Enhanced Deep Learning Intrusion Detection in IoT Heterogeneous Network with Feature Extraction

By Deris Stiawan

WORD COUNT 4469
TIME SUBMITTED 02-NOV-2021 09:47AM
PAPER ID 78362146

Indonesian Journal of Electrical and Engineering and Informatics (IJEEI)
Vol. 9, No. 3, September 2021, pp. 747~755
ISSN: 2089-3272, DOI: 10.52549/ijeei.v9i3.3134

Enhanced Deep Learning Intrusion Detection in IoT Heterogeneous Network with Feature Extraction

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

Article Info

Article history:

Received May 1, 2021
Revised Aug 14, 2021
Accepted Sep 6, 2021

Keywords:

IDS
Features Extraction
PCA
Deep Learning
Heterogeneous
IoT

ABSTRACT

Heterogeneous network is one of the challenges that must be overcome in Internet of Thing Intrusion Detection System (IoT IDS). The difficulty of the IDS significantly is caused by various devices, protocols, and services, that make the network becomes complex and difficult to monitor. Deep learning is one algorithm for classifying data with high accuracy. This research work incorporated Deep Learning into IDS for IoT heterogeneous networks. There are two concerns on IDS with deep learning in heterogeneous IoT networks, i.e.: limited resources and excessive training time. Thus, this paper uses Principle Component Analysis (PCA) as features extraction method to deal with data dimensions so that resource usage and training time will be significantly reduced. The results of the evaluation show that PCA was successful reducing resource usage with less training time of the proposed IDS with deep learning in heterogeneous networks environment. Experiment results show the proposed IDS achieve overall accuracy above 99%.

Copyright © 2021 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

Deris Stiawan,
Department of Computer Science,
Universitas Sriwijaya, Indonesia,
deris@unsri.ac.id

1. INTRODUCTION

The growth in the number of complex and diverse (heterogeneous) traffic as well as spreading of device distribution makes Internet of Things (IoT) security even more complex and challenging. In addition, the attacks detection in an IoT environment is different from detection systems on conventional networks such as resource limitations, low latency, distribution, scalability, and mobility [1]. Therefore it is necessary to design an IoT IDS that can more precisely detect attacks on heterogeneous networks. Deep learning (DL) technique is a potential candidate solution as it has features to identify small changes in a complex system.

Diro and Chilamkurti [2] state that traditional machine learning cannot detect complex intrusions, due to training process of traditional machine learning fails to identify small changes in attack scenario, more specifically, because traditional machine learning cannot extract invisible features of a dataset. In fact, attacks evolve 99% and only 1% left with similar concept. The success of deep learning technique in identifying small changes of data such as changes on pixels in image recognition shows its reliability.

There are two concerns to note on IoT Intrusion Detection System (IDS) using deep learning, i.e.: the use of resources and excessive training time. Since IoT has a limited resource, it needs to design an appropriate and optimized IDS method for IoT system with less resources consumption without scarifying the accuracy of the detection. Furthermore, as deep learning takes a relatively long time for training process, thus, a mechanism is needed to reduce the processing time to train the IoT IDS on heterogeneous networks.

Previous studies on IDS IoT have implemented deep learning techniques combined with feature extraction such as [3][4][5]. Yan and Han [3] propose deep learning as a solution to intrusion detection challenges because of its outstanding performance in handling large-scale complex data. The work uses Autoencoder stack model to perform unattended dimension reduction of intrusion detection samples. As a result, feature extraction can reduce high-dimensional dataset to its low-dimensional that in turn increases deep learning performance.

Sharipuddin *et al* [4] and Zyad *et al* [5] have discussed a hybrid Principal Component Analysis (PCA) with deep learning to improve accuracy and time detection of IDS. The proposed methods reduce the dimensions of the training data. Therefore, the training process of the deep learning model becomes faster without high resources requirement.

This research aims to propose PCA-based feature extraction method for IoT-IDS in heterogeneous network then the proposed method is combined with a deep learning technique to improve the performance of -IoT IDS performance in heterogeneous networks. PCA is used to reduce the dimensions of heterogeneous data without losing the characteristics of the original data. Thus, in turn, the feature extraction reduces the use of resource and training process time of the deep learning.

This paper is organized in 5 sections. Section 2 provides background and related work on heterogeneous network, IDS on IoT, PCA and deep learning. Section 3 presents the proposed method. Section 4 discusses experiment and results. Finally, Section 5 concludes some findings and suggests for future works.

2. RELATED WORKS

IDS with Deep Learning. DL has been implemented in many fields and one of them is network security, i.e.: IDS. The research work in [5] implements four key DL models used in IDS literature and evaluates them on four datasets: CICIDS 2017 and CICIDS 2018, KDD'99, NSL-KDD. The DL models have been chosen from the top three types of the taxonomy. They represent different methods to build DL models. First is the LSTM network that classifies sequences of flows. Second is the feed-forward neural network that classifies flow instances. Third is deep belief networks and autoencoder that are trained in a semi-supervised manner with both unlabeled as well as labeled data. This comparison in this research aims to address the difficulty of comparing models by results reported in research works due to differences in datasets and evaluation metrics.

Research work in [6] proposes Deep Neural Network (DNN) for classifying the attacks in IoT networks. The method of IDS can only be developing if there is availability of an effective dataset for the training process. The performance of DNN to classify attacks has been evaluated using several datasets such as KDD-Cup'99, NSL-KDD, and UNSW-NB15. The results show that the accuracy of the proposed method using DNN is 90%. Alrawashdeh and Purdy [7] have proposed deep learning with DNN to improve the IoT IDS by comparing it with other algorithms.

Principle Component Analysis. The selection of features is important for processes in IDS. The accuracy of an IDS changes when IDS gave different input features. IoT networks have a large amount of traffic with high dimensional features which will affect results of classification [8]. In IoT networks, IDS requires FE to reduce computation in IDS-IoT [9]. The feature extraction (FE) have been proposed to extract features of datasets from existing features and change features into small dimension to reduce training and improving accuracy [10]. The following are some of the previous studies related to the use of PCA that have been conducted [11][12][13][14][15][16][17].

Liu, et al. [11] have built a detection system for monitoring online computing for misuse detection and anomaly detection. In this work, the PCA method is applied to reduce the dimension of the dataset by combining the Artificial Neural Network (ANN) method so that it was known as PCANN. The results of experiments on DARPA dataset show accuracy of up to 98.58%.

Bharti & Singh [12] have applied a hybrid method to reduce high data dimensions with involving two stages. The first stage is to select features from the dataset and select several important ones. The result will be obtained in the form of a list of sub-dataset then the PCA method is applied to reduce the overall dimensions of the original dataset without losing a lot of information.

In research works by Hamid et al. [13], Taguchi [14], Taguchi [15], Taguchi & Murakami [16], Thaseen & Kumar [17], FE method using Principal Component Analysis (PCA) is also being applied to reduce the dataset dimension. It is not only applied to the detection system but also to other aspects. In addition, Kuang et al. [18] propose a Support Vector Machine (SVM) approach model by combining PCA with genetic algorithm (GA) for IDS. In the proposed method, hybrid-SVM is used to classify an activity as an attack or normal. The model of KPCA is used to SVM preprocessor with the aim to reduce features dimensions and training time. The function is to reduce noise caused by different features, improve SVM

performance and kernel function (N-RBF). The experimental results show that the proposed method has performed higher accuracy and faster in the detection.

3. RESEARCH METHOD

3.1 Architecture of the Proposed Method

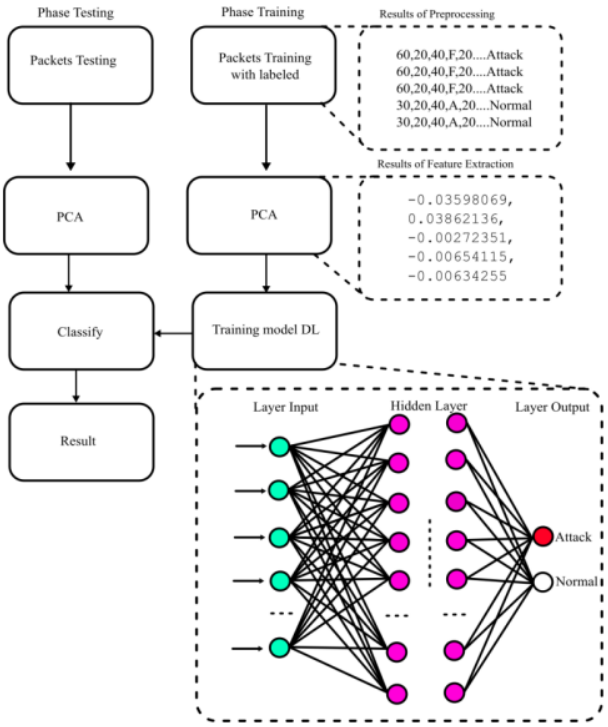

Figure 1. Design architecture IDS-DL

Figure 1 shows architecture of the proposed method to reduce the use of resources and training time of IDS with deep learning in heterogeneous IoT network. It consists of two phases namely training phase and testing phase. Prior to the phases, firstly, a preprocessing is performed, i.e.: dividing the dataset into two portions; for training and for testing. Next, is the process of reducing the dimension of the dataset, without losing its characteristics and then followed by designing IDS with deep learning model for the IoT environment. Lastly, is evaluating the IDS-Deep Learning model.

3.2 Dataset and Preprocessing

Two initial preparation works are carried out, i.e.: dataset creation and preprocessing on the created dataset. This work creates its own heterogeneous IoT dataset that consists of several devices, sensors, transport (wire, wireless), services, and protocols. Thus, the dataset represents an IoT heterogeneous network in a real environment. The hardware used include: sensors (soil moisture, MQ2, Fundulno, DHT22, etc.), devices as nodes (PC, Raspy, and Arduino). The middleware used include: XBee, w1d D1, and WIFI to connect among middleware and to server in Figure 2. The type of attack is Denial of Service.

Next, is the preprocessing on the dataset as depicted in Figure 3. This stage is required to collect the attributes (then become features) to identify patterns of the traffic packets. The dataset in Pcap files is difficult for humans to identify and to find important information (features), as they have different structures and hidden layers depend on protocols. The results of the WIFI dataset preprocessing are converted to 96 features while the XBee dataset is converted to 64 features. The details of the dataset features are shown in pseudocode in *functions def extract_xbee()* for Xbee and *def extract_wifi()* for WIFI.

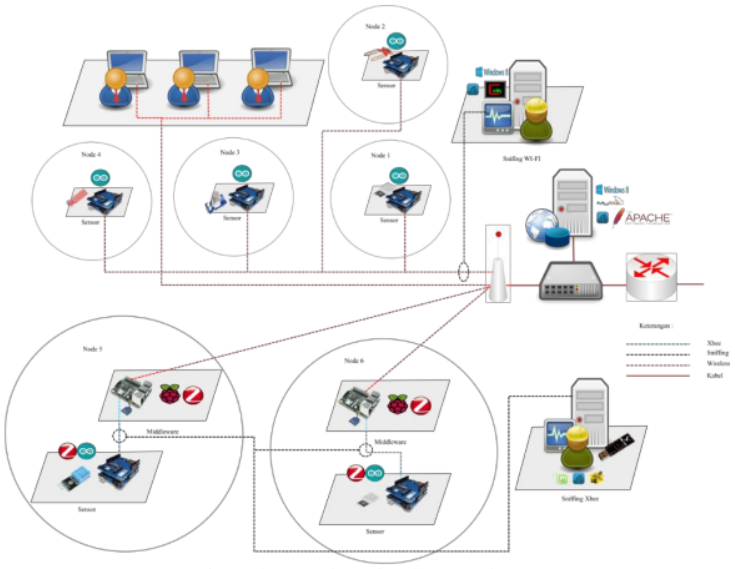

Figure 2. Topology for the experiment

Input : DtI (Dataset)
dI = {dI1, dI2,…,dIi} (i number of packets)
Output: Hd (Result of Preprocessing)
def extract_xbee(dIi)
  for i C dIi do
    data ← ppi-flags-dynamic, time-relative, len-,cap-len, marked-, data-len- encap-type, time-epoch, wpan-frame-type-, ppi-flags-reserved, ppi-common-tsft-, ppi-common-flags-fcs-, ppi-common-chan-flags-ofdm-,ppi-common-chan-flags-2ghz-,ppi-common-chan-flags-5ghz-, ppi-common-fhss-hopset, wpan-dst-addr-mode-,wpan-version-, wpan-dst-pan-, wpan-dst64-, ppi-common-flags-tsft, data-data-, time-,offset-shift, time-delta, ppi-common-flags-fcs-invalid-,ppi-common-flags-phy-err-, ppi-common-chan-freq-, ppi-common-chan-flags-turbo-, ppi-common-dbm-antsignal-,ppi,common-dbm-antnoise-,wpan-frame-length-,wpan-fcf-, ppi-dlt, , ppi-field-type, ppi-field-len-, ppi-common-fhss-pattern, wpan-src16-, wpan-src-addr-mode-, time-delta-displayed, ppi-common-rate, wpan-,security-, wpan-fcs-ok-,-, wpan-seqno-suppression-, wpan-ie-present-, wpan-pending-,wpan-ack-requestwpan-src64-,wpan-fcs-, protocol-,ppi-version-,ppi-flags-,ppi-flags-alignment-, ppi-flags-gfsk, number-, ignored-, ppi-length, ppi-common-flags-, ppi-common-chan-flags-passive-, wpan-seq-no-, ppi-common-chan-flags-, ppi-common-chan-flags-cck, wpan-pan-id-compression-,
    end
return data
def extract_wifi(dIi)
  for i C dIi do
    data ← eth-dst-, flags-urg-, flags-res-, stream, flags-cwr-, eth-addr-, time-relative-, eth-ig-, eth-src-, eth-lg-, eth-src-resolvedeth-addr-resolved-, analysis-acks-frame-, eth-lg-, eth-addr-, eth-type-, version-, eth-ig-, hdr-len-, dsfield-, flagsflags-ack-, dsfield-ecn-, len-, id-, flags-, -, flags-ns-, dsfield-dscp-,, flags-ecn-, eth-dst-resolved-, flags-rb-, flags-mf-, frag-offset-, flags-df-, ttl-, proto-, checksum-status-, src-, addr-, checksum-, src-host-, dst-, port-, analysis-initial-rtt-, host-, addr-, dstport-, flags-reset-, ws-expert-, connection-sack-, ws-expertmessage-, ws-expert-severitywindow-size-, option-kind-, checksum-, checksum-status-, urgent-pointer-, option-, option-mss, option-len-, options-mss-val-, analysis-, analysis-ack-rtt-, -, seqhdr-len-, time-relative-, ack-, host-, len-,-, flags-str-, window-size-value-, flags-syn-, , ws-expert-groupflags-fin-, dst-hostsrcport-, eth-addr-resolved-, time-epoch-, offset-shift-, time-delta-, time-delta-, encap-type-, time-delta-displayed-, cap-len-, len-, marked-, time-, protocol-, coloring-rule-name, number-, coloring-rule-string-, ignored-, port-, nxtseq-, flags-push-,
  end
return data
def main()
  dI ← read(DtI)
  for dIi C dI do
    if dIi = WIFI then
      dL ← extract_wifi(dIi)
      Hd ←dL
    end
    if dIi = WIFI then
      dL ← extract_xbee(dIi)
      Hd ←dL
    end
  end
end

Figure 3. Pseudocode for Preprocessing the Dataset

3.3 Feature extraction

This stage is a process of reducing number of data dimensions of the dataset. In this work, PCA is used as a FE method. Figure 4 is the pseudocode for the PCA designed for IoT IDS-Deep Learning. The performance of the PCA as extraction method will be evaluated using two experimental scenarios. First, the WIFI dataset consisting of 96 features is converted into 5 and 8 features. Second, the Xbee dataset, which consists of 64 features, is also converted into 5 and 8 features.

```
Input : Dtl (Dataset)
Output: Hpca (Result of PCA)
from sklearn import decomposition
data ← load_dataset(Dtl)
def main()
    Z ← read(data)
    PCA = decomposition.PCA(n_components=8)
    PCA.fit(Z)
    Z = PCA.transform(Z)
   Hpca←Z
 end
```

Figure 4. Pseudocode of Feature Extraction using PCA

3.4 Proposed Method

The proposed IDS-Deep Learning model uses Input Layer that consists of 96 entries for WIFI dataset and 64 entries for Xbee dataset and 12 nodes. The Hidden Layer consists of 8 layers and 8 nodes. The Output Layer as classifier to classify two-classes, i.e.: attack or normal. Computer used in the experiment is a notebook with hardware specification: Intel Core i7, 12GB RAM, running Ubuntu 20.04 LTS operating system. The platform used to develop IDS-Deep Learning are: TensorFlow (Keras) and Scikit-learn for the feature extraction stage. Table 1 lists the deep learning setup variable, while Figure 5 shows the pseudocode of the proposed IoT IDS with Deep learning technique.

Table 1. Variable for the Deep Learning

Variable	Value
Number of Layers	3 (1 input layer, 6 hidden layer, 1 output layer)
Node	20, 10, 10 , 2
Activation	ReLU, ReLU, ReLU, ReLU, ReLU, ReLU, ReLU, sigmoid
Input dimension	96 WIFI and 64 Xbee
Epoch	150

```
Input : Dtl (Dataset)
Output: Hdl (Result of DL)
import decomposition
 import Sequential
 import Dense
dataset = open"Dtl ",'r')
reader = dataset.readlines()
Z_train, Z_test, a_train, a_test = train_test_split(dataset,test_size=0.4, random_state=5)
AS = decomposition.PCA(n_components=8)
Z_train_z = sc.fit_transform(Z_train_z)
Z_test_Z = sc.transform(Z_test_Z)
modelPCA-DL = Sequential()
modelPCA-DL(Dense(12, input_dim=len(Z_train_z[0]), activ='relu'))
modelPCA-DL(Dense(10, activ='relu'))
modelPCA-DL(Dense(10, activ='relu'))
modelPCA-DL(Dense(10, activ='relu'))
modelPCA-DL(Dense(10, activ='relu'))
modelPCA-DL(Dense(10, activ='relu'))
modelPCA-DL(Dense(10, activ='relu'))
modelPCA-DL(Dense(1, activ='sigmoid'))
modelPCA-DL.compile()
modelPCA-DL.fit(Z_train_z, a_train1, epochs=1)
accuracy = model.predict(Z_test_Z)
print(confusion_matrix)
print(classification_report)
print(accuracy_score)
```

Figure 5. Pseudocode of the proposed IoT IDS with Deep learning

3.5 Performance Metric

This work only considers accuracy as a metric for the performance evaluation of the proposed IDS-Deep Learning model, as shown in (1).

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

4. RESULTS AND ANALYSIS

Table 2. Type of protocol and number of packets in Dataset

Protocols	Total of Packet
WIFI	
UDP	114
TCP	3132331
ARP	626
Xbee	
ZigBee	11
IEEE 802.15.4	19374
LwMesh	11
IPv6	9
6LowPAN	21

Table 2 shows the number of packets of each protocol as results of running experiments that produces 3,133,071 traffic packets. Three types of protocols, i.e.: UDP, TCP, and ARP are captured in the WIFI dataset. In other hand, Xbee dataset has 19,426 traffic packets. The preprocessing stage has obtained 95 attributes for the Wi-Fi protocol and 64 for the Xbee. Table 3 shows the results of PCA in extracting the features in the dataset from the initial attributes into 5 and 8 attributes. The results of the PCA will be converted into numeric in order to be able to be inputted into the IDS-Deep Learning.

Table 3. Result of Feature Extraction with PCA

No	Original Feature	Number of component	Result of Feature Extraction
	Extraction feature of WIFI dataset		
1	58,58,0,0,1,4,20,0,0,0,44,12923,16384,0,1,0, 0,128,6,13050,2,1,80,30569,0,80,30569,523,0, 0,0,1,24,18,0,0,0,0,0,1,0,0,2097152,33554432 ,0,13,8192,8192,7681,2,0,1,1,1,1,2,4,1460,1, 4064,0.00011,0,0.00011,0.00011,Normal	5	0.09873893,0.0869453 4,0.02496807,- 0.09827725,0.0405119 6
2	342,342,0,0,1,4,20,0,0,0,328,1474,0,0,0,0,0, 128,6,40599,2,1,30569,80,0,30569,80,523,288, 1,289,1,20,24,0,0,0,0,0,1,1,0,2097152,335544 32,0,1,5840,5840,7383,2,0,0,0,0,0,0,1,0,0,0. 004093,0.004743,0.00065,Attack	8	-0.01009107, -0.02344507, -0.03873531, 0.10514533, -0.00119132, 0.01065651, 0.03553167, 0.04040863
	Extraction feature of Xbee dataset		
3	54,54,0,0,1,4,20,0,0,0,40,9027,16384,0,1,0,0 ,128,6,16948,2,1,80,12199,0,80,12199,1000,0, 222,222,299,20,17,0,0,0,0,0,1,0,0,2097152,33 554432,1,12,64240,64240,6633,2,0,0,0,0,0,0,0 ,0,0,0,0.049065,0.00037,Attack	5	-0.03598069 ,0.03862136,- 0.00272351,- 0.00654115,- 0.00634255
4	42,42,0,0,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, 1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,,Normal	8	0.01294474, -0.07776417, 0.0351635, -0.03764663, -0.05535734, -0.05293076, -0.02199441, 0.0401086

Figure 6 shows a snapshot of experiments that have been carried out. The results of experiments have obtained several parameters as shown in Figure 6. It shows parameters such as precision, recall, f1-score, and accuracy. But in this study only the results of the accuracy are compared because it is the main parameter to measure the performance of the IDS in this work.

```
[[      0    4512]
 [      0 1249599]]
              precision    recall  f1-score   support

           0       0.00      0.00      0.00      4512
           1       1.00      1.00      1.00   1249599

    accuracy                           1.00   1254111
   macro avg       0.50      0.50      0.50   1254111
weighted avg       0.99      1.00      0.99   1254111

(99.64022323382858, '%')
```

Figure 6. A Snapshot of Running IDS-DL with PCA

Table 4. Results of Confusion Matrix of Packet Recognition by IDS-Deep Learning

	WIFI		Xbee	
	Normal	Attack	Normal	Attack
Normal	36,707	36	4593	40
Attack	10	569,457	6	678,439

Table 4 displays the results of confusion matrix IDS-DL experiment using the portion of 60% for training and 40% for testing. In WIFI dataset, 569,457 traffic packets are recognized as attacks and 36,707 packets are recognized as normal with error detection reaches 1%. Meanwhile, in XBee dataset, 678,439 packets are recognized as attacks and 4,593 packets are recognized as normal. The experiment is repeated 5 times, and the 60% for training data portion is distributed 50% for training phase and 10% for validation.

Table 5 shows the results for accuracy on the testing phase. The proposed IDS with deep learning in heterogeneous IoT network is able to detect attacks with accuracy level above 99%. These results show that the PCA increases the accuracy of IDS-Deep Learning. Figure 6 shows a comparison of experimental results on accuracy of IDS-Deep Learning with PCA and without PCA.

Table 5. Results of Testing Classification

Data distribution Training: Testing	**Accuracy (%)**	
	All	**PCA**
	WIFI	
50:50	94.96	99.3
60:40	89.95	99.4
70:30	93.91	99.3
80:20	88.07	99.6
90:10	92.93	99.3
	Xbee	
50:50	93.96	99.2
60:40	90.95	99.5
70:30	89.97	99.4
80:20	92.97	99.5
90:10	91.97	99.3

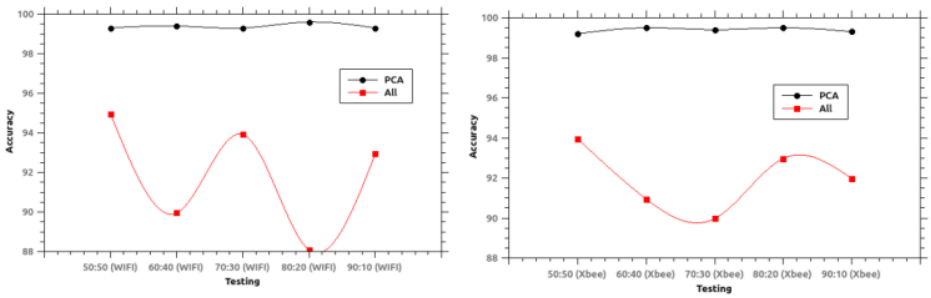

Figure 7. Result of Testing DL WIFI and Xbee

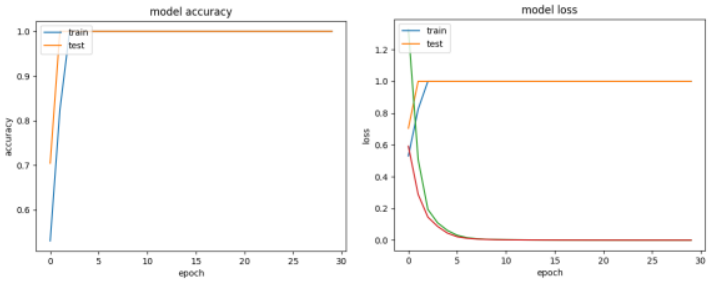

Figure 8. Graph of Model Accuracy and Model Loss of Training Process

Figure 7 are graphs of results of deep learning model training during the experiment. The image shows the comparison of testing and training accuracy. Testing of deep learning model was carried out with 30 epochs for the deep learning model training process. From the graphs, it can be seen that there several errors in initial time of training. After several training epochs, the deep learning model will stable when it reaches epoch 0 epochs to 5 epochs. Figure 8 are graphs of the results of the deep learning model training of loss. This loss model is an error training of data per epoch. The function of the loss model shows that at beginning of training there are still errors in forming a deep learning model when classifying training data.

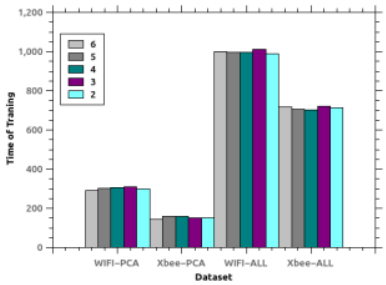

Figure 9. Graph of Execution Time During Training Process

Figure 9 is execution time comparison of the training process. Figure 9 shows there is the significant reduction in time from the training process. The IDS running on dataset with PCA is faster than the IDS running on dataset with all features (without PCA).

Table 6. Comparison Method

Method FE	Accuracy (%)
All	91.96
Factor Analysis	94.67
Non-Negative Matrix Factorization	96.97
PCA	99.39

Table 6 depicts comparisons of the accuracy of FE methods and without using FE in IDS-Deep Learning. The comparison shows accuracy of IDS-Deep Learning with PCA feature extraction produces the highest accuracy more than Factor Analysis, Non-Negative Matrix Factorization, and without feature extraction. Previous research in [19] only used the WIFI dataset for experimentation. Whereas, this research work extends the work in [19], i.e.: through introducing more complex dataset, namely WIFI and Xbee datasets. The results of this work show that the performance of IDS-DL with WIFI and Xbee datasets is not different from the previous research work. Therefore, it can be concluded that IDS-DL with feature extraction is able to enhance the performance of IDS IoT with heterogeneous networks.

5. CONCLUSION

Incorporating deep learning into IDS for IoT heterogeneous network can increase the performance of accuracy detection. The issues that need to be solved in IoT IDS with Deep Learning are limited resources and excessive training time. One of the solutions is implementation of feature extraction method in IDS IoT. This work has proposed the Principle Component Analysis (PCA) as the extraction method. The initial

results of the experiments show that the proposed IDS-Deep Learning is able to reduce the use of resources and faster training time. The experimental results show that the performance of the proposed IDS-Deep Learning increases significantly and reach accuracy level above 99%. In the near future, the authors plan to proceed with other methods on feature extraction such as Autoencoder method for automatic feature extraction.

ACKNOWLEDGMENTS

This work supported by Universitas Dinamika Bangsa and COMNETS Lab Universitas Sriwijaya.

REFERENCES

[1] H. E. Hudson, V. Forsythe, and S. G. Burns, "Fog Computing for the Internet of Things: Security and Privacy Issues," Published by the IEEE Computer Society, vol. 4, no. 2, pp. 157–161, 2017.
[2] 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.
[3] B. Yan and G. Han, "Effective Feature Extraction via Stacked Sparse Autoencoder to Improve Intrusion Detection System," IEEE Access, vol. 6, no. c, pp. 41238–41248, 2018.
[4] Sharipuddin et al., "Features Extraction on IoT Intrusion Detection System Using Principal Components Analysis (PCA)," Proc. EECSI 2020 - 1-2 October 2020, pp. 114–118, 2020.
[5] E. Zyad, A. Taha, and B. Mohammed, "Improve R2L attack detection using trimmed PCA," Proceedings - 2019 International Conference on Advanced Communication Technologies and Networking, CommNet 2019, pp. 1–5, 2019.
[6] 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, p. 102767, 2020.
[7] 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.
[8] K. Alrawashdeh and C. Purdy, "Toward an online anomaly intrusion detection system based on deep learning," Proceedings - 2016 15th IEEE International Conference on Machine Learning and Applications, ICMLA 2016, pp. 195–200, 2017.
[9] Y. N. Kunang, S. Nurmaini, D. Stiawan, A. Zarkasi, and F. Jasmir, "Automatic Features Extraction Using Autoencoder in Intrusion Detection System," Proceedings of 2018 International Conference on Electrical Engineering and Computer Science, ICECOS 2018, vol. 17, pp. 219–224, 2019.
[10] Kurniabudi, D. Stiawan, Darmawijoyo, M. Y. Bin Bin 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.
[11] G. Liu, Z. Yi, and S. Yang, "A hierarchical intrusion detection model based on the PCA neural networks," Neurocomputing, vol. 70, no. 7–9, pp. 1561–1568, 2007.
[12] K. K. Bharti and P. K. Singh, "Hybrid dimension reduction by integrating feature selection with feature extraction method for text clustering," Expert Systems with Applications, vol. 42, no. 6, pp. 3105–3114, 2015.
[13] Y. Hamid, M. Sugumaran, and L. Journaux, "A fusion of feature extraction and feature selection technique for network intrusion detection," International Journal of Security and its Applications, vol. 10, no. 8, pp. 151–158, Aug. 2016.
[14] Y. H. Taguchi, "Principal component analysis based unsupervised feature extraction applied to budding yeast temporally periodic gene expression," BioData Mining, vol. 9, no. 1, pp. 1–23, 2016.
[15] Y. H. Taguchi, "Principal components analysis based unsupervised feature extraction applied to gene expression analysis of blood from dengue haemorrhagic fever patients," Scientific Reports, vol. 7, no. August 2016, pp. 1–14, 2017.
[16] Y. H. Taguchi and Y. Murakami, "Principal Component Analysis Based Feature Extraction Approach to Identify Circulating microRNA Biomarkers," PLoS ONE, vol. 8, no. 6, 2013.
[17] I. Sumaiya Thaseen and C. Aswani Kumar, "Intrusion detection model using fusion of chi-square feature selection and multi class SVM," Journal of King Saud University - Computer and Information Sciences, vol. 29, no. 4, pp. 462–472, 2017.
[18] F. Kuang, W. Xu, and S. Zhang, "A novel hybrid KPCA and SVM with GA model for intrusion detection," Applied Soft Computing Journal, vol. 18, pp. 178–184, 2014.
[19] S. Sharipuddin *et al*., "Intrusion detection with deep learning on internet of things heterogeneous network," *IAES Int. J. Artif. Intell*., vol. 10, no. 3, p. 735, 2021, doi: 10.11591/ijai.v10.i3.pp735-742.

Enhanced Deep Learning Intrusion Detection in IoT Heterogeneous Network with Feature Extraction

ORIGINALITY REPORT

26%
SIMILARITY INDEX

MATCH ALL SOURCES (ONLY SELECTED SOURCE PRINTED)

★www.hindawi.com 9%
Internet

EXCLUDE QUOTES	OFF	EXCLUDE MATCHES	OFF
EXCLUDE BIBLIOGRAPHY	OFF		