IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 10, No. 3, September 2021, pp. 735~742
ISSN: 2252-8938, DOI: 10.11591/ijai.v10.i3.pp735-742

Intrusion detection with deep learning on internet of things heterogeneous network

Sharipuddin[1], Benni Purnama[2], Kurniabudi[3], Eko Arip Winanto[4], Deris Stiawan[5], Darmawijoyo Hanapi[6], Mohd. Yazid Idris[7], Rahmat Budiarto[8]
[1,2,3,4,5,6]Department of Computer Science, Universitas Sriwijaya, Indonesia
[1,2,3]Computer Enginering, Universitas Dinamika Bangsa, Jambi Indonesia
[4,7]School of Computing, Universiti Teknologi Malaysia, Malaysia
[8]College of Computer Science & IT, Albaha University, Saudi Arabia

Article Info

Article history:

Received Mar 3, 2020
Revised May 10, 2021
Accepted May 22, 2021

Keywords:

Deep learning
Features extraction
Heterogeneous
Intrusion detection system
Principal component analysis

ABSTRACT

The difficulty of the intrusion detection system in heterogeneous networks is significantly affected by devices, protocols, and services, thus the network becomes complex and difficult to identify. Deep learning is one algorithm that can classify data with high accuracy. In this research, we proposed deep learning to intrusion detection system identification methods in heterogeneous networks to increase detection accuracy. In this paper, we provide an overview of the proposed algorithm, with an initial experiment of denial of services (DoS) attacks and results. The results of the evaluation showed that deep learning can improve detection accuracy in the heterogeneous internet of things (IoT).

Corresponding Author:

Deris Stiawan
Department of Computer Science
Universitas Sriwijaya
Indonesia
Email: Deris@unsri.ac.id

1. INTRODUCTION

The in the number of complex and diverse (heterogeneous) traffic and distribution of internet of things (IoT) devices or services makes IoT security more complex and challenging [1]. According to [2]-[4] from 2013 to 2020, there will be around 24-50 billion new IoT devices that will be connected to the internet. With so many devices connected, it raises serious security problems, and it was proven [5] that in 2016 the biggest DDoS attack had occurred through an IoT device. Therefore, one solution is to implement an intrusion detection system in the heterogeneous network.

Diro and Chilamkurti [6] authors states that traditional machine learning cannot detect complex cybercrime actions, because the traditional machine learning train process fails to recognize small changes in the packet attack scenario and because it cannot extract invisible features. This is consistent with the fact that many attacks have mutated (around 99%) and only (1%) are still in the previous concepts and ways. The success of deep learning in detecting small changes such as small changes in image pixels shows the reliability of DL in the training process.

In research [7] shows that the application of deep learning not only can be applied to big data but can also be implemented in the classification of network traffic and intrusion detection systems. Several previous studies have used deep learning to detect attack traffic, including [8] using and combining deep learning and shallow learning for NIDS on KDD'99 and NSL-KDD datasets, besides [9] using deep learning sparse autoencoder and soft-max regression for detecting NSL-KDD datasets. Research [10] has been

proposed applying hybrid deep learning and autoencoder to improve the performance of accurate detection IDS. Then, detection time will faster with reducing dimensions of the dataset.

However, some attack detection studies on IoT using deep learning still use the KDD '99 and NSL-KDD dataset so that the results of deep learning testing become a major issue, therefore an IoT dataset will be built with several services that can describe heterogeneous networks. The purpose of this study is to apply deep learning for IoT intrusion detection systems on heterogeneous networks.

2. RELATE WORKS

DL has been implemented in various fields. for instance in network security (IDS). From literature [11] has been surveyed on the use of DL in IDS with several deep learning algorithms such as AE, RNN, CNN, RBM, and DBN. In [12] authors has been conducting research by comparing several conventional machine learning methods such as logistic regression, J45, SVM, RF, and DL were able to achieve the best accuracy. In addition, [12] also conducted a study attempting to apply deep learning using TensorFlow and tested on the MAWILeb's 2017 dataset also achieved satisfactory detection results.

Some propose a hybrid deep learning algorithm [9], namely AE and DBN whose purpose is to use AE for automatic feature extraction and DBN for detection or classification. From literature, previous researchers [13]-[18] result evaluate the proposed method using NSL-KDD and KDD'99 Cup datasets the research is not much different. The interesting is in [19] has been described as a few public datasets that can be used for testing with deep learning.

Sharipuddin *et al.* [20], the authors have been proposed deep learning with DBN to improve the intrusion detection system on IoT by comparing it with existing DGAs standards. Besides, [6] also proposes to use DL for detection systems on the IoT network, and the results of the research reach 99 percent accuracy. However, this study was evaluated with NSL-KDD and KDD CUP 99 datasets so it needs to be tested on real IoT networks to obtain accurate results.

3. RESEARCH METHOD

3.1. Deep learning algorithms

DL is the metamorphosis of machine learning from an ANN. DL are one of great innovations that pushing a lot of organization to advantage artificial Intelligence. DL is algorithms capable of founding the features such as the human brain. DL are developed with step by step of ANN. DL is consists of large neuron connections that can to high-level extraction of data features. In DL function learned by a neuron is evaluated and calculated by 1000s sub-neuron that outcome a comprehensive classification. DL has several forms, this work proposed to use deep belief network (DBN) to detect attacks in IDS-IoT. In the DBN process of learning and training data is in input. The DBN has features to pre-processes the data to clean the noise of the data that not suitable. There are a few DBN invariants in the range Figure 1. The normalization process in DBN is to prevent the decision that misguided. DBN can use the procedure of probabilistically to reconstruct data inputs, so the layer itself describes feature detectors [21], [22].

Deep belief network is consists of stacked few layers such as a multi-stage restricted boltzmann machine (RBM). The hidden layers in DBN composed of one number to allow the learning process faster. Often called log-linear, the RBM algorithm is constructed based on markov random field (MRF). The RBM energy function has its free parameters to increase accuracy. RBM is a block part of a deep trust network. The connections among neurons in visible layer are shown in Figure 1. The hidden layers exist between the layers in Figures 2. The neuron stores the results of computations at each layer. Each node can randomly input weights.

There are two steps of the DBN training process [23]: The first, train each layer of RBM separately in an unsupervised manner. The second, BP neural network in the last layer of DBN. We set the output vector from the last RBM as the input vector of the BP neural network, and then conduct supervised training for classifier relations.

3.2. Topology heterogeneous

We built a testbed topology to get heterogeneous real IoT datasets by using several different end-devices, services, and protocols so that they will depict heterogeneous networks in the real. In this work used hardware to develop testbed such as soil moisture, MQ2, Fundulno and DHT22. There are some nodes as end-devices and middleware. The middleware to communicate are using XBee, w1d D1 and wireless routers to connecting among middleware and monitoring server. The topology proposed in this study is shown in Figure 3. The attack used in this paper is DoS. The patterns of normal and attack obtained with analyzed through attributes [20], thus who can manually identify normal or attack data.

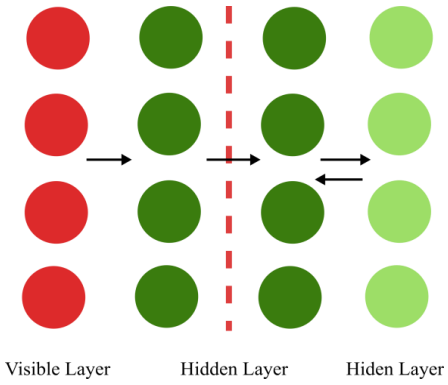

Figure 1. The architecture of RBM

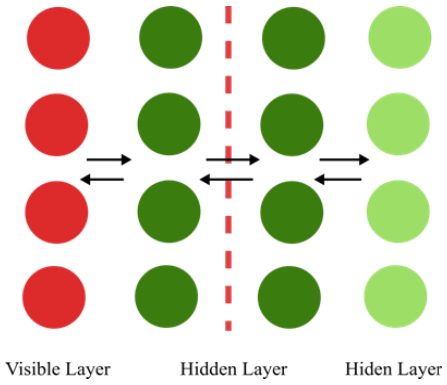

Figure 2. The architecture of DBN

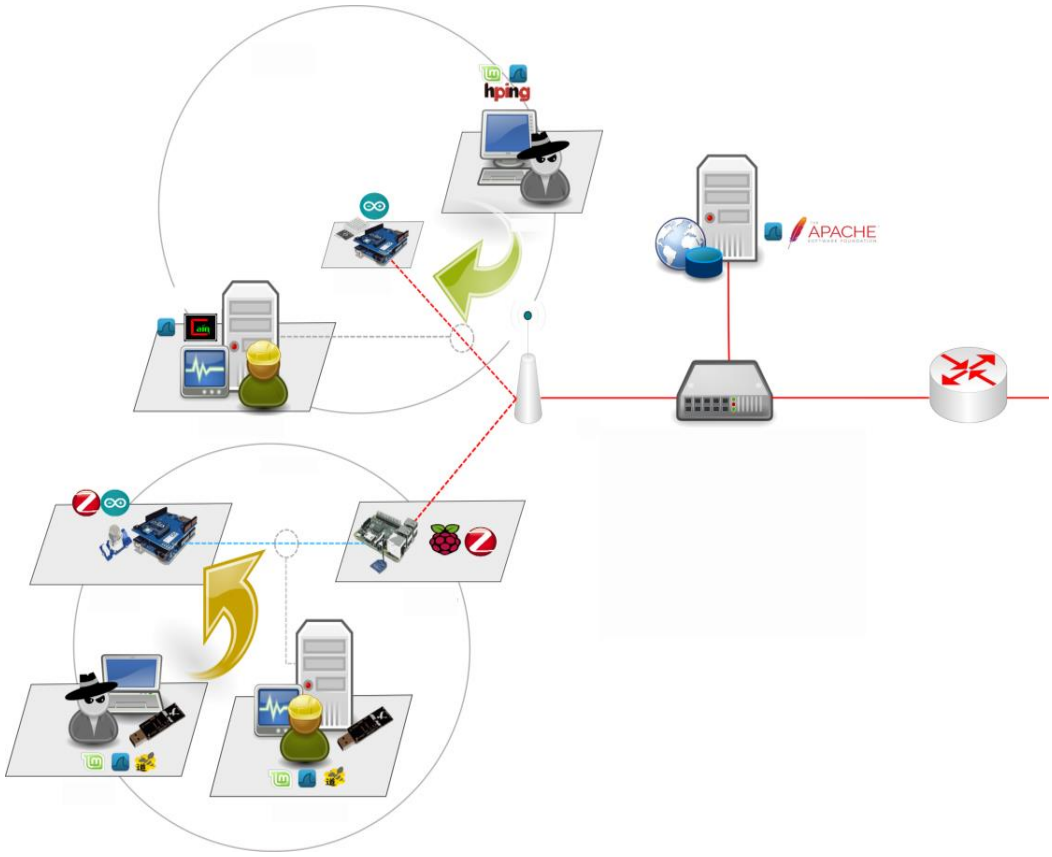

Figure 3. Dataset topology

3.3. Data preprocessing

This stage is the pre-processing of the dataset from the dataset that was obtained previously. This process is needed to extract the parameters needed to find and identify common basic patterns. Pcap files obtained from the sniffing process are difficult for humans to read because they have different header structures, and have hidden layers depending on different protocols and encapsulation processes. We proposed to two mechanisms to pre-processing dataset namely data conversion and normalization. Data conversions is to converted traffic features of nominal to numeric and ensure all numeric data to be processed by the detection system model. The following is the pseudocode from the process extract parameters dataset.

```
Input : DtI (Dataset)
dI = {dI1, dI2,…,dIi} (i number of packets)
Output: Hd (Result of Preprocessing)
```

```
def ekstrak(dli)
 for i ∈ dli do
       data ←), ip(dst, dsfield, ttl, src, addr ,flags-mf, dsfield-ecn, len, id, flags,
           src-host hdr-len, flags-df, checksum, checksum-status, host, frag-offset, addr,
           dst-host, proto, hos, dsfield-dscp, flags.rb), frame(time-epoch, time-delta,
           encap-type, offset-shift, time-relative, cap-len, len, marked, time, protocols,
           , number, ignored, time-delta-displayed, coloring-rule-name, tcp(flags-ns,
           analysis-initial-rtt, -ws-expert-group, srcport, flags-fin, nxtseq, port, seq,
           checksum-status,hdr-len, flagsack, flags-ecn, flags-ack, flagss-reset, analysis-
           acks-frame, -ws-expert, connection-sack, option-len, flags-res, time-relative,
           option-kind, -ws-expert-message, analysis, options-mss-val, flags-str, stream,
           window-size, -ws-expert-severity, checksum, len, flags-push, flags-cwr, options-
           mss, urgent-pointer, lags-syn, options, analysis-ack-rtt, flags-cwr, window-
           size-value, time-delta, dstport, port), eth(dst-resolved, type, ig, addr-
           resolved, addr, src, src-resolvedeth-addr-resolved, dst, lg, version, addr),
 end
return data

def main()
 dI ← read(DtI)
 for dIi ∈ dI do
 if dIi = ipv4 then
 dL ← ekstrak(dIi)
 Hd ←dL
 end
 end
end
```

Normalization is implemented to reduce high variants of features to a certain scale of values [21]. Zero values will be eliminated in process normalization. The method to normalize, we proposed to use the minimum-maximum method to scaling values features among zero and one.

$$Xi[0-1] = \frac{Xi-Xmin}{Xmax-Xmin}$$

X*i* is data point *i*. X*min* is smaller value of data points. X*max* is highly value of data points. X*i*[0-1] is result data point i normalized become range between 0 to 1. Duo some of columns contain only NaN value and in this particular case, NaN has been generated to zero.

3.4. IDS-DBN

In this paper, deep learning were useful to identify DoS attacks with dataset has captured. Figure 4 is flowchart proposed method IDS-DBN. First, dataset was captured consist of DoS attacks and benign behaviors from the heterogeneous network. Then the dataset must were normalized. Next flow is samples were split become data training and data testing. The data training consist five parts with number of dataset 50%, 60%, 70%, 80, 90% and to data testing are 50%, 40%, 30%, 20, 10%. The models of IDS-DBN were developing with basis on the data training. Last, the models was develop need evaluate with data testing. The outcome of IDS-BBN was measure performance of models developed. IDS-DBN consists of two hidden layers with number of neurons 8 respectively. The activation function has proposed to IDS-DBN model are relu and sigmoid. The number of neurons and hidden layer to IDS-DBN model changed depending of performances that obtained. In this work, we selected numbers of it based on the models accuracy. On the other hand, we did not apply feature selection method to IDS-DBN and we used all features of normalization. The future work, we will use different artificial intelligence approaches to define optimum values and applied feature extraction or feature selection to reduce the dimension of data input.

Figure 4 shown main of steps of IDS-DBN [24], [25]. The First, define of number of dataset of result preprocessing dataset become two data training and data testing. Second, Normalize the dataset is step to convert value of dataset become value with range 0 to 1. In addition, unrelated features like time, value is NaN, infinity, and empty will converted to zero. Third, develop IDS-DBN models that used to process detection with learn based on data training. The last is evaluated of IDS-DBN models.

In this work, the IDS-DBN models consist of a few layers. First layer is input layer with 62 dimensions and 12 nodes. Second layer are hidden layer that consists of 2 layers and 8 nodes. The last layer is layer output with key activation sigmoid that produce two class attack and normal.

3.5. Performance metrics

We use four metrics the most common validations to measure of performance IDS-DBN model explained by:

$$Precision = \frac{(TP)}{(TP+FP)}$$

$$F1 - score = 2 \text{ x } \frac{(Precision\ x\ Recall)}{(Precision + Recall)}$$

$$Recall = \frac{(TP)}{(TN+FP)}$$

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

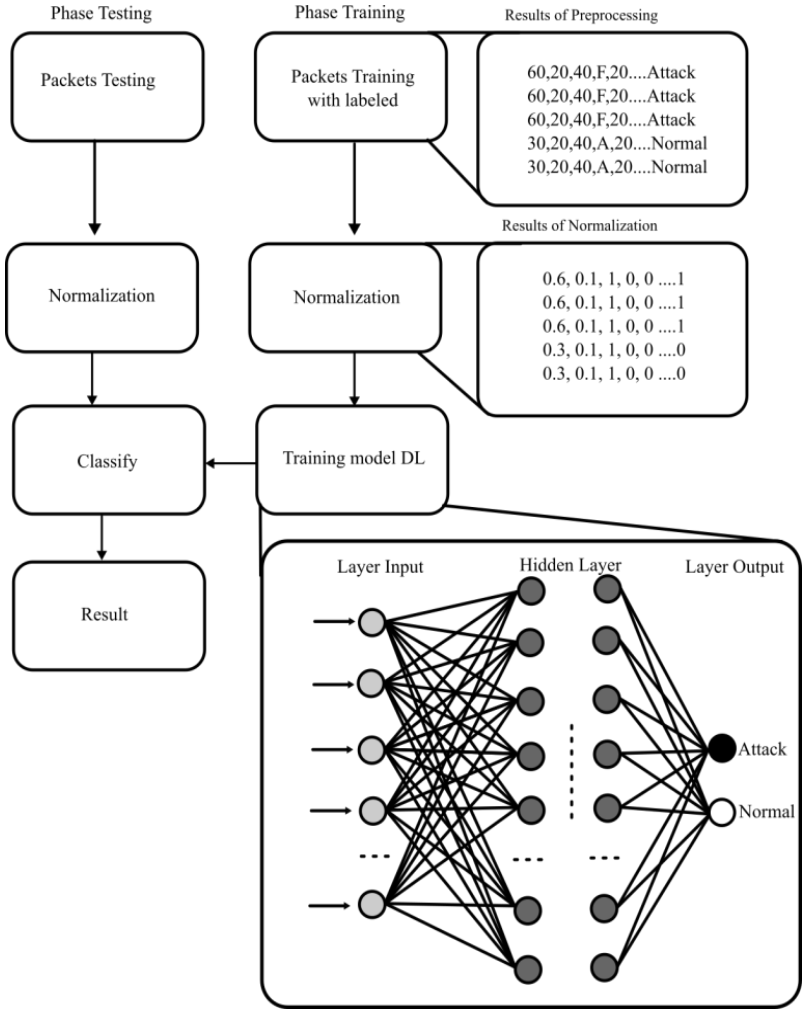

Figure 4. Design IDS-DBN

4. THE RESULTS AND ANALYSIS

For the experiment, we used a Dell notebook with Intel Core i7, 256 SSD, 12GB memory, and the Ubuntu 18.04 LTS. The frameworks to build the DBN are use TensorFlow python and Scikit-learn to the dataset normalization process. Here are the deep learning setup variables. This section is discussing the results of experiment that have carried out. In experiment of topology, there are two data testing is dataset benign and attack with a five-minute observation period.

Table 1 is the number of packets from the results of experiments that have been carried out the amount to 1213299, there are a few protocols namely TCP, UDP and ARP. The number of packets is consisting of an attack of 1139179 and a normal amount to 74121. The preprocessing process has obtained attributes from these process 95 features to Wi-Fi protocol. Next is the process of normalization. The goal of normalization is to eliminate irrelevant features used to training and testing process of IDS-DBN to 62

attributes that can be seen in Table 2. It displays the results of the normalization of the extracted data attribute results and then the attributes of the typed string converted to numeric. Then, data transformed to a scale of 0 to 1 as shown in Figure 5, so that it can be input in the IDS-DBN detection system.

Table 1. Number of packets

Type packet	Amount packet data
TCP	1212329
UDP	139
ARP	832

Table 2. Results of normalization

Attributes
54,54,0,0,1,4,20,0,0,0,40,80,0,0,0,0,0,64,6,58657,2,1,5340,80,0,5340,80,3463,0,1,1,1997724222,20,1,0,0,0,0,0,0,0,4194304,2097152,150994944,33554432,1,2,512,512,45206,2,0,0,0,0,0,0,0,0,0,0,0.0,0.0,Attack
54,54,0,0,1,4,20,0,0,0,40,1048,0,0,0,0,0,128,6,41312,2,1,19656,80,0,19656,80,4,0,1,1,1,20,16,0,0,0,0,0,1,0,0,0,0,0,1,5840,5840,45053,2,0,0,0,0,0,0,1,30,0.004183,0.004287,0.004287,0.004183,Normal

```
0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
0.00000000e+00 0.00000000e+00 4.39374981e-01 0.00000000e+00
0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
0.00000000e+00 0.00000000e+00 4.56907865e-01 0.00000000e+00
0.00000000e+00 9.93148699e-01 1.22072175e-03 0.00000000e+00
9.93148699e-01 1.22072175e-03 8.34879717e-01 0.00000000e+00
9.62855794e-01 9.62855794e-01 1.76843821e-01 0.00000000e+00
0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
0.00000000e+00 6.66666667e-01 6.25000000e-02 2.77777778e-02
1.00000000e+00 1.00000000e+00 1.00000000e+00 2.98023224e-08
5.96046448e-08 7.81273843e-03 1.74672489e-02 1.80120243e-01
3.11332503e-05 0.00000000e+00 0.00000000e+00 0.00000000e+00
0.00000000e+00 0.00000000e+00 0.00000000e+00 4.13730559e-07
0.00000000e+00]
```

Figure 5. Result data pre-processing

The step after preprocessing of dataset is developing IDS-DBN model. In this paper, we have implemented IDS to multi-class classification attack based on DBN, the IDS-DBN models to training use a sequence of RBMs. Each row of dataset has 62 features as input IDS-DBN model and two outputs. In epochs in the proses training is 100 and batch sizes of layers is 10. The detailed of variables for build model IDS-DBN show in Table 3.

In Table 4 is the result of deep learning testing using data sharing by 50 percent for training and 50 percent for testing. In this testing, the attack detection results were 569457 packages and 36707 normal packages with error detection in this test reaching 0 percent. This test will be carried out 5 times by sharing data from 50 percent to 10 percent for testing data.

Table 3. Variable deep learning

Layers	Three (one input layer, two hidden layer, one output layer)
Node	12 node, 8 node, 8 node, 2 node
Activation	Relu, relu, relu, sigmoid
Input dimension	62
Epoch	100
batch_size	10

Table 4. Results of testing DL

	Normal	Attack
Normal	36707	0
Attack	0	569457

From Table 5 shows the results of performance metrics from the tests that have been carried out and obtained that deep learning can detect packet traffic on a heterogeneous IoT network to reach 100 percent acuity. These results may also be influenced by the lack of packet types in the dataset that has been built. There is an interesting thing that is obtained from the results of this test is that deep learning can conduct training and testing with a large dimension dataset that reaches 62 features successfully. The following Figure 6 shows the experimental accuracy of respect for the percentage of data is used for training.

Table 5. Results of testing classification

Testing (%)	Precision	Recall	F1-score	Accuracy (%)
50:50	1.00	1.00	1.00	100
60:40	1.00	1.00	1.00	100
70:30	1.00	1.00	1.00	100
80:20	1.00	1.00	1.00	100
90:10	1.00	1.00	1.00	100

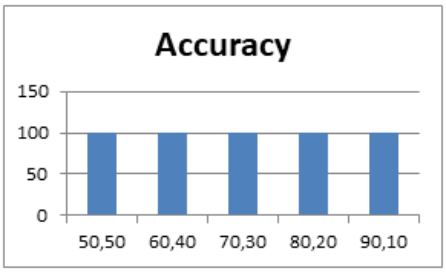

Figure 6. Result of testing

5. CONCLUSION

This work proposes to use deep learning to IDS IoT with a deep belief network to detect attacks on heterogeneous networks with considerable dimensional features. The result of the evaluation is deep learning successful to identify attacks that occur in heterogeneous networks. The accuracy detection achieves around 99 percent. In future research, the IDS IoT application of feature extraction to reduce features of dimensions of the data so the resources that can less.

ACKNOWLEDGMENTS

This work funded by UNAMA (Universitas Dinamika Bangsa) by HR development programs and support by COMNETS Lab Universitas Sriwijaya.

REFERENCES

[1] D. N. Jha, P. Michalak, Z. Wen, R. Ranjan, and P. Watson, "Multiobjective Deployment of Data Analysis Operations in Heterogeneous IoT Infrastructure," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 11, pp. 7014-7024, 2020, doi: 10.1109/TII.2019.2961676.

[2] M. I. Ali *et al.*, "Security challenges and cyber forensic ecosystem in IoT driven BYOD environment," *IEEE Access*, vol. 8, pp. 172770-172782, 2020, doi: 10.1109/ACCESS.2020.3024784.

[3] N. M. Karie, N. M. Sahri, and P. Haskell-Dowland, "IoT Threat Detection Advances, Challenges and Future Directions," *Proceedings - 2020 Workshop on Emerging Technologies for Security in IoT, ETSecIoT 2020*, 2020, pp. 22-29, doi: 10.1109/ETSecIoT50046.2020.00009.

[4] M. Sikimic, M. Amovic, V. Vujovic, B. Suknovic, and D. Manjak, "An Overview of Wireless Technologies for IoT Network," *2020 19th International Symposium INFOTEH-JAHORINA, INFOTEH 2020 - Proceedings*, no. March, 2020, pp. 18-20, doi: 10.1109/INFOTEH48170.2020.9066337.

[5] D. Erhan and E. Anarim, "Hybrid DDoS Detection Framework Using Matching Pursuit Algorithm," *IEEE Access*, vol. 8, pp. 118912-118923, 2020, doi: 10.1109/ACCESS.2020.3005781.

[6] 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, doi: 10.1016/j.future.2017.08.043.

[7] Kim, M. Park, and D. H. Lee, "AI-IDS: Application of Deep Learning to Real-Time Web Intrusion Detection," *IEEE Access*, vol. 8, pp. 70245-70261, 2020, doi: 10.1109/ACCESS.2020.2986882.

[8] D. Preethi and N. Khare, "Performance Evaluation of Shallow learning techniques and Deep Neural Network for

Cyber Security," in *International Conference on Emerging Trends in Information Technology and Engineering, ic-ETITE 2020*, 2020, pp. 2-6, doi: 10.1109/ic-ETITE47903.2020.128.
[9] 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, doi: 10.1109/ACCESS.2020.3028690.
[10] S. Zavrak and M. Iskefiyeli, "Anomaly-Based Intrusion Detection from Network Flow Features Using Variational Autoencoder," *IEEE Access*, vol. 8, pp. 108346-108358, 2020, doi: 10.1109/ACCESS.2020.3001350.
[11] S. Gamage and J. Samarabandu, "Deep learning methods in network intrusion detection: A survey and an objective comparison," *Journal of Network and Computer Applications*, vol. 169, no. February, Art. No. 102767, 2020.
[12] Alsughayyir, A. M. Qamar, and R. Khan, "Developing a network attack detection system using deep learning," *2019 International Conference on Computer and Information Sciences, ICCIS 2019*, pp. 1-5, 2019, doi: 10.1109/ICCISci.2019.8716389.
[13] L. Wu and T. Deng, "Computer Network Security Analysis Modeling Based on Spatio-temporal Characteristics and Deep Learning Algorithm," *Journal of Physics: Conference Series*, vol. 1648, no. 4, Art. No. 042111, 2020.
[14] Z. Lv, L. Qiao, J. Li, and H. Song, "Deep Learning Enabled Security Issues in the Internet of Things," *IEEE Internet of Things Journal*, vol. 4662, no. 2017, pp. 1-1, 2020, doi: 10.1109/JIOT.2020.3007130.
[15] H. Hindy, R. Atkinson, C. Tachtatzis, J. N. Colin, E. Bayne, and X. Bellekens, "Utilising deep learning techniques for effective zero-day attack detection," *Electronics (Switzerland)*, vol. 9, no. 10. pp. 1-16, 2020, doi: 10.3390/electronics9101684.
[16] M. Haggag, M. M. Tantawy, and M. M. S. El-Soudani, "Implementing a Deep Learning Model for Intrusion Detection on Apache Spark Platform," *IEEE Access*, vol. 8, no. Dl, pp. 163660-163672, 2020, doi: 10.1109/ACCESS.2020.3019931.
[17] J. Zhang, F. Li, and F. Ye, "An Ensemble-based Network Intrusion Detection Scheme with Bayesian Deep Learning," *IEEE International Conference on Communications*, vol. 2020-June, 2020, doi: 10.1109/ICC40277.2020.9149402.
[18] S. Choudhary and N. Kesswani, "Analysis of KDD-Cup'99, NSL-KDD and UNSW-NB15 Datasets using Deep Learning in IoT," *Procedia Computer Science*, vol. 167, no. 2019, pp. 1561-1573, 2020, doi: 10.1016/j.procs.2020.03.367.
[19] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study," *Journal of Information Security and Applications*, vol. 50, Art. No. 102419, 2020, doi: 10.1016/j.jisa.2019.102419.
[20] Sharipuddin *et al.*, "Measurement of Component Performance (Sensor) on Internet of Thing (IoT)," *Proceedings of 2018 International Conference on Electrical Engineering and Computer Science, ICECOS 2018*, 2019, pp. 339-344, doi: 10.1109/ICECOS.2018.8605265.
[21] M. M. Hassan, A. Gumaei, A. Alsanad, M. Alrubaian, and G. Fortino, "A hybrid deep learning model for efficient intrusion detection in big data environment," *Information Sciences*, vol. 513, pp. 386-396, 2019, doi: 10.1016/j.ins.2019.10.069.
[22] M. Z. Alom, V. Bontupalli, and T. M. Taha, "Intrusion detection using deep belief networks," *Proceedings of the IEEE National Aerospace Electronics Conference, NAECON*, vol. 2016-March, pp. 339–344, 2016.
[23] A. A. Süzen, "Developing a multi-level intrusion detection system using hybrid-DBN," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 2, pp. 1913–1923, 2021.
[24] I. Sohn, "Deep belief network based intrusion detection techniques: A survey," *Expert Systems with Applications*, vol. 167, Art. No. 114170, 2021.
[25] K. Singh and K. J. Mathai, "Performance Comparison of Intrusion Detection System Between Deep Belief Network (DBN) Algorithm and State Preserving Extreme Learning Machine (SPELM) Algorithm," *Proceedings of 2019 3rd IEEE International Conference on Electrical, Computer and Communication Technologies, ICECCT 2019*, 2019, pp. 1–7.