Dimensionality reduction with Fast ICA for IoT Botnet Detection

By Deris Stiawan

WORD COUNT	8680	TIME SUBMITTED	05-MAY-2022 12:23AM
		PAPER ID	85555719

Dimensionality reduction with Fast ICA for IoT Botnet Detection

Susanto[a,b], Deris Stiawan*[c], Dian Palupi Rini[c], M. Agus Syamsul Arifin[a,b], Mohd Yazid Idris[d], Nizar Alsharif[e], and Rahmat Budiarto[e]

[a]Faculty of Engineering Universitas, Sriwijaya, Palembang, Indonesia; [b]Faculty of Computer, Universitas Bina Insan, Lubuklinggau, Indonesia; [c]Department of Computer Engineering, Faculty of Computer Science, Universitas Sriwijaya, Palembang, Indonesia; [d]School of Computing, Faculty of Engineering, Universiti Teknologi, Johor Bharu, Malaysia; [e]College of Computer Science and IT, Albaha University, Saudi Arabia.

Correspondence:

Deris Stiawan, Department of Computer Engineering, Faculty of Computer Science, Universitas Sriwijaya, Palembang, Indonesia.

Email: deris@unsri.ac.id

Abstract

Internet of Things (IoT) has unique characteristic with a minimalist design, and has network access with great scalability, which makes difficulty in controlling the access.. Setting up Intrusion Detection System (IDS) on an IoT system with taking into account its unique characteristic is a big challenge. In this paper, we propose a dimensional reduction approach utilizing the fast Independent Component Analysis (fast ICA) method to address scalability issues of IDS for IoT systems. The experimental results show that the reduction of dimensions by fast ICA method overall improves the IDS execution time and does not significantly affect accuracy.

Keyword
IoT Botnet, IDS, Dimensionality reduction, Fast ICA, Scalability

I. Introduction

Intrusion detection system (IDS) is one of the effective techniques to protect data and networks (Aljanabi, Ismail and Ali, 2021). Constructing the IDS has various kinds of such as difficulties in estimating the distribution of high dimensional data (Khraisat *et al*., 2019); data is high-scale and has high redundancy (Wenjuan Wang *et al*., 2020); maximizes accuracy by minimizing false alarms (Thaseen and Kumar, 2016); and high false-positive rate (Moustakidis and Karlsson, 2020). To address the challenges, researchers use dimensional reduction technique.

Dimensionality reduction is a process of projecting high-dimensional data into lower-dimensional data (Sarveniazi, 2014). Vlachos (2011) explains that Dimensionality reduction is a process of mapping data from n-dimension, into the lower-k-dimensional data space or what is called a data compression method. Then this process encourages the visualization of data in two or three dimensions. Furthermore, the dimensionality reduction technique removes irrelevant and redundant features and increases processing speed which means reducing the execution time (Ayesha, Hanif and Talib, 2020).

Dimensionality reduction techniques, such as: fast ICA (Minguan *et al*., 2015), Principal Component Analysis (PCA) (Sipola, Juvonen and Lehtonen, 2012; Syarif, Prugel-Bennett and Wills, 2012; Platonov and Semenov, 2014; Daniel Perez *et al*., 2019; Zong, Chow and Susilo, 2019), Multidimensional Scaling (MDS) (Nziga, 2011), Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO) (Alsaadi *et al*., 2020) have been widely used. In botnet detection research, Jabbar and Mohammed (2020) use Correlation Attributes Evaluation and PCA in reducing the dimension of CIC-IDS2017 dataset and apply an ensemble classifier. Furthermore, Alshamkhany *et al*. (2020) maximize the function of the Naïve Bayes algorithm, K-Nearest Neighbor, Support Vector Machine, and Decision Trees by reducing data dimension on the UNSW-NB15 dataset using PCA for detecting botnets. Popoola *et al*. (2021) reduce high-scale dimensions to low dimensions on IoT networks on the Bot-IOT dataset using an autoencoder to detect IoT botnets. Then, Bahsi, Nomm and La Torre (2018) minimize the number of features in the classification by using Fisher interpretable score to improve the classification accuracy of k-NN method and decision tree on N-BaIoT dataset for detecting IoT botnets. The proposed method shows that fewer features can achieve a very high degree of accuracy and provide interpretable results with a decision tree classifier. As mentioned by Velliangiri, Alagumuthukrishnan and Thankumar Joseph (2019), the reduction of dimensions in the dataset can be beneficial for smaller data storage, less computation time, removal of redundant, irrelevant data, improvement of data quality, help the algorithm performs efficiently and improves accuracy, make it easier to visualize the data, and simplify the classification process and increase efficiency. In the detection of IoT botnets that have a minimalist design, there are still few researchers who consider storage space, computation time, and simplification of the classification process.

This study uses N-BaIoT dataset, which has a large amount of traffic data and a large number of features that must be considered. To show the workableness of the proposed approach, it is also implemented on MedBIoT dataset. We use fast ICA dimensionality reduction method in reducing high-scale to low-scale data and removing redundant then continue with classification using AdaBoost (AB), k-Nearest neighbor (k-NN), Random Forest (RF), Decision Tree (DT), Gradient Boosting (GB), and Naïve Bayes (NB) to detect IoT botnets. Lastly, the performance of the detection results is evaluated in order to determine the effects of dimensional reduction. Fast ICA is chosen because it has effective performance (Kasturiwale and Mizwan, 2014), efficiencies (Huang and Zeng, 2012) and it also has the best performance (Chen and Bickel, 2006).

This research work aims to find out a minimum size of N-BaIoT dataset that yet provide high accuracy as well as low false alarm through a comparison with IDS without data dimensional reduction procedure. Thus, this work contributes towards minimizing the number of network information required for detecting IoT intrusion while preserving high accuracy and low false alarm.

The rest of this paper is structured as follows. In Section 2, we present and summarize related works in dimensionality reduction on IDS. Section 3 explains the dataset, experimental setup, fast ICA, classification algorithm, and analysis tool. Section 4 presents the experiment results along with analysis. Section 5 presents the conclusions and further work on dimensionality reduction of botnet IoT dataset.

II. Related work

Research works on dimensionality reduction have been carried out, especially in intrusion detection systems. Zhao et al. (Zhao, Li, Zia and Albert Y. Zomaya, 2017) reduce the dimensions of the dataset from large amounts of data to small amounts of data by using Principal Component Analysis (PCA). The PCA-Softmax regression combination results in low computational complexity. Abbas (2017) reduces features and removes redundant data and irrelevant features using PCA and singular value decomposition (SVD) algorithms. The algorithm contributes to minimizing memory used and execution time. Zheng and Zhou (2017) perform an analysis in improving the functionality of PCA method. Experiment result gives a better effect of reduction in dimensions at an accuracy of 99.7689%.

Hamid *et al*. (2017) mitigate the curse of dimensionality using the t-SNE nonlinear dimensionality reduction method. The proposed method shows the effectiveness in reducing the dimensions provided that the target dimensions are not too low, to prevent the classes from collapsing with each other. Zhang *et al*. (2018) perform an analysis of IDS using an improved PCA. The improved PCA method can reduce the detection time up to 60%, shorten the detection time to 0.5 seconds and increase the accuracy up to 91.06%. The detection accuracy value was 86% with cross-validation. Nomm and Bahsi (2019) perform an analysis on reduction of the number of features using Hopkins statistics, entropy, and variance methods. The proposed technique shows high accuracy in the unsupervised learning model. Abdulhammed *et al*. (2019) reduce features numbers using Autoencoder technique and PCA. The proposed technique shows better performance in terms of Detection Rate (DR), F-Measure, False Alarm Rate (FAR), and Accuracy level of 99.6%.

Salo, Nassif and Essex (2019) handle redundant data and irrelevant features by combining the information gain and PCA method. The proposed approach contributes to providing more important features in achieving high accuracy and low false alarms. Pajouh *et al*. (2019) analyze the reduction of high to low data dimensions using two-layer dimensionality reduction PCA and Linear Discriminant Analysis (LDA) methods. The proposed method improves intrusive activity detection performance. Sreenivasa Chakravarthi and Jagadeesh Kannan (2019) reduce the high data dimension to the low data dimension using an autoencoder technique. The proposed technique contributes to reducing the false alarm rate.

Table 1 summarizes the research works on dimensionality reduction in intrusion detection systems over the past five (5) years. After reviewing the results of previous studies, we conclude that dimensionality reduction can reduce the false alarm rate, produce low computational complexity, through removal of redundant data and irrelevant features.

Table 1. Summary of related work

(Author & Year)	Dataset (feature#)	DR Method	Result
(Zhao, Li, Zia and Albert Y. Zomaya, 2017)	KDD CUP 99 (41)	PCA	Able to optimally identify benign & malware behaviors
(Abbas, 2017)	KDD CUP 99 (41)	PCA and SVD	Has a high detection rate
(Zheng and Zhou, 2017)	KDD CUP 99 (41)	PCA	Has a better effect in intrusion detection
(Hamid *et al*., 2017)	KDD CUP 99 (41)	t-SNE	Effective in intrusion detection systems
(Bahsi, Nomm and La Torre, 2018)	N-BaIoT (115)	Fisher's scores	The Decision Tree classification can be easily interpreted and has high accuracy
(Zhang *et al*., 2018)	KDD CUP 99 (41)	PCA	Has a detection rate of 86%.
(Nomm and Bahsi, 2019)	N-BaIoT (115)	Hopkins statistics, entropy & variance	Has low computational complexity
(Abdulhammed *et al*., 2019)	CICIDS 2017 (82)	PCA, Autoencoder	Promotes better erformance in multiple metrics as well as classification speed
(Salo, Nassif and Essex, 2019)	ISCX2012 (19), NSL-KDD (41), Kyoto 2006 (24)	Hybrid IG-PCA	Contributes more important & significant feature, achieving high accuracy and low false alarm rate
(Pajouh *et al*., 2019)	NSL-KDD (41)	PCA, LDA	Accurate in distinguishing between attack types and normal
(Sreenivasa Chakravarthi and Jagadeesh Kannan, 2019)	UNSW-NB15 (47)	Autoencoder	Reduces the false alarm rate from anomaly detection
(Mutlaq, Madhi and Kareem, 2020)	NSL-KDD (41)	Genetic Algorithm	Improve the accuracy of the IDS
(Andalib and Vakili, 2020)	UNSW-NB15 (47), NSL-KDD(41) and TON IoT20 (44)	Autoencoder	There was no decrease in accuracy in IDS detection
(Alsaadi *et al*., 2020)	NSL-KDD (41)	ACO and PSO	Efficient at finding optimal feature subsets and reducing redundant, useless and irrelevant features.

Mutlaq, Madhi and Kareem (2020) reduce the data dimension by using genetic algorithm. The proposed algorithm contributes to producing a subset of relevant features. Andalib and Vakili (2020) reduce the dataset using Autoencoder. The proposed method is used to reduce bandwidth and overhead. Alsaadi *et al*. (2020) handle the problem of high data dimensions using the Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO) algorithms. The proposed algorithm contributes to generating the relevant feature subsets.

III. Proposed methodology

This section describes the N-BaIoT dataset, fast ICA, classification algorithms, experimental setup, and analysis tool.

A. Dataset

This study uses comma-separated value (CSV) data of N-BaIoT dataset (Meidan *et al*., 2018) from the University California Irvine (UCI) Machine Learning repository, which was extracted using statistical methods (Mirsky *et al*., 2018). The N-BaIoT dataset is taken from nine (9) IoT devices, each in the form of a file named after the attack label with a comma-separated value.

N-BaIoT dataset was selected with consideration of large number of features (115 features) and large number of records (7,062,606 records). In addition, it is also up to date, as can be seen the number was only 6,273,053 records when it used by Bahsi et al.'s work, conducted in 2018 (Bahsi, Nomm and La Torre, 2018). Moreover, N-BaIoT dataset is the only real dataset that has been used in literatures to detect IoT botnet attacks (Al Shorman, Faris and Aljarah, 2019). In detail, the attack data traffic (label/filename) shown in the second column of Table 2, consists of normal traffic (benign) and attack traffic with 10 types of attacks.

Table 2. Data distribution in N-BaIoT dataset of new label on 20%

New Label Feature	Label File	Number of Data
Benign	Benign	111179
	Combo	103030
	Junk	52158
Bashlite	Scan	51022
	Tcp	171969
	Udp	192873
	Ack	128764
	Scan	107596
Mirai	Syn	146660
	Udp	246001
	Udpplain	104660
Total		1415912

B. Fast ICA

Fast ICA is a variant of Independent Component Analysis methods which is based on a point iteration scheme (A. Hyvarinen and E. Oja, 2000). Fast ICA has its simplicity and fast convergence involving preprocessing and fixed-point iteration schemes (Acharya and Panda, 2008). The following is the basic formula of the fast ICA algorithm for one unit.

1. Choose an initial random weight vector w.
2. Let $w^{+} = E\{xg(w^{T}x)\} - E\{g^{l}(w^{T}x)\}w$ (1)
3. Let $w = w^{+}/||w^{+}||$ (2)
4. If not convergence, go back to 2.

Note: convergence means that the old and new values of w point in the same direction. Then the maximum estimation of the negentropy $w^{T}x$ is obtained at a given optimum of $E\{g^{l}(w^{T}x)\}$. While the basic formulae of the fast ICA algorithm for several units are as follows.

1. Let $w = w/\sqrt{||ww^{T}||}$ (3)

Repeat step (2) until convergence.

2. Let $w = 3/2(w) - ½(ww^{T}w)$ (4)

Where w is the matrix $(w_1,\ldots w_n)^{T}$ of the vectors. The formula in (1) can all be norm matrices, for example, the number of matrices of two rows (or columns) is the absolute largest (but not the Frobenius norm).

Xin *et al*. (2009) use the fast ICA method to reduce the dimensions of the hyperspectral image. Then, Fang *et al*. (2015) analyze the fast ICA method and improve it to reduce the dimensions of the hyperspectral image.

C. Classification algorithm

Machine learning-based IDS systems with classification algorithms are being implemented as a potential solution for detecting intrusions throughout the network in an efficient manner (Ahmad *et al*., 2020). Classifying algorithms in machine learning are algorithms that learn from training datasets which then assign new data points to certain classes in predicting class labels with the help of mapping functions for a new data entry (Sen, Hajra and Ghosh, 2020). We use six classification algorithms, i.e.: AB, k-NN, RF, DT, GB, and NB. These classification algorithms are used to investigate the impact of the fast ICA implementation on features number reduction on IDS performances.

1. Adaboost

AdaBoost is a classification method in machine learning, work by building a global and optimal combination of weak classifiers based on reweighting samples (Wu and Nagahashi, 2015). Mazini et al. (Mazini, Shirazi and Mahdavi, 2019) use AB classification to obtain a high detection rate (DR) with a low false-positive rate (FPR) for network-based anomalies detection. Then, Hu and Hu (2005) build a network-based intrusion detection system using AB. This classification method contributes to a very low false-positive rate while keeping high detection rate and very low computation complexity.

2. K-Nearest Neighbor

Based on survey paper by Bhatia and Vandana (2010), k-NN is divided into two categories namely, Structure-less and Structure-based. In the structure less category, the k-closest neighbor is in the first category where all data are classified into training data and sample data points. The distance is evaluated from all practice points to the sample point and the point with the lowest distance is called the nearest neighbor. This technique is very easy to implement, however the value of k affects the results in some cases. Whereas in the Structure-Based category, the second category of closest neighbor technique is based on data structures such as the Ball Tree, KD Tree, Principal Axis Tree (PAT), Orthogonal Structure Tree (OST), Nearest Feature Line (NFL), Center Line (CL) and etc. This technique increases the speed of classification k-NN performance. The leaves of the tree contain relevant information and internal nodes are used to guide efficient searches through the foliage. The k-dimensional tree divides the training data into two parts, a right node and a left node. The left or right side of the tree is searched by query records. Upon reaching the terminal node, the records at the terminal node are checked to find the closest data node to the query record.

In IDS, k-NN is used to separate abnormal behavior from normal behavior, and to analyze the parameter selection and error rate of the intrusion detection system. This method has high accuracy and detection speed (Li *et al*., 2014). Furthermore, Liao and Vemuri (2002) use k-NN to classify normal or intrusive behavior. The proposed method shows that the k-NN classifier can effectively detect intrusive attacks and achieve a low false-positive ratio.

3. Random forest

The random forest is an ensemble classifier that generates multiple decision trees, using a randomly selected subset of samples and training variables (Negandhi, Trivedi and Mangrulkar, 2019). Farnaaz and Jabbar (2016) have developed a model of an intrusion detection system using random forest. The proposed model shows its efficiency with a low false alarm rate and high detection rate. In another study, Negandhi, Trivedi and Mangrulkar (2019) increase IDS security by also using Random Forest algorithm. The proposed IDS not only runs faster but also has higher accuracy.

4. Gradient boosting

The Gradient Boosting learning procedure sequentially adjusts the new model to provide a more accurate estimation of the response variable by constructing a new basic learner so that it is maximally correlated with the negative gradient of the loss function, which is related to the entire ensemble. The applied loss function can be arbitrary, however to provide better intuition, if the error function is a classical error-squared loss, the learning procedure will result in sequential error adjustment (Natekin and Knoll, 2013).

5. Naïve bayes

Naïve Bayes is a classification algorithm based on the Bayes theorem with the assumption of strong independence and the features are independent of the given class (Kaviani and Dhotre, 2017). Mukherjee and Sharma (2012) apply the Naïve Bayes classification method on a reduced dataset to produce an efficient and effective IDS. Furthermore, Gujar and Patil (2014) use the Naïve Bayes classification to construct an IDS that classifies whether an attack exists or not. The proposed method shows good performance for real-time data.

D. Hardware and software

The experiments are conducted through simulations on a computer with specification of a 9th gen. Intel core i7 processor, 16GB DDR4 RAM, 512GB SSD, and NVIDIA GTX1660 Ti GPU. The computer runs Windows 10 operating system. We develop analysis tool using Python 3.7.4 programming language.

E. Experimental setup

The experimental set up is presented in Figure 1, with the following stages.

- This experiment only uses ± 20% of the N-BaIoT dataset, which is taken the average from all dataset files. Next, add a class label to each record file. Then the dataset files are combined into one file.
- Reduce data dimension with 3, 5, 10, and 15 components (features) using the fast ICA method and without reduce data. The used number of feature is chosen, randomly.
- Train the six classifiers of the IDS using Training dataset (70% of the total dataset).
- Classify the results of data reduction using the six classification methods, i.e.: AB, k-NN, RF, DT, GB, and NB.
- Assess the experimental results on the following parameters: execution time, TPR, accuracy, precision, sensitivity, specificity, FPR, and FNR.

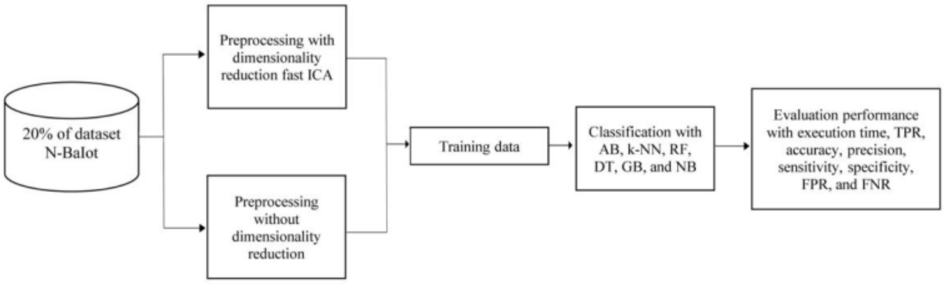
20% of dataset N-BaIot
Preprocessing with dimensionality reduction fast ICA
Preprocessing without dimensionality reduction
Training data
Classification with AB, k-NN, RF, DT, GB, and NB
Evaluation performance with execution time, TPR, accuracy, precision, sensitivity, specificity, FPR, and FNR

Figure 1. Experimental setup

IV. Experimental result, analysis, and comparison

A. Dataset preparasion

Experiment in this work only uses 20% of data records from each N-BaIoT dataset file. The N-BaIoT dataset consists of eighty-nine files which are then combined into one file using Python's *pd.concat* function. The class labeling on the N-BaIoT dataset lies in the file name according to the attack traffic of eleven labels. Then it is summarized into three labels as presented in Table 2. In the process of naming of dataset label, which is still located in the file name, for the purpose of training data, a new feature is added to each content of the dataset file. In this experiment, we add one feature with the name out, so that the total features are 116 features.

B. Dimensionality reduction using Fast ICA

Dimensionality reduction method decreases the scale of data from high to low scale through the transformation of the data into a matrix. The data is represented as a matrix X=AS. X is a matrix obtained from expansion of non-Gaussian (independent) linier combination of vector S that contains independent components are matrix A as linear mixing matrix (A. Hyvarinen and E. Oja, 2000). Fast ICA tries to 'un-mix' the data through determining an un-mixing matrix W, that satisfies $S = W K X$. Fast ICA has ability to estimate as many as numbers of attributes as features that makes it possible to reduce the data size by arranging the number of components < number of features, K is not rectangle matrix and A is estimated as pseudo-invers of WK.

In this work, the dimensionality reduction method is implemented in Python software using SKlearn decomposition fast ICA library. Thus, fast ICA transforms the N-BaIoT dataset that has 115 features into matrixes with 3,5,10 and 15 features.

C. Experimental result

Algorithm 1 presented the overall dimension reduction process. In the performance evaluation of reducing high-scale to low-scale dimensions of the fast ICA method with six classification algorithms, eight measurement metrics are considered, i.e.: accuracy, precision, sensitivity, specificity, true positive rate (TPR), FPR, false-negative rate (FNR), and execution time. Execution time is measured during the training time (time measured from the start to the end of the classification process). In the experiment, each dimensional reduction was classified by the AB, k-NN, RF, DT, GB, and NB classification algorithms (classifiers).

Algorithm 1 Fast ICA Dimensionality Reduction

```
Procedure process()
Input: N-BaIoT (20%)
Fast ICA : (n-component = (3, 5, 10, 15), algorithm = parallel, whiten = true, fun =
  logcosh, fun_args = none, max_iter = 200, tol = 0.1, w_init = none, random_state =
  0)
Apply Classifier
  C1 = AdaBoost; C2 = k-Nearest neighbor; C3 = Random Forest; C4 = Decision
  Tree; C5 = Gradient Boosting; C6 = Naïve Bayes
Calculate TPR, Accuracy, Precision, Sensitivity, specificity, FPR, and FNR
Compare Accuracy of C1, C2, C3, C4, C5, and C6
```

The performance of the dimension reduction classification using three features is presented in Table 3. k-NN has the highest accuracy of 99.95%. In addition, k-NN was also the best performance in detecting Bashlite, benign, and Mirai, with TPR reaches 0.9998, 0.9996, and 0.972. The worst performance was by NB in detecting benign, which only reaches 0.3033 of TPR. Significantly, DT and k-NN have the lowest FPR value of 0.0001 compared to other algorithms. To detect benign traffic, NB and AB have the lowest TPR values.

The performance of the dimension reduction classification using five features is presented in Table 4. RF has the highest accuracy of 99.99%. In addition, RF is also the highest in detecting Bashlites with TPR reaching 1. Likewise in detecting RF and DT mirrors it reaches a value of 1 of TPR. This is inversely proportional to the lowest NB in detecting benign, which only reaches 0.3815 of TPR. Significantly, DT and RF have the lowest FPR of 0 compared to others.

Table 3. Comparison performance metrices using three feature

Detection	k-NN	DT		NB	RF	AB	GB
Bashlite	0.9998	0.9963	0.9987	0.9949	0.9403	0.9045	0.9998
Benign	0.9972	0.9986	0.3022	0.9985	0.3273	0.9783	0.9972
Mirai	0.9996	0.9999	0.6518	0.9999	0.8299	0.9918	0.9996
Accuracy	99.95	99.84	76.43	99.78	83.5	95.56	99.95
Precision	0.9993	0.9987	0.8569	0.9981	0.8724	0.9683	0.9993
Sensitivity	0.9977	0.9999	0.3489	0.9999	0.3518	0.9829	0.9977
Specificity	0.9997	0.9998	0.9993	0.9997	0.9963	0.9984	0.9997
FPR	0.0001	0	0.0007	0.0003	0.0037	0.0015	0.0001
FNR	0.0023	0.0001	0.6511	0.0001	0.6481	0.0171	0.0023

Table 4. Comparison performance metrices using five feature

Detection	k-NN	DT	NB	RF	AB	GB
Bashlite	0.9998	0.9956	0.7297	1	0.8988	0.9793
Benign	0.9981	0.9998	0.3815	0.9998	0.9114	0.9903
Mirai	0.9981	1	0.8849	1	0.9899	0.9977
Accuracy	99.97	99.82	78.27	99.99	94.7	98.97
Precision	0.9995	0.9988	0.8193	0.9999	0.8951	0.9867
Sensitivity	0.9983	0.9998	0.4019	0.9998	0.9669	0.9912
Specificity	0.9999	1	0.997	1	0.9381	0.9956
FPR	0.0001	0	0.0029	0	0.0618	0.0043
FNR	0.0017	0.0001	0.5981	0.0001	0.033	0.0087

The performance of the dimension reduction classification using ten features is presented in Table 5. DT and RF have the highest accuracy of 99.99%. In addition, DT and RF are also the highest in detecting Bashlite, Mirai, and benign with TPR reaches 1, 1, and 0.9999. This result is consistent with their FPR values, which are also the lowest with values of 0. NB has the worst performance in detecting Bashlite, benign, and Mirai.

Table 5. Comparison performance metrices using ten feature

Detection	k-NN	DT	NB	RF	AB	GB
Bashlite	0.9998	0.9956	0.7297	1	0.8988	0.9793
Benign	0.9981	0.9998	0.3815	0.9998	0.9114	0.9903
Mirai	0.9981	1	0.8849	1	0.9899	0.9977
Accuracy	99.97	99.82	78.27	99.99	94.7	98.97
Precision	0.9995	0.9988	0.8193	0.9999	0.8951	0.9867
Sensitivity	0.9983	0.9998	0.4019	0.9998	0.9669	0.9912
Specificity	0.9999	1	0.997	1	0.9381	0.9956
FPR	0.0001	0	0.0029	0	0.0618	0.0043
FNR	0.0017	0.0001	0.5981	0.0001	0.033	0.0087

Dimensional reduction classification performance using fifteen features is presented in Table 6. DT and RF have the highest accuracy of 100%. In addition, DT and RF are also the highest in detecting Bashlite, Mirai and benign with TPR values reach 1. This result is also consistent with their FPR values, which are also the lowest with values of 0. NB is the worst performance in detecting Bashlite, Mirai and benign.

Table 6. Comparison performance metrices using fifteen feature

Detection	k-NN	DT	NB	RF	AB	GB
Bashlite	0.9999	1	0.9305	1	0.9478	0.9907
Benign	0.998	0.9999	0.5623	0.9999	0.9471	0.997
Mirai	0.9998	1	0.8416	1	0.9806	0.9992
Accuracy	99.97	99.99	85.55	99.99	96.47	99.56
Precision	0.9996	0.9999	0.8879	0.9999	0.9699	0.9963
Sensitivity	0.9983	0.9999	0.6225	0.9999	0.948	0.9972
Specificity	0.9999	1	0.9967	1	0.9974	0.9996
FPR	3.7512	0	0.0032	0	0.0026	0.0003
FNR	0.0007	0.0001	0.3777	0.0001	0.0519	0.0029

Classification performances without dimensional reduction are shown in Table 7. DT and RF have the highest accuracy, i.e.: 100%, followed by k-NN and GB with 99.9% and AB with 99.97% accuracy. The lowest accuracy was achieved by NB with 84.2% accuracy. For FPR level, DT and RF have the lowest, which is contrast to k-NN, AB and GB that have very high FPR level. Meanwhile, NB relatively has low level of FPR. Overall, DT and RF have the best values in term of TPR, precision, sensitivity, specificity, and FNR compared to other classifiers.

Table 7. Comparison performance metrices without dimensionality reduction

Detection	k-NN	DT	NB	RF	AB	GB
Detection	k-NN	DT	NB	RF	AB	GB
Bashlite	0,9999	1	0,6979	1	0,9998	0,9999
Benign	0,9990	1	0,5539	1	0,9998	0,9994
Mirai	0,9999	1	0,9991	1	0,9996	0,9999
Accuracy	99,99	100	84,27	100	99,97	99,99
Precision	0,9998	1	0,9022	1	0,9997	0,9999
Sensitivity	0,9993	1	0,6889	1	0,9998	0,9994
Specificity	0,9999	1	0,9975	1	0,9999	0,9999
FPR	1,2511	0	0,0025	0	8,2619	5,0043

D. Experimental result

Overall, the experimental results show that the proposed fast ica method performs well in reducing the number of features of n-baiot from 115 features to lower-scale dimensions; each consists of 3, 5, 10 and 15 features. These results show that fast ica dimensionality reduction provides different impacts on the accuracy performance. The implementation of dimensionality reduction accelerates the execution time of botnet detection. The use of lower scales datasets with 3, 5, 10 and 15 features decreases slightly the classification accuracy of k-nn,random forest, decision tree, gradient boosting, and naïve bayes. Nevertheless, still achieves more than 99%. The fpr value for k-nn decreases significantly with the use of 3, and 5 features. This situation differs from decision tree and random forest where the fpr values are not affected by the reduction in data dimensions.

On the other hand, the reduction in data dimension increases both, the precision value and the false-negative rate for naïve bayes, while for ab, dt, rf, and gb the precision value decreases significantly. Only k-nn has a relatively small decrease. Meanwhile, the false-negative rate on the dt is stable and the rf is relatively stable. Similar to naïve bayes, the false-negative rate also increases.

The sensitivity values of k-nn, dt, rf, and gb are relatively good, which is different from naïve bayes, where the sensitivity value drops dramatically. With regards to the specificity value, all classification methods achieve good values.

It is observed that the number of features significantly affects the detection performance. Table 8 shows a summary of the classification accuracy which is influenced by the number of features used by the proposed fast ica method compared to without features reduction. The proposed approach achieves a stable accuracy for k-nn with an accuracy of up to 99.95% when using low scale dimensions. On the other hand, the reduction of low scale dimensions decreases the accuracy compared to without features reduction. The decrease in accuracy is quite significant when using three features that occur in ab, gb, and nb. Nevertheless, nb achieves higher accuracy level when using 5 features compared to without feature reduction.

Table 8. Comparison level of accuracy without and with the proposed dimensionality reduction approach

Detection	With				Without
	3	5	10	15	
k-NN	**99.95**	99.97	k-NN	**99.95**	99.97
DT	99.84	99.82	DT	99.84	99.82
RF	99.78	**99.99**	RF	99.78	**99.99**
AB	83.5	94.7	AB	83.5	94.7
GB	95.56	98.97	GB	95.56	98.97
NB	76.63	78.82	NB	76.63	78.82

Furthermore, the reduction of dimensions also affects the precision value. Table 9 shows summary of the precision which is affected by the number of features used. The proposed approach achieves a stable precision for k-nn with an accuracy of 99.93% when using low scale dimensions. On the other hand, the reduction of low-scale dimensions decreases the precision. In ab, the decrease in precision is quite significant when using three features. This fact is inversely proportional to the nb. In general, comparing to classification without features reduction shows that the lesser the number of features, the lesser the precision.

Table 9. Comparison level of precision without and with the proposed dimensionality reduction approach

Detection	With				Without
	3	5	10	15	
k-NN	**0.9993**	0.9995	0.9996	0.9996	0.9998
DT	0.9987	0.9988	**0.9999**	**1**	**1**
RF	0.9981	**0.9999**	**0.9999**	**1**	**1**
AB	0.8724	0.8951	0.9699	0.9796	0.9997
GB	0.9683	0.9867	0.9963	0.997	0.9999
NB	0.8569	0.8193	0.8878	0.7929	0.9022

Next, we analyze the value of sensitivity. Table 10 shows the summary of the sensitivity affected by the number of features. The proposed approach achieves a relatively stable sensitivity for k-NN, DT, RF, and GB when using low scale dimensions. On the other hand, the reduction of the low-scale dimension decreases the sensitivity. The decrease in sensitivity is quite significant when using three features that occur in AB. This is inversely proportional to NB. Comparing to classification without features reduction, the lesser the features used, the lower the sensitivity. However, for k-NN, DT and RF, the sensitivity values are stable for the use of 15 features and above.

The results of the performance experiment on the specificity are presented in Table 11, which shows a summary of the specificity influenced by the number of features used. The proposed approached has stable specificity for k-NN, DT, RF, and GB when using low scale dimensions. On the other hand, for NB there was an increase when using a low scale dimensions from fifteen to three features with specificity reaches 0.9963 while AB reaches 0.9993. Comparing to classification without features reduction, with the use of small number of feature, the specificity values decrease, however, for k-NN, DT and RF the specificity values are stable.

Table 10. Comparison level of sensitivity without and with the proposed dimensionality reduction approach

Detection	With				Without
	3	5	10	15	
k-NN	0.9977	0.9983	0.9983	0.9986	0.9993
DT	**0.999**	**0.9998**	**0.9999**	**1**	**1**
RF	**0.999**	**0.9998**	**0.9999**	**1**	**1**
AB	0.3518	0.9669	0.948	0.9588	0.9998
GB	0.9829	0.9912	0.9972	0.9971	0.9994
NB	0.3489	0.4019	0.6225	0.4267	0.6889

Table 11. Comparison level of specivicity without and with the proposed dimensionality reduction approach

Detection	With				Without
	3	5	10	15	
k-NN	0.9997	0.9999	0.9999	0.9999	0.9999
DT	**0.9998**	**1**	**1**	**1**	**1**
RF	0.9997	**1**	**1**	**1**	**1**
AB	0.9963	0.9381	0.6674	0.9575	0.9999
GB	0.9984	0.9956	0.9996	0.9995	0.9999
NB	0.9993	0.997	0.9967	0.9688	0.9975

The reduction of dimensions affects the FPR, as shown in Table 12. As for k-NN, there was a very significant decrease when using three and five features provide fpr value of 0.0001. In contrast, when using 10 and 15 features, the FPR value increases very sharply to 3.7512 and 4.2532. In the case of NB, there was also a decrease with the FPR to 0.0007 when using 3 features. This fact is inversely proportional to RF. For RF there is a slight increase but it is still very low with the FPR values from 0 to 0.0003. As for DT, the value of 0 in each feature is the lowest. These results are different for AB and GB that have FPR values very high without performing features reduction.

The results of the FNR performance evaluation are shown in Table 13. For NB, there was an increase when the dimension was reduced, with the FNR value reaches 0.6511 using three features. FNR of k-NN also increases but not too significant with FNR value of 0.0023 using three features. AB experiences a significant increase when using three features with a value of 0.6481 but has low value of FNR, i.e.: 0.033 when using five features. Meanwhile, for DT and RF the FPR value is relatively stable with the value of

0.0001 if using 15 features and below, and value of 0 if using above 15 features. These results are different for k-NN, AB, GB, and NB that have lower FNR values for classifying without performing features reduction.

Table 12. Comparison level of FPR without and with the proposed dimensionality reduction approach

Detection	With				Without
	3	5	10	15	
k-NN	0.0001	0.0001	3.7512	4.2532	1.2511
DT	**0**	**0**	**0**	**0**	**0**
RF	0.0003	**0**	**0**	**0**	**0**
AB	0.0037	0.0618	0.0026	0.0025	8.2619
GB	0.0015	0.0043	0.0003	0.0005	5.0043
NB	0.0007	0.0029	0.0032	0.0312	0.0025

Table 13. Comparison level of FNR without and with the proposed dimensionality reduction approach

Detection	With				Without
	3	5	10	15	
k-NN	0.0023	0.0017	0.0017	0.0014	0.0007
DT	**0.0001**	**0.0001**	**0.0001**	**0**	**0**
RF	**0.0001**	**0.0001**	**0.0001**	**0**	**0**
AB	0.6481	0.033	0.0519	0.0412	0.0002
GB	0.0017	0.0087	0.0029	0.0029	0.0006
NB	0.6511	0.5981	0.3777	0.5732	0.311

The reduction in data dimensions also greatly affects the execution time, as shown in Table Table 14. NB classifier is the fastest with average execution time of 52.83 seconds, and DT classifier is the second best with the average execution time of 64.137 seconds. To see the effect of the implementation on the execution time, experiment without using the proposed dimensionality reduction method was conducted and the execution time for each classifier algorithm is recorded. The results are shown in Table 14. The longest execution time of the proposed approach implementation is considered. From the table, it can be observed obviously for k-NN, DT, RF, AB and GB classifiers, the proposed dimensionality reduction method speeds up the execution time. In contrast, it is different for NB classifier where without dimensionality reduction method provides faster execution time.

Table 14. Comparison level of execution time without and with the proposed dimensionality reduction approach

Detection	With				Without
	3	5	10	15	
k-NN	79.11	82.73	89.87	126.03	30908.87
DT	51.63	60.17	71.98	72.77	163.75
RF	171.35	201.78	306.68	333.96	675.74
AB	126.12	139.55	405.36	463.05	1143.27
GB	362.45	516.22	2135.85	2535.23	5404.97
NB	**49.18**	**48.05**	**53.71**	**62.62**	**26.78**

E. Comparison of Implementation on other dataset

Table 15 shows the comparison results between the implementation of the proposed methods on the two data sets. It is observed that the proposed dimensional reduction method also works well with MedBIoT dataset (Guerra-Manzanares *et al*., 2020). Experiment on other dataset, i.e.: MedBIoT is conducted. The data were captured from 83 IoT devices, consist of physical devices: 2 switches and 1 light bulb, and virtual devices: 20 locks, 20 fans, 20 switches, and 20 light bulbs. Besides, the dataset has total records of 17,845,567 with 4 types of traffic data, i.e.: Normal, Bashlite, Mirai, and Torii. For the experiment, only 2,728,266 records of data were being used (15% are taken from each device).

Table 15. Comparison of implementation result on other dataset

Dataset	Method	# of Feature			
		3	5	10	15
N-BaIoT	k-NN	**99.95**	99.97	99.97	99.97
	DT	99.84	99.82	**99.99**	**100**
	RF	99.78	**99.99**	**99.99**	**100**
	AB	83.50	94.70	96.47	97.97
	GB	95.56	98.97	99.56	99.74
	NB	76.43	78.82	85.55	82.01
MedBIoT	k-NN	**99.07**	**99.71**	99.84	99.85
	DT	76.01	93.00	62.26	**99.99**
	RF	92.47	93.44	**99.98**	**99.99**
	AB	73.05	55.01	57.25	57.83
	GB	90.56	75.44	86.16	61.37
	NB	68.23	77.32	83.64	81.16

F. Comparison with other works

Benchmarking is done to the similar approaches that use same dataset (Bahsi, Nomm and La Torre, 2018; Nomm and Bahsi, 2019) as well as using different dataset (Zhao, Li, Zia and Albert Y Zomaya, 2017). The accuracy of the proposed method is higher than other methods as shown in Table 16.

Table 16. Comparison of implementation result on other method

Author & Year	Method	# of Feature				
		2	3	5	10	15
(Bahsi, Nomm and La Torre, 2018)	Fisher score + DT	98.43	98.51			98.97
(Nomm and Bahsi, 2019)	Entropi + SVM		93.15	92.33		88.27
(Zhao, Li, Zia and Albert Y Zomaya, 2017)	PCA + Softmax		84.99		84.44	84.41
Susanto et al. 2021	Prop. Model		**99.95**	**99.97**		**99.97**

V. Discussion

In this work, we analyze classification accuracy, precision, specifity, sensitivity, FPR, and execution time of each algorithm when we use different low scales of the dataset. We observe that k-NN, DT, and RF have accuracy above 99%, compared to AB, GB and NB when we use low scales N-BaIoT dataset. Observation on precision value, DT provides the highest, while DT and RF achieve the highest on sensitivity as well as on Specifity, where DT outperforms k-NN and RF when using dataset with three features. Observation on FNR value, again DT and RF achieve the best for each feature scale. Meanwhile, DT and RF have the lowest values for FPR where DT has more stable values with value of zero (0) for each feature number.

Observation on execution time, NB has the best execution time, but with low accuracy. Among methods that have accuracy above 99%, DT has the best execution time, followed by k-NN and RF.

Experiment results on two datasets N-BaIoT and MedBIoT show that the use of 3 features has higher accuracy for both datasets. When using 5 features, RF has higher accuracy on N-BaIoT dataset, whereas the use of 10 features and above, k-NN and DT have high accuracy for both datasets. It is also observed that implementation on MedBIoT dataset; k-NN has higher accuracy, in general. Further investigation shows that DT and RF classifiers with 10 features and above provide highest accuracy on both dataset, however, only RF provides a consistent high accuracy on MedBIoT dataset.

Overall, implementation on N-BaIoT dataset accuracy averages for k-NN, DT and RF are above 99%, while on MedBIoT dataset, only k-NN classifier has accuracy average above 99%. k-NN classifier performs well on both datasets. Its average accuracy level is above 99% for all features used. K-NN has better accuracy when using low scale data dimensions, i.e.: 3 and 5 features, as shown from the consistent performance of the implementation on comparison dataset MedBIoT, which has almost double number of records compared to N-BaIoT dataset. On the other hand, DT and RF, which have high average accuracy, i.e.: above 99% on N-BaIoT dataset, significantly drop their accuracy levels when implemented on MedBIoT dataset.

Comparison with previous works show that the proposed model outperforms the proposed Bahsi, Nomm and La Torre (2018) and Zhao, Li, Zia and Albert Y Zomaya (2017) when using 3 and 10 features, as well as Nomm and Bahsi (2019) when using 3,4, and 10 features.

From the experimental results of the implementation of fast ICA dimensionality reduction method combined with k-NN, DT, RF, AB, GB, and NB classification algorithms, we conclude that combination of fast ICA method with k-NN algorithm provide the most effective detection. This conclusion is in line with similar works that using the same dataset as well as different datasets.

VI. Conclussion

This study has investigated the impact of the dimensional reduction on scalability issues on IoT botnet detection through experiments. Fast ICA was chosen based on the rationale of its ability to reduce dimensions through point iterations and fast convergence in the preprocessing stage.

Overall, k-NN, DT and RF have the most stable accuracy which is above 99%. We observed that combination of fast ICA method with k-NN classification algorithm outperforms other classification algorithm combinations during the experiments using low scale dimensions. This claim is supported by experimental results using different datasets as well as comparison with similar works that using same datasets. Overall, it has the most stable accuracy with accuracy level above 99% , and relatively fast execution time. The combination with k-NN achieves fastest execution time with average execution time of 94.43 seconds. Overall, evaluation results of accuracy, precision, sensitivity, specificity, FPR, FNR and execution time using 3 features are 99.95%, 0.9993, 0.9977, 0.9997, 0.0001, 0.0023, 79.11 seconds, respectively, and 99.97%, 0.9995, 0.9983, 0.9999, 0.0001, 0.0017, 82.73 seconds when using 5 features, respectively.

The use of fast ICA eliminates irrelevant and unnecessary features in the dataset, which in turn, impacts positively to the decrement of the detection processing time.

It was observed from the experimental results that the use of low scale dimensions decrease significantly the accuracy of several classification algorithms, thus, we plan to conduct researches on optimizing the classification accuracy while considering the reduction in data dimensions.

Reference

A. Hyvarinen and E. Oja (2000) 'Independent component analysis: algorithms and applications', *Neural networks*, 13(5), pp. 411–430.

Abbas, S. H. (2017) 'Ids Feature Reduction Using Two', *International Journal of Civil Engineering and Technology (IJCIET)*, 8(3), pp. 468–478.

Abdulhammed, R., Musafer, H., Alessa, A., Faezipour, M., and Abuzneid, A. (2019) 'Features dimensionality reduction approaches for machine learning based network intrusion detection', *Electronics (Switzerland)*, 8(3), pp. 1–27.

Acharya, D. P. and Panda, G. (2008) 'A review of independent component analysis techniques and their applications', *IETE Technical Review (Institution of Electronics and Telecommunication Engineers, India)*, 25(6), pp. 320–332.

Ahmad, Z., Khan A. S., Shiang, C. W., Abdullah, J., and Ahmad, F. (2020) 'Network intrusion detection system: A systematic study of machine learning and deep learning approaches', *Transactions on Emerging Telecommunications Technologies*, 32(1), pp. 1–29.

Aljanabi, M., Ismail, M. A. and Ali, A. H. (2021) 'Intrusion Detection Systems, Issues, Challenges, and Needs', *International Journal of Computational Intelligence Systems*, 14(1), pp. 560–571.

Alsaadi, H. I., Almuttairi, R. M., Bayat, O., and Ucani, O. N. (2020) 'Computational intelligence algorithms to handle dimensionality reduction for enhancing intrusion detection system', *Journal of Information Science and Engineering*, 36(2), pp. 293–308.

Alshamkhany, M., Alshamkhany, W., Mansour, M., Khan, M., Dhou, S., Aloul, F. (2020) 'Botnet Attack Detection using Machine Learning', in *Proc. 14th International Conference on Innovations in Information Technology, IIT*. Al ain, United Arab Emirates, pp. 203–208.

Andalib, A. and Vakili, V. T. (2020) 'A Novel Dimension Reduction Scheme for Intrusion Detection Systems in IoT Environments', *arXiv*.

Ayesha, S., Hanif, M. K. and Talib, R. (2020) 'Overview and comparative study of dimensionality reduction techniques for high dimensional data', *Information Fusion*. Elsevier B.V., 59, pp. 44–58.

Bahsi, H., Nomm, S. and La Torre, F. B. (2018) 'Dimensionality Reduction for Machine Learning Based IoT Botnet Detection', in *Proc. 2018 15th International Conference on Control, Automation, Robotics and Vision, ICARCV*. Singapore, Singapore: IEEE, pp. 1857–1862.

Bhatia, N. and Vandana (2010) 'Survey of Nearest Neighbor Techniques', *International Journal of Computer Science and Information Security*, 8(2), pp. 302–305.

Chen, A. and Bickel, P. J. (2006) 'Efficient independent component analysis', *Annals of Statistics*, 34(6), pp. 2825–2855.

Perez, P., Alonso, S., Moran, A., Prada, M. A., Fuertes, J. J., and Dominguez, M. (2019) 'Comparison of network intrusion detection performance using feature representation', in *International Conference on Engineering Applications of Neural Networks*. Xersonisos, Greece, pp. 463–475.

Fang, M., Yu, Y., Zhang, W., Wu, H., Deng, M., and Fang, J. (2015) ‘High Performance Computing of Fast Independent Component Analysis for Hyperspectral Image Dimensionality Reduction on MIC-Based Clusters’, in *Proceedings of the International Conference on Parallel Processing Workshops*. Massachusetts, Amerika Serikat, pp. 138–145.

Farnaaz, N. and Jabbar, M. A. (2016) ‘Random Forest Modeling for Network Intrusion Detection System’, in *Procedia Computer Science*. Bangalore, India: The Author(s), pp. 213–217.

Guerra-Manzanares, A., Medina-Galindo, J., Bahsi, H., and Nomm, S. (2020) ‘MedBIoT: Generation of an IoT botnet dataset in a medium-sized IoT network’, *ICISSP 2020 - Proceedings of the 6th International Conference on Information Systems Security and Privacy*, pp. 207–218.

Gujar, S. S. and Patil, B. M. (2014) ‘Intrusion Detection Using Naïve Bayes for Real Time Data’, *International Journal of Advances in Engineering & Technology*, 7(2), pp. 568–574.

Hamid, Y., Journaux, L., Lee, J. A., Sautot, L., Nabi, B., Sugumara, M. (2017) ‘Large-scale nonlinear dimensionality reduction for network intrusion detection’, in *ESANN 2017 - Proceedings, 25th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*. Bruges, Belgium, pp. 153–158.

Hu, W. and Hu, W. (2005) ‘Network-based intrusion detection using adaboost algorithm’, in *Proceedings - 2005 IEEE/WIC/ACM InternationalConference on Web Intelligence, WI 2005*. Compiegne, France, pp. 712–717.

Huang, X. and Zeng, H. (2012) *Efficient variant of FastICA algorithm for speech features extraction*, *Lecture Notes in Electrical Engineering*. Springer.

Jabbar, A. F. and Mohammed, I. J. (2020) ‘Development of an Optimized Botnet Detection Framework based on Filters of Features and Machine Learning Classifiers using CICIDS2017 Dataset’, in *IOP Conference Series: Materials Science and Engineering*, pp. 1–14.

Kasturiwale, H. and Mizwan, Z. (2014) ‘Comparison and Performance Analysis of various ICA Algorithms for ECG signals’, *International Journal of Engineering Research & Technology*, 3(4), pp. 674–677.

Kaviani, P. and Dhotre, S. (2017) ‘Short Survey on Naive Bayes Algorithm’, *International Journal of Advance Engineering and Research Development*, 4(11), pp. 607–611.

Khraisat, A., Gondal, I., Vamplew, P., and Kamruzzaman, J. (2019) ‘Survey of intrusion detection systems: techniques, datasets and challenges’, *Cybersecurity*. Cybersecurity, 2(1), pp. 1–22.

Kotsiantis, S. B. (2013) ‘Decision trees: A recent overview’, *Artificial Intelligence Review*, 39(4), pp. 261–283.

Kumar, M. and Hanumanthappa, M. (2012) ‘Intrusion detection system using decision tree algorithm’, in *IEEE 14th International Conference on Communication Technology*. Chengdu, Cina, pp. 730–734.

Li, W., Yi, P., Wu, Y., Pan, L., and Li, J. (2014) ‘A new intrusion detection system based on KNN classification algorithm in wireless sensor network’, *Journal of Electrical and Computer Engineering*, 2014, pp. 1–8.

Liao, Y. and Vemuri, V. R. (2002) ‘Use of k-nearest neighbor classifier for intrusion detection’, *Computers and Security*, 21(5), pp. 439–448.

Mazini, M., Shirazi, B. and Mahdavi, I. (2019) ‘Anomaly network-based intrusion detection system using a reliable hybrid artificial bee colony and AdaBoost algorithms’, *Journal of King Saud University - Computer and Information Sciences*. King Saud University, 31(4), pp. 541–553.

Meidan, Y., Bohadana, M., Mathov, Y., Mirsky, Y., Breitenbacher, D., Shabtai, A., and Elovici, Y. (2018) ‘N-BaIoT-Network-based detection of IoT botnet attacks using deep autoencoders’, *IEEE Pervasive Computing*, 17(3), pp. 12–22.

Minguan, F., Haifang, Z., Weiming, Z., and Shen, X. L. (2015) ‘A parallel algorithm of FastICA dimensionality reduction for hyperspectral image on GPU’, *Journal of National University of Defence Technology*, 37(4), pp. 65–70.

Mirsky, Y., Doitshman, T., Elovici, Y., Shabtai, A. (2018) ‘Kitsune: An ensemble of autoencoders for online network intrusion detection’, *arXiv*, (February), pp. 18–21.

Moustakidis, S. and Karlsson, P. (2020) ‘A novel feature extraction methodology using Siamese convolutional neural networks for intrusion detection’, *Cybersecurity*. Cybersecurity, 3(16), pp. 1–13.

Mukherjee, S. and Sharma, N. (2012) ‘Intrusion Detection using Naive Bayes Classifier with Feature Reduction’, in *Procedia Technology*. Hooghly, India: Elsevier B.V., pp. 119–128. Available at:

Mutlaq, K. A. A., Madhi, H. H. and Kareem, H. R. (2020) ‘Addressing big data analytics for classification intrusion detection system’, *Periodicals of Engineering and Natural Sciences*, 8(2), pp. 693–702.

Natekin, A. and Knoll, A. (2013) ‘Gradient boosting machines, a tutorial’, *Frontiers in Neurorobotics*, 7(DEC), pp. 1–21.

Negandhi, P., Trivedi, Y. and Mangrulkar, R. (2019) ‘Intrusion Detection System Using Random Forest on the NSL-KDD Dataset’, in *Advances in Intelligent Systems and Computing*. Springer.

Nomm, S. and Bahsi, H. (2019) ‘Unsupervised Anomaly Based Botnet Detection in IoT Networks’, in *Proc.- 17th IEEE International Conference on Machine Learning and Applications, ICMLA*. Orlando, Amerika Serikat: IEEE, pp. 1048–1053.

Nziga, J. P. (2011) ‘Minimal dataset for network intrusion detection systems via dimensionality reduction’, in *2011 6th International Conference on Digital Information Management, ICDIM 2011*. Melbourne, Australia, pp. 168–173.

Pajouh, H. H., Javidan, R., Dehghantanha, A., Choo, K. R. (2019) ‘A Two-Layer Dimension Reduction and Two-Tier Classification Model for Anomaly-Based Intrusion Detection in IoT Backbone Networks’, *IEEE Transactions on Emerging Topics in Computing*, 7(2), pp. 314–323.

Platonov, V. V. and Semenov, P. O. (2014) ‘Dimension reduction in network attacks detection systems’, *Nonlinear Phenomena in Complex Systems*, 17(3), pp. 284–289.

Popoola, S. I., Adebisi, B., Hammoudeh, M., Gui, G., and Gacanin, H. (2021) ‘Hybrid Deep Learning for Botnet Attack Detection in the Internet-of-Things Networks’, *IEEE Internet of Things Journal*, 8(6), pp. 4944–4956.

Salo, F., Nassif, A. B. and Essex, A. (2019) ‘Dimensionality reduction with IG-PCA and ensemble classifier for network intrusion detection’, *Computer Networks*. Elsevier B.V., 148, pp. 164–175.

Sarveniazi, A. (2014) ‘An Actual Survey of Dimensionality Reduction’, *American Journal of Computational Mathematics*, 04(02), pp. 55–72.

Sen, P. C., Hajra, M. and Ghosh, M. (2020) ‘Supervised Classification Algorithms in Machine Learning: A Survey and Review’, *Advances in Intelligent Systems and Computing*. Springer Singapore, 937, pp. 99–111.

Al Shorman, A., Faris, H. and Aljarah, I. (2019) ‘Unsupervised intelligent system based on one class support vector machine and Grey Wolf optimization for IoT botnet detection’, *Journal of Ambient Intelligence and Humanized Computing*. Springer Berlin Heidelberg, 11, pp. 2809–2825.

Sipola, T., Juvonen, A. and Lehtonen, J. (2012) ‘Dimensionality reduction framework for detecting anomalies from network logs’, *Engineering Intelligent Systems*, 20(1–2), pp. 95–105.

Sivatha Sindhu, S. S., Geetha, S. and Kannan, A. (2012) ‘Decision tree based light weight intrusion detection using a wrapper approach’, *Expert Systems with Applications*. Elsevier Ltd, 39(1), pp. 129–141.

Sreenivasa Chakravarthi, S. and Jagadeesh Kannan, R. (2019) ‘Non-linear dimensionality reduction-based intrusion detection using deep autoencoder’, *International Journal of Advanced Computer Science and Applications*, 10(8), pp. 168–174.

Syarif, I., Prugel-Bennett, A. and Wills, G. (2012) ‘Data mining approaches for network intrusion detection: from dimensionality reduction to misuse and anomaly detection’, *Journal of Information Technology Review*, 3(2), pp. 70–83.

Thaseen, S. and Kumar, C. A. (2016) ‘Intrusion Detection Model using PCA and Ensemble of Classifiers’, *Advances in Systems Science and Application*, 16(2), pp. 15–38.

Velliangiri, S., Alagumuthukrishnan, S. and Thankumar Joseph, S. I. (2019) ‘A Review of Dimensionality Reduction Techniques for Efficient Computation’, in *Procedia Computer*

Science. Chennai, India: Elsevier B.V., pp. 104–111.

Vlachos, M. (2011) ‘Dimensionality Reduction’, in *Encyclopedia of Machine Learning*. Springer US.

Wang, W., Du, X., Shan, D., Qin, R., and Wang, N. (2020) ‘An Intrusion Detection Method based on Stacked Autoencoder and Support Vector Machine’, *IEEE Transactions on Cloud Computing*, 1453(1), pp. 1–14.

Wu, S. and Nagahashi, H. (2015) ‘Analysis of generalization ability for different AdaBoost variants based on classification and regression trees’, *Journal of Electrical and Computer Engineering*, 2015, pp. 1–17.

Xin, Q., Yongjian, N., Jianwei, W., and Linghua, S. (2009) ‘Dimensionality reduction for hyperspectral imagery based on fastica’, *Journal of Electronics*, 26(6), pp. 831–835.

Zhang, B., Liu, Z., Jia, Y., Ren, J., and Zhao, X. (2018) ‘Network Intrusion Detection Method Based on PCA and Bayes Algorithm’, *Security and Communication Networks*, 2018, pp. 1–11.

Zhao, S., Li, W., Zia, T. and Zomaya, A. Y. (2017) ‘A dimension reduction model and classifier for anomaly-based intrusion detection in internet of things’, in *Proceedings - 2017 IEEE 15th International Conference on Dependable, Autonomic and Secure Computing,* Orlando,
5 Amerika Serikat, pp. 836–843.

Zhao, S., Li, W., Zia, T. and Zomaya, A. Y. (2017) ‘A Dimension Reduction Model and Classifier for Anomaly-Based Intrusion Detection in Internet of Things A Dimension Reduction Model and Classifier for Anomaly-Based Intrusion Detection in Internet of Things’, in *2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing*. Orlando, Amerika Serikat, pp. 836–843.

Zheng, J. and Zhou, Y. (2017) ‘Intrusion detection algorithm based on improved principal component analysis’, *Journal of Discrete Mathematical Sciences & Cryptography*, 20(5), pp. 1007–1016.

Zong, W., Chow, Y. W. and Susilo, W. (2019) ‘Dimensionality Reduction and Visualization of Network Intrusion Detection Data’, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11547 LNCS, pp. 441–455.

Dimensionality reduction with Fast ICA for IoT Botnet Detection

ORIGINALITY REPORT

5%
SIMILARITY INDEX

PRIMARY SOURCES

1	archive.org Internet	116 words —	1%
2	www.coursehero.com Internet	62 words —	1%
3	umexpert.um.edu.my Internet	61 words —	1%
4	www.mdpi.com Internet	58 words —	1%
5	researchoutput.csu.edu.au Internet	51 words —	1%
6	acikerisim.uludag.edu.tr Internet	50 words —	1%
7	www.scribd.com Internet	49 words —	1%

EXCLUDE QUOTES	OFF	EXCLUDE SOURCES	< 1%
EXCLUDE BIBLIOGRAPHY	OFF	EXCLUDE MATCHES	OFF