

Improvement detection system on complex network using hybrid deep belief network and selection features

Sharipuddin¹, Eko Arip Winanto², Zulwaqar Zain Mohtar³, Kurniabudi², Ibnu Sani Wijaya¹,
Dodi Sandra¹

¹Department of Informatics, Faculty of Computer Science, Universitas Dinamika Bangsa, Jambi, Indonesia

²Department of Computer Engineering, Faculty of Computer Sciences, Universitas Dinamika Bangsa, Jambi, Indonesia

³School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia, Johor Bahru, Malaysia

Article Info

Article history:

Received Sep 2, 2022

Revised Feb 21, 2023

Accepted Mar 12, 2023

Keywords:

Complex network

Deep belief network

Feature extraction

Feature selection

Intrusion detection system

ABSTRACT

The challenge for intrusion detection system on internet of things networks (IDS-IoT) as a complex networks is the constant evolution of both large and small attack techniques and methods. The IoT network is growing very rapidly, resulting in very large and complex data. Complex data produces large data dimensions and is one of the problems of IDS in IoT networks. In this work, we propose a dimensional reduction method to improve the performance of IDS and find out the effect of the method on IDS-IoT using deep belief network (DBN). The proposed method for feature selection uses information gain (IG) and principle component analysis (PCA). The experiment of IDS-IoT with DBN successfully detects attacks on complex networks. The calculation of accuracy, precision, and recall, shows that the performance of the combination DBN with PCA is superior to DBN with information gain for Wi-Fi datasets. Meanwhile, the Xbee dataset with information gain is superior to using PCA. The final result of measuring the average value of accuracy, precision, and recall from each IDS-DBN test for IoT is 99%. Other results also show that the proposed method has better performance than previous studies increasing by 4.12%.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Eko Arip Winanto

Department of Computer Engineering, Faculty of Computer Sciences, Universitas Dinamika Bangsa

36125 Kota Baru, Jambi, Indonesia

Email: ekoaripwinanto@unama.ac.id

1. INTRODUCTION

The growing number of complex and diverse traffic (complexity) and distribution of internet of things (IoT) devices or services make the security of IoT increasingly complex and challenging [1]. In addition, IoT attack detection is also different from detection systems on conventional networks such as limited resources, low latency, distribution, scalability, and mobility [2]. According to the findings in [3], conventional machine learning techniques are not capable of identifying sophisticated cybercrime activities. The training process of conventional machine learning approaches is unable to identify minor variations or mutations in the patterns of attack packets due to its inability to extract latent features. This is consistent with the observation that many attacks have evolved and only a limited number of them remain within the confines of the original concept and methodology. On the other hand, deep learning has demonstrated a remarkable capability to identify small variations, such as subtle changes in image pixels, thereby showcasing its reliability in the training process. Therefore, the current study seeks to leverage deep learning techniques for the detection of complex cyber attacks on IoT networks.

Research [4]-[6] shows the application of deep learning not only be applied to big data but can also be implemented into network traffic classification and intrusion detection systems (IDS). There are several previous studies that have used deep learning. Thakkar and Lohiya [7] proposed a deep belief network (DBN)

based approach to detect attacks on IoT networks and managed to achieve 99.0% accuracy. Balakrishnan *et al.* [8] has proposed the DBN method for IoT networks. In addition, Sharipuddin *et al.* [9] proposed a hybrid model to improve the performance of DBN with autoencoder, the detection results show that there is an increase in performance than a single DBN. Therefore, there are challenges that need to be remedied, one of which is improving the performance of deep learning for IDS on complex networks. Wang and Wei [10] one may employ feature selection or feature extraction to improve the performance of deep learning on IDS-IoT. This can be done either manually or automatically.

The goal of feature selection is to select a subset of variables from the input data in an effective manner so that the variables can describe the input data while at the same time reducing the effects of noise or irrelevant variables and still providing good predictive results [11], [12]. Feature selection was developed with this purpose in mind. Feature extraction, on the other hand, is the process of removing features from original features that already exist and converting features to lower dimensions in order to accelerate the training process and increase accuracy results [13], [14]. Selection of features is a deeply important process in an IDS, and the performance or accuracy of a IDS will change drastically when given different feature inputs. In addition, a large amount of traffic on the IoT network and high-dimensional features will affect the results of the classification process [15], [16]. In order to improve the performance of the detection system on complex IoT networks, in-depth research on the influence of feature selection or extraction is crucial.

Therefore, in this work focus on comparing the increasing effects of using feature selection and feature extraction in complex network IDS using DBN. In addition, this research has several contributions as: i) finding dataset features on complex IoT networks using feature selection and feature extraction, ii) proposing a detection system on complex IoT networks using DBN, and iii) identify the effects of features selection and features extraction to improve the performance of IDS using DBN. This paper is organized into four sections. Section 1 is the introduction. Section 2 provides a brief discussion of the experimental dataset and setup used in the study. Section 3 provides a more detailed description of the experiment and the findings of the study. Finally, section 4 presents the conclusions of the study and suggests potential avenues for future research.

2. METHOD

This work focuses on comparing feature selection improvements using deep learning. In this section, we describe the steps to complete this research. This section describes datasets, experimental configurations, feature selection techniques, classification algorithms, and experimental tools.

2.1. Dataset

We utilize a complex IoT dataset from Comnets Lab Unsri [17] for this work. To depict an IoT complex network in a real environment. The hardware employed consists of nodes (PC, Raspy, and Arduino) and sensors (soil moisture, MQ2, and Fundulno). To connect middleware to the server, XBee, w1d D1, and WIFI are all utilized as middleware. Table 1 in this dataset presents numerous attack scenarios and kinds, including benign, TCP flood, and zbassocflood on Xbee. Two different sorts of datasets using the Wi-Fi and Xbee protocols. In this work, the features dataset consists of 96 characteristics for WIFI and 65 attributes for Xbee. The IoT Comnets Lab Unsri dataset was selected in order to have a dataset that represents the current real-world network traffic in the experiment.

Table 1. Dataset

File name	Types of traffic	Numbers of record
normal_server	Benign	12792
serangan_server	TCP flood, Benign	3135393
normalxserangan_server	TCP flood, Benign	3709681
normal_mid1	Benign	1739
serangan_mid1	TCP flood, Benign	1175059
normalxserangan_server	TCP flood, Benign	1191320
normal_mid2	Benign	2102
serangan_mid2	TCP flood, Benign	1555706
normalxserangan_mid2	TCP flood, Benign	1603038
normal_node_wifi	Benign	7806
serangan_node_wifi	TCP flood, Benign	2399420
normalxserangan_node_wifi	TCP flood, Benign	2426599
normal_node_xbee	Benign	568
serangan_node_xbee	Zbassocflood, Benign	19426
normalxserangan_node_xbee	Zbassocflood, Benign	22441

2.2. Experiment setup

The present study has a considerable emphasis on the experimental phase for IDS using deep learning. The experiment encompasses a comprehensive examination of various aspects, such as the utilization of feature selection and feature extraction techniques, the distribution of datasets for training and testing purposes, and the design and configuration of the DBN method and its associated variables. In general, there are three stages of experimental setting as shown in Figure 1, which can be described as:

- Feature selection, this study uses information gain for feature selection and principal component analysis (PCA) for feature extraction.
- Furthermore, the results of each feature group or feature subset are classified using DBN. The analysis considers parameters such as precision, recall, and accuracy. The test was carried out using the results of the information gain and PCA features.
- Finally, compare and analyze precision, recall, and accuracy for each dataset type, and reduction method from IG or PCA.

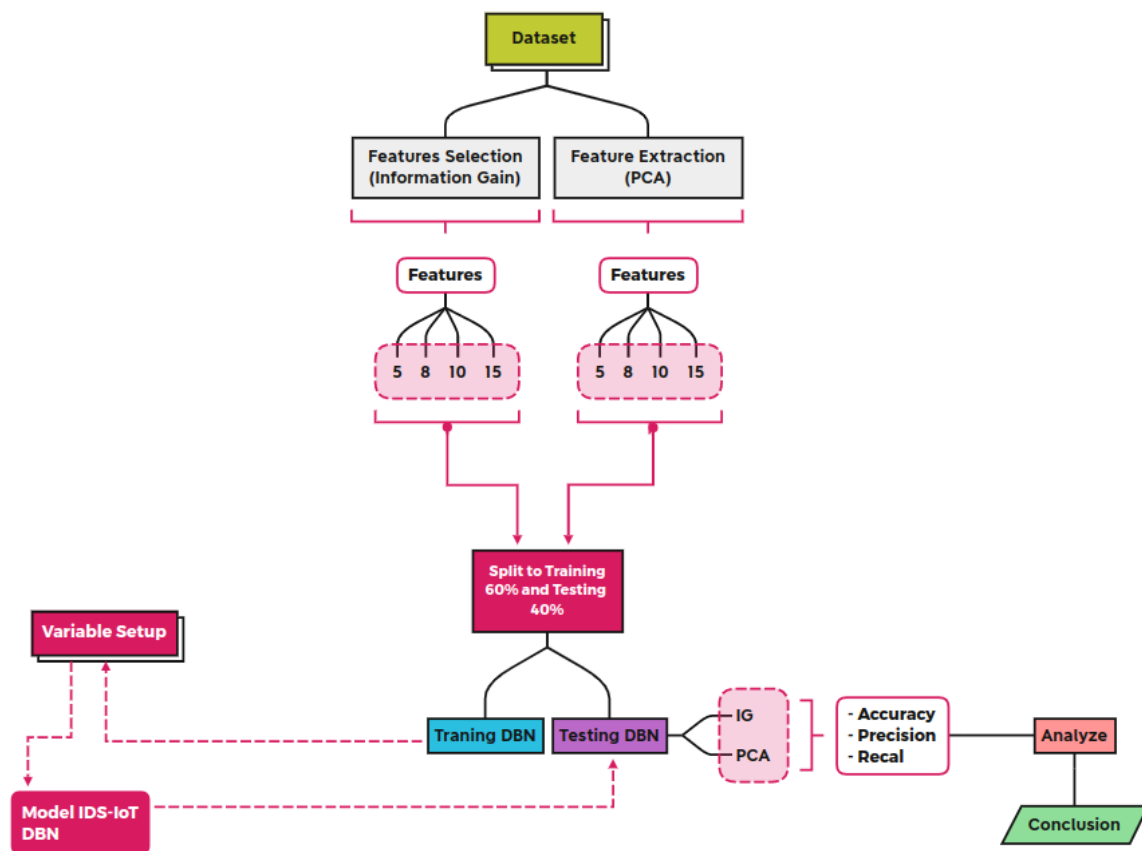


Figure 1. Experiment design

2.3. Features selection using information gain

The method of feature selection known as information gain is the one that is most widely used. It is a method of selecting features that is predicated on an information gain filter, and it works by first reducing the noise brought on by irrelevant features through the use of simple attribute ranking, and then identifying those features that have the majority of the information base in a given class [18]. Calculating the entropy of a feature is one way to determine which one is superior to others. Entropy is a measure of uncertainty that can be used to infer feature distributions in a concise form [19]. Entropy can be used to infer feature distributions from its value.

This work chose information gain as feature selection because it is a filter-based technique that provides a more stable selected feature set due to its strong nature against overfitting [20]. Thus, the use of feature selection techniques that produce significant, relevant features, fewer features, and less computational complexity will reduce the execution time of the classification algorithms used in the attack detection process

on heterogeneous IoT networks. Its features are divided into two parts, namely features for Wi-Fi and Xbee datasets. Information gain ranks feature based on their weight values and minimum weights. In this work, we divide into four groups, namely 5, 8, 10, and 15 features. Thus, feature groups are obtained and each feature group will have a different number of features. Furthermore, all feature groups will be validated using a classification algorithm, so that it can be determined which feature groups are effective enough to be used for the detection process with DBN.

2.4. Features extraction using principal component analysis

PCA is a mathematical equation that converts high-dimensional data to low-dimensional data that contains most of the information from high-dimensional data [19]. PCA is one of the data dimension reduction techniques and is a multivariate analysis of data tables where observations are described by several correlated quantitative dependent variables. The goal is to extract important information from the table to represent it as a new set of orthogonal variables called principal components, and to display patterns of similarity of observations and variables as points on the map.

The three main components in PCA calculations are covariance, eigen value, and eigen vector. The values of the three main components were calculated using a mathematical equation obtained using PCA. In this work, we will reduce the dimensions with PCA into four groups similar to the information gain, namely 5, 8, 10, and 15 features. These features will later be used for the IDS-IoT training process using DBN.

2.5. Deep belief network

DBN is a kind of deep neural network, which consists of stacked layers of restricted boltzmann machines (RBM). This is a generative model and was proposed by [20]. DBN can be used to complete unsupervised learning tasks to reduce feature dimensions, and can also be used to complete supervised learning tasks to build classification models or regression models. To train a DBN, there are two steps, layer-by-layer training and fine-tuning [21]. Layer-by-layer training refers to the unsupervised training of each RBM, and fine-tuning refers to the use of an error back-propagation algorithm to fine-tune the DBN parameters after the unsupervised training is complete [22]. The DBN model of the combined distribution between the observed vector x and l hidden layers h_k is as shown in (1):

$$P(x, h^1, \dots, h^l) = (\prod_{k=0}^{l-2} P(h^k | h^{k+1})) P(h^{l-1}, h^l) \quad (1)$$

where $x=h_0$, $P(h^k | h^{k+1})$ is the conditional distribution for visible units conditioned on hidden RBM units at level k , and $P(h^{l-1}, h^l)$ is the combined visible-hidden distribution in top-level RBMs.

2.6. Analysis tools

The simulations were run on a computer with an Intel Core i7 processor running at 2.60 GHz and 12 GB RAM, with Ubuntu 20.04.3 LTS as the operating system. For analysis purposes, Weka 3.9 was used as feature selection software with a heap size of 3072 MB. Additionally, scikit-learn and tensorflow were used for machine learning, and keras was used for deep learning.

3. RESULTS AND DISCUSSION

This comprehensive examination outlines the preparation of the dataset, the methodology employed for feature selection and extraction, the implementation of the IDS-IoT DBN, and a thorough analysis of the experimental outcomes, including a discussion of the results. This section presents in more detail the results obtained from the experiment. In addition, it also analyzes experimental results and compares experimental results with other previous studies.

3.1. Dataset preparation

The initial preparation of the preparation dataset is normalization. The meaning of normalization is to remove features that have no value from the dataset (.csv). In addition, it also means eliminating irrelevant features to become attack pattern features such as time, and IP address. The results of this dataset preparation will be used for feature selection and feature extraction processes. The number of features after this process is 66 features originating 96 for WiFi datasets. Meanwhile, in the Xbee dataset, there is no feature reduction for the feature selection and extraction process.

3.2. Result of information gain

This section is the result of the feature selection process using information gain to select the sub-sections of the Wi-Fi and Xbee dataset features. Table 2 are the results of the feature selection process using

the information gain method. This feature extraction stage is carried out using a dataset of 20000 lines in the form of normal data and data.

Feature selection testing is done using the Weka application for ranking information gain. 66 attributes out of a total of 96 attributes are used in the feature selection phase of the Wi-Fi package. Based on the results of this stage, there are 49 features that affect the characteristics of the Wi-Fi dataset. The effect of each attribute can be seen from the weight value of the ranking results using information gain. A feature weight with a value of 0 means that it has no effect on the characteristics of the dataset for the IDS-IoT classification process with DBN.

Table 3 is the result of feature selection in the Xbee dataset. The feature extraction features used in the Xbee dataset are 65 after the normalization process is carried out. The results of feature selection using information gain on the Xbee dataset show that only 27 features affect the characteristics of the Xbee dataset. Table 3 is the result of feature selection in the study which will be divided into 4 types, namely 5/8/10/15 features.

Table 2. Feature selection using information gain

WI-FI				Xbee			
No	Weight	Feat. ID	Feat. name	No	Weight	Feat. ID	Feat. name
1	0.972	50	tcp.flags.str	1	0.998	62	data.data
2	0.972	51	tcp.window_size	2	0.998	9	frame.cap_len
3	0.972	38	tcp.flags	3	0.998	63	data.len
4	0.971	36	tcp.ack	4	0.998	42	wpan.frame_length
5	0.939	25	ip.ttl	5	0.998	8	frame.len
...
66	0	1	frame.encap_type	65	0	2	frame.time

Table 3. Feature selected

WI-FI		Xbee	
Number of selected features	New features subset	Number of selected features	New features subset
5	50,51,38,36,25	5	62,9,63,42,8
8	50,51,38,36,25,2,31,19	8	62,9,63,42,8,43,4,5
10	50,51,38,36,25,2,31,19,27,3	10	62,9,63,42,8,43,4,5,12,54
15	50,51,38,36,25,2,31,19,27,3,32,52,49,29,44	15	62,9,63,42,8,43,4,5,12,54,48,60,59,56,53

3.3. Result of PCA

This stage is to reduce the dataset into smaller dimensions. The goal is to reduce the training data processing and improve the performance of IDS. In this work, we propose to use the PCA method to reduce the dimensions of the dataset features without losing the characteristics of the data. Table 4 are the results of feature extraction using the PCA method. From Table 4, the results can be seen in this study, the dataset will be converted into four categories. The dataset is converted into 15, 10, 8, and 5 features. The results of this feature extraction will be used to process IDS training data using DBN. In this PCA process, the dataset is converted to a value with a range of 0 to 1. The value of the dataset after being converted into a smaller range. The aim is to reduce the resource usage of IDS-IoT machines with DBN.

Table 4. Feature extraction using PCA

WI-FI		Xbee	
Number of selected features	New features subset	Number of selected features	New features subset
5	0.02793, -0.09260, 0.02802, -0.00393, -0.01220	5	0.00620, -0.02867, -0.00259, -0.01495, 0.01134
8	0.00234, 0.01818, -0.0957, 0.06531, 0.01215, 0.02118, 0.02293, -0.01749	8	0.05068, -0.00189, 0.06662, 0.09061, 0.10891, 0.02286, 0.01770, -0.03581
10	0.03598, 0.03862, -0.0027, -0.00654, -0.00634, -0.03874, 0.0207, -0.05040, 0.00210, -0.00051	10	-0.07090, -0.04464, 0.03906, -0.03321, -0.01257, -0.03450, -0.02499, -0.00259, 0.06773, -0.01350
15	0.06350, 0.05068, -0.00189, 0.06662, 0.09061, 0.10891, 0.02286, 0.01770, -0.03581, 0.00306, 0.03906, -0.03321, -0.01257, -0.03450, -0.02499	15	-0.00188, -0.04464, -0.05147, -0.02632, -0.00844, -0.01916, 0.07441, -0.03949, -0.06832, -0.09220, 0.04445, -0.00567, -0.04559, -0.03419, -0.03235, -0.04069, -0.01944, -0.06899, -0.0792, 0.04127

3.4. Result of IDS-IoT using DBN

To analyze the performance of IDS-IoT using DBN enhanced with information gain, and PCA, 3 (three) measurements are used, namely precision, recall, and accuracy. In the experiment, each subset of IG and PCA features were classified by DBN into 4 types: 5/8/10/15 features. In this study, two steps must be taken to use DBN for an attack detection system on a complex IoT network. The first step is to carry out a learning process to obtain hierarchical weights and biases from the DBN network using layered RBM and setting variables in Table 5. The result of the learning process from DBN is a value of weights and biases that will be used in the detection or prediction process. Next is the prediction process using DBN to detect attacks on IoT networks.

Table 5. Variable DBN

Variable name	Description
Number of Layer	4(1 input, 2 hidden, 1 output)
Node	12 node, 8 node, 8 node, 2 node
Input dimension	Relu, relu, relu, sigmoid
Output dimension	5/8/10/15 (result of PCA or IG)
Epoch	100
batch_size	10

The IDS-IoT DBN testing was carried out on as many as 96 dataset files. The 96 files were obtained from the feature extraction and feature selection processes, the details of which are shown in Table 5. The IDS-DBN testing process is done by dividing each file in half. Each dataset is divided into training data and testing data. The distribution of the dataset is based on 60% for training data and 40% for testing data. Table 6 is an example of the accuracy results from DBN testing on IDS-IoT. The accuracy results consist of 4 groups according to the number of features, namely 5/8/10 and 15 features. Unsatisfactory accuracy results were obtained from testing the Xbee dataset on features 8 and 10. The highest average accuracy was obtained from the results of 5 features which reached 92-99%.

Table 6. Accuracy using IG

Dataset	Accuracy			
	Number of features			
	5	8	10	15
TCP (WI-FI)				
normal_server	100	100	100	100
serangan_server	99.33	66.33	66.33	93.01
normalxserangan_server	100	99.85	99.92	99.98
normal_mid1.pcap	99.99	99.67	99.65	99.65
serangan_mid1	100	100	100	100
normalxserangan_server	81.23	79.77	79.72	79.71
normal_mid2	78.89	78.45	78.45	78.45
serangan_mid2	100	100	100	100
normalxserangan_mid2	79.77	78.71	79.56	78.71
normal_node_wifi	79.11	78.04	77.96	77.96
serangan_node_wifi	100	100	100	100
normalxserangan_node_wifi	93.33	93.41	93.33	93.41
Xbee				
normal_node_xbee	100	100	100	80.04
serangan_node_xbee	99.97	50.14	50.21	100
normalxserangan_node_xbee.pcap	99.94	49.70	49.68	99.98

To simplify the presentation of the data, in this work, it will be displayed in the form of the average value of each dataset. The division is divided into two datasets, namely WIFI and Xbee datasets. Table 7 is the results of the average accuracy of the IG and PCA tests. From the comparison table of accuracy results, it is found that the use of IG on IDS-IoT with DBN on the Xbee dataset is superior to the use of PCA. Then the TCP dataset is higher on IDS-DL PCA than on IDS-DL IG.

Table 7 also shows the results of the IDS-DBN precision parameters using IG and PCA. Precision is the level of accuracy of IDS-DBN in detecting or predicting an attack on heterogeneous IoT networks. The second best result is on the TCP dataset which varies the largest reaching 100% and the smallest reaching 0.53 or 53% on the attack_server dataset in 8 features. Finally, the Xbee dataset achieves the lowest precision of 0.49 on 8 and 10 features. Furthermore, in IDS-DBN with PCA, the best results are on the TCP dataset which is quite varied which reaches 100% and the smallest reaches 0.78. These results show that PCA results are

better than IG for TCP datasets on precision parameters. The last is the Xbee dataset which shows unsatisfactory test results with a precision level that only reaches 50%.

The next result is the recall measurement which can be seen in Table 7. Table 7 also shows the data from the recall using GI to reduce the dimensions of the dataset as DBN input. From the results of the table, it can be seen that the results do not vary, which means that the results are very satisfactory, reaching 1.00 or 100%. Furthermore, the results from PCA show unsatisfactory recall results on the Xbee dataset in groups 10 and 15 features.

Table 7. Feature extraction using PCA

Dataset	No of feat.	Accuracy		Precision		Recall	
		IG	PCA	IG	PCA	IG	PCA
Average recall of WIFI dataset	5	92.08	91.70	0.92	0.91	1	1
	8	91.41	91.76	0.87	0.91	0.92	1
	10	91.41	91.77	0.91	0.91	0.99	1
	15	91.33	91.75	0.91	0.92	0.94	1
Average recall of Xbee dataset	5	99.33	67.51	1	0.66	1	0.99
	8	66.33	67.65	0.83	0.66	0.83	0.98
	10	66.33	75.80	0.91	0.74	1	0.75
	15	92.01	67.36	0.91	0.65	0.93	0.66

3.5. Analysis

The implementation of IG and PCA on IDS-IoT DBN can be seen in the previous tables. Next is to analyze the results of accuracy, precision, and recall as well as the effect of implementing IG and PCA on IDS-DBN on heterogeneous IoT networks. The results of accuracy, precision, and recall in this study can be seen in Table 7. However, there are unsatisfactory results in the Xbee dataset test. From these results, it can be observed that the best results for accuracy, precision, and recall are obtained sequentially from the TCP dataset and the last is Xbee. In addition, it can also be concluded that the results of each parameter show the results of 5 components or the number of features that produce the best performance.

Figure 2 is the result of the comparison of the accuracy of the IDS-DBN using PCA and IG. From the graph, it can be concluded that the accuracy parameter in the TCP dataset is superior to the results of using PCA than IG. However, the results of the Xbee dataset on IG show a performance that is superior to the results from PCA. Next is a comparison of measurement parameters, namely the precision of using DBN for detection methods on IoT networks. Precision is the level of accuracy of the IDS-DBN in detecting an attack.

Figure 3 is a graph of the precision comparison of the results of this research. From the graph, it can be concluded that DBN with PCA produces superior performance on TCP datasets. While the Xbee DBN dataset with IG produces better precision with a significant difference.

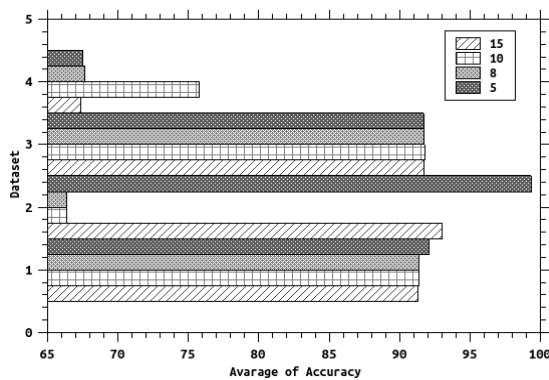


Figure 2. Result of experiment to average of accuracy

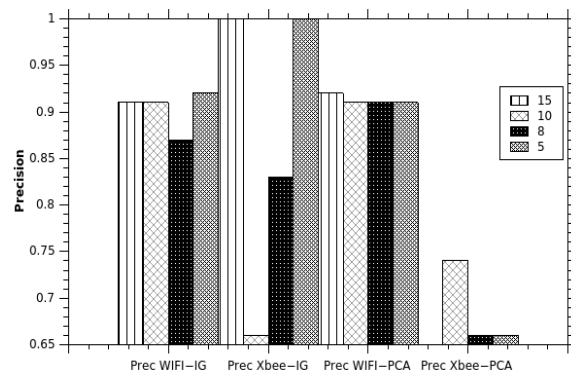


Figure 3. Result of experiment to average of precision

The last is a measurement parameter of recall from the results of using IDS-DBN in predicting an attack. Recall itself can be interpreted as the success rate of DBN in detecting an attack. Figure 4 is a comparison graph of the average recall of this study. From the graph, it can be said that PCA is superior to the TCP dataset. As for Xbee, IG produces a better recall rate.

The final conclusion from this comparison is to show that the use of feature extraction, namely PCA, shows a higher performance than the IDS-DBN test that uses feature selection, namely IG. This is due to the feature extraction method that allows converting dataset rows into smaller dimensions, without losing the characteristics of the data. While feature selection is selecting a sub-section of the overall feature. So, it will really depend on the right method to select the feature that can represent the characteristics of the data row.

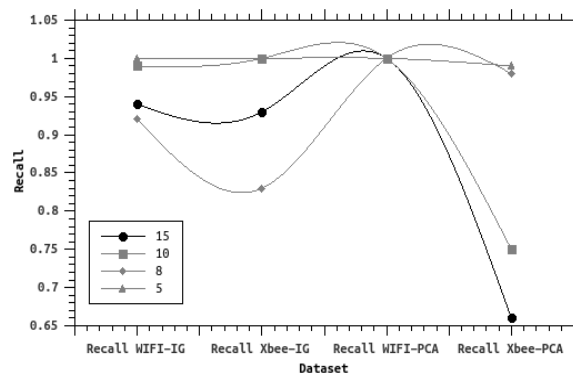


Figure 4. Average of recall

3.6. Comparison IDS-IoT method

The final step is to compare some previous studies with results of this work. Table 8 shows several IDS studies using several deep learning techniques. This comparison only compares detection accuracy results for DoS or DDoS attack types, as it is fit dataset for this work. Table 8 shows success of this work using DBN to detect attacks that occur on complex IoT networks. The result is significantly improved compared to several previous studies using other methods. DBNs with preprocessing by PCA or IG are better at detecting attacks on complex IoT networks.

Table 8. Comparison with previous method

Ref	Methods	Dataset	Result of accuracy
Zavrak and Iskefiyeli [23]	IDS-CNN	UNSW-NB15	79.03
Zhang <i>et al.</i> [24]	IDS-DNN	CICIDS2017	85,5
Haggag <i>et al.</i> [25]	IDS-AutoEncoder	NSL-KDD 5 class	87.96
Proposed method	IDS-DBN + PCA	Complex network	91,771
Proposed method	IDS-DBN + IG	Complex network	92,08

4. CONCLUSION

The increase of devices and network complexity such as protocols, end-devices, sensors, and data. (heterogenous) connected to the internet network will increase the vulnerability of these devices. One promising solution is to propose using deep learning on IDS in heterogeneous IoT networks. In this work, DBN is used as a detection method for IDS IoT. This work focuses on improving DBN performance with feature selection (IG) and feature extraction (PCA). The goal is to determine the features that will be used in the training process and determine the effect of IG/PCA for IDS-IoT. The result of this work is the DBN succeeded in detecting attacks on complex IoT networks. In this study, three measurement parameters were used: accuracy, precision, and recall. The results show that the performance of DBN with feature extraction (PCA) is superior to DBN with feature selection, namely information gain for TCP datasets. As for Xbee IG dataset, the result is superior to using PCA. The final result is the average value of accuracy, precision, and recall from each IDS-DBN testing for IoT reaching 99. Future research proposes the use of deep learning for IDS IoT with complexity networks even better. One of them proposes using other deep learning methods such as CNN, RNN, and others to identify attacks on complex IoT networks and optimize deep learning to detect attacks on Xbee datasets.

ACKNOWLEDGEMENTS




This research was supported by the University of Dinamika Bangsa through the development of human resource programs.

REFERENCES




- [1] Q. A. Al-Haija and S. Zein-Sabatto, "An efficient deep-learning-based detection and classification system for cyber-attacks in IoT," *Electronics*, vol. 9, no. 2152, 2020, doi: 10.3390/electronics9122152.
- [2] S. A. Khanday, H. Fatima, and N. Rakesh, "Implementation of intrusion detection model for DDoS attacks in lightweight IoT Networks," *Expert Systems with Applications*, vol. 215, no. November 2022, 2023, doi: 10.1016/j.eswa.2022.119330.
- [3] Sharipuddin *et al.*, "Features extraction on iot intrusion detection system using principal components analysis (PCA)," *In 2020 7th International Conference on Electrical Engineering*, vol. 2020, 2020, pp. 114-118, doi: 10.23919/EECSI50503.2020.9251292.
- [4] H. Min, J. Woo, and H. Kang, "In-vehicle network intrusion detection using deep convolutional neural network," *Vehicular Communications*, vol. 21, p. 100198, 2020, doi: 10.1016/j.vehcom.2019.100198.
- [5] M. Almiyani, A. Abughazleh, A. Al-Rahayfeh, S. Atiewi, A. Razaque, and C. Information, "Simulation modelling practice and theory deep recurrent neural network for iot intrusion detection system," *Simulation Modelling Practice and Theory*, vol. 101, no. Jul. 2019, 2020, doi: 10.1016/j.simpat.2019.102031.
- [6] Z. Lv, L. Qiao, J. Li, and H. Song, "Deep learning enabled security issues in the internet of things," *IEEE Internet Things Journal*, vol. 4662, no. 2017, pp. 1-1, 2020, doi: 10.1109/jiot.2020.3007130.
- [7] A. Thakkar and R. Lohiya, "A review on machine learning and deep learning perspectives of IDS for IoT: recent updates, security issues, and challenges," *Archives of Computational Methods in Engineering*, vol. 28, pp. 3211-3243, 2021, doi: 10.1007/s11831-020-09496-0.
- [8] 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, p. 100112, 2019, doi: 10.1016/j.iot.2019.100112.
- [9] S. Sharipuddin *et al.*, "Enhanced deep learning intrusion detection in IoT heterogeneous network with feature extraction," *Indonesian Journal of Electrical Engineering and Informatics (IJEI)*, vol. 9, no. 3, pp. 747-755, 2021, doi: 10.52549/ijeei.v9i3.3134.
- [10] Q. Wang and X. Wei, "The detection of network intrusion based on improved adaboost algorithm," in *ICCSP 2020: Proceedings of the 2020 4th International Conference on Cryptography, Security and Privacy*, no. 2, 2020, pp. 84-88, doi: 10.1145/3377644.3377660.
- [11] M. Catillo, A. Pecchia, and U. Villano, "Botnet detection in the internet of things through all-in-one deep autoencoding," in *The 17th International Conference on Availability, Reliability and Security (ARES 2022)*, 2022, pp. 23-26, doi: 10.1145/3538969.3544460.
- [12] Kurniabudi, D. Stiawan, Darmawijoyo, M. Y. B. Idris, A. M. Bamhdi, and R. Budiarto, "CICIDS-2017 dataset feature analysis with information gain for anomaly detection," *IEEE Access*, vol. 8, pp. 132911-132921, 2020, doi: 10.1109/ACCESS.2020.3009843.
- [13] Y. Cong, T. Guan, J. Cui, and X. Cheng, "LGBM: an intrusion detection scheme for resource-constrained end devices in internet of things," *Security and Communication Networks*, vol. 2022, no. 1, pp. 1-12, 2022, doi: 10.1155/2022/176165.
- [14] W. Han, J. Xue, Y. Wang, L. Huang, Z. Kong, and L. Mao, "MalDAE: detecting and explaining malware based on correlation and fusion of static and dynamic characteristics," *Computer Security*, vol. 83, pp. 208-233, 2019, doi: 10.1016/j.cose.2019.02.007.
- [15] S. Sharipuddin *et al.*, "Intrusion detection with deep learning on internet of things heterogeneous network," *International Journal of Artificial Intelligence (IJAI)*, vol. 10, no. 3, p. 735, 2021, doi: 10.11591/ijai.v10.i3.pp735-742.
- [16] S. Gavel, A. S. Raghuvanshi, and S. Tiwari, "An optimized maximum correlation based feature reduction scheme for intrusion detection in data networks," *Wireless Networks*, vol. 28, no. 6, pp. 2609-2624, 2022, doi: 10.1007/s11276-022-02988-w.
- [17] A. Verma and V. Ranga, "Machine learning based intrusion detection systems for IoT," *Wireless Personal Communications*, vol. 111, no. 4, pp. 2287-2310, 2020, doi: 10.1007/s11277-019-06986-8.
- [18] D. Protic and M. Stankovic, "A hybrid model for anomaly-based intrusion detection in complex computer networks," in *2020 21st International Arab Conference on Information Technology (ACIT)*, 2020, pp. 1-8, doi: 10.1109/ACIT50332.2020.9299965.
- [19] D. Arivudainambi, K. A. V. Kumar, and P. Visu, "Malware traffic classification using principal component analysis and artificial neural network for extreme surveillance," *Computer Communications*, vol. 147, no. June, pp. 50-57, 2019, doi: 10.1016/j.comcom.2019.08.003.
- [20] I. Sohn, "Deep belief network based intrusion detection techniques: A survey," *Expert System Application*, vol. 167, 2021, doi: 10.1016/j.eswa.2020.114170.
- [21] Y. Zhou, J. Wang, and Z. Wang, "Multisensor-based heavy machine faulty identification using sparse autoencoder-based feature fusion and deep belief network-based ensemble learning," *Jornal Sensors*, vol. 2022, pp. 1-26, 2022, doi: 10.1109/ACIT50332.2020.9299965.
- [22] 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, doi: 10.1007/s12652-020-02271-w.
- [23] 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.
- [24] J. Zhang, F. Li, and F. Ye, "An ensemble-based network intrusion detection scheme with bayesian deep learning," *IEEE In ICC 2020-2020 IEEE International Conference on Communications (ICC)*, 2020, pp. 1-6, doi: 10.1109/ICC40277.2020.9149402.
- [25] 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, pp. 163660-163672, 2020, doi: 10.1109/access.2020.3019931.

BIOGRAPHIES OF AUTHORS




Dr. Sharipuddin    received a Doctor of Engineering from Universitas Sriwijaya. He is currently a Senior Lecturer at the Faculty of Computer Science, Universitas Dinamika Bangsa, Indonesia. His research interests include information technology and information security. He can be contacted at email: sharipuddin@unama.ac.id.






Eko Arip Winanto    received the B.Sc. degree in computer science from the University of Sriwijaya, Indonesia, the M.Phil. degree in computer science from the Universiti Teknologi Malaysia, Malaysia. He is currently a Lecturer at the Faculty of Computer Science, Universitas Dinamika Bangsa, Indonesia. His research interests include IoT, machine learning, blockchain, and network security. He can be contacted at email: ekoaripwinanto@unama.ac.id.






Zulwaqar Zain Mohtar    Zulwaqar Zain Mohtar received a B.Sc. degree in Software Engineering from Universiti Teknologi Malaysia in 2019. He is currently a research officer at Media and Game Innovation Center of Excellence under Institute of Human Centered Engineering (iHumEn), Universiti Teknologi Malaysia and undergoing his Master of Philosophy in Software Engineering in Universiti Teknologi Malaysia. His research interest in Software Engineering include blockchain, distributed file system, access control and digital signature. He can be contacted at email: zzain2@live.utm.my.






Dr. Kurniabudi    received a Doctor of Engineering degree from Universitas Sriwijaya. He is currently a Senior Lecturer at the Faculty of Computer Science, Universitas Dinamika Bangsa, Indonesia. His research interests include technology adoption, information technology, information security, and network security. He can be contacted at email: kbudiz@yahoo.com.



Ibnu Sani Wijaya    received the S.Kom. degree in computer science from the University Yarsi, Indonesia, the M.S.I. degree in Megister Information System' from the Universiti Dinamika Bangsa, Indonesia. He is currently a Lecturer at the Faculty of Computer Science, Universitas Dinamika Bangsa, Indonesia. His research interests include IoT, human computer interaction, mobile computing, and expert system. He can be contacted at email: ibnu_sw17@unama.ac.id.



Dodi Sandara    received the B.Sc. degree in computer science from the STIKOM Dinamika Bangsa, Indonesia, the M.S.I. degree in computer science from the STIKOM Dinamika Bangsa, Indonesia. He is currently a Lecturer at the Faculty of Computer Science, Universitas Dinamika Bangsa, Indonesia. His research interests include information systems and network security. He can be contacted at email: doedy235@gmail.com.