Journal of Artificial Intelligence and Data Mining (JAIDM), Vol. 12, No. 1, 2024, 37-55.

Shahrood University of Technology

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

Research paper

Intrusion Detection for IoT Network Security with Deep Learning

Roya Morshedi[1], S. Mojtaba Matinkhah[*1] and Mohammad Taghi Sadeghi[2]

1. Department of Computer Engineering, *Yazd University, Yazd, Iran.*
2. Department of Electrical Engineering, Yazd University, Yazd, Iran.

Article Info

Article History:
Received 06 September 2023
Revised 12 October 2023
Accepted 10 November 2023

DOI:10.22044/jadm.2023.13539.2471

Keywords:
Neural Networks, Internet of Things, DDoS, CICIDS2017, Dense Network, LSTM, CNN, Deep Learning Models.

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

Abstract

Intrusion Detection Systems (IDSs) are critical components for safeguarding IoT networks against cyber threats. This study presents an advanced approach to IoT network intrusion detection, leveraging deep learning techniques and pristine data. We utilize the publicly available CICIDS2017 dataset, which enables comprehensive training and testing of intrusion detection models across various attack scenarios, such as Distributed Denial of Service (DDoS) attacks, port scans, and botnet activity. Our goal is to provide a more effective method than the previous methods. Our proposed deep learning model incorporates dense transition layers and LSTM architecture, designed to capture both spatial and temporal dependencies within the data. We employed rigorous evaluation metrics, including sparse categorical cross-entropy loss and accuracy, to assess the model performance. The results of our approach show outstanding accuracy, reaching a peak of 0.997 on the test data. Our model demonstrates stability in loss and accuracy metrics, ensuring reliable intrusion detection capabilities. Comparative analysis with other machine learning models confirms the effectiveness of our approach. Moreover, our study assesses the model's resilience to Gaussian noise, revealing its capacity to maintain accuracy in challenging conditions. We provide detailed performance metrics for various attack types, offering insights into the model's effectiveness across diverse threat scenarios.

1. Introduction

IoT technology holds the potential to enhance and facilitate personal, professional, and societal aspects of our lives. Furthermore, there are numerous cost-effective IoT devices available, catering to a diverse range of users who may not possess extensive technological knowledge. Unfortunately, this accessibility and widespread usage also render IoT vulnerable to cyber-attacks. Endpoint devices in the IoT ecosystem, such as home security cameras and appliances, are particularly susceptible to cyber-attacks within the network. These devices possess limited computational power, storage capacity, and network capabilities compared to more complex endpoints like routers, smartphones, and laptops [2]. Attackers exploit the vulnerabilities present in a significant number of IoT devices, enabling them to execute large-scale assaults on internet resources. [4]-[7]. DDoS attacks occur when multiple compromised machines or devices flood a targeted machine or device with requests, overwhelming its server and causing it to crash. DDoS attacks are highly effective, as even highly configured machines experience degraded performance, exemplified by attacks on cloud services and virtual machines [8]. Moreover, IoT devices connected to 5G networks can also become targets of DDoS attacks [11]. Many IoT devices lack the capability to detect such cyber-attacks [12].

In summary, while IoT technology offers numerous benefits and conveniences, it also presents significant cybersecurity challenges. Securing IoT devices and networks is essential to prevent them from being exploited in large-scale cyberattacks, such as DDoS attacks orchestrated through compromised IoT devices. Efforts to improve IoT security should involve manufacturers, users, and service providers working together to mitigate these risks.

These attacks typically exploit open ports on IoT operating systems [10]. Furthermore, IoT devices connected through 5G networks are also susceptible to DDoS attacks [14]. Due to inherent limitations, many IoT devices lack the capability to detect and mitigate such cyber-attacks [12].

It's worth noting that while achieving high accuracy is promising, the model's performance in real-world scenarios may vary due to factors such as evolving attack techniques, noisy network data, and adversarial attacks. Regular updates and continuous monitoring are essential to ensure the model remains effective over time. Additionally, considering the computational and resource constraints of IoT devices, model optimization and efficient deployment are critical considerations. To address these challenges, we developed a deep neural network-based intrusion detection model using the CICIDS-2017 dataset. Our proposed model is capable of detecting DDoS attacks, along with other cyber-attacks targeting IoT networks. Specifically, we employed a convolutional neural network (CNN) model and a hybrid CNN and Long Short-Term Memory (LSTM) model to detect DDoS attacks. Through our experiments, we found that Dp-model demonstrated superior accuracy, achieving 99.77% accuracy on our test data. The proposed model can be integrated into an Intrusion Detection System (IDS), providing an additional layer of security to mitigate various threats.

Figure 1 depicts the network architecture utilized in this research work. The objectives of this study are as follows:

1. Propose a deep learning model for detecting DDoS attacks and other cyber-attacks targeting IoT networks.
2. Evaluate the performance of the proposed model using the CICIDS-2017 dataset.
3. Compare the performance of the proposed model with selected machine learning algorithms.

By accomplishing these goals, we aim to contribute to the advancement of intrusion detection systems and bolster cybersecurity measures within IoT networks.

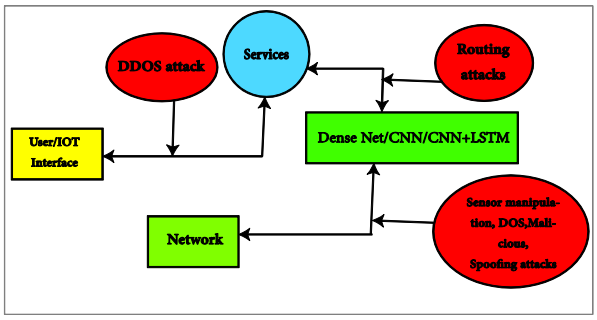


Figure 1. IoT network architecture

2. Related Works

This section presents a comparative analysis of the recent papers that focus on enhancing security in Internet of Things (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 considering the importance of deep learning models in intrusion detection in IoT networks. Our research project presents an intrusion detection model to identify distributed denial of service attacks along with several other cyber attacks in IoT networks using the CICIDS-2017 dataset.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. The need for further research to develop effective and robust IDSs for IoT networks is emphasized, as highlighted in Table 1.

Gyamfi and Jurcut [1] propose a lightweight NIDS based on an online incremental support vector data description (OI-SVDD) anomaly detection system on the industrial IoT devices and an adaptive sequential extreme learning machine (AS-ELM) on the multiaccess edge computing (MEC) server. However, the article by Gyamfi and Jurcut has several shortcomings. Firstly, it only focuses on IIoT and does not generalize to other IoT networks, limiting its applicability. Secondly, the argument that conventional signature-based NIDS are not suitable for IIoT network security is not entirely accurate as they can be updated with new signatures. Additionally, the proposed solution based on OI-SVDD and AS-ELM may be complex and difficult to implement in practice, especially for resource-constrained IIoT devices. Finally, the

evaluation of the proposed NIDS is limited to two datasets, which limits the generalizability of the results.

Deng et al. [2] propose a novel approach for label-limited network intrusion detection in IoT networks using Flow Topology based Graph Convolutional Networks (FT-GCNs). The authors suggest that traditional machine learning based NIDS approaches require a large amount of labeled traffic flow data, which hinder their application in highly dynamic IoT networks with limited labeling. However, the proposed FT-GCN leverages the underlying traffic flow patterns through a flow topological structure to unlock the full potential of the traffic flow data with limited labeling.

Wu et al. [3] propose an intelligent intrusion detection algorithm based on big data mining, fuzzy rough set, generative adversarial network (GAN), and convolutional neural network (CNN). Their method aims to address the challenges of implementing big data-enabled intrusion detection algorithms on resource-limited edge nodes.

While Wu et al. [4] propose a method for effective intrusion detection in large-scale, scarcely labeled IoT domains.Wu et al. [4] 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.

Ruzafa-Alcázar et al. [5] focus specifically on evaluating differential privacy techniques for federated learning in the context of an intrusion detection system for industrial IoT. While this is a valuable contribution, it focuses on the specific context of industrial IoT and federated learning, our study has broader implications for IDS accuracy in IoT networks more generally. Furthermore, their evaluation is limited to a single dataset (ToNIoT) and does not provide insights into the generalizability of the proposed approach across different datasets. Additionally, the article does not discuss the potential trade-offs between accuracy and privacy in the proposed approach, which is a crucial factor in the development of IDSs for IoT networks.

While Long et al. [6] propose a novel approach for intrusion detection in IoT networks, their study fails to compare their proposed approach with existing intrusion detection systems. Additionally, the study only evaluates their model on a single dataset, which limits the generalizability of their findings to other datasets and real-world settings.

Oseni et al. [7] focus specifically on the Internet of Vehicles (IoV) rather than the broader Internet of Things (IoT) network. This may limit the generalizability of the findings to other types of IoT networks, such as those used in industrial or home settings. 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.

The article by Mehedi et al.[8] emphasizes the importance of attribute selection in identifying normal and attack scenarios with a small amount of labeled data, which is crucial in real-world scenarios where obtaining labeled data can be challenging.

Bebortta et al. [9] propose a fog-enabled intelligent network intrusion detection framework for internet of things applications. First, while the Equilibrium Optimization-based Artificial Neural Network (EO-ANN) model proposed in their study shows promising results in detecting network attacks in IoT systems, it is unclear how the model performs in the presence of noisy data, which is a common challenge in real-world settings.

Alani and Awad [10] propose an intelligent two-layer intrusion detection system for IoT which can be deployed in an IoT network without affecting the normal operation of the devices and applications. This enables organizations to enhance the security of their IoT networks without incurring significant costs or disrupting existing operations.

Wu et al. [11] article relies heavily on transferring knowledge from a data-rich domain (network intrusion detection) to a data-scarce domain (IoT intrusion detection) using a complex graph alignment method. This may introduce additional complexity and potential sources of error in the intrusion detection process, particularly if the alignment is not perfect

Thakkar and Lohiya [12] focus primarily on addressing the issue of class imbalance in intrusion detection datasets using an ensemble learning approach, rather than evaluating the accuracy of the intrusion detection system itself. While addressing class imbalance is certainly an important consideration for developing a coherent and potent intrusion detection and classification system, it is also important to evaluate the accuracy of the system in detecting and preventing intrusions. The article by Thakkar and Lohiya does mention that the performance of the proposed approach is evaluated using various evaluation metrics, including accuracy, precision, recall, f-score, and False Positive Rate (FPR), but it would be important to assess how well the proposed

approach performs in comparison to other existing intrusion detection systems, especially those that have also addressed the issue of class imbalance.

While Sharadqh et al. [13] propose an interesting approach to intrusion detection in IoT networks, their work appears to be focused on a very specific solution involving the use of blockchain and optimization algorithms. This limits the generalizability and applicability of their approach to different network environments and may not address the root cause of the problem, which is the need for accurate intrusion detection. Additionally, their proposed framework involves a complex process that may result in high computational costs and may not be scalable in larger networks.

Ma et al. [14] propose a collaborative learning-based intrusion detection framework called ADCL for IoT networks. While the proposed framework seems to address the limitations of a single model and improve detection performance, the article is unclear how it mitigates the limitations of a single model. Additionally, the article does not provide a detailed evaluation of the proposed framework, which makes it difficult to assess the effectiveness of the proposed approach.

By using a deep learning approach, Kandhro et al. [15] demonstrate a significant performance increase in terms of accuracy, reliability, and efficiency in detecting all types of attacks. Additionally, the use of a generative adversarial network for intrusion detection is a novel approach that could have implications for improving network security in the future. However, the use of a deep learning-based approach may require a large amount of labeled data to achieve high accuracy.

Telikani et al. [16] present an innovative approach to address the problem of imbalanced data distribution in Industrial IoT environments for intrusion detection. Limited number of malicious activities compared to normal activities causes machine learning models being biased towards the majority class (normal activities) and therefore failing to detect rare malicious activities. The authors propose a hybrid model of stacked autoencoders (SAE) and convolutional neural networks (CNNs) with a new cost-dependent loss function, called EvolCostDeep, to optimize the model's parameters. They also introduce a fog computing-enabled framework, called DeepIDSFog, to parallelize the EvolCostDeep model and mitigate attacks in IIoT environments. However, the article does not provide a detailed comparison with other state-of-the-art intrusion detection systems, making it difficult to evaluate the effectiveness of the proposed EvolCostDeep model and DeepIDSFog framework in comparison to existing solutions.

Abdel Wahab [17] discusses the challenges of maintaining the accuracy of machine learning-based intrusion detection systems in dynamic environments, specifically in the IoT network. The author proposes a solution that involves a drift detection technique using principal component analysis (PCA) to detect data and concept drifts, an online outlier detection technique to identify outliers, and an online deep neural network (DNN) that adjusts the sizes of hidden layers based on the hedge weighting mechanism. However, 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. Moreover, 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 it is important to address drifts in dynamic IoT environments, it is also important to consider other factors that can impact the effectiveness of an IDS.

The article by Liang et al. [18] gives the idea to offer a promising solution to the problem of imbalanced learning in microservice-oriented intrusion detection in distributed IoT systems, however, more information is needed to fully evaluate the effectiveness of the proposed approach. For example, 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 optimized intra/inter-class-structure-based variational few-shot learning (OICS-VFSL) model was compared to the baseline methods.

Zhou et al. [19] article focuses on the development of an attack generation method for testing intrusion detection systems in IoT networks. They also introduce a new method for generating adversarial examples for intrusion detection systems in IoT networks. While this method may be effective in testing the robustness of these systems, it also raises concerns about the potential for attackers to use similar methods to bypass these systems in real-world attacks. Therefore, it could be argued that the Zhou et al. article may actually undermine the effectiveness of intrusion detection systems in

IoT networks by making it easier for attackers to evade detection.

Booij et al. [20] raise important points about the importance of data sets and standardization efforts in IoT security research, our article provides a more specific and actionable contribution to the field by proposing and evaluating a concrete intrusion detection method for IoT networks.

Zeeshan et al. [21] proposed architecture uses a deep learning technique to achieve high accuracy, they do not address the issue of noisy data or the impact of noise on classification accuracy. This is a significant limitation as noisy data is a common issue in IoT networks, and it can significantly impact the performance of an IDS

Siddharthan et al. [22] proposes an intrusion detection system for IoT networks that uses Elite Machine Learning (EML) algorithms and a lightweight protocol to manage time constraints. The article claims to achieve an accuracy of above 99% for the considered system model.

Muthanna et al. [23] propose an intelligent and efficient framework for threat detection in IoT environments, leveraging Cuda Long Short Term Memory Gated Recurrent Unit (cuLSTMGRU) and Software-Defined Networking (SDN) technologies. The proposed model achieved a high detection accuracy of 99.23% with a low false-positive rate, outclassing other models and benchmark algorithms in terms of speed efficiency, detection accuracy, precision, and other standard evaluation metrics. The study used a state-of-the-art IoT-based dataset and standard evaluation metrics, and employed 10-fold cross-validation to ensure unbiased results. However, the article does not clearly state which dataset was used for evaluation and how it was collected.

Miranda et al. [24] article focuses on a specific optimization mechanism for preventing rank attacks in SDN-based deployments of 6LoWPAN networks. They discuss the use of reinforcement learning (RL) to complement an SDN controller in achieving cost-efficient route optimization and QoS provisioning to prevent rank attacks in low-power IoT networks. The use of RL is a novel approach to address the issue of non-optimized routes for packet forwarding in the face of rank attacks.

Jayalaxmi et al. [25] provide a useful survey of existing studies on intrusion detection and prevention systems in IoT networks. The article mainly presents a mapping technique for risk factor analysis and proposes a hybrid framework for security model development. Moreover, it provides a comparative analysis of various AI-based techniques, tools, and methods used for intrusion detection and prevention in IoT networks. In contrast, our study presents a clear research question and methodology for evaluating the accuracy of an intrusion detection system in IoT networks.

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 comparison with deep learning-based approach	IoT networks	The use of a deep learning-based approach for intrusion detection may require a large amount of 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	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.
Jayalaxmi et al. [25]	Mapping technique for risk factor analysis and a hybrid framework for security model development	Intrusion detection and prevention in IoT networks	The article does not provide empirical results on the accuracy of an IDS in IoT networks. For some types of threats, the algorithms have shown a lack of detection. Yet, the algorithm's performance has not been tested on a live network.

3. Data sets and Methodology

3.1. Datasets

The CICIDS2017 dataset consists of both benign network traffic and malicious network traffic, allowing researchers to train and test their IDS on a diverse range of attack scenarios. This paper used a publicly available dataset MachineLearn-ingCSV, a piece of the CICIDS-2017 dataset from ISCX Consortium. It consists of eight real-world traffic monitoring sessions in a comma-separated value (CSV) file. This dataset was collected and distributed by researchers from the Canadian Institute of Cyber Security. It contains up-to-date

attacks containing both benign and malicious network traffic traces. It is a labeled dataset including 84 features. Very last feature of the dataset is the class label, which classifies the sample as an attack or benign traffic. There are 14 kinds of attacks in this dataset. Some of the attack types are significant in aspect IoT detection. The attacks included in the dataset range from botnet attacks to DoS and DDoS attacks, port scans, and more. 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 researchers to gain insight into the characteristics of the traffic. The availability of the CICIDS2017 dataset has been a significant development in the field of cybersecurity research, allowing 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. CICIDS2017 contains more than 80 million labeled flows with 84 features. the CICIDS2017 dataset indeed includes both benign and malicious network traffic, providing a comprehensive set of data for researchers to develop, train, and evaluate intrusion detection systems (IDS) and security algorithms. This diversity allows for testing the effectiveness of IDS in identifying various types of attacks and helps in creating more robust and reliable security solutions.The CICIDS2017 (Canadian Institute for Cybersecurity Intrusion Detection System 2017) dataset is a widely used benchmark dataset in the field of cybersecurity and intrusion detection. It was created to support research and development in the area of network security and intrusion detection systems. The dataset contains a diverse set of network traffic data, including both benign and malicious traffic, captured in a controlled environment.

Key features of the CICIDS2017 dataset include:

Variety of Attacks: The dataset covers a wide range of cyber attacks, including but not limited to DoS (Denial of Service), DDoS (Distributed Denial of Service), port scanning, brute force attacks, and more. This variety allows for comprehensive testing and evaluation of intrusion detection systems.

Realistic Traffic: The dataset includes real-world network traffic, making it more representative of actual network environments. This helps researchers in developing and testing intrusion detection systems under conditions that closely resemble those encountered in practice.

Detailed Attributes: The dataset provides detailed attributes and features for each network flow, such as source and destination IP addresses, port numbers, protocol types, and various statistical features derived from the traffic data.

Large Scale: The dataset is relatively large in scale, containing a substantial number of network flow records, which is beneficial for training and evaluating machine learning and data mining algorithms.

Researchers and practitioners in the field of cybersecurity use the CICIDS2017 dataset for various purposes, including training machine learning models, evaluating the performance of intrusion detection systems, and benchmarking new intrusion detection techniques. Its availability has contributed to advancements in the development of more effective and robust security solutions to combat evolving cyber threats.

The CICIDS2017 dataset is a comprehensive collection of network traffic data designed for research and development in the field of cybersecurity, particularly in the area of intrusion detection systems (IDS).

Data Preprocessing: The dataset has undergone preprocessing to remove noise and irrelevant data, making it suitable for training and testing machine learning models and intrusion detection algorithms.

Research and Development: Researchers and practitioners in the cybersecurity domain use the CICIDS2017 dataset to develop and evaluate intrusion detection systems, test the effectiveness of security algorithms, and benchmark the performance of various detection techniques.

Overall, the CICIDS2017 dataset serves as a valuable resource for studying network security, evaluating intrusion detection techniques, and enhancing the resilience of systems against cyber threats. Its comprehensive nature and diverse range of features make it a widely used benchmark for cybersecurity research and development. In this dataset, the network flow data has been recorded for five days, the general information of which is:

- First day: includes normal activities
- Second day: includes normal and abnormal activities of Brute Force, FTP-Patator and SSH-Patator.
- Third day: includes normal and abnormal activities of DDOS, DoS slowloris, DoS Slowhttptest, DoS Hulk, DoS GoldenEye, Heartbleed Port 444.

• Fourth day: includes normal and abnormal activities Web Attack - Brute Force, Web Attack - XSS, Web Attack - Sql Injection, Infiltration - Dropbox download, Infiltration - Cool disk - MAC, Infiltration - Dropbox download.

• Fifth day: includes normal and abnormal activities of Botnet ARES, Port Scan, DDoS LOIT.

Table 2. Description of the CIC-IDS2017 dataset

File name	Available attack	Count
Mondey-WorkingHours.pcap ISCX.csv	Benign	529,918
Tuesday-WorkingHours.pcap ISCX.csv	Benign	432,074
	FTP-Patator	7,938
	SSH-Patator	5,897
Wednesday-WorkingHours.pcap ISCX.csv	Benign	440,031
	DoS GoldenEye	10,293
	DoS Hulk	231,073
	DoS Slowhttptest	5,499
	DoS slowloris	5,796
	Heartbleed	11
Thursday-WorkingHours-Morning-WebAttacks.pcap ISCX.csv	Benign	168,186
	Web Attack – Brute Force	1,507
	Web Attack – Sql Injection	21
	Web Attack – XSS	652
Thursday-WorkingHours-Afternoon-Infilteration.pcap ISCX.csv	Benign	288,566
	Infiltration	36
Friday-WorkingHours-Morning.pcap ISCX.csv	Benign	189,067
	Bot	1,966
Friday-WorkingHours-Afternoon-PortScan.pcap ISCX.csv	Benign	127,537
	PortScan	158,930
Friday-WorkingHours-	Benign	97,718
	DDoS	128,027

3.2. Methodology

The dataset utilized in this study was collected across various time segments. Despite the temporal variations, the features remained consistent throughout the data collection process. In order to consolidate the available data from multiple timelines, all the collected information was merged. Upon examination, it was observed that 1362 rows within the dataset contained "NaN" values, which accounted for less than one percent of the entire dataset. Consequently, these instances were eliminated from further analysis as they included null and NaN values. For the purpose of training our models, the dataset was randomly divided into an 80% training set and a 20% testing set. Additionally, a random validation split of 10% was introduced within the training dataset to facilitate model assessment and fine-tuning. To prepare the dataset for utilization in deep learning models, categorical variables were encoded into integer values. This encoding process was accomplished using a label encoder that assigned integer values ranging from 0 to 14 to the different categories. While categorical variable encoding was conducted within the deep learning models themselves, prior to commencing the training process, the dataset underwent standardization through scalar transformation. The standardization procedure employed the StandardScaler method, which eliminates the mean and scales the data to achieve unit variance. However, it is important to note that outliers may influence the calculation of both the empirical mean and standard deviation, subsequently constricting the range of characteristic values. These divergences in the initial feature distributions have the potential to cause difficulties for numerous machine learning models.One particular issue arising from disparate feature scales is encountered when employing distance-based models. If a given feature exhibits a wide range of values, it will disproportionately govern the resulting distance calculations. This discrepancy can introduce bias and adversely affect the performance of the model. To mitigate this potential problem, data standardization is necessary, wherein the data is transformed to possess a mean (μ) of 0 and a standard deviation (σ) of 1. Such standardization ensures that variables with disparate scales do not exert disproportionate influence on model fitting and the associated learning algorithm. Consequently, prior to integrating the data into a machine learning model, it is customary to standardize the dataset to achieve the desired properties ($\mu = 0$, $\sigma = 1$). The preprocessing consisted of three sub-steps: scaling, normalization, and data cleaning. Then, the dataset was labeled. The next classification step was performedusing CNN, CNN- LSTM, and DenseNet. This step adopts different models to predict the attack. We classification step used three different types of neural networks to classify the attacks. These NNs were CNN, LSTM, and DenseNet. We employed TensorFlow and Keras to implement CNN, CNN-LSTM, and DenseNet and Python to implement the neural network models. Afterward, we trained, tested, and evaluated our model.

4. Results

In this section, we have reviewed the results of the tests, first we have reported the three main networks and then we have expressed our proposed method.

4.1 Implementation of algorithms

4.1.1 CNN Implementation

To compare the effectiveness of the experimental flow, a Convolutional Neural Network (CNN) was set up with similar layer parameters as the DenseNet implementation. The CNN started with an input shape of (9 x 9 x 1). To match the parameter size of 20,943, 3 extra columns were added as padding with 0 values. Dense transition layers were added with sizes (120 x 2), (60 x 3), and (30 x 4). The activation layers had 15 parameters, similar to DenseNet. The transition layers used the ReLU activation function, and the final classification layer used SoftMax. The loss and accuracy metrics were evaluated using sparse categorical cross-entropy. The Adam optimizer was used as well. The batch size for training the CNN was set to 1024, and the number of epochs was set to 120. The validation split, where a portion of the data is used for validation, was set to 90:10 for all experiments. By setting up this experiment, the goal is to compare the outcome and effectiveness of the CNN architecture with the DenseNet architecture. The parameters and settings were adjusted to match as closely as possible to ensure a fair comparison. Here is the pseudo-code for the GetCNNDPProcess() function:

```
    def GetCNNDPProcess(self):
        ReadData(1)
        data_features = ds_data[:, 0:81]
        data_classes = ds_data[:, 81:82]
        data_classes = np_utils.to_categorical(data_classes)
        scaler = StandardScaler()
        data_features = scaler.fit_transform(data_features)
        data_features = data_features.reshape(len(data_features), 9, 9, 1)
```

```
features_train, features_test, classes_train, classes_test = train_test_split(data_features, data_classes, test_size=0.2, random_state=4, shuffle=True)

model = Sequential()
model.add(tf.keras.Input(shape=(9, 9, 1)))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(Conv2D(120, 2))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(Conv2D(60, 3))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(Conv2D(30, 4))
model.add(Flatten())
model.add(Dense(data_classes.shape[1], activation='softmax'))
model.summary()

model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy', 'mse'])

history = model.fit(features_train[0:5000, :], classes_train[0:5000, :], epochs=self.epoch, validation_data=(features_test[0:100, :], classes_test[0:100, :]))

GetPlot('CNN Model Accuracy', history.history['accuracy'], history.history['val_accuracy'], 'accuracy')
GetPlot('CNN Model Loss', history.history['loss'], history.history['val_loss'], 'loss')
```

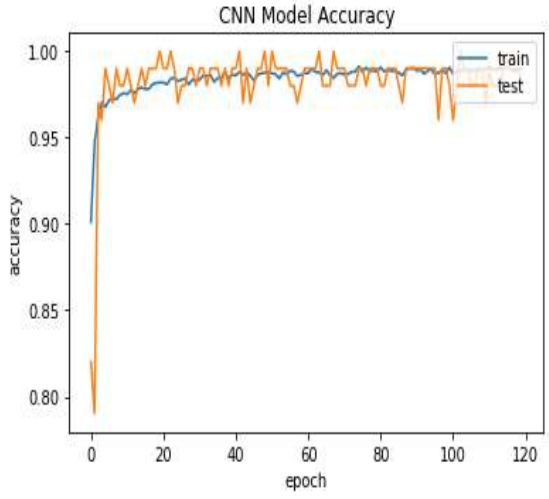


Figure 2. Accuracy graph of CNN.

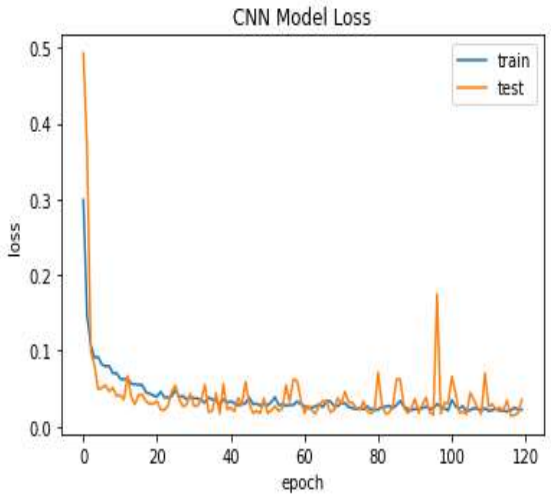


Figure 3. Loss gragh of CNN.

4.1.2 Dense net implementation

DenseNet is a deep learning architecture that consists of multiple dense blocks. Each dense block contains different filters, but the dimensions within each block are similar. The transition layer is used to downsample the feature maps and applies batch normalization. In this specific implementation, the input shape is (78 x 1). Three transition layers are used with filter sizes of 128, 64, and 32 sequentially. These transition layers are connected using the ReLU activation function. After the transition layers, a classification layer with 15 outputs is added, followed by a SoftMax function for prediction. The dense blocks in this implementation have filter sizes of 1 x 1 and 3 x 3. The second layer receives a total of 79 parameters (78 predictors and 1 target), and the third layer has a total of 10112 parameters (78+1 x 128). Including the classification layer, the total number of parameters is 20,943 (0+10112+8256+2080+495). The Adaptive Moment Optimizer (Adam) is used as the optimizer for training the model. Adam adjusts the learning rate based on the exponential moving average of the gradient and squared gradient. Sparse categorical cross-entropy is used as the loss function and validation accuracy metric. Sparse categorical cross-entropy is preferred for multiclass classification tasks and takes into account the mutual exclusiveness of classes. The model is trained with a batch size of 1024 and 120 epochs. The early stop parameter is enabled, which stops the training process if the consecutive accuracy does not improve. This helps prevent overfitting and improves efficiency when training on a large amount of data.

Here is the pseudo-code for the GetDenseNetDPProcess() function:

```
def GetDenseNetDPProcess(self):
    ReadData()
```

```
        data_features = ds_data[:, 0:78]
        data_classes = ds_data[:, 78:79]
        data_classes = np_utils.to_categorical(data_classes)
        scaler = StandardScaler()
        data_features = scaler.fit_transform(data_features)
        data_features = data_features.reshape(len(data_features), 78, 1)
        features_train, features_test, classes_train, classes_test = train_test_split(data_features, data_classes, test_size=0.2, random_state=4, shuffle=True)

        dense_block_size = 3
        layers_in_block = 4
        growth_rate = 12
        classes = classes_train.shape[1]
        model = dense_net(growth_rate * 2, growth_rate, classes, dense_block_size, layers_in_block)
        model.summary()

        optimizer = Adam(learning_rate=0.0001, beta_1=0.9, beta_2=0.999, epsilon=1e-08)
        model.compile(optimizer=optimizer, loss='categorical_crossentropy', metrics=['accuracy', 'mse'])
        history = model.fit(features_train[0:5000, :], classes_train[0:5000, :], epochs=self.epoch, validation_data=(features_test[0:100, :], classes_test[0:100, :]))

        GetPlot('DenseNet Model Accuracy', history.history['accuracy'], history.history['val_accuracy'], 'accuracy')
    GetPlot('DenseNet Model Loss', history.history['loss'], history.history['val_loss'], 'loss')
```

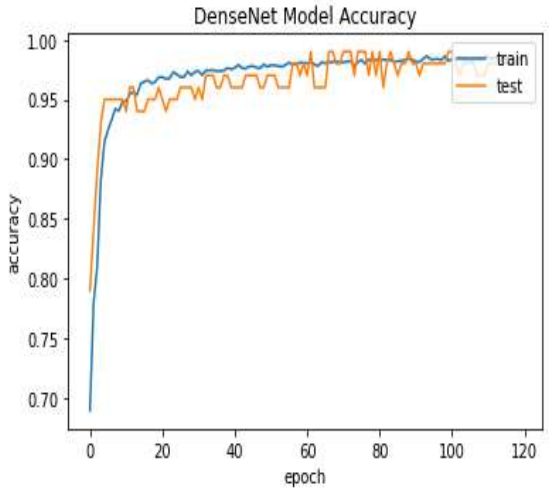


Figure 4. Accuracy graph of dense.

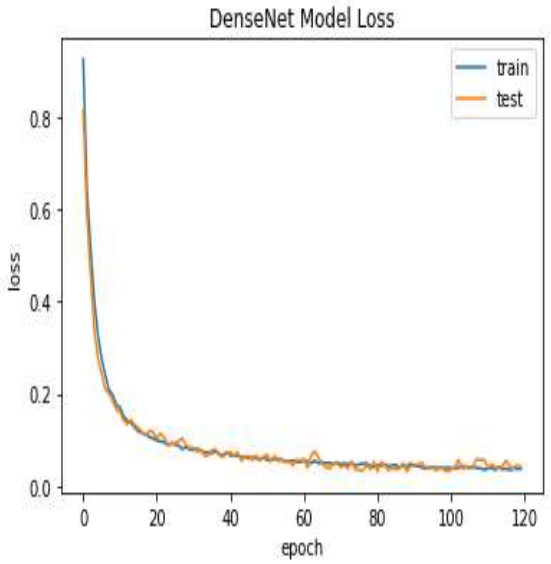


Figure 5. Loss gragh of dense.

4.1.3 Implementation of CNN-LSTM

To achieve the best performance for intrusion detection, a hybrid algorithm combining CNN and LSTM was developed. The architecture started with a single-dimensional convolution layer with (128 x 3) and 78 inputs, followed by a max-pooling layer of size 2. The same single convolution layers were added twice, followed by another max-pooling layer of size 2 before feeding the parameters to the LSTM model. The LSTM layer started with 256 nodes, followed by a dropout layer of 0.1 to reduce the unused node size. The classification layer had 15 outputs, and the softmax function was used. The convolution layers used the ReLU activation function, and sparse categorical cross-entropy was used for loss and accuracy functions. The Adam optimizer was used for adaptive learning rate based on weight updates. The batch size for training the CNN-LSTM hybrid algorithm was reduced to 512, and the number of epochs was kept at 120. The goal of this hybrid algorithm was to combine the strengths of both CNN and LSTM architectures to improve the performance of intrusion detection. By using CNN for feature extraction and LSTM for sequence learning, the hybrid algorithm can potentially achieve better accuracy and efficiency compared to using either architecture alone.

Here is the pseudo-code for the GetCNN_LSTMDPProcess() function:

```
    def GetCNN_LSTMDPProcess(self):
        ReadData()
        data_features = ds_data[:, 0:78]
        data_classes = ds_data[:, 78:79]
        data_classes = np_utils.to_categorical(data_classes)
        scaler = StandardScaler()
```

```
        data_features =
scaler.fit_transform(data_features)
        data_features =
data_features.reshape(len(data_features), 78, 1)
        features_train, features_test, classes_train,
classes_test = train_test_split(data_features,
data_classes, test_size=0.2, random_state=4,
shuffle=True)
        model = Sequential()
        model.add(Conv1D(128, 3,
activation='relu', input_shape=(78, 1)))
        model.add(Conv1D(128, 3,
activation='relu'))
        model.add(MaxPooling1D(2))
        model.add(Conv1D(128, 3,
activation='relu'))
        model.add(MaxPooling1D(2))
        model.add(LSTM(256))
        model.add(Dropout(0.1))
        model.add(Dense(data_classes.shape[1],
activation='softmax'))
        model.summary()

model.compile(loss='categorical_crossentropy',
optimizer='adam', metrics=['accuracy', 'mse'])

        history = model.fit(features_train[0:5000, :],
classes_train[0:5000, :], epochs=self.epoch,
validation_data=(features_test[0:100, :],
classes_test[0:100, :]))

        GetPlot('CNN-LSTM Model Accuracy',
history.history['accuracy'],
history.history['val_accuracy'], 'accuracy')
        GetPlot('CNN-LSTM Model Loss',
history.history['loss'], history.history['val_loss'],
'loss')
```

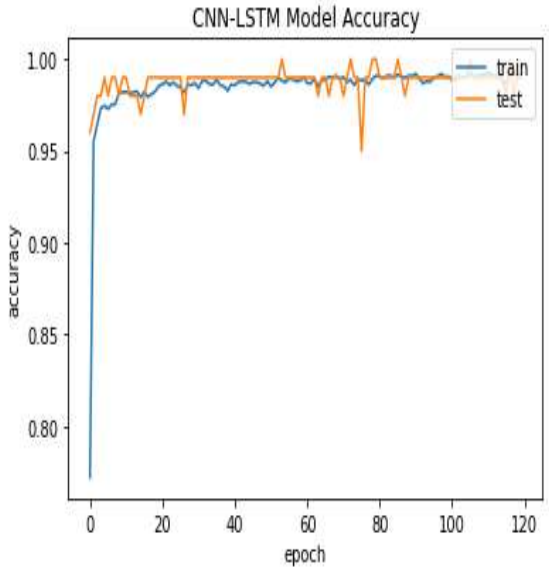


Figure 6. Accuracy graph of CNN+LSTM.

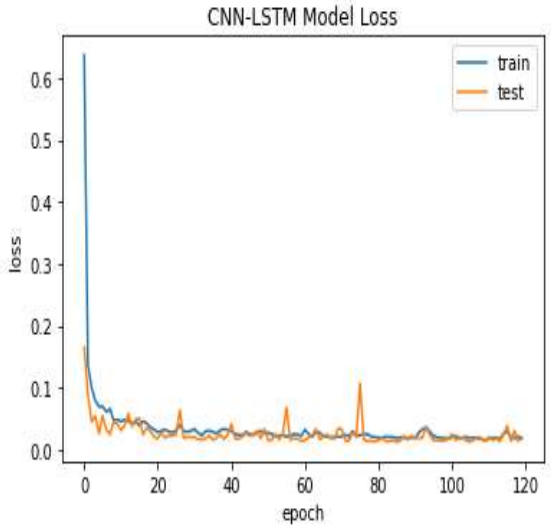


Figure 7. Loss gragh CNN-LSTM.

4.1.4 Main Dp process network

By using this proposed model, we were able to present a new model that has better accuracy than the other three models. In this proposed model (dp-model), we used a 1D CNN layer in the first layer of this model, and then we used a Flatten layer, then a Dense layer and a Dropout layer to reduce the dimensions. and after that, a dense layer has been used again. Here is a pseudo-code representation of the provided code snippet:

```
    function GetMainDPProcess():
        ReadData()
        data_features = ds_data[:, 0:78]
        data_classes = ds_data[:, 78:79]
        data_classes =
np_utils.to_categorical(data_classes)
        scaler = StandardScaler()
        data_features =
scaler.fit_transform(data_features)
        data_features =
data_features.reshape(len(data_features), 78, 1)
        features_train, features_test, classes_train,
classes_test = train_test_split(data_features,
data_classes, test_size=0.2, random_state=4,
shuffle=True)

        model = Sequential()
        model.add(Input(shape=(78, 1)))
        model.add(Conv1D(64, 2,
activation='relu'))
        model.add(Flatten())
        model.add(Dense(32, activation='relu'))
        model.add(Dropout(0.1))
        model.add(Dense(data_classes.shape[1],
activation='softmax'))
        model.summary()
```

```
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy', 'mse'])
        history = model.fit(features_train, classes_train, epochs=self.epoch, validation_data=(features_test, classes_test))

        GetPlot('Proposed Model Accuracy', history.history['accuracy'], history.history['val_accuracy'], 'accuracy')
        GetPlot('Proposed Model Loss', history.history['loss'], history.history['val_loss'], 'loss')
```

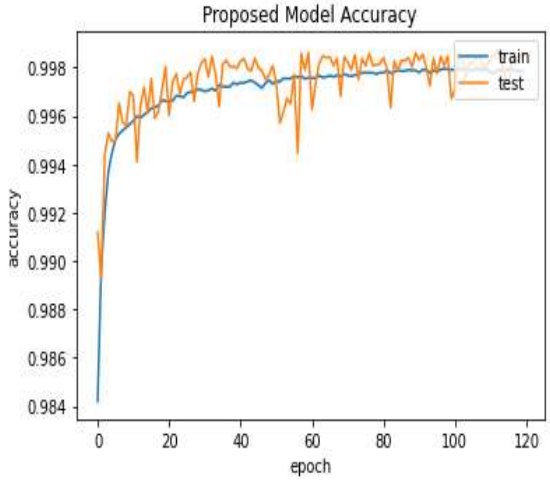


Figure 8. Accuracy graph of dp-model.

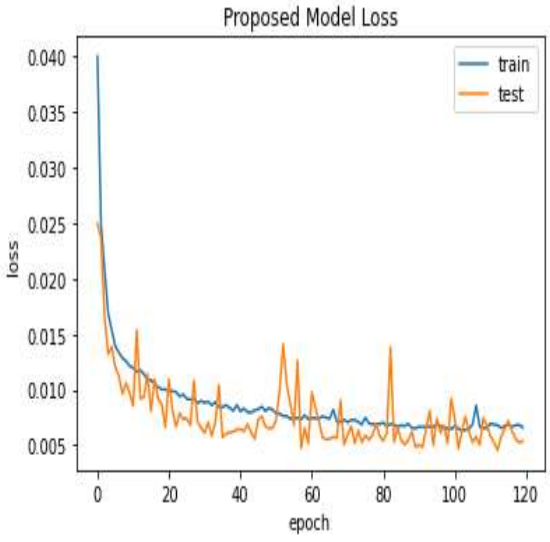


Figure 9. Loss gragh of dp-model.

This study aimed to compare the effectiveness of different neural network architectures using the CIC-IDS2017 dataset in the context of deploying machine learning models in IoT infrastructure. The research focused on identifying suitable architectures that could achieve high performance while minimizing computational requirements, rather than increasing the depth of the neural networks.

The proposed model demonstrated the highest accuracy of 0.997 on the test data. It was observed that the absence of dropout and convolutional layers, which are commonly used in CNN and CNN-LSTM architectures, allowed the data to remain intact and contribute efficiently to the classification layer. This finding is consistent with the belief that the proposed model benefits from providing all combinations of a feature vector to the classification layer, considering the mutual exclusiveness among the 15 categorical inputs. It highlights the advantage of the loss function that takes this into account [14].

The analysis of Figure 3 revealed that the loss per epoch stabilizes after the initial iterations, indicating convergence of the training process. Similarly, Figure 2 displayed consistent accuracy over epochs after a certain number of iterations, suggesting the model's ability to maintain stable performance. These trends indicate that the proposed model(dp-model), along with CNN and CNN-LSTM architectures, performed exceptionally well, achieving accuracies of 0.997, 0.988, and 0.99, respectively.

Precision and recall scores further indicated that all three algorithms are effective in operating in an IoT infrastructure after deployment. The retraining process also exhibited consistency, with CNN demonstrating excellent optimization of loss and maintaining stability throughout the 120 epochs. On the other hand, DenseNet and CNN-LSTM showed slight fluctuations during mid-training, which were considered insignificant [30]. Figures 4 and 5 provided visual representations of the model loss and accuracy loss for each deep learning algorithm.

In summary, the results of this article large the effectiveness of the proposed neural network architecture, along with CNN and CNN-LSTM, in achieving high accuracy in classifying data from the CIC-IDS2017 dataset. The findings support the suitability of these architectures for deployment in IoT infrastructure. The stability and consistency observed in the training process further reinforce their reliability. However, further research and optimization may be required to address the minor fluctuations observed in the DenseNet and CNN-LSTM architectures during mid-training.

4.2 Analysis of effect of noise on algorithms of proposed model

To evaluate the effectiveness of the proposed models, we introduced Gaussian noise to the dataset. Through these tests, it became evident that the proposed method outperformed other approaches. Even when subjected to Gaussian

noise distribution, the models displayed minimal impact on accuracy, as depicted in graphs 10, 11, 12, 13, 14, 15,16 and 17.

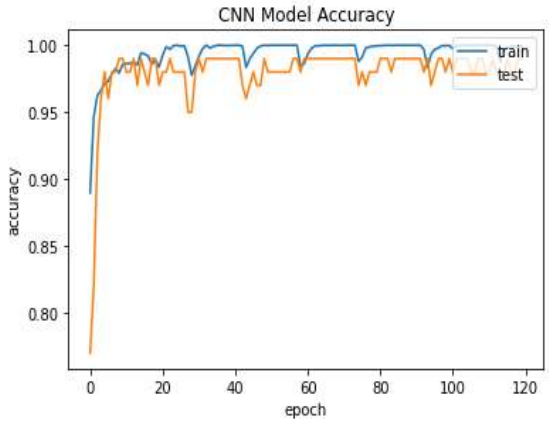


Figure 10. Accuracy graph of CNN with Gaussian noise distribution.

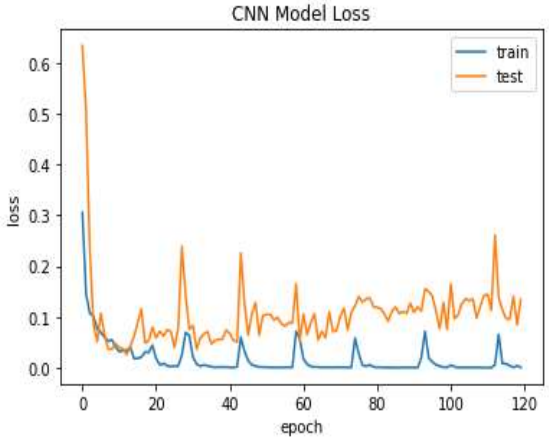


Figure 11. Loss gragh of CNN with Gaussian noise distribution.

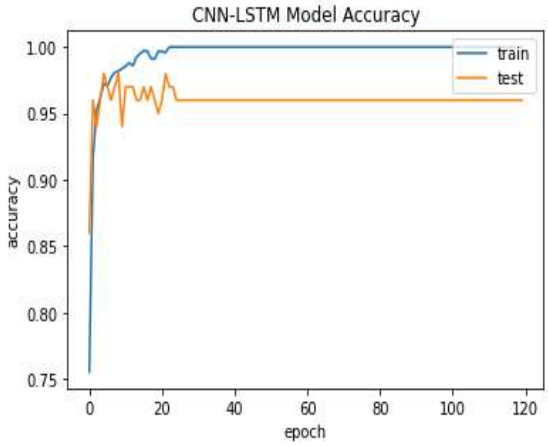


Figure 12. Accuracy graph of CNN+LSTM with Gaussian noise distribution.

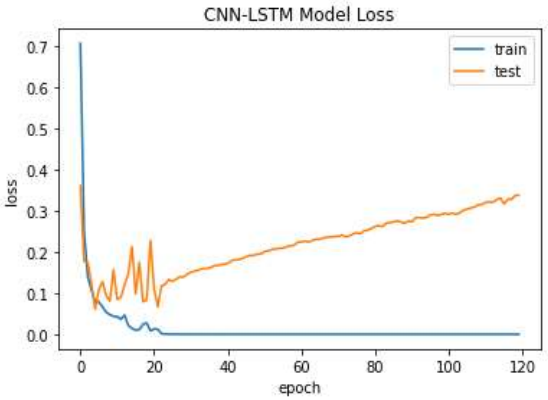


Figure 13. Loss gragh CNN-LSTM with Gaussian noise distribution.

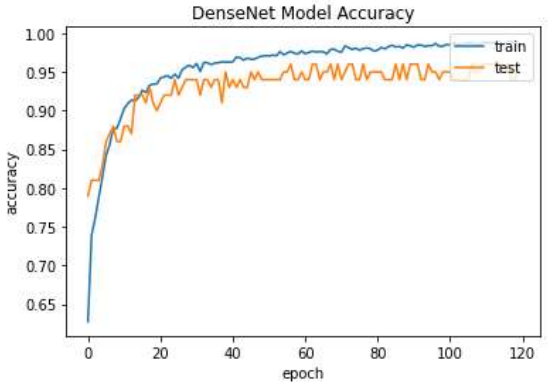


Figure 14. Accuracy graph of dense with Gaussian noise distribution.

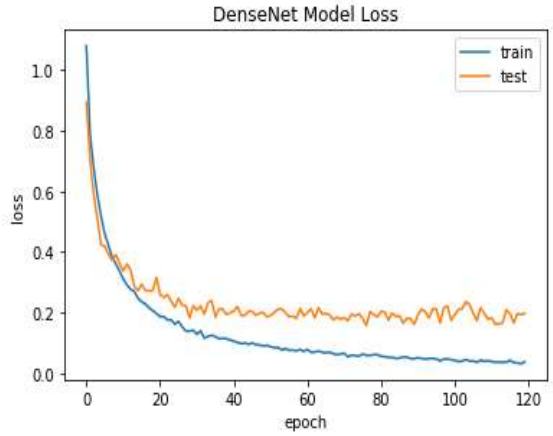


Figure 15. Loss gragh of dense with Gaussian noise distribution.

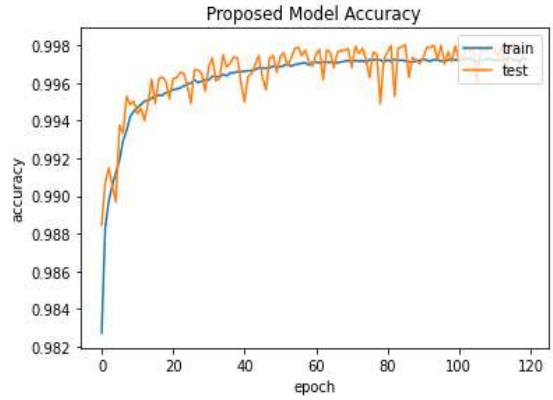


Figure 16. Accuracy graph of dp-model with Gaussian noise distribution.

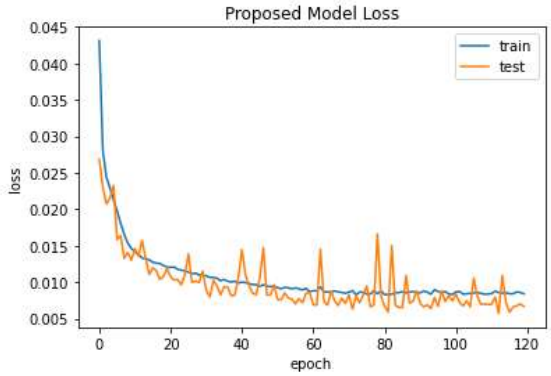


Figure 17. Loss gragh of dp-model with Gaussian noise distribution.

4.3 Performance evaluation metrics

Prior research in network security has typically presented the results of intrusion detection techniques using binary classification performance metrics [31], [32], [33]–[34]. In the context of Intrusion Detection Systems (IDSs) for Wireless Sensor Networks (WSNs) and the Internet of Things (IoT), these results commonly include a subset of binary classification metrics such as detection rate, true and false positive alarms, and accuracy rate. However, we propose the use of an extended set of performance metrics to gain a deeper understanding of the suitability and effectiveness of detection models. The binary classification performance measurements revolve around categorizing the system's alarm types. The raised alarms are quantified and classified as either True or False alarms. True Positive (TP) and True Negative (TN) values represent correct predictions made by the model. TP alarms indicate instances where the model correctly identifies malicious activities, while TN values indicate correct identification of benign activities with no alarm raised. On the other hand, False Positive (FP) and False Negative (FN) values denote misclassifications made by the model. FP values refer to undetected malicious local node activities, while FN values indicate benign node activities that triggered an alarm.

Table 3. Equations used.

Equations	number
$\text{Recall}/TPR = \frac{TP}{TP+FP}$	(1)
$\text{Precision}/PPV = \frac{TP}{TP+FP}$	(2)
$ACC = \frac{TP+TN}{TP+TN+FP+FN}$	(3)
$F1score = 2*\frac{\text{Precision}*\text{Recall}}{\text{Precision}+\text{Recall}}$	(4)

Recall, also known as True Positive Rate (TPR) or sensitivity, is a metric that denotes the ratio of True Positives (malicious node alarms) to the total number of alarms raised (equation 1). This metric is often accompanied by Precision or Positive Predictive Value (PPV), which represents the percentage of correctly identified malicious nodes within the network (equation 2) [34]. In some works, the term Detection Rate has been used interchangeably with Precision/PPV [31], [33]. Accuracy (ACC) measures the ratio of correctly classified input local node activities (benign or malicious) to the total number of local node activities used in the specific experimental analysis. The equation for ACC is shown in Equation 3, and it has been utilized in previous studies [32], [34]. The F1 score, commonly preferred by researchers, provides an overall measure of the quality of a classifier's predictions as it captures both precision and recall. It is calculated as the harmonic mean of precision and recall scores (equation 4). The F1 score ranges from 0% to 100%, where a higher value indicates a better quality classifier. In summary, these performance metrics offer a comprehensive evaluation of intrusion detection models, enabling a more nuanced assessment of their effectiveness and suitability for WSNs and IoT environments.

4.4 Discuss Comparison

In this section, we will examine and compare our method with others' methods. We presented an intrusion detection model in Internet of Things networks based on deep neural network using CICIDS-2017 dataset. The purpose of this research work is to provide a more efficient method than previous methods that has the ability to detect all types of attacks in the Internet of Things network.

Table 4. Performance Comparison of result CNN model.

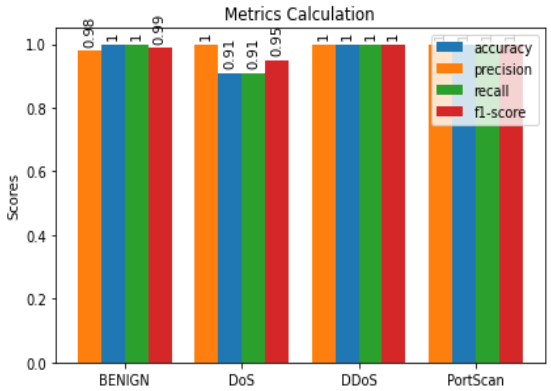


Table 5. Performance Comparison of result CNN-LSTM model.

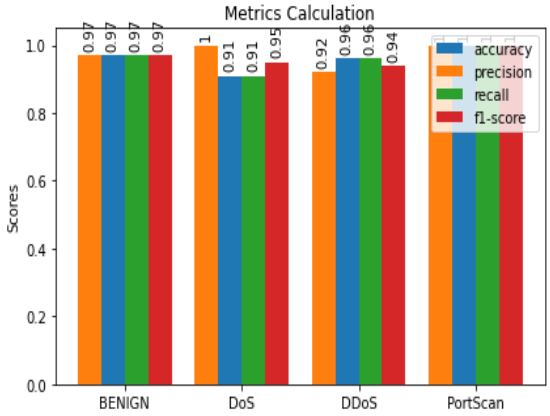


Table 6. Performance Comparison of result dense model.

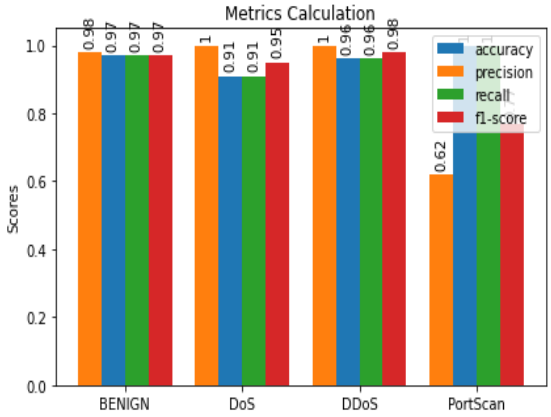
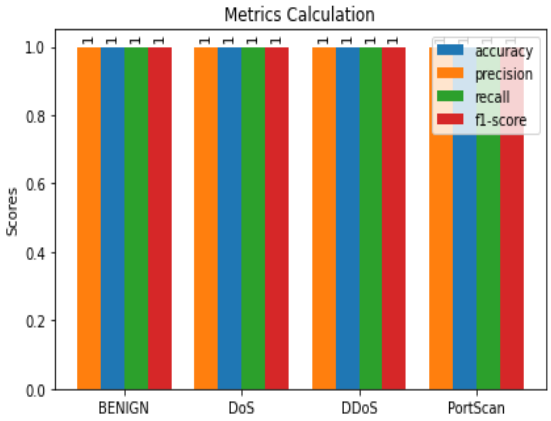


Table 7. Performance Comparison of result DP- model



4.4 Discuss Comparison

In this section, we will examine and compare our method with others' methods. We presented an intrusion detection model in Internet of Things networks based on deep neural network using CICIDS-2017 dataset. The purpose of this research work is to provide a more efficient method than previous methods that has the ability to detect all types of attacks in the Internet of Things network. In this research work, two methods of dimension reduction and deep learning are used to implement the proposed method. In addition, we have examined effective deep learning models to demonstrate cyber security knowledge in Internet of Things networks, including CNN, DenseNet, and a hybrid model of CNN and LSTM, and a proposed model called dp-model, which has brought the accuracy of the proposed model to 0.997. This study aimed to compare the effectiveness of different neural network architectures using the CIC-IDS2017 dataset in the context of deploying machine learning models in IoT infrastructure. The research focused on identifying suitable architectures that could achieve high performance while minimizing computational requirements, rather than increasing the depth of the neural networks.

Also, we were able to improve the accuracy of LSTM & CNN, DenseNet, CNN model in the article of Mr. Mostofa Ahsan et al.[30] to 0.988, 0.985, 0.99 respectively. Also, our analysis showed that our proposed models can detect several other cyber-attacks targeting IoT devicesAs future work, we want to improve the present models'model prediction performance and test them against additionalrouting attack types. We're looking at adding more features to create a single model that can detect many forms of cyber-attacks against IoTs.

Table8.Comparison analysis of the similar works using various methods.

Article/Year	Model	Accuracy	Dataset
-/2020	-	-	CICIDS-2017
-/2021	-	-	CICIDS-2017
30/2022	Dense Net CNN CNN-LSTM	97.77 96.87 96.92	CICIDS-2017
Our proposed method/2023	Dense Net CNN CNN-LSTM DP-Model	98.5 98.8 99.0 99.7	CICIDS-2017

5.Conclusion

IoT technology and systems offer promising opportunities for integrating various technological advancements, services, and management capabilities from the digital world. As technology continues to advance, individuals are becoming increasingly connected to these IoT systems, emphasizing the need for robust security measures to protect the vast number of users and their extensive data.

In this research study, we have proposed a solution aimed at detecting intrusions in IoT networks by evaluating and comparing different deep neural network architectures. Through our investigation, we have determined that DP-model(proposed model) outperforms other deep learning models and machine learning algorithms, achieving an accuracy rate of 0.997. Additionally, our analysis has revealed that our proposed models exhibit the capability to detect various cyber-attacks targeting IoT devices.

Abbreviations

IOT	Internet of Things
IDS	Intrusion Detection Systems
CNN	convolutional neural network
LSTM	Long short-term memory
NNs	Neural Networks
DL	Deep Learning
ACC	Accuracy
GAN	generative adversarial network
IOV	Internet of Vehicles
EO-ANN	Equilibrium Optimization-based Artificial Neural Network
DDoS	distributed denial-of-service
Dos	denial-of-service
DNN	deep neural network
DT	Decision tree
OICS-VFSL	Optimized intra/inter-class-structure-based variational few-shot learning
EML	Elite Machine Learning
SAE	stacked autoencoders
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, pp. 73-85.

[28] L. khalvati; M. Keshtgary; N. Rikhtegar, Intrusion Detection based on a Novel Hybrid Learning Approach, *Journal of AI and Data Mining*, vol. 6, no. 1 , March 2018, , pp. 157-162, doi.org/10.22044/jadm.2017.979 .

[29] M. Rahimi, A. A. Taheri, and H. Mashayekhi, "Learning a Nonlinear Combination of Generalized Heterogeneous Classifiers," *Journal of Artificial Intelligence & Data Mining*, vol. 11, no. 1, pp. 77–93, 2023.

[30] M. Ahsan, N. Rifat, M. Chowdhury, and R. Gomes, "Intrusion Detection for IoT Network Security with Deep Neural Network," in 2022 IEEE *International Conference on Electro Information Technology (eIT)*, 2022, doi: 10.1109/eit53891.2022.9814006.

[31] S. Raza, L. Wallgren, and T. Voigt, "SVELTE: Real-time Intrusion Detection in the Internet of Things," *Ad hoc networks*, vol. 11, no. 8, pp. 2661–2674, 2013.

[32] C. Ioannou, V. Vassiliou, and C. Sergiou, "An Intrusion Detection System for Wireless Sensor Networks," in 24th International Conference on Telecommunications, ICT 2017, Limassol, *Cyprus*, May 3-5, 2017, pp. 1-5.

[33] C. Cervantes, D. Poplade, M. Nogueira, and A. Santos, "Detection of sinkhole attacks for supporting secure routing on 6LoWPAN for Internet of Things," in Proceedings of the 2015 *IFIP/IEEE International Symposium on Integrated Network Management*, IM 2015, June 2015, doi: 10.1109/INM.2015.7140344.

[34] E. Kabir, J. Hu, H. Wang, and G. Zhou, "A novel statistical technique for intrusion detection systems," Future Generation Computer Systems: *the international journal of grid computing*: *theory, methods and applications*, vol. 79, pp. 303-318, 2018, doi:10.1016/j.future.2017.01.029.

تشخیص نفوذ برای امنیت شبکه اینترنت اشیا با استفاده از یادگیری عمیق

رؤیا مرشدی [۱]، سید مجتبی متین خواه [۱،*] و محمدتقی صادقی [۲]

[۱] گروه مهندسی کامپیوتر، دانشگاه یزد، یزد، ایران.

[۲] گروه مهندسی برق، دانشگاه یزد، یزد، ایران.

ارسال ۲۰۲۳/۰۹/۰۶؛ بازنگری ۲۰۲۳/۱۰/۱۲؛ پذیرش ۲۰۲۳/۱۱/۱۰

چکیده:

سیستم‌های تشخیص نفوذ (IDS) اجزای حیاتی برای محافظت از شبکه‌های اینترنت اشیا در برابر تهدیدات سایبری هستند .این مطالعه یک رویکرد پیشرفته برای تشخیص نفوذ شبکه اینترنت اشیا، استفاده از تکنیک‌های یادگیری عمیق و داده‌های بکر ارائه می‌کند .ما از مجموعه داده CICIDS2017 در دسترس عموم استفاده می‌کنیم که آموزش و آزمایش جامع مدل‌های تشخیص نفوذ را در سناریوهای مختلف حمله، مانند حملات انکار سرویس توزیع‌شده(DDoS) ، اسکن‌های پورت، فعالیت بات‌نت و موارد دیگر امکان‌پذیر می‌سازد. هدف ما ارائه روشی موثرتر از روش های قبلی است. مدل یادگیری عمیق پیشنهادی ما از لایه‌های انتقال متراکم و معماری LSTM استفاده می‌کند که برای ثبت وابستگی‌های مکانی و زمانی درون داده‌ها طراحی شده است.ما از معیارهای ارزیابی دقیق، از جمله از دست دادن متقابل آنتروپی طبقه‌بندی شده و دقت، برای ارزیابی عملکرد مدل استفاده کردیم .نتایج رویکرد ما دقت فوق‌العاده‌ای را نشان می‌دهد که در داده‌های آزمون به اوج ۰,۹۹۷ رسیده است .مدل ما ثبات در معیارهای تلفات و دقت را نشان می‌دهد و قابلیت‌های تشخیص نفوذ قابل اعتماد را تضمین می‌کند .تجزیه و تحلیل مقایسه ای با سایر مدل های یادگیری ماشینی اثربخشی رویکرد ما را تأیید می کند .علاوه بر این، مطالعه ما انعطاف‌پذیری مدل را در برابر نویز گاوسی ارزیابی می‌کند و ظرفیت آن را برای حفظ دقت در شرایط چالش‌برانگیز آشکار می‌کند .ما معیارهای دقیق عملکرد را برای انواع مختلف حملات ارائه می‌کنیم و بینش‌هایی در مورد اثربخشی مدل در سناریوهای مختلف تهدید ارائه می‌کنیم.

کلمات کلیدی: شبکه های عصبی، اینترنت اشیا، حملات DDOS، مجموعه داده CICIDS2017 ، شبکه های عصبی کانولوشنی، شبکه های عصبی بازجریانی.