Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 9304689, 13 pages
https://doi.org/10.1155/2022/9304689

Research Article

Intrusion Detection System on IoT with 5G Network Using Deep Learning

Neha Yadav,[1] Sagar Pande,[1] Aditya Khamparia,[2] and Deepak Gupta[3]

[1]*School of Computer Science, Lovely Professional University, Phagwara, Punjab, India*
[2]*Department of Computer Science, Babasaheb Bhimrao Ambedkar University (Central University), Satellite Centre, Amethi, UP, India*
[3]*Department of Computer Science and Engineering, Maharaja Agrasen Institute of Technology, Delhi, India*

Correspondence should be addressed to Deepak Gupta; deepakgupta@mait.ac.in

Received 15 June 2021; Revised 14 February 2022; Accepted 17 February 2022; Published 10 March 2022

Academic Editor: Carles Gomez

The Internet of Things (IoT) cyberattacks of fully integrated servers, applications, and communications networks are increasing at exponential speed. As problems caused by the Internet of Things network remain undetected for longer periods, the efficiency of sensitive devices harms end users, increases cyber threats and identity misuses, increases costs, and affects revenue. For productive safety and security, Internet of Things interface assaults must be observed nearly in real time. In this paper, a smart intrusion detection system suited to detect Internet of Things-based attacks is implemented. In particular, to detect malicious Internet of Things network traffic, a deep learning algorithm has been used. The identity solution ensures the security of operation and supports the Internet of Things connectivity protocols to interoperate. An intrusion detection system (IDS) is one of the popular types of network security technology that is used to secure the network. According to our experimental results, the proposed architecture for intrusion detection will easily recognize real global intruders. The use of a neural network to detect attacks works exceptionally well. In addition, there is an increasing focus on providing user-centric cybersecurity solutions, which necessitate the collection, processing, and analysis of massive amounts of data traffic and network connections in 5G networks. After testing, the autoencoder model, which effectively reduces detection time as well as effectively improves detection precision, has outperformed. Using the proposed technique, 99.76% of accuracy was achieved.

1. Introduction

Deep learning frameworks became an active field for the role of network intrusion detection in cybersecurity. While many excellent surveys cover a growing research field on this topic, the literature fails to make an impartial comparison of different deep learning models, especially in recent datasets for intrusion detection, in a controlled setting. Cybersecurity is a critical issue in today's world [1, 2]. Firewalls, for example, have long been used to secure data confidential information [1]. IDS analyses network traffic or a specific computer environment to detect signs of malicious action [2]. The rapid growth in interest in artificial intelligence (AI) development has resulted in major advances in mechanisms including pattern recognition and anomaly detection.

For these issues, neural networks are a suitable option, so their use is no longer limited. This is largely due to higher in the number of computing resources available. This situation caused researchers to make changes to neural network architectures to incorporate or improve IDS [3–5].

1.1. Internet of Things Devices Using 5G Networks. 5G is a key to the Internet of Things although it provides a faster network with greater capacity to meet connectivity needs. The 5G spectrum broadens the frequencies used by mobile communication technologies to transmit data. Because there is a wide range available for use, a whole bandwidth of mobile networks rises, enabling more devices to connect. To resolve current issues, 5G certainly requires control of the response period of the network and network infrastructure. Notice that

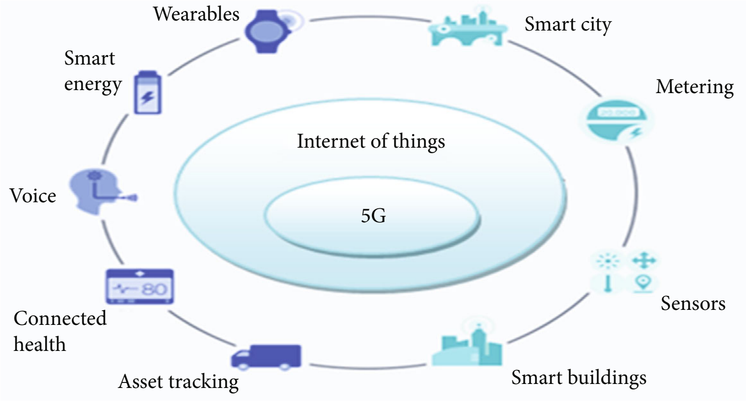

Figure 1: Applications of Internet of Things [5].

the model Internet of Things has a different system of interaction technology, including wireless sensor networks. Therefore, in this case, the role of fog calculation is also evident. Fog computing or fogging mostly consists of effective distribution of data, transmission, stocking, and applications between data sources and the cloud, through a decentralized networking and computing framework. Various applications of 5G network are depicted in Figure 1 which was reproduced from the article [5].

1.2. The Architecture of 5G Technology. 5G has an integrated infrastructure that usually updates network modules and terminals to include a new scenario. Advanced technology may often be used by service companies to easily take advantage of value-added offers. However, upgradability is based on cognitive radio technology which has many important characteristics such as the ability of devices to recognize their place, voice, sensors, health, energy, environment, temperature, etc. In its working environment, cognitive radio equipment works as a transceiver (beam) that can collect radio signals perceptually and respond to them. Furthermore, it detects changes in its environment automatically and thus responds to continuous continuity.

As soon as 4G becomes publicly accessible, package companies will be forced to adopt a 5G network [6]. To satisfy customer demands and resolve conflicts in the 5G environment, a fundamental change in the 5G wireless cellular technology growth is needed. According to the researchers' major findings in [7], the majority of wireless customers spend essentially 80% of their time indoors and 20% of their time outdoors (2018)). That is a narrow blend of both NFV and SDN innovations which are efficiently detected by 5G networks and cyberattacks that mitigate them. To address this challenge, a new concept or design technique for planning the 5G cellular architecture has emerged: distinguishing between outside and inside setups [8]. The accessibility loss across the building's boundaries will be slightly decreased with this design technique. The user details would be filtered by other user computers as in the device-to-device communications, so the anonymity of this information is the key concern. Closed access ensures the confidentiality of programs at the system stage. The system lists such trustworthy devices as consumers situated near you or in your place of business, you know; otherwise, a trusted entity, e.g., an organization, will easily link and maintain a degree of confidentiality, whereas devices that are not included in this list can interact on the macrocell phase.

One of the important issues in 5G network lies in the component which is used in the designing as well as at the deployment phase, as every element needs authentication with all of the other elements in the network architecture even before initiating any operation, whereas in physical layer phase the network, components are also required to be developed through the trusty worthy network components. As the traffic on internet is growing continuously, the domain is also constantly updating 5G and IoT technologies, lot of security breaches are there which can be easily affected due to intrusion-based attacks like Denial-of-Service Attacks (DDoS) which can not only affect the application layer of the Open Systems Interconnection (OSI) model but also affects the network layer as well. In this paper, the dataset used for the implementation purpose consists of all such kinds of attacks which are feasible on not only 5G networks but also IoT-based system. Hence, a novel technique to detect such type of attacks is elaborated in Section 3 of this paper.

1.3. The Contribution of the Paper Is as Follows

(i) Autoencoder-based novel deep learning technique is implemented for the detection of network attacks

(ii) Several machine learning algorithms are used for implementation purposes

(iii) A recent benchmark dataset is used for the implementation purpose

(iv) A comparative analysis of the work with the existing framework has been provided

1.4. The Structure of This Paper Is as Follows. Section 2 outlines the literature review based upon various intrusion detection systems. The third section reveals the research

Table 1: Comparative analysis of various existing frameworks.

Year	Performance evaluation parameters	Dataset	Algorithm/technique/ approach	Findings
2017 [10]	Accuracy, precision, recall, F1	RedIRIS	RNN and CNN	In this work, the authors have improvised the existing deep learning algorithms by modifying the hidden layers.
2017 [11]	Accuracy, precision, recall, F1, FAR	KDD 99 Cup	GRU and RF	Minimizing the loss function has helped the researchers to achieve better results.
2018 [12]	Accuracy, precision, recall, F1 score, MR, FAR, detection time	UNSW_ NB15	BLSTM RNN	In this work, feature normalization and conversion of categorical features to numeric values have helped them to generate improvised results.
2018 [13]	Accuracy, precision, recall, F1 measure	NSL-KDD	DNN	In this work, SGD was used to minimize the loss function of DNN.
2019 [14]	Accuracy, precision, recall	CICIDS2017	MLP, 1d-CNN, LSTM, and CNN+LSTM	In this work, researchers have balanced the dataset by performing data processing in which they have duplicated the records.
2019 [15]	Precision, recall (TPR), F1 score	NSL-KDD	DBN	In this work, deep neural network was optimized by assigning a cost function to each layer of their proposed model.
2019 [16]	Accuracy, precision, recall, F1 measure	NSL-KDD	SDPN	SMO algorithm is used for optimal selection of features.
2020 [17]	Accuracy, precision, recall, F1 measure	NSL-KDD	RF	In this work, Weka tool was used for evaluation purposes.
2021 [18]	Accuracy, precision, recall, F1 measure	NSL-KDD, KDD99	ANN	In this work, the stack-based feature selection technique has been proposed to optimize the computation time.
2021 [19]	F1 score	Bot-IoT	RF, NB, and MLP	In this work, a hierarchical approach was used for intrusion detection.
2021 [20]	Accuracy, precision, recall, F1 measure	CICIDS2017	HW-DBN	In this work, low frequency attack was detected.

methodology by providing a theoretical description of IDS and deep neural network (DNN) concepts along with the details of the proposed model. Following that, the fourth section provides the results and discussion of intrusion detection with system configuration and comparative analysis. The fifth section provides the conclusion based upon the presented work along with its future scope.

2. Literature Survey

2.1. Intrusion Detection System-Based Detection Systems. To identify possible computer intrusions, intrusion detection calls for monitoring and analysing running networks and networking traffic. The IDS system is a collection of methods and mechanisms for this purpose. In general, most IDSs have standard capabilities to secure the network [9]. An IDS starts with data collection from the observed incidents. It does detailed logging and compares operations with event-related data from different networks. The detector which employs various methods and related techniques is at the core of an IDS, depending on the situation.

There would also be mitigation capability. It is here that the process of identification and avoidance of intrusion is called as Intrusion Detection and Prevention System.

2.2. Anomaly-Based Detection Strategy. In anomaly-based recognition methods, different patterns, like generative, discriminatory, and hybrid structures, can be used. Where the assault is allocated to one form of attack, clustering should be binary if normal and until behaviours are to be distinguished. Divide into multisubclass hierarchical structures may be further broken down into many classes. Any studies utilizing hierarchical datasets have been performed. The detector often operates online or offline to support applications in real time. In the last several years, various research articles have been published on intrusion detection techniques, many of which are focused on the detection of network attacks using various machine learning and deep learning approaches. In Table 1, various existing frameworks are compared.

2.3. Summary of the Related Work. After reviewing various research papers based upon the intrusion detection system, it can be observed that commonly used datasets NSL-KDD and KDD99 were utilized for the implementation purpose. No doubt both the datasets are benchmark dataset seems to be outdated, and hence, work needs to be carried out on new dataset. Also, very few researchers have implemented the system based upon the Internet of Things and 5G network [20–24]. In this work, the UNSW-NB15 benchmark dataset which was created by the IXIA PerfectStorm tool in the Cyber Range Lab of UNSW Canberra was used [25, 26].

```
Generic              39496
Normal               19488
Exploits             16187
DoS                   1791
Fuzzers               1731
Reconnaissance        1703
Analysis               564
Worms                  114
Backdoor                99
Name: attack_cat, dtype: int64
```

Figure 2: Attack category.

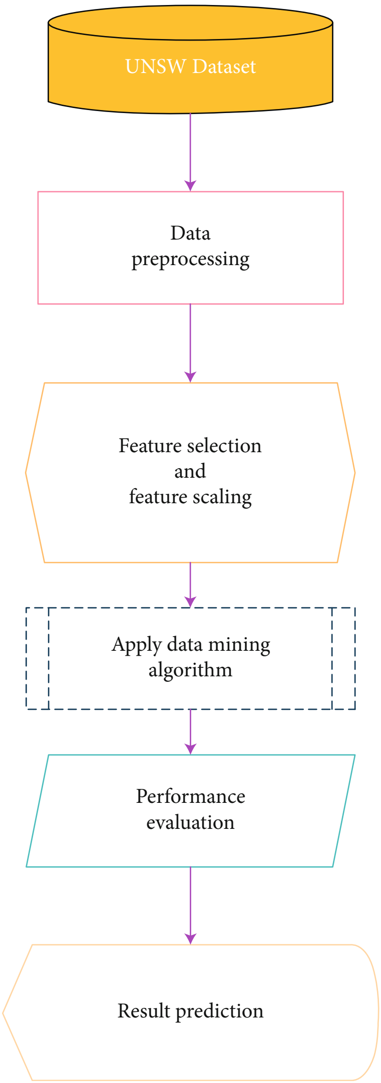

Figure 3: Generalized system flow chart.

3. Proposed Methodology

3.1. Background of Deep Learning Architectures. Deep learning is one of the popular techniques of data mining. Tiefskin's learning is a valuable algorithm for modelling abstract subjects and relationships over two neural layers [27]. Deep learning is currently being studied in several areas, including the identification of images, speech, natural language processing, social network filtering, and so forth. In addition to finding correlations between vast data from different sources to carry out attribute learning, classification, or classification tasks at the same time, deep-sea learning algorithms vary in their ability. Various deep

```
Input Given: Complete Dataset D=($a_1$, $a_2$.... $a_n$) where $a_i$ ➔X
Output: Prediction result for binary class label denoted by variable b
Step 1: Data Pre-processing (S) ⟶ S'
Step 2: Correlated sample Obtained was S"
Step 3: Training using AE $(S, F_\varphi, G_\theta)$ ⟶ Cust AE
Step 4: Apply AE (S' , Cust AE) ⟶ Full_AE
Step 5: S" = S', Where Full AE ⟶ Correlated Features
Step 6: Training using DNN (S"') ⟶ Cust DNN
Step 7: Add Cust DNN after Cust AE to form Cust AE+ Cust DNN
Step 8: Input Testing Data to generate class label b.
```

Algorithm 1: Proposed customized AE and DNN detection input.

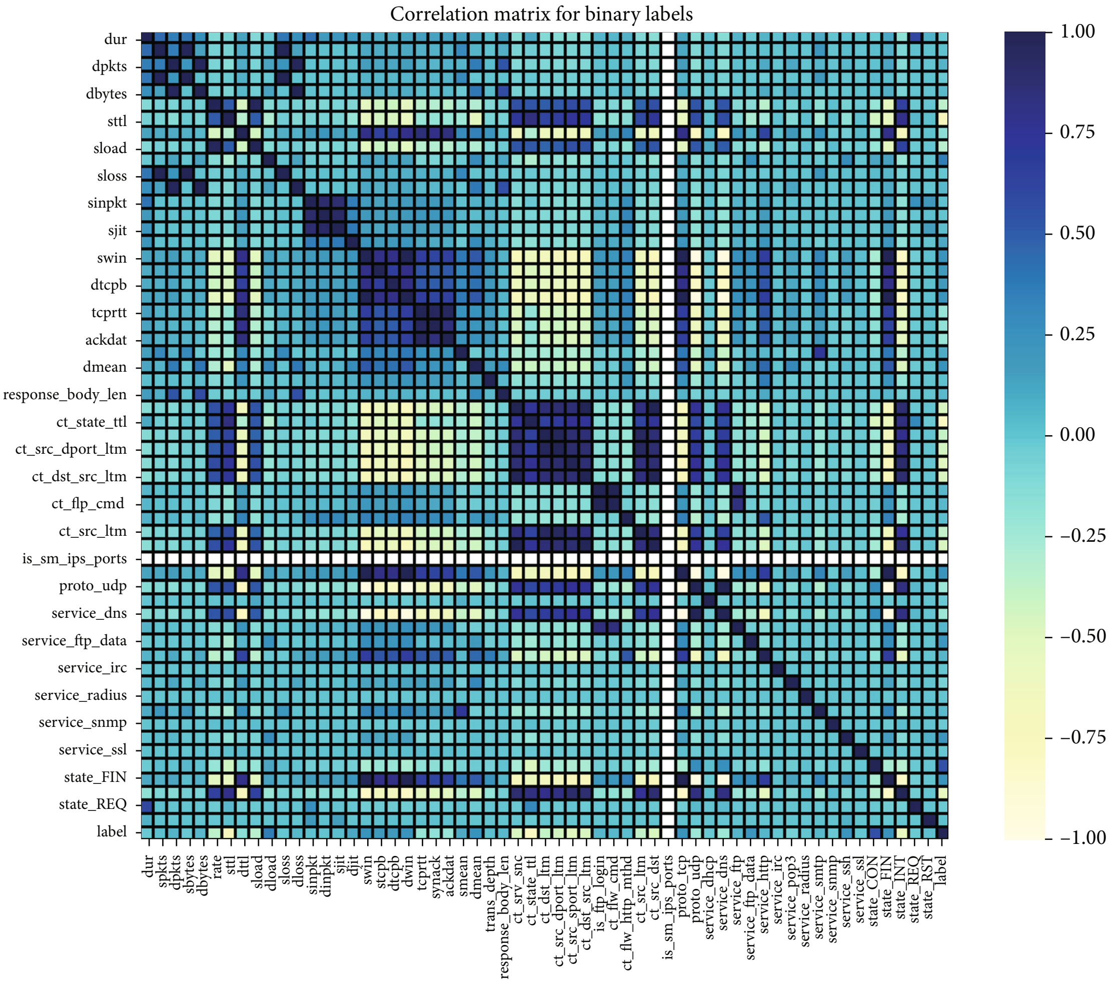

Figure 4: Heatmap for binary classification.

learning techniques which are used by different researchers for the development of intrusion detection system are discussed as follows:

(i) *Generative Architectures*. Unlisted raw data dynamically train algorithms to carry out different activities. This is the most general architecture in the architectural class category.

(ii) *Autoencoder (AE)*. Gao et al. [28] is a massive neural network widely used to minimize dimensionality by having improved data representation compared to raw inputs. The AE contains layers of the same number of feature vectors, in addition to a hidden layer with a low-dimensional representation. An AE incorporates and trains an encoder and decoder with a backpack. As knowledge is

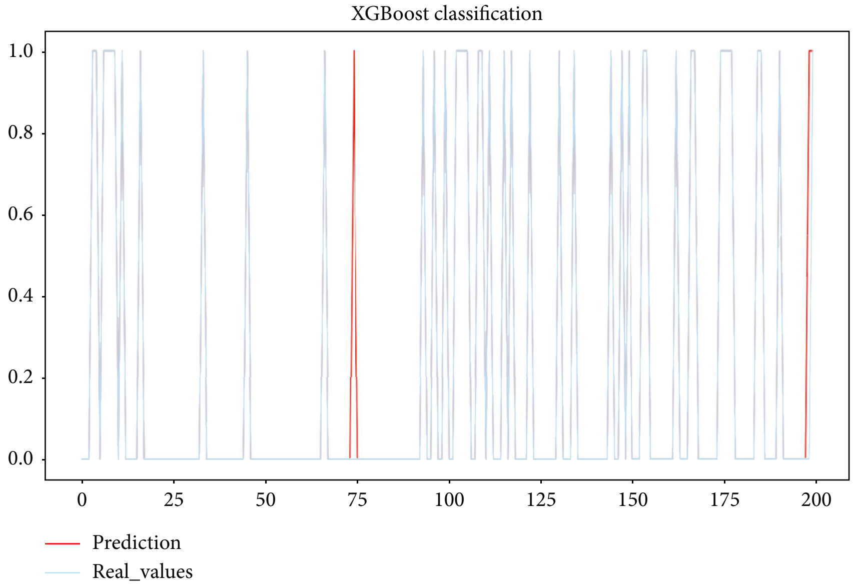

FIGURE 5: XGboost classification results.

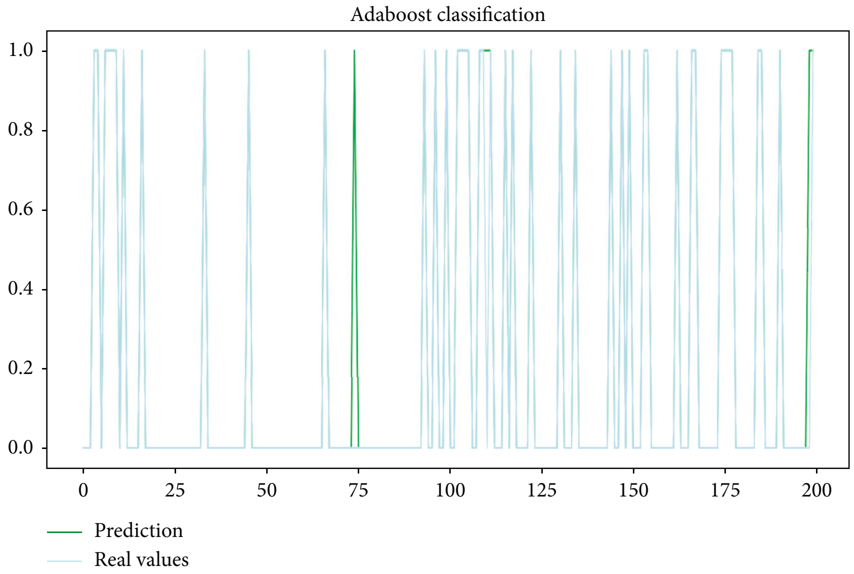

FIGURE 6: Adaboost classification results.

translated into small abstraction, it captures brutal characteristics and learns the representation of details. Afterward, the decoder receives small displays and reconstructs original functions [29]. Some AE extensions like AE (SAE), sparse AE, and denoise AE are available.

(iii) *SAE*. Cascade through a vast network and SAE via more than one hidden sheet. The features used to create a new data display are more thoroughly studied [30].

(iv) *Sparse AE*. Units covered in scarce AE are of a little sparse size. While there are many hidden units available to research data representations, AE remains valuable. Sparsity constraints are intended to ensure that most neurons are inactive [31] in the low average output.

(v) *Denoising AE*. Denoise is built on the use of skewed data for refined data view where hidden layers only use stable characteristics vectors [32].

(vi) *Restricted Boltzmann Machine (RBM)*. As a probabilistic neural network, the Boltzmann (BM) was created by Hinton and Sejnovsk. A BM network consists of binary units symmetrically connected which specifies which units are permitted. Several interactions between units, however, contribute to very slow learning [33]. RBM is the unidirectional

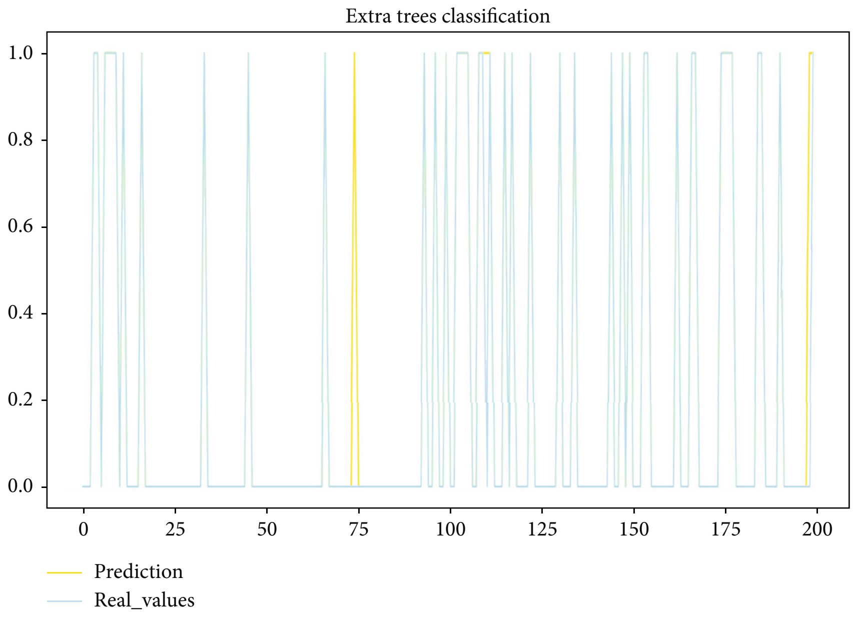

Figure 7: ExtraTreeClassifier results.

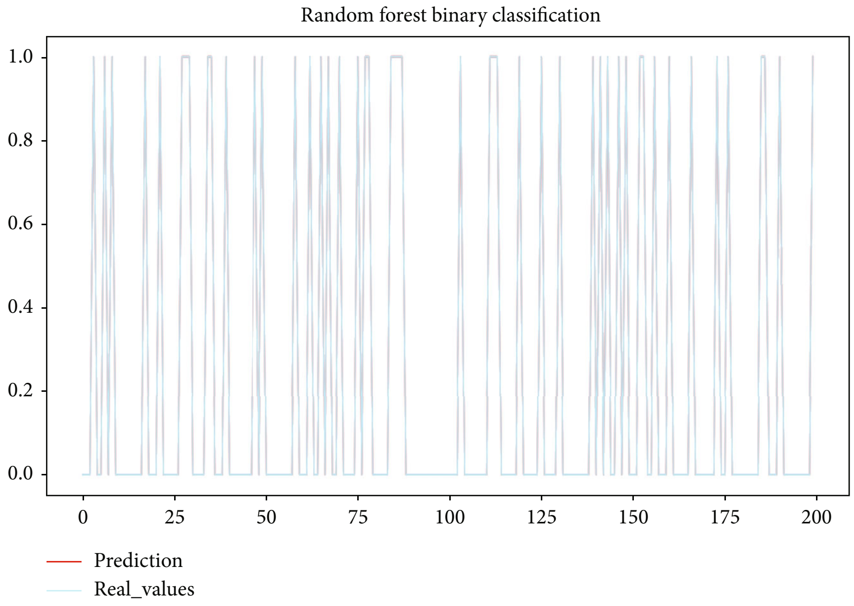

Figure 8: Random Forest Classifier results.

paradigm of Smolensky 1986 which solves BM's uncertainty. The principle of RBM is that neuronal connections are removed on the same sheet. RBM contains a translucent layer and an occult layer of latent (hidden) variables for initial input variables. Both units are connected to the hidden layer in a clear layer with corresponding weights. The feature distribution learns the units covered by the input variables. As an initial stage, RBM is typically employed as a preprocessing feature extractor or for initializing other network parameters, for another learning network. RBM may be used as a grouping model as well. A nonlinear, autonomous classification of Larochell and Bengaio was taught to be the discriminatory Boltzmann (DRBM) device. If some of the Boltzmann press are waterfalls, this is called a deeper Boltzmann (DBM).

(vii) *Deep Belief Network (DBN)*. The DBN consists of stacked RBMs, which have been trained in a pessimistic way. In comparison to the previous RBM, each RBM is trained and represents a contribution

```
Model : "sequential_2"
________________________________________________________________
Layer (Type)                     Outout shape          Param #
================================================================
Dense_ features (Densefeature)   Multiple                0
________________________________________________________________
Dense_6 (Dense)                  Multiple              22912
________________________________________________________________
Dense_7 (Dense)                  Multiple              16512
________________________________________________________________
Dense_8 (Dense)                  Multiple               129
================================================================
Total params : 39, 553
Trainable params : 39, 553
Non–trainable params : 0
```

Figure 9: Model summary for DNN.

to the hidden layer of the RBM. The algorithm for profound learning is efficient and rapid to move. In case an additional layer of bias is applied, DBN is generalized for both dimensional reduction and independent classification for practical applications.

(viii) *Recurrent Neural Network (RNN)*. Holand suggested RNN is a dynamic network of nerve feed in 1982. In normal transmission, depending upon the neural network architecture and its dependency, the output of each layer will consist of the same unit of the neuron. The discrepancy in the neural feed system feeds from hidden layers to RNNs. Various models of the memory unit such as long, temporary memory were extended, and the recurrent gated unit can be used.

(ix) *Long Shorter Memory Time (LSTM)*. The RNN gradient is addressed by LSTM. It will learn long-term dependencies by utilizing the gate scheme. Every LSTM device is equipped with a memory cell containing old states.

(x) *Gated System Recurrent (GRU)*. Lightweight is the GRU version of the LSTM. The architecture has been streamlined, the doors merged, and the states integrated.

(xi) *Neural Classic Network (NCN)*. The perceptron is multilayered and is further known as the fully connect network. The model must be modified to binary entries in simple records.

(xii) *Linear Function (LF)*. Rightly called, it is a single line multiplying the input by a constant multiplier.

(xiii) *Nonlinear Function (NLF)*. Furthermore, the nonlinear function is split into three subsets: curve sigmoid, which is an S-shaped feature with several zeros to 1. The S-shaped curve with a scale of -1 to 1 relates to the hyperbolic tangent (tanh).

```
Model : "functional_1"
________________________________________________________________
Layer (Type)                  Outout shape             Param #
================================================================
Input–1 (InputLayer)          ((None, 14))                0
________________________________________________________________
Dense (Dense)                  (None, 14)                210
________________________________________________________________
Dense_1 (Dense)                (None, 7)                 105
________________________________________________________________
Dense_2 (Dense)                (None, 7)                  56
________________________________________________________________
Dense_3 (Dense)                (None, 14)                112
================================================================
Total params : 483
Trainable params : 483
Non–trainable params : 0
```

Figure 10: Proposed model summary.

	No.	Name	Type	Description
0	1	Srcip	Nominal	Source IP address
1	2	Sport	Integer	Source port number
2	3	Dstip	Nominal	Destination IP address
3	4	Dsport	Integer	Destination port number
4	5	Proto	Nominal	Transaction Protocol

Figure 11: Sample feature description.

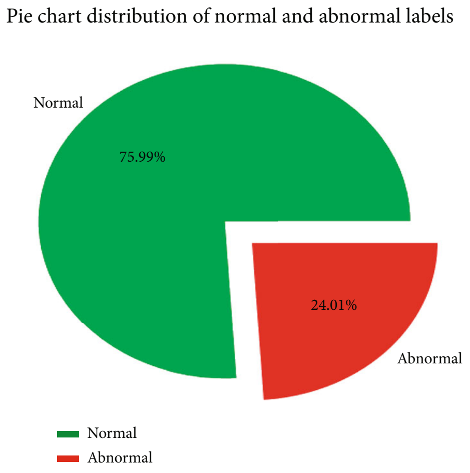

Figure 12: Dataset distribution.

3.2. Proposed Architecture. In this work, UNSW 2015 benchmark dataset was used. Initially, the dataset was analysed in the data preprocessing phase where the null value-based columns were dropped. Further, the updated dataset was provided to feature selection and feature scaling phase. In this phase, important features were considered using the Pearson correlation technique. Attack category used for the analysis purpose is depicted in Figure 2. After obtaining the important features, the categorical features were

Table 2: Results obtained using various deep learning approaches.

Sr. No.	Algorithm	Accuracy (%)
1.	Deep neural network	91.72
2.	Recurrent neural network+convolutional neural network [10]	96.12
2.	Gated recurrent neural networks [11]	98.91
3.	Bidirectional long short-term memory recurrent neural network [12]	95.71
4.	Distributed deep model [13]	99.20
5.	Convolutional neural network+long short-term memory [14]	97.16
6.	Spider monkey optimization+stacked-deep polynomial network [16]	99.02
7.	Random Forest [17]	99.70%
8.	Artificial neural network [18]	99.00%
9.	Hybrid Weighted Deep Belief Network [20]	99.38
10.	Proposed methodology	99.76

Table 3: Results obtained using various machine learning algorithms.

Sr. No.	Algorithm	Accuracy	R2 score	Precision	Recall	F1 score	MAE	MSE	RMSE
1.	XGboostClassifier	0.984	0.910	0.98	0.99	0.99	0.015	0.015	0.125
2.	AdaboostClassifier	0.983	0.910	0.98	0.99	0.99	0.016	0.0164	0.128
3.	ExtraTreeClassifier	0.983	0.910	0.98	0.99	0.99	0.016	0.016	0.128
4.	Random Forest Classifier	0.986	0.924	0.99	0.99	0.99	0.013	0.013	0.117

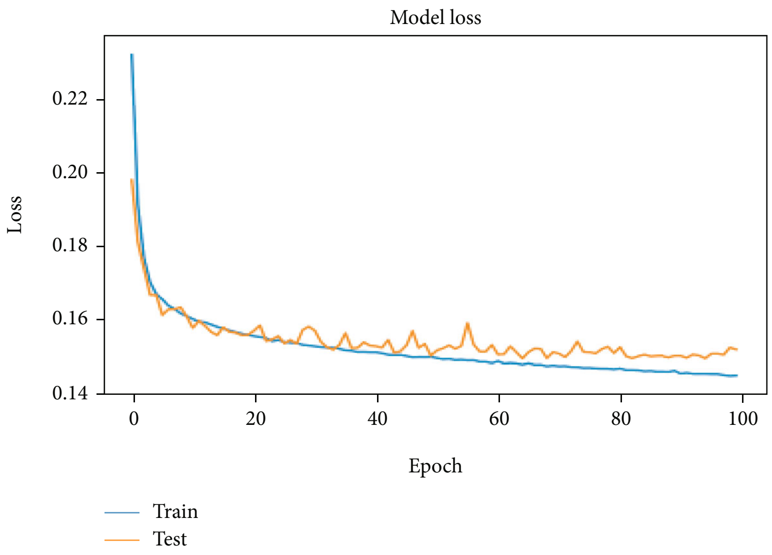

Figure 13: Training validation accuracy of the proposed model for 100 epochs.

converted to the numeric features using the one-hot encoding technique. Later for scaling, the feature normalization and standardization techniques were used. Finally, various machine learning and deep learning algorithms were used for training the model. In the training phase, 80% of the dataset was utilized, while for testing the model, remaining 20% of the dataset was used. The proposed autoencoder technique has outperformed compared to other models. Initially, a deep neural network and proposed model were implemented using 100 epochs. Finally, the proposed model has got more promising results when it was trained with 5000 epochs. The generalized flow chart of the proposed model is depicted in Figure 3. The proposed system algorithm has been represented in Algorithm 1 as follows:

For identifying the highly correlated features in the dataset, Pearson correlation technique was used. The

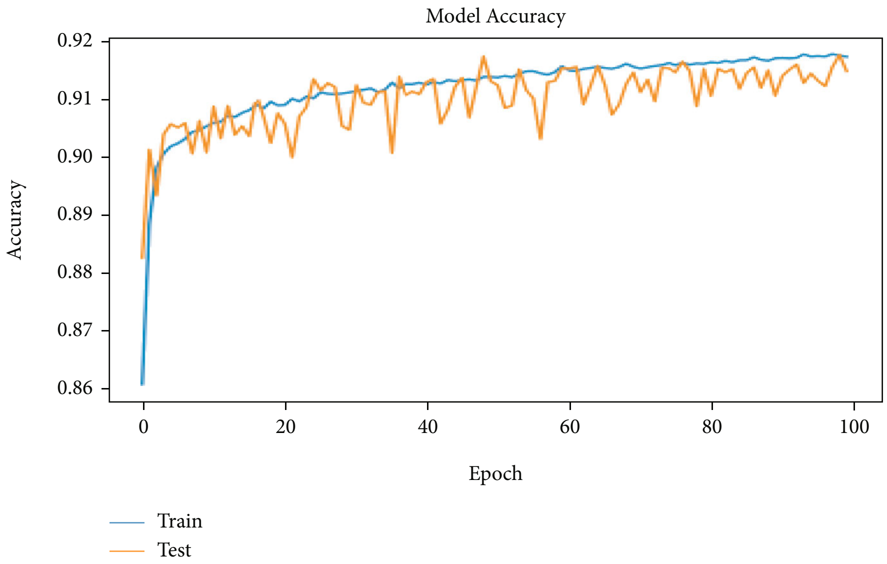

Figure 14: Training validation loss of the proposed model for 100 epochs.

correlated features were analysed through heatmap as provided in Figure 4.

For implementation purpose, ensemble-based machine learning techniques were considered. XGboost, Adaboost, ExtraTreeClassifier, and Random Forest Classifier were used for implementation purpose. Results obtained through various machine learning algorithms are depicted in Figures 5–8.

Figures 9 and 10 represent the model summary of deep neural network and the proposed methodology, respectively.

4. Results and Discussions

4.1. Dataset Description and Preprocessing. In this work, the UNSW-NB15 dataset was used for the implementation. This dataset consists of a total of 175341 rows and 45 attributes. In the dataset preprocessing step, initially, the null values were dropped. Due to this, the dataset was converted into almost half of its size. Further, to handle the categorical features, encoding techniques were used. Later for performing feature scaling, the normalization technique was applied. Figure 2 provides the details of the attack category available in the dataset. The sample of feature description is depicted in Figure 11. Dataset distribution after splitting in training and testing samples is depicted in Figure 12. For generating efficient results of the proposed model, the implementation is done with high configuration architecture which is comprised of AMD RYZEN 9 processor with 8 cores, 64-bit Windows 10 OS, 16 GB of RAM, and 6 GB GTX 1660 TI GPU. The complete model script was implemented on Jupyter Notebook tool using python programming language.

4.2. Evaluation Metrics. The main aim of the evaluation metrics is to depict the implications of enriching the IDS with the proposed model using some of the following important parameters:

(i) Maximize detection rate (DR)

$$\text{Detection rate} = \frac{\text{true positive}}{(\text{true positive} + \text{false negative})} \tag{1}$$

(ii) Maximize accuracy (AC)

$$\text{False alarm rate} = \frac{\text{false positive}}{(\text{false positive} + \text{true positive})} \tag{2}$$

(iii) Minimize false alarm rate (FA)

$$\text{Maximize accuracy} = \frac{\text{true positive} + \text{true negative}}{(\text{true negative} + \text{true positive} + \text{F N} + \text{false positive})} \tag{3}$$

Our model obtained higher accuracy compared to the existing model as depicted in Table 2. The proposed model was tested on various performance metrics, and classification accuracy was used as a comparison parameter with the existing model.

Table 3 provides the details of the results obtained using various machine learning algorithms. Results obtained using the deep learning algorithm and proposed methodology are provided in Table 2. In Table 2, only results obtained from those researchers are considered who have used accuracy as their comparative parameter.

In Figures 13 and 14, the accuracy results and loss results of the proposed methodology computed for 100 epochs are depicted. An appendix of all the acronyms is given in Table 4.

Table 4: Appendix of all the acronyms.

Sr. No.	Term	Acronym
1	Internet of Things	IoT
2	Intrusion detection system	IDS
3	Denial of Service	DoS
4	Artificial intelligence	AI
5	Deep neural network	DNN
6	Wireless sensor network	WSN
7	Recurrent neural network	RNN
8	Convolutional neural network	CNN
9	Gated recurrent unit	GRU
10	Bidirectional long short-term memory recurrent neural network	(BLSTM RNN)
11	False alarm rate	FAR
12	Stochastic gradient descent	SGD
13	Multilayer perceptron	MLP
14	Long short-term memory	LSTM
15	Spider monkey optimization	SMO
16	Stacked-deep polynomial network	SDPN
17	Artificial neural network	ANN
18	Random Forest	RF
19	Naïve Bayes	NB
20	Autoencoder	AE
21	Sparse AE	SAE
22	Restricted Boltzmann machine	RBM
23	Boltzmann	BM
24	Discriminatory Boltzmann	DRBM
25	Deeper Boltzmann	DBM
26	Deep belief network	DBN
27	Recurrent neural network	RNN
28	Neural classic networks	NCN
29	Fully connect network	FCN
30	Linear function	LF
31	Nonlinear function	NLF
32	Mean absolute error	MAE
33	Mean squared error	MSE
34	Root mean squared error	RMSE
35	Miscalculation rate	MR
36	Hybrid Weighted Deep Belief Network	HW-DBN
37	Denial-of-Service Attacks	DDoS
38	Open Systems Interconnection	OSI

5. Conclusion and Future Scope

The proposed algorithm is trained using the SoftMax classifier to identify the attack types in the dataset. The benchmark dataset, UNSW-NB15, was used for training and testing the model. For training the hidden layers, there are many options, such as linear, SoftMax, sigmoid, and corrected linear functionality which can be used as an activation function. A novel autoencoder technique was used for training and testing the model. The proposed model has achieved comparatively high accuracy than the existing system. Several machine learning and deep learning approaches were used for implementation purposes. Further, to extend the work, stack-based autoencoder technique can be used for reducing the computational resources. Also, more focus can be given to optimizing the computational time.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

References

[1] B. Susilo and R. F. Sari, "Intrusion detection in IoT networks using deep learning algorithm," *Information*, vol. 11, no. 5, p. 279, 2020.

[2] A. Aldweesh, A. Derhab, and A. Z. Emam, "Deep learning approaches for anomaly-based intrusion detection systems: a survey, taxonomy, and open issues," *Knowledge-Based Systems*, vol. 189, p. 105124, 2020.

[3] L. Fernández Maimó, A. Huertas Celdrán, M. Gil Pérez, F. J. García Clemente, and G. Martínez Pérez, "Dynamic management of a deep learning-based anomaly detection system for 5G networks," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 8, pp. 3083–3097, 2019.

[4] V. Yihua Liao, "Rao Vemuri, Use of K-nearest neighbor classifier for intrusion detection11An earlier version of this paper is to appear in the Proceedings of the 11th USENIX Security Symposium, San Francisco, CA, August 2002," *Computers & Security*, vol. 21, no. 5, pp. 439–448, 2002.

[5] N. Abdullah, *Advances in Cyber Security: Third International Conference, ACeS 2021, Penang, Malaysia, August 24–25, 2021, Revised Selected Papers*, Springer Nature, Penang, Malaysia, 2021, https://link.springer.com/book/10.1007/978-981-16-8059-5.

[6] E. Hodo, X. Bellekens, A. Hamilton, C. Tachtatzis, and R. Atkinson, "Shallow and deep networks intrusion detection system: a taxonomy and survey," 2017, https://arxiv.org/abs/1701.02145.

[7] Y. Bengio, "Learning Deep Architectures for AI," *Found. Trends Mach. Learn Foundations and Trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[8] J. Cannady, "Artificial neural networks for misuse detection," *InNational information systems security conference*, vol. 26, pp. 443–456, 1998.

[9] A. Khamparia, S. Pande, D. Gupta, A. Khanna, and A. K. Sangaiah, "Multi-level framework for anomaly detection in social networking," *Library Hi Tech*, vol. 38, no. 2, pp. 350–366, 2020.

[10] M. Lopez-Martin, B. Carro, A. Sanchez-Esguevillas, and J. Lloret, "Network traffic classifier with convolutional and recurrent neural networks for Internet of Things," *IEEE Access*, vol. 5, pp. 18042–18050, 2017.

[11] M. Kumar, *Deep Learning Approach for Intrusion Detection System (IDS) in the Internet of Things (IoT) Network Using Gated Recurrent Neural Networks (GRU)*, Wright State University, 2017, https://corescholar.libraries.wright.edu/etd_all/1848/.

[12] 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 *in 2018 28th International Telecommunication Networks and Applications Conference (ITNAC)*, pp. 1–6, Sydney, NSW, Australia, November 2018.

[13] A. A. Diro and N. Chilamkurti, "Distributed attack detection scheme using deep learning approach for Internet of Things," *Future Generation Computer Systems*, vol. 82, pp. 761–768, 2018.

[14] M. Roopak, G. Y. Tian, and J. Chambers, "Deep learning models for cyber security in IoT networks," in *in 2019 IEEE 9th annual computing and communication workshop and conference (CCWC)*, pp. 452–457, Las Vegas, NV, USA, January 2019.

[15] G. Thamilarasu and S. Chawla, "Towards deep-learning driven intrusion detection for the internet of things," *Sensors*, vol. 19, no. 9, p. 1977, 2019.

[16] Y. Otoum, D. Liu, and A. Nayak, "DL-IDS: a deep learning-based intrusion detection framework for securing IoT," *Transactions on Emerging Telecommunications Technologies*, article e3803, 2019.

[17] S. Pande, A. Khamparia, D. Gupta, and D. N. H. Thanh, "DDOS Detection Using Machine Learning Technique," in *Recent Studies on Computational Intelligence. Studies in Computational Intelligence,*, A. Khanna, A. K. Singh, and A. Swaroop, Eds., vol. 921, Springer, Singapore, New Delhi, India, 2021.

[18] S. Pande, A. Khamparia, and D. Gupta, "An intrusion detection system for health-care system using machine and deep learning," *World Journal of Engineering*, 2021.

[19] G. Bovenzi, G. Aceto, D. Ciuonzo, V. Persico, and A. Pescapé, "A hierarchical hybrid intrusion detection approach in IoT scenarios," in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, pp. 1–7, Taipei, Taiwan, December 2020.

[20] Z. K. Maseer, R. Yusof, S. A. Mostafa, N. Bahaman, O. Musa, and B. A. Al-rimy, "DeepIoT.IDS: hybrid deep learning for enhancing IoT network intrusion detection," *CMC-Computers, Materials & Continua*, vol. 69, no. 3, pp. 3945–3966, 2021.

[21] T. O'Shea and J. Hoydis, "An Introduction to deep learning for the physical layer," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563–575, 2017.

[22] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescapé, "Mobile encrypted traffic classification using deep learning: experimental evaluation, lessons learned, and challenges," *IEEE Transactions on Network and Service Management*, vol. 16, no. 2, pp. 445–458, 2019.

[23] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, "Kitsune: an ensemble of autoencoders for online network intrusion detection," 2018, https://arxiv.org/abs/1802.09089.

[24] M. Almiani, A. AbuGhazleh, Y. Jararweh, and A. Razaque, "DDoS detection in 5G-enabled IoT networks using deep Kalman backpropagation neural network," *International Journal of Machine Learning and Cybernetics*, vol. 12, no. 11, pp. 3337–3349, 2021.

[25] N. Moustafa and J. Slay, "The evaluation of network anomaly detection systems: statistical analysis of the UNSW-NB15 dataset and the comparison with the KDD99 dataset," *Information Security Journal: A Global Perspective*, vol. 25, no. 1-3, pp. 1–14, 2016.

[26] A. Bhardwaj, V. Mangat, and R. Vig, "Hyperband tuned deep neural network with well posed stacked sparse autoencoder for detection of DDoS attacks in cloud," *IEEE Access*, vol. 8, pp. 181916–181929, 2020.

[27] L. Yuancheng, M. Rong, and J. Ruhai, "A hybrid malicious code detection method based on deep learning," *Journal of Security and Its Applications*, vol. 9, no. 5, pp. 205–216, 2015.

[28] N. Gao, L. Gao, Q. Gao, and H. Wang, "An intrusion detection model based on deep belief networks," in *in 2014 Second*

International Conference on Advanced Cloud and Big Data, pp. 247–252, Huangshan, China, November 2014.

[29] C. Zhou and R. C. Paffenroth, "Anomaly detection with robust deep autoencoders," in *InProceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 665–674, New York, NY, United States, August 2017.

[30] M. Abadi and D. G. Andersen, "Learning to protect communications with adversarial neural cryptography," 2016, https://arxiv.org/abs/1610.06918.

[31] 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, 2016.

[32] D. Kwon, H. Kim, J. Kim, S. C. Suh, I. Kim, and K. J. Kim, "A survey of deep learning-based network anomaly detection," *Cluster Computing*, vol. 22, pp. 1–13, 2019.

[33] M. Usama, J. Qadir, A. Raza et al., "Unsupervised machine learning for networking: techniques, applications and research challenges," 2017, https://arxiv.org/abs/1709.06599.