Hindawi
Journal of Advanced Transportation
Volume 2023, Article ID 9801729, 1 page
https://doi.org/10.1155/2023/9801729

Retraction

Retracted: An Improved Deep Belief Network IDS on IoT-Based Network for Traffic Systems

Journal of Advanced Transportation

Received 12 December 2023; Accepted 12 December 2023; Published 13 December 2023

This article has been retracted by Hindawi, as publisher, following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of systematic manipulation of the publication and peer-review process. We cannot, therefore, vouch for the reliability or integrity of this article.

Please note that this notice is intended solely to alert readers that the peer-review process of this article has been compromised.

Wiley and Hindawi regret that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] R. Malik, Y. Singh, Z. A. Sheikh, P. Anand, P. K. Singh, and T. C. Workneh, "An Improved Deep Belief Network IDS on IoT-Based Network for Traffic Systems," *Journal of Advanced Transportation*, vol. 2022, Article ID 7892130, 17 pages, 2022.

Hindawi
Journal of Advanced Transportation
Volume 2022, Article ID 7892130, 17 pages
https://doi.org/10.1155/2022/7892130

Review Article

An Improved Deep Belief Network IDS on IoT-Based Network for Traffic Systems

Rayeesa Malik,[1] **Yashwant Singh**,[1] **Zakir Ahmad Sheikh**,[1] **Pooja Anand**,[1] **Pradeep Kumar Singh**,[2] **and Tewabe Chekole Workneh**[3]

[1]*Department of Computer Science and Information Technology, Central University of Jammu, J&K 181143, India*
[2]*Department of Computer Science, KIET Group of Institutions, Delhi-NCR Ghaziabad, Uttar Pradesh, India*
[3]*Admas University, Addis Ababa, Ethiopia*

Correspondence should be addressed to Yashwant Singh; yashwant.csit@cujammu.ac.in and Tewabe Chekole Workneh; tewabe.chekole@amu.edu.et

Received 25 January 2022; Revised 8 March 2022; Accepted 23 March 2022; Published 25 April 2022

Academic Editor: Muhammad Arif

Internet of things (IoT) services are turning out to be more domineering with the rising security considerations fading with time. All this owes to the propagating heterogeneity and budding technologies teamed up with resource-constrained IoT systems, sculpting smart systems to be more susceptible to cyber-attacks. The security challenges such as privacy, scalability, authenticity, trust, and centralization thwart the quick adaptation of the smart services; hence, effective solutions are needed to be in place. Traditional approaches of intrusion detection mechanisms have become irrelevant now, as the bad actors often use obfuscation techniques to evade detections. Moreover, these techniques collapse, while detecting zero-day attacks. Hence, there is a need to use an intelligent mechanism based on machine learning (ML) and deep learning (DL), to detect attacks. In this study, the authors have proposed an intrusion detection engine with a deep belief network (DBN) being the core. The implementation of DBN_Classifier is performed using TensorFlow 2.0 and evaluated using a sample of the TON_IOT_Weather dataset. The findings indicate that the proposed engine outperforms the other state-of-the-art techniques with an average accuracy of 86.3%.

1. Introduction

In the present era, diverse objects connected to the Internet all around the world have paved a way for the smart world around us. The connected objects are individually recognizable and are capable of sensing, acting, and communicating without the need for human intervention [1] owing to the IoT, bequeathed by Kevin Asthon in 1999 [2]. IoT has taken its place in almost all the arenas seeing from health care, smart grid, smart cities, smart farming, industries, and transportation. Thus, IoT-based services have created a tremendous impact on people's lives. Their exponential growth is depicted in Figure 1 with the projected IoT-powered (interconnected) devices crossing 100 billion by 2040 [3]. Participant's solutions, such as IoT assistance, enable impaired people to experience freedom and social involvement [4]. By enabling continuous tracking of health conditions, IoT has changed the lives of people, especially older patients. Wearables in the form of appliances, such as fitness bands, cuffs for heart rate monitoring, and glucometer, provide access to customized attention for patients [5]. The use of smart grids and smart meters has optimized the daily electricity usage and the proper maintenance of the supply-demand ratio. The use of smart agriculture facilitates the identification and isolation of disease-prone crop areas, prediction of crop yield, and fertilizer requirements [2].

With the accelerated growth, the IoT also complements the entrenched security challenges because the communication stack for IoT systems has oodles of vulnerabilities to enter into the system. Consequently, it leads to a substantial range of cyber-attacks. Manufacturers owe a lot by disregarding the security concerns and producing devices that can be readily hacked [6]. Currently, nine billion things are connected to the Internet as of now. It is seen that 75% of interconnected gadgets are vulnerable to cyber threats [7].

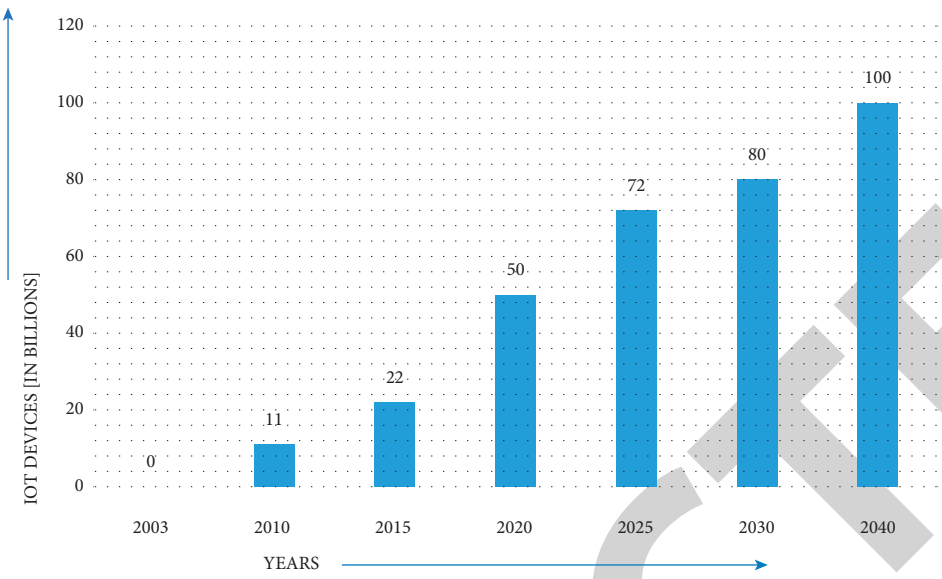


Figure 1: Growth of IoT devices.

In addition, 25% of attacks on the industrial sector have been attributed to infect the IoT systems by the end of 2020 according to statistics [8]. As evidenced from the cyber-attacks launched successfully by hewed IoT devices, with Mirai (2016), Hajime botnet (2016), and Persirai (2017), Memcached (2018) demonstrates the severity of the problem [9, 10]. These exploits are launched by hiring a squad of compromised IoT devices that seek to spread the malware by targeting more and more hewed devices. For example, the Mirai attack in 2016 was a botnet attack that attempted a DDOS attack using telnet, thus shutting down the various internet infrastructure. In 2018, the Memcached attack became apparent. A Memcached DDoS attack aimed at Memcached, to speed up the network with traffic to crash the servers [11]. In 2018, India's national ID database named under the Aadhaar has been targeted, in which 1.1 billion records were lost, and it is taken as the biggest record leak as of January 2020 [12].

To the rescue, several intrusion detection systems (IDS) have already come to the picture. Although, the issues need to be resolved accompanying these IDS. These IDS systems work upon different approaches such as the signature-based approach, which compares current system data to a documented signature of an intrusion attack that is saved in the IDS database. When the IDS detects a match, it classifies it as an intrusion. But these signature databases must be maintained regularly, and the device may be hacked before the next intrusion attack is patched [13]. Moreover, it has other downsides such as overloading the network, high signature matching prices, and a high number of false alarms.

Another way goes with the anomaly-based or behavior-based approach, which detects an intrusion when the device behaves abnormally. However, this technique has poor accuracy and a high rate of false alarms as a hindrance [14]. Putting together, there is a hybrid approach, detecting identified attacks using a signature-based approach and unknown attacks using an anomaly-based approach. Though this approach results in more precise detection, with incrementing potential to be inefficient and higher computational costs [15].

With the list of downsides, including the inability to distinguish new malicious threats, the need to be modified, poor accuracy, a high rate of false alarms, and the inability to detect zero-day attacks, the learning-based methods have the breakthrough. These methods have come a long way in recent years, and artificial intelligence has gone from being a curiosity in the lab to being used in a variety of critical applications. With the ability to intelligently track, IoT devices offer major protection against new attacks named under the zero-day attacks. They have proven to be effective data discovery methods for learning about "normal" and "abnormal" actions in the IoT environment based on how IoT components and devices work.

Furthermore, through learning from current instances, learning-based approaches may intelligently predict unknown attacks, which are often variants of previous attacks. As a result, learning methods are useful in transforming IoT protection from merely enabling safe communication between devices and intelligent secure systems [16]. To serve the purpose, several learning techniques have been used for the detection of attacks in IoT including Naive-Bayes [17], KNN [18], decision tree [19], SVM [20], and ANN [21].

The versatility, scalability, and low CPU load of ML techniques will enable us to develop a variety of analytical models for attack and anomaly detection that are more accurate and have lower false alarm rates. In our proposed model, we have used DL-based IDS known as DBN_Classifier because of its low rate of false alarms and better classification. It achieves a higher detection rate to detect attacks and high-level feature extraction, as it is a probabilistically generated model. It is used to efficiently initialize the DBN's parameters by reducing data dimensions.

1.1. Key Contributions. The key contributions of this study are as follows:

(i) The background preliminaries and the importance of IDS in IoT have been discussed. Also, the need for developing an intelligent IDS over traditional IDS has been discussed.

(ii) Various techniques and datasets used for IDS in IoT networks using ML and DL are discussed.

(iii) Also, the unique categorization of IoT attacks and intrusion detection approaches is proposed.

(iv) We proposed the DBN-based intrusion detection engine, DBN_Classifier, and evaluated it on the TON_IOT_Weather dataset.

1.2. Methods and Materials. To direct the proposed work appropriately, a systematic approach is followed to analyze the different aspects of IoT, particularly the security challenges and the different ways that IDS systems work in general with their downsides. The role of learning-based methods to secure the IoT system is also studied. This research is steered using various articles, blogs, research publications, and white papers. This research is mainly focused on IoT attacks, vulnerabilities, threats, and anomalies. To obtain valid data for the intended research, quality checks have been carried out. The ones from SCI journals with many citations are generally selected. The relevant research publications are found in high-quality database journals and prominent conferences such as IEEE Xplore, Springer, MDPI, ACM, Elsevier, and Google Scholar. The important keywords such as vulnerabilities, security, threats, IoT, attacks, ML, and DL are used to get the relevant literature.

1.3. Organization and Roadmap. To ensure the logical flow of content, we split our study into sections, and the organization of study is depicted in Figure 2. Moreover, all the acronyms used in this study are mentioned in Table 1. After discussing the introduction in Section 1, the rest of the study is organized as follows: Section 2 presents the research background and preliminaries in general, which is followed by Section 3 that in particular discusses the ML- and DL-based IDS for IoT. The proposed IDS model and datasets considered are discussed in Section 4 and Section 5, respectively. Section 6 includes results and discussion, which is followed by the conclusion in Section 7.

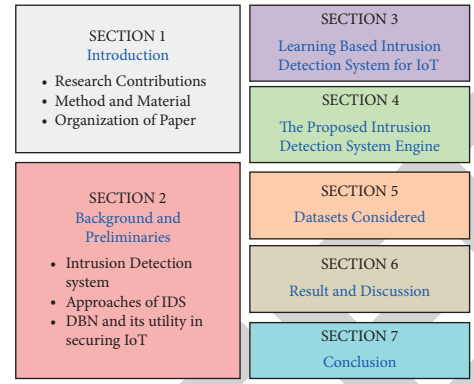


Figure 2: Organization of study.

Table 1: Acronyms used in this study and their meanings.

Acronym	Meaning
IoT	Internet of things
DBN	Deep belief network
RBM	Restricted Boltzmann machines
IDS	Intrusion detection system
HIDS	Host intrusion detection system
NIDS	Network intrusion detection system
DoS	Denial of service
R2L	Remote to user
U2R	User to root
CNN	Convolution neural network
ANN	Artificial neural network
LSTM	Long short term memory
ML	Machine learning
DL	Deep learning
NB	Naive-Bayes
PR	Precision
RC	Recall
F ms	F-measure
ACC	Accuracy
FAR	False alarm rate
RF	Random forest
DT	Decision tree
SVM	Support vector machine
KNN	K-nearest neighbour

2. Background and Preliminaries

The rising trend of smart services in society being followed by threats and attacks raises a serious concern for its sustenance. As the IoT becomes more deeply ingrained in our daily lives and communities, it is high time to take action and step up cybersecurity seriously [22]. Because of the "IoT" involvement in different applications, the risk of unauthorized access is far greater. Today's cyber-attacks on communication networks are extremely powerful and troubling. Cyber-attacks are becoming more complex, posing greater difficulties in detecting intruders. If intrusions are not prevented, security services such as data confidentiality, transparency, and availability will be at high risk [23]. There exist threats of different levels of severity. The more severe the threat is, the more should be the priority assigned to deal with it, so as to reduce the chances of higher consequences. Threats in the form of various IoT attacks have been categorized by authors in the existing works. The authors in reference [24] have classified IoT attacks based on architectural layers, that is, perception layer threats, network layer threats, support layer threats, and application layer threats. The study [25] has also classified IoT attacks based on architectural layers and medium, that is, physical attack, network attack, software attack, and encryption attack. Similarly, in study [26], the IoT attacks are classified based on device property, location, strategy, access level, and protocol. The study [27] has classified IoT attacks based on device property, location, strategy, access level, protocol,

host, information damage level, and communication stack protocol. Based on the existing works, we have prepared a unique categorization of IoT attacks based on various criteria and attributes, as depicted in Figure 3. It presents readers with a unique and all-in-one categorization of IoT attacks based on various properties and criteria. Possible categorizations in literature have been deeply reviewed to prepare a unique classification of IoT attacks.

2.1. Intrusion Detection System (IDS). The software program that tracks the malicious behavior of a network or system is called the intrusion detection system (IDS). It is also defined as the act of detecting behavior intended to compromise a resource's confidentiality, integrity, and availability [28]. Activities that render services of computers unresponsive to legitimate users are called intrusions. These systems are classified based on different attributes such as deployment location and working approach. Based on the deployment location, the IDS is categorized into network-based intrusion detection system (NIDS), host-based intrusion detection system (HIDS) [29], and hybrid. An IDS framework that uses the action of a network is called NIDS. Network activity is obtained by mirroring network components, such as switches and routers, using network equipment to detect attacks and potential threats hidden in network traffic [30]. A method was proposed by Martin et al. [31] on unsupervised NIDS for IoT environments. It was based on a conditional variation autoencoder (CVAE). Since it can retrieve missing features from incomplete training datasets, this technique is effective. Dataset used was the updated release of NSL-KDD3. Their work was experimentally complex as compared to other NIDS. The metrics used for the classification such as accuracy, precision, recall, and F-measure were better than CNN and linear SVM. They noted to increase efficiency. HIDS is an IDS system that, to detect attacks, uses several log files on the local host machine to record device activities. The HIDS is typically based on host-environment measurements, such as computer system log files. These metrics or features are fed into the HIDS's decision engine as data. As a result, the foundation for any HIDS is feature extraction from the host environment [32].

2.2. Approaches of IDS. To decide whether or not an intrusion attempt has been made, IDS relies on a few approaches. The first is a signature-based approach, which compares current system data to a documented signature of an intrusion attack that is saved in the IDS database. When the IDS detects a match, it classifies it as an intrusion. This method allows for fast and precise detection. The signature database must be maintained regularly, which is a drawback. Also, the device may be hacked before the next intrusion attack is patched [13]. Moreover, it has other drawbacks such as network overloading, high signature matching prices, and a high number of false alarms. SIDS, which analyzes attacks that span several packets, are difficult to detect by using network packets and matching signatures against a signature database. With the complexity of modern malware, the signature extraction will be needed. The IDS would also need to carry the contents of previous packets [33].

The second method is anomaly based or behavior based, in which the IDS detects an intrusion when the device behaves abnormally. Both known and unknown threats can be detected using this tool. However, this technique has poor accuracy and a high rate of false alarms as disadvantages [14].

A hybrid approach, on the other hand, mixes signature and anomaly-based approaches. This system detects identified attacks using a signature-based approach and unknown attacks using an anomaly-based approach. Putting together the two methods can result in more precise detection, but they have the potential to be inefficient and raise computational costs [15]. This approach will help to resolve the limitations of a single process, thus improving the overall IoT system's reliability. The obvious disadvantage is that the entire IDS can grow in size and complexity. This will make operating the system more complex and will necessitate more resources. The intrusion detection method can consume a lot of resources and time, particularly if there are a lot of protocols in the IoT framework [34].

Traditional IDSs have several drawbacks, including the inability to distinguish new malicious threats, the need to be modified, poor accuracy, a high rate of false alarms, and the inability to detect zero-day attacks. Signature detection was used in reference [35] to detect attacks on Android phones by searching for unique patterns to detect intrusion and malicious activities. The device detects intrusions and automatically notifies the user of an unauthorized or malicious attempt as well as the intruder's location. To detect intrusions more flexibly and effectively, this approach was used to model and build the framework using actual intrusion features and processes. The most difficult challenge for signature-based NIDS is keeping up with large volumes of incoming traffic, as each packet must be matched to any signature in the database. As a result, handling all of the traffic takes a long time and slows down the system's throughput. SIDS strategies have become less successful as the number of zero-day attacks has increased [33], owing to the lack of a signature for all such attacks. In a simulated environment that generates synthetic data, Hasan et al. [36] contrasted the mechanisms for detecting anomalies in various machine learning techniques (ANN, RF, DT, SVM, and LR). However, this does not guarantee that RF can behave in this manner in case of big data and other unknown issues. As a result, further research will be needed. A standard pattern of data is created using data from regular users and then compared to current data patterns in real time for detecting anomalies [37]. An anomaly-based IDS detects deviations from behavior that is normal in the computing environment by building a normal behavior model in the computing system that is constantly updated based on data from normal users and that is used for detecting any variation from normal behavior [38].

IDSs are mostly classified based on their approach of working or deployment location. The former classifies it into signature based and anomaly based, whereas the latter classifies it into NIDS, HIDS, and hybrid. Based on the

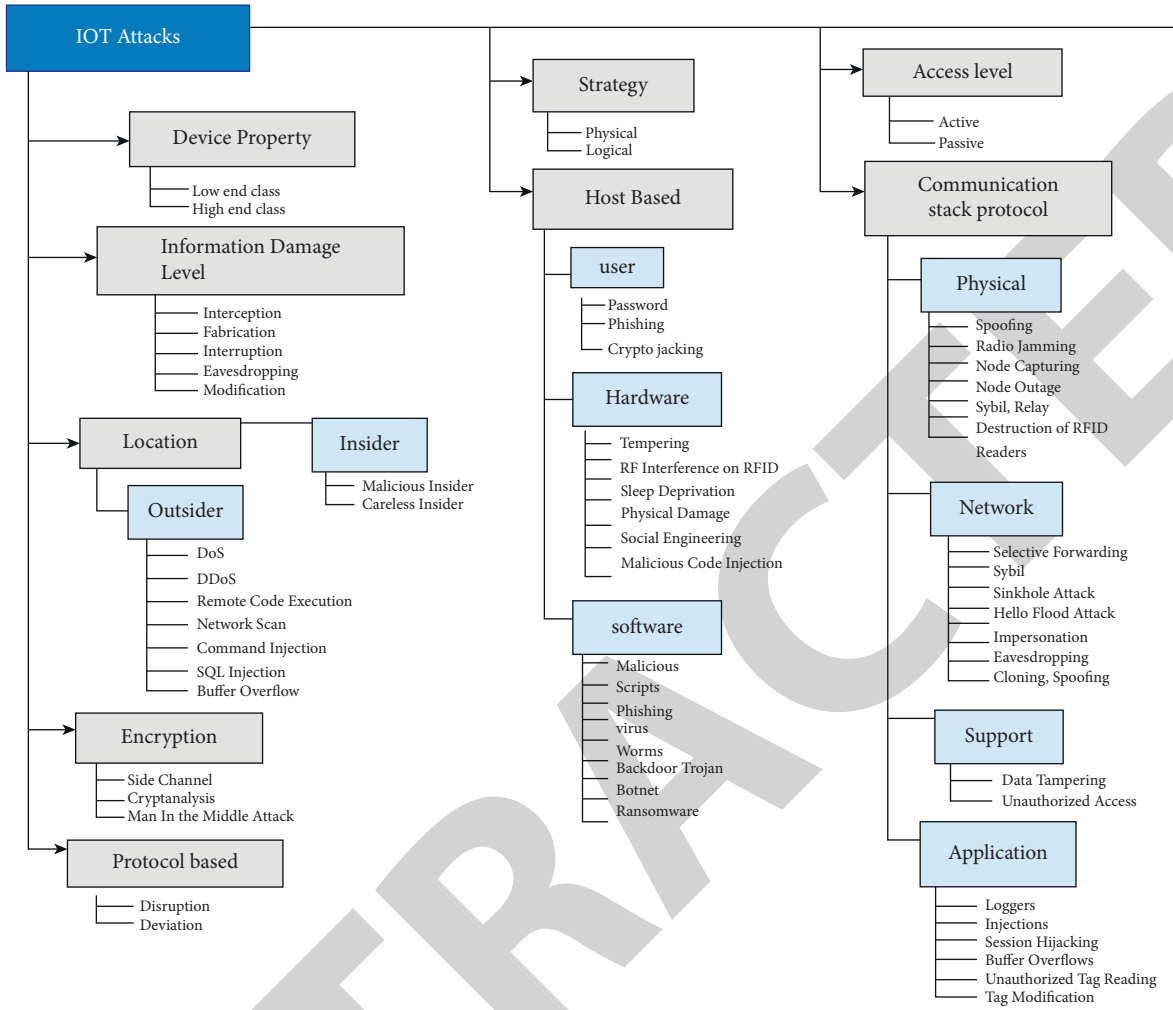


Figure 3: Multidimensional classification of IoT attacks.

literature review, we found many other classifications of IDS, and accordingly, we have prepared a unique categorization of IDS based on various attributes, as mentioned in Figure 4.

2.3. DBN and Its Utility in Securing IoT. Alom et al. also proposed a DBN-based IDS model. The DBN structure used in the model of IDS was not described in detail. The used data set was NSL KDD to test the IDS proposed. Using 40% of the training data, the authors recorded an accuracy of 97.5% [14].

Ding et al. looked at the use of DBN for malware detection. Pretraining was done in each layer for 30 epochs with RBM, and fine-tuning was done with the back-propagation algorithm and five-fold cross-validation training. A total of 3000 benevolent records and malicious 3000 records were used to test the classification results. The proposed model was tested with different data used for training (features) and an accuracy of 96.1% can be achieved with 400 features [39].

The DBN and probabilistic neural network (PNN) based on a hybrid anomaly detection model were presented by Zhao et al. Furthermore, the algorithm was used to increase the proposed model's efficiency, namely, particle swarm optimization. The proposed model of IDS with DBN was evaluated on dataset KDD cup99, where records together with four attack classes and normal were 10000 and were selected randomly for the testing. The proposed model can produce a false alarm rate, accuracy, and detection rate equal to 0.615%, 93.25%, and 99.14%, respectively [40].

Diro and Chilamkurti [41] introduced the DBN based on the distributed detection system. In the DBN, softmax is used as a categorization component that is comparable to prior DBN-based IDS. The distributed model outperformed the centralized model in the testing, with detection rate, accuracy, and false alarm rate of 99.27%, 99.20%, and 0.85% for the 2-class case and 96.5%, 98.27%, and 2.5% for the 4-class case, respectively.

A novel intrusion detection model focused on multiple DBNs and a fuzzy aggregation approach was presented by Yang et al. [42]. In addition, traffic data are clustered using the MDPCA to decrease the data's imbalanced state. MDPCA stands for modified density peak clustering algorithm. To train and evaluate the suggested model, NSL-KDD and UNSW-NB15 data sets are used. According to the results of the experiment, the suggested MDPCA-DBN

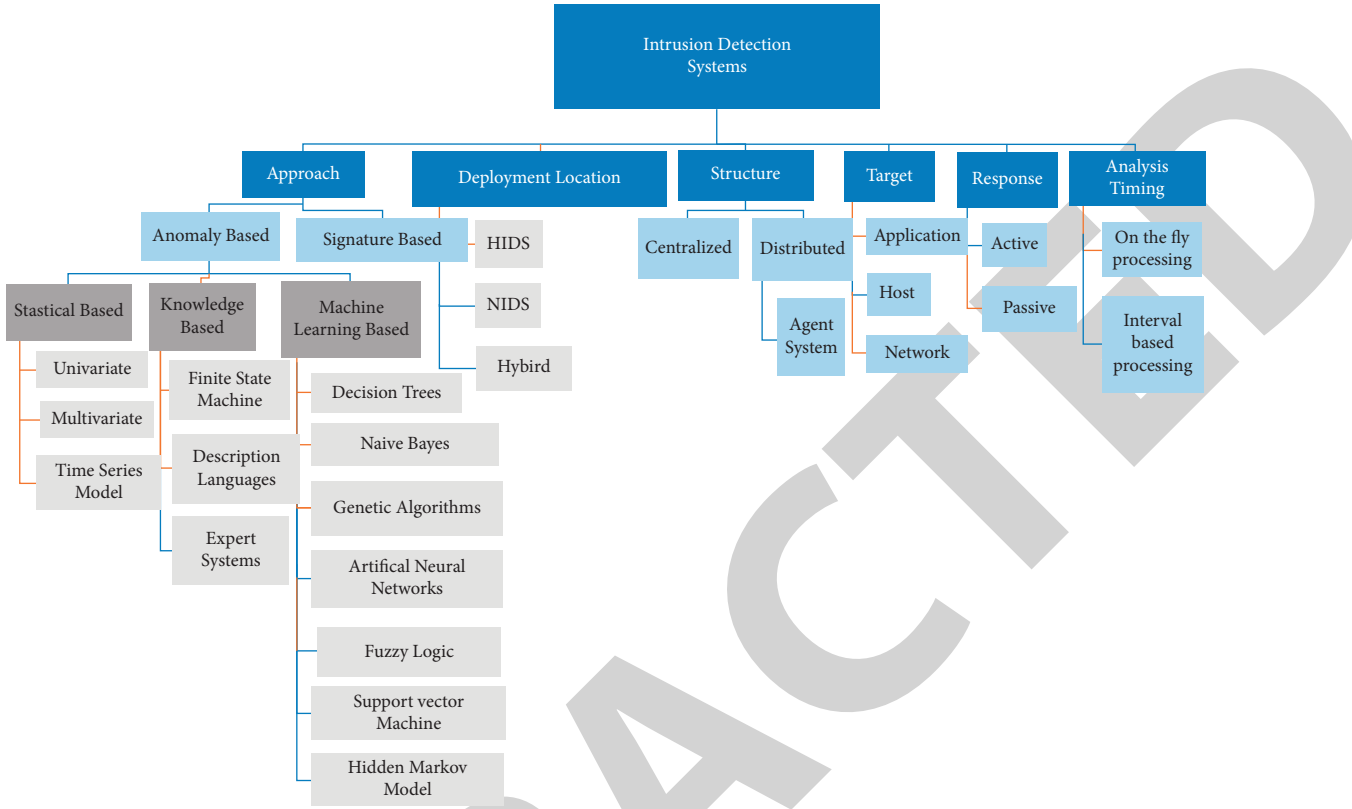


Figure 4: Different types of intrusion detection systems.

technique outperformed the DBN technique, with precision, accuracy, false-positive rate, and detection rate of 92.27%, 82.08%, 2.62%, and 70.51% for the NSL-KDD dataset and 87.3%, 90.21%,17.15%, and 96.22% for UNSW-NB15 dataset, respectively.

For wireless sensor networks, the authors in reference [43] implemented the DBN-based IDS architecture. The presented DBN was trained in the same way as prior DBN-based IDS models with three hidden layers, but in each hidden layer, the number of units was not specified. The presented IDS for WSN under attack was evaluated on a dataset, namely, KDD Cup 99. The authors reported a performance metric of 99.12% and a detection rate accuracy of 99.91% from the experiment.

Balakrishnan et al. [44] proposed the hybridization of IDS with DBN on real-time data. It was noticed that during the comparison the suggested algorithm needs to be strengthened by enhancing the dataset used for training. The attacks included are Dos attempt, overflow attempt, cache poisoning attempt, and malware infection, and the accuracy for this was 85%.

The IDS model by the authors in reference [45] uses DBN with a pretraining process based on a PSO algorithm. The proposed approach uses a two-stage PSO-based algorithm, and with the selection of features on the higher level and the lower level, there is hyperparameter selection. Using the NSL-KDD and CICIDS2017 datasets, the proposed IDS model's efficiency was evaluated. The binary classification for proposed IDS accuracy, precision, and recall for the NSL_KDD data set were 99.79%, 99.83%, and 99.81%, respectively, according to the experiment.

In a real-time network attack scenario, reference [46] implemented a model of IDS based on DBN. The DBN is divided into two sections: classification of attack and attack detection for real time. The multidimensional data are processed using a genetic algorithm, and the smallest number of features is chosen to enhance attack detection. The DBN classification of attack-based module processes data that have been defined as an attack for classification of the attack type. The dataset 2017CICIDS was used to assess the suggested IDS model's detection efficiency. Recall and precision of 97.67% and 97.74%, respectively, were recorded from the experiment results.

The intrusion detection system presented in reference [47] also uses the DBN algorithm concept. The performance of the current model of IDS is estimated using the CICIDS 2017 dataset. The experimental findings show that the suggested method achieves greater accuracy, recall, precision, F1-score, and detection rate than other methods. This approach has 97.93% accuracy in the normal class, 97.71% in the Botnet class, 96.67% in the Brute Force class, and 96.37% in the Dos/DDoS class.

Parul [48] proposed a neuro-fuzzy interference system that can predict software's reliability. This model was developed using a neuro-fuzzy tool and its results were more accurate when compared with the state-of-art soft computing techniques. This model helped the researchers to select the best software in terms of reliability.

An optimized energy-efficient secure routing protocol (OEESR) was proposed by Ripty [49] that is used in wireless body area networks, which minimizes the network congestion and provides security during the transmission of data. The results showed that OEESR is highly secure and efficient.

Thirumoorthy [50] proposed a multi-sensor data synchronization scheduling framework that is used in wireless sensor networks. This framework is secure and efficient for data aggregation in wireless sensor networks. The results proved that this framework increased the lifetime of the network and reduced the energy consumption by 51%.

3. Learning-Based Intrusion Detection for IoT

Learning-based methods ML/DL have come a long way in recent years, and artificial intelligence has gone from being a curiosity in the lab to being used in a variety of critical applications. The ability to intelligently track IoT devices offers major protection against new attacks also known as zero-day attacks. Effective data discovery methods are ML/DL for learning about "normal" and "abnormal" actions in the IoT environment based on how IoT components and devices work. As a result, ML and DL techniques are crucial in transforming IoT protection from merely enabling safe communication between devices and intelligence-based security systems. Having the ability to track IoT devices allows you to intelligently respond to new or zero-day threats. The input data from each component of the IoT system can be obtained and analyzed to determine normal patterns of interaction, allowing for the early detection of malicious actions. Furthermore, through learning from current instances, learning-based approaches may intelligently predict unknown attacks. These methods can be useful in predicting new attacks, which are often mutations of previous attacks. As a result, for successful and safe systems, IoT systems must move from simply forwarding secure communication among gadgets to security-based intelligence allowed by learning-based methods [16].

3.1. Machine Learning for Intrusion Detection in IoT. In 1959, Arthur Samuel coined the word "machine learning," defining it as an "area of research that allows computers to learn without being specifically programmed." It entails creating a model that depicts a specific action or attribute and then using that model to predict characteristics in both seen and unseen situations. The versatility, adaptability, and low CPU load of ML techniques will encourage us to create a variety of analytical models for attack and anomaly detection that are more accurate and have lower false alarm rates. Furthermore, knowledge of various ML methods is needed to assess their suitability for a variety of attacks and anomalies. Some of the benefits of using machine learning-based IDS instead of signature-based IDS are as follows:

(i) Signature-based IDS can be easily circumvented by making small changes to an attack sequence, while supervised machine learning-based IDS can identify attack variants as they learn the behavior of traffic flow.

(ii) Novel attacks can be detected by some ML-based IDS, especially those based on unsupervised learning algorithms.

(iii) Because ML-based IDS do not evaluate all signatures in the signature database such as signature-based IDS, they have a low-to-moderate CPU load.

Based on this approach, ML can be supervised, unsupervised, or semi-supervised.

3.1.1. Supervised Learning. It is a method of extracting features from a training dataset. The primary aim is to estimate the mapping function, such that the right output labels for the new data can be predicted. It is classified into classification and regression based on the nature of target labels. The technique is extremely useful for detecting faults and detecting intrusions based on misuse. For learning purposes, the dataset availability with signatures for documented attacks is a prerequisite for implementing supervised ML algorithms in IoT. For attack detection in IoT, various supervised learning methods such as KNN, decision tree, SVM, Naive-Bayes, and ANN are used.

3.1.2. Unsupervised Learning. Due to the lack of a labeled dataset, it is particularly useful for modeling the fundamental or hidden structure of data. The lack of availability of labeled dataset distinguishes it from the supervised approach, allowing for a more thorough analysis of the results. Clustering, dimensionality reduction, and density estimation are the three parts of the program. As a result, these methods are useful for identifying new anomalies and outliers. Furthermore, PCA and other dimensionality reduction techniques help to eliminate features that do not affect the class separability.

3.1.3. Reinforcement Learning. This method is concerned with the use of acceptable software agent behavior in a given environment to maximize the accumulated reward. It can also be referred to as "learning from the world" in a broader sense. Policy search and value function approximation are two of the most popular reinforcement learning techniques. Q-learning, TD-learning, and R-learning are the three main categories.

Based on the above literature survey, various authors analyze the performance evaluation of ML techniques, as listed in Table 2.

3.2. Deep Learning for Intrusion Detection in IoT. Deep learning is a successor to machine learning, capable of simulating the human brain and therefore falling under the category of artificial intelligence. Because of their multi-layered structure, deep networks can achieve higher precision in terms of predictions and classifications. When paired with IDS, DL networks can achieve superhuman efficiency in terms of detecting new attacks and anomalies. The main

advantage of this technology over ML is that manual feature selection is no longer necessary, and nonlinear relationships can now be modeled. Also, the ability to manage big data supports the use of technology in IoT. The nonlinearity activation function is crucial in achieving this aim. It was discovered that a deep-learning model could improve accuracy, allowing for the most successful mitigation of attacks on an IoT network. Several DL algorithms are used for the detection of intrusion in IoT, as shown in Table 3.

4. Proposed Intrusion Detection System Engine

In this section, the DBN_Classifier [63] is proposed and evaluated its performance on the TON_IOT_Weather dataset, which is a subset of TON_IOT Combined_IoT_Dataset. Figure 5 depicts the overall architecture for DBN-based IDS. DBN is an encouraging algorithm that uses the attack dataset/cases to train and make decisions. A Deep belief network (DBN) is a technique for stacking multiple unsupervised networks that use the hidden layer of each network as the input to the next layer. This is usually done with a stack of restricted Boltzmann machines (RBMs) or autoencoders. The ultimate aim is to develop a faster-unsupervised training protocol for each subnetwork that relies on contrastive divergence. DBN is a stochastic model made up of stacked restricted Boltzmann Machine (RBM) modules. The RBM is a model based on undirected energy with two layers of visible and hidden units, with only relations between layers. The contrastive divergence protocol is used to train each RBM module one at a time in an unsupervised manner.

In DBN, each stage's output (learned features) is fed into the next RBM stage as data. Later, the supervised learning is used to train the entire network to enhance classification accuracy (fine-tuning method) [64]. DBN consists of two steps: pretrain step and fine-tune step. The pretrain step is made up of several layers of RBN, while the fine-tuning step is made up of a feed-forward neural network [65]. During the training phase of DBN, the inputs are preprocessed, which retrieves the relevant primary data, according to the DBN architecture. The training phase entails feeding the network's experience, such as the specifics of the attack, to the network. The features are recognized and fed to the next hidden layer in various forms after the first input layer. The number of hidden layers varies by application and the default section can be customized to the intended usage before the start of the training. The third layer, like the second, gathers information for the learning process. Through classification, the target decision is mapped in the output layer. Because the output layer of the network is a binary decision network, logical 0 is mapped to the secure network (i.e., no intrusion), and logical 1 is for detection of intrusion. Before the actual performance evaluation of DBN, we have preprocessed the data, extracted the sample dataset, and split the training and testing set. The overall methodology is mentioned in Algorithm 1 and the same is depicted in Figure 6 as well.

The proposed DBN-based IDS performance evaluation algorithm is provided as Algorithm 1.

In this study, the authors have used a model and considered sample data of 30000 tuples (entries or rows) out of the TON_IOT_Weather dataset because of the unavailability of high computational power. The authors have worked on this sample dataset of the TON_IOT_Weather dataset for the performance evaluation of DBN_Classifier. To get an unbiased representational sample, the authors have shuffled the original dataset before extracting a sample from it. Afterward, the columns that contain string values are converted into numeric values using a label encoder. Also, the authors have normalized the sample data using MinMaxScaler with the range of 0 and 1. With 0 being the least value and 1 being the highest value. The normalization helps to train the model in the least time. Still, the training time will be very large for DBN_Classifier, as deep learning algorithms take a large time for training the model as compared to machine learning models. Then, we split the sample dataset into the training set and testing set in the ratio of 0.8 and 0.2, respectively. The model is generated by training DBN_Classifier on the training dataset. The performance of this model is then evaluated on the testing dataset. The structure of our DBN_Classifier is listed in Table 4. The authors have used two hidden layers of sizes 256 and 256, respectively. The number of epochs has been taken to 30 due to the limited computational power. These attributes are also known as hyperparameters.

To surmise the proposed engine, the DBN_Classifier model has been used for binary classification on the label attribute of the dataset, which contains two values, that is, 0 and 1, representing normal and attacks, respectively. Accordingly, for the same, the performance has been evaluated. The subclass attack identification (multiple classifications) is not considered as the sample data could not learn well to identify each subclass attack because of limited entries (i.e., 24k entries of training data). Also, the dataset contains a small number of attack entries for some individual subclass attacks such as Scanning and XSS, and as there are only 529 and 866 entries in the whole dataset, their contribution is very small in the sample dataset.

5. Datasets Considered

IoT datasets play a major role in developing IoT analytics. IoT datasets in the real world produce more data, which improves the accuracy of deep learning algorithms. The evaluation datasets are critical for the validation of any intrusion detection approach because they enable us to evaluate the proposed method's ability to detect intrusive conduct. Due to privacy concerns, datasets for analyzing network packets are not readily available. However, few datasets are freely accessible such as TON_IOT, being studied in this section.

5.1. TON_IOT Dataset. This dataset contains telemetry data from IoT systems, along with operating system logs and network data from an IoT system that was obtained from a practical depiction of a medium-scale network at the UNSW Canberra Cyber Range and IoT labs (Australia). It is a

Table 2: Machine learning approaches for IDS.

Author	Year	Dataset used	Algorithm	Challenge	Performance evaluation
Anthi et al. [51]	2018	Dataset developed by making a smart home testbed	NB	No clustering of homogeneous devices, limited attacks	Scan attack: PR: 97.7% RC: 97.7% F ms: 97.7% SYN: PR: 80.8% RC: 68.8% F ms: 65.8% UDP: PR: 8% RC: 68.8% F ms - 65.8%
Divyatmika et al. [52]	2016	NSL-KDD	Clustering + KNN data classification + MLP misuse detection + reinforcement anomaly detection	Considers only TCP/IP flows, not host based	ACC: 99.5% with false alarms that are reduced
Christiana et al. [53]	2019	Own testbed created	SVM	IDS deployment in high-energy gateway nodes	ACC: 100% and ACC: 81% when topology is changed
Pajouh et al. [54]	2019	NSL-KDD	PCA + LDA (feature selection), NB + CF-KNN classification	To perform anomaly and intrusion detection at the device and support layers, various network protocols are taken into account	ACC: Probe attack:87.32% DOS attack: 88.20% U2R: 70.15% R2L: 42% Overall detection rate: 84.86% FAR: 4.86
Shahid et al. [55]	2020	Dataset generated by creating the testbed	RF, DT, ANNKNN, and GNB (Gaussian naïve-Bayes)	Exploring unsupervised deep learning and integration of anomaly detection models with software defined	ACC: RF: 99.9% DT: 99.5% SVM: 99.3% KNN: 98.9% ANN: 98.6% GNB: 91.6%
Srinivasan et al. [56]	2019	Two random networks	RF	Testing different ML algorithms	ACC: 97%
Moustafa et al. [57]	2019	NIMS and UNSW-NB15	DT + NB + ANN ensemble model	Other IoT protocols are being considered, with a focus on further zero-day attacks	ACC with DNS data source is 99.54% ACC with HTTP data source is 98.97%
Zhao et al. [58]	2018	KDD cup 99	PCA (dimension reduction) + KNN (classification + softmax regression) = PCA + KNN + softmax regression (classification)		With three dimensions, ACC is 85.24%; with six dimensions, ACC is 85.19% With three dimensions, ACC is 84.999%, with six dimensions, ACC is 84.436%, and with ten dimensions, ACC is 84.406%
Hasan et al. [56]	2019	DS2OS	LR, SVM, ANN, RF, and DT	Robust algorithms are needed, system construction inspection is needed, and real-time attack detection requires more attention	ACC: LR: 98.3% SVM: 98.2% DT: 99.4% RF: 99.4% ANN: 99.4%

Table 3: Deep learning approaches for IDS.

Author	Year	Dataset	Algorithm	Challenges	Performance evaluation
Farhan et al. [59]	2019	Google code jam	DNN	Overfitting problem	ACC: 96%
Roopak et al. [60]	2019	CICIDS	MLP, one-dimensional CNN (convolutional neural network), LSTM, LSTM + CNN	There are not enough deep learning models that can handle highly unbalanced datasets	ACC: 1-d CNN: 95.14% MLP: 86.34% LSTM: 96.24% LSTM + CNN: 97.16%
Vigranesaram et al. [61]	2018	KDD	DNN-3	Lack of real-time dataset	ACC: 93%
Shone et al. [62]	2020	NSL-KDD, KDD cup 99	NDAE (nonsymmetric deep autoencoders)	Inadequate research due to a lack of real-time traffic	ACC: Dos: 94.58%, probe: 94.67%, R2L: 3.82%, U2R: 2.70%
Diro et al. [14]	2018	NSL-KDD	A deep learning model with three layers was used	Implementation of technique on different datasets	ACC: 96%–99% (for two classes such as normal and anomalous) ACC: 98.27% (for 4 classes such as normal, dos, probe, U2R, and R2L)
Alsaedi et al. [15]	2020	TON_IOT	LR, LDA, KNN, RF, CART, NB, SVM, and LSTM	On the proposed datasets, further work should be done to boost the efficiency of the baseline methods	ACC: LR: 0.61% LDA: 0.62% KNN: 0.72% RF: 0.71% CART: 0.77% NB: 0.54% SVM: 0.60% LSTM: 0.68%
Balakrishnan et al. [16]	2019	Real-time network traffic	DBN	It needs further improvement by using the dataset	For other attacks PR 86%, RC 74%, and F1-score 79%

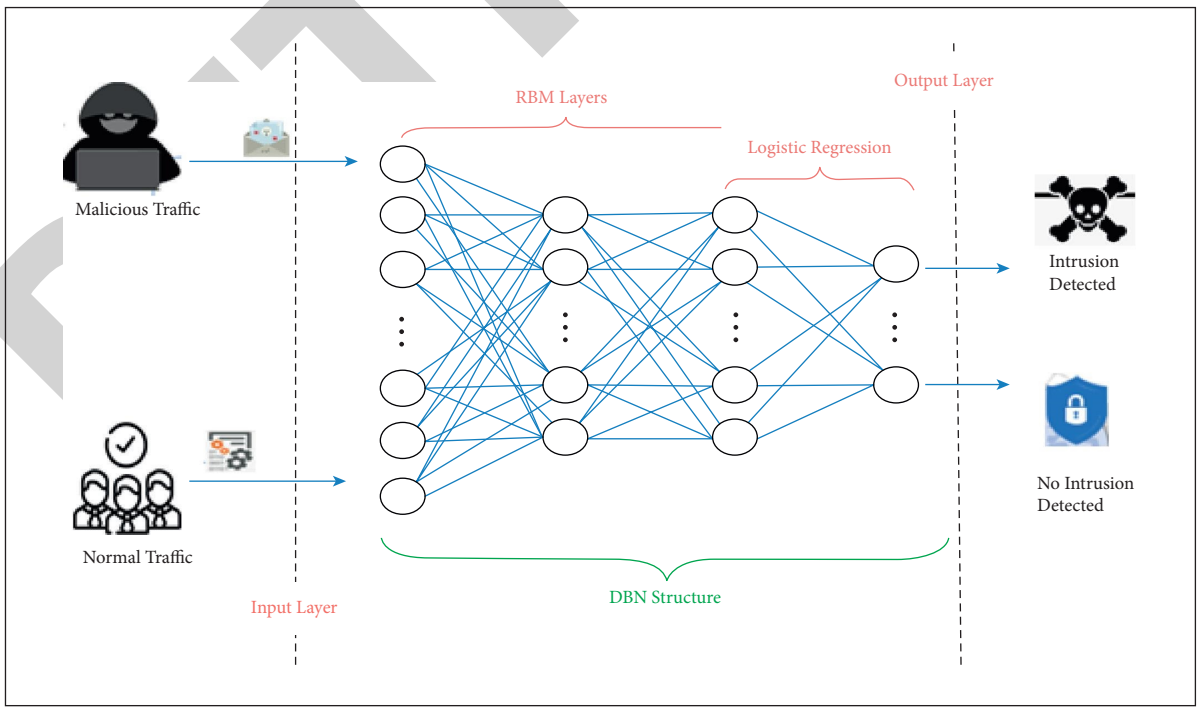


Figure 5: DBN-based intrusion detection system architecture.

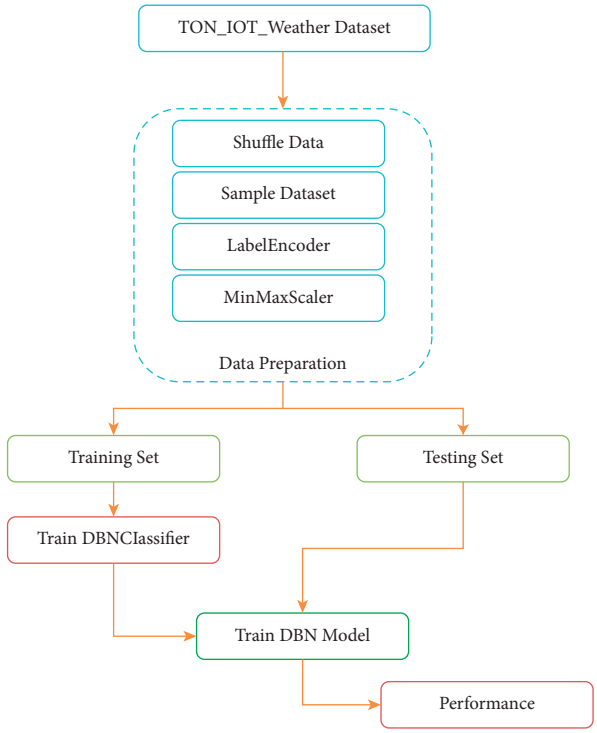


Figure 6: Proposed DBN-based IDS training and testing mechanism.

publicly accessible dataset at the ToN-IoT repository [66]. To collect their telemetry data, seven IoT sensors were used, including weather and modbus sensors. TON_IoT has various advantages as follows:

(i) It includes a variety of standard and attack events for various IoT/IIoT services.

(ii) It contains a nonuniform data source. Furthermore, for multi-class classification problems, the datasets presented were labeled with a marked characteristic, which indicates whether an examination is natural or attack and a feature type indicating subclasses of the attacks.

DoS, DDoS, scanning, ransomware, backdoor, code injection, cross-site scripting (XSS), Man-in-the-Middle (MITM), and password cracking are among the nine forms of cyber-attacks that were launched across the IoT network against various IoT sensors [67]. The data generated from sensors were kept in CSV files. Processed datasets and train test datasets are the two key directories for IoT datasets. The processed datasets folder includes a processed and filtered category of the datasets in CSV format, along with their regular features and labels. For train test, dataset samples are used in a CSV format in the "train test datasets" folder for testing the accuracy and effectiveness of deep-learning models. Seven Train-Test IoT datasets are available, one for each of the IoT devices: refrigerator, GPS tracker, motion light, garage door, modbus, thermostat, and weather. All IoT datasets were merged into a single combined_IoT _dataset CSV format. To merge all IoT datasets automatically to one CSV file having 22 features in total, a python script was implemented. Table 5 depicts 22 features and a description of combined_IoT_dataset. Figure 7 illustrates train test data for the combined_IoT_dataset.

The Combined_IoT_dataset contains 22 attributes (features). These features are combined from 7 datasets namely fridge sensor, garage door, GPS sensor, modbus, light/motion, thermostat, and weather. Each of these datasets contains a set of common attributes namely ts, date, time, label, and type. In Combined_IoT_dataset, these attributes are used only once [67].

In this study, the authors have evaluated the performance of the deep belief network on the weather dataset. This dataset contains 8 attributes: ts, date, time, temperature, pressure, humidity, label, and type. Their description is mentioned in Table 5. The statistical features of the weather dataset are enumerated in Figure 8. TON_IOT_Weather dataset contains 650242 entries, of which 559718 are normal entries and the rest, that is, 90524 are attack entries. The different attack categories considered in this dataset are password, scanning, XSS, DDOS, backdoor, ransomware, and injection. Scanning and XSS contain a very small number of entries, that is, 529 and 866, respectively. This small number of entries for a particular attack category can hinder machine learning or deep learning models to learn its detection with high accuracy.

6. Results and Discussion

We have evaluated the performance of the DBN_Classifier on the TON_IOT_Weather sample dataset of size 30k entries. In the sample dataset, 24k entries are used for training and 6k entries for testing purposes. We have used a hard-coded value of 30 epochs and 10 backpropagation iterations for training our model. We have executed the model in the HP system containing Windows 8.1 OS, 8 GB RAM, 600 GB hard disk, and processor specifications as Intel(R) Core(TM) i3-4010U CPU @ 1.70GHZ. The parameters considered for performance evaluation are accuracy, precision, recall, and F1-score. Given confusion metrics, with values true positive (TP), false positive (FP), true negative (TN), and false negative (FN), the performance parameters can be calculated.

Accuracy is the most widely accepted performance measure and is a ratio of correctly predicted observations to the total observations:

$$\textbf{Accuracy} = \frac{\textbf{TP} + \textbf{TN}}{\textbf{TP} + \textbf{TN} + \textbf{FP} + \textbf{FN}}. \quad (1)$$

The precision is the ratio of miss and false hit rates, or we can say a ratio of correctly predicted positive observations to the total predicted positive observations. It is calculated as follows:

$$\textbf{Precision} = \frac{\textbf{TP}}{\textbf{TP} + \textbf{FP}}. \quad (2)$$

The recall is also known as sensitivity or true positive rate (TPR). It is the ratio of correctly predicted positive

```
Input: TON_IOT_Weather dataset
Initialization: Hidden_Layer_Structure = 256,256, RBM_Learning_Rate = 0.05, Learning_Rate = 0.1, Epochs = 30,
Fine_Tuning_Iterations = 10, Batch_Size = 32, Activation_Function = "Relu", Dropout = 0.2;
Output: DBN_Classifier Performance on TON_IOT_Weather Dataset
(1) Start
(2)    Data ← TON_IOT_Weather
(3)    Data ← Shuffle_Data
(4)    Sample_Data ← Extract Sample of 30k entries from Data
(5)    Sample_Data ← Apply LabelEncoder on String Columns of Sample_Data
(6)    Sample_Data ← Apply MinMaxScaler on Sample_Data to scale values between 0 and 1
(7)    Train_Data, Test_Data ← Split Sample_Data
(8)    Train DBN_Classifier on Train_Data
(9)    Test DBN_Classifier on Test_Data
(10)   Print Performance
(11) End
```

Algorithm 1: Proposed DBN based IDS performance evaluation algorithm.

Table 4: DBN_Classifier structure and hyperparameters.

DBN attribute	Value
Hidden layer structure	256, 256
Learning rate for RBM	0.05
Learning rate	0.1
Number of epochs	30
Backpropagation iterations	10
Batch size	32
Activation function	Relu
Dropout	0.2

Table 5: Features of combined TON_IOT dataset.

Feature	Description
Ts	Sensor reading data timestamp
Data	Telemetry data from logging sensors
Time	Sensor telemetry data logging time
fridge_temperature	A fridge sensor's temperature is measured.
temp_condition	Based on whether the temperature is high or low, conditions of a temperature of a fridge sensor are based on a predefined threshold value
Label	Determines normal and attack records, where 0 specifies normal and 1 specifies attack
Type	A tag that can be used for standard or attack subgroups, for example, DoS, DDoS, and backdoor attacks
door_state	Specify whether a door is open or closed
sphone_signal	Specify receiving door signal state on a phone, where a signal can either be true or false
Latitude	GPS tracking sensor's latitude value
Longitude	GPS tracking sensor's longitude value
motion_status	Indicates motion sensor's status, that is, on or off. 1 specifies on and 0 specifies off
light_status	Indicates the status of light sensor whether on or off
FC1_Read_Input_Register	A modbus function code whose responsibility is to read an input register
FC2_Read_Discreate_Value	A modbus function code whose responsibility is to read discrete values.
FC3_Read_Holding_Register	A modbus function code whose responsibility is to read a holding register
FC4_Read_Coil	A modbus function code whose responsibility is to read a coil
current_temperature	Specify thermostat sensor's current temperature
thermostat_status	Thermostat sensor's status (on/off)
Temperature	Weather sensor's temperature readings
Pressure	Weather sensor's pressure measurements
Humidity	Weather sensor's humidity measurements

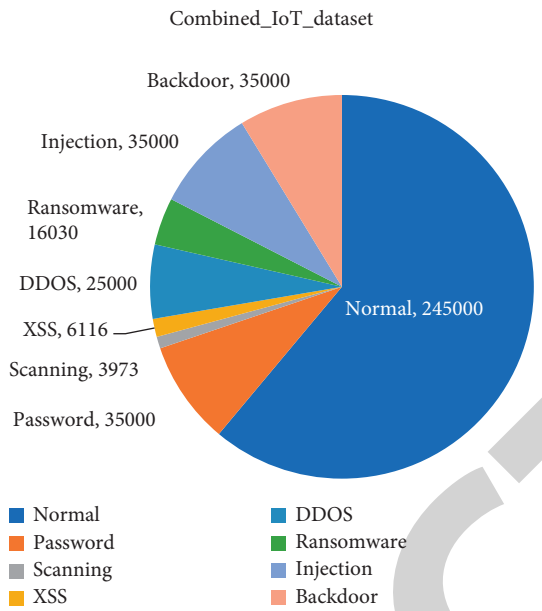


Figure 7: Statistics of combined TON_IOT dataset.

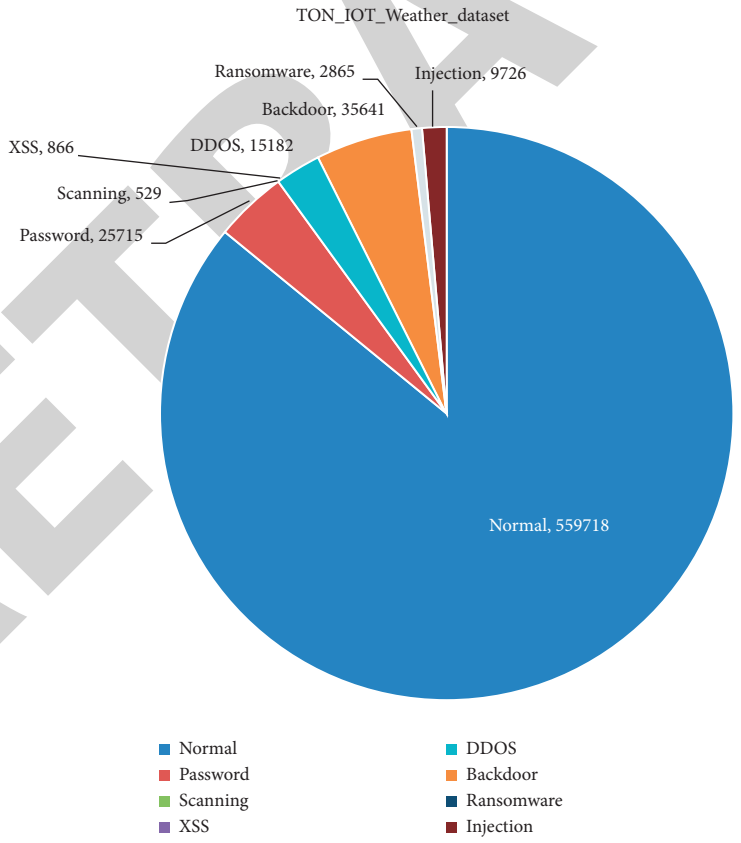


Figure 8: Statistics of TON_IOT_Weather dataset.

Table 6: DBN_Classifier performance on TON_IOT_Weather sample dataset.

Metric	Precision	Recall	F1-score accuracy
Average	0.78	0.90	0.84 86.3

observations to the sum of correctly predicted positives and incorrectly predicted negatives. It is calculated as follows:

$$\textbf{Recall} = \frac{\textbf{TP}}{\textbf{TP} + \textbf{FN}}. \tag{3}$$

The F1-score defines the balance between precision and recall and is calculated as follows:

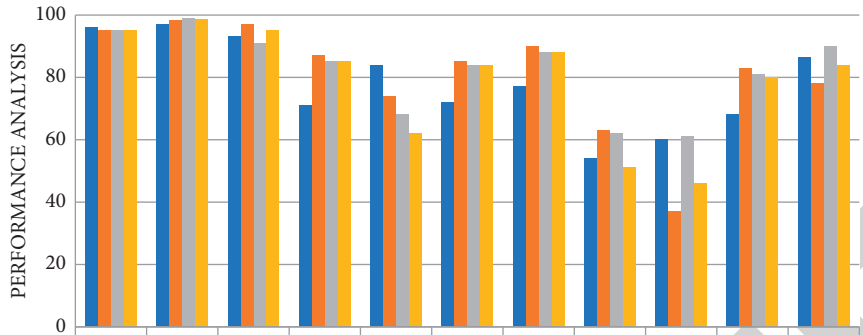
$$\textbf{F1 Score} = \frac{2 * \textbf{Precision} * \textbf{Recall}}{\textbf{Precision} + \textbf{Recall}}. \tag{4}$$

In our methodology, we have got an accuracy of 86.33% for classifying whether the entry is an attack or normal. The other performance metrics evaluated are given in Table 6.

The model is evaluated on a sample dataset and with a small number of epochs, keeping in view the limited computational power available. The performance can be enhanced by increasing the number of epochs and utilizing the effective hyperparameter evaluation mechanism. This not only increases the performance but also helps in reducing the training time. The authors have passed hardcoded hyperparameters to evaluate DBN performance on the TON_IOT_Weather dataset.

The Intrusion detection system (IDS) faces certain challenges such as identification of attack source, effectiveness in voluminous network flows, and response against attacks. Also, the traditional IDS mechanisms failed to maintain security against unknown attacks. Considering these challenges, the intelligent security methods based on ML and DL invoke greater effectiveness than the traditional approaches. Our approach is an intelligent approach to ensure security but needs performance assessment on unknown attacks as well through mechanisms such as transfer learning (domain adoption) that will be considered in our future research. We have checked the effectiveness of our model in the IoT-based TON dataset and it has shown an accuracy of 86.3% and an F1-Score of 84%. Our achieved performance needs further enhancement to be effective for IoT systems. The dataset and/or hardcoded model structure can be reasons for low performance. This can be achieved by considering mechanisms such as hyperparameter optimization, parameter optimization, and feature engineering. Hyperparameters are variables whose values influence the learning process and affect the model parameters that a learning algorithm learns. Hyperparameters are significant because they directly regulate the training algorithm's behavior and have a major impact on the model's performance. Hyperparameter tuning optimizes a model for the metric we choose given a set of input features (hyperparameters). To address a regression problem, hyperparameter tuning makes educated judgments about which hyperparameter combinations are most likely to produce the best results and then conducts training tasks to verify these assumptions. The use of parameter values that are optimal are recommended whenever the objective function is minimized for a specific dataset. Weights and biases are the parameters of the network. The settings of the parameters define how accurately the model executes the task for a specific architecture. We look for good values by defining a loss function to assess the model's performance. The goal is to reduce the loss as much as possible and so obtain parameter values that correspond to reality. The act of selecting, altering, and transforming raw data into features that can be used in supervised learning is known as feature engineering. In simple terms, feature engineering is the process of transforming raw observations into desired characteristics through statistical or machine learning methods. Some of its techniques are log transform, scaling, etc. A successful feature engineering process results in a more efficient model. Algorithms that are easier to use and fit the data. Algorithms will have an easier time detecting patterns in the data. Intelligent models are also vulnerable to adversarial attacks, for example, the data flow can be perturbated/modified to evade detection and model parameters can be changed. Hence, there is a need to deal with adversaries either by considering techniques such as adversarial learning. The proposed model has been proposed for IoT systems and tested on the IoT-based TON dataset. This proposed approach is a general IDS approach, which can be used in any system/scenario such as IT system and cyber-physical system but needs evaluation prior to its utilization on the respective traffic flows or real-time scenarios. The different algorithms such as DNN, LSTM + CNN, DNN3, RF, LDA, KNN, CART, NB, SVM, and LSTM from existing works are compared with our DBN model in terms of accuracy, precision, recall, and F1-score, as shown in Figure 9.

Our proposed model DBN (TON) got an accuracy of 86.3%, which is better than the accuracy achieved by RF (TON) [36], LDA (TON) [36], KNN (TON) [36], CART (TON) [36], NB (TON) [36], SVM (TON) [36], and LSTM (TON) [36], but some of these existing models were better in terms of precision and F1-score as compared to our work. DNN (Google code jam) [31], LSTM + CNN(CICIDS) [32], and DNN3 (KDD) [33] have achieved better performance and we will also intend to enhance our performance by utilizing model optimization and feature engineering techniques.



	DNN (Google code jam) [31]	LSTM +CNN (CICIDS)[32]	DNN3 (KDD) [33]	RF (TON) [36]	LDA (TON) [36]	KNN (TON) [36]	CART (TON) [36]	NB (TON) [36]	SVM (TON) [36]	LSTM (TON) [36]	DBN (TON)
Accuracy	96	97	93	71	84	72	77	54	60	68	86.33
Precision	95.16	98.44	97	87	74	85	90	63	37	83	78
Recall	95.1	99.1	91	85	68	84	88	62	61	81	90
F1 score	95.13	98.76	95	85	62	84	88	51	46	80	84

Figure 9: Performance comparison with existing work.

7. Conclusion

The mass production of insecure IoT devices incrementing the smart services exponentially poses a high risk to the widely spread smart systems. Although, a vast amount of literature is available in respect to securing IoT systems with several security standards proposed by security boards and regulation groups. The existing IDS engines with lesser accuracy and higher false alarm rates bait the research community to develop more reliable engines with higher deployment rates. For this, the authors have proposed an intrusion detection engine based on a learning model, DBN_Classifier, and implemented using TensorFlow. The learning model is trained on the subset of the TON_IOT_Weather dataset. A representational subset is extracted by shuffling the original TON_IOT_Weather dataset. The results claim that the proposed system outperforms the existing models in terms of accuracy, precision, recall, and F1-score.

A future leeway to the presented intrusion detection engine in this work is the addendum with real-time dataset and complete TON dataset. Also, the available hyperparameters evaluation mechanisms will be studied in-depth and worked upon to improve the efficiency of the proposed engine. [68].

Data Availability

No new data are generated in this manuscript.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

All authors contributed equally to this manuscript.

References

[1] B. B. Gupta and M. Quamara, "An overview of Internet of Things (IoT): architectural aspects, challenges, and protocols," *Concurr. Comput.*vol. 32, no. 21, pp. 1–24, 2020.

[2] P. V Dudhe, N. V Kadam, R. M. Hushangabade, and M. S. Deshmukh, "Applications," in *Proceedings of the 2017 Interational Conference Energy Communication Data Analysi Soft Computing*, pp. 2650–2653, Chennai, India, August 2017.

[3] "softbank-son-iot-1000-devices-2040@ www.techinasia.-com," May 2021, https://www.techinasia.com/softbank-son-iot-1000-devices-2040.

[4] M. C. Domingo, "An overview of the Internet of Things for people with disabilities," *Journal of Network and Computer Applications*, vol. 35, no. 2, pp. 584–596, 2012.

[5] I. Gudymenko and M. Hutter, "Security in the Internet of Things Supervisor," in *Proceedings of the Intensive Program Information Communication Security*, pp. 1–7, Itt, Kolkata, India, December 2011.

[6] P. Anand, Y. Singh, A. Selwal, M. Alazab, S. Tanwar, and N. Kumar, "IoT vulnerability assessment for sustainable computing: threats, current solutions, and open challenges," *IEEE Access*, vol. 8, pp. 168825–168853, 2020.

[7] S. Naik and V. Maral, "Cyber security - IoT," in *Proceedings of the RTEICT 2017 - 2nd PIEEE International Conference Recent Trends Electronics Information Communication Technology Proc*, pp. 764–767, Bangalore india, May 2017.

[8] "att-cybersecurity-insights-vol-5-the-ceos-guide-to-data-security @," 2021, http://www.business.att.comhttps://www.business.att.com/learn/research-reports/att-cybersecurity-insights-vol-5-the-ceos-guide-to-data-security.html.

[9] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "DDoS in the IoT: Mirai and other botnets," *Computer*, vol. 50, no. 7, pp. 80–84, 2017.

[10] S. Tanwar, S. Tyagi, and S. Kumar, "The role of internet of things and smart grid for the development of a smart city," *Intelligent Communication and Computational Technologies*, vol. 19, pp. 23–33, 2018.

[11] "ddos-reflection-attack-memcached-udp @ www.akamai.-com," 2021, https://www.akamai.com/us/en/resources/our-thinking/threat-advisories/ddos-reflection-attack-memcached-udp.jsp.

[12] "Number of compromised data records in selected data breaches as of January 2021," 2021, https://www.statista.com/statistics/290525/cyber-crime-biggest-online-data-breaches-worldwide/.

[13] 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, Article ID 105124, 2020.

[14] M. Z. Alom, V. Bontupalli, and T. M. Taha, "Intrusion detection using deep belief networks," in *Proceedings of the IEEE National. Aerospace Electronics Conference NAECON*, pp. 339–344, Dayton, OH, USA, 2016-March.

[15] A. Khraisat, I. Gondal, P. Vamplew, J. Kamruzzaman, and A. Alazab, "Hybrid intrusion detection system based on the stacking ensemble of C5 decision tree classifier and one class support vector machine," *Electron*, vol. 9, no. 1, 2020.

[16] M. A. Al-Garadi, A. Mohamed, A. K. Al-Ali, X. Du, I. Ali, and M. Guizani, "A survey of machine and deep learning methods for internet of things (IoT) security," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1646–1685, 2020.

[17] M. Martfnez-Arroyo and L. E. Sucar, "Learning an optimal naive Bayes classifier," in *Proceedings of the International Conference Pattern Recognition*, vol. 3, pp. 1236–1239, Hong Kong, China, August 2006.

[18] U. Noor, Z. Anwar, T. Amjad, and K.-K. R. Choo, "A machine learning-based FinTech cyber threat attribution framework using high-level indicators of compromise," *Future Generation Computer Systems*, vol. 96, pp. 227–242, 2019.

[19] J. R. Quinlan, *Induction of Decision Trees*, Springer link, New York, NY, USA, pp. 81–106, 2007.

[20] S. Kaplantzis, A. Shilton, and N. Mani, "Detecting Selective Forwarding Attacks in Wireless Sensor Networks Using Support Vector Machines," in *Proceedings of the 2007 3rd International Conference on Intelligent Sensors, Sensor Networks and Information*, pp. 335–340, MelbourneAustralia, December 2007.

[21] B. Yegnanarayana, *Artificial Neural Networks,* PHI Learning Pvt. Ltd., 2009.

[22] M. U. Farooq, M. Waseem, S. Mazhar, A. Khairi, and T. Kamal, "A review on internet of things (IoT)," *International Journal of Computer Application*, vol. 113, no. 1, pp. 1–7, 2015.

[23] S. Rizvi, R. Orr, A. Cox, P. Ashokkumar, and M. R. Rizvi, "Identifying the attack surface for IoT network," *Internet of Things*, vol. 9, Article ID 100162, 2020.

[24] H. F. Atlam, A. Alenezi, M. O. Alassafi, A. A. Alshdadi, and G. B. Wills, "Security, cybercrime and digital forensics for IOT," *Intelligent Systems Reference Library*, vol. 174, pp. 551–577, 2019.

[25] J. Deogirikar, *Security Attacks inIoT : A Survey*, in *Proceedings of the International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC 2017)*, 2017.

[26] M. Nawir, A. Amir, N. Yaakob, O. B. Lynn, and C. Engineering, "2014 2nd International Conference on Electronic Design ICED 2014," in *Proceedings of the 2014 2nd International Conference Electronics Designs ICED*, p. 542p, Penang, Malaysia, August 2014.

[27] M. U. Chowdhury, R. Doss, B. Ray, S. Rajasegarar, and S. Chowdhury, "IoT insider attack - survey," *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, vol. 324, pp. 28–41, 2020.

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

[29] M. Almseidin, M. Alzubi, S. Kovacs, and M. Alkasassbeh, "Evaluation of machine learning algorithms for intrusion detection system," in *Proceedings of the SISY 2017 - IEEE 15th International Symposium Intelligent System Informatics*, pp. 277–282, Subotica, Serbia, September 2017.

[30] O. Al-Jarrah and A. Arafat, "Network intrusion detection system using attack behavior classification," in *Proceedings of the 2014 5th International Conference Information Communication System ICICS*, Irbid, Jordanac, April 2014.

[31] M. Lopez-Martin, B. Carro, A. Sanchez-Esguevillas, and J. Lloret, "Conditional variational autoencoder for prediction and feature recovery applied to intrusion detection in iot," *Sensors*, vol. 17, no. 9, 2017.

[32] S. K. Gautam and H. Om, "Computational neural network regression model for Host based Intrusion Detection System," *Perspectives in Science*, vol. 8, pp. 93–95, 2016.

[33] A. Khraisat and A. Alazab, "A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges," *Cybersecurity*, vol. 4, no. 1, 2021.

[34] C. Liang, B. Shanmugam, S. Azam et al., "Intrusion detection system for the internet of things based on blockchain and multi-agent systems," *Electron*, vol. 9, no. 7, pp. 1–27, 2020.

[35] O. O. Cyril, T. Elmissaoui, M. C. Okoronkwo, M. Ihedioha Uchechi, C. H. Ugwuishiwu, and O. B. Onyebuchi, "Signature based network intrusion detection system using feature selection on android," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 6, pp. 551–558, 2020.

[36] 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, Article ID 100059, 2019.

[37] P. Mishra, E. S. Pilli, V. Varadharajan, and U. Tupakula, "Intrusion detection techniques in cloud environment: a survey," *Journal of Network and Computer Applications*, vol. 77, pp. 18–47, 2017.

[38] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network anomaly detection: methods, systems and tools," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 303–336, 2014.

[39] Y. Ding, S. Chen, and J. Xu, "Application of Deep Belief Networks for opcode based malware detection," in *Proceedings of the International Jt. Conference Neural Networks*, pp. 3901–3908, Vancouver, Canada, July 2016.

[40] G. Zhao, C. Zhang, and L. Zheng, "Intrusion detection using deep belief network and probabilistic neural network," in *Proceedings of the 2017 IEEE International Conference Computer Science Engineering IEEE/IFIP Internationl Conference Embeded Ubiquitous Computing CSE EUC*, vol. 1, pp. 639–642, Guangzhou, China, July 2017.

[41] 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.

[42] Y. Yang, K. Zheng, C. Wu, X. Niu, and Y. Yang, "Building an effective intrusion detection system using the modified density peak clustering algorithm and deep belief networks," *Applied Sciences*, vol. 9, no. 2, 2019.

[43] S. Otoum, B. Kantarci, and H. T. Mouftah, "On the feasibility of deep learning in sensor network intrusion detection," *IEEE Networking Letters*, vol. 1, no. 2, pp. 68–71, 2019.

[44] 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*, vol. 14, Article ID 100112, 2019.

[45] W. Elmasry, A. Akbulut, and A. H. Zaim, "Evolving deep learning architectures for network intrusion detection using a double PSO metaheuristic," *Computer Networks*, vol. 168, 2020.

[46] H. Zhang, Y. Li, Z. Lv, A. K. Sangaiah, and T. Huang, "A real-Time and ubiquitous network attack detection based on deep belief network and support vector machine," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 3, pp. 790–799, 2020.

[47] S. Manimurugan, S. Al-Mutairi, M. M. Aborokbah, N. Chilamkurti, S. Ganesan, and R. Patan, "Effective attack detection in internet of medical things smart environment using a deep belief neural network," *IEEE Access*, vol. 8, pp. 77396–77404, 2020.

[48] P. Gandhi, M. Z. Khan, R. K. Sharma, O. H. Alhazmi, S. Bhatia, and C. Chakraborty, "Software Reliability Assessment Using Hybrid Neuro-Fuzzy Model," *Computer Systems Science & Engineering*, vol. 41, no. 3.

[49] R. Singla, N. Kaur, D. Koundal, S. A. Lashari, S. Bhatia, and M. K. Imam Rahmani, "Optimized energy efficient secure routing protocol for wireless body area network," *IEEE Access*, vol. 9, pp. 116745–116759, 2021.

[50] T. Palanisamy, D. Alghazzawi, S. Bhatia, A. Abbas Malibari, P. Dadheech, and S. Sengan, "Improved energy based multi-sensor object detection in wireless sensor networks," *Intelligent Automation & Soft Computing*, vol. 33, no. 1, pp. 227–244, 2022.

[51] E. Anthi, L. Williams, and P. Burnap, "Pulse: An Adaptive Intrusion Detection for the Internet of Things," in *Proceedings of the Living in the Internet of Things: Cybersecurity of the IoT*, pp. 4–6, London, June 2018.

[52] Divyatmika and M. Sreekesh, "A two-tier network based intrusion detection system architecture using machine learning approach," in *Proceedings of the International Conference Electrical Electronics Optimaization Technology ICEEOT*, pp. 42–47, Chennai, India, March 2016.

[53] C. Ioannou and V. Vassiliou, "Classifying security attacks in IoT networks using supervised learning," in *Proceedings of the 15th Annual International Conference Distriuting Computer Sense System DCOSS*, pp. 652–658, Santorini, Greece, May 2019.

[54] H. H. Pajouh, R. Javidan, R. Khayami, A. Dehghantanha, and K.-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.

[55] M. Shahid, G. Blanc, Z. Zhang, and H. Debar, "Machine learning for IoT network monitoring," in *Proceedings of the RESSI 2019: Meeting of Information Systems Security Research and Education*, pp. 1–3, Seoul, Republic of Korea, July 2019.

[56] S. M. Srinivasan, T. Truong-Huu, and M. Gurusamy, "Machine learning-based link fault identification and localization in complex networks," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6556–6566, 2019.

[57] N. Moustafa, B. Turnbull, and K.-K. R. Choo, "An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4815–4830, 2019.

[58] S. Zhao, W. Li, T. Zia, and A. Y. Zomaya, "A dimension reduction model and classifier for anomaly-based intrusion detection in internet of things," in *Proceedings of the 2017 IEEE 15th International Conference Dependable, Auton. Secure Computing*, pp. 836–843, Orlando, FL, USA, November 2017.

[59] F. Ullah, H. Naeem, S. Jabbar et al., "Cyber security threats detection in internet of things using deep learning approach," *IEEE Access*, vol. 7, pp. 124379–124389, 2019.

[60] M. Roopak, G. Yun Tian, and J. Chambers, "Deep learning models for cyber security in IoT networks," in *Proceedings of the 2019 IEEE 9th Annual Computing Communication Work. Conference CCWC*, pp. 452–457, Vegas, NV, USA, January 2019.

[61] V. K. Rahul, R. Vinayakumar, K. Soman, and P. Poornachandran, "Evaluating shallow and deep neural networks for network intrusion detection systems in cyber security," in *Proceedings of the 2018 9th International Conference Computing Communication Network Technology ICCCNT*, Bengaluru, India, July 2018.

[62] S. Moraboena, G. Ketepalli, and P. Ragam, "A deep learning approach to network intrusion detection using deep autoencoder," *Revue d'Intelligence Artificielle*, vol. 34, no. 4, pp. 457–463, 2020.

[63] A. Albertbup, "Python Implementation of Deep Belief Networks Built upon NumPy and TensorFlow with Scikit-Learn Compatibility," 2021, https://github.com/albertbup/deep-belief-network.

[64] Deep belief network, https://www.sciencedirect.com/topics/engineering/deep-belief-network, 2021.

[65] Deep belief networks an introduction, https://medium.com/black-feathers-labs/deep-belief-networks-an-introduction-1d52bb867a25, 2021.

[66] TON_IoT datasets, https://cloudstor.aarnet.edu.au/plus/s/ds5zW91vdgjEj9i, 2021.

[67] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and A. Anwar, "TON_IoT telemetry dataset: a new generation dataset of IoT and IIoT for data-driven intrusion detection systems," *IEEE Access*, vol. 8, pp. 165130–165150, 2020.

[68] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 41–50, 2018.