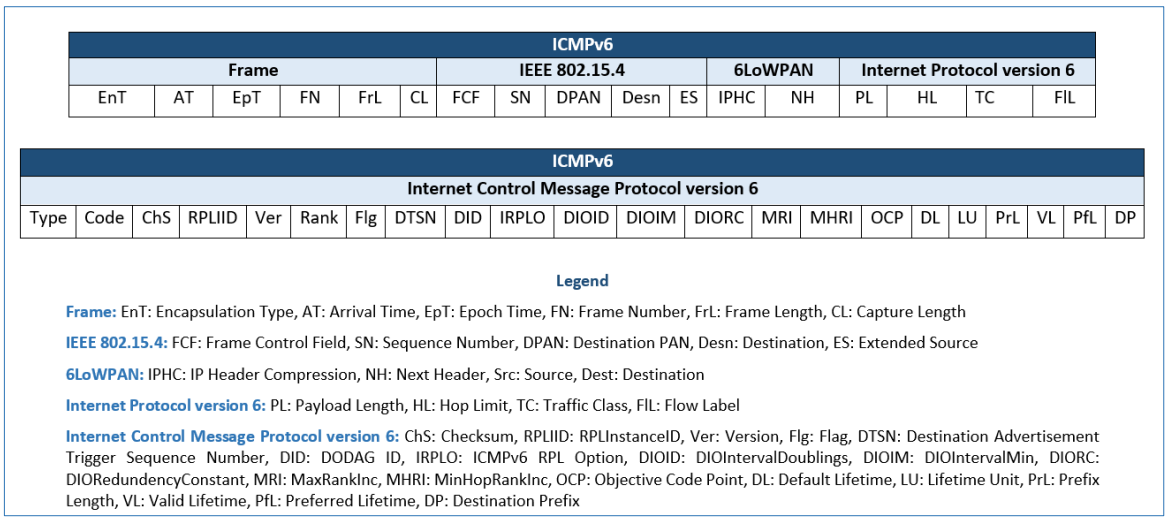

Figure 4. Dataset columns with ICMPv6 filter-related features.

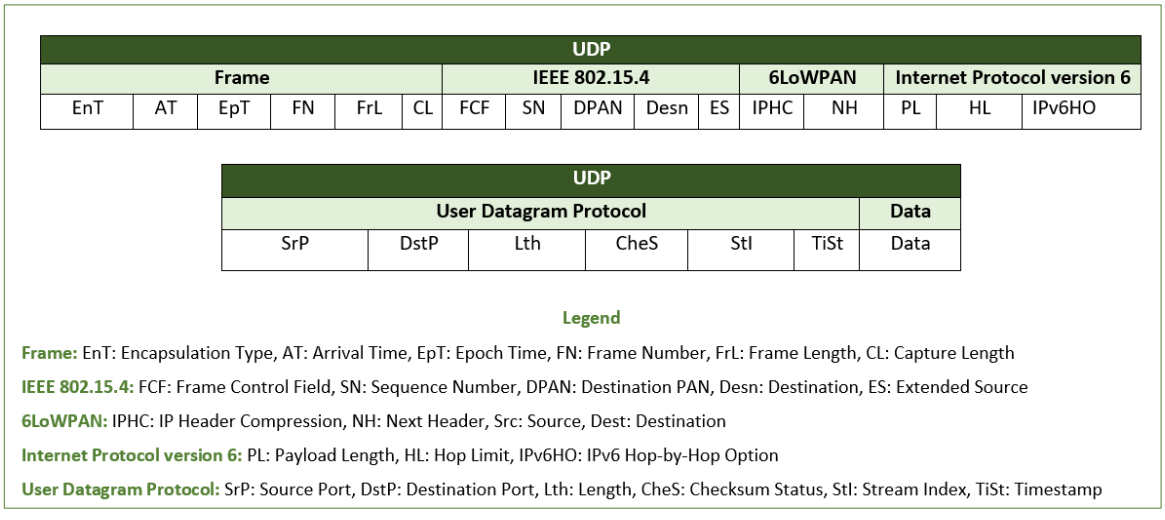

Figure 5. Dataset columns with UDP filter-related features.

IEEE 802.15.4																		
Frame						IEEE 802.15.4 Ack												
EnT	AT	EpT	FN	FrL	CL	FCF											SN	FCS
-						FrT	SecE	FrP	AckReq	PANIDC	Res	SNS	IEP	DAM	FrVer	SAM	-	

Legend

Frame: EnT: Encapsulation Type, AT: Arrival Time, EpT: Epoch Time, FN: Frame Number, FrL: Frame Length, CL: Capture Length

IEEE 802.15.4 Ack: FCF: Frame Control Field, SN: Sequence Number, FCS: Frame Check Sequence

FCF: FrT: Frame Type, SecE: Security Enabled, FrP: Frame Pending, AckReq: Acknowledgment Request, PANIDC: PAN ID Compression, Res: Reserved, SNS: Sequence Number Suppression, IEP: Information Elements Present, DAM: Destination Address Mode, FrVer: Frame Version, SAM: Source Addressing Mode

Figure 6. Dataset columns with IEEE 802.15.4 filter-related features.

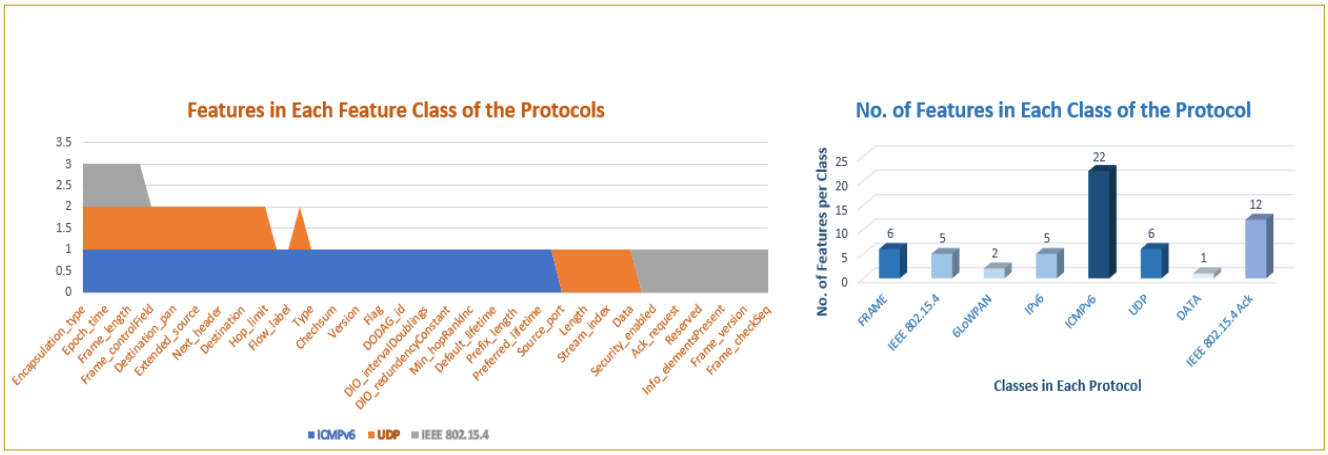

Figure 7. Features in each class of the protocol.

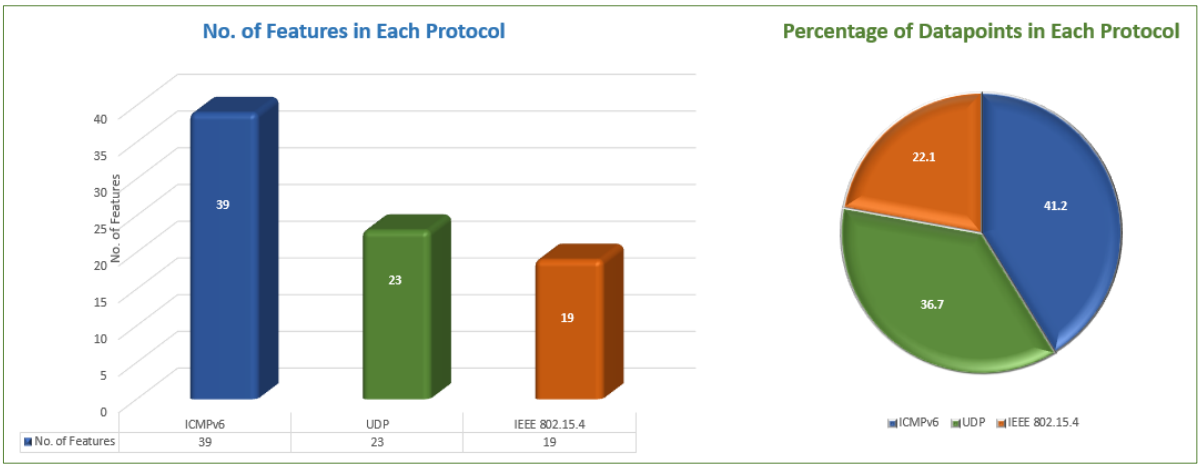

Figure 8. Number of features and datapoint percentage in each protocol.

Figure 9 presents the instance percentage of each feature class in the protocol-based traffic collected and extracted through Wireshark in the Contiki Cooja 3.0 simulator. It shows that there are five classes in ICMPv6 protocol. The fifth feature, that is, ICPMv6, encompass a little over half the percentage of all the classes while the remaining classes have similar features spreading across, except for 6LoWPAN, which is comparatively less at about five percent of the total features. For the UDP classes, the feature distribution is higher in three classes, while the Data class has the lowest percentage among all others. In the IEEE 802.15.4 class, the ratio is sixty-seventy to thirty-three.

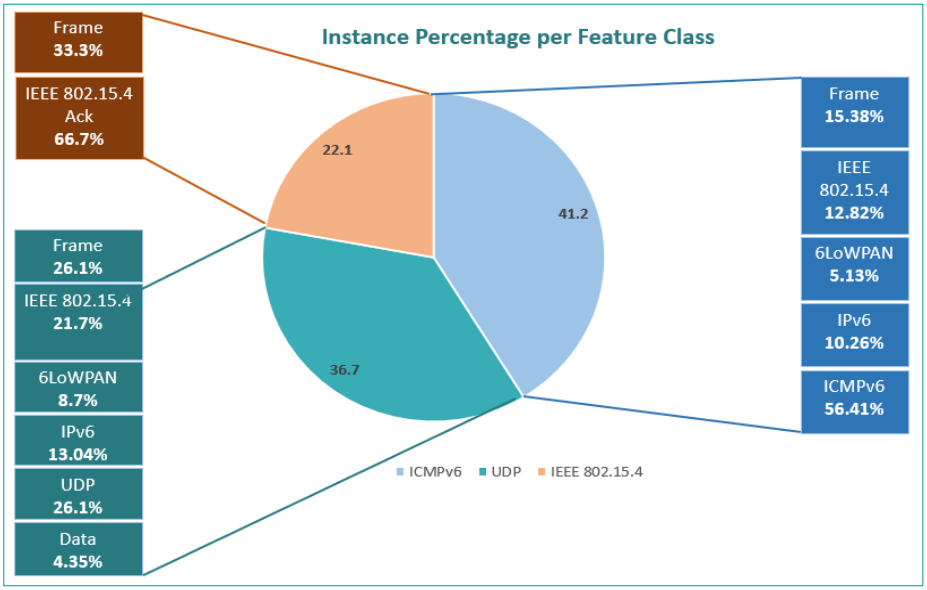

Figure 9. Percentage of instances per feature class.

The analysis reflects that ICMPv6-related features are important as they reveal information about the control messages in RPL-based routing happening in the IoT network. In the next section, the data is prepared for feature selection to build the ProSenAD model for addressing the routing attacks in RPL-based IoT networks.

3.4. Modification Phase in SEMMA

After the data collection process is completed, the raw data needs to be shaped into a useful dataset that can be employed for model development in the Model stage of SEMMA. For this purpose, it is necessary to perform certain requisite steps. In the ML paradigm, they are known as data preparation, also referred to as data preprocessing, and feature engineering. Several tools and techniques are available that can be used for preparing the dataset. It primarily depends on the requirements, the nature of the data, the objective to be achieved, and the expected outcome. In this paper, the research objective is to classify three classes, and detect two types of routing attacks, which are reflected in the network traffic. Therefore, it is necessary to include benign network traffic as a baseline for observation of normal traffic transmission and its characteristics. This factor indicates that there are at least three target classes in the data, which are benign network traffic class, PS-R attack class, and SN-W attack class. To prepare and engineer the dataset, it is important to address issues such as structural formatting, missing or incomplete records, and feature repetition. These issues can be addressed using certain standard practices observed in the literature, which include data cleaning, feature selection, feature engineering, and dimensionality reduction. Since the data preparation step is dependent and highly specific to the dataset and the project, it makes the process straightforward. The tasks performed in the previous section are foundational and pivotal for performing the data preparation and feature engineering in the next steps.

The previous section stipulates that the data collected in LIoTN-RPL data pool requires cleaning in terms of missing record values, repetitive features, and other aforementioned issues. To address these issues, Python on Anaconda [61] has been used. It is a package management framework that accommodates various libraries, software notebooks, programming environments, and data science-related tools. The data is loaded to the Jupyter notebook [62], provided by the Anaconda framework. It is visualized through this notebook, and a Python data analysis library called 'pandas' [63] is used for preprocessing. For instance, the missing values are replaced with 'NaaN', the repetitive features are removed,

and categorical features are converted using one-hot encoding. The selected features and their descriptive statistics are presented in Table 4.

Table 4. Descriptive statistics of features in the LIoTN-RPL dataset.

Instance		Number/Value		
		Normal	**PS-R**	**SN-W**
Source ID		3259 instances	6213	4408
Destination ID		3259 instances	6213	4408
Protocol	IEEE 802.15.4	2991	241	3324
	UDP	1980	1314	2152
	ICMPv6	1279	4899	2256
Rank	Number of rank values observed	114	172 distinct values observed in the dataset	Number of rank values observed 176
Message	DODAG Information Object (DIO)	816	1424	1218
	DODAG Information Solicitation (DIS)	-	Not considered	23
	DODAG Advertisement Object (DAO)	463	3475	1015
	Acknowledgment (Ack)	2991	241	3324
	UDP	1980	Not considered	2152
MaxRankInc	The threshold set for a maximum rank increase for stable network performance and controlled DIO	Normal value: 896, 816 values in the normal traffic dataset	Value in attack scenario is set to 0 Normal traffic dataset showed it to be set to 896 in the simulated network model Total values observed in the rank attack traffic dataset: 1424	The threshold set for a maximum rank increase for stable network performance and controlled DIO Normal value: 896, 1225 values in the normal traffic dataset
MinHopRankInc	Minimum hop rank increase for stable network performance and for avoiding loops	Normal value: 128, 816 values observed in the normal traffic dataset	Value in attack scenario is set to 0 Normal traffic dataset showed it to be set to 128 in the simulated network model Total values observed in the rank attack traffic dataset: 1442	Minimum hop rank increase for stable network performance and for avoiding loops Normal value: 128, 1225 values observed in the normal traffic dataset
RErr	Rank error, RPL's inherent method to detect any errors in the rank values	True: 3 False: 1976 Total: 1979	True: 0 False: 1315	Not considered
DIOIntervalMin	The interval threshold set for sending DIOs	The normal threshold value is observed to be 12 in the simulated network, and a total of 816 values are observed in the normal traffic dataset	Threshold value observed to be decreased during the attack: 7 Occurrences: 1424	Not considered

Table 4. *Cont.*

Instance		Number/Value		
		Normal	**PS-R**	**SN-W**
DIORedConst	The DIO redundancy constant set for controlling the redundant DIOs and maintaining a stable network	The threshold value is set to 10 in the simulated benign network, and a total of 816 values are found in the normal traffic dataset	Threshold value observed to be increased during the attack: 15 Occurrences: 1424	Not considered
Lost	Lost packets during the attack simulation	Minimum calculated in normal traffic	Not considered	610 instances are observed in the SN-W attack traffic dataset, and they are denoted by 1
Hop count	Smallest hop count broadcasted by malicious nodes	Average calculated in normal traffic	Not considered	1244 instances are observed in the SN-W attack traffic dataset, and they are denoted by 1
Class label	Normal	0	Not applicable	Not applicable
	PS-R	Not applicable	1	Not applicable
	SN-W	Not applicable	Not applicable	2

3.5. Modeling Phase in SEMMA

This section introduces the development of the proposed ML-based model, ProSenAD, for protocol-specific and SN-inherited attack detection in RPL-based IoT networks. From the learning paradigms, the type and characteristics of the collected data identified through statistical analysis, research objectives, and consideration for the nature of the RPL-based IoT network, multiclass classification is adopted for attack detection through the ProSenAD model. ML-LGBM [12] is leveraged and optimized for multiclass classification tasks in the proposed ProSenAD model. The optimization elements are discussed later in this section. The benchmark research and the ML model variants are used for comparative analysis of the performance using carefully selected evaluation metrics. To make sure that both categories of attacks are covered, ML-LGBM [12], GAN-C [14], GRU-DL [13] are used for benchmarking. Model variants including gradient boosting (GB), extreme gradient boosting (XGBoost), and baseline light gradient boosting machine (LGBM) are also considered for comparison. The authors of [12] have only focused on protocol-specific version attack, whereas [13] have focused on SN-inherited hello flood attacks. The authors of [14] have addressed two protocol-specific and one SN-inherited attacks. However, wormhole is not considered in [14]. Therefore, these research studies have been used as benchmarks along with the models mentioned earlier in this section.

The most important constituents of an ML-based model are its parameters and hyperparameters. In the next subsections, different types of parameter categories and metrics are discussed for ProSenAD development. They include core parameters, learning control parameters, input and output workflow parameters, objective parameters, and metric parameters.

3.5.1. Fundamental Parameters for ProSenAD

The core parameters for developing the LGBM-based ProSenAD model include the following: (1) task, (2) objective, (3) boosting, (4) number of iterations, (5) learning rate, and (6) number of leaves. These parameters are fundamental for achieving the objective of attack detection in RPL-based IoT networks through the ProSenAD model. The aforementioned parameters have several metrics that can be used for building the model. They are explained further in this section.

The LGBM is typically used for binary classification as in [12]. However, our task is to optimize it for multiclass classification. Therefore, the most suitable and fundamental core metrics from the first parameter are selected for ProSenAD. They include, train, predict/test. From the second parameter, multiclass and softmax objective function is selected following the research objective of this paper. Other metrics from the second parameter employed by the ProSenAD include the determination of multiclass classification application metrics, and the number of classes. The multiclass application metrics include direct multiclass classification and binary one-vs-all-based multiclass classification. Furthermore, the third parameter allows for the implementation of the following methods: (1) gradient-based one-side sampling, (2) dropouts meet multiple additive regression trees, (3) random forest, and (4) conventional gradient boosting decision tree. By default, LGBM employs gradient boosting decision tree for some iterations before activating gradient-based one-side sampling depending on the learning rate. However, it should be tuned with other metrics for enhanced performance. For instance, if the learning rate is set to 0.1, then LGBM will run gradient boosting decision tree for the first ten iterations internally, followed by gradient-based one side sampling starting from the eleventh iteration. Therefore, we have experimented with the aforementioned metrics and tuned them accordingly to get optimal results.

The *number of iterations* parameter consists of metrics for the number of trees, number of rounds, number of estimators, and number of iterations. The default number of trees created by the model in multiclass classification is equal to the multiple of the number of classes and number of iterations. However, it can be exploited and tuned to meet the requirements and achieve the target accordingly. The fifth and sixth parameters, that is, *learning rate* and *number of leaves*, can also be tuned based on the model performance. ProSenAD incorporates these parameters and hyperparameters for building the attack detection model for RPL-based IoT networks. The next section covers the learning control parameters considered for ProSenAD.

3.5.2. Learning Control Parameters for ProSenAD

The learning control parameters are required for optimizing the model based on the objective and the characteristics of the dataset. The values in these parameters are responsible for and control the learning process of the model. The important control parameters for ProSenAD can be classified into boosting-specific, tree-specific, and heterogenous parameters. They include those that are optimized for solving the accuracy, regularization, overfitting, and training speed-related issues. The important learning control parameters include depth of the tree ($Tdepth$), minimum data in a leaf (d_{min} in L_i), feature fraction ($F_{fraction}$) and bagging fraction ($F_{bagging}$), bagging frequency ($f_{bagging}$).

3.5.3. Input and Output Workflow Parameters for ProSenAD

The input and output workflow parameters for developing the ProSenAD model include the following: (1) dataset-related, and (2) prediction-related parameters. The dataset parameters consist of methods for selecting maximum bins, minimal data, or information in one bin, feature bundling, and specification of categorical features. These methods are used to train, tune, and optimize the model. However, because there are few categorical features in LIoTN-RPL dataset, this step is performed during preprocessing in the Modify phase of SEMMA. Other aforementioned parameters have been experimented upon to build the multiclass model in ProSenAD. The prediction-related parameters can only be used while making predictions on the dataset. They include specification of iteration from where a prediction should be initialized. The value can be set to zero or less, where zero means the prediction should be started from the first iteration. ProSenAD uses this parameter for reference and comparison purposes. Other parameters include number of iterations considered for prediction and early stopping which is specific for classification tasks in LGBM. It is set to false in prediction task to avoid possible negative impact on the accuracy. The next section discusses parameter tuning and optimization.

3.5.4. Baseline Model, Parameter Tuning, and ProSenAD Optimization

The baseline model is used for contextualization of the model results and to perform optimization with reference to the objective to be achieved. For ProSenAD model development, first the LGBM baseline model is built and trained using basic parameters and default values. The model variant-related benchmarks are also trained with multiclass objective for comparison with the proposed model. Both direct multiclass and binary class-based multiclass models are considered to be used for reference. For instance, Ref. [12] have used GOSS boosting method for binary classification of protocol-specific attack. Therefore, the same method has been used in this paper in addition to dart booster for performance comparison. Furthermore, GB, XGBoost, and LGBM with GBDT boosting methods are built. The results of the ProSenAD are presented in the next section along with the comparative performance analysis for both types of benchmarks that are considered in this research. Based on the results obtained from the baseline model, the fine-tuning step is performed for ProSenAD. The important parameters, hyperparameters, and metrics used in ProSenAD for attack detection are presented in Table 5. (3) in Table 5 means that the parameter/metric has been implemented while (7) means that it has not been implemented.

Table 5. Parameters, hyperparameters, and metrics considerations.

Parameter Class	Parameters/Metrics	Baseline	ProSenAD	Constraints
Core	Task	Training, prediction/testing	Training, prediction/testing	N/A
	Objective	Multiclassova (one-vs-all)	Multiclass	N/A
	Number of classes	3	3	N/A
	Boosting	Gbdt	Goss, dart	N/A
	Iterations	Smallest multiple of number of classes and number of iterations	✓ (default = 100)	Greater than or equal to 0
	Learning rate	0.1		Greater than 0
	Number of leaves	✗	✓ (default = 31)	Greater than 0 and less than or equal to 131,072
Learning Control	Tree depth (maximum)	✗	✓ (default = −1)	N/A
	Minimal information in one leaf	✗	✓ (default = 20)	Greater than or equal to 0
	Feature fraction	✓	✓ (default = 1)	Greater than 0 and less than or equal to 1
	Bagging fraction	✗	✓ (default = 1)	Greater than 0 and less than or equal to 1 Bagging frequency must be set to a non-zero value to use this parameter
	Bagging frequency	✗	✓ (default = 0)	Bagging fraction should less than 1 to use this parameter
	Early stopping	✗	✓ (when the best performance is observed)	N/A

Table 5. *Cont.*

Parameter Class	Parameters/Metrics	Baseline	ProSenAD	Constraints
Input and out workflow	Number of bins (maximum)	✗	✓ (default = 255)	Should be greater than 1
	Feature bundling	✓	✓	N/A
	Prediction initialization	✗	✓ (default = 0)	N/A
	Number of iterations for prediction	✗	✓ (default = -1)	N/A
Objective	Number of classes	3	3 (default = 1)	Should be greater than 0
	Sigmoid	✓ (default = 1)	✗	Should be greater than 0
Metric	Softmax/multi_logloss	✗	✓	N/A
	Cross entropy	✓	✓	N/A

Parameter tuning is performed to optimally improve accuracy, handle overfitting, regularization, and reduce the training time, because parameters and hyperparameters determine the overall performance of the model. Therefore, they are classified as: (1) parameters that contribute to improved accuracy, (2) parameters that determine and impact the structural and learning process of the tree, and (3) parameters that handle overfitting issues. These are important to consider for optimization. Therefore, in ProSenAD, each of these categories are addressed in the parameter tuning process. The first category includes learning rate (lr), number of iterations (I), and number of bins (B_n). They are tuned in inverse relation to each other to improve accuracy and avoid overfitting by experimenting with a range of values. The second category consists of structural parameters such as number of leaves (L_n), maximum tree depth (T_{depth}), and minimal information in one leaf (d_{min} *in* L_i). The last parameter of this category is important as it also contributes to resolving regularization and overfitting issues. A range between hundreds and thousands is optimal if the dataset is above ten thousand records. Therefore, this range is experimented upon to find the optimal value. The third category comprises parameters for solving regularization and overfitting problems. Some of the most important parameters that contribute to resolving these issues include feature fraction and bagging fraction. In ProSenAD, feature fraction ($F_{fraction}$), bagging fraction ($F_{bagging}$), and bagging frequency ($f_{bagging}$) are considered, where $F_{bagging}$ cannot be used without $f_{bagging}$. Therefore, these requirements are fulfilled to develop and optimize the ProSenAD. The results are presented in the next section with detailed comparative analysis through different use cases.

3.6. Assessment Phase in SEMMA

In this phase, several factors are considered for the selection of evaluation metrics to test the performance of the proposed model. One of them is the type of learning paradigm considered for developing the model. Confusion matrix parameters are used for deriving and formulating the metrics. Furthermore, the type of classification being employed is also taken into consideration. In multiclass classification, advanced parameters are required for model evaluation in addition to classic ML metrics to counter any accuracy-related bias. Therefore, Cross Entropy (CE) is used from the metrics parameters of LGBM, Cohen's Kappa (CK) for inter-rater agreement and reliability, and the Matthews Correlation Coefficient (MCC) for extensive evaluation and validation. These metrics and evaluation methods are discussed in the next section where the results are presented using different use cases to assess the model's performance.

4. Results and Discussion

In this section, the metric parameters and evaluation methods used for the evaluation of the proposed model are explained. Comparative performance analysis is also performed using different benchmarks and scenarios, which is followed by a detailed discussion of results and findings.

4.1. Performance Parameters, Evaluation and Comparison

The performance of the proposed ML-based ProSenAD is compared with existing relevant research works [12–14] and gradient boosting variants including GB, XGBoost, and LGBM based on the Classification Accuracy (CA), precision/Positive Predictive Values (PPV), recall/sensitivity, CE, CK, and MCC calculation. Multiple network model simulations are performed under two cases and scenarios to obtain the dataset which is split into training and testing set. The testing set environment is used for the performance evaluation. The cases include benign model simulation, PS-R attack simulation, and SN-W attack simulation. The data from the first case is used as a baseline to determine the other two attack cases and the attack-focused results are presented in the next subsections. PS-R attack is identified as Case A and SN-W attack is identified as Case B.

4.1.1. Classification Accuracy

The first metric used for evaluation of the ProSenAD model is classification accuracy (CA). Although accuracy is prone to biased results in some cases, specifically in multiclass classification and unbalanced dataset, it is important to use in evaluation as it forms a frame of reference for comparison of results with benchmarks and other evaluation metrics. Equation (1) presents the equation for accuracy calculation, where *TP* = *True Positive*, *TN* = *True Negative*, *FP* = *False Positive*, and *FN* = *False Negative*.

$$\frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

The accuracy shows the value of correctly predicted data instances among all the available instances of benign, and attack classes. Figure 10 shows the average classification accuracy of ProSenAD, GAN-C, ML-LGBM, GB, XGBoost, and LGBM, comprising data from the RPL-based IoT environment. ProSenAD outperforms GAN-C by 8.7% in protocol-specific attack detection, GB by 10.6% in protocol-specific attack detection and 12.2% in SN-inherited attack detection. It exceeds in classifying the target classes by 7% in comparison with the XGBoost where the former's classification accuracy is 0.997 and latter's is 0.927. As for the SN-inherited attack detection scenario, ProSenAD outperforms XGBoost by 7.7% indicating better overall performance in both attack use cases. Baseline LGBM shows an accuracy of 0.952, which is 4.5% less than ProSenAD for protocol-specific attack scenario. As for the SN-inherited use case, the proposed model exceeds the LGBM performance with accuracy improved by 5%. Although ML-LGBM and ProSenAD show similar performance in protocol-specific use cases, the former is not designed for SN-inherited attack detection. Therefore, ProSenAD outperforms in terms of number and simultaneity of attacks for detection. GAN-C is also designed for only one attack scenario; the comparative performance for this method is presented earlier. Similarly, GRU-DL shows negligible difference in performance, but it also works only in the SN-inherited attack detection scenario. It does not address protocol-specific attacks; thus, ProSenAD outperforms the benchmarks, overall.

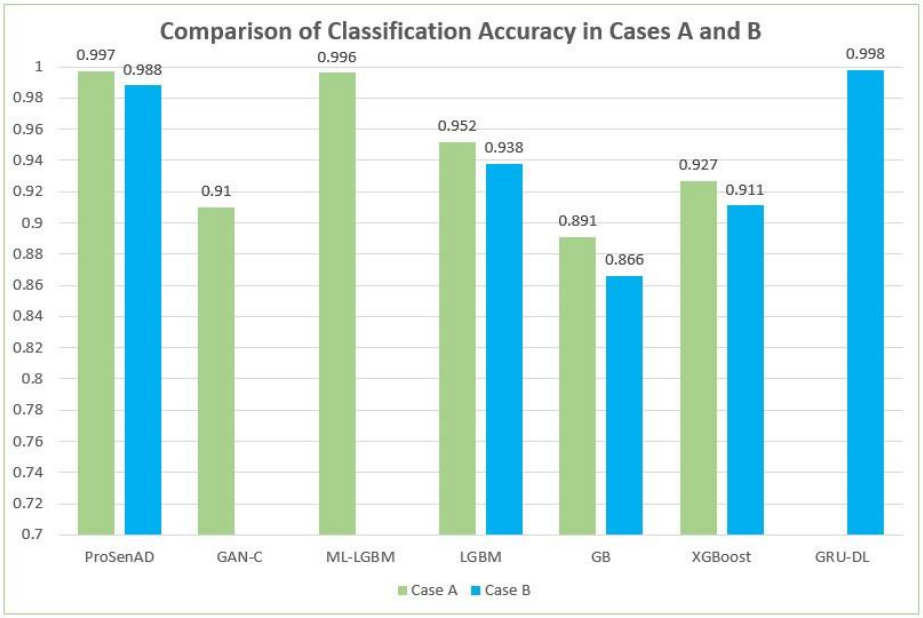

Figure 10. Comparison of classification accuracy.

4.1.2. Precision

The second metric used for ProSenAD performance evaluation is precision, also known as PPV. It is a more scrutinized evaluation metric as compared to the CA. It calculates the correctly predicted positive instances over all positive instances predicted by the model, which can also be explained as the quantification of accurate classification of positive class using the *TP* and *FP* parameters. Equation (2) is typically used to calculate the precision of a model.

$$\frac{TP}{TP + FP} \tag{2}$$

Figure 11 shows the average precision/PPV results of ProSenAD, GAN-C, ML-LGBM, GB, XGBoost, and baseline LGBM. GAN-C shows the highest difference in performance in the protocol-specific attack scenario, with a precision of 0.84. The proposed model outperforms it by 15%. Next, GB performs the poorest among all benchmarks (except for the GAN-C in protocol-specific attack detection) with an average precision of 0.883 in protocol-specific attack scenario and 0.853 in SN-inherited attack scenario. ProSenAD shows improvement in both scenarios with an improved performance of 10.7% and 12%, respectively, for this benchmark. Next, the XGBoost shows an average precision of 0.905 in the first attack scenario, and 0.893 in the second attack scenario. ProSenAD outperforms it by 8.5% and 8%, respectively. The proposed model also shows better PPV in comparison with the LGBM for both attack scenarios, with an increase of 5.3% for protocol-specific attack detection and 5.1% for SN-inherited attack detection. The difference between the performance of the proposed model and ML-LGBM is almost similar, but the latter is designed only for the detection of one type of attack, i.e., the protocol-specific attack. It does not address SN-inherited attacks. Therefore, ProSenAD outperforms it in terms of number and simultaneity of attack detection. Overall, the proposed model shows improved performance as shown through the comparative analysis.

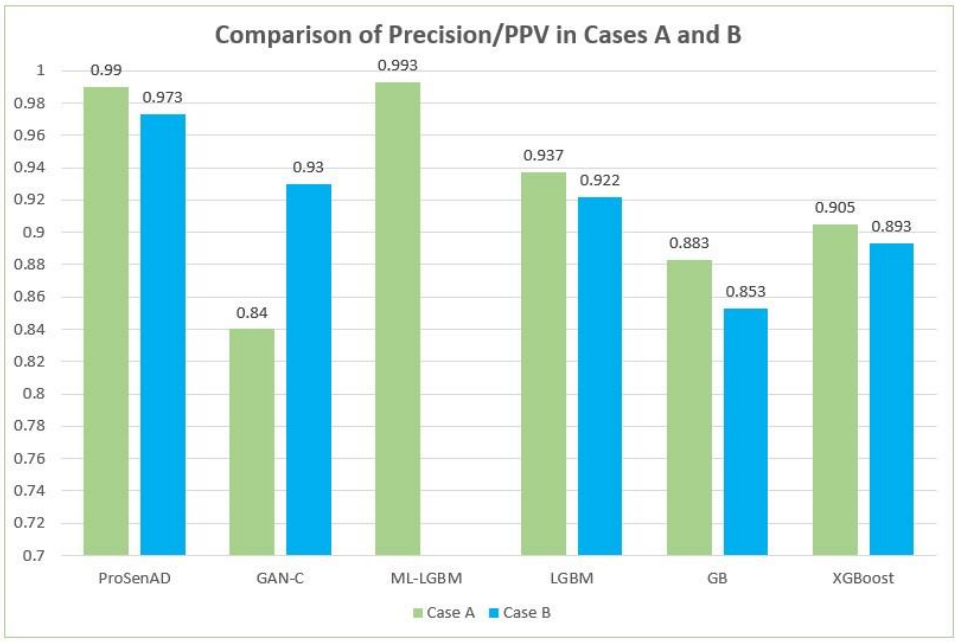

Figure 11. Comparison of precision/PPV.

4.1.3. Sensitivity

The third evaluation metric used for ProSenAD performance measurement is recall. It is also known as sensitivity, which is the ratio between the number of attacks detected by the model and the total attacks present in the dataset. Equation (3) is used to calculate the sensitivity/recall.

$$\frac{TP}{TP + FN} \tag{3}$$

Figure 12 shows the average recall/sensitivity results of ProSenAD, GAN-C, ML-LGBM, GB, XGBoost, and baseline LGBM. The comparative analysis shows that proposed model outperforms the approaches used as benchmarks. For instance, it shows an increased average recall for protocol-specific attack scenario in comparison with GAN-C where the difference between their results is 16.1%. SN-inherited attack scenario also shows that the ProSenAD performs well with a 4.2% increase in recall when compared with the same approach. Next, the comparison with GB in the first scenario shows the proposed model's high performance with a 13% improved recall, followed by 16.8% improvement of the proposed model when compared with GB in the second attack detection scenario. XGBoost performs with a 0.837 recall in the first scenario and 0.844 recall in the second scenario, as compared to ProSenAD which performs better with a recall of 0.981 in the first attack case and 0.962 in the second attack case, indicating a 14.4% and 11.8% increase in the comparative performance, respectively. Next, the LGBM shows an average recall of 0.927 in protocol-specific attack detection, and 0.909 recall in SN-inherited attacks detection. ProSenAD outperforms these by an increase of 5.4% and 5.3% in the sensitivity, respectively. The performance comparison of ML-LGBM and the proposed model shows a 0.9% increase in recall of the latter. Although the difference is small, ProSenAD performs better in terms of the number of attacks that it detects as compared to ML-LGBM.

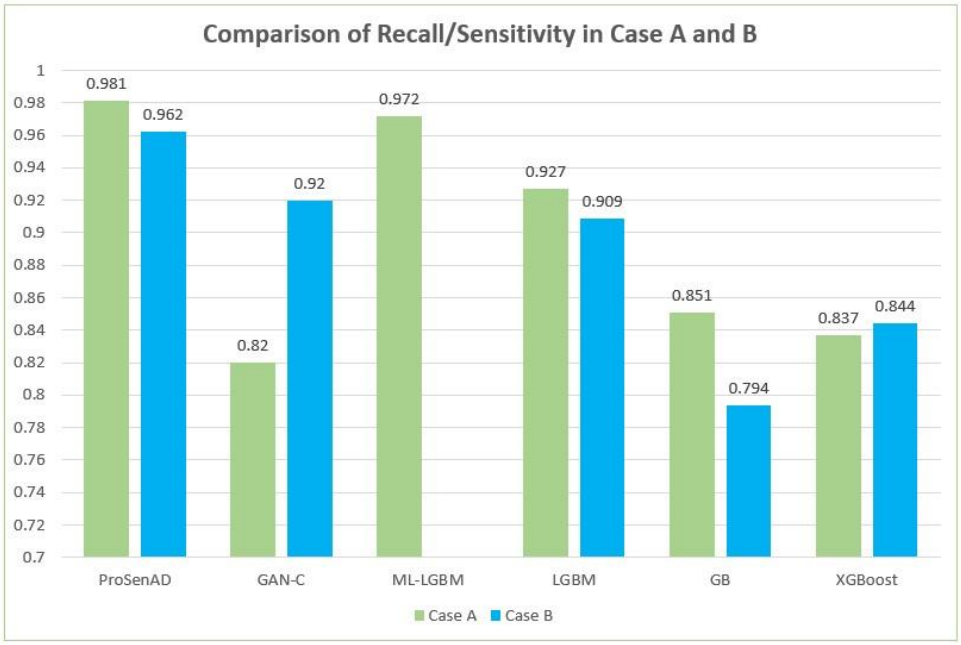

Figure 12. Comparison of recall/sensitivity.

4.1.4. Cross Entropy

CE is used to measure cross entropy loss in ML-based models. The values between the range of 0 and 1 are used to measure the performance. The lower the value, the better a model performs. It is useful to employ this metric for model evaluation in multiclass classification cases or when the dataset is unbalanced to avoid accuracy bias. Figure 13 presents the cross entropy of the proposed model. The comparison of ProSenAD shows that it outperforms other approaches. GB shows the poorest results with a value of 0.591, which means that the cross-entropy loss is greater in GB as compared to ProSenAD. This is also the case with XGBoost and LGBM where cross entropy is 0.409 and 0.276, respectively. The results indicate that the ProSenAD performs better as compared to the benchmarks.

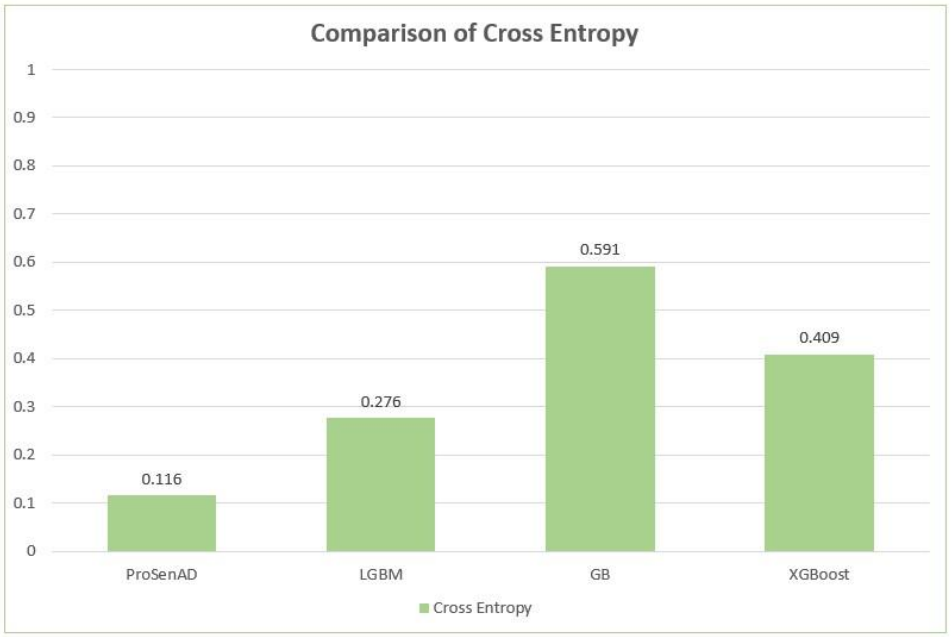

Figure 13. Comparison of cross entropy.

4.1.5. Cohen's Kappa

CK is a statistical evaluation metric which is used to measure the inter-rater agreement and reliability. It considers TP, TN as agreement values, and FP, FN as agreement by chance or disagreement values for evaluating a model within the range of -1 and 1. The closer the value is to 1, it indicates that the better the model has performed. CK is represented by κ. The equation to calculate CK is presented below (Equation (6)) where p_0 and p_e can be calculated using the confusion matrix parameters through Equations (4) and (5), respectively.

$$p_0 = \frac{TP + TN}{N} \tag{4}$$

where N is equal to the sum of all the observations.

$$p_e = \frac{z_1 \times y_1}{N^2} + \frac{z_2 \times y_2}{N^2} + \frac{z_3 \times y_3}{N^2} \tag{5}$$

where z_1, y_1, z_2, y_2, z_3, and y_3 are derived from the confusion matrix presented in Figure 14.

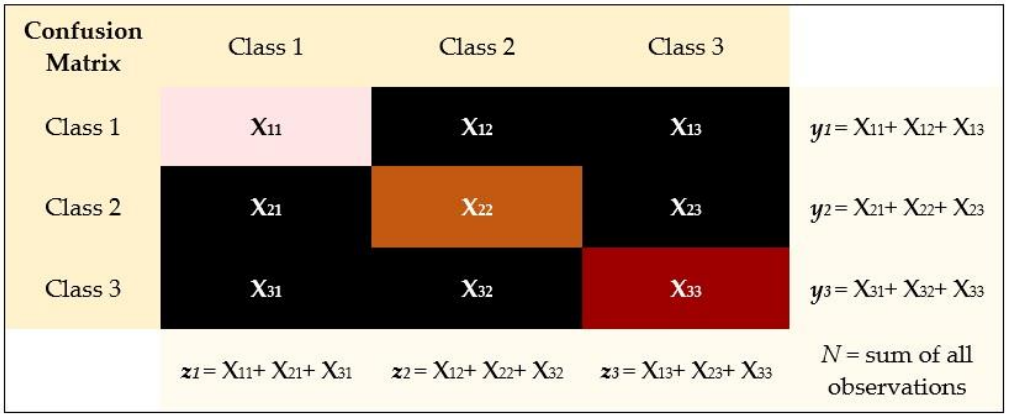

Figure 14. Derivation of variables for CK calculation.

A value less than 0 indicates that there is no agreement between the actual classes and the predicted classes, while a range of 0–20 shows imperceptible agreement. The ranges between 21–40, 41–60, 61–80, 81–99 indicate fair, moderate, substantial, and excellent agreement, while a value of 1 is designated for perfect performance of the model.

$$\kappa = \frac{p_0 - p_e}{1 - p_e} \tag{6}$$

Figure 15 shows the performance of ProSenAD in terms of kappa coefficient. The comparison indicates that the proposed model performs better than the benchmarks. GB shows a value of 0.827 in the graphical representation of results which indicates it gaining the poorest results among all methods used for comparison, while LGBM performs better than XGBoost. However, ProSenAD outperforms LGBM with a value of 0.95.

4.1.6. The Matthews Correlation Coefficient

MCC in another metric used to further evaluate the model in addition to simple accuracy. It utilizes the parameters in confusion matrix to evaluate the model, and specifically considers true negatives, unlike other metrics. This metric is useful to apply when the classification is multiclass type, or the data is unbalanced. Therefore, it is employed to further evaluate the ProSenAD. Equation (7) is used to calculate the MCC.

$$\text{MCC} = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{7}$$

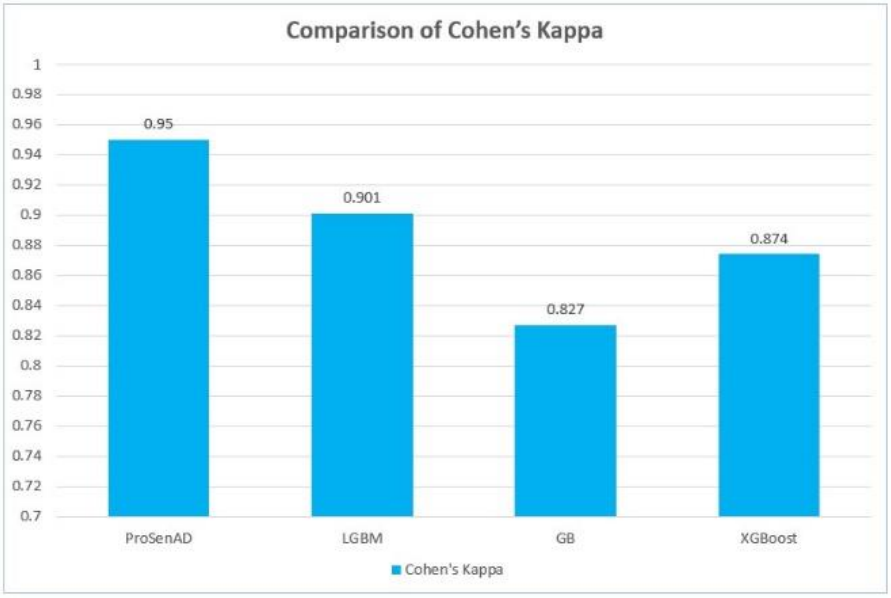

Figure 15. Comparison of Cohen's kappa.

Figure 16 shows the MCC results of ProSenAD, LGBM, GB, and XGBoost, in RPL-based IoT environment for protocol-specific and SN-inherited attack detection. The comparative analysis shows that the proposed model outperforms the approaches used as benchmarks. For instance, the comparison with GB shows the proposed model's high performance with an 18% improved MCC. XGBoost performs with 0.825 MCC value, as compared to ProSenAD. The latter performs better with 0.942 MCC, indicating an 11.7% increase in the comparative performance. Lastly, the LGBM shows an MCC value of 0.893. ProSenAD exceeds in performance by 4.9% based on this metric when compared to LGBM. Overall, the proposed model outperforms the benchmarks with an average improvement of 11.5% based on MCC evaluation metric.

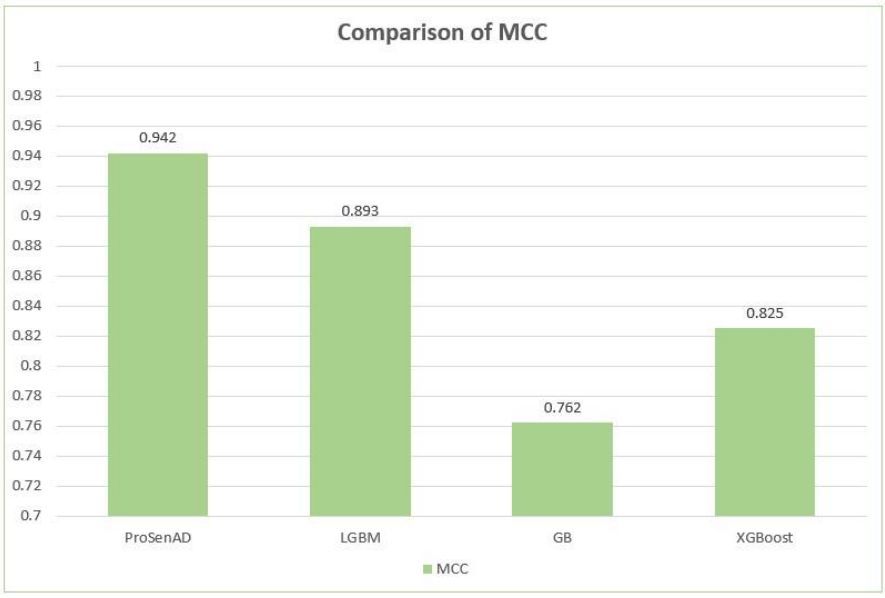

Figure 16. Comparison of MCC.

4.2. Discussion

The growing implementation of IoT-enabled applications, devices, and smart systems emphasizes the need for enhancing the security of these networks. In this research paper,

the focus is on RPL-based IoT networks, particularly, the detection of PS-R and SN-W attacks. The ML-based approach is selected because of the various advantages that it provides such as automation, pattern recognition, and feature extraction as well as selection of useful features. Based on the critically chosen parameters from different parameter categories, including the core, learning control, input and output workflow, objective, and metric parameters, an ML-based model is designed and developed. The performance evaluation is carried out using the testing set of the proposed LIoTN-RPL dataset which plays the role of model testing environment. The proposed ProSenAD model outperforms the benchmarks and provides security against PS-R and SN-W attacks in terms of number of attacks detected, attack classification accuracy, PPV, recall, CE, reliability (CK), and MCC.

The novelty and usefulness of the proposed ML-based model is summarized, as follows:

- It considers both the categories of attacks prevalent in RPL-based IoT networks and provides simultaneity of attack detection using a multiclass approach.
- The LIoTN-RPL dataset used in the development of the model is prepared such that it allows the detection of multiple attacks from both attack categories.
- It is comparatively lightweight as compared to the other models due to the usage of lightweight boosting methods and feature bundling techniques that it employs to build the model.
- It addresses the CIA triad-related security compromises that occur due to the attacks including confidentiality, integrity due to PS-R attacks and confidentiality, availability due to SN-W attacks.
- It overcomes the accuracy bias and regularization-related issues by employing appropriate metric parameters and learning control parameters, respectively.

Overall, the performance of ProSenAD is significantly better when compared with the model-related benchmarks including GB, XGBoost, and LGBM, as well as research-related benchmarks including GAN-C [14] in terms of accuracy, precision, and recall, and ML-LGBM [12] and GRU-DL [12,13] in terms of the number and simultaneity of the attacks addressed.

5. Conclusions

The IoT is emerging in several application-specific smart devices that are used in different domains, such as healthcare, homes, cities, urban and industrial infrastructures, transport, etc. The number of IoT implementations is exponentially escalating and is expected to reach billions in the next few years. Both IoT networks and RPL are resource-constrained; therefore, the security risks are higher in RPL-based IoT networks. Several solutions have been proposed to address the security attacks using different approaches. Machine learning is one such solution domain that has gained recognition for attack detection model development due to the big data generated by IoT devices. In the existing literature, the concurrent detection of PS-R and SN-W attacks is insufficiently addressed. Therefore, to address this gap, ProSenAD model is developed based on multiclass classification approaches. This research critically analyzes the parameters important for developing such a system. Evaluation metrics are also explored for performance analysis because standard classification-related metrics such as accuracy, precision, and recall are inadequate for ProSenAD evaluation. The proposed ML-based model has outperformed the ML-LGBM and GRU-DL benchmarks in terms of the number of attacks detected, and GB, XGBoost, and LGBM, in terms of accuracy, PPV, sensitivity, CE, CK, and MCC. ProSenAD shows improved performance in both use cases, i.e., PS-R and SN-W, respectively.

For future work, the LIoTN-RPL dataset will be further diversified by simulating other prevalent attacks from the protocol-specific and SN-inherited categories. Currently, the dataset is limited to benign network traffic data and attack data from two categories of the attacks identified in the paper. We plan to evaluate the results considering additional parameters and benchmarks for the two attack categories mentioned earlier.

Author Contributions: Conceptualization, F.Z., N.Z.J., S.N.B., and N.A.K.; data curation, F.Z., N.Z.J., S.N.B., M.M., S.A., and N.A.K.; formal analysis, F.Z., N.Z.J., S.N.B., M.M., S.A., and N.A.K.; methodology, F.Z., N.Z.J., S.N.B., and N.A.K.; resources, F.Z., N.Z.J., S.N.B., M.M., S.A., and N.A.K.; supervision, N.Z.J., S.N.B., and N.A.K.; writing—original draft, F.Z., N.Z.J., S.N.B., and N.A.K.; writing—review and editing, F.Z., N.Z.J., S.N.B., M.M., S.A., and N.A.K. All authors have read and agreed to the published version of the manuscript.

Funding: Taif University Researchers Supporting Project Number (TURSP-2020/73), Taif University, Taif, Saudi Arabia.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data will be available on request.

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

References

1. Ahmad, T.; Zhang, D. Using the Internet of Things in Smart Energy Systems and Networks. *Sustain. Cities Soc.* **2021**, *68*, 102783. [CrossRef]
2. Patil, S.; Gokhale, P. Systematic Review of Resource Allocation Methods Using Scheduling for M2M (Machine to Machine Communication) in IoT Network. *Stud. Syst. Decis. Control* **2021**, *341*, 213–224. [CrossRef]
3. Javaid, M.; Haleem, A.; Singh, R.P.; Rab, S.; Suman, R. Significance of Sensors for Industry 4.0: Roles, Capabilities, and Applications. *Sens. Int.* **2021**, *2*, 100110. [CrossRef]
4. Rathee, G.; Balasaraswathi, M.; Chandran, K.P.; Gupta, S.D.; Boopathi, C.S. A Secure IoT Sensors Communication in Industry 4.0 Using Blockchain Technology. *J. Ambient. Intell. Humaniz. Comput.* **2020**, *12*, 533–545. [CrossRef]
5. Vailshery, L. Number of Internet of Things (IoT) Connected Devices Worldwide from 2019 to 2030 (In Billions). Available online: https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/ (accessed on 13 June 2022).
6. La Rosa, R.; Livreri, P.; Trigona, C.; di Donato, L.; Sorbello, G. Strategies and Techniques for Powering Wireless Sensor Nodes through Energy Harvesting and Wireless Power Transfer. *Sensors* **2019**, *19*, 2660. [CrossRef]
7. Adu-Manu, K.S.; Adam, N.; Tapparello, C.; Ayatollahi, H.; Heinzelman, W. Energy-Harvesting Wireless Sensor Networks (EH-WSNs): A Review. *ACM Trans. Sens. Netw.* **2018**, *14*, 1–50. [CrossRef]
8. Kaw, J.A.; Gull, S.; Parah, S.A. SVIoT: A Secure Visual-IoT Framework for Smart Healthcare. *Sensors* **2022**, *22*, 1773. [CrossRef]
9. Verma, A.; Ranga, V. The Impact of Copycat Attack on RPL Based 6LoWPAN Networks in Internet of Things. *Computing* **2020**, *103*, 1479–1500. [CrossRef]
10. Raoof, A.; Matrawy, A.; Lung, C.H. Routing Attacks and Mitigation Methods for RPL-Based Internet of Things. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 1582–1606. [CrossRef]
11. Liu, L.; Xu, X.; Liu, Y.; Ma, Z.; Peng, J. A Detection Framework against CPMA Attack Based on Trust Evaluation and Machine Learning in IoT Network. *IEEE Internet Things J.* **2021**, *8*, 15249–15258. [CrossRef]
12. Osman, M.; He, J.; Mokbal, F.M.M.; Zhu, N.; Qureshi, S. ML-LGBM: A Machine Learning Model Based on Light Gradient Boosting Machine for the Detection of Version Number Attacks in RPL-Based Networks. *IEEE Access* **2021**, *9*, 83654–83665. [CrossRef]
13. Cakir, S.; Toklu, S.; Yalcin, N. Rpl Attack Detection and Prevention in the Internet of Things Networks Using a Gru Based Deep Learning. *IEEE Access* **2020**, *8*, 183678–183689. [CrossRef]
14. Nayak, S.; Ahmed, N.; Misra, S. Deep Learning-Based Reliable Routing Attack Detection Mechanism for Industrial Internet of Things. *Ad Hoc Netw.* **2021**, *123*, 102661. [CrossRef]
15. Karmakar, S.; Sengupta, J.; Bit, S.D. LEADER: Low Overhead Rank Attack Detection for Securing RPL based IoT. In Proceedings of the 2021 International Conference on COMmunication Systems and NETworkS, COMSNETS 2021, Bangalore, India, 5–9 January 2021; pp. 429–437. [CrossRef]
16. Zahra, F.T.; Jhanjhi, N.Z.; Brohi, S.N.; Malik, N.A. Proposing a Rank and Wormhole Attack Detection Framework using Machine Learning. In Proceedings of the MACS 2019—13th International Conference on Mathematics, Actuarial Science, Computer Science and Statistics, Karachi, Pakistan, 14–15 December 2019. [CrossRef]
17. Le, A.; Loo, J.; Lasebae, A.; Vinel, A.; Chen, Y.; Chai, M. The Impact of Rank Attack on Network Topology of Routing Protocol for Low-Power and Lossy Networks. *IEEE Sens. J.* **2013**, *13*, 3685. [CrossRef]
18. Zahra, F.; Jhanjhi, N.Z.; Brohi, S.N.; Khan, N.A.; Masud, M.; AlZain, M.A. Rank and Wormhole Attack Detection Model for RPL-Based Internet of Things Using Machine Learning. *Sensors* **2022**, *22*, 6765. [CrossRef] [PubMed]
19. Hu, Y.C.; Perrig, A. Wormhole Attacks in Wireless Networks. *IEEE J. Sel. Areas Commun.* **2006**, *24*, 370–379. [CrossRef]
20. Dutta, N.; Singh, M.M. Wormhole Attack in Wireless Sensor Networks: A Critical Review. In *Advances in Intelligent Systems and Computing*; Springer: Singapore, 2019; Volume 702.

21. Gobinath, T.; Kalaiyarasi, T.; Kumar, P. Features monitoring system to defend wormhole attacks in wireless sensor networks. In Proceedings of the International Conference on Emerging Trends in Science, Engineering and Technology: Recent Advancements on Science and Engineering Innovation, INCOSET, Tiruchirappalli, Tamil Nadu, India, 13–14 December 2012.
22. Sookhak, M.; Akhundzada, A.; Sookhak, A.; Eslaminejad, M.; Gani, A.; Khan, M.K.; Li, X.; Wang, X. Geographic Wormhole Detection in Wireless Sensor Networks. *PLoS ONE* **2015**, *10*, e115324. [CrossRef]
23. Pongle, P.; Chavan, G. Real Time Intrusion and Wormhole Attack Detection in Internet of Things. *Int. J. Comput. Appl.* **2015**, *121*, 1–9. [CrossRef]
24. Zahra, F.T.; Jhanjhi, N.Z.; Brohi, S.N.; Malik, N.A.; Humayun, M. Proposing a Hybrid RPL Protocol for Rank and Wormhole Attack Mitigation using Machine Learning. In Proceedings of the 2020 2nd International Conference on Computer and Information Sciences, ICCIS 2020, Sakaka, Saudi Arabia, 13–15 October 2020. [CrossRef]
25. Gad, A.R.; Nashat, A.A.; Barkat, T.M. Intrusion Detection System Using Machine Learning for Vehicular Ad Hoc Networks Based on ToN-IoT Dataset. *IEEE Access* **2021**, *9*, 142206–142217. [CrossRef]
26. Samy, A.; Yu, H.; Zhang, H. Fog-Based Attack Detection Framework for Internet of Things Using Deep Learning. *IEEE Access* **2020**, *8*, 74571–74585. [CrossRef]
27. Chen, M.; Liu, W.; Wang, T.; Zhang, S.; Liu, A. A Game-Based Deep Reinforcement Learning Approach for Energy-Efficient Computation in MEC Systems. *Knowl.-Based Syst.* **2022**, *235*, 107660. [CrossRef]
28. Ullah, I.; Mahmoud, Q.H. Design and Development of a Deep Learning-Based Model for Anomaly Detection in IoT Networks. *IEEE Access* **2021**, *9*, 103906–103926. [CrossRef]
29. Shafiq, M.; Tian, Z.; Bashir, A.K.; Du, X.; Guizani, M. CorrAUC: A Malicious Bot-IoT Traffic Detection Method in IoT Network Using Machine-Learning Techniques. *IEEE Internet Things J.* **2021**, *8*, 3242–3254. [CrossRef]
30. Apostol, I.; Preda, M.; Nila, C.; Bica, I. Iot Botnet Anomaly Detection Using Unsupervised Deep Learning. *Electronics* **2021**, *10*, 1876. [CrossRef]
31. Qaddoura, R.; Al-Zoubi, A.M.; Almomani, I.; Faris, H. A Multi-Stage Classification Approach for Iot Intrusion Detection Based on Clustering with Oversampling. *Appl. Sci.* **2021**, *11*, 3022. [CrossRef]
32. Tien, C.W.; Huang, T.Y.; Chen, P.C.; Wang, J.H. Using Autoencoders for Anomaly Detection and Transfer Learning in Iot. *Computers* **2021**, *10*, 88. [CrossRef]
33. Pratomo, B.A.; Burnap, P.; Theodorakopoulos, G. Unsupervised Approach for Detecting Low Rate Attacks on Network Traffic with Autoencoder. In Proceedings of the 2018 International Conference on Cyber Security and Protection of Digital Services, Cyber Security 2018, Glasgow, Scotland, 11–12 June 2018.
34. Abdel-Basset, M.; Hawash, H.; Chakrabortty, R.K.; Ryan, M.J. Semi-Supervised Spatiotemporal Deep Learning for Intrusions Detection in IoT Networks. *IEEE Internet Things J.* **2021**, *8*, 12251–12265. [CrossRef]
35. Ravi, N.; Mercy Shalinie, S. Semisupervised-Learning-Based Security to Detect and Mitigate Intrusions in IoT Network. *IEEE Internet Things J.* **2020**, *7*, 11041–11052. [CrossRef]
36. Vu, L.; Nguyen, Q.U.; Nguyen, D.N.; Hoang, D.T.; Dutkiewicz, E. Deep Transfer Learning for IoT Attack Detection. *IEEE Access* **2020**, *8*, 107335–107344. [CrossRef]
37. Kelli, V.; Argyriou, V.; Lagkas, T.; Fragulis, G.; Grigoriou, E.; Sarigiannidis, P. IDS for Industrial Applications: A Federated Learning Approach with Active Personalization. *Sensors* **2021**, *21*, 6743. [CrossRef]
38. Al-Garadi, M.A.; Mohamed, A.; Al-Ali, A.K.; Du, X.; Ali, I.; Guizani, M. A Survey of Machine and Deep Learning Methods for Internet of Things (IoT) Security. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 1646–1685. [CrossRef]
39. Hussain, F.; Hussain, R.; Hassan, S.A.; Hossain, E. Machine Learning in IoT Security: Current Solutions and Future Challenges. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 1686–1721. [CrossRef]
40. Amanullah, M.A.; Habeeb, R.A.A.; Nasaruddin, F.H.; Gani, A.; Ahmed, E.; Nainar, A.S.M.; Akim, N.M.; Imran, M. Deep Learning and Big Data Technologies for IoT Security. *Comput. Commun.* **2020**, *151*, 495–517. [CrossRef]
41. Alimi, O.A.; Ouahada, K.; Abu-Mahfouz, A.M. A Review of Machine Learning Approaches to Power System Security and Stability. *IEEE Access* **2020**, *8*, 113512–113531. [CrossRef]
42. Azad, S.; Sabrina, F.; Wasimi, S. Transformation of smart grid using machine learning. In Proceedings of the 2019 29th Australasian Universities Power Engineering Conference, AUPEC 2019, Nadi, Fiji, 26–29 November 2019.
43. Li, W.; Li, Z.; Sun, J.; Wang, Y.; Liu, H.; Yang, J.; Gui, G. Spear and Shield: Attack and Detection for CNN-Based High Spatial Resolution Remote Sensing Images Identification. *IEEE Access* **2019**, *7*, 94583–94592. [CrossRef]
44. Zhang, Z.; Wang, Y.; Xie, L. A Novel Data Integrity Attack Detection Algorithm Based on Improved Grey Relational Analysis. *IEEE Access* **2018**, *6*, 73423–73433. [CrossRef]
45. Lee, S.; Abdullah, A.; Jhanjhi, N.; Kok, S. Classification of Botnet Attacks in IoT Smart Factory Using Honeypot Combined with Machine Learning. *PeerJ Comput. Sci.* **2021**, *7*, e350. [CrossRef]
46. Gopi, R.; Sathiyamoorthi, V.; Selvakumar, S.; Manikandan, R.; Chatterjee, P.; Jhanjhi, N.Z.; Luhach, A.K. Enhanced Method of ANN Based Model for Detection of DDoS Attacks on Multimedia Internet of Things. *Multimed. Tools Appl.* **2021**, *81*, 26739–26757. [CrossRef]
47. Chen, M.; Liu, W.; Zhang, N.; Li, J.; Ren, Y.; Yi, M.; Liu, A. GPDS: A Multi-Agent Deep Reinforcement Learning Game for Anti-Jamming Secure Computing in MEC Network. *Expert. Syst. Appl.* **2022**, *210*, 118394. [CrossRef]

48. Ren, Y.; Liu, W.; Liu, A.; Wang, T.; Li, A. A Privacy-Protected Intelligent Crowdsourcing Application of IoT Based on the Reinforcement Learning. *Future Gener. Comput. Syst.* **2022**, *127*, 56–69. [CrossRef]
49. Abdalgawad, N.; Sajun, A.; Kaddoura, Y.; Zualkernan, I.A.; Aloul, F. Generative Deep Learning to Detect Cyberattacks for the IoT-23 Dataset. *IEEE Access* **2022**, *10*, 6430–6441. [CrossRef]
50. Cvitic, I.; Perakovic, D.; Gupta, B.B.; Choo, K.K.R. Boosting-Based DDoS Detection in Internet of Things Systems. *IEEE Internet Things J.* **2022**, *9*, 2109–2123. [CrossRef]
51. Sivanathan, A.; Gharakheili, H.H.; Loi, F.; Radford, A.; Wijenayake, C.; Vishwanath, A.; Sivaraman, V. Classifying IoT Devices in Smart Environments Using Network Traffic Characteristics. *IEEE Trans. Mob. Comput.* **2019**, *18*, 1745–1759. [CrossRef]
52. Muthanna, M.S.A.; Alkanhel, R.; Muthanna, A.; Rafiq, A.; Abdullah, W.A.M. Towards SDN-Enabled, Intelligent Intrusion Detection System for Internet of Things (IoT). *IEEE Access* **2022**, *10*, 22756–22768. [CrossRef]
53. Sharafaldin, I.; Lashkari, A.H.; Ghorbani, A.A. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In Proceedings of the 4th International Conference on Information Systems Security and Privacy, ICISSP 2018, Funchal, Portugal, 22–24 January 2018.
54. Zeeshan, M.; Riaz, Q.; Bilal, M.A.; Shahzad, M.K.; Jabeen, H.; Haider, S.A.; Rahim, A. Protocol-Based Deep Intrusion Detection for DoS and DDoS Attacks Using UNSW-NB15 and Bot-IoT Data-Sets. *IEEE Access* **2022**, *10*, 2269–2283. [CrossRef]
55. Alharbi, A.; Alsubhi, K. Botnet Detection Approach Using Graph-Based Machine Learning. *IEEE Access* **2021**, *9*, 99166–99180. [CrossRef]
56. Saheed, Y.K.; Arowolo, M.O. Efficient Cyber Attack Detection on the Internet of Medical Things-Smart Environment Based on Deep Recurrent Neural Network and Machine Learning Algorithms. *IEEE Access* **2021**, *9*, 161546–161554. [CrossRef]
57. Said, A.M.; Yahyaoui, A.; Yaakoubi, F.; Abdellatif, T. Machine learning based rank attack detection for smart hospital infrastructure. In *Lecture Notes in Computer Science, International Conference on Smart Homes and Health Telematics, Hammamet, Tunisia, 24–26 June 2020*; Springer: Cham, Switzerland, 2020; pp. 28–40. [CrossRef]
58. Al-Shargabi, B.; Aleswid, M. Performance of RPL in Healthcare Wireless Sensor Network. *Int. J. Emerg. Trends Eng. Res.* **2020**, *8*, 797–803. [CrossRef]
59. Hariharakrishnan, J.; Bhalaji, N. Adaptability Analysis of 6LoWPAN and RPL for Healthcare Applications of Internet-of-Things. *J. ISMAC* **2021**, *3*, 69–81. [CrossRef]
60. Gara, F.; ben Saad, L.; ben Ayed, R.; Tourancheau, B. RPL Protocol adapted for healthcare and medical applications. In Proceedings of the IWCMC 2015—11th International Wireless Communications and Mobile Computing Conference, Dubrovnik, Croatia, 24–28 August 2015; pp. 690–695. [CrossRef]
61. Anaconda | Anaconda Distribution. Available online: https://www.anaconda.com/products/distribution (accessed on 14 October 2022).
62. Project Jupyter | Jupyter Notebook. Available online: https://jupyter.org/ (accessed on 25 June 2022).
63. Pandas—NumFOCUS. Available online: https://pandas.pydata.org/ (accessed on 26 June 2022).