SCHOLARLY
RESEARCH AND REVIEWS

International Journal of Scholarly Research in
Engineering and Technology

Journal homepage: https://srrjournals.com/ijsret/

ISSN: 2961-3280 (Online)

Int. J. Scholarly Res. Eng. Technol.
ISSN: 2961-3280 (Online)
International Journal of Scholarly Research in
Engineering and Technology
SRR Publications

(RESEARCH ARTICLE)

A novel intrusion detection system based on deep learning and random forest for digital twin on IOT platform

Swati Lipsa [*] and Ranjan Kumar Dash

Odisha University of Technology and Research, Bhubaneswar, 751029, Odisha, India.

International Journal of Scholarly Research in Engineering and Technology, 2023, 02(01), 051–064

Publication history: Received on 02 February 2023; revised on 12 March 2023; accepted on 15 March 2023

Article DOI: https://doi.org/10.56781/ijsret.2023.2.1.0020

Abstract

Digital Twin (DT) has bright prospects for a broad spectrum of applications, such as supply chain, healthcare, predictive maintenance, etc. On one hand, the intrinsic properties of DT make it highly relevant to real-world applications, while on the other hand, they render it vulnerable to cyber threats. So, the security of DT is of utmost importance for the security of the communication and computational infrastructure underneath it. As cybercriminals are always coming up with new attack techniques, it is essential to safeguard the digital twin against malicious attacks. This can be accomplished by putting in place an efficient security mechanism, one of which is an intrusion detection system (IDS). The work done in this paper introduces a novel hybrid model for IDS in digital twins that integrates deep learning (DL) and random forest (RF). The effectiveness of the developed model is evaluated in comparison to that of k-Nearest Neighbors (KNN), Naive Bayes, Logistic Regression (LR), Support Vector Machine (SVM), Random Forest (RF), and Long Short Term Memory (LSTM), and it's evident from the outcome that the proposed method has higher accuracy than the competing models by at least 5%. In addition to its high accuracy, our proposed model also promises impressively low computing time requirements when compared to other competing models.

Keywords: Digital Twin; Deep Learning; Random Forest; Intrusion Detection System

1 Introduction

The digital technologies are rapidly expanding their domain-specific applications in different industries around the globe. These technologies include cloud computing, Internet-of-Things (IoT), edge computing, fog computing, and digital twins, to name a few [1]. Among them, the digital twin serves as an interface between these technologies by virtualizing the physical objects sensed by IoT devices and facilitating their digital copies for people at the top of industries' hierarchical working structures. Thus, by remodeling or reassembling existing models or creating new ones using digital tools, the risk, and expense are substantially reduced.

The success of the digital twin relies largely on its skeletal working networks i.e. IoT [2, 3]. The Internet-of-things, which is an interconnection of sensor nodes, has mostly three working layers: the physical layer, the network layer, and the application layer [4]. The implementation of the digital twin is possible due to the IoT or, in other words, the IoT provides a working platform for digital twins [2,3]. The adoption of digital twins by different industries like manufacturing, healthcare, smart cities, supply chain, etc. is growing at a rapid pace.

The digital twin facilitates much-sophisticated flexibility to its end users as long as there is a one-to-one mapping between the physical objects and their replicas. However, any tiny discriminated replica may cause substantial harm to the operationality of the whole system. Under such a scenario, it can even be disruptive to the industries that have adopted it. Out of many reasons, security threats are the main reason for such discrimination between physical objects and their replicas. Like other networks, the IoT is also vulnerable to many security threats [5]. Moreover, as IoT is the

[*] Corresponding author: Swati Lipsa

backbone of the digital twin, security attacks on IoT would also affect the working of the digital twin [6]. There are numerous approaches for mitigating security threats from general networks to very specific networks such as IoT, with the intrusion detection system(IDS) being the most promising [7].

The intrusion detection system provides a defense mechanism to mitigate many attacks. A detail of IDS along with its classification can be found in [7]. Further, the work [8] suggested the use of machine learning as a part of IDS to detect ~~the~~ attacks. The machine learning algorithms employed for IDS are supervised, i.e., they are trained and validated on a dataset that contains the attributes or features that define the type of attack. The accuracy of such an algorithm largely depends on the type and size of the relevant dataset in which the target function can be expressed in terms of features [9, 10]. So, one of the intrinsic steps towards the design of an IDS is to select the most promising dataset that maps the problem statement properly or to use real-time data.

The work carried out in this paper is an attempt to propose a hybrid model of deep learning and random forest for the design of IDS to mitigate different types of network-specific attacks. Deep learning is employed for feature selection or dimensionality reduction, whereas the random forest algorithm is used for multi-class classification.

The rest of the paper is organized as follows: Related work pertaining to the use of machine learning for the design of IDS is discussed in Section 2; the proposed model, i.e., DL-RF based IDS, is presented in Section 3; Section 4 is devoted to the analysis of the dataset and its suitability towards a real-time working environment for the digital twin; the simulated results, along with their discussion, are demonstrated in Section 5; Section 6 concludes the paper with its future scope.

1.1 Related work

Machine learning [8] is a competitive approach toward the detection of security threats like denial of service (DoS), distributed denial of service (DDoS), data theft, etc. Hodo et al. [11] proposed an artificial neural network (ANN) based IDS to combat attacks like DoS and DDoS with a reported accuracy of 99.4%. They trained the model with 2313 records, validated it, and tested it with 496 instances for each case. Zhang et al. [12] used a hybrid model of an improved genetic algorithm and a deep belief network to propose an IDS for the IoT. They considered the NSL-KDD dataset to train and validate their method to mitigate attacks like DoS, probe, user to root (U2R), and remote to local (R2L) with an overall accuracy of 99%.

Ge et al. [13] applied deep learning to the BoT-IoT dataset to detect attacks like DoS, DDoS, reconnaissance, and information theft. The accuracy of the model has been expressed in terms of attack types. The overall accuracy, as mentioned by them, is 99%. Smys et al. [14] used LSTM to design an IDS for the IoT. The dataset used by them to train and validate their model is UNSW NB15. They performed a binary classification with an accuracy of 98.6%. Almian et al. [15] proposed an IDS based on a recurrent neural network. They used the NSL-KDD dataset to classify the attacks like DoS, probe, U2R, and R2L with an average of 90% accuracy. Qiu et al. [16] introduced an IDS based on deep learning with an accuracy of 94.31%. The dataset considered by them is the Mirai dataset to mitigate attacks like DoS, DDoS, and ARP scans.

Bovenzi et al. [17] presented an IDS based on a multimodal deep autoencoder and used the BoT-IoT dataset to train and validate their model to mitigate attacks like DoS, DDoS, reconnaissance, and information theft. Akbarian et al. [18] used IDS in a digital twin environment to detect ramp attacks and scaling attacks with 99.4% accuracy. Kumar et al. [19] presented an IDS for an IoT environment by using a decision tree and its variant to detect four types of attacks: generic, DoS, probe, and exploit. They considered the UNSW-NB15 dataset to train and validate their model.

Otoum et al. [20] proposed deep learning-based IDS to detect R2L, U2R, DoS, and probe attacks. The dataset used by them is the NSL-KDD benchmark data set and the reported accuracy is 99.02%. Lo et al. [21] presented an E-GraphSAGE (Graph SAmple and aggreGatE) based IDS for the IoT. They reported the accuracy of their model to be 99.99%, validating it over BoT-IoT. The attack types they considered are DoS, DDoS, reconnaissance, and information theft. Su et al. [22] developed a hybrid model of LSTM and federated learning for IDS to detect DDoS attacks. The dataset used by them is the KDD CUP 1999 dataset. The works [23] introduced an IDS for mitigating timing attacks on the DT system which can be extended for reliable and intelligently deployed sensors [24, 25].

The works summarized in Table 1 make it quite evident that the BoT- IoT dataset can be used effectively as a real-time compatible dataset for the detection of network intrusions. Further, the employed technique for IDS must be both accurate and time-efficient because of the huge amount of continuous data received from different sensor nodes. A less time-efficient ID may slow down the working environment of the digital twin merely based on detecting the different attacks, which in turn hinders the basic characteristics of the digital twin, i.e., its capability of quickly digitizing the physical objects and their mapping as well. This necessitates proposing a time-efficient ID with greater accuracy by using a hybrid machine learning technique, i.e., the integration of deep learning and random forest. The obvious reason to choose deep learning for feature extraction is due to its inherent capability of extracting features during the learning process while the use of random forest lies in the fact that it is an efficient ensemble technique that outperforms other techniques in terms of multi-class classification accuracy.

Table 1 Summary of works related to IDS based on machine learning

Paper	Model	Dataset	Threats	Accuracy(%)
Hodo et al. [11]	ANN	Training-2313, Testing, and validation- 496	DoS, DDoS	99.4
Zhang et al. [12]	Improved GA and DBN	NSL-KDD	DoS, probe, U2R, R2L	99
Ge et al. [13]	DL	BoT-IoT	DoS, DDoS, reconnaissance, information theft	99
Smys et al. [14]	LSTM	UNSW NB15	normal vs attack	98.6
Almian et al. [15]	RNN	NSL-KDD dataset	DoS, probe, U2R, R2L	90
Qiu et al. [16]	DL	Mirai dataset	DoS, DDoS, ARP scan	94.31
Bovenzi et al. [17]	DL	BoT-IoT	DoS, DDoS, reconnaissance, information theft	99
Akbarian et al. [18]	IDS in DT	NA	ramp attack, scaling attack	
Kumar et al. [19]	Decision tree	UNSW-NB15	generic, DoS, probe, exploit	88.92
Otoum et al. [20]	Deep learning	NSL-KDD	R2L, U2R, DoS, probe attacks	99.02
Lo et al. [21]	E- GraphSAGE	BoT-IoT	DoS, DDoS, reconnaissance, informa tion theft	99.99
Su et al. [22]	LSTM and federated learning	KDD CUP 1999	DDoS	

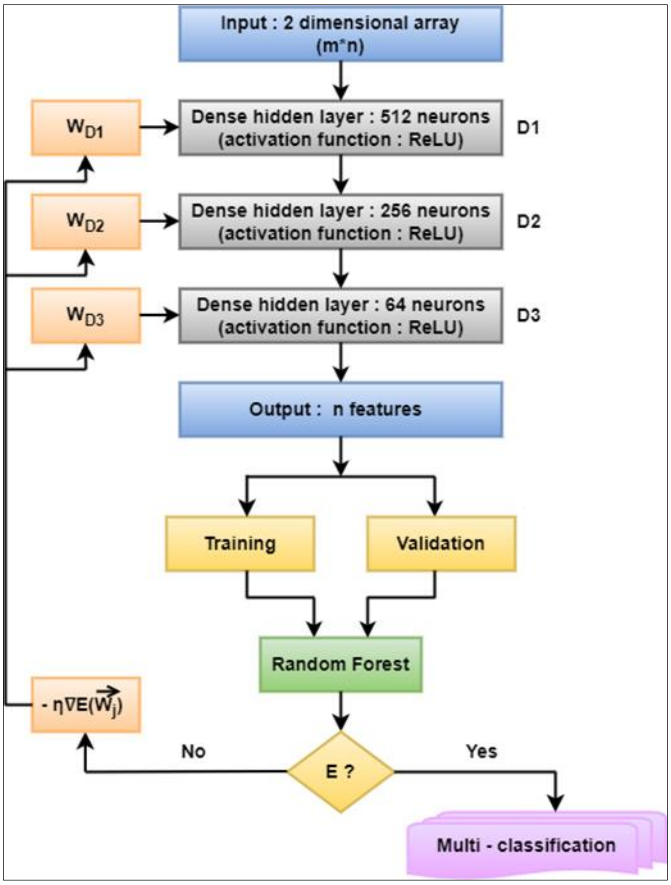
Input : 2 dimensional array (m*n)
W_D1
Dense hidden layer : 512 neurons (activation function : ReLU)
D1
W_D2
Dense hidden layer : 256 neurons (activation function : ReLU)
D2
W_D3
Dense hidden layer : 64 neurons (activation function : ReLU)
D3
Output : n features
Training
Validation
Random Forest
- η∇E(W_j)
No
E ?
Yes
Multi - classification

Figure 1 Proposed DL-RF model for IDS

2 Material and methods

2.1 Proposed DL-RF based IDS

The proposed model for IDS is a hybrid model of DL and random forest. The DL is used for extracting important features, while the random forest is used as an ensembler for classification purposes. The developed model is depicted in Figure 1. The feature extraction is accomplished by the DL with three ~~number of~~ dense layers. The first two layers have 512 and 256 neurons while the third layer has 64 neurons to store the extracted features. The activation function used in these layers is ReLU. The input to this DL is the m n array where m is the number of records while n is the number of features from a suitable dataset or a real-time environmental input.

The first layer of DL (D1) receives n features as input and outputs 512 features activated by the ReLU function. Similarly, the second dense layer of DL (D2) yields 256 features, while the third one (D3) provides 64 features. The 64 extracted features are then fed to the last dense layer, the output of which is the n features. The features are extracted across all cases, allowing us to statistically assess their importance in the final classification. Dimensionality reduction is carried out intrinsically in response to their statistical score. The proposed dimensional reduction technique does not rely on explicit instance transformation, as in principal component analysis or linear discrimination. Thus, it avoids the time-consuming transformation of instances to and from.

The detection of attacks and their types is a multi-classification problem. The random forest classifier is used to detect multiple attacks. It is trained over the extracted features from the DL while validated over validation or testing instances.

Let $< X_{i,1}, X_{i,2} \cdots X_{i,n}, t_i >$ be the ith tuple, where X is the feature, n is the number of features and t is target class with $t_i \in \{0,1,\cdots c\}$ and C is the number of classes. The weights associated with the above-mentioned layers are $\overrightarrow{W}_{D,3}, \overrightarrow{W}_{D,2}, \overrightarrow{W}_{D,1}$ respectively with each $\overrightarrow{W}_{D,i} = \{w_{D1}, w_{D2}, \cdots, w_{Dn}\}$ where i {=1, 2 or 3}. The sum square error (E) yielded by the random forest over the input data can be expressed in terms of the weights associated with three hidden layers as :

$$E = \left(\overrightarrow{W}_{D,3}, \overrightarrow{W}_{D,2}, \overrightarrow{W}_{D,1}\right) = \underset{\overrightarrow{W}_{D3},\overrightarrow{W}_{D2},\overrightarrow{W}_{D1}}{\operatorname{argmin}} \sum_{i=1}^{m}(t_i - \hat{t}_i)^2 \ldots\ldots\ldots(1)$$

Where,

$$\overrightarrow{W}_{D,3}, \overrightarrow{W}_{D,2}, \overrightarrow{W}_{D,1} \in \Re^{n+1} \ldots\ldots\ldots(2)$$

The sum square error (E) of this multi-class classification can be minimized by finding the optimal weights associated with each hidden layer. In order to minimize this error, the optimal weights can be searched by starting with random valued weights and then updating these weights by small values in the direction of the steepest descent along the error surface. This direction can be determined by the gradient of E with respect to the weight vectors. Thus, the gradient of E with respect to $\overrightarrow{W}_{D,i} = \{w_{D1}, w_{D2}, \cdots, w_{Dn}\}$, where i={1,2 or 3} can be expressed as:

$$\nabla E\left(\overrightarrow{W}_{D,3}\right) = \left[\frac{\delta E}{w_{3,1}}, \frac{\delta E}{w_{3,1}} \cdots \frac{\delta E}{w_{3,n}}\right] \ldots\ldots\ldots\ldots(3)$$

$$\nabla E\left(\overrightarrow{W}_{D,2}\right) = \left[\frac{\delta E}{w_{21}}, \frac{\delta E}{w_{2,2}} \cdots \frac{\delta E}{w_{321}}\right] \ldots\ldots\ldots\ldots (4)$$

$$\nabla E\left(\overrightarrow{W}_{D1}\right) = \left[\frac{\delta E}{w_{11}}, \frac{\delta E}{w_{12}} \cdots \frac{\delta E}{w_{1,n}}\right] \ldots\ldots\ldots\ldots (5)$$

The optimal values of weights for each hidden unit can be obtained by iterating through the input data. The gradient of errors is propagated along the hypothesis space mentioned in the steepest descent direction with the following weight update rule:

$$\overrightarrow{W}_j = \overrightarrow{W}_j - \eta \nabla E\left(\overrightarrow{W}_j\right), where\ j \epsilon \{\overrightarrow{W}_{D3}\ or\ \overrightarrow{W}_{D2}\ or\ \overrightarrow{W}_{D1}\} \ldots\ldots\ldots\ldots(6)$$

In order to avoid a large number of negative weights, the following rectified linear unit (ReLU) is used:

$$ReLu(x) = \begin{cases} x & for\ x > 0 \\ 0 & for\ x \leq 0 \end{cases} \ldots\ldots\ldots(7)$$

The derivative of ReLu function mentioned in Eq.(7) is

$$\frac{\delta(ReLu(x))}{\delta x} = \begin{cases} 1 & for\ x > 0 \\ 0 & for\ x \leq 0 \end{cases} \ldots\ldots\ldots(8)$$

The whole process which has been discussed above is depicted in Figure 1. The proposed model converges to optimal error which should be less than some pre-defined and acceptable sum of square error (E).

2.2 Dataset selection and its features

BoT-IoT dataset [26, 27, 28] contains the data generated in real-time scenarios which can be used to train and validate a model. The trained model is equally robust to detect the attacks if deployed in a real-time environment. The normal and benign data from the IoT test bed is distributed around 74 CSV files in the BoT-IoT dataset. 5% of the entire data split for training (2934817 number of records with 19 features) and testing (733705 number of records with 19 features) with the most important features are also available. The extracted features are presented in Table 1 along with their explanation.

2.3 Analysis of the BoT-IoT dataset

The IP address of attacking devices are enlisted in Table 3. The packets are mainly transmitted over protocols TCP and UDP (Table 4). From Table 3 and Table 4, it can be observed that both udp and tcp are more prone to different attacks as compared to other protocols like arp, icmp, and ipv6- icmp. However, this does not necessitate avoiding these protocols for the transmission of packets. The IP address mentioned in Table 3 can be used to trace back and block the devices if they are found to be sending malicious packets.

Table 2 Features of BoT-IoT dataset

Features	Is numeric	Std. Deviation	Distinct values
pkSeqID	Yes	1059057.75	-
proto	False	-	5
saddr	False	-	20
sport	False	-	65541
daddr	False	-	81
dport	False	-	6906
seq	Yes	75786.99	-
stddev	Yes	0.8036	-
N_IN_Conn_P_SrcIP	Yes	24.390	-
min	Yes	1.48355	-
State_number	Yes	1.187	-
mean	Yes	1.51	-
N_IN_Conn_P_DstIP	Yes	18.16	-
drate	Yes	56.23	-
srate	Yes	784.54	-
max	Yes	1.86	-
attack	Yes	0.011	-
category	False	-	5
subcategory	False	-	8

The dataset contains four different types of attacks, such as DDoS, DoS, reconnaissance, and theft, in addition to normal packets (Table 5). The table reveals the dataset to be highly imbalanced, which may lead the classifier to be overfitted. In order to balance the number of instances of normal packets, additional records are extracted from the earlier mentioned CSV files (Table 6). These records are refined to contain the features enlisted in Table 2.

Table 3 The details of the IP address used to generate normal or malicious packets

IP address	Number
192.168.100.147	761360
192.168.100.148	738642
192.168.100.150	712260
192.168.100.149	711466
192.168.100.3	6609
192.168.100.5	4107
192.168.100.6	272
192.168.100.7	34
192.168.100.4	17
192.168.100.1	14

192.168.100.27	9
192.168.100.46	8
fe80::250:56ff:febe:254	5
192.168.100.55	3
fe80::2c6a:ff9b:7e14:166a	2
fe80::250:56ff:febe:c038	2
fe80::c0c0:aa20:45b9:bdd9	2
fe80::250:56ff:febe:89ee	2
fe80::250:56ff:febe:26db	2
fe80::250:56ff:febe:e9d9	1

Table 4 Protocol type and their numbers

Protocol type	Number	Attack type	number
udp	1596819	2	826321
		1	758279
		3	11930
		0	289
tcp	1330598	1	782999
		2	493777
		3	53723
		4	62
		0	37
icmp	7228	3	7206
		1	12
		2	10
arp	166	3	60
		2	40
		0	38
		1	25
		4	3
ipv6-icmp	6	0	6

Table 5 Attack type and their number

Attack type	Numeric value	Number of instances	
		Training	Validation

DDoS	1	1541315	385309
DoS	2	1320148	330112
Reconnaissance	3	72919	18163
Normal	0	370	107
Theft	4	65	14

Table 6 Attack type and their number after the addition of extra instances

Attack type	**Numeric value**	**Number of instances**	
		Training	**Validation**
DDoS	1	1541315	385309
DoS	2	1320148	330112
Reconnaissance	3	72919	18163
Normal	0	2103370	601107
Theft	4	65	14

3 Results and discussion

The proposed DL-RF based IDS has been simulated in a Google Colab environment. The developed IDS is trained and validated over the training dataset and validation dataset respectively.

3.1 Performance of DL-RF model

Here are the different performance metrics that are considered to evaluate the performance of the proposed DL-RF models:

- Precision(P) - $P = \frac{TP}{TP+FP}$, where TP is the true positive and FP is the false positive
- Recall(R) - $R = \frac{TP}{TP+FN}$, where FN is the false negative
- F1-score - $F1 - score = \frac{2\times P\times R}{P+R}$
- Support - Support is the number of tuples belonging to certain classes
- Cohen's Kappa score (k) - Cohen's Kappa score is a statistical measure to find inter-rater for categorical items by integrating class imbalance and measurement to chance.
- Matthew's correlation coefficient(MCC) - $\frac{(TP\times TN-FP\times FN)}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$
- Log loss(LL) - Log loss is a statistical measure to obtain the model's un- certainty while predicting the target classes by considering their probability score.
- Receiver Operating Characteristic Curve(ROC)

A detailed description of precision, recall, F1-score, and ROC can be found in [10]. The performance of the proposed DL-RF model during training and validation is presented in Table 7. This table makes it evident that the proposed model is not affected by data overfitting as the log loss during training is less than that during validation. As a result, when compared to training, the loss and accuracy are both lower during validation.

The features extracted from DL require a little more processing to find the score of individual features as shown in Figure 2.

Table 7 Performance of the proposed DL-RF model during training and validation

	Attack	**P**	**R**	**F1-score**	**Support**	**CKS**	**MMC**	**LL**	**Acc**

	0	0.99	0.98	0.9	2103370				
	1	1.00	0.99	1.00	1541315				
Training	2	1.00	0.99	1.00	132014	0.99	0.99	0.001	0.999
	3	1	0.95	0.99	72919				
	4	1.00	0.95	0.98	165				
	0	0.98	0.97	0.98	601107				
	1	1.00	0.99	1.00	385309				
Validation	2	1.00	0.99	1.00	330112	0.99	0.99	0.004	0.99
	3	0.93	0.99	0.96	18163				
	4	1.00	0.93	0.96	14				

3.2 Comparison of DL-RF against other classifiers

The performance of the proposed model is compared against that of k- nearest neighbors (KNN), Naive Bayes, logistic regression (LR), support vector machine (SVM), random forest (RF), and long short-term memory (LSTM). The performance metrics considered here for comparison are precision, recall, F1-score, and accuracy. The attack type-specific values of these metrics for each model are depicted in Table 8. The precision of these classifiers is almost above 90% except for attack. However, the proposed DL-RF model has a value of 0.93 even for attacks when the others are found to be underperforming. The value of recall is almost similar for all the models, including our proposed model. Observations analogous to those made for precision can also be drawn for the F1-score. The accuracy of the proposed model surpasses at least 5% more than that of other models. ROC of different attack types is shown in Figure 3.

The confusion matrices for each model are shown in Figure 4(a), Figure 4(b), Figure 4(c), Figure 4(d), Figure 4(e), and Figure 4(f).

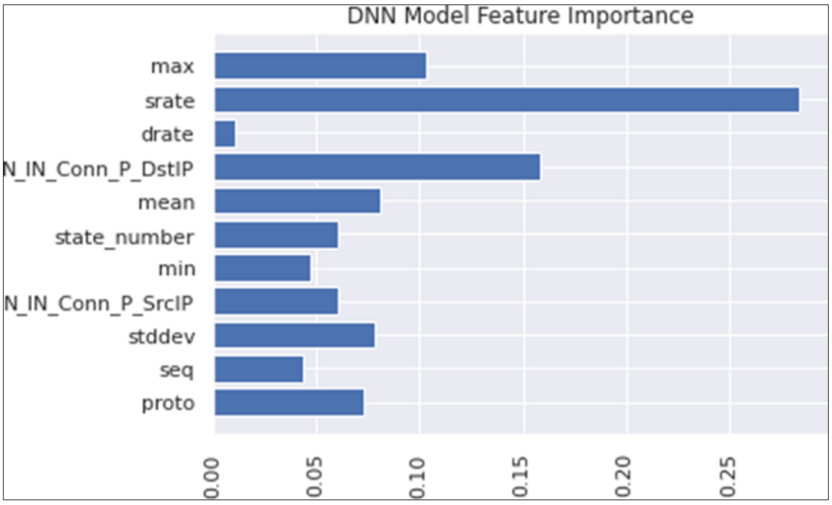

Figure 2 Extracted features with their score

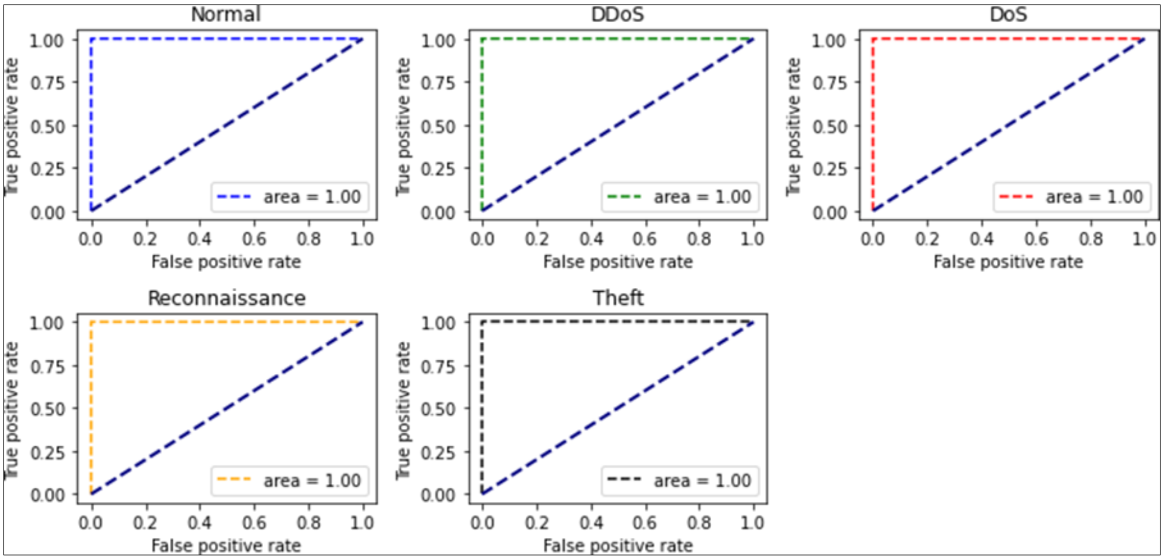

Figure 3 ROC of different attack types

Table 8 Performance comparison of the proposed DL-RF model against other machine learning classifiers

Model	**Attack type**	**P**	**R**	**F1-score**	**Accuracy**
	0	0.91	0.92	0.91	
	1	0.98	0.94	0.96	
kNN	2	0.97	0.94	0.96	0.94
	3	0.54	0.94	0.68	
	4	1.00	0.86	0.92	
	0	0.91	0.93	0.91	
	1	0.98	0.93	0.95	
Naive Bayes	2	0.97	0.93	0.95	0.93
	3	0.50	0.93	0.65	
	4	1.00	0.86	0.92	
	0	0.91	0.93	0.91	
	1	0.98	0.93	0.95	
LR	2	0.97	0.93	0.95	0.93
	3	0.50	0.93	0.65	
	4	1.00	0.86	0.92	
	0	0.92	0.96	0.92	
	1	0.99	0.96	0.97	
SVM	2	0.98	0.96	0.97	0.95
	3	0.64	0.96	0.77	
	4	1.00	0.93	0.96	
	0	0.91	0.94	0.92	
	1	0.99	0.96	0.97	

RF	2	0.98	0.96	0.97	0.95
	3	0.64	0.96	0.77	
	4	1.00	0.93	0.96	
	0	0.91	0.94	0.92	
	1	0.98	0.95	0.97	
LSTM	2	0.98	0.95	0.96	0.94
	3	0.59	0.95	0.72	
	4	1.00	0.93	0.96	
	0	0.98	0.97	0.98	
	1	1.00	0.99	1.00	
Proposed model	2	1.00	0.99	1.00	0.99
	3	0.93	0.99	0.96	
	4	1.00	0.93	0.96	

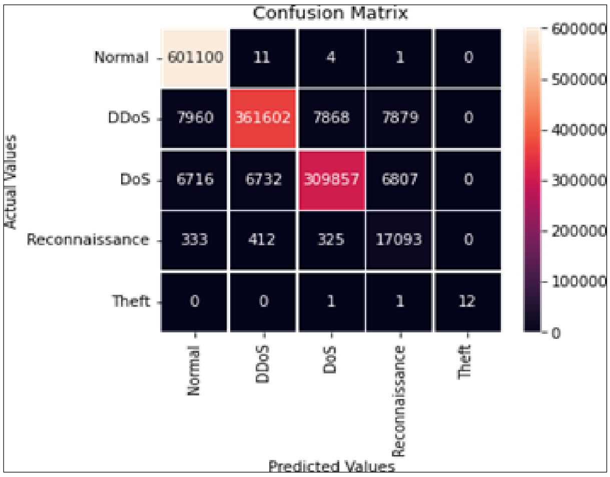

Figure 4(a) Confusion matrix of k-Nearest Neighbors

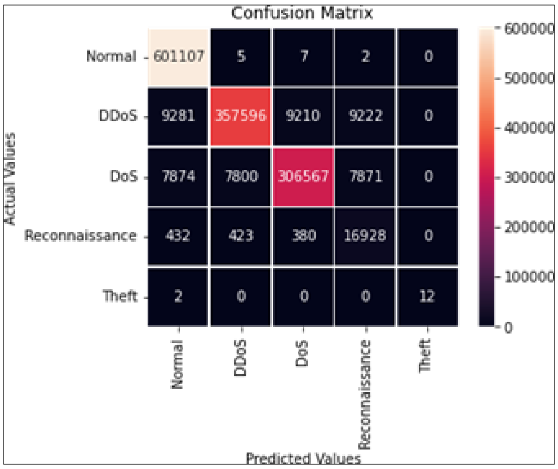

Figure 4(b) Confusion matrix of Logistic regression

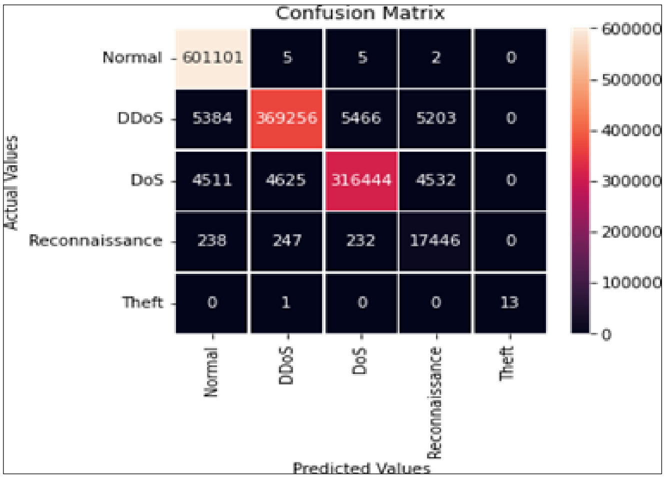

Figure 4(c) Confusion matrix of Random forest

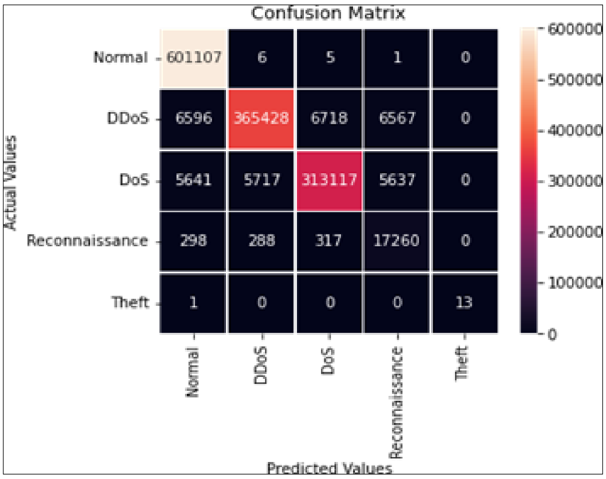

Figure 4(d) Confusion matrix of LSTM

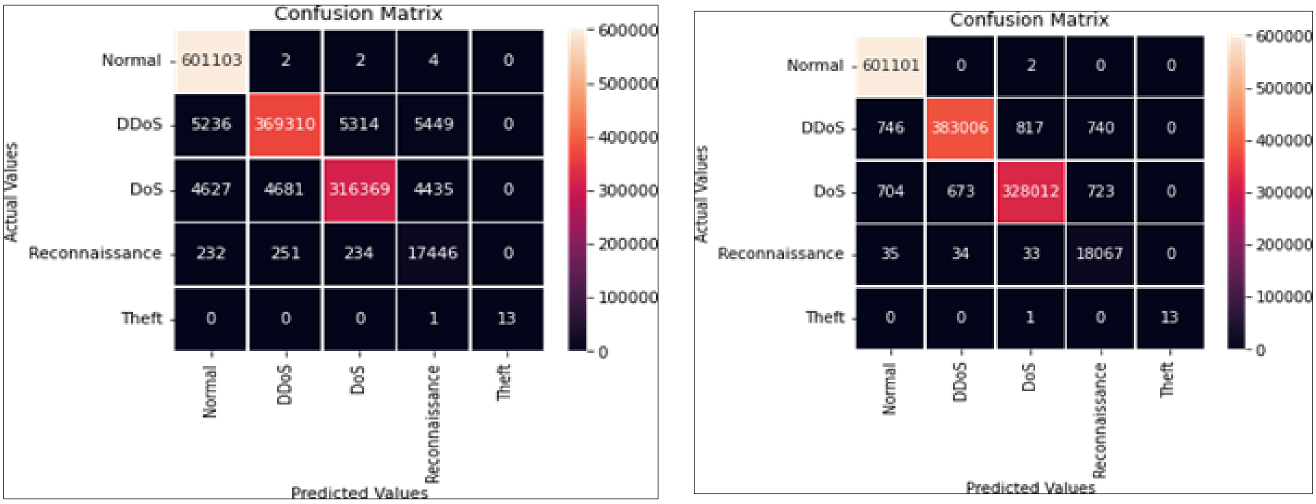

Figure 4(e) Confusion matrix of Support vector machine **Figure 4(f)** Confusion matrix of Proposed DL-RF

Figure 4 Confusion matrix generated by different classifiers

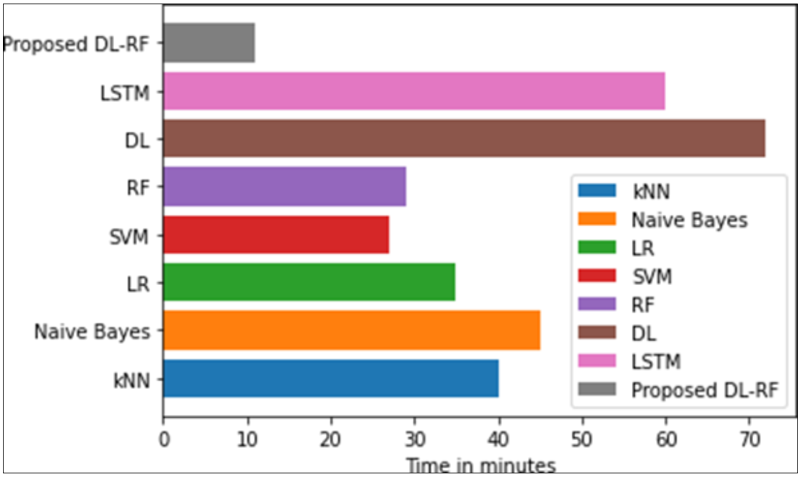

Figure 5 Comparison of computational time by proposed DL-RF based IDS against other methods

3.3 Comparison of computational time to train and validate the models

The existing works [13, 14, 17, 19, 21] on IDS that use the BoT-IoT dataset reported an accuracy of almost 99% by using the deep learning technique or its variant. In addition to the classifier's accuracy, a crucial factor to take into account, in this case, is the classifier's total training and validation time. The same architecture of deep learning mentioned in [13] has been simulated to train and validate the dataset discussed earlier. An accuracy of 99% was observed for 20 epochs to train and validate the deep learning model. The overall processing time was found to be 72 minutes and 17 seconds, compared to the proposed method's 11 minutes and 28 seconds, which included 7 minutes and 23 seconds for deep learning-based feature extraction and an additional 4 minutes and 5 seconds for classification. Thus, the time saved by the proposed DL-RF is found to be 64 minutes 49 seconds. The time consumed by the rest classifiers, as mentioned earlier, is also presented in Figure 5.

4 Conclusion

As digital twin is used across a multitude of applications, they generate massive amounts of network traffic data, making them susceptible to cyberattacks. In this work, we developed a random forest-based deep learning model for an intrusion detection system, which was then trained using the BoT-IoT dataset, that encompasses various types of cyberattacks. The proposed model is compared to that of k-Nearest Neighbors (KNN), Naive Bayes, Logistic Regression (LR), Support Vector Machine (SVM), Random Forest (RF), and Long Short Term Memory (LSTM) to analyze its performance. The results reveal that the proposed model is at least 5% more ac- curate than the other models being compared. Finally, our proposed model not only offers high accuracy but also drastically reduced computing time requirements in comparison to other methods.

Compliance with ethical standards

Acknowledgments

The authors acknowledge their gratefulness to the anonymous reviewers for their valuable comments to enhance the quality of this article.

Disclosure of conflict of interest

The authors have no conflict of interest to disclose.

References

[1] Jones D, Snider C, Nassehi A, Yon J, Hicks B. Characterising the Digital Twin: A systematic literature review. CIRP Journal of Manufacturing Science and Technology. 2020 May 1, 29:36-52.

[2] Baghalzadeh Shishehgarkhaneh M, Keivani A, Moehler RC, Jelodari N, Roshdi Laleh S. Internet of Things (IoT), Building Information Modeling (BIM), and Digital Twin (DT) in Construction Industry: A Review, Bibliometric, and Network Analysis. Buildings. 2022 Sep 22, 12(10):1503.

[3] Al-Ali AR, Gupta R, Zaman Batool T, Landolsi T, Aloul F, Al Nabulsi A. Digital twin conceptual model within the context of internet of things. Future Internet. 2020 Sep 26, 12(10):163.

[4] Han Y, Niyato D, Leung C, Kim DI, Zhu K, Feng S, Shen X, Miao C. A dynamic hierarchical framework for iot-assisted digital twin synchronization in the metaverse. IEEE Internet of Things Journal. 2022 Aug 23, 10(1):268-84.

[5] Deogirikar J, Vidhate A. Security attacks in IoT: A survey. In2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC) 2017 Feb 10 (pp. 32-37). IEEE.

[6] Alcaraz C, Lopez J. Digital twin: A comprehensive survey of security threats. IEEE Communications Surveys & Tutorials. 2022 Apr 29.

[7] Sherasiya T, Upadhyay H, Patel HB. A survey: Intrusion detection system for internet of things. International Journal of Computer Science and Engineering (IJCSE). 2016, 5(2):91-8.

[8] Lipsa, S., & Dash, R. K. (2023, January). A Novel Dimensionality Reduction Strategy Based on Linear Regression with a Fine-Pruned Decision Tree Classifier for Detecting DDoS Attacks in Cloud Computing Environments. In Artificial Intelligence: First International Symposium, ISAI 2022, Haldia, India, February 17-22, 2022, Revised Selected Papers (pp. 15-25). Cham: Springer Nature Switzerland.

[9] Lipsa S, Dash RK. MalNet–an Optimized CNN based method for Malaria Diagnosis. In2022 2nd International Conference on Intelligent Technologies (CONIT) 2022 Jun 24 (pp. 1-6). IEEE.

[10] Lipsa S, Dash RK. GASVR-A Model to Predict and Analyze Crude Oil Price. In2022 2nd Asian Conference on Innovation in Technology (ASIANCON) 2022 Aug 26 (pp. 1-5). IEEE.

[11] Hodo E, Bellekens X, Hamilton A, Dubouilh PL, Iorkyase E, Tachtatzis C, Atkinson R. Threat analysis of IoT networks using artificial neural network intrusion detection system. In2016 International Symposium on Networks, Computers and Communications (ISNCC) 2016 May 11 (pp. 1-6). IEEE.

[12] Zhang Y, Li P, Wang X. Intrusion detection for IoT based on improved genetic algorithm and deep belief network. IEEE Access. 2019 Mar 7, 7:31711-22.

[13] Ge, M., Fu, X., Syed, N., Baig, Z., Teo, G., & Robles-Kelly, A. Deep learning-based intrusion detection for IoT networks. In 2019 IEEE 24th pacific rim international symposium on dependable computing (PRDC) 2019 December (pp. 256-25609). IEEE.

[14] Smys S, Basar A, Wang H. Hybrid intrusion detection system for internet of things (IoT). Journal of ISMAC. 2020 Sep 30, 2(04):190-9.

[15] Almiani M, AbuGhazleh A, Al-Rahayfeh A, Atiewi S, Razaque A. Deep recurrent neural network for IoT intrusion detection system. Simulation Modelling Practice and Theory. 2020 May 1, 101:102031.

[16] Qiu H, Dong T, Zhang T, Lu J, Memmi G, Qiu M. Adversarial attacks against network intrusion detection in IoT systems. IEEE Internet of Things Journal. 2020 Dec 30, 8(13):10327-35.

[17] Bovenzi G, Aceto G, Ciuonzo D, Persico V, Pescapé A. A hierarchical hybrid intrusion detection approach in IoT scenarios. InGLOBECOM 2020-2020 IEEE Global Communications Conference 2020 Dec 7 (pp. 1-7). IEEE.

[18] Akbarian F, Fitzgerald E, Kihl M. Intrusion detection in digital twins for industrial control systems. In2020 International Conference on Software, Telecommunications and Computer Networks (SoftCOM) 2020 Sep 17 (pp. 1-6). IEEE.

[19] Kumar V, Das AK, Sinha D. UIDS: a unified intrusion detection system for IoT environment. Evolutionary intelligence. 2021 Mar, 14:47-59.

[20] Otoum Y, Liu D, Nayak A. DL-IDS: a deep learning–based intrusion detection framework for securing IoT. Transactions on Emerging Telecommunications Technologies. 2022 Mar, 33(3):e3803.

[21] Lo WW, Layeghy S, Sarhan M, Gallagher M, Portmann M. E-graphsage: A graph neural network based intrusion detection system for iot. InNOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium 2022 Apr 25 (pp. 1-9). IEEE.

[22] Su D, Qu Z. Detection DDoS of Attacks Based on Federated Learning with Digital Twin Network. InKnowledge Science, Engineering and Management: 15th International Conference, KSEM 2022, Singapore, August 6–8, 2022, Proceedings, Part III 2022 Jul 19 (pp. 153-164). Cham: Springer International Publishing.

[23] Mohamed T, Kezunovic M, Lusher J, Liu JC, Ren J. The use of digital twin for timing intrusion detection in synchrophasor systems. In2022 IEEE Power & Energy Society General Meeting (PESGM) 2022 Jul 17 (pp. 01-05). IEEE.

[24] Dash RK, Cengiz K, Alshehri YA, Alnazzawi N. A new and reliable intelligent model for deployment of sensor nodes for IoT applications. Computers and Electrical Engineering. 2022 Jul 1, 101:107959.

[25] Lipsa S, Nguyen TN, Dash RK. A New Signature-Based Blockchain Paradigm: Foreseeable Impact on Healthcare Applications. IEEE Internet of Things Magazine. 2022 Sep, 5(3):146-51.

[26] Moustafa N. The bot-iot dataset. IEEE Dataport. 2019 Oct, 5.

[27] Koroniotis N, Moustafa N, Sitnikova E, Turnbull B. Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset. Future Generation Computer Systems. 2019 Nov 1, 100:779-96.

[28] Anthi E, Williams L, Słowińska M, Theodorakopoulos G, Burnap P. A supervised intrusion detection system for smart home IoT devices. IEEE Internet of Things Journal. 2019 Jul 2, 6(5):9042-53.