INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT (IJSREM)
VOLUME: 08 ISSUE: 03 | MARCH - 2024 SJIF RATING: 8.448 ISSN: 2582-3930

Enhancing Intrusion Detection through Unsupervised Deep Learning Models

S.P.Renjith[1,*] ,Dr.Vargheese [2]
[1] ME Student, Department of CSE, PSN College of Engineering & Technology
[2] Professor, Department of CSE, PSN College of Engineering & Technology

***Abstract*—** The Internet of Things (IoT) has experienced significant growth since its inception, representing a groundbreaking technological advancement. In essence, IoT involves the seamless integration of devices and data to automate and centralize various processes. This transformative technology is revolutionizing business operations and reshaping society as a whole. As IoT continues to evolve, the importance of detecting vulnerabilities and weaknesses becomes paramount in order to thwart unauthorized access to critical resources and business functions, which could potentially render the entire system unavailable. One prevalent threat in this context is Denial of Service (DoS) and Distributed DoS attacks. In this project, propose an innovative architecture known as Protocol Based Deep Intrusion Detection (PB-DID). To create our dataset, we gathered packets from IoT traffic and compared features from two well-known datasets: UNSWNB15 and Bot-IoT. We focused on flow and Transmission Control Protocol (TCP) characteristics. Our primary objective was to accurately classify network traffic into three categories: non-anomalous, DoS, and DDoS, while addressing issues like data imbalance and overfitting. Utilizing deep learning (DL) techniques, we achieved an impressive classification accuracy of 96.3%. This level of accuracy demonstrates the potential of our PB-DID architecture in effectively identifying and mitigating intrusion attempts in the context of IoT, thereby enhancing the security and reliability of IoT systems.

Keywords—: *Intrusion detection in IoT, Deep learning for intrusion detection, DoS detection, DDoS detection.*

I. INTRODUCTION

In the age of the IoT, our world is witnessing unprecedented levels of convenience and efficiency. However, the self-configuring and open nature of the IoT renders it vulnerable to a wide range of attacks. IoT devices often need more manual controls and have limited memory and computational power resources. Despite these limitations, the IoT's high dependence and rapid growth have led to increased security risks, making network security solutions crucial. While detecting some attacks can be challenging, some systems currently do an excellent job [1,2]. The volume of information transmitted across networks is growing quickly, leading to an increase in the number of attacks on networks. This has made it crucial to develop quick and effective ways to detect attacks and reduce the risks associated with the widespread adoption of IoT technology. Denial of Service (DoS) is one of the most damaging attacks, as it prevents legitimate users from accessing services. DoS attacks can have severe consequences for critical applications such as healthcare, leading to fatal delays in medical services. In 2016, the Mirai botnet was launched, which hacked into CCTV cameras using default user IDs and credentials to initiate DDoS attacks on DNS servers, bringing internet access in some parts of the US to a standstill. Another botnet, Mozi, capable of launching various DDoS attacks, was identified in April 2020. The architecture of IoT networks is shown in Figure 1 [3]. There is a need for more innovative approaches to strengthen network security and deliver embedded intelligence in an environment involving the IoT. Intrusion Detection Systems (IDS) monitor the host or network for security breaches and notify the administrator when they are detected. Entry events are entered through sensors into a database and employ a set of criteria to generate alerts for security incidents that have occurred. However, stealth systems are still in the early stages of research, and several issues need to be addressed to achieve such high accuracy and low false alarm rates. Signature, anomaly, and specification are the three types of IDS classified based on their detection methods. Signature-IDS compares network traffic patterns to the previously stored patterns of attack.

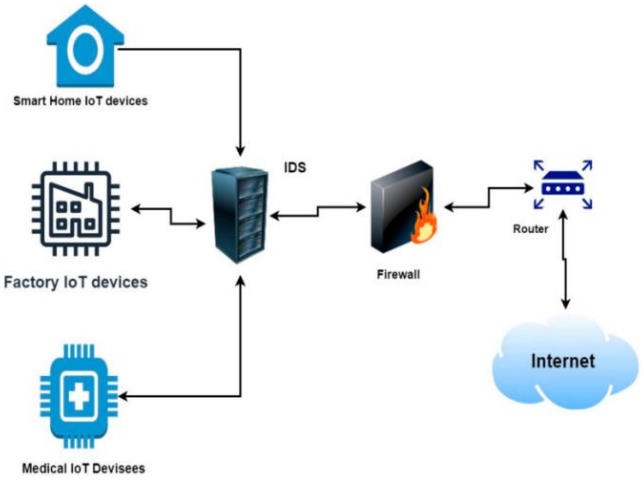

Figure 1. Network Architecture for IoT.

An alert is triggered if a match is discovered. Signature-IDS has high accuracy and a minimum number of false alarms, but it cannot identify new assaults. A specification based IDS network compares traffic behaviour to the present rule set and values to detect malicious activity. A security expert determines these standards manually [4–6]. IoT devices produce a large volume of information, and conventional data gathering, processing and storage techniques might be unable to handle it. The data heterogeneity the IoT creates is causing issues for current data processing systems. In order to be able to predict, and evaluate the enormous amounts of data, new mechanisms must be developed to handle this overwhelming extent of information. Thus, ML is one of the best computational models for providing intelligence to IoT devices. ML could support systems and intelligent devices in understanding data that machines or humans generate. ML has the capability of automating behaviour for smart devices, based on the knowledge that this is the essential role of IoT solutions [7].

In recent years, machine learning, with its rapid development, has found extensive applications in the field of intrusion detection. Machine learning algorithms offer unique advantages compared to traditional detection methods. They can not only learn complex patterns and rules from large volumes of data but also handle high-dimensional and nonlinear data, making them more suitable for intrusion detection in complex systems. Furthermore, with the advancement of networks, a significant amount of network data, including samples of various intrusion and abnormal behaviors, has been accumulated. This rich dataset provides ample training samples for machine learning, ensuring excellent detection performance of machine learning algorithms. However, despite the achievements of machine learning algorithms, there are still some challenges and issues that need to be addressed.

The real-time capability of an IoT device intrusion detection system is a crucial consideration for ensuring network security. IoT devices typically have limited computational resources and storage capacity. Due to these resource constraints, the system may struggle to efficiently process and analyze a large amount of network traffic data, leading to delays or inability to meet real-time requirements in detection. Additionally, in the face of frequent network attacks on IoT devices and the constant evolution of attack techniques, intrusion detection systems are under increasing pressure to detect such attacks [8]. The generation of a vast amount of network traffic data introduces significant redundancy and irrelevant features. Redundant features in the data can lead to overfitting during the model learning process, ultimately diminishing detection performance [9]. One effective approach to address feature redundancy is feature selection. Feature selection plays a crucial role in machine learning-based intrusion detection systems, reducing the dimensionality of the dataset, lowering training time and computational costs, while improving model performance [10].

In this proposed work, the PB-DID architecture utilizes comprehensive data from both the Bot-IoT and UNSW-NB15 datasets. These datasets encompass major attack scenarios and are based on realistic traffic scenarios. The architecture involves training an LSTM-based unsupervised deep model using 26 features. For classification purposes, flow and TCP features are selected, as they are particularly relevant to IoT devices. One-hot encoding is applied to one feature at a time, and all features are simultaneously used in the embedding layer. Additionally, the issue of data imbalance was addressed during the merging of datasets.

This project aims to achieve the following objectives:

- Comparison of the UNSW-NB15 and the Bot-IoT datasets to identify common features between them.
- Union of both data-sets using common features which fall in flow and TCP categories.
- Removal of issues pertaining to the imbalance nature of data-sets and biased classification.
- Employing LSTM based un-supervised deep learning model for the detection of DoS and DDoS attacks.

II. LITERATURE SURVEY

Recently, machine learning techniques have been widely applied in the field of intrusion detection in the Internet of Things (IoT) and have achieved excellent results. Intrusion detection systems based on machine learning are typically divided into two parts. The first part is data preprocessing, which involves preprocessing the data before feeding it into the model. This includes feature selection and handling imbalanced datasets, to provide better inputs to the model. The second part is the classifier, where selecting an appropriate model can maximize the intrusion detection rate. Therefore, many researchers have focused their efforts on these two aspects to create powerful intrusion detection systems. In this section, we will review the recent work.

Lazzarini et al. [11] built an IoT intrusion detection system using an ensemble stacking approach. They combined four different deep learning models (MLP, DNN, CNN, and LSTM) to detect and classify attacks in IoT environments. Binary and multi-class experiments were conducted on the Ton-IoT and CIC-IDS2017 datasets. The results showed that the proposed method was able to detect the majority of attacks with particularly low false positive (FP) and false negative (FN) rates. However, this approach integrates four different models, which requires a significant amount of resources and further evaluation of its performance on real IoT devices. Alani [12] used feature importance-based recursive feature elimination (RFE) for feature selection on the dataset, selecting the top 11 most important features. They used a

decision tree (DT) classifier for classification and Shapley additive explanation (SHAP) to explain the selected features and classifier. The proposed method achieved an accuracy of 0.9997 on the WUSTL-IIOT-2021 dataset.

Nizamudeen [13] employed integer-grading normalization (I-GN) for data preprocessing and used opposition-based learning (OBL)-rat inspired optimizer (RIO) for feature selection to retain important features. Experiments on a combined dataset (NF-UQ-NIDS) showed improved detection accuracy compared to other state-of-the-art methods. Sharma et al. [14] proposed an IoT intrusion detection system based on a deep neural network (DNN) model to better protect the security of internet devices. They used a generative adversarial network (GAN) to synthesize minority attack class data and employed the Pearson's correlation coefficient (PCC) filter method for feature selection. Experimental results on the UNSW-NB15 dataset achieved an accuracy of 91% with balanced data.

Kareem et al. [15] proposed a feature selection algorithm using the algorithm for bird swarms (BSA) to improve the performance of the gorilla troops optimizer (GTO). Experiments on the NSL-KDD, CICIDS-2017, UNSW-NB15, and Bot-IoT datasets demonstrated that the proposed GTO-BSA achieved better convergence speed and performance. Mohy-eddine et al. [16] presented an IoT intrusion detection system based on the K-nearest neighbors (K-NN) algorithm, utilizing principal component analysis (PCA), univariate statistical tests, and genetic algorithm (GA) for feature selection. Experiments on the BotIoT dataset achieved a high accuracy of 99.99% while significantly reducing the prediction time. Liu et al. [17] addressed the issue of excessive flow features affecting detection speed in IoT intrusion detection systems by proposing a feature selection method based on a genetic algorithm. Extensive experiments on the Bot-IoT dataset selected six features from 40 features, achieving an accuracy of 99.98% and an F1 score of 99.63%.

Alweshah et al. [18] proposed a novel wrapping feature selection algorithm that employed the emperor penguin colony (EPC) to explore the search space, selecting K-nearest neighbors (KNN) as the classifier. Experimental results on well-known IoT datasets showed improved accuracy and reduced feature size compared to methods such as the multi-objective particle swarm optimization (MOPSO). Hassan et al. [19] used an improved binary manta ray foraging algorithm for feature selection to remove redundant and irrelevant features from the dataset, and utilized a random forest (RF) classifier for classification. The proposed method was evaluated on the NSL-KDD and CIC-IDS2017 datasets, selecting 22 and 38 features, respectively, and achieved accuracies of 98.8% and 99.3%. Mohiuddin et al. [20] proposed a modified wrapper-based whale sine-cosine method to reduce the complexity of feature selection optimization, selecting important features, and used XGBoost as the classifier. Experimental results on the UNSW-NB15 dataset achieved accuracy rates of 99% and 91% for binary and multi-class classification, respectively, and an accuracy of 98% for binary classification on the CIC-IDS2017 dataset.

Shareena et al. present a deep-learning-based intrusion detection system for IoT DDoS botnet attacks using a dataset created in a realistic network environment. A highly extensible Deep Neural Network (DNN) is developed and evaluated for headstrong detection of IoT botnet attacks. The results show that the proposed DNN outperforms existing systems with high accuracy and precision, demonstrating its potential for effectively detecting IoT DDoS botnet attacks [21]. Alghanam et al. present an improved PIO feature selection model for intrusion detection. The algorithm uses ensemble learning for detection. The dataset and architecture used in these modes is not complex [22].

Syed et al. discuss an RNN-based model for an intrusion detection system in IoT networks. In comparison to models trained on the full feature set, the models generated on the smaller dataset had higher recall rates without losing the capacity to distinguish between classes [23]. In the past, few researchers applied deep learning algorithms for intrusion detection systems. Many intrusion detection systems currently in use do not take into account the issue of dataset imbalance or model maintenance [24, 25]. This can result in high levels of bias, as well as high false-positive and false-negative rates, which in turn can lead to security breaches. It is important to address these issues in order to improve the accuracy and effectiveness of intrusion detection systems, and ultimately reduce the risk of security breaches.

III. SYSTEM OVERVIEW

The process involves features comparison to find similar features in the UNSW-NB15 and the Bot-IoT datasets, features selection, data pre-processing and selection and model training using un-supervised LSTM deep learning model are shown in Figure 1.

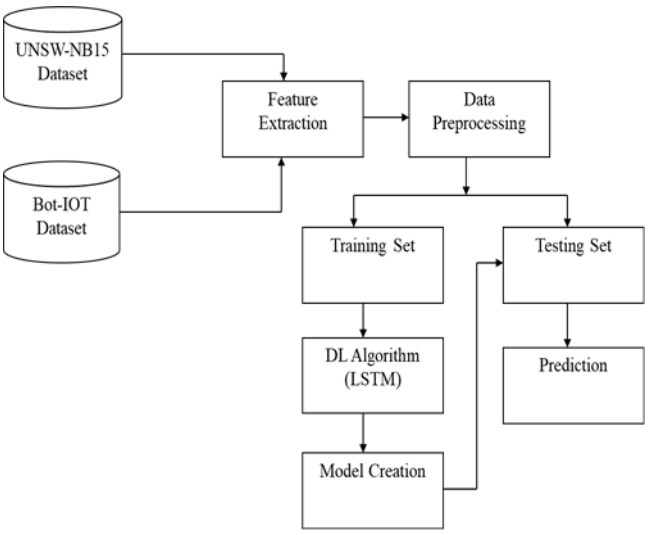

Figure 2. System Architecture

A. DATA-SETS

Two widely recognized raw network packet datasets, namely UNSW-NB15 and Bot-IoT, were utilized for training and validation. In contrast to numerous other studies that employed partial datasets, our approach involves using the complete data for model training. A brief description of both datasets is provided below.

1) UNSW-NB15

The dataset was released in 2015 and comprises a simulation of over 2.5 million network packets. It encompasses nine types of attacks, including Exploits, Reconnaissance, DoS, Generic, Shellcode, Fuzzers, Backdoors, Worms, and Analysis, alongside non-anomalous packets. Notably, the dataset exhibits high imbalance, with over 87% of packets categorized as non-anomalous. For further details about this dataset, additional information can be referenced.

2) BOT-IoT DATA-SET

This data-set is the latest in the field. It consists of more than 72 million records with the mixture of simulated and real time scenarios. It has four categories of attacks but major portion of data-set has DoS and DDoS type of packets. This data-set is imbalanced just like the UNSW-NB15 data-set. Records distribution of the Bot-IoT data-set is given. More details about the data-set can be found.

B. FEATURES COMPARISON

In the UNSW-NB15, there are 49 features including 48th as a multi-class label and 49th as a binary label. In the BotIoT data-set, there are 46 features and the last three are label features. In proposed PB-DID, the features in both data-set are compared and we found that 29 features in the Bot-IoT are similar or can be evaluated in the UNSW-NB15 data-set as well.

C. FEATURES SELECTION

In the proposed PB-DID architecture, clusters of features in both data-sets are created according to flow, Domain Name System (DNS)/File Transfer Protocol (FTP)/Hypertext Transfer Protocol (HTTP), Message Queuing Telemetry Transport (MQTT) and TCP. Figure 3 illustrate the proposed structure of PB-DID for attack classification utilizing DL. The major portion of features falls into two clusters i.e. flow and TCP. Here the clusters are created by analyzing each feature's description reported by the authors. We kept a minimum number of features while covering the application and transport layer. Major contributions from the application layer are the flow features whereas from the transport layer are of TCP protocol. Therefore, both of these clusters are chosen to create optimized scenarios by keeping maximum information of a packet. This approach significantly reduces the computational time required during the learning phase.

D. DATA PRE-PROCESSING

In this section we explain different data pre-processing steps.

1) DATA TYPE RESOLUTION

In PB-DID, certain features like 'saddr,' 'daddr,' and 'proto' are categorical and need conversion into a form executable by algorithms. 'saddr' and 'daddr' represent source and destination IP addresses, respectively, while 'proto' signifies the protocol type in the flow. Numerical values were assigned to all IP addresses, with 49 in the UNSW-NB15 dataset and 301 in the Bot-IoT dataset. During the merging of both datasets, IP addresses were replaced with 350 randomly generated unique integer numbers to anonymize them and prevent overfitting. This anonymization is crucial for certain features, such as NINConnPS rcIP and NINConnPDstIP, which rely on IP addresses for evaluation. Additionally, the 'proto' feature was converted into an integer type. This process ensures meaningful representation of features dependent on IP addresses and aids in preventing overfitting.

2) MISSING PORT NUMBERS

In the Bot-IoT full data-set, the packets using ARP protocol have missing source and destination port numbers, which is understandable. In mentioned that they have given -1 as port numbers where ARP protocol is used in the 5% extracted the Bot-IoT data-set. We used the same value in PB-DID and assigned it to port numbers in full dataset where ARP protocol is used.

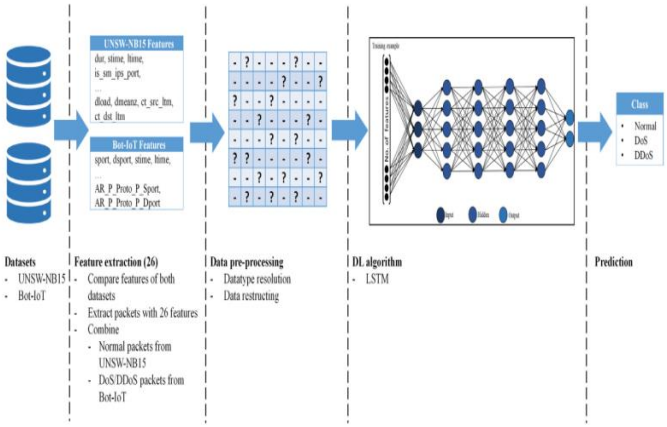

Figure 3. Proposed structure of PB-DID for attack classification using DL.

3) RESOLVING THE DATA IMBALANCE ISSUE

Imbalanced data, a common challenge in machine learning, arises when the distribution of classes is uneven. This can range from a slight imbalance to a severe one. Training a model on a severely imbalanced dataset often leads to poor predictive performance for minority classes. The UNSW-NB15 and Bot-IoT datasets exemplify imbalanced data, with 87.35% of UNSW-NB15 data being non-anomalous, and only 0.013% of Bot-IoT data being non-anomalous. Additionally, in the Bot-IoT dataset, approximately 52.5% of data corresponds to DDoS attacks, and around 45% is attributed to DoS attacks. Due to this imbalance, neither dataset alone is sufficient for training and predicting non-anomalous, DDoS, or DoS packets. Therefore, merging both datasets becomes essential to ensure meaningful predictions.

The data merging process from UNSW-NB15 and BoT-IoT datasets is outlined. UNSW-NB15 contains approximately 2.218 million non-anomalous packets, while BoT-IoT includes 38.5 million DDOS packets and 33 million DoS packets. In PB-DID, a complete data unit consists of 2.218 million non-anomalous packets. We create 14 equal data chunks, each containing 2.218 million non-anomalous, DDoS, and DoS packets. Consequently, each chunk comprises a total of 6.654 million packets, employing a strategy where non-anomalous samples are repeated, and DDoS and DoS samples are unique within each chunk. PB-DID is trained and validated separately for each chunk, and an average prediction accuracy is computed over all 14 chunks. This merging strategy utilizes 100% of non-anomalous samples from UNSW-NB15, 80.65% of DDoS samples, and 94.1% of DoS samples from BoT-IoT, resulting in a total of 64.322 million samples for training and validation. This approach effectively addresses data imbalance, minimizing overfitting, and maximizes the utilization of samples from both datasets for PB-DID model training and validation.

We created batches of 128 packets of one category and give them one label. To fulfill this requirement, we kept the closest multiple of 128 which came out to be 2,218,240. In the final configuration, PB-DID has 17330 packets of batch size 128 of all three categories.assures equal distribution of the data for the three classes in all chunks and mitigates the problem of over-fitting.

E. DEEP INTRUSION DETECTION

Deep Learning is a subclass of ML which mainly uses hierarchical stages in Artificial Neural Networks (ANN). Just like human brain, ANN's are built as a web linking neuron nodes. Although standard algorithms linearly build insights of data, the hierarchy of DL systems allows computers to interpret the data in a nonlinear way.

1) MODEL SELECTION

The PB-DID architecture employs an LSTM model featuring an input layer, two hidden layers, and an output layer. The input layer is an embedding layer that takes batches of size (26 x 128) as input, producing a 16-dimensional output fed into the first of two hidden LSTM layers. The embedding layer creates a vector for each training example, akin to the one-hot encoding function in Keras, but with the advantage of simultaneously using all features. The vector entries are initialized with random weights, and the embedding layer learns these weights iteratively. Both LSTM layers consist of 20 nodes, yielding a 20-dimensional output. Activation, recurrent activation functions, and dropout, recurrent dropout functions are applied in these LSTM layers. Two types of output layers are employed: one for binary classification and another for multi-class classification. In binary classification, the last layer is a dense layer with two neurons, representing non-anomalous vs. DDoS, non-anomalous vs. DoS, and DDoS vs. DoS classifications. For multi-class, the last layer is a dense layer with three neurons, outputting probabilities for each class. Notably, the input for classification includes only packets relevant to the specific classification being performed.

IV. RESULT

To evaluate the performance of algorithms, we used different metrics. Most of them are based on the confusion matrix. Confusion matrix is a tabular representation of a classification model performance on the test set. Here used two metrics for measuring the performance of the proposed PB-DID model namely confusion matrices and accuracy score. In a confusion matrix, there are four possible options, true positive (TP), true negative (TN), false positive (FP) and false negative (FN). The first part in every option shows whether the prediction is true or false and second part shows that prediction is positive or negative. The accuracy score shows the accuracy score of the predictions by the underlying deep model.

Accuracy: Accuracy is often the most used metric representing the percentage of correctly predicted observations, either true

or false. To calculate the accuracy of a model performance, the following equation can be used:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

In most cases, high accuracy value represents a good model, but considering the fact that we are training a classification model in our case, an article that was predicted as true while it was actually false (false positive) can have negative consequences; similarly, if an article was predicted as false while it contained factual data, this can create trust issues. Therefore, we have used three other metrics that take into account the incorrectly classified observation, i.e., precision, recall, and F1-score.

Recall: Recall represents the total number of positive classifications out of true class. In our case, it represents the number of articles predicted as true out of the total number of true articles.

$$R = \frac{TP}{TP + FN}$$

Precision: Conversely, precision score represents the ratio of true positives to all events predicted as true. In our case, precision shows the number of articles that are marked as true out of all the positively predicted (true) articles:

$$P = \frac{TP}{TP + FP}$$

F1-Score; F1-score represents the trade-off between precision and recall. It calculates the harmonic mean between each of the two.)us, it takes both the false positive and the false negative observations into account. F1-score can be calculated using the following formula:

$$F1 = \frac{2PR}{P + R}$$

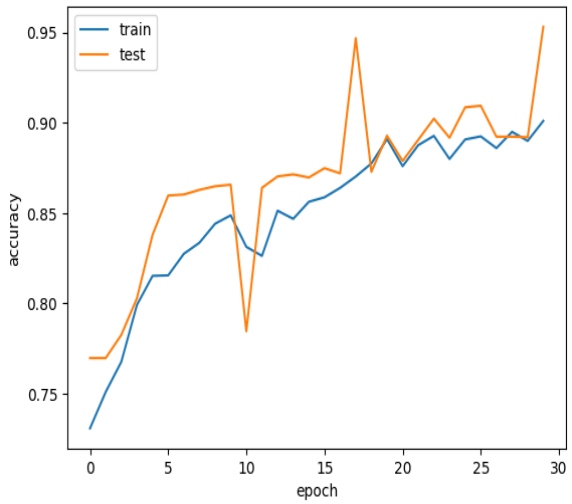

Figure 4. depicts the model accuracy.

In Figure 4, accuracy is depicted as a vital metric for assessing classification models. Informally, it signifies the proportion of correctly identified predictions. Formally, accuracy is defined as the ratio of correct predictions to the total predictions, serving as a clear and intuitive measure of the model's overall performance. This metric offers insights into the model's efficacy in making accurate classifications across the entire dataset.

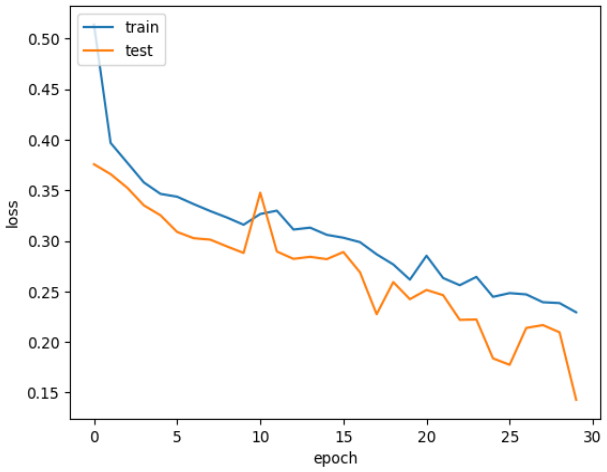

Figure 5. Model loss

In Figure 5, the model loss of the Unsupervised LSTM Deep Learning Model for Intrusion Detection indeed underscores its effectiveness in minimizing the difference between predicted and actual values during training, reflecting efficient learning. This indicates the model's skill in optimizing parameters to improve its predictive capabilities, thereby establishing its suitability for robust Intrusion Detection.

Table1 Comparison of different methods.

Model	Accuracy	Precision	Recall	F1-Score
SVM	0.9062	0.9087	0.9062	0.9066
DNN	0.9245	0.9245	0.9245	0.9245
DNN-LSTM	0.9531	0.9553	0.9531	0.9525

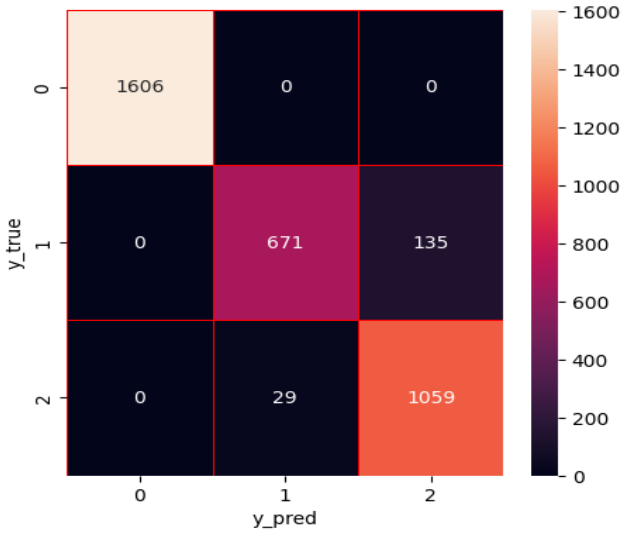

Fig. 6 Confusion matrices of classification models

Table 1 reveals the promising evaluation results of the proposed Unsupervised LSTM Deep Learning Model for Intrusion Detection, demonstrating its effectiveness compared to SVM and a basic DNN. With an impressive overall accuracy of 95.31%, surpassing SVM (90.62%) and DNN (92.45%), the model exhibits consistently high precision (95.53%), recall (95.31%), and F1-Score (95.25%). This balanced performance positions the Proposed-DNN-LSTM as a robust solution for Intrusion Detection, suggesting its potential for deployment in real-world scenarios crucial for securing systems and networks with timely and accurate intrusion detection.

In Fig. 6, the confusion matrices for DoS, DDoS, and BoT classification models are presented. These matrices offer a comprehensive overview of the performance of each model in terms of correctly and incorrectly classifying instances related to Denial of Service (DoS), Distributed Denial of Service (DDoS), and Bot (BoT) attacks.

V. CONCLUSION AND FUTURE WORK

In this study, we introduce PB-DID, a novel approach that compares the features of two prominent benchmark datasets, namely the UNSW-NB15 and Bot-IoT, both developed by researchers at the University of New South Wales. PB-DID analyzes and integrates the standard features of flow and TCP categories from both datasets, addressing issues like data imbalance and over-fitting by selecting an equal number of packets from each category. We utilize deep learning (DL) techniques to classify non-anomalous, DoS, and DDoS traffic, achieving a remarkable accuracy of 96.3% while covering both datasets comprehensively. A unique aspect of this work is the significant reduction in the number of features required for identifying malicious traffic, almost halving the feature count, and encompassing two latest benchmarked datasets.

Future endeavors will focus on enhancing the feature comparison and selection techniques by incorporating additional renowned and benchmark datasets. The goal is to broaden the spectrum of attack types, ensuring comprehensive coverage of threats to IoT devices in classification.

VI. REFERENCES

[1] Balaji, R.; Deepajothi, S.; Prabaharan, G.; Daniya, T.; Karthikeyan, P.; Velliangiri, S. Survey on Intrusions Detection System using Deep learning in IoT Environment. In Proceedings of the 2022 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS), Erode, India, 7–9 April 2022; IEEE: Manhattan, NY, USA; pp. 195–199.

[2] Alsamiri, J.; Khalid, A. Internet of Things cyber attacks detection using machine learning. Int. J. Adv. Comput. Sci. Appl. 2019, 10, 627–634. [CrossRef]

[3] Velliangiri, S.; Karthikeyan, P. Hybri4d optimization scheme for intrusion detection using considerable feature selection. Comput. Appl. 2020, 32, 7925–7939.

[4] Verma, A.; Virender, R. Machine learning based intrusion detection systems for IoT applications. Wirel. Pers. Commun. 2020, 111, 2287–2310. [CrossRef]

[5] Derhab, A.; Aldweesh, A.; Emam, A.Z.; Khan, F.A. Intrusion detection system for the Internet of Things based on temporal convolution neural network and efficient feature engineering. Wirel. Commun. Mob. Comput. 2020, 2020, 6689134. [CrossRef]

[6] Abusafat, F.; Pereira, T.; Santos, H. Proposing a Behavior- Based IDS Model for IoT Environment. In EuroSymposium on Systems Analysis and Design; Springer: Cham, Switzerland, 2018; pp. 114–134.

[7] Hussain, F.; Hussain, R.; Hassan, S.A.; Hossain, E. Machine learning in IoT security: Current solutions and future challenges. IEEE Commun. Surv. Tutor. 2020, 22, 1686–1721. [CrossRef]

[8] Elaziz, M.A.; Al-qaness, M.A.A.; Dahou, A.; Ibrahim, R.A.; El-Latif, A.A.A. Intrusion detection approach for cloud and IoT environments using deep learning and Capuchin Search Algorithm. Adv. Eng. Softw. 2023, 176, 103402. [CrossRef]

[9] Halim, Z.; Yousaf, M.N.; Waqas, M.; Sulaiman, M.; Abbas, G.; Hussain, M.; Ahmad, I.; Hanif, M. An effective genetic algorithmbased feature selection method for intrusion detection systems. Comput. Secur. 2021, 110, 102448. [CrossRef]

[10] Dubey, G.P.; Bhujade, R.K. Optimal feature selection for machine learning based intrusion detection system by exploiting attribute dependence. Mater. Today Proc. 2021, 47, 6325–6331. [CrossRef]

[11] Lazzarini, R.; Tianfield, H.; Charissis, V. A stacking ensemble of deep learning models for IoT intrusion detection. Knowl.-Based Syst. 2023, 279, 110941. [CrossRef]

[12] Alani, M.M. An explainable efficient flow-based Industrial IoT intrusion detection system. Comput. Electr. Eng. 2023, 108, 108732. [CrossRef]

[13] Nizamudeen, S.M.T. Intelligent Intrusion Detection Framework for Multi-Clouds–Iot Environment Using Swarm-Based Deep Learning Classifier. J. Cloud Comput. 2023, 12, 134. [CrossRef]

[14] Sharma, B.; Sharma, L.; Lal, C.; Roy, S. Anomaly based network intrusion detection for IoT attacks using deep learning technique. Comput. Electr. Eng. 2023, 107, 108626. [CrossRef]

[15] Kareem, S.S.; Mostafa, R.R.; Hashim, F.A.; El-Bakry, H.M. An effective feature selection model using hybrid metaheuristic algorithms for iot intrusion detection. Sensors 2022, 22, 1396. [CrossRef]

[16] Mohy-eddine, M.; Guezzaz, A.; Benkirane, S.; Azrour, M. An efficient network intrusion detection model for IoT security using K-NN classifier and feature selection. Multimed. Tools Appl. 2023, 82, 23615–23633. [CrossRef]

[17] Liu, X.; Du, Y. Towards Effective Feature Selection for IoT Botnet Attack Detection Using a Genetic Algorithm. Electronics 2023, 12, 1260. [CrossRef]

[18] Alweshah, M.; Hammouri, A.; Alkhalaileh, S.; Alzubi, O. Intrusion detection for the internet of things (IoT) based on the emperor penguin colony optimization algorithm. J. Ambient Intell. Humaniz. Comput. 2023, 14, 6349–6366. [CrossRef]

[19] Othman, S.M.; Ba-Alwi, F.M.; Alsohybe, N.T.; Al-Hashida, A.Y. Intrusion detection model using machine learning algorithm on Big Data environment. J. Big Data 2018, 5, 34. [CrossRef]

[20] Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey wolf optimizer. Adv. Eng. Softw. 2014, 69, 46–61. [CrossRef]

[21] Shareena, J.; Ramdas, A. APH Intrusion detection system for iot botnet attacks using deep learning. SN Comput. Sci. 2021, 2, 205.

[22] Alghanam, O.A.; Almobaideen, W.; Saadeh, M.; Adwan, O. An improved PIO feature selection algorithm for IoT network intrusion detection system based on ensemble learning. Expert Syst. Appl. 2023, 213, 118745. [CrossRef]

[23] Syed, N.F.; Ge, M.; Baig, Z. Fog-cloud based intrusion detection system using Recurrent Neural Networks and feature selection for IoT networks. Comput. Netw. 2023, 225, 109662. [CrossRef]

[24] Alhanaya, M.; Ateyeh Al-Shqeerat, K.H. Performance Analysis of Intrusion Detection System in the IoT Environment Using Feature Selection Technique. Intell. Autom. Soft Comput. 2023, 36, 3709–3724. [CrossRef]

[25] Khanday, S.A.; Fatima, H.; Rakesh, N. Implementation of intrusion detection model for DDoS attacks in Lightweight IoT Networks. Expert Syst. Appl. 2023, 215, 119330. [CrossRef] 23. Srivastav, D.; Srivastava, P. A two-tier hybrid ensemble learning pipeline for intrusion detection systems in IoT networks. J. Ambient Intell. Humaniz. Comput. 2023, 14, 3913–3927. [CrossRef]