Computers, Materials & Continua
DOI:10.32604/cmc.2021.018466
Article

Tech Science Press

Towards Machine Learning Based Intrusion Detection in IoT Networks

Nahida Islam[1], Fahiba Farhin[1], Ishrat Sultana[1], M. Shamim Kaiser[1], Md. Sazzadur Rahman[1], Mufti Mahmud[2], A. S. M. Sanwar Hosen[3] and Gi Hwan Cho[3,*]

[1]Institute of Information Technology, Jahangirnagar University, Dhaka, Bangladesh
[2]Department of Computer Science, Nottingham Trent University, Nottingham, UK
[3]Division of Computer Science and Engineering, Jeonbuk National University, Jeonju, 54896, Korea
[*]Corresponding Author: Gi Hwan Cho. Email: ghcho@jbnu.ac.kr
Received: 08 March 2021; Accepted: 09 April 2021

Abstract: The Internet of Things (IoT) integrates billions of self-organized and heterogeneous smart nodes that communicate with each other without human intervention. In recent years, IoT based systems have been used in improving the experience in many applications including healthcare, agriculture, supply chain, education, transportation and traffic monitoring, utility services etc. However, node heterogeneity raised security concern which is one of the most complicated issues on the IoT. Implementing security measures, including encryption, access control, and authentication for the IoT devices are ineffective in achieving security. In this paper, we identified various types of IoT threats and shallow (such as decision tree (DT), random forest (RF), support vector machine (SVM)) as well as deep machine learning (deep neural network (DNN), deep belief network (DBN), long short-term memory (LSTM), stacked LSTM, bidirectional LSTM (Bi-LSTM)) based intrusion detection systems (IDS) in the IoT environment have been discussed. The performance of these models has been evaluated using five benchmark datasets such as NSL-KDD, IoTDevNet, DS2OS, IoTID20, and IoT Botnet dataset. The various performance metrics such as Accuracy, Precision, Recall, F1-score were used to evaluate the performance of shallow/deep machine learning based IDS. It has been found that deep machine learning IDS outperforms shallow machine learning in detecting IoT attacks.

Keywords: IoT; shallow machine learning; deep learning; data science; IDS

1 Introduction

Internet of things (IoT) are growing exponentially and playing a vital role in our everyday life. IoT nodes can use internet protocol address and connect to internet. These self-configured smart nodes are driving beyond many cutting-edge applications such as process automation, home automation, smart cars, decision analytics, smart grids, health care system, educational development, industrial development and so on [1,2]. Analysts are predicting that there will be a society with more connected devices than people living on this planet. The International Data

Corporation (IDC) forecasted that there would be 41.6 billion connected IoT devices producing 79.4 zettabytes (ZB) of data in 2025 compared to the estimated population of 8.1 billion [3].

In IoT systems, heterogeneous nodes are connected to a complex network architecture and pose security concerns. The key challenge is to ensure security in resource constraint IoT nodes [4]. Otherwise, these IoT nodes are vulnerable to different types of attacks. IDSs are a rudimentary and powerful security mechanism in maintaining sufficient network protection in any IoT embedded environment [5,6]. They are proficient in monitoring, analyzing, and detecting real-time data packets through passive traffic collection even-if they are intruders or not. IDSs are traditionally organized into network-based (NIDS) and host-based IDS (HIDS) based on the detection places. Any IDS aims to monitor traffic and recognize different malware activities immediately [7,8**].** With the escalating number of anomalies, the upgrading and development of IDSs have become exceedingly important as the main challenge in intrusion detection is to find out the obscure attacks from the routine traffic flow [9]. Due to the dynamic approach of drawing a fine line between malware and benign data with high detection accuracy, shallow Machine Learning (ML) has become the center of attention of many researchers to upgrade the performance of IDSs [10]. Many supervised and unsupervised ML tools are effectively introduced for this purpose i.e., fuzzy logic (FL), support vector machine (SVM), artificial neural network (ANN), K-nearest neighbor (KNN), logistic regression (LR), hidden Markov model (HMM), genetic algorithm (GA), naive Bayes (NB), random forest (RF), decision tree (DT), decision forest (DF), decision jungle (DJ) and compacted hybrid algorithms, etc. [11–13].

In this research, data analysis-based techniques have been used as it works faster than others and performs better for the unspecified cases raised from unknown attacks. The essential objective of the framework is to build up a keen, secured, and trusted IoT based system that can differentiate its vulnerability, provide a protected firewall against all cyberattacks, and recuperate itself consequently. Thus, a learning-based methodology is proposed here which can recognize and ensure the infrastructure's security when it is in the anomalous condition. Despite the prevalence of utilizing conventional shallow ML strategies for classification problems, they still have numerous inadequacies that should be tended to, for example, the point of view of full representation of features, problem complexity, and static classification limitations. Deep learning (DL), having hierarchical architectures, is considered as a class of ML techniques that comprises numerous layers of data processing for classification and pattern recognition. Instead of conquering the previous lack of customary machine learning techniques, it puts an extraordinary mark on many research explorations recently. In light of DL achievement and stability, it has been effectively utilized in a wide scope of uses these days, for example, natural language processing, computer vision, and cybersecurity systems. For this errand, three shallow ML classifiers and five DL models have been exploited. Another vital part of this paper is that it has made the comparison of a simple model like DT or RF with a complex network like deep belief network (DBN), long short-term memory (LSTM), bidirectional long short-term memory (Bi-LSTM) for anomaly detection.

The main contributions of this research can be summarized as follows:

- A comprehensive workflow is proposed to predict optimal attack detection model in IoT systems.
- In addition to two most commonly used network attack datasets (such as NSL-KDD and DS2OS) three new datasets (such as IoTDevNet, IoTID20, and IoT Botnet) have been used.
- The performance of some shallow and deep machine learning algorithms has been evaluated using these five datasets through extensive experimentations.

The rest of the paper describes the state-of-the-art of this field and IoT threats in Section 2 and proposes methodologies, detailed dataset descriptions, learning model summary in Section 3. In Section 4, experimental setup, performance analysis, and comparative study with other existing works are explained. In the end, Section 5 presents concluding remarks with future scopes.

2 Literature Survey

2.1 State-of-the-art

This section includes some of the researches of various ML algorithms and classifiers integrated IDSs to detect intrusions in IoT networks. Roy et al. [14] has introduced a Bi-LSTM recurrent neural network (Bi-LSTM RNN) approach for intrusion detection aiming to identify a binary classification of normal and attack patterns. The implemented model has been trained using the UNSW-NB15 dataset and it achieves over 95% accuracy in IoT attack detection. Le et al. [15] developed a Botnet detection model using CNN with one class classification which used features derived from the system call graph. The model achieved an accuracy of 97% and an F-measure of 98.33% [16,17]. Almiani et al. [18] proposed a multi-layered deep RNN model to be implemented for IoT devices. The performance was evaluated using NSL-KDD dataset and found the detection rates of DoS, Probe, U2R, and R2L attacks are 98.27%, 97.35%, 64.93%, and 77.25%, respectively. Along with the detection rate, the overall performance has experimented with other works [19–22] in terms of accuracy, precision, false positive, and negative rate with less execution time. Another two-performance metrics have been added as Mathew correlation and Cohen’s Kappa coefficient reached 84.44% and 84.36%, respectively. According to the performance and experimental analysis, this work provides an almost perfect detection strategy against cyberattacks.

A novel intrusion detection system proposed by Xu et al. [23] analyzed the performance of some basic and hybrid RNN models named BGRU+MLP, GRU+MLP, BLSTM+MLP, LSTM+MLP, GRU, LSTM, and MLP to provide security against IoT attacks. The defined models are trained and tested using both KDD’99 and NSL-KDD dataset. For both of the datasets, BGRU+MLP provides the highest detection rate of 99.84% and 99.24%, respectively. Li et al. [24] implemented LSTM, GRU, Bi-LSTM, and Broad Learning System (BLS) algorithms on the NSL-KDD dataset for various known intrusion classification. The performance analysis determines that the BLS reduces the model training time with an overall accuracy of 84.15% and 72.64%, corresponding to KDDTest+ and KDDTest-21 datasets. Elmasry et al. [25] presented an empirical study on intrusion detection using the variations of ML and DL models: DF, DJ, DNN, DBN, LSTM-RNN, and GRU-RNN. Four datasets, namely, KDD CUP 99, NSL-KDD, CIDDS, and CICIDS2017, have been employed to scrutinize the performances of these algorithms to detect and classify anomalies in terms of 22 different evaluation metrics. However, the experiment result shows that DL models exploit the ML models, specifically DBN enhances the detection accuracy rate from 5% to 10% than others. Furthermore, a heuristic approach for intrusion detection designed by Ayyaz et al. [26] showcased an accuracy from 85.5% to 95.25% for RNN-IDS. The IDS is trained by gradient descent algorithm beforehand and later on again trained and tested with KDD20+ and KDDTest+ dataset. The performance of RNN-IDS surpasses the other applied algorithms, namely, J48, SVM, NB, NB Tree, MLP, RF, RF Tree, and ANN.

A hybrid sampling-based intrusion detection by Jiang et al. [27] is experimented with the combination of NSL-KDD and UNSW-15 datasets, separately. The combination of SMOTE and OSS is applied to construct a balanced dataset to train the models developed using RF, CNN, BiLSTM, CNN-BiLSTM, AlexNet, and LeNet-5 classifiers. The statistical result claims that the

CNN-BiLSTM outperformed other algorithms with an accuracy of 83.58% and 77.16% for the mentioned datasets, respectively. Dushimimana et al. [28] introduced a Bi-RNN based intrusion detection system to investigate the efficiency of some other DL algorithms, namely RNN and GRNN. The performance is evaluated using 10% KDD dataset and the Bi-RNN records the best accuracy of 99.04% compared to others.

Hasan et al. [29] discussed some paradigmatic ML techniques for intrusion detection in IoT networks leading to system failure. Five-fold cross-validation has been performed on the DS2OS dataset using each of the considered ML techniques: LR, SVM, DT, RF, and ANN. Among these, RF performs more accurately, about 99.4%, to detect IoT attacks though the performance of RF will be this better is not guaranteed in case of real-time or vast enormous unknown data. Cheng et al. [30] proposed a semi-supervised Hierarchical Stacking Temporal Convolutional Network (HS-TCN) to detect anomalies in IoT communication. The experiment conducted using two of the variations of the original dataset DS2OS-the data gathered for 11 days (DS20S-A) and the other one is the under-sampled value of these collected data (DS2OS-UA). The HS-TCN model performs better in comparison to LSTM and SVM for both of the modified datasets. Another ML-based anomaly detection method was devised by Sahu et al. [31] through the implementation of LR and ANN classification algorithms in a two-fold way. Both LR and ANN obtain around 99.4% accuracy using the full dataset while the accuracy of 99.99% is achieved after the omission of around 1,05,952 data from the original dataset. In both of the cases, the dataset is split into a proportion of 75% and 25%, respectively.

Latif et al. [32] introduced a novel strategy for attack detection in industrial IoT using an advanced and lightweight scheme of ANN, the Random Neural Network (RaNN). The suggested RaNN is trained by the gradient descent (GD) algorithm. The dataset is processed by discarding the feature ‘Source ID’ to obtain an accuracy of more than 99%, which eventually outperformed the accuracy score of ANN, SVM, and DT. A DNN architecture is presented by Reddy et al. [33] to secure the applications of future smart cities. Around a number of 2,198 null rows are excluded from the original dataset, and the rest of the data partitioned into the ratio of 70% and 30% for training and testing purposes, respectively. The result analysis shows that around 98.26% accuracy is achieved by this DNN strategy even with a different number of layers and neurons in a comparison of conventional ML classifiers.

Above surveyed studies regarding IoT security are summarized according to their datasets, models and best accuracy result in Tab. 1.

Table 1: IoT security literature survey summary

Ref.	Method	Dataset	Contribution	Limitation
[14]	Bi-LSTM	UNSW-NB15	IDS classifier detected normal or attack types with 95% accuracy.	It failed to identify various types of attack. Besides, parameters were not optimized.
[15]	CNN	System Call Graph	It used CNN based model to detect Botnet using system call graphs with an accuracy of 97% and an F-measure of 98.33%.	No experiment was done for other malicious lines on IoT devices.

(Continued)

Table 1: Continued

Ref.	Method	Dataset	Contribution	Limitation
[18]	RNN	NSL-KDD	Fog computing-based IDS having multi-layered deep RNN with higher detection rate (DoS: 98.27%, Probe: 97.35%, U2R: 64.93%, R2L: 77.25%).	It explored a single dataset only without explaining the hyper-parameters tuning.
[23]	BGRU+MLP, GRU+MLP, BLSTM+MLP, LSTM+MLP, GRU, LSTM, MLP	KDD99, NSL-KDD	Four types of attacks such as DOS, Probe, U2R and R2L had been detected using multiple models with high accuracy where BGRU+MPL performs well achieving 99.24% accuracy.	More generic attacks were detected exploring a single dataset only.
[24]	LSTM, GRU Bi-LSTM, BLS	NSL-KDD	Three types of RNN and BLS models were applied to detect intrusions where BLS outperforms with 84.14% accuracy and 84.68% F-measures.	A single simple network dataset was considered only. No hyper-parameter tuning was done.
[25]	DF, DJ, DNN, DBN, LSTM, GRU	NSL-KDD, KDD99, CICIDS	An empirical extensive study on NIDS as multiclass classification using four DL models and two shallow ML models with multiple evaluation matrices where DBN outperforms with an accuracy of 96.9%.	No IoT attack datasets are studied.
[26]	J48, SVM, NB, NB Tree, MLP, RF, RF Tree, ANN and RNN-IDS	NSL-KDD	NIDS using the feed-forward nature of Random Neural Networks (RNN-IDS) was proposed and compared with multiple ML algorithms where RNN-IDS accuracy reached up to 95.2%.	Only ML models and a single dataset are considered for performance comparison. No hyper-parameter tuning.
[27]	RF, CNN, Bi-LSTM, CNN-BiLSTM, AlexNet, LeNet-5	NSL-KDD, UNSW-15	Hybrid sampling and a deep hierarchical network constructed on CNN and BiLSTM were proposed with an accuracy of 83.58%.	No performance comparison with related studies. Multiple models were tested with two similar types of datasets.
[28]	Bi-RNN, RNN, GRNN	10% KDD	IDS was designed and evaluated using the Bi-RNN model which outperformed RNN and GRNN with a 99.04% detection rate.	Only 10% of the full KDD dataset was used, failed to analyze the performance using a large amount of data, and to classify multi-class attack data.

(Continued)

Table 1: Continued

Ref.	Method	Dataset	Contribution	Limitation
[29]	LR, SVM, DT, RF, ANN	DS2OS	Data-analysis based IDS model was explored where RF model achieved higher accuracy (99.4%) than other ML models, which could detect multi-class attacks more accurately.	With a single dataset, only ML models were implemented. The efficiency of the RF model for a large volume of data was not examined.
[30]	TCN, LSTM, SVM	DS2OS	A semi-supervised HS-TCN model outperformed the other two models with 98.15% accuracy. Besides, a balanced version of the dataset has also been evaluated along with the original one.	Lack of efficiency optimization of semi-supervised models and failed to identify multi-class attack types.
[31]	LR, ANN	DS2OS	The dataset was experimented with a data-analysis based technique in a two-fold way, using LR and ANN models. In both cases, LR and ANN detected attacks with equal accuracy (99.4%, 99.99%), respectively.	Indistinct analysis of achieved result and no parameter optimization.
[32]	RaNN	DS2OS	Achieved 99.20% attack detection accuracy with less prediction time using an advanced and lightweight scheme of ANN, the RaNN model.	Only a single dataset was used. No statistical comparison of attack data for experimented SVM and DT models.
[33]	DNN	DS2OS	Seven types of attacks were classified with 98.26% accuracy using an optimized DNN model.	Complex DL models had not analyzed.

In Tab. 1, we found that these models are not created with several IoT and other attack datasets; many studies did not consider hyper parameters tuning. Further, there is no discussion of comparison between shallow ML and DL models, complex DL models are less researched, and training and validation have not been widely observed. That is why we have proposed the IDS model based on shallow ML/DL with five datasets which includes this gap discussed above.

2.2 Iot Threats

With the distributed nature, an IoT network is a layered architecture where every layer sequentially maintains individual tasks to run this whole platform efficiently. Intrusion happens in every layer to breach the security where researchers have found multiple attacks occurring in the whole network including the protocol and gateways [34]. Fig. 1 shows the IoT attack scenario inclusively where the IoT framework is considered as a combination of various IoT sensors and devices, networks, platforms, applications, management, and services. Some intrusions have recently gained much attention to security analysts due to their increasing rate and DoS, Probe, U2R, and R2L are considered as the encapsulated form of these attacks occurred in the whole network [35].

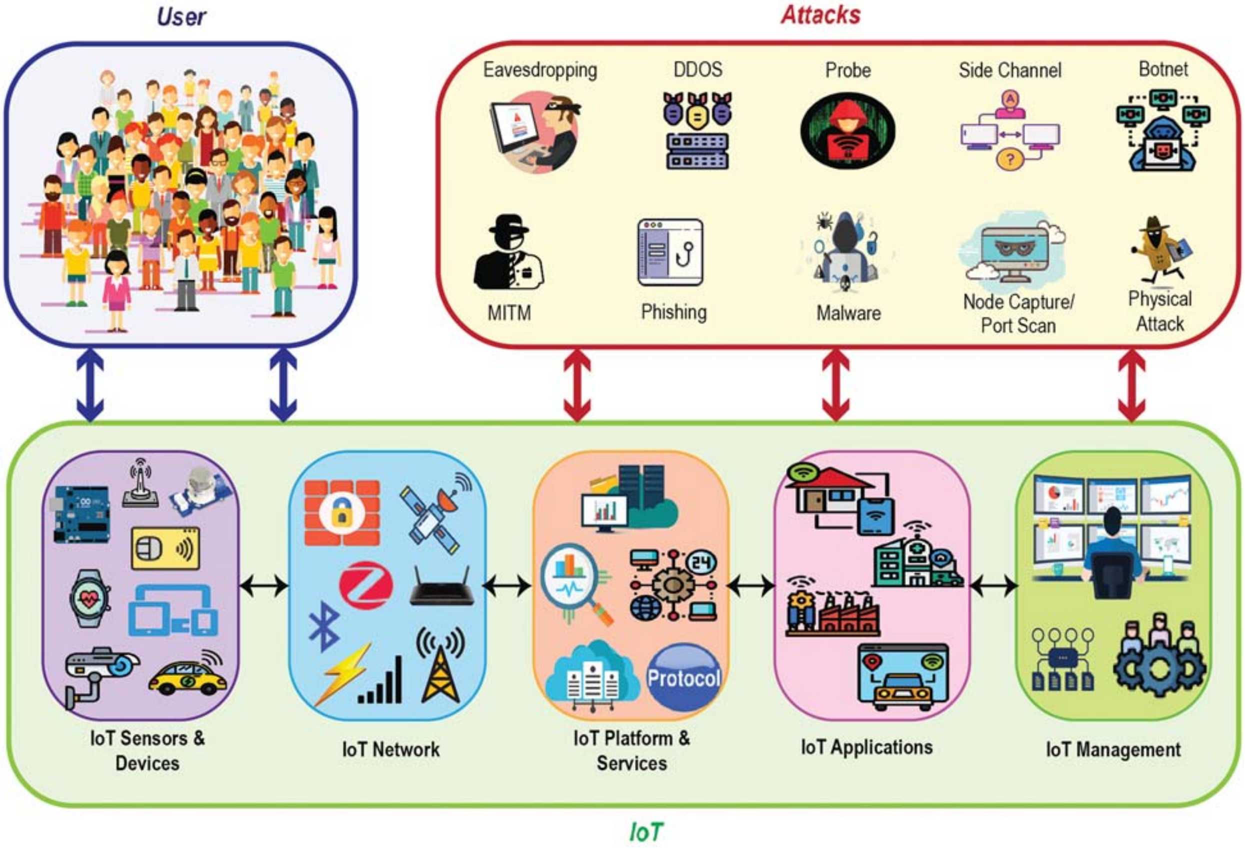

Figure 1: Relationship visualization of IoT ecosystem and potential threats

3 Methodology

The strategic framework to detect attacks on an IoT network follows the combination of some elementary steps. Fig. 2 portrays the pictorial vision of this whole structure, where the first and foremost step is to collect the dataset, observe thoroughly, and analyze the features and their data types. In the next step, the pre-processing of the dataset is performed to be feedable to learning algorithms. The processed dataset is split into train and test sets with a ratio of 80:20, respectively. The train set is used for the training process with the considered ML/DL learning algorithms. Lastly, the final models are assessed using the test set in terms of the considered performance metrics.

3.1 Dataset Collection and Description

In this research work, five datasets have been used to train and evaluate three shallow ML and five DL models. To analyze intrusion detection methods, two widely used datasets (NSL-KDD, DS2OS) and other three new datasets (IoT Device Network Logs, IoT Intrusion Dataset 2020, IoT Botnet Dataset 2020) have been chosen based on the IoT attack variation. The first dataset is NSL-KDD, extensively used as a benchmark dataset improvised from the original KDD’99 dataset by eliminating the redundancy of 78% and 75% train and test set records, respectively [36]. The dataset contains around 42 features (41 independent, 1 dependent) and a separated train and test set, denoted as KDDTrain+ and KDDTest+ with total records of 1,25,973 and 22,543, respectively. Besides, there is a total of 39 attacks classified into four attack classes: DoS, Prob, U2R, and R2L. As for the proposed experiment, the 20% records of the entire train set, KDDTrain+_20, and the subset of the test set, KDDTest-21 (deprecated of most difficult traffic records, a score of 21), are considered [37]. Around 25,192 and 11,850 records with the same number of features are assigned to the corresponding datasets, KDDTrain+_20 and KDDTest-21, respectively. The second dataset IoT Device Network Logs is discovered from Kaggle [38] and preprocessed according to the network-based intrusion detection systems in IoT devices. To monitor the network and collect the records of events, Ultrasonic Sensor with Arduino and NodeMCU are used. The collected network logs are sent to the server via NodeMCU with the ESP8266 Wi-Fi module. The dataset contains a total of 4,77,426 data with 14 distinct features and classified into five classes: Normal, Wrong Setup, DDoS, Data Type Probing, Scan attack, and MITM.

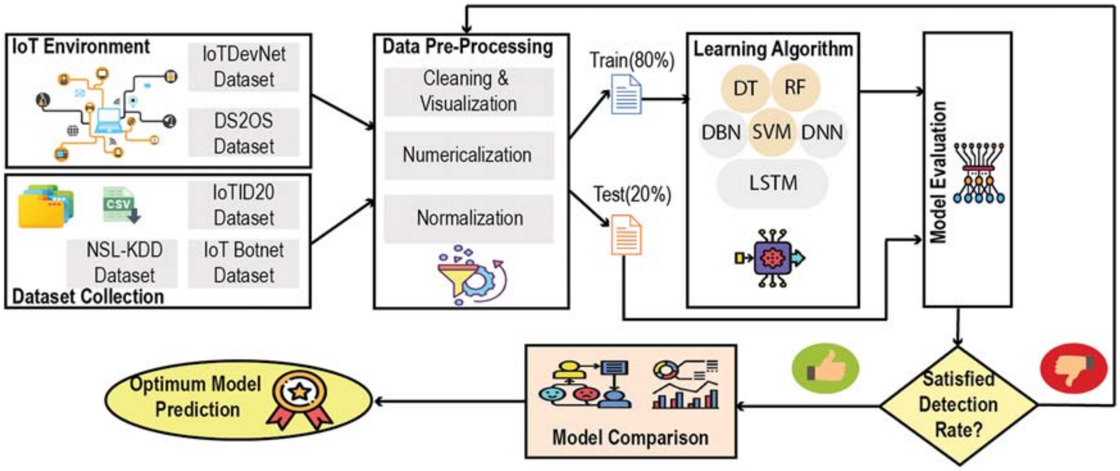

Figure 2: The proposed workflow for predicting optimum attack detection model

Tab. 2 presents the nuts and bolts of the attack and normal records of the corresponding dataset. Tab. 3 lists the features and their data types. The third dataset, 'Distributed Smart Space Orchestration System' acronym as DS2OS, an open-source synthetic dataset collected from Kaggle and provided by M. Pahl et al. [39]. They have generated the dataset by capturing traffic traces from the application layer of four different IoT sites with divergent types of services: light controller, thermometer, movement sensors, washing machines, batteries, thermostats, smart doors, and smartphones for a duration of 24 hrs. There are a total of 3,57,952 records and 13 features in the dataset along with eight non-identical classifications: DoS, Data type Probing, Malicious Control, Malicious Operation, Scan, Spying, Wrong Setup, and Normal.

The 'IoT Intrusion Dataset 2020' acronym as IoTID20 is the fourth dataset, adopted by I. Ullah et al. [40] generated from [41]. The IoTID20 dataset contains 83 network features and three label features along with 625,783 records. The total records are categorized into five classes specifically, Mirai, Scan, DoS, Normal, and MITM. These classes are again categorized into seven sub-classes: Mirai Brute force, Mirai HTTP Flooding, Mirai UDP Flooding, Scan Host Port, Scan Port OS, Syn Flooding, and ARP Spoofing respectively. Tab. 4 represents the overall distribution of attack and normal records accurately.

Table 2: Record distribution of IoTDevNet dataset

IoT Device Network Logs Dataset			
Class	frequency count	% of total records	% of attack records
Normal	79,035	16.55%	-
Wrong Setup	82,285	17.24%	20.65%
DDoS	79,020	16.55%	19.83%
Data Type Probing	79,002	16.55%	19.83%
Scan	79,052	16.56%	19.84%
MITM	79,032	16.55%	19.84%

Table 3: Feature description of IoTDevNet dataset

SI.	Features	Data Types	SI.	Features	Data Types
1	frame.number	numeric	8	ip.proto	numeric
2	frame.time	numeric	9	ip.len	numeric
3	frame.len	numeric	10	tcp.len	numeric
4	eth.src	numeric	11	tcp.srcport	numeric
5	eth.dst	numeric	12	tcp.dstport	numeric
6	ip.src	numeric	13	Value	numeric
7	ip.dst	numeric	14	normality	numeric

Table 4: Record distribution of IoTID20 dataset

IoT Intrusion Dataset 2020			
Class	frequency count	% of total records	% of attack records
Mirai	4,15,677	6.43%	70.97%
Scan	75,265	12.03%	12.85%
DoS	59,391	9.49%	10.14%
Normal	40,073	6.40%	-
MITM ARP Spoofing	35,377	5.65%	6.04%

The fifth dataset is 'IoT Botnet Dataset 2020' developed based on a comprehensive IoT network by I. Ullah et al. [42] using a network traffic flow analyzer to improve and increase the number of flow and network features. The original one has 46 network features and two label features with a limited number of flow features. But the developed one has 83 network features with three labeling features, namely, 'Label', 'Cat', and 'Sub_cat'. It contains a total of 1,940,389 records (10% of the full dataset) and classified into two labels: normal or anomalous, and five classes: DoS, DDOS, Reconnaissance, Normal, and Theft, which is again organized into eleven sub-classes: Normal, DDoS-HTTP, DDoS-TCP, DDoS-UDP, DoS-HTTP, DoS-TCP, DoS-TCP, OS-Fingerprint, Service-Scan, Keylogging, and Data-Exfiltration. The detailed record distribution of the collected dataset is analyzed in Tab. 5. Tab. 6 lists the features and their data types for both IoTID20 and IoT Botnet Dataset 2020.

Table 5: Record distribution of IoT Botnet dataset 2020

IoT Botnet Dataset 2020			
Class	frequency count	% of total records	% of attack records
DoS	6,51,122	33.56 %	35.33 %
DDoS	6,36,539	32.8 %	34.53 %
Reconnaissance	5,55,011	28.60 %	30.11 %
Normal	97,197	5.00 %	-
Theft	520	0.027 %	0.028%

Table 6: Feature description of IoTID20 and IoT Botnet dataset 2020

SI.	Features	Data types	SI.	Features	Data types	SI.	Features	Data types
1	Flow_ID	object	30	Fwd_IAT_Max	float64	59	Pkt_Size_Avg	float64
2	Src_IP	object	31	Fwd_IAT_Min	float64	60	Fwd_Seg_Size_Avg	float64
3	Src_Port	int64	32	Bwd_IAT_Tot	float64	61	Bwd_Se_Size_Avg	float64
4	Dst_IP	object	33	Bwd_IAT_Mean	float64	62	Fwd_Byts/b_Avg	int64
5	Dst_Port	int64	34	Bwd_IAT_Std	float64	63	Fwd_Pkts/b_Avg	int64
6	Protocol	int64	35	Bwd_IAT_Max	float64	64	Fwd_Blk_Rate_Avg	int64
7	Timestamp	‘0’	36	Bwd_IAT_Min	float64	65	Bwd_Byts/b_Avg	int64
8	Flow_Duration	int64	37	Fwd_PSH_Flags	int64	66	Bwd_Pkts/b_Avg	int64
9	Tot_Fwd_Pkts	int64	38	Bwd_PSH_Flags	int64	67	Bwd_Blk_Rate_Avg	int64
10	Tot_Bwd_Pkts	int64	39	Fwd_URG_Flags	int64	68	Subflow_Fwd_Pkts	int64
11	TotLen_Fwd_Pkts	float64	40	Bwd_URG_Flags	int64	69	Subflow_Fwd_Byts	int64
12	TotLen_Bwd_Pkts	float64	41	Fwd_Header_Len	int64	70	Subflow_Bwd_Pkts	int64
13	Fwd_Pkt_Len_Max	float64	42	Bwd_Header_Len	int64	71	Subflow_Bwd_Byts	int64
14	Fwd_Pkt_Len_Min	float64	43	Fwd_Pkts/s	float64	72	Init_Fwd_Win_Byts	int64
15	Fwd_Pkt_Len_Mean	float64	44	Bwd_Pkts/s	float64	73	Init_Bwd_Win_Byts	int64
16	Fwd_Pkt_Len_Std	float64	45	Pkt_Len_Min	float64	74	Fwd_Act_Data_Pkts	int64
17	Bwd_Pkt_Len_Max	float64	46	Pkt_Len_Max	float64	75	Fwd_Seg_Size_Min	int64
18	Bwd_Pkt_Len_Min	float64	47	Pkt_Len_Mean	float64	76	Active_Mean	float64
19	Bwd_Pkt_Len_Mean	float64	48	Pkt_Len_Std	float64	77	Active_Std	float64
20	Bwd_Pkt_Len_Std	float64	49	Pkt_Len_Var	float64	78	Active_Max	float64
21	Flow_Byts/s	float64	50	FIN_Flag_Cnt	int64	79	Active_Min	float64
22	Flow_Pkts/s	float64	51	SYN_Flag_Cnt	int64	80	Idle_Mean	float64
23	Flow_IAT_Mean	float64	52	RST_Flag_Cnt	int64	81	Idle_Std	float64
24	Flow_IAT_Std	float64	53	PSH_Flag_Cnt	int64	82	Idle_Max	float64
25	Flow_IAT_Max	float64	54	ACK_Flag_Cnt	int64	83	Idle_Min	float64
26	Flow_IAT_Min	float64	55	URG_Flag_Cnt	int64	84	Label	int64
27	Fwd_IAT_Tot	float64	56	CWE_Flag_Count	int64	85	Cat	Object
28	Fwd_IAT_Mean	float64	57	ECE_Flag_Cnt	int64	86	Sub_Cat	Object
29	Bwd_IAT_Mean	float64	58	Down/Up_Ratio	float64	-	-	-

3.2 Dataset Pre-processing

Any ML/DL research requires radical data analysis to be compatible with the learning algorithms to accomplish accurate performance. Hence, it is necessary to perform pre-processing to convert the data from categorical values to numeric. There are two fundamental processes of pre-processing: Numericalization and Normalization. Before performing these, a few steps need to execute as dataset cleaning, finding out the missing values, and replacing the 'NaN' (Not a Number) values.

3.2.1 Data Cleaning

Both the KDDTrain+_20 and KDDTest-21 datasets contain 42 identical features, whereas only one feature has missing values, namely, 'num_outbound_cmds'. IoT Network Device Logs Dataset is abbreviated as 'IoTDevNet' for the supremacy of the further experimental work. As it contains no missing or 'NaN' values, therefore, there is no need for further data filtering. As for DS2OS, there are two columns- 'Accessed Node Type' and 'Value' containing missing values. The data types of these columns are categorical and continuous, respectively. The 'Accessed Node Type' column has a total of 148 rows containing 'NaN' values needed to replace or remove. Removal of these 148 rows might results in a loss of valuable data, eventually in substandard performance. Hence, the 'NaN' values are restored with 'Malicious' values. There are also some unassigned data in the 'Value' column as 'False', 'True', 'Twenty', and 'None' transformed into values '0.0', '1.0', '20.0', and '0.0', respectively.

The IoT Botnet Dataset 2020 and the IoTIDS20 dataset both contain similar type and number of features with slightly variated attack classes. Before performing normalization on these two datasets, the empty spaces are dropped, useless indices are deprecated and the data types are converted into appropriate type (float) to avoid errors. In these two datasets, there are 16 features containing missing values which are removed to compute accurate result.

3.2.2 Numericalization

For each dataset, the data types of the features need to be determined beforehand for numericalization. In NSL-KDD, there are three categorical nominal variables in the 'Service', 'Flag', and 'Protocol_type' columns as well as in DS2OS; except 'Value' and 'Timestamp'; other columns contain categorical nominal variables [29,36]. According to the inspection from Tab. 3 and Tab. 6, there is no categorical value in IoTDevNet dataset, but there are object values in two columns of both IoTID20 and IoT Botnet Dataset 2020, namely, 'Cat' and 'Sub_Cat'. These categorical variables are converted into numeric by applying label encoding. Label encoding yields the same dimension of the dataset as before and assigns the same values to the repeated labels with low memory consumption. Hence, label encoding is applied to all of these four datasets to convert the categorical type features into numeric values.

3.2.3 Normalization

After numericalization, the StandardScaler method is applied to the continuous numerical data (particularly the high range ones) of the five concerned datasets for normalization. Standard-Scalar follows a normal distribution and scales the data by subtracting the mean 0 and dividing by the standard deviation 1.

$$\frac{x_i - mean(x)}{stdev(x)} \tag{1}$$

For feature x_i, the value of mean and standard deviation of x features are calculated and x_i is scaled based on the above Eq. (1) [43].

3.3 *Learning Algorithms*

3.3.1 *Shallow Machine Learning*

ML is an Artificial Intelligence (AI) branch that is closely related to computational statistics which uses mathematical optimization to emphasis on prediction making. It is known to be unsupervised learning for different individuals to learn and develop baseline behavioral profiles and then to identify meaningful abnormalities. Shallow Learning is a form of ML in which models learn from predefined features represented by data. This study considers three shallow ML techniques, namely, DT, RF, and SVM. DT is a flowchart-like classification or regression model that works by separating a dataset into several smaller subsets while gradually evolving a related decision tree with decision and leaf nodes simultaneously [11]. RF, a supervised ML algorithm, justifies its name by creating the forest with several DTs. RF constructs DTs on randomly chosen data samples, gets a forecast from each tree, and picks the best solution through voting [44]. SVM is another well-known supervised ML technique used for regression and classification analysis by discovering a max-margin separation hyperplane in the n-dimension space [45].

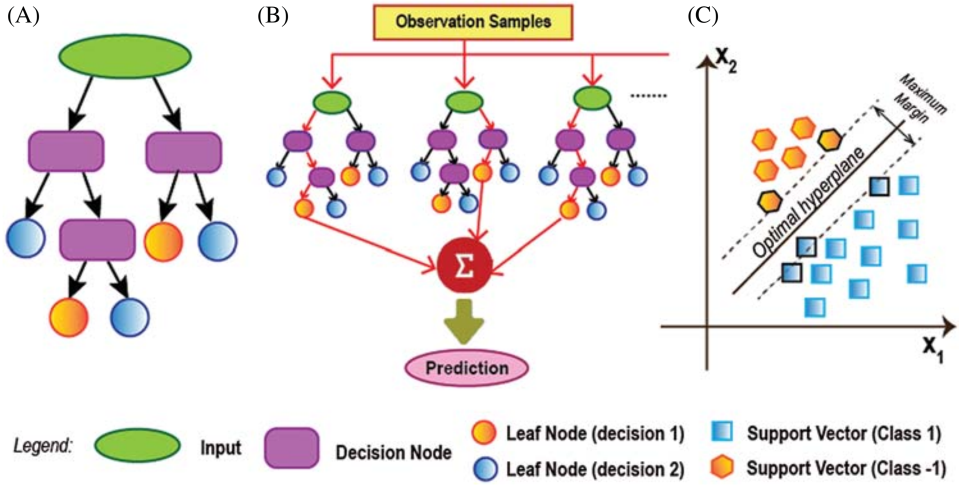

Figure 3: Different machine learning techniques: (a) working process of DT that takes different decision depending on various conditions. (b) decision-making technique of RF which calculates the final prediction by summarizing different DT's result. (c) classification procedure of linear SVM in two-dimensional case that creates a hyper-plane among the support vectors of different classes

3.3.2 *Deep Learning*

DL is a subfield of shallow ML and is much more implicated for processing big data and ensuring IoT network security. In this subsection, an overview of five different DL models has been introduced, namely, DNN, DBN, LSTM, Stacked LSTM, and Bi-LSTM. DNN is a sort of neural network demonstrated as a multilayer perceptron (MLP) prepared with algorithms to take in portrayals from datasets with no manual design of feature extractors [46]. On the other hand, DBN is a probabilistic generative model framed by stacking Restricted Boltzmann Machines (RBMs) with already pre-trained hidden layers (more or less) [47]. LSTM is an extension of conventional RNN capable enough to solve exploding and vanishing gradient problems to upgrade the performance. A stacked LSTM model is nothing but a composition of multiple LSTM layers and Bi-LSTM is a variant of LSTM, with bidirectional long-term temporal dependencies, facilitates the processing in two opposite direction-backward and forward [48,49].

Figs. 3 and 4 illustrate the detailed architecture of three shallow ML and five DL algorithms discussed in this section.

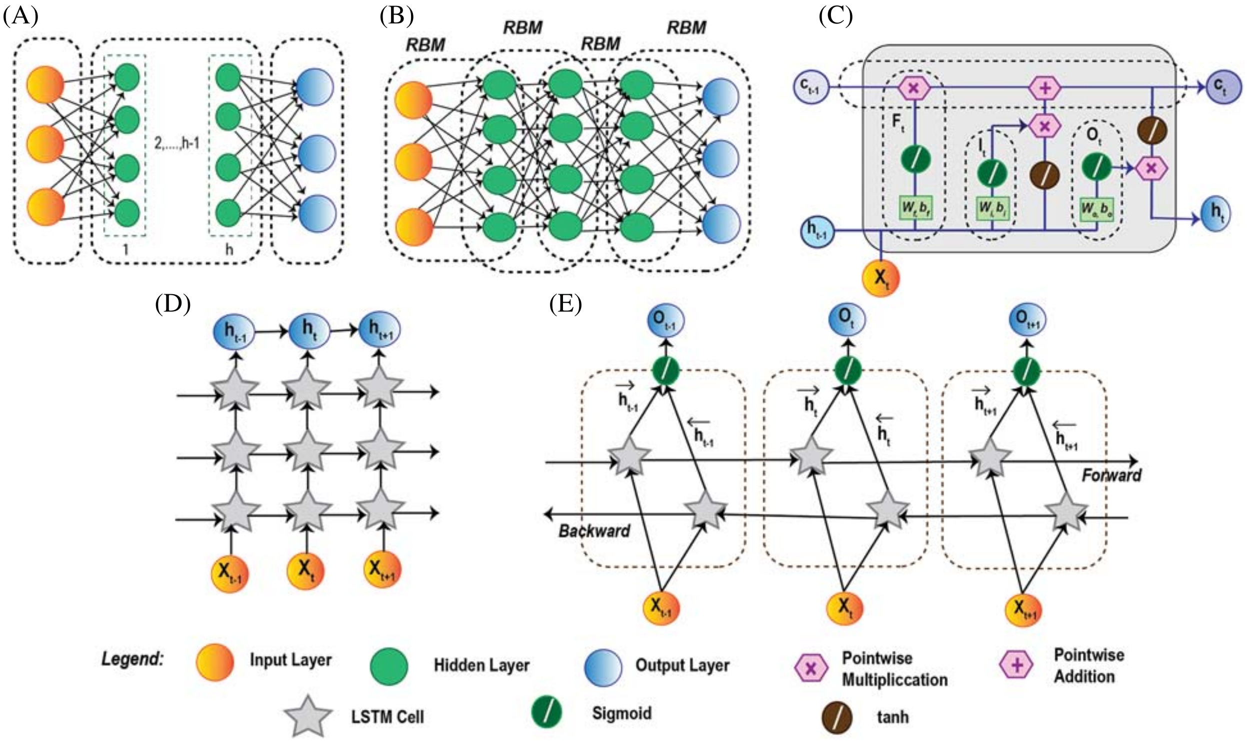

Figure 4: Deep learning algorithm architecture: (a) DNN architecture with fully connected input, output and h number of hidden layers. (b) illustration of DBN structure with multiple layers of stacked RBM. (c) LSTM cell architecture with input X_t. It has three gates: F_t (forget), I_t (input), and O_t (output) with corresponding weights, bias values, and activation functions sigmoid, tanh. (d) Stacked LSTM architecture concatenated with multiple LSTM layers. The previous LSTM layer's output h $_{t-1}$ works as the input of the next layer h $_t$ to produce the final output h $_{t+1}$. (e) Bi-LSTM architecture composed of three hidden LSTM layers in two opposite directions, forward and backward. X $_{t-1}$, X_t, X $_{t+1}$ are the inputs of the corresponding hidden layers with outputs O $_{t-1}$, O_t, O$_{t+1}$, respectively

4 Evaluation and Result Analysis

4.1 Experimental Environment

The experiment is conducted on a personal HP laptop where the microprocessor is 2.5 GHz Intel Core i7-6500U with Intel® HD Graphics 520, 8 GB RAM, and Windows 10 operating system. The models are executed on an open-source platform Google Colab Notebook. Experimented shallow ML and DL models are implemented from the Keras layer using TensorFlow 1.14.0. As for loading, cleaning data, Pandas, and NumPy frameworks are used. Again, Matplotlib and Seaborn frameworks are implemented for data visualization. Finally, the performance of the experiment is analyzed using the scikit-learn framework.

4.2 Parameter Optimization

Shallow ML models are self-biased learning algorithms that work best on moderate data, whereas DL algorithms perform more efficiently on intensive and larger data which requires complex hidden patterns consideration. All shallow ML/DL techniques are trained and evaluated for multi-class classifications with five mentioned datasets. Five-fold cross-validation is performed on each of the datasets using all of these techniques to estimate the skill of learning algorithms on unseen data.

During the simulation of the proposed models, the parameters listed in Tabs. 7 and 8 are tuned in such a way so that both shallow ML and DL models can achieve better results with the same values. As for ML models, the parameters of DT, RF, and SVM are optimized based on the input data volume. The IoT Botnet dataset is the largest for any ML model to consider for performance acceptance among the datasets discussed in Section 3.1. Hence, the parameters of the ML models have been fine-tuned for this dataset. For DT, the parameter 'max_depth' is optimized only, while the index, splitter, and other parameters are set to default. Using the variation of 'max_depth' as 2, 3, 4, and 5, the tree produces an accuracy of 84.45%, 99.72%, 99.99%, and 100% with a time variation of 651 s, 663 s, 781 s, and 713 s, respectively. As for RF, 'max_depth' and 'estimators' are tuned in the combination of (2, 100), (3, 100), (3, 50), and (4, 100), respectively. With these values, the RF tree results in an accuracy of 90.86%, 98.30%, 98.27%, and 99.46% with a time variation of 8510 s, 12156 s, 8612 s, and 15578 s, respectively. From the achieved results, DT with 'max_depth' = 3 performs better in less time than other ones and RF with 'max_depth' = 3 and 'n_estimators' = 100 performs better in terms of accuracy and time though with the 'max_depth' = 5, it produces the better result, but it takes a lot of time. With these parameters tuning of DT and RF, other datasets are trained and evaluated with satisfactory results (discussed in Section 4.3). As the learning rate of SVM is very high, it is not recommended to train SVM with larger datasets, hence the IoTID20 and IoT Botnet datasets are not trained with SVM. According to the data volume of the DS2OS dataset, the 'regularization parameter' is set to 1000, while kernel type, degree, tolerance, and rest of the parameters are selected as default. With a variation of the 'regularization parameter' as 0.1, 1, 10, 100, and 1000, the SVM classifier obtains 99.13%, 99.49%, 99.40%, and 99.40% detection rate in 9270 s , 14908s, 10163 s, and 8154 s, respectively. As per the inspection, the achieved score for 'regularization parameter' = 1000 is the higher one with the lowest time than others. Furthermore, with the same tuning parameters, SVM also performs better for the other two datasets-NSL-KDD and IoTDevNet. Therefore, the evaluation of similarly tuned three shallow ML models on five different datasets is quite acceptable.

Table 7: Parameter values for performance evaluation of three shallow ML algorithms

Parameters	Shallow ML Algorithms		
	DT	RF	SVM
max_depth	3	3	–
n_estimators	–	100	–
regularization parameter (C)	–	–	1000

Table 8: Parameter values for performance evaluation of five DL algorithms

Parameters	DL Algorithms				
	DNN	DBN	LSTM	Stacked LSTM	Bi-LSTM
Input neuron	800	256	4	16	80
Hidden neuron	400	–	–	16	40, 128
Epochs	1000/10	10 (rbm)	10	10	15
Batch size	64	32	64	64	64
Optimizer	adam	–	adam	adam	adam
Dropout rate	0.9	0.2	0.6	0.4	0.1
Activation	relu, softmax	Relu	softmax	softmax	relu, softmax
Loss	categorical	–	categorical	categorical	categorical

DL models are complex neural networks with input, hidden and output layers having multiple neurons. Parameter optimization regarding mentioned five algorithms is mostly related to these hidden number of neurons. Dropout regularization is a strategy for minimizing overfitting and enhancing deep neural network generalization. Fixed dropout for each of the individual model is considered here for better performance which is measured based on accuracy and evaluation time. So, to acquire optimized result, hidden neuron number and the number of epochs in each fold are tuned for each model. At first, considering DNN for the DS2OS dataset, it is considered as a sequential model with three layers of hidden neurons with dropouts. The model is experimented with (400, 200), (800, 400), (1200, 800), and (100, 50) neuron sets which achieve 98.84%, 99.11%, 99.12%, and 97.17% accuracy with a time variation of 892 s, 1661 s, 3620 s, and 437 s. So, considering test and train accuracy with the training time, (800, 400) hidden neurons with a dropout 0.9, optimizer = 'adam', and loss = 'categorical_crossentropy' combination set is contemplated as optimum. The other four datasets also perform accordingly better in such arrangements. As for DBN model, the parameters combination ('n_epochs_rbm', 'n_iter_back-prop') is tuned in a variation of (10, 100), (10, 50), (5, 100), and (5, 50) for NSL-KDD dataset. As per the performance analysis, the tuning with (10, 100) combination yields a higher testing accuracy of 91.97% with an evaluation time of 10,514 s. The same parameter tuning is applicable for the other four datasets too. As LSTM is a modified DNN, it is also constructed as a sequential model with the above parameters. While three variations of the LSTM model are explored in this study, the various combinations of hidden neuron set and no. of epoch have been examined for each of the models. The LSTM model is tested with two tuned parameters [('hidden neuron'), 'epoch'] for the IoTID20 dataset, and the combination sets [(4), 10], [(4), 15], [(4), 5], and [(2), 10] obtain 99.91%, 99.25%, 94.35%, and 95.33% test accuracy with time 1128 s, 1807 s, 650 s, and 1141 s, respectively. These statistics indicate that the first combination [(4), 10] with a dropout of 0.6 beats the others, so the rest of the datasets are analyzed with it and performed as anticipated. Now, Stacked LSTM model is trialed with neuron-epoch combination [(8), 10], [(16), 10], [(16), 5], and [(16), 15] for IoTDevNet dataset, where test accuracy is found 82.94%, 99.63%, 98.28%, and 99.53% with time variance of 1535 s, 1641 s, 887 s, and 2549 s. So, the performance of [(16), 10] combination with a dropout of 0.4 provides an optimum result which is tested for other datasets as well. Finally, for the NSL-KDD dataset, the Bi-LSTM model with the tuned parameter sets [(80, 40, 128), 15], [(80, 40, 128), 5], [(80, 40, 128), 10], and [(40, 20, 64), 15] achieves 99.5%, 98.78%, 99.06%, and 99.26% test accuracy with processing times of 55 s, 32 s, 47 s, and 63 s.

The results reveal the correctness of [(80, 40, 128), 15] set with a dropout of 0.1 to get a most favorable outcome. Furthermore, other datasets also justify these sets of tuned parameters through the performance.

4.3 Result Analysis

To analyze the performances of the executed models, the widely used popular multi-class performance metric 'Accuracy' is evaluated in this experiment. Besides, the scores of Precision, Recall, and F1-score have also been computed. These metrics depend on four basic qualitative model quality indicators, namely, true positive, true negative, false positive, and false negative [50]. A brief analogy of the resulting training and validation accuracy of these techniques corresponding to each dataset is specified below.

Figs. 5 and 6 depict the visualization of training and validation accuracy, precision, recall, and f1-score of each shallow ML and DL model for five corresponding datasets. From these figures, it is specified that among shallow ML models (DT, RF, and SVM), SVM achieves the highest train and test accuracy, as, (99.81%, 99.81%), (99.82%, 99.84%), and (99.42%, 99.44%) corresponding to NSL-KDD, IoTDevNet, and DS2OS, respectively. For the other two datasets, IoTID20 and IoT Botnet, DT obtains the highest train and test accuracy, as (100%, 100%) and (99.71%, 99.71%), respectively. Bi-LSTM outperforms the other four DL models (DNN, DBN, LSTM, and Stacked LSTM) in terms of train and test accuracy as (99.27%, 99.27%), (99.97%, 99.97%), (99.43%, 99.39%), (99.996%, 99.99%), and (99.998%, 99.991%) corresponding to NSL-KDD, IoTDevNet, DS2OS, IoTID20, and IoT Botnet dataset, respectively.

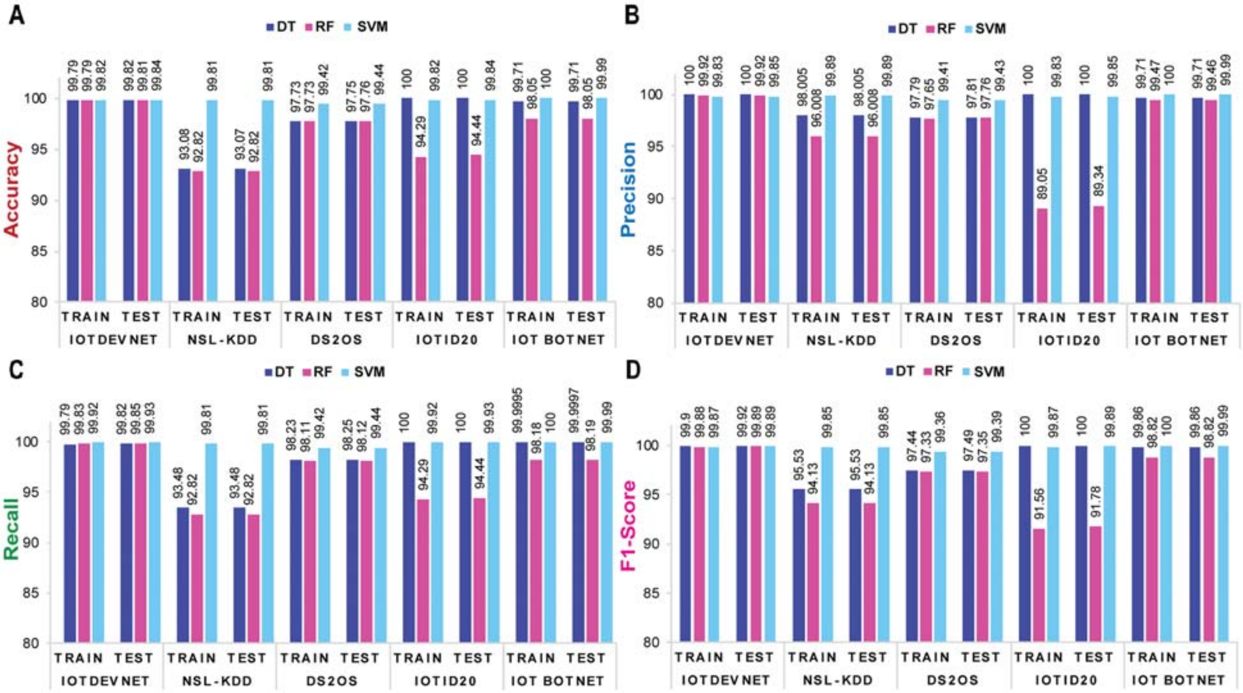

Figure 5: Training and validation matrices of ML models for five discussed datasets-(a) accuracy (b) precision (c) recall and (d) f1-score where all depict that SVM and DT perform slightly better than RF in attack detection

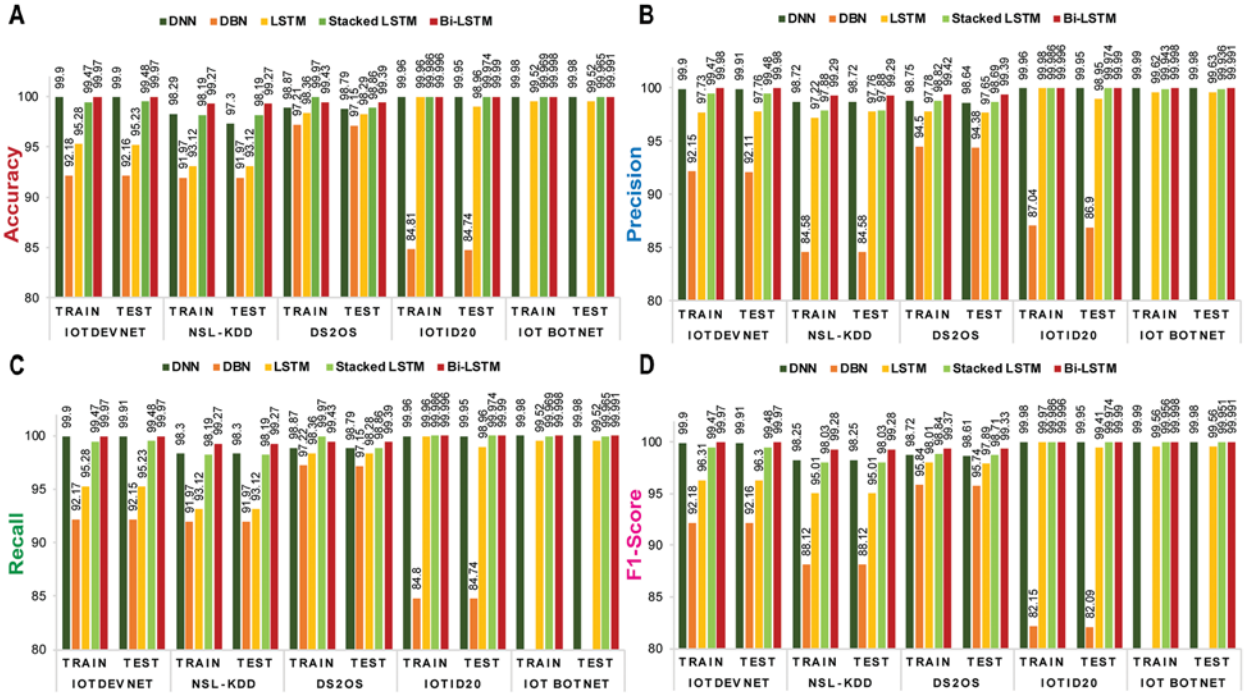

Figure 6: Training and validation matrices of ML models for five discussed datasets-(a) accuracy (b) precision (c) recall and (d) f1-score which showcase Bi-LSTM to be most efficient and accurate in attack detection

4.4 *Comparative Study*

Fig. 7 can intuitively compare the evaluation index of all presented models to in advance IDS models mentioned in the state-of-the-art based on DS2OS and NSL-KDD dataset, respectively. In these figures, each set of colors exhibits a comparison set of different learning models from some notable studies.

The use of DS2OS dataset is quite recent in the study of IDSs. Therefore, the number of directed researches related to network security are very few with this dataset. Among them, [29] is worth mentioning. In this study, numerous ML techniques are considered while SVM acquired 98.2% accuracy which is lesser than our trained SVM model of 99.44% accuracy. Cheng et al. considered SVM, LSTM, and TCN in [30] with an accuracy of 86.98%, 93.84%, and 98.15%, respectively while our presented similar architecture of LSTM and Bi-LSTM obtain 93.84% and 99.39% accuracy. However, a DNN model with higher accuracy, 98.29%, is presented by Reddy et al. in [33] whereas our considered DNN can detect anomalies 98.79% accurately.

For NSL-KDD dataset, Elmasry et al. acquired 73.35% accuracy using decision forest and 73.38 % using decision jungle in [25] while our simpler DT and RF can be more accurate with 93.07% and 92.82% accuracy individually. Similarly, this study also considers DNN, DBN, and LSTM-RNN with the accuracy of 86.53%, 93.78%, and 91.16%. Furthermore, Almiani et al. presented a cascaded RNN based approach with an accuracy of 92.18% in [18] which is comparable with the architecture of the proposed stacked LSTM with an accuracy of 98.19%. In study [27], accuracy 74.71%, 81.75%, 83.58%, and 79.43% are acquired using RF, CNN, CNN-BiLSTM, and Bi-LSTM chronologically while the presented RF, LSTM, Bi-LSTM outperforms with the accuracy of 92.82%, 93.12%, and 99.27% accordingly. Dushimimana et al. obtained

99.04% accuracy in their proposed Bi-RNN based IDS [28], which is very much comparable to our Bi-LSTM.

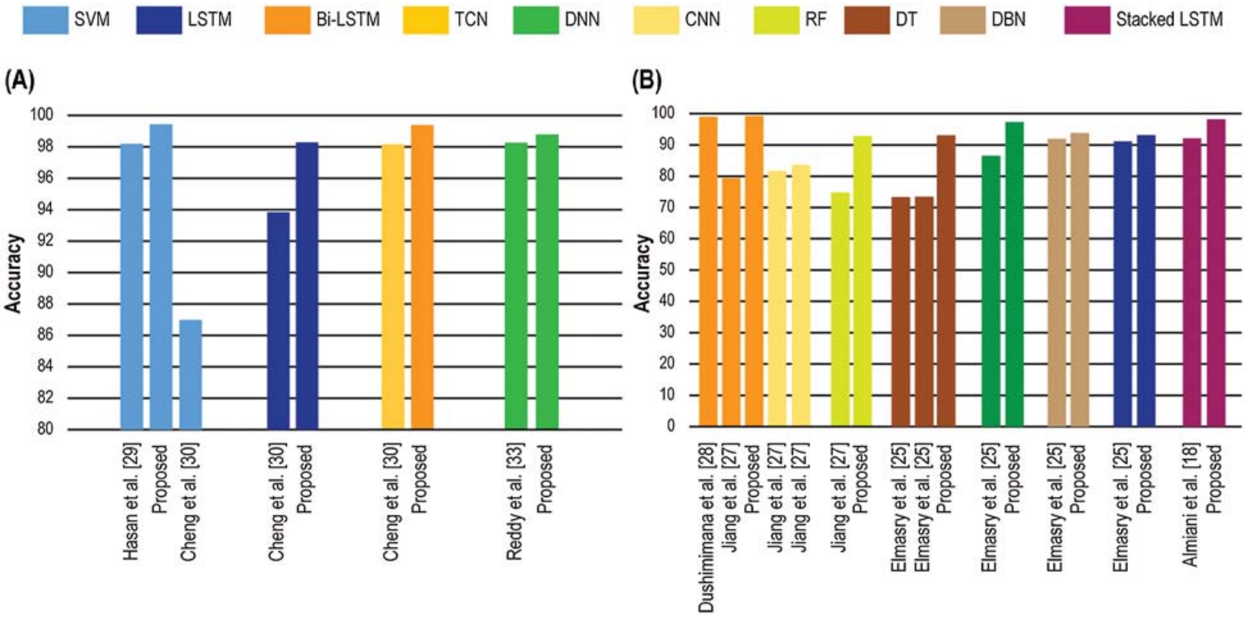

Figure 7: Accuracy comparison among the researches mentioned in the state-of-the-art and presented systems: (a) DS2OS (b) NSL-KDD datasets

5 Conclusion

IoT has been utilized extensively due to its potential of communicating with the actual devices of different application spaces to clients through the Web. In any case, the interconnected structure of IoT and the capacity of devices to interact with one another has risen security issues in IoT networks. So, a legitimate security system for IoT networks and devices should be created. In this paper, we have presented a data analysis technique for intrusion detection in the IoT environment. We begin with the state-of-the-art of different intrusion detection systems with a general introduction to IoT possible threats. Thereafter, the paper exhibits the nut and bolts of five datasets, among them two are known-NSL-KDD and DS2OS while another three are comparatively new-IoTDevNet, IoTID20, and IoT Botnet. Henceforth, this study discusses three ML and five DL techniques for distinguishing IoT attacks from a known or even an obscure environment. The structure overcomes implementation problems of heavy DL techniques directly on low space IoT devices, recognizes a few threats with high accuracy and detection rates, and maintaining a detection system by updating it accordingly for better attack identification. Relying on the experimental investigation, it can be concluded that Bi-LSTM outperforms best among the studied DL techniques and for this particular study of multiple datasets. However, it does not guarantee that on account of the big data and other obscure conditions Bi-LSTM will play out thusly. Hence, further investigation is required on the problem based on real-time data and power-time optimization. In the IoT network, micro-services behave distinctively at different events which trigger deviations in ordinary conduct in IoT services thus subsequently making an inconsistency.

So, further analysis is required to interpret these issues in a more inside and out manner which may end in designing a hybrid algorithm of multiple techniques.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] H. Lin, J. Hu, W. Xiaoding, M. F. Alhamid and M. J. Piran, "Towards secure data fusion in industrial IoT using transfer learning," *IEEE Transactions on Industrial Informatics,* pp. 1–9, 2020, (Early Access).

[2] A. Rehman, S. U. Rehman, M. Khan, M. Alazab and G. T. R., "CANintelliIDS: Detecting in-vehicle intrusion attacks on a controller area network using CNN and attention-based GRU," *IEEE Transactions on Network Science and Engineering,* pp. 1–11, 2021, (Early Access).

[3] H. F. Atlam and G. B. Wills, "IoT security, privacy, safety and ethics," *In Digital Twin Technologies and Smart Cities,* 1st ed., Switzerland, Springer Nature Switzerland AG 2020, chp. 8, pp. 123–149, 2020.

[4] S. Hameed, F. Idris Khan and B. Hameed, "Understanding security requirements and challenges in internet of things (IoT): A review," *Journal of Computer Networks and Communications,* vol. 2019, pp. 1–14, 2019.

[5] S. Anwar, J. Mohamad Zain, M. F. Zolkipli, Z. Inayat, S. Khan *et al.*, "From intrusion detection to an intrusion response system: Fundamentals, requirements, and future directions," *Algorithms,* vol. 10, no. 2, pp. 1–24, 2017.

[6] M. Letafati, A. Kuhestani, K. K. Wong and M. J. Piran, "A lightweight secure and resilient transmission scheme for the internet of things in the presence of a hostile jammer," *IEEE Internet of Things Journal,* vol. 8, no. 6, pp. 4373–4388, 2021.

[7] S. Suganth and D. Usha, "A survey of intrusion detection system in IoT devices.," *International Journal of Advanced Research,* vol. 6, pp. 23–30, 2018.

[8] G. Srivastava, G. Thippa Reddy, N. Deepa, B. Prabadevi and P. K. Reddy M, "An ensemble model for intrusion detection in the internet of softwarized things," in *Adjunct Proc. of the 2021 Int. Conf. on Distributed Computing and Networking*, New York, NY, USA, pp. 25–30, 2021.

[9] A. Khraisat, I. Gondal, P. Vamplew and J. Kamruzzaman, "Survey of intrusion detection systems: Techniques, datasets and challenges," *Cybersecurity,* vol. 2, no. 1, pp. 1–22, 2019.

[10] R. M. Swarna Priya, P. K. R. Maddikunta, M. Parimala, S. Koppu, T. R. Gadekallu *et al.*, "An effective feature engineering for DNN using hybrid PCA-gWO for intrusion detection in IoMT architecture," *Computer Communications,* vol. 160, pp. 139–149, 2020.

[11] H. Liu and B. Lang, "Machine learning and deep learning methods for intrusion detection systems: A survey," *Applied Sciences,* vol. 9, no. 20, pp. 4396–4424, 2019.

[12] M. Mamdouh, M. A. I. Elrukhsi and A. Khattab, "Securing the internet of things and wireless sensor networks via machine learning: A survey," in *2018 Int. Conf. on Computer and Applications*, Beirut, Lebanon, pp. 215–218, 2018.

[13] F. Farhin, I. Sultana, N. Islam, M. S. Kaiser, M. S. Rahman *et al.*, "Attack detection in internet of things using software defined network and fuzzy neural network," in *2020*, *Joint 9th Int. Conf. on Informatics, Electronics Vision and 2020 4th Int. Conf. on Imaging, Vision Pattern Recognition*, Kitakyushu, Japan, pp. 1–6, 2020.

[14] B. Roy and H. Cheung, "A deep learning approach for intrusion detection in internet of things using bi-directional long short-term memory recurrent neural network," in *2018, 28th Int. Telecommunication Networks and Applications Conf.*, Sydney, NSW, Australia, pp. 1–6, 2018.

[15] H.-V. Le, Q.-D. Ngo and V.-H. Le, "Iot botnet detection using system call graphs and one-class CNN classification," *Int. J. Innov. Technol. Exploring Eng,* vol. 8, no. 10, pp. 937–942, 2019.

[16] Q. Miao, J. Liu, Y. Cao and J. Song, "Malware detection using bilayer behavior abstraction and improved one-class support vector machines," *International Journal of Information Security,* vol. 15*,* no. 4*,* pp. 361–379, 2016.

[17] E. Burnaev and D. Smolyakov, "One-class SVM with privileged information and its application to malware detection," in *2016, IEEE 16th Int. Conf. on Data Mining Workshops*, Barcelona, Spain, pp. 273–280, 2016.

[18] M. Almiani, A. AbuGhazleh, A. Al-Rahayfeh, S. Atiewi and A. Razaque, "Deep recurrent neural network for IoT intrusion detection system," *Simulation Modelling Practice and Theory,* vol. 101*,* pp. 102031–102056, 2020.

[19] H. H. Pajouh, R. Javidan, R. Khayami, A. Dehghantanha and K. R. Choo, "A two-layer dimension reduction and two-tier classification model for anomaly-based intrusion detection in IoT backbone networks," *IEEE Transactions on Emerging Topics in Computing,* vol. 7*,* no. 2*,* pp. 314–323, 2019.

[20] P. Illy, G. Kaddoum, C. M. Moreira, K. Kaur and S. Garg, "Securing fog-to-things environment using intrusion detection system based on ensemble learning," in *2019 IEEE Wireless Communications and Networking Conf.*, Marrakesh, Morocco, pp. 1–7, 2019.

[21] V. V. Kumari and P. R. K. Varma, "A semi-supervised intrusion detection system using active learning SVM and fuzzy c-means clustering," in *2017 Int. Conf. on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, Palladam, pp. 481–485, 2017.

[22] B. Mohammadi and M. Sabokrou, "End-to-end adversarial learning for intrusion detection in computer networks," in *2019 IEEE 44th Conf. on Local Computer Networks*, Osnabrueck, pp. 270–273, 2019.

[23] C. Xu, J. Shen, X. Du and F. Zhang, "An intrusion detection system using a deep neural network with gated recurrent units," *IEEE Access,* vol. 6*,* pp. 48697–48707, 2018.

[24] Z. Li, P. Batta and L. Trajkovic, "Comparison of machine learning algorithms for detection of network intrusions," in *2018 IEEE Int. Conf. on Systems, Man, and Cybernetics*, Miyazaki, Japan, pp. 4248–4253, 2018.

[25] W. Elmasry, A. Akbulut and A. H. Zaim, "Empirical study on multiclass classification-based network intrusion detection," *Computational Intelligence,* vol. 35*,* no. 4*,* pp. 919–954, 2019.

[26] B. Ayyaz-ul-Haq Qureshi, H. Larijani, J. Ahmad and N. Mtetwa, "A heuristic intrusion detection system for internet-of-things (IoT)," in *Intelligent Computing: Proc. of the 2019 Computing Conf.*, London, United Kingdom, pp. 86–98, 2019.

[27] K. Jiang, W. Wang, A. Wang and H. Wu, "Network intrusion detection combined hybrid sampling with deep hierarchical network," *IEEE Access,* vol. 8*,* pp. 32464–32476, 2020.

[28] A. Dushimimana, T. Tao, R. Kindong and A. Nishyirimbere, "Bi-directional recurrent neural network for intrusion detection system (IDS) in the internet of things (IoT)," *International Journal of Advanced Engineering Research and Science,* vol. 7*,* no. 3*,* pp. 524–539, 2020.

[29] M. Hasan, M. M. Islam, M. I. I. Zarif and M. M. A. Hashem, "Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches," *Internet of Things,* vol. 7*,* pp. 100059–100073, 2019.

[30] Y. Cheng, Y. Xu, H. Zhong and Y. Liu, "Leveraging semisupervised hierarchical stacking temporal convolutional network for anomaly detection in IoT communication," *IEEE Internet of Things Journal,* vol. 8*,* no. 1*,* pp. 144–155, 2020.

[31] N. K. Sahu and I. Mukherjee, "Machine learning based anomaly detection for IoT network: (anomaly detection in IoT network)," in *2020 4th Int. Conf. on Trends in Electronics and Informatics (48184)*, Tirunelveli, India, pp. 787–794, 2020.

[32] S. Latif, Z. Zou, Z. Idrees and J. Ahmad, "A novel attack detection scheme for the industrial internet of things using a lightweight random neural network," *IEEE Access,* vol. 8*,* pp. 89337–89350, 2020.

[33] D. K. Reddy, H. S. Behera, J. Nayak, P. Vijayakumar, B. Naik *et al.*, "Deep neural network based anomaly detection in internet of things network traffic tracking for the applications of future smart cities," *Transactions on Emerging Telecommunications Technologies,* pp. 1–26, 2020.

[34] F. Farhin, M. S. Kaiser and M. Mahmud, “Secured smart healthcare system: blockchain and Bayesian inference based approach,” in *Proc. of Int. Conf. on Trends in Computational and Cognitive Engineering*, Singapore, pp. 455–465, 2021.

[35] N. Islam, I. Sultana and M. S. Rahman, “HKMS-Ami: A hybrid key management scheme for AMI secure communication,” in *Proc. of Int. Conf. on Trends in Computational and Cognitive Engineering*, Singapore, pp. 383–392, 2021.

[36] L. Dhanabal and S. P. Shantharajah, “A study on NSL-kDD dataset for intrusion detection system based on classification algorithms,” *International Journal of Advanced Research in Computer and Communication Engineering,* vol. 4*,* no. 6*,* pp. 446–452, 2015.

[37] M. Tavallaee, E. Bagheri, W. Lu and A. A. Ghorbani, “A detailed analysis of the KDD CUP 99 data set,” in *2009,I EEE Symposium on Computational Intelligence for Security and Defense Applications*, Ottawa, ON, Canada, pp. 1–6, 2009.

[38] Kaggle, “Iot device network logs,” 2020, [Online]. Available: https://www.kaggle.com/speedwall10/iot-device-network-logs.

[39] M. Pahl and F. Aubet, “All eyes on you: Distributed multi-dimensional IoT microservice anomaly detection,” in *2018 14th Int. Conf. on Network and Service Management*, Rome, Italy, pp. 72–80, 2018.

[40] I. Ullah and Q. H. Mahmoud, “A scheme for generating a dataset for anomalous activity detection in IoT networks,” in *Canadian Conference on Artificial Intelligence*, Ottawa, ON, Canada, pp. 508–520, 2020.

[41] H. Kang, D. H. Ahn, G. M. Lee, J. D. Yoo, K. H. Park *et al.*, “Iot network intrusion dataset,” *IEEE Dataport,* 2019. https://dx.doi.org/10.21227/q70p-q449.

[42] I. Ullah and Q. H. Mahmoud, “A technique for generating a botnet dataset for anomalous activity detection in IoT networks,” in *2020 IEEE Int. Conf. on Systems, Man, and Cybernetics*, Toronto, ON, Canada, pp. 134–140, 2020.

[43] V. N. G. Raju, K. P. Lakshmi, V. M. Jain, A. Kalidindi and V. Padma, “Study the influence of normalization/transformation process on the accuracy of supervised classification,” in *2020 Third International Conference on Smart Systems and Inventive Technology*, Tirunelveli, India, pp. 729–735, 2020.

[44] S. M. Tahsien, H. Karimipour and P. Spachos, “Machine learning based solutions for security of internet of things (IoT): A survey,” *Journal of Network and Computer Applications,* vol. 161*,* pp. 102630–102648, 2020.

[45] A. L. Buczak and E. Guven, “A survey of data mining and machine learning methods for cyber security intrusion detection,” *IEEE Communications Surveys & Tutorials,* vol. 18*,* no. 2*,* pp. 1153–1176, 2015.

[46] G. Zhao, C. Zhang and L. Zheng, “Intrusion detection using deep belief network and probabilistic neural network,” in *2017,I EEE Int. Conf. on Computational Science and Engineering and IEEE Int. Conf. on Embedded and Ubiquitous Computing*, Guangzhou, pp. 639–642, 2017.

[47] N. Balakrishnan, A. Rajendran, D. Pelusi and V. Ponnusamy, “Deep belief network enhanced intrusion detection system to prevent security breach in the internet of things,” *Internet of Things,* pp. 1–8, 2019, (In Press).

[48] C.-C. Wei, “Development of stacked long short-term memory neural networks with numerical solutions for wind velocity predictions,” *Advances in Meteorology,* vol. 2020*,* pp. 1–18, 2020.

[49] A. Samy, H. Yu and H. Zhang, “Fog-based attack detection framework for internet of things using deep learning,” *IEEE Access,* vol. 8*,* pp. 74571–74585, 2020.

[50] M. Grandini, E. Bagli and G. Visani, “Metrics for multi-class classification: An overview,” *arXiv preprint*, vol. arXiv: 2008.05756, pp. 1–17, 2020.