Features Extraction on IoT Intrusion Detection System Using Principal Components Analysis (PCA)

Sharipuddin
Department of Computer Engineering Universitas Dinamika Bangsa Jambi & faculty of Engineering Universitas Sriwijaya)
Indonesia
sharipuddin@stikom-db.ac.id

Benni Purnama
Department of Computer Engineering Universitas Dinamika Bangsa Jambi & faculty of Engineering Universitas Sriwijaya)
Indonesia
bennipurnama@stikom-db.ac.id

Kurniabudi
Department of Computer Engineering Universitas Dinamika Bangsa Jambi & faculty of Engineering Universitas Sriwijaya)
Indonesia
kbudiz@stikom-db.ac.id

Eko Arip Winanto
School of Computing Universiti Teknologi Malaysia
Malaysia
ekoaripwinanto@gmail.com

Deris Stiawan
Faculty of Computer Science Universitas Sriwijay
Indonesia
deris@unsri.ac.id

Darmawijoyo Hanapi
Faculty of Mathematics and Natural Science Universitas Sriwijay
Indonesia
darmawijoyo@yahoo.com

Mohd Yazid bin Idris
School of Computing Universiti Teknologi Malaysia
Malaysia
yazid@utm.my

Rahmat Budiarto
College of Computer Science & IT) Albaha University)
Saudi Arabia
rahmat@bu.edu.sa

***Abstract*— Feature extraction solves the problem of finding the most efficient and comprehensive set of features. A Principle Component Analysis (PCA) feature extraction algorithm is applied to optimize the effectiveness of feature extraction to build an effective intrusion detection method. This paper uses the Principal Components Analysis (PCA) for features extraction on intrusion detection system with the aim to improve the accuracy and precision of the detection. The impact of features extraction to attack detection was examined. Experiments on a network traffic dataset created from an Internet of Thing (IoT) testbed network topology were conducted and the results show that the accuracy of the detection reaches 100 percent.**

Keywords—component, formatting, style, styling, insert (*key words****)***

I. INTRODUCTION

Features classification or selection is very important process in intrusion detection system (IDS), and the performance or accuracy of the IDS will change drastically when given different input features. In addition, the large number of traffic on Internet of Things creates high dimensionality of the traffic features and will affect the classification results [1].

With the increasing amount of data processed within IDS, it is necessary to perform feature extraction to reduce the computation cost in processing the raw traffic data in IoT IDS [2]. The purpose of feature extraction is to extract features of existing original features and change the features into a lower dimension structures to boost training process and improve accuracy result [3].

PCA is a method of attributes extraction, which has been already used widely and effectively in various fields such as in Bio-engineering [4], Bio-informatics [5], medical [6] as well as in in intrusion detection system [7, 8].

PCA transforms the data by fetching some linear combination data from attributes or already existing features. The thing that makes PCA distinct from other feature selection algorithm is the process that other feature selection algorithms select certain groups of the origin features while PCA extracts the features with several linear feature combinations. Once the features are transformed into its lower dimension structure, then the best principal components can be selected and use them for classification process [8].

This research implements the PCA as feature extraction algorithm on an IoT network IDS. The main goal is to investigate the impact of feature extraction process in improving the accuracy of attack detection on IoT networks. Experimental results of running IoT testbed network is compared with K- nearest neighbor (KNN) classification algorithm computation results for evaluation purpose.

II. RELATED WORKS

There are several feature extraction methods, which have been proposed by previous researchers to improve the accuracy results of IDSs. Dalmazo et al. [9] aims are to propose a new approach for detecting anomalies in cloud network traffic. The anomaly detection mechanism works on the basis of a Support Vector Machine (SVM). The key requirement for improving the accuracy of the SVM model utilizes the Poisson Moving Average predictor to the feature extraction approach and is able to handle the vast amount of information generated over time.

In [8], Hamid et al. has been proposed Principal Component Analysis (PCA) used to extract the features.

And after that on the transformed dataset Correlation-based Feature Selection (CFS) is used to select a subset of important features. The reduced dimension dataset is tested with Support Vector Machines (SVM). Obtained results demonstrate improved detection accuracy, computational efficiency with minimal false alarms, and fewer system resources utilization.

Datti et al. [10] have been proposed to compare the performance of two features reduction techniques on the dataset. These feature reduction techniques include Principal Component Analysis and Linear Discriminant Analysis. After the reduction Error Back-Propagation Algorithm is used for classification and the results show that PCA performs better than LDA.

Aburomman and Reaz [7], has been tried to maximize the effectiveness of each single feature extraction algorithm and to develop an efficient intrusion detection system, an ensemble of Linear Discriminant Analysis (LDA) and Principle Component Analysis (PCA) feature extraction algorithms are implemented. This ensemble PCA-LDA method has led to good results and showed a greater proportion of precision in comparison to a single feature extraction method.

Liu et al. [12], Thaseen et al. [13], and Kuang et al. [14] have been proposed to use the PCA to perform feature extraction to reduce training time and testing on IDS to improve the detection accuracy. They used PCA for feature extraction and conduct classify with features from PCA, results show that the mentioned approach outperformed all its competitors in terms of detection rate and computational time.

III. Research Method

This research starts with the dataset creation through the implementation of an IoT network testbed. Then capture the traffic and store them in a dataset. The PCA is implemented as the feature extraction method to obtain a dataset with lower dimension of features/attributes. Next, the feature classification is applied to the IDS to recognize attacks and measure the accuracy. Lastly, KNN is run on the same dataset as comparison. The experimental framework is seen in Fig 1 followed by a description for each point.

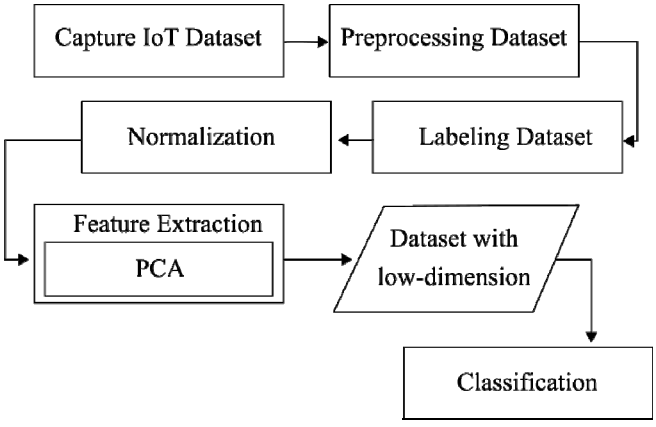

Fig. 1. Experimental Frameworks

A. Principle Component Analysis

The main objective of the PCA is to minimize the dimensionality of the dataset while preserving as much variation as possible in the datasets. The popularity of PCA derives from three main properties. First, it is the optimal (in terms of mean squared error) linear scheme for compressing a set of high dimensional vectors into a set of lower dimensional vectors and then reconstructing the original set. Second, the model parameters can be computed directly from the data - for example by diagonalzing the sample covariance matrix. Third, compression and decompression are easy operations to perform given the model parameters - they require only matrix multiplication [8].

B. Testbed Topology

One of the challenges in IoT IDS research is the lack of public available dataset for testing process. To overcome this challenge, we set up a topology as shown in Figure 1 for the testbed network in order to create a dataset. The testbed network consists of devices include DHT22 sensor, MQ2 sensor, Fundulno sensor, soil moisture sensor and other types of sensors. Several nodes are linked through middleware1 that uses XBee version 1, other nodes are linked through middleware2 that uses XBee version 2, wemos D1 and wireless routers as the connecting device between middleware and a PC acts as a monitoring.

C. Data Capturing

Data capture is performed to generate two types of dataset, i.e.: normal dataset and anomaly or attack dataset.

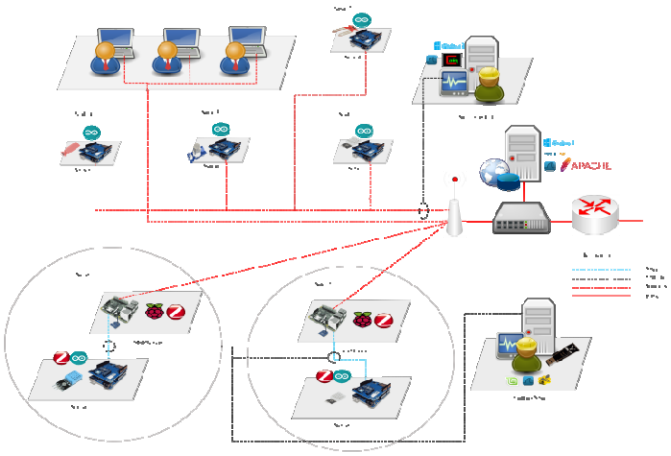

Fig. 2. Testbed Topology

Attack type traffic data used in this work is Denial of Service (DoS) attack. Normal packet pattern and DoS packet pattern of the obtained data will be analyzed through their attributes [15], so manually is able to distinguish the normal data from anomaly data or attack. Data capture is performed by using *Wireshark*/tcpdump tool for 5 minutes duration and hping3 tool is used to simulate DoS attack. The sniffed packets are stored into log file.

D. Data Preprocessing

Having done capturing the packets from the traffic using *Wireshark*, then preprocessing on acquired dataset is performed. The sniffed packets resulted by *Wireshark* are in *.Pcap* format, which are unreadable by human. Thus, firstly, this type of data should be converting into comma separated value (CSV) file format and labeled. Next, is to normalize

the data, because the features/attributes have different type of data (numeric, IP address, characters, etc.) or have redundant [16]. Later on, relevant features will be selected as input to PCA features extraction process. Not all of the created features are selected, as an example; the 'time' attribute is an unselected attribute for PCA feature extraction process. Figure 2 shows the flowchart of preprocessing.

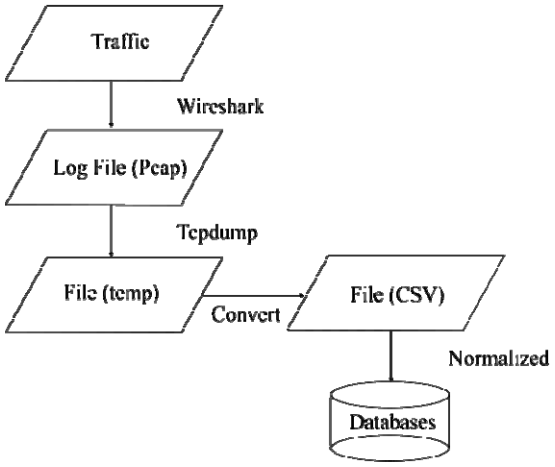

Fig. 3. Flowchart of preprocessing

E. Classification

In this experiment, we use K-nearest neighbor (KNN) classification algorithm to evaluate the impact of feature extraction result in the form of two classes, attack class and normal class. KNN [17] is a method for classification of objects based on the training data, which is closest to the object. Figure 3 illustrates how KNN works with the value of K=3.

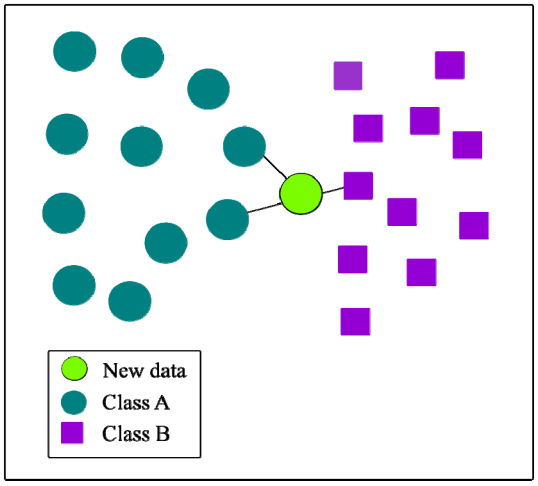

Fig. 4. Illustration of KNN for k =3

Distance calculation in K-nearest neighbor usually uses the Euclidean formula, where for two n-feature position, for example X = (x_1, x_2 ,..., x_n) and Y = (y_1, y_2, ..., y_n), then Euclidean distance between data can be calculated as:

$$dist(X,Y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_n - y_n)^2}$$
$$= \sqrt{1 \cdot (x_1 - y_1)^2 + 1 \cdot (x_2 - y_2)^2 + \cdots + 1 \cdot (x_n - y_n)^2} \quad (1)$$

IV. Experimental Results and Discussion

This section discusses about testing result which have been done. In testing topology testbed there are two types of packet data, i.e.: normal and attack with time observation for five minutes.

TABLE I. Number Of Packets In The Created Dataset

Protocol	Number of Packet Data
UDP	139
TCP	1212329
ARP	832

Table 1 show the number of packets resulted from the experiment. Overall, the experiment provides 1,213,299 packets with three types of protocols: UDP, TCP and ARP. It consists of 1,139,179 attack packets and 74,121 normal packets.

Table 2 shows the attributes obtained from the preprocessing stage. The results obtained from this process are 95 attributes for Wi-Fi protocol.

TABLE II. Result Of Reading The Attributes

Attributes
frame.encap_type,frame.time,frame.offset_shift,frame.time_epoch,frame.time_delta,frame.time_delta_displayed,frame.time_relative,frame.number,frame.len,frame.cap_len,frame.marked,frame.ignored,frame.protocols,frame.coloring_rule.name,frame.coloring_rule.string,eth.dst,eth.dst_resolved,eth.addr,eth.addr_resolved,eth.lg,eth.ig,eth.src,eth.src_resolved,eth.addr,eth.addr_resolved,eth.lg,eth.ig,eth.type,ip.version,ip.hdr_len,ip.dsfield,ip.dsfield.dscp,ip.dsfield.ecn,ip.len,ip.id,ip.flags,ip.flags.rb,ip.flags.df,ip.flags.mf,ip.frag_offset,ip.ttl,ip.proto,ip.checksum,ip.checksum.status,ip.src,ip.addr,ip.src_host,ip.host,ip.dst,ip.addr,ip.dst_host,ip.host,tcp.srcport,tcp.dstport,tcp.port,tcp.port,tcp.stream,tcp.len,tcp.seq,tcp.nxtseq,tcp.ack,tcp.hdr_len,tcp.flags,tcp.flags.res,tcp.flags.ns,tcp.flags.cwr,tcp.flags.ecn,tcp.flags.urg,tcp.flags.ack,tcp.flags.push,tcp.flags.reset,tcp.flags.syn,_ws.expert,tcp.connection.sack,_ws.expert.message,_ws.expert.severity,_ws.expert.group,tcp.flags.fin,tcp.flags.str,tcp.window_size_value,tcp.window_size,tcp.checksum,tcp.checksum.status,tcp.urgent_pointer,tcp.options,tcp.options.mss,tcp.option_kind,tcp.option_len,tcp.options.mss_val,tcp.analysis,tcp.analysis.acks_frame,tcp.analysis.ack_rtt,tcp.analysis.initial_rtt,tcp.time_relative,tcp.time_delta

The attributes in Figure 2 will be normalized. The goal is to remove irrelevant attributes for feature extraction process and to reduce it into 62 attributes.

TABLE III. Result Of Normalization

Attributes
54,54,0,0,1,4,20,0,0,0,40,80,0,0,0,0,0,64,6,58657,2,1,5340,80,0,5340,80,3463,0,1,1,1997724222,20,1,0,0,0,0,0,0,0,0,4194304,2097152,150994944,33554432,1,2,512,512,45206,2,0,0,0,0,0,0,0,0,0,0,0,0,0,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

Table 3 shows the result of attribute normalization process and then the non-numeric attributes will be converted into numeric values first. After that, the data will be transformed using PCA into 10 components or 10 attributes and able to be used into next step of classification.

TABLE IV. RESULT OF TESTING CLASSIFICATION KNN

Training: Testing	Accuracy (%)		Precision	
	Original	PCA	Original	PCA
50:50	99.96	100	1.00	1.00
60:40	99.95	100	1.00	1.00
70:30	99.97	100	1.00	1.00
80:20	99.97	100	1.00	1.00
90:10	99.97	100	1.00	1.00

Table 4 shows the results of the use of attack detection using KNN on the filtered/normalized dataset. For different combinations of training and testing data, the accuracy of KNN is 100%. Besides, the classification processing time is lesser, because the process of distance calculation between attributes is smaller. Figure 5 depicts a snapshot of running program of the KNN classification using PCA.

```
Number of Data Testing : 606164
Number of Data Training : 606164
Resum Parameter :
 KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
           metric_params=None, n_jobs=None, n_neighbors=3, p=2,
           weights='uniform')
Confusion Matrix :
 [[569721      0]
 [     0  36443]]
classification_report :
              precision    recall  f1-score   support

      Attack       1.00      1.00      1.00    569721
      Normal       1.00      1.00      1.00     36443

   micro avg       1.00      1.00      1.00    606164
   macro avg       1.00      1.00      1.00    606164
weighted avg       1.00      1.00      1.00    606164

Akurasi : 100.0 %
```

Fig. 5. Snapshot of running program KNN classification with PCA

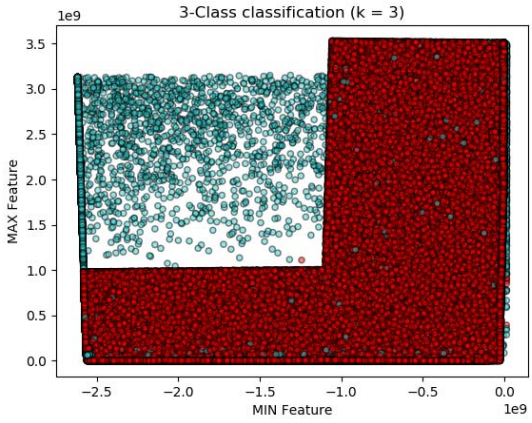

Fig. 6. Result classification KNN without PCA

Figure 6 and 7 show different attacks and normal data classification results using PCA and without PCA. In Figure 6, for the result of classification without PCA it is observed that there is no class difference between normal and attack data. While in Figure 7, the result of classification using PCA, it is clearly shown that there is vibrant class division, red Dot is plot for attack data while the blue dot is normal data.

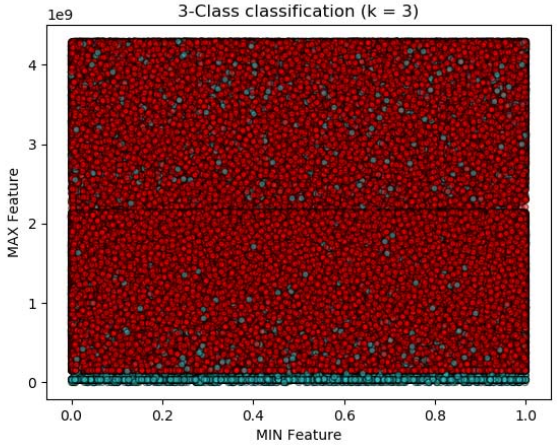

Fig. 7. Result classification KNN with PCA

TABLE V. FEATURE EXTRACTION METHOD COMPARISON

Method	Accuracy
Original	99.91
Factor Analysis	99.97
Non-Negative Matrix Factorization	99.979
PCA	100

In table 5 is a comparison of the results of the accuracy of several feature extraction methods and without using feature extraction. The comparison results show that accuracy with PCA produces the highest accuracy than FA, NMF, and without feature extraction.

V. CONCLUSION AND FUTURE WORKS

Experiments in this research work showed that the PCA significantly improve the accuracy and precision of attack detection on IoT network. The accuracy reached up to 100% for various combinations of training data and testing data allocation. Involving more attack types, and more complicated IoT network topology are considered as future works of this research.

ACKNOWLEDGMENT

This research is supported by Universitas Dinamika Bangsa through human resource development programs and collaboration with Comnet Lab Universitas Sriwijaya.

REFERENCES

[1] 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.

[2] Y. N. Kunang, S. Nurmaini, D. Stiawan, A. Zarkasi, and F. Jasmir, "Automatic Features Extraction Using Autoencoder in Intrusion Detection System," *Proc. 2018 Int. Conf. Electr. Eng. Comput. Sci. ICECOS 2018*, vol. 17, pp. 219–224, 2019.

[3] Kurniabudi, B. Purnama, Sharipuddin, D. Stiawan, Darmawijoyo, and R. Budiarto, "Preprocessing and Framework for Unsupervised Anomaly Detection in IoT: Work on Progress," *Proc. 2018 Int. Conf. Electr. Eng. Comput. Sci. ICECOS 2018*, pp. 345–350, 2019.

[4] Y. H. Taguchi, "Principal component analysis based unsupervised feature extraction applied to budding yeast temporally periodic gene expression," *BioData Min.*, vol. 9, no. 1, pp. 1–23, 2016.

[5] 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.

[6] Y. H. Taguchi, “Principal components analysis based unsupervised feature extraction applied to gene expression analysis of blood from dengue haemorrhagic fever patients,” *Sci. Rep.*, vol. 7, no. August 2016, pp. 1–14, 2017.

[7] A. A. Aburomman and M. B. I. Reaz, “Ensemble of binary SVM classifiers based on PCA and LDA feature extraction for intrusion detection,” *Proc. 2016 IEEE Adv. Inf. Manag. Commun. Electron. Autom. Control Conf. IMCEC 2016*, pp. 636–640, 2017.

[8] Y. Hamid, M. Sugumaran, and L. Journaux, “A fusion of feature extraction and feature selection technique for network intrusion detection,” *Int. J. Secur. its Appl.*, vol. 10, no. 8, pp. 151–158, Aug. 2016.

[9] M. C. Bruno L Dalmazo, Jo˜ao P. Vilela, Paulo Sim˜oes, “Expedite Feature Extraction for Enhanced Cloud Anomaly Detection,” *IEEE/IFIP NOMS 2016 Work. 2nd Work. Secur. Emerg. Distrib. Netw. Technol. (DISSECT*, vol. 21, no. 3, pp. 284–290, May 2016.

[10] R. Datti and S. Lakhina, “Performance Comparison of Features Reduction Techniques for Intrusion Detection System,” vol. 8491, pp. 332–335, 2012.

[11] N. Yala, B. Fergani, and A. Fleury, “Towards improving feature extraction and classification for activity recognition on streaming data,” *J. Ambient Intell. Humaniz. Comput.*, vol. 8, no. 2, pp. 177–189, 2017.

[12] 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.

[13] I. S. Thaseen and C. A. Kumar, “Intrusion detection model using fusion of PCA and optimized SVM,” *Proc. 2014 Int. Conf. Contemp. Comput. Informatics, IC3I 2014*, pp. 879–884, 2014.

[14] F. Kuang, W. Xu, and S. Zhang, “A novel hybrid KPCA and SVM with GA model for intrusion detection,” *Appl. Soft Comput. J.*, vol. 18, pp. 178–184, 2014.

[15] Sharipuddin *et al.*, “Measurement of Component Performance (Sensor) on Internet of Thing (IoT),” *Proc. 2018 Int. Conf. Electr. Eng. Comput. Sci. ICECOS 2018*, pp. 339–344, 2019.

[16] D. Stiawan, M. Y. Idris, R. F. Malik, S. Nurmaini, N. Alsharif, and R. Budiarto, “Investigating Brute Force Attack Patterns in IoT Network,” *J. Electr. Comput. Eng.*, vol. 2019, pp. 1–13, 2019.

[17] M. Y. Su, “Real-time anomaly detection systems for Denial-of-Service attacks by weighted k-nearest-neighbor classifiers,” *Expert Syst. Appl.*, vol. 38, no. 4, pp. 3492–3498, 2011.