Journal of Artificial Intelligence and Data Mining (JAIDM), Vol. 11, No. 4, 2023, 609-626.

Shahrood University of Technology

Journal of Artificial Intelligence and Data Mining (JAIDM)
Journal homepage: http://jad.shahroodut.ac.ir

Research paper

Exploring Impact of Data Noise on IoT Security: a Study using Decision Tree Classification in Intrusion Detection Systems

S. Mojtaba Matinkhah*, Roya Morshedi and Seyed Akbar Mostafavi

Department of Computer Engineering, Yazd University, Yazd, Iran.

Article Info

Article History:
Received 17 April 2023
Revised 15 July 2023
Accepted 24 August 2023

DOI:10.22044/jadm.2023.12964.2442

Keywords:
Classification Accuracy, Clean Data, Decision Tree Classifier, Intrusion Detection System, IoT Networks.

**Corresponding author: matinkhah@yazd.ac.ir (S. M. Matinkhah).*

Abstract

The Internet of Things (IoT) has emerged as a rapidly growing technology that enables seamless connectivity between a wide variety of devices. However, with this increased connectivity comes an increased risk of cyber-attacks. In the recent years, the development of intrusion detection systems (IDSs) has become critical for ensuring the security and privacy of IoT networks. This article presents a study that evaluates the accuracy of an intrusion detection system (IDS) for detecting network attacks in the IoT network. The proposed IDS uses the decision tree classifier and is tested on four benchmark datasets: NSL-KDD, BOT-IoT, CICIDS2017, and MQTT-IoT. The impact of noise on the training and test datasets on classification accuracy is analyzed. The results indicate that clean data has the highest accuracy, while noisy datasets significantly reduce accuracy. Furthermore, the study finds that when both training and test datasets are noisy, the accuracy of classification decreases further. The findings of this study demonstrate the importance of using clean data for training and testing an IDS in IoT networks to achieve accurate classification. This research work provides valuable insights for the development of a robust and accurate IDS for IoT networks.

1. Introduction

Detecting network attacks in the Internet of Things (IoT) is crucial because IoT devices are increasingly becoming interconnected, and are used in various critical applications, such as healthcare, transportation, and industrial control systems. These devices are also highly vulnerable to cyber-attacks due to their limited resources, lack of security measures, and outdated software. A successful cyber-attack on an IoT device can lead to significant consequences, such as loss of data, system failure, financial losses, and even physical harm. Therefore, having an effective intrusion detection system (IDS) is crucial to detect and prevent attacks on IoT networks, ensuring the security and reliability of these systems.

The decision tree classifier is helpful in detecting network attacks in the IoT because it is a widely used and effective classification algorithm that can handle both the categorical and numerical data. In an IDS for IoT networks, the decision tree classifier can be trained on a dataset of network traffic features and their corresponding labels (normal or malicious traffic) to learn the patterns and rules that distinguish normal traffic from attack traffic. The resulting decision tree can then be used to classify new network traffic as normal or malicious based on the features observed in the traffic. The decision tree classifier is also computationally efficient and easy to interpret, which makes it a suitable algorithm for real-time intrusion detection in resource-constrained IoT devices. Overall, the Decision Tree Classifier is a helpful tool in detecting network attacks in IoT networks due to its ability to handle diverse data types, its accuracy, and its computational efficiency.

The main problem addressed in this paper is the accuracy of an intrusion detection system (IDS) for detecting network attacks in the IoT network, and how the accuracy is impacted by noisy data in the training and test datasets. The study aims to evaluate the proposed IDS and analyze the effect of noise on classification accuracy to highlight the importance of using clean data for developing a robust and accurate IDS for IoT networks. This paper proposes an IDS for the IoT by utilizing decision tree classifier. We evaluate the accuracy of the proposed IDS on four well-known publicly available datasets, namely NSL-KDD, BOT-IoT, CICIDS2017, and MQTT-IoT. Accuracy is measured with varying levels of noise variances in the training and test datasets, and the results are analyzed through the generated graphs. Moreover, the paper provides insights that show how clean data has the highest accuracy, and accuracy is decreased when both the training and test datasets have noise. Similarly, when the class label contains noise, accuracy is further affected, and C4.5 has a destructive influence. Furthermore, the findings demonstrate that the accuracy of decision tree classification decreases with an increase in noise variance. This research work contributes to determining the significance of having a clean training and test data for an accurate classification in IDS for IoT networks. It emphasizes the importance of developing a more robust and reliable IDS for IoT networks that can identify common attacks in modern datasets.

In the followings, we first review the most recent studies on application of decision tree classification in IoT intrusion detection. In the next section, the dataset used as benchmarked is introduced, and at that point our methodology of how we implemented the model is described. Finally, we discussed our results and conclude with some lessons learned for future improvements.

2. Related Works

This section presents a comparative analysis of the recent papers that focus on enhancing security in IoT using Intrusion Detection Systems (IDSs) with Artificial Intelligence (AI) methodologies. By comparing and analyzing various aspects of recent research on this topic, we aim to gain a better understanding of the current state-of-the-art in IDS research and identify potential areas for future investigation. We justify the need for our approach by noting the importance of understanding how data noise affects the accuracy of IDSs in IoT networks. Our research project evaluates the accuracy of an IDS for detecting network attacks in an IoT network using the decision tree classifier, and highlights the importance of using clean data for training and testing an IDS in order to achieve accurate classification. Summarized in Table 1 are the recent papers on IDSs for IoT networks, with a focus on proposed methodologies, network applications, and limitations of their approaches. While some papers propose novel IDS methods using various machine learning techniques, such as deep learning-based approaches, decision tree classifiers, reinforced learning, and graph convolutional networks, many have limitations such as limited evaluation on benchmark datasets, lack of generalizability to other IoT networks, and difficulty implementing solutions on resource-constrained devices. The need for further research to develop effective and robust IDSs for IoT networks is emphasized, as highlighted in the table 1.

Table 1. Comparative analysis of IDSs for enhancing IoT security with AI methodologies: a review of recent research work.

Authors	proposal	Network	Limitations
Gyamfi and Jurcut [1]	Lightweight NIDS based on OI-SVDD and AS-ELM	Industrial IoT devices	Only focuses on industrial IoT, proposed solution may be difficult to implement for resource-constrained IIoT devices, evaluation limited to two datasets, and generalization to other IoT networks is uncertain.
Deng et al. [2]	Flow topology based graph convolutional networks	Network Intrusion Detection	Requires some labeled traffic flow data for training, may be difficult to obtain in highly dynamic IoT networks, complexity may pose challenges for deployment and maintenance. Therefore while, they focus on a specific approach for intrusion detection in IoT networks and does not address the issue of noise in the datasets used for training and testing, they address different aspects of the problem.
Wu et al. [3]	Using big data mining, fuzzy rough set, generative adversarial network (GAN), and convolutional neural network (CNN) for intelligent intrusion detection	implementing on resource-limited edge nodes	Limited to evaluating one specific intrusion detection system and its performance under certain conditions. complex GAN architectures, such Wasserstein GANs, have not been studied as thoroughly. Another factor that has not been covered is that CNN approaches used in multi-class classification are not successful. focus on proposing a new intelligent intrusion detection algorithm based on big data mining, while our article focuses on evaluating the accuracy of an existing intrusion detection system using decision tree classifier and the study of the impact of noise on the

			training and test datasets and how it affects the classification accuracy of the IDS.
Wu et al. [4]	Multisource heterogeneous domain adaptation, semantic transfer, geometric similarity-aware Pseudo-label refinement	effective intrusion detection in large-scale, scarcely labeled IoT Domains	Computationally intensive and difficult to implement on resource-limited edge nodes in IoT Networks. The proposed model has been applied only for few types of attacks features. use a complex method involving multisource heterogeneous domain adaptation, semantic transfer, and geometric similarity-aware pseudo-label refinement, which may be computationally intensive and difficult to implement on resource-limited edge nodes in IoT networks. In contrast, we proposed a simpler decision tree classifier and focuses on the impact of noise on classification accuracy, which has practical implications for the development of a robust and accurate IDS for IoT networks.
Ruzafa-Alcázar et al. [5]	Evaluating differential privacy techniques for federated learning in the context of an intrusion detection system for industrial IoT	industrial IoT and federated learning	Limited evaluation to a single dataset (ToNIoT); no insights into generalizability of proposed approach across different datasets; no discussion on potential trade-offs between accuracy and privacy in the proposed approach, which is crucial for IDS development in IoT networks.
Long et al. [6]	A regularized cross-layer ladder network	IoT networks	Not compared with existing intrusion detection systems; evaluated on a single dataset, limiting generalizability to other datasets and real-world settings
Oseni et al. [7]	Explainable deep learning-based IDS	internet of vehicles (IoVs)	Limited generalizability to other types of IoT networks, unclear performance on other datasets or in other IoT network environments, need for further evaluation. Additionally, while the article proposes an explainable deep learning-based intrusion detection framework, it is unclear how well this framework would perform on other datasets or in other IoT network environments. Therefore, it may be necessary to evaluate the performance and applicability of the proposed framework on a wider range of datasets and IoT network scenarios to determine its usefulness in improving the transparency and resiliency of deep learning-based IDS in IoT networks.
Mehedi et al.[8]	Deep transfer learning-based IDS with attribute selection	IoT networks	Limited evaluation on benchmark datasets, may not be as comprehensive as the decision tree classifier approach. the article presented here evaluates the accuracy of an IDS for detecting network attacks in IoT networks using the Decision Tree Classifier which provides more comprehensive insights and results that are valuable for the development of dependable IDS models in IoT networks.
Bebortta et al. [9]	Equilibrium Optimization-based Artificial Neural Network (EO-ANN)	Fog-enabled IoT	Unclear performance in the presence of noisy data, lack of evaluation on real-world datasets and scenarios.
Alani and Awad [10]	Two-layer intrusion detection system for IoT	IoT networks	Lack of comprehensive analysis of the impact of noisy data on accuracy, less thorough evaluation compared to our study using decision tree classifier.
Wu et al. [11]	Heterogeneous domain adaptation using graph alignment method	data-scarce domain IoT	Potential complexity and errors in intrusion detection process due to imperfect alignment, reliance on pseudo-labels may lead to errors in classification. In contrast, our approach relies on the use of clean data for training and testing.
Thakkar and Lohiya [12]	Ensemble learning-based deep neural network	IoT network	Primarily focused on addressing class imbalance rather than evaluating accuracy of intrusion detection system, limited comparison to other existing systems, no analysis of impact of noise on performance.
Sharadqh et al. [13]	Hybrid chain: Blockchain enabled framework for bi-level intrusion detection and graph-based mitigation for security provisioning in edge assisted IoT environment	IoT Networks	The proposed approach for intrusion detection in IoT networks using blockchain and optimization algorithms has limitations in its generalizability, scalability, and ability to address the root cause of the problem, which is accurate intrusion detection.
Ma et al. [14]	Collaborative learning-based intrusion detection framework called ADCL for IoT networks	IoT Networks	The article does not provide a detailed evaluation of the proposed framework, making it difficult to assess the effectiveness of the approach. There is also no comparison with other state-of-the-art methods, limiting the contribution of this article. The paper is unclear about how the proposed framework mitigates the limitations of a single model.
Kandhro et al. [15]	Decision Tree Classifier for intrusion detection in IoT networks and	IoT Networks	The use of a deep learning-based approach for intrusion detection may require a large amount of

	comparison with Deep Learning-based approach		labeled data to achieve high accuracy, which may be a limitation in practical scenarios where labeled data is scarce or expensive to obtain. In contrast, this article uses a decision tree classifier which can be trained on smaller datasets and requires less computational resources. The paper demonstrates the effectiveness of this approach in terms of accuracy, reliability, and efficiency in detecting all types of attacks.
Telikani et al. [16]	Hybrid model of stacked autoencoders (SAE) and convolutional neural networks (CNNs) with a new cost-dependent loss function called EvolCostDeep and fog computing-enabled framework called DeepIDSFog	imbalanced data distribution in industrial IoT environments for Intrusion detection	The article does not provide a detailed comparison with other state-of-the-art intrusion detection systems, which makes it difficult to evaluate the effectiveness of the proposed EvolCostDeep model and DeepIDSFog framework in comparison to existing solutions.
Abdel Wahab [17]	Drift detection technique using principal component analysis (PCA), online outlier detection technique, and online deep neural network (DNN) with hedge weighting mechanism	accuracy of machine learning-based intrusion detection systems in dynamic IoT environments	It is unclear how the IoT-based intrusion detection dataset was selected or how the performance of the proposed solution was compared to the static DNN model. Additionally, it is not clear how the proposed drift detection and outlier detection techniques compare to existing techniques for addressing data and concept drift in machine learning-based IDSs. The article does not address other potential challenges such as adversarial attacks, resource constraints, or the need for interpretability and explainability in decision-making. While addressing drifts in dynamic IoT environments is important, it is also important to consider other factors that can impact the effectiveness of an IDS.
Liang et al. [18]	Optimized intra/inter-class-structure-based variational few-shot learning (OICS-VFSL) model for microservice-oriented intrusion detection in distributed IoT systems	imbalanced learning in microservice-oriented intrusion detection in distributed IoT systems	It is not clear how the proposed model performs in terms of detecting attacks that are not novel or in detecting attacks on different types of IoT devices or networks. Moreover, it is unclear how the two public datasets were selected or how the performance of the proposed OICS-VFSL model was compared to the baseline methods. More information is needed to evaluate the effectiveness of the proposed approach.
Zhou et al. [19]	Attack generation method for testing intrusion detection systems in IoT networks and a new method for generating adversarial examples for intrusion detection systems in IoT networks	testing robustness of intrusion detection systems in IoT networks	The proposed approach may raise concerns about the potential for attackers to use similar methods to bypass these systems in real-world attacks, undermining the effectiveness of intrusion detection systems in IoT networks. The article does not provide any solution to address this limitation.
Booij et al. [20]	Decision tree classifier for intrusion detection in IoT networks and evaluation of its accuracy on four different benchmark datasets	intrusion detection in IoT networks	The article acknowledges the importance of data sets and standardization efforts in IoT security research, which is an important consideration for the development and evaluation of intrusion detection systems in IoT networks.
Zeeshan et al. [21]	Protocol-based deep intrusion detection	UNSW-NB15 and bot-IoT data-sets	Noisy data issue and its impact on classification accuracy were not addressed
Siddharthan et al. [22]	Elite machine learning algorithms (EML)	IoT networks	Lack of details on the dataset used for testing and evaluation methodology, unclear how proposed IDS handles noisy datasets or the impact of noisy datasets on classification accuracy. Furthermore, it is unclear how the proposed IDS handles noisy datasets or the impact of noisy datasets on classification accuracy.
Muthanna et al. [23]	Intelligent and efficient framework for threat detection in IoT using cuLSTMGRU and SDN technologies	intrusion detection system for IoT environments	Not clearly stated which dataset was used and how it was collected. The author has not discussed implementing a real-time SDN for existing networks. Additionally, the experimental studies for classification were conducted extensively only for small- sample intrusion and normal network requests.
Miranda et al. [24]	Reinforcement learning preventing rank attacks in low-power IoT networks	SDN controller in low-power IoT networks	The article does not discuss the implementation challenges of the proposed scheme. Methodology to apply high volumes of data was not discussed, demanding real- time forecast and the sense to reduce the data's dimensionalities.

3. Datasets and Methodology

3.1. Datasets

Intrusion Detection Systems (IDSs) are critical in network security to detect and respond to suspicious activity or attacks. Evaluating an IDS algorithm's effectiveness is crucial to ensure that it can identify various types of attacks accurately and efficiently. Benchmark datasets, such as the KDD Cup 99 or the NSL-KDD dataset, provide a standardized way to evaluate IDS algorithms' performance. Using a benchmark dataset for IDS evaluation allows the researchers to make a comparison between different IDS algorithms using the same dataset. This comparison can be based on various metrics such as detection rate, false alarm rate, and accuracy. Additionally, benchmark datasets provide a consistent testing environment, which is essential when testing IDS algorithms across different research studies. Another advantage of using benchmark datasets is their scalability. These datasets can simulate various network scenarios, including large-scale attacks, that would be difficult to reproduce in a lab environment. The researchers can test their IDS algorithms' scalability in handling an increased number of attacks without the need for physical hardware changes.

NSL-KDD, BOT-IoT, CICIDS2017, and MQTT-IoT datasets are all popular benchmark datasets which we used for evaluating intrusion detection and classification systems in networking and IoT environments as described in the following:

The NSL-KDD dataset, also known as the "NSL-KDD Cup 99" dataset, is a benchmark dataset that has been used by intrusion detection researchers to evaluate the effectiveness of their intrusion detection algorithms. This dataset is an improved version of the KDD Cup 99 dataset, which was used in the KDD Cup 1999 data mining competition to create efficient detection models for network intrusion detection systems. The NSL-KDD dataset contains a mix of normal and attack patterns, making it a realistic representation of network traffic. The dataset consists of 41 features, which include 34 continuous and 7 categorical features. The continuous features include attributes such as duration of the connection, number of bytes transferred, and number of packets exchanged, while the categorical features include information such as protocol type and service type. The NSL-KDD dataset has been widely used in research publications for the evaluation of intrusion detection techniques. The researchers have used this dataset to develop and test various machine learning and deep learning models for intrusion detection. The dataset has been used to evaluate the performance of various classification algorithms, such as decision trees, neural networks, and support vector machines, among others.

One of the major advantages of using the NSL-KDD dataset is that it provides a standardized and realistic dataset for the evaluation of intrusion detection algorithms. This ensures that the results obtained from different studies can be compared and replicated, which is important for the advancement of the field of intrusion detection. Furthermore, since the dataset contains a mix of normal and attack traffic, it provides a realistic representation of the types of traffic that intrusion detection systems must be able to handle. Another advantage of the NSL-KDD dataset is that it is publicly available, which allows the researchers from around the world to access and use it for their studies. This makes it possible for researchers to compare their results with other studies and to collaborate with other researchers in the field of intrusion detection.

NSL-KDD is a network intrusion detection (IDS) dataset that is used to train and evaluate IDS systems. It is a revised version of the KDD'99 dataset, which was released in 1999. The NSL-KDD dataset was created by the Network Security Laboratory at the University of New Brunswick in Canada. The dataset is divided into two parts: a training set of 125,973 records and a test set of 22,544 records. The training set is used to train IDS systems, and the test set is used to evaluate the performance of those systems. The NSL-KDD dataset has several advantages over the KDD'99 dataset. First, it is more balanced, with a more even distribution of normal and attack records. Second, it is more realistic, with a more representative mix of attack types. Third, it is more efficient, with a smaller number of features and records. The NSL-KDD dataset is a valuable resource for researchers and developers of IDS systems. It is a widely used benchmark dataset that can be used to compare the performance of different IDS algorithms. the NSL-KDD dataset consists of 148,171 records, each of which represents a single network connection. The records are labeled as either normal or one of five types of attack: Denial of Service (DoS), Remote to Local (R2L), User to Root (U2R), Probing, and Web Application Attack (Web).

Table 2.Description of the NSL-KDD Dataset.

Attack category	Attack type		
		Training set	Testing set
		KDDTrain+20Percent	KDDTest+
	Neptune	8282	4657
	Smurf	529	665
	Pod	38	41
Denial of Service (DoS)	Land	1	7
	Back	196	359
	Apache2	-	737
User to Root (U2R)	Spy	1	-
	Xterm	-	13
	Http-tunnel	-	133
	Sql-attack	-	2
	Snmp-guess	-	331
	Multihop	2	18
Remote to Local (R2L)	Warezmaster	7	944
	Warezclient	181	-
	Named	-	17
	Xlock	-	9
	Xsnoop	-	4
Probe	Nmap	301	73
	Satan	691	735
	Saint	-	319
	Mscan	-	996

The Bot-IoT dataset is a significant development in the field of cybersecurity research, providing a valuable resource for the researchers and industry professionals to develop more effective detection algorithms for IoT botnets. The dataset contains traffic data collected from a variety of IoT devices, providing a comprehensive view of the behavior and characteristics of IoT botnets. The data has been preprocessed to extract features such as IP addresses, ports, and protocols used, allowing the researchers to analyze and study the traffic patterns of IoT botnets. The Bot-IoT dataset is publicly available, allowing the researchers to use it for various research studies to develop more effective detection techniques. These techniques can then be used to develop more robust security measures to protect IoT devices from botnet attacks. The availability of this dataset has led to several research studies, including the development of machine learning algorithms for detecting botnets, the identification of botnet command and control servers, and the analysis of botnet behavior. The significance of the Bot-IoT dataset cannot be overstated, as it provides a valuable resource for the researchers and industry professionals to develop more effective security measures for IoT devices. With the rise of IoT devices, the threat of botnet attacks is only set to increase. Therefore, it is crucial that the researchers and industry professionals work together to improve IoT security and protect these devices from malicious actors. The BoT-IoT dataset was created by designing a realistic network environment in the Cyber Range Lab of UNSW Canberra. The network environment incorporated a combination of normal and botnet traffic. The dataset’s source files are provided in different formats, including the original pcap files, the generated argus files and csv files. The files were separated, based on attack category and subcategory, to better assist in labeling process.

Bot-IoT is a recent and publicly available dataset that represents botnet attack traffic in Internet of Things (IoT) networks. About 9,000 of the roughly 73,000,000 instances in the dataset are labeled as normal traffic. In this study, we provide an easy-to-learn approach for Bot-IoT. Our method involves the use of a minimum number of dataset features and a simple learning algorithm for accurate classification. To be more specific, our contribution revolves around the use of only 3 out of the 29 Bot-IoT features and the Decision Tree classifier. In keeping with our definition of easy-to-learn, we require that predictive models have Area Under the Receiver Operating Characteristic Curve (AUC) and Area Under the Precision-Recall Curve (AUPRC) mean scores greater than 0.99. Our results demonstrate that the Bot-IoT dataset yields an easy-to-learn Decision Tree model. The Bot-IoT dataset is a publicly available dataset of network traffic that can be used to train and evaluate machine learning models for botnet detection in Internet of Things (IoT) networks. The dataset was created by researchers at the University of New South Wales in Australia. The Bot-IoT dataset contains over 72 million records of network traffic, including both normal and botnet traffic.

Table 3.Description of the B0T-IOT dataset.

Category	Attack Type	Flow count	Training	Test
BENIGN	**BENIGN**	9543	7634	1909
Information gathering	**Service scanning**	1,463,364	117,069	29,267
	OS Fingerprinting	358,275	28,662	7166
	DDoS TCP	19,547,603	1,563,808	390,952
DDoS attack	**DDoS UDP**	18,965,106	1,517,208	379,302
	DDoS HTTP	19,771	1582	395
	DoS TCP	12,315,997	985,280	246,320
DoS attack	**DoS UDP**	20,659,491	1,652,759	413,190
	DoS HTTP	29,706	2376	594
Information theft	**Keylogging**	1469	1175	294
	Data theft	118	94	24
Total	/	73,370,443	5,877,647	1,469,413

The CICIDS2017 dataset is a publicly available dataset that contains network traffic for various network attacks. The dataset consists of both benign network traffic and malicious network traffic, allowing the researchers to train and test their IDS on a diverse range of attack scenarios. The attacks included in the dataset range from botnet attacks to DoS and DDoS attacks, port scans, and more. The dataset is labeled, with both benign and malicious network flows being labeled. The dataset also includes full packet payloads in pcap format, which can be used to analyze network traffic in detail. Furthermore, the dataset includes profiles of each network flow, allowing the researchers to gain insight into the characteristics of the traffic. The labeled flows and CSV files for machine and deep learning purposes are also publicly available for researchers. The availability of the CICIDS2017 dataset has been a significant development in the field of cybersecurity research, allowing the researchers to train and test their IDS on a diverse range of network attacks. The labeled dataset provides a benchmark for researchers to compare the effectiveness of different IDS and to develop new intrusion detection techniques. Moreover, the availability of the full packet payloads in pcap format provides a detailed view of network traffic, allowing researchers to analyze the traffic in detail and gain insights into the characteristics of different attacks. The CICIDS2017 dataset has been used in several research studies, including the development of machine learning algorithms for intrusion detection and the analysis of network traffic for botnet detection. The dataset has also been used to evaluate the effectiveness of various IDS, allowing the researchers to compare different systems and identify areas for improvement. The dataset can be used for training and evaluating network intrusion detection systems (NIDS) and other security-related applications. It is publicly available for academic and research purposes. CICIDS2017 contains more than 80 million labeled flows with 85 features. The CICIDS2017 dataset is a publicly available dataset of network traffic that can be used to train and evaluate machine learning models for intrusion detection. The dataset was created by researchers at the Canadian Institute for Cybersecurity (CIC) at the University of New Brunswick in Canada. The CICIDS2017 dataset contains over 2.8 million records of network traffic, including both benign and malicious traffic. The malicious traffic includes a variety of attack types, such as DoS attacks, DDoS attacks, brute-force attacks, and web application attacks. The CICIDS2017 dataset is a valuable resource for researchers and developers of intrusion detection systems. It is one of the largest and most comprehensive publicly available datasets of network traffic, and it includes a wide range of attack types. The CICIDS2017 dataset is a valuable resource for the cybersecurity community, and it is helping to advance the development of more effective intrusion detection systems. Here are some of the key features of the CICIDS2017 dataset: It is a large dataset, containing over 2.8 million records of network traffic. It is a comprehensive dataset, including both benign and malicious traffic, as well as a variety of attack types. It is a realistic dataset, created using a realistic testbed environment.

Table 4. Description of the CIC-IDS2017 dataset.

	Category	Total	Total (-Rows with Lack Info)	Training	Test
BENIGN	**BENIGN**	2,273,097	2,271,320	20,000	20,000
DOS	**DDoS**	128,027	128,025	2700	3300
	DoS slowloris	5796	5796	1350	1650
	DoS Slowhttptest	5499	5499	2171	1169
	DoS Hulk	231,073	230,124	4500	5500
	DoS GoldenEye	10,293	10,293	1300	700
	Heartbleed	11	11	5	5
PortScan	**PortScan**	158,930	158,804	3808	4192
Bot	**Bot**	1966	1956	936	624
Brute-Force	**FTP-Patator**	7938	7935	900	1100
	SSH-Patator	5897	5897	900	1100
Web Attack	**WebAttack-Brute Force**	1507	1507	910	490
	Web Attack-XSS	652	652	480	160
	WebAttack-SQL Injection	21	21	16	4
Infiltration	**Infiltration**	36	36	24	6
Total Attack		471,454	470,365	20,000	20,000
Total		2,830,743	2,827,876	40,000	40,000

Finally, The MQTT-IoT dataset is a publicly accessible dataset that contains a large volume of MQTT messages that were collected from a range of IoT devices. The MQTT protocol is a popular messaging protocol that is widely used for facilitating communication between IoT devices and servers. MQTT is lightweight and efficient, making it an ideal choice for IoT devices with limited resources. The dataset includes more than 13 million MQTT messages that were collected from diverse IoT devices, such as temperature and humidity sensors, light bulbs, and power meters. The dataset is a valuable resource for the researchers and practitioners who are interested in analyzing IoT data and developing machine learning algorithms for IoT applications. The messages in the dataset were collected over a period of several months from publicly available MQTT brokers. MQTT brokers are the central hub for all MQTT communication, and they act as a mediator between IoT devices and servers. The dataset was collected from a diverse range of MQTT brokers, ensuring that the messages represent a broad range of IoT devices and scenarios. The researchers and practitioners can use the dataset for various purposes, such as developing intrusion detection systems for IoT networks, analyzing network traffic patterns in IoT environments, and developing machine learning algorithms for IoT applications. The dataset is available for download, and it is an excellent resource for anyone interested in IoT data analysis and machine learning. The MQTT-IoT-IDS2020 dataset is a publicly available dataset that can be used to train and evaluate machine learning models for intrusion detection in MQTT-based IoT networks. The dataset was created by researchers at the University of Strathclyde in the United Kingdom. The MQTT-IoT-IDS2020 dataset contains over 1.2 million records of network traffic, including both normal and attack traffic. The attack traffic includes a variety of attack types, such as brute-force attacks, denial-of-service attacks, and man-in-the-middle attacks. The MQTT-IoT-IDS2020 dataset is a valuable resource for researchers and developers of intrusion detection systems for MQTT-based IoT networks. It is one of the first and most comprehensive publicly available datasets of MQTT network traffic, and it includes a variety of attack types. Here are some of the key features of the MQTT-IoT-IDS2020 dataset:

- It is a large dataset, containing over 1.2 million records of network traffic.
- It is a comprehensive dataset, including both normal and attack traffic, as well as a variety of attack types.
- It is a realistic dataset, created using a realistic testbed environment.
- It is a well-labeled dataset, with each record labeled as either normal or attack traffic.

The MQTT-IoT-IDS2020 dataset is available for download from the IEEE DataPort website.

The MQTT-IoT-IDS2020 dataset is a valuable resource for the cybersecurity community, and it is helping to advance the development of more effective intrusion detection systems for MQTT-based IoT networks.

Table5. Description MQTT_IoT_IDS2020 dataset.

Category	With redundancy	Without redundancy
Normal	3343318	167159
MQTT-Brutforce	2002780	2001972
Scan-A	31245	292276
Scan-u	33404	27843
Sparta	1252259	1217198
Total	3654006	3443448

In conclusion, benchmark datasets are vital in evaluating IDS algorithms accurately and objectively. They provide a standardized and scalable way to test the performance of different IDS algorithms against various types of attacks. Overall, these datasets differ in their size, complexity, and the types of attacks they simulate. NSL-KDD and CICIDS2017 are larger and more complex datasets that simulate a wider range of attacks, while BOT-IoT and MQTT-IoT focus specifically on botnet and MQTT protocol traffic. Therefore, the choice of dataset is crucial for demonstrating the validity of the exactness of our proposed method.

3.2. Methodology

This section provides an overview of how Decision Tree Classification can be used to improve security in IoT networks. It explains the basic principles of Decision Tree Classification and how it can be used to detect and classify different types of network attacks, such as DDoS attacks, malware infections, and unauthorized access attempts. The text also mentions several datasets that can be used to train and test the algorithm for IoT network security. Furthermore, the steps involved in implementing decision tree classification in Python programming language using the scikit-learn library is outlined. It explains how to load and preprocess the dataset, split it into a training and test set, create the decision tree classifier, evaluate its performance, and analyze the results. We mention several metrics that can be used to evaluate the performance of the algorithm.

decision tree classification is a powerful machine learning algorithm that can be used to improve security in IoT networks. It works by constructing a tree-like model of decisions and their possible consequences. Each internal node of the tree represents a test on a specific feature, and each branch represents the outcome of the test. The leaves of the tree represent the final decisions or classifications. The algorithm is called a "decision tree" because it makes a sequence of decisions that lead to a final decision or classification.

In the context of IoT security, decision tree classification can be used to detect and classify different types of network attacks, such as DDoS attacks, malware infections, and unauthorized access attempts. To do this, the algorithm is trained on a set of labeled data, which contains examples of normal network traffic as well as examples of different types of attacks. The algorithm uses this training data to build a model of the normal behavior of the network and to learn how to detect anomalous behavior that may indicate an attack.

Several datasets are commonly used in the research to evaluate the performance of decision tree classification for IoT network security. The NSL-KDD dataset, for example, contains a set of network traffic data that has been preprocessed to remove duplicate and irrelevant records, as well as to reduce the number of features. The BOT-IoT dataset contains data on network traffic in IoT environments, including traffic generated by different types of IoT devices. The CICIDS2017 dataset contains data on different types of network attacks, including DDoS attacks and malware infections. Finally, the MQTT-IoT dataset contains data on network traffic in MQTT-based IoT environments. The decision tree classifier (DT) is indeed a powerful approach to multistage decision making. It is a supervised machine learning algorithm that builds a tree-like model of decisions and their possible consequences. The basic idea behind a multistage approach is to divide a complex decision problem into a series of simpler decisions, each leading to a subsequent stage of decision-making.

DTs excel at breaking down a complex decision-making process into a collection of simpler decisions. The algorithm constructs a tree structure, where each internal node represents a decision based on a specific feature or attribute, and each leaf node represents the outcome or class label. By traversing the tree from the root to a leaf, the DT algorithm makes a series of decisions based on the feature values, leading to a final prediction or decision.

One of the significant advantages of DTs is their interpretability. The resulting tree structure provides a clear and intuitive representation of the decision process. It allows humans to understand and interpret the decision rules followed by the algorithm. This interpretability is valuable in domains where transparency and explainability are crucial, such as medicine, finance, or legal systems. Furthermore, DTs can handle both numerical and categorical features and can handle missing values without requiring extensive preprocessing. They are also robust to outliers and can capture nonlinear relationships between features and the target variable through recursive partitioning.

However, it is important to note that DTs have certain limitations. They can be sensitive to small changes in the training data, leading to different tree structures and potentially different decisions. DTs can also be prone to overfitting, where the model becomes too complex and performs poorly on unseen data. Several techniques, such as pruning and setting regularization parameters, can mitigate these issues.

In summary, Decision Tree classifiers are powerful tools for multi stage decision making. They excel at breaking down complex decision processes into simpler components and provide interpretable solutions. However, they also have some limitations that need to be considered and addressed to ensure their optimal performance.

This algorithm can be implemented using Python programming language and the scikit-learn library. The algorithm can be trained and tested on datasets that contain network traffic data, such as the NSL-KDD, BOT-IoT, CICIDS2017, and MQTT-IoT datasets. These datasets are commonly used for benchmarking intrusion detection systems (IDSs) in IoT networks.

To illustrate how decision tree classification can be used for IoT network security, consider the following example. We had couple of datasets of network traffic in an IoT environment, and we used decision tree classification to detect DDoS attacks. We first preprocessed the dataset to remove irrelevant features and to reduce the number of features to a manageable size. We then did split the dataset into training data and test data, and we used the training data to train the Decision Tree Classification algorithm. Once the algorithm is trained, we used the test data to evaluate its performance. To implement decision tree classification for enhancing security in IoT networks, the following steps are followed:

1. Learning dataset: This dataset contains 20% of the samples (573 samples). The learning dataset is used exclusively for training the decision tree model. The recursive partitioning procedure, described earlier, is applied to this dataset to construct the decision tree.
2. Single-split validation dataset: This dataset also contains 20% of the samples (573 samples). It is used for validating the learning process and avoiding overfitting. Overfitting occurs when the model becomes too specific to the training data and performs poorly on unseen data. The validation dataset does not participate in the training process but is used to assess the performance of the decision tree model. The model's hyperparameters, such as the splitting criterion, maximal depth, and minimal leaf size, can be tuned based on the performance on this validation dataset.
3. Testing dataset: This dataset comprises the remaining 60% of the samples (1720 samples). These samples are not used in any way during the training process. They serve solely for the final performance assessment of the decision tree model. By evaluating the model on this testing dataset, we can gain insights into its real performance and its ability to generalize to new, unseen data.

   This partitioning strategy allows for a comprehensive evaluation of the decision tree model's performance. The learning dataset trains the model, the single-split validation dataset helps optimize the model's hyperparameters, and the testing dataset provides a reliable measure of the model's performance on unseen data.

By analyzing the results of the classifier at different levels of noise variance, patterns and trends can be identified that help to determine the optimal level of noise variance for the model. This is the point at which the model performs best and has the highest accuracy, indicating that it is well-suited for real-world scenarios.

Once the optimal noise variance is identified, the decision tree classifier can be trained and deployed with confidence, knowing that it will perform well in real-world situations. This ensures that the model is reliable and accurate, and can effectively classify new data points based on the patterns and trends identified during the analysis. This is critical for decision-making and other applications where accurate predictions are necessary.

4. Results

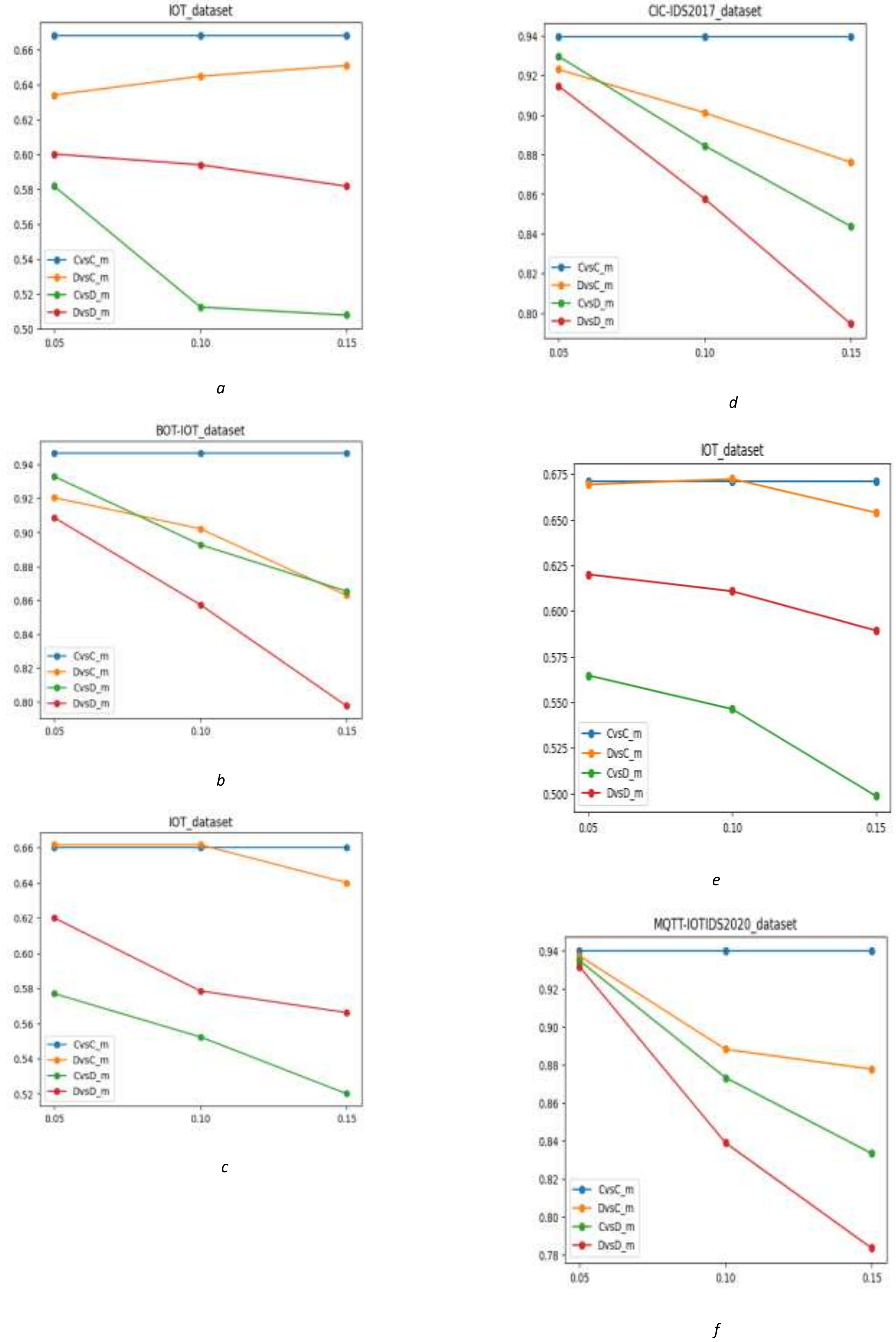
4.1 Analysis of effect of noise on decision tree

Decision tree classifiers are a type of supervised learning algorithm used for both classification and regression tasks. They work by recursively splitting the dataset into smaller subsets based on the most significant attribute at each node of the tree. This process continues until the data in each subset belongs to the same class or has similar characteristics. The nodes of the decision tree represent the attributes or features, and the branches represent the outcomes or decisions based on those attributes. The decision tree algorithm uses various measures such as Gini impurity or information gain to determine the best attribute to split the data at each node. One of the main advantages of decision trees is that they are easy to interpret and visualize, making them useful for understanding the decision-making process. They can handle both numerical and categorical data and are robust to outliers and missing values. However, decision trees are prone to overfitting, especially when the tree becomes too deep or complex. To address this, techniques such as pruning, setting a maximum depth, or using ensemble methods like random forests can be applied. In the field of machine learning, decision tree classifiers are commonly used for classification tasks. However, the performance of these classifiers can be affected by the presence of noise in the data. To ensure the accuracy and reliability of the decision tree classifier, it is essential to determine the optimal level of noise variance that it can handle. This is where the process of analyzing the results comes in. By analyzing the performance of the decision tree classifier at different levels of noise variance, it is possible to identify the optimal noise variance that produces the highest accuracy. Once the optimal noise variance is determined, the decision tree classifier can be confidently trained and deployed in real-world scenarios, knowing that it will perform well. In this context, this section will discuss how analyzing the results can lead to the identification of the optimal noise variance and enable the development of a high-performing decision tree classifier.

We analyzed the classification accuracy of decision trees for each dataset by varying noise variances of 5%, 10%, and 15%. The results indicate that the accuracy of the decision tree is highest for clean data (cvsc), followed by (DvsC). Specifically, the accuracy of the decision tree is better when the training data has noise rather than the test data (CvsD) or both training and test data (DvsD) having noise. However, for the datasets CvsD and DvsD, the accuracy is contrary to the idea that DvsD is better. This is only applicable to datasets with numerical data, where the same amount of noise is applied to both the training and test data. Consequently, the accuracy is better than CvsD, although the accuracy decreases with an increase in noise. It is worth noting that in some cases, an increase in noise can improve accuracy. This may be because with the increase or decrease in noise, the numerical data can either become closer to reality or move away from it. In the following figures, the horizontal axis represents the level of noise in different features of the datasets, while the vertical axis represents the classification accuracy of the datasets. The first figure focuses on the IoT dataset, which contains continuous data with six distinct classes. The second, fourth, sixth, and eighth figures correspond to the BOT-IoT, CICIDS-2017, MQTT-IoT-IDS2020, and NSL-KDD datasets, which are discrete and have two output classes. The first figure depicts the impact of feature noise on the IoT dataset.

Figure 1(a) to Figure 1(h) clearly shows that the accuracy of decision tree classification decreases as the amount of noise in the data increases. The highest classification accuracy is achieved when both the training and test data sets are clean or without noise (mode CvsC). Following this, the highest accuracy is related to DvsC, and CvsD. However, in the IoT dataset, there are instances where the precision of DvsD is higher than that of CvsD. This may be due to the fact that when noise is introduced into both the training and test data, the nature of continuous data causes these data points to increase or decrease together. As a result, there are instances where the DvsD mode can achieve better accuracy than the CvsD mode. Additionally, in the IoT dataset, noise in the test data (CvsD) has a more destructive effect than noise in the training data (DvsC). This suggests that it is better for the training data to have noise rather than the test data.

In the datasets BOT-IoT, CICIDS-2017, MQTT-IoT-IDS2020, and NSL-KDD, when both the training and test data sets are noisy (mode DvsD), the classification accuracy is always the lowest. Moreover, as the amount of noise variance increases, the accuracy of classification decreases further. However, in the IoT dataset, there are cases where the accuracy of classification may slightly improve with increased noise. This could be due to the fact that the IoT dataset contains continuous data points. By reducing or increasing noise in the data, the data points can be brought closer to or further away from the actual values they represent. Therefore, it is possible that the accuracy of classification can improve slightly in some cases when noise is added to the dataset. However, in general, noise has a negative impact on the accuracy of classification, and it is desirable to minimize noise in the dataset to achieve higher accuracy. The highest classification accuracy is achieved when both the training and test data sets are clean or without noise (mode CvsC). Following this, the highest accuracy is related to DvsC, and CvsD. in the IoT dataset, there are instances where the precision of DvsD is higher than that of CvsD.


IOT_dataset
CIC-IDS2017_dataset
BOT-IOT_dataset
IOT_dataset
IOT_dataset
MQTT-IOTIDS2020_dataset
CvsC_m
DvsC_m
CvsD_m
DvsD_m
0.05
0.10
0.15
0.50
0.52
0.54
0.56
0.58
0.60
0.62
0.64
0.66
0.78
0.80
0.82
0.84
0.86
0.88
0.90
0.92
0.94
0.500
0.525
0.550
0.575
0.600
0.625
0.650
0.675

a

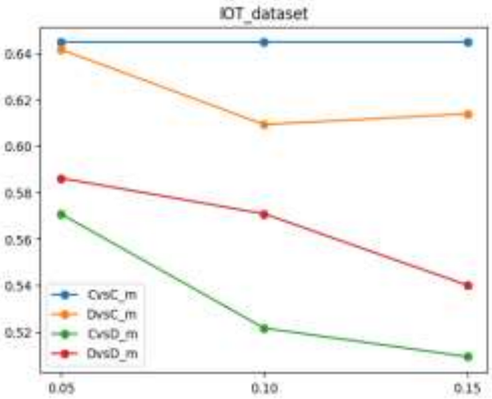
d

b

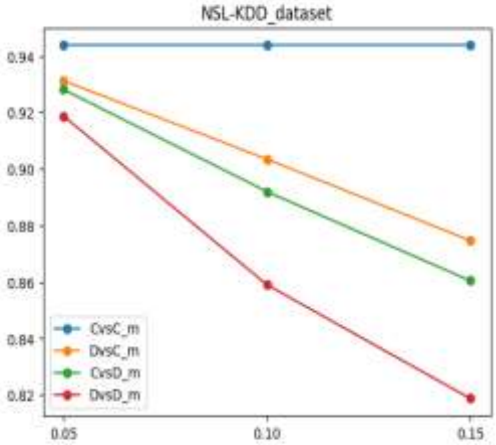
e

c

f



g



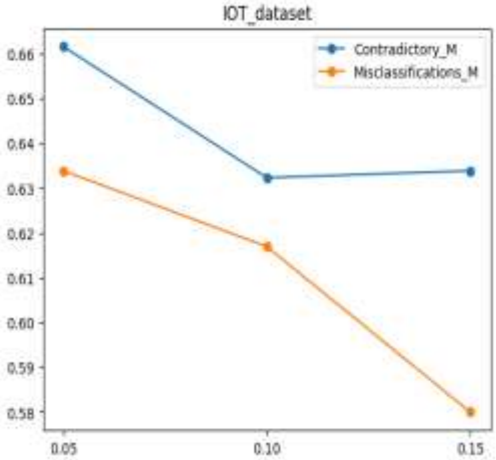
h

Figure 1. a) Feature noise on the dataset IoT b) Feature noise on the BOT-IoT dataset c) Feature noise on the dataset IoT d)Feature noise on the CICIDS-2020 dataset e) Feature noise on the dataset IoT f) Feature noise on the MQTT-IoT-IDS2020 datase g) Feature noise on the dataset IoT h) Feature noise on the NSL-KDD dataset.

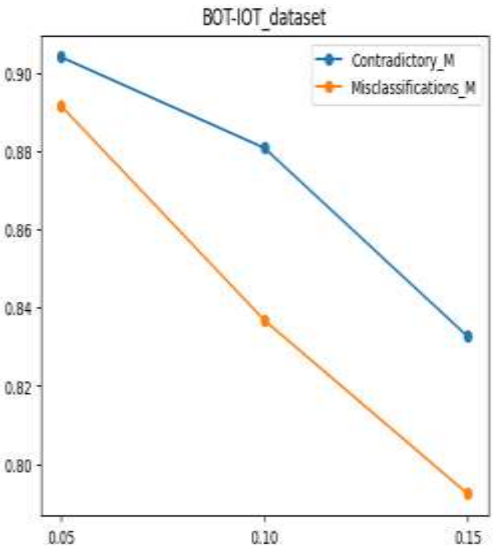
4.2. Analysis of effect of class noise

In our study, we created a graph to illustrate the relationship between classification accuracy and class noise. We measured class noise at three different levels, namely 5%, 10%, and 15%. The results were presented in the 9th, 11th, 13th, and 15th figures of the graphs which were related to the IoT dataset. Additionally, the second figure of the graphs was dedicated to the datasets BOT-IoT, CICIDS-2017, MQTT-IoT-IDS2020, and NSL-KDD. These graphs revealed that higher levels of class noise corresponded with lower accuracy of decision tree classification.

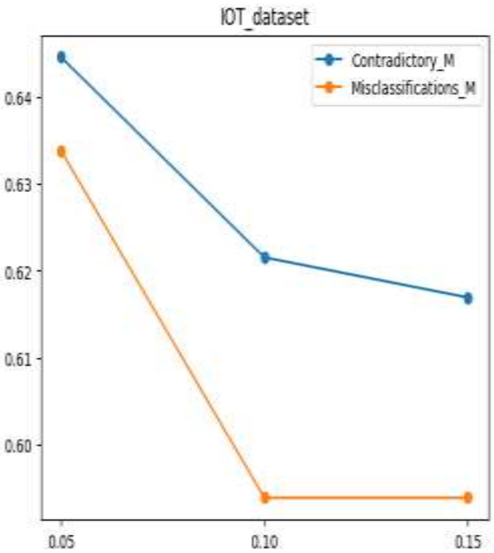
In the case of the IoT dataset, the accuracy was at its highest level when there was no noise present in either the training or test data (mode CvsC). However, the highest accuracy was achieved when the training data was clean, but the test data contained noise (mode DvsC). On the other hand, the presence of noise in the test data had a greater destructive impact on the accuracy of classification (mode CvsD) than noise in the training data (mode DvsD). It is worth noting that in some cases, increased noise in the IoT dataset resulted in slightly better accuracy of classification. This could be due to the fact that the data in the IoT dataset are continuous and noise can bring the data points closer to or further away from the actual values they represent. Therefore, our study highlights the importance of minimizing class noise in datasets to achieve higher accuracy of decision tree classification. By understanding the impact of noise on the accuracy of classification, we can optimize our data collection and cleaning methods to ensure high-quality results. The effect of noise on class label in different datasets has been analyzed and presented in Figure 2(i) to Figure 2(p) The accuracy of classification in the BOT-IoT, CICIDS-2017, MQTT-IoT-IDS2020, and NSL-KDD datasets is generally higher than that of the IoT dataset. This indicates that noise has a more significant impact on the IoT dataset. The reason behind this is that noise in the class label affects the selection of the best internal node, leading to an increase in the height of the tree and making it larger. This ultimately causes C4.5 to be negatively impacted by noise. It has been observed that the accuracy in both datasets decreases as the class label noise increases. Moreover, mislabeled data noise (misclassification) has a more destructive effect compared to the noise of conflicting data (contradictory examples), leading to an increase in the number of misclassified noisy data and a decrease in accuracy. In conclusion, the impact of noise on class label should be taken into account when applying the C4.5 decision tree algorithm for classification.



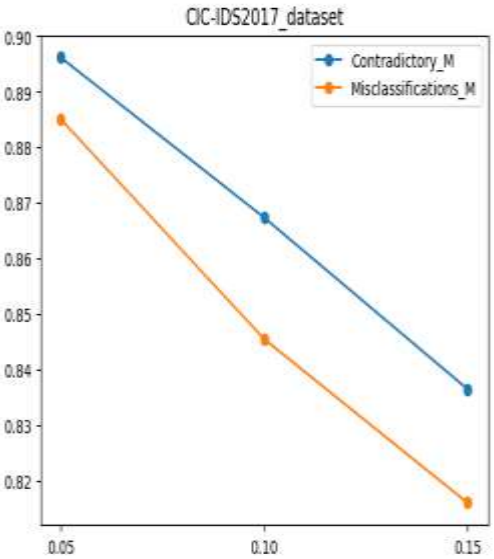
i


BOT-IOT_dataset
Contradictory_M
Misclassifications_M
0.90
0.88
0.86
0.84
0.82
0.80
0.05
0.10
0.15

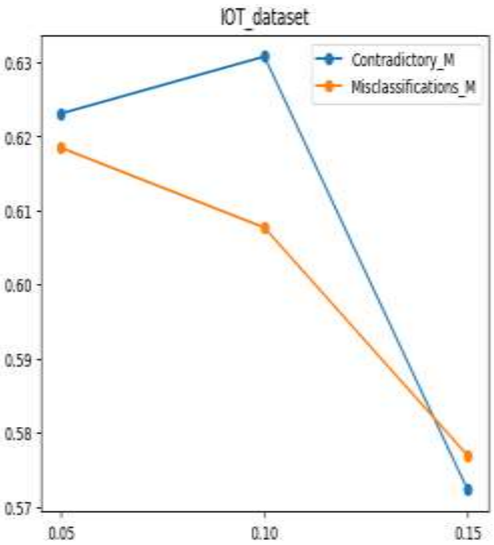
j


IOT_dataset
Contradictory_M
Misclassifications_M
0.64
0.63
0.62
0.61
0.60
0.05
0.10
0.15

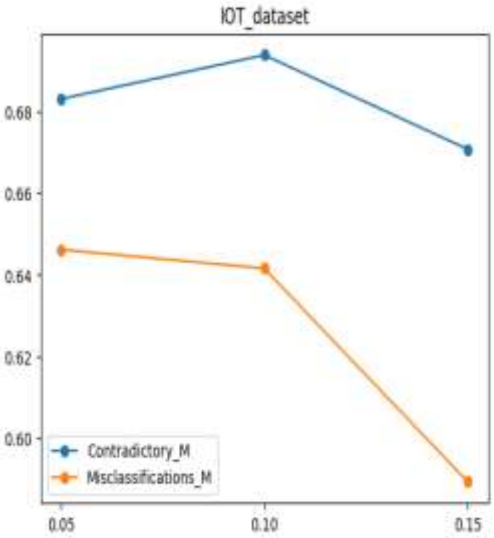
k


CIC-IDS2017_dataset
Contradictory_M
Misclassifications_M
0.90
0.89
0.88
0.87
0.86
0.85
0.84
0.83
0.82
0.05
0.10
0.15

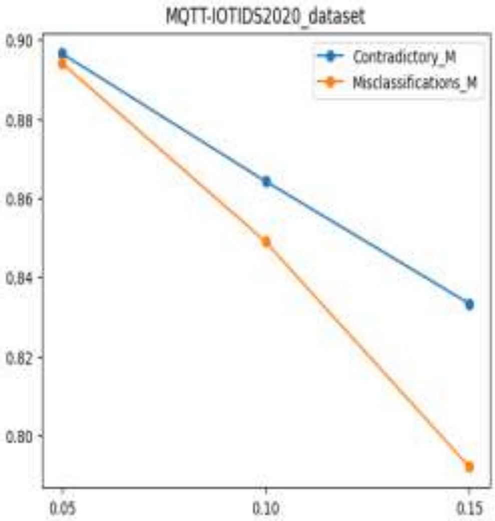
l


IOT_dataset
Contradictory_M
Misclassifications_M
0.63
0.62
0.61
0.60
0.59
0.58
0.57
0.05
0.10
0.15

m


IOT_dataset
Contradictory_M
Misclassifications_M
0.68
0.66
0.64
0.62
0.60
0.05
0.10
0.15

n


MQTT-IOTIDS2020_dataset
Contradictory_M
Misclassifications_M
0.90
0.88
0.86
0.84
0.82
0.80
0.05
0.10
0.15

o

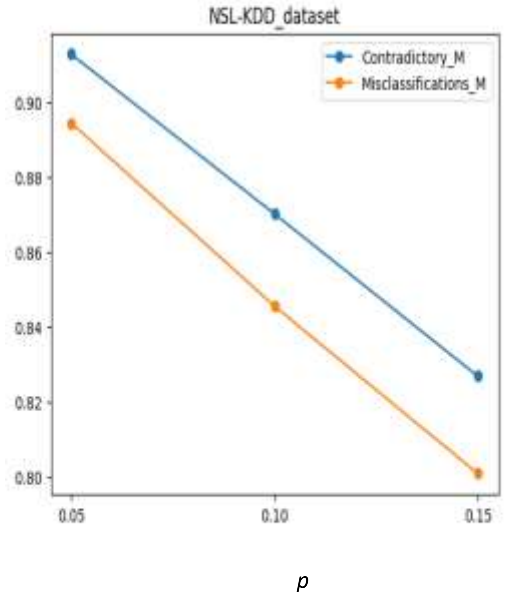


Figure 2. i) Class noise on the IoT dataset j) Class noise on the BOT-IoT dataset k) Class noise on the IoT dataset l) Class noise on the CICIDS2017 dataset m) Class noise on the IoT dataset n) Class noise on MQTT-IoT-IDS2020 dataset o) Class noise on the IoT dataset p) Class noise on the NSL-KDD dataset.

4.3. Discussion

The Internet of Things (IoT) has become a crucial part of the digital world as it offers immense opportunities for technological advancement, improved services, and enhanced management capabilities. With the increasing integration of technology in people's lives, IoT has become an essential tool for daily life. However, with the vast volume of data and users, it is crucial to ensure the security of IoT to protect against potential cyber-attacks. In this paper, we proposed a solution to detect network intrusion in IoT using three different deep neural network architectures. The study indicates that the proposed model can identify multiple cyber-attacks aimed at IoT devices, thus making it a more robust solution for detecting intrusion in IoT networks. Overall, the study highlights the importance of developing effective security solutions for IoT networks to protect the vast amount of data and users connected to them.

The effect of noise on class label in different datasets has been analysed and presented in the figures. The accuracy of classification in the BOT-IoT, CICIDS-2017, MQTT-IoT-IDS2020, and NSL-KDD datasets is generally higher than that of the IoT dataset. This indicates that noise has a more significant impact on the IoT dataset. The reason behind this is that noise in the class label affects the selection of the best internal node, leading to an increase in the height of the tree and making it larger. This ultimately causes C4.5 to be negatively impacted by noise, specifically data with contradictory examples, leading to an increase in the number of misclassified noisy data and a decrease in accuracy. In conclusion, the impact of noise on class label should be taken into account when applying the C4.5 decision tree algorithm for classification.

Moreover, it was also observed that mislabelled data noise has a more destructive effect compared to the noise of conflicting data. The noise of contradictory examples leads to an increase in the number of misclassified noisy data and subsequently decreases the accuracy of the classification. Therefore, it is crucial to reduce noise in the class label to ensure accurate classification results.

It has been observed that the accuracy in both datasets decreases as the class label noise increases. Moreover, mislabelled data noise (Misclassification) has a more destructive effect compared to the noise of conflicting examples. Increasing class label noise correlates with decreased accuracy in both datasets, emphasizing the critical need to address misclassification errors to maintain accurate classification results in IoT networks.

4.4. Discussion comparison

We proposed a solution to detect network intrusion in IoT using three different deep neural network architectures. The study indicates that the proposed model can identify multiple cyber-attacks aimed at IoT devices, thus making it a more robust solution for detecting intrusion in IoT networks. Overall, the study highlights the importance of developing effective security solutions for IoT networks to protect the vast amount of data and users connected to them.

We measured class noise at three different levels, namely 5%, 10%, and 15%. The results were presented in the 9th, 11th, 13th, and 15th figures of the graphs which were related to the IoT dataset. Additionally, the second figure of the graphs was dedicated to the datasets BOT-IoT, CICIDS-2017, MQTT-IoT-IDS2020, and NSL-KDD. These graphs revealed that higher levels of class noise corresponded with lower accuracy of decision tree classification.

The effect of noise on class label in different datasets has been analysed and presented in the figures. The accuracy of classification in the BOT-IoT, CICIDS-2017, MQTT-IoT-IDS2020, and NSL-KDD datasets is generally higher than that of the IoT dataset. The reason behind this is that noise in the class label affects the selection of the best internal node, leading to an increase in the height of the tree and making it larger. Furthermore, the findings demonstrate that the accuracy of decision tree classification decreases with an increase in noise variance.

This indicates that noise has a more significant impact on the IoT dataset. It has been observed that the accuracy in both datasets decreases as the class label noise increases. in our article compared to other similar methods with different methods, which we have shown in the table below.

Table6. Comparison between similar works

Article	Year	Dataset
Intrusion Detection Systems using decision trees and Support vector machines	2004	**KDD99**
Intrusion Detection system using decision tree algorithm	2012	**KDD99**
A decision tree classifier for intrusion detection priority tagging	2015	**ISCX Synthetic**
An anomaly intrusion detection system using C5 decision tree classifier	2018	**NSL-KDD**
Our proposed method	2023	**BOT-IOT** **CIC_IDS2017** **MQTT_IOTIDS2020** **NSL-KDD**

5. Conclusion

The use of decision trees can be an effective approach to detect network attacks in the Internet of Things network. However, the accuracy of the model can be affected by noise in the data, particularly in the class label. It is better to have clean training data and, if noise exists, it is better to have it in the training data rather than in the test data. The research also highlights the importance of carefully selecting the dataset and noise variances to achieve the best results. To improve this research work, future studies could consider using other machine learning algorithms in addition to decision trees, such as deep learning and ensemble methods. Moreover, incorporating more complex features or integrating domain knowledge could enhance the accuracy of the model. Additionally, more research work can be done on how to effectively deal with class label noise, as it can have a significant impact on the performance of the model. Finally, expanding the study to cover other datasets and a wider range of noise variances can provide more insight into the performance of IDSs in IoT networks.

Abbreviations

IoT	Internet of Things
IDS	Intrusion Detection Systems
CNN	convolutional neural network
LSTM	Long Short-Term Memory
ACC	Accuracy
GAN	Generative Adversarial Network
IoV	Internet of Vehicles
DTC	Decision Tree Classifier
EO-ANN	Equilibrium Optimization-based Artificial Neural Network
OICS-VFSL	Optimized Intra/Inter Class Structure based-Variational Few-Shot Learning
EML	Elite Machine Learning
SAE	Stacked Auto Encoders
PCA	Principal Component Analysis

References

[1] E. Gyamfi and A. D. Jurcut, “Novel online network intrusion detection system for industrial IoT based on OI-SVDD and AS-ELM,” *IEEE Internet of Things Journal*, vol. 10, no. 5, pp. 3827–3839, Mar. 2023, doi: 10.1109/JIOT.2022.3172393.

[2] X. Deng, J. Zhu, X. Pei, L. Zhang, Z. Ling, and K. Xue, “Flow topology-based graph convolutional network for intrusion detection in label-limited IoT networks,” *IEEE Transactions on Network and Service Management*, vol. 20, no. 1, pp. 684–696, Mar. 2023, doi: 10.1109/TNSM.2022.3213807.

[3] Y. Wu, L. Nie, S. Wang, Z. Ning, and S. Li, “Intelligent intrusion detection for internet of things security: A deep convolutional generative adversarial network-enabled approach,” *IEEE Internet of Things Journal*, vol. 10, no. 4, pp. 3094–3106, Feb. 2023, doi: 10.1109/JIOT.2021.3112159.

[4] J. Wu et al., “Joint semantic transfer network for IoT intrusion detection,” *IEEE Internet of Things Journal*, vol. 10, no. 4, pp. 3368–3383, Feb. 2023, doi: 10.1109/JIOT.2022.3218339.

[5] P. Ruzafa-Alcázar et al., “Intrusion detection based on privacy-preserving federated learning for the

industrial IoT," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 2, pp. 1145–1154, Feb. 2023, doi: 10.1109/TII.2021.3126728.

[6] J. Long, W. Liang, K.-C. Li, Y. Wei, and M. D. Marino, "A regularized cross-layer ladder network for intrusion detection in industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 2, pp. 1747–1755, Feb. 2023, doi: 10.1109/TII.2022.3204034.

[7] A. Oseni et al., "An explainable deep learning framework for resilient intrusion detection in IoT-Enabled transportation networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 1, pp. 1000–1014, Jan. 2023, doi: 10.1109/TITS.2022.3188671.

[8] Sk. T. Mehedi, A. Anwar, Z. Rahman, K. Ahmed, and R. Islam, "Dependable intrusion detection system for IoT: A deep transfer learning based approach," *IEEE Transactions on Industrial Informatics*, Vol. 19, No. 1, pp. 1006–1017, Jan. 2023, doi: 10.1109/TII.2022.3164770.

[9] S. Bebortta, S. K. Das, and S. Chakravarty, "Fog-enabled intelligent network intrusion detection framework for internet of things applications," in *13th international conference on cloud computing, data science & engineering (confluence)*, Jan. 2023, pp. 485–490. doi: 10.1109/Confluence56041.2023. 10048841.

[10] M. M. Alani and A. I. Awad, "An intelligent two-layer intrusion detection system for the internet of things," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 1, pp. 683–692, Jan. 2023, doi: 10.1109/TII.2022.3192035.

[11] J. Wu, H. Dai, Y. Wang, K. Ye, and C. Xu, "Heterogeneous domain adaptation for IoT intrusion detection: A geometric graph alignment approach," *IEEE Internet of Things Journal*, pp. 1–1, 2023, doi: 10.1109/JIOT.2023.3239872.

[12] A. Thakkar and R. Lohiya, "Attack classification of imbalanced intrusion data for IoT network using ensemble learning-based deep neural network," *IEEE Internet of Things Journal*, pp. 1–1, 2023, doi: 10.1109/JIOT.2023.3244810.

[13] A. A. M. Sharadqh, H. Hatamleh, A. M. A. Alnaser, S. S. Saloum, and T. A. Alawneh, "Hybrid chain: Blockchain enabled framework for bi-level intrusion detection and graph-based mitigation for security provisioning in edge assisted IoT environment," *IEEE Access*, pp. 1–1, 2023, doi: 10.1109/ACCESS.2023.3256277.

[14] Z. Ma, L. Liu, W. Meng, X. Luo, L. Wang, and W. Li, "ADCL: Towards an adaptive network intrusion detection system using collaborative learning in IoT networks," *IEEE Internet of Things Journal*, pp. 1–1, 2023, doi: 10.1109/JIOT.2023.3248259.

[15] I. A. Kandhro et al., "Detection of real-time malicious intrusions and attacks in IoT empowered cybersecurity infrastructures," *IEEE Access*, vol. 11, pp. 9136–9148, 2023, doi: 10.1109/ACCESS.2023.3238664.

[16] A. Telikani, J. Shen, J. Yang, and P. Wang, "Industrial IoT intrusion detection via evolutionary cost-sensitive learning and fog computing," *IEEE Internet of Things Journal*, vol. 9, no. 22, pp. 23260–23271, Nov. 2022, doi: 10.1109/JIOT.2022.3188224.

[17] O. Abdel Wahab, "Intrusion detection in the IoT under data and concept drifts: Online deep learning approach," *IEEE Internet of Things Journal*, vol. 9, no. 20, pp. 19706–19716, Oct. 2022, doi: 10.1109/JIOT.2022.3167005.

[18] W. Liang, Y. Hu, X. Zhou, Y. Pan, and K. I.-K. Wang, "Variational few-shot learning for microservice-oriented intrusion detection in distributed industrial IoT," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 8, pp. 5087–5095, Aug. 2022, doi: 10.1109/TII.2021.3116085.

[19] X. Zhou, W. Liang, W. Li, K. Yan, S. Shimizu, and K. I.-K. Wang, "Hierarchical adversarial attacks against graph-neural-network-based IoT network intrusion detection system," *IEEE Internet of Things Journal*, vol. 9, no. 12, pp. 9310–9319, Jun. 2022, doi: 10.1109/JIOT.2021.3130434.

[20] T. M. Booij, I. Chiscop, E. Meeuwissen, N. Moustafa, and F. T. H. den Hartog, "ToN_IoT: The Role of Heterogeneity and the Need for Standardization of Features and Attack Types in IoT Network Intrusion Data Sets," *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 485–496, Jan. 2022, doi: 10.1109/JIOT.2021.3085194.

[21] M. Zeeshan et al., "Protocol-based deep intrusion detection for DoS and DDoS attacks using UNSW-NB15 and bot-IoT data-sets," *IEEE Access*, vol. 10, pp. 2269–2283, 2022, doi: 10.1109/ACCESS.2021.3137201.

[22] H. Siddharthan, T. Deepa, and P. Chandhar, "SENMQTT-SET: An intelligent intrusion detection in IoT-MQTT networks using ensemble multi cascade features," *IEEE Access*, vol. 10, pp. 33095–33110, 2022, doi: 10.1109/ACCESS.2022.3161566.

[23] M. S. A. Muthanna, R. Alkanhel, A. Muthanna, A. Rafiq, and W. A. M. Abdullah, "Towards SDN-Enabled, intelligent intrusion detection system for internet of things (IoT)," *IEEE Access*, vol. 10, pp. 22756–22768, 2022, doi: 10.1109/ACCESS.2022.3153716.

[24] C. Miranda, G. Kaddoum, A. Boukhtouta, T. Madi, and H. A. Alameddine, "Intrusion prevention scheme against rank attacks for software-defined low power IoT networks," *IEEE Access*, vol. 10, pp. 129970–129984, 2022, doi: 10.1109/ACCESS.2022.3228170.

[25] P. L. S. Jayalaxmi, R. Saha, G. Kumar, M. Conti, and T.-H. Kim, "Machine and deep learning solutions for intrusion detection and prevention in IoTs: A survey," *IEEE Access*, vol. 10, pp. 121173–121192, 2022, doi: 10.1109/ACCESS.2022.3220622.

[26] Y. Yang, K. Zheng, C. Wu, and Y. Yang, Improving the classification effectiveness of intrusion detection by using improved conditional variational autoencoder and deep neural network, *Sensors*, vol. 19, pp02528, 2019.

[27] M. Yadollahzadeh Tabari; Z. Mataji, Detecting Sinkhole Attack in RPL-based Internet of Things Routing Protocol, *Journal of AI and Data Mining*, vol. 9, no. 1, January 2021, Pages73-85.

[28] L. khalvati; M. Keshtgary; N. Rikhtegar, Intrusion Detection based on a Novel Hybrid Learning Approach, *journal of AI and Data Mining*, Volume 6, Issue 1 , March 2018, Pages 157-162, doi.org/10.22044/jadm.2017.979 .