SINERGI Vol. 28, No. 1, February 2024: 31-42
http://publikasi.mercubuana.ac.id/index.php/sinergi
http://doi.org/10.22441/sinergi.2024.1.004

Effective and efficient approach in IoT Botnet detection

Susanto[1], Deris Stiawan[2*], M. Agus Syamsul Arifin[1], Mohd Yazid Idris[3], Rahmat Budiarto[4]
[1]Faculty of Engineering Science, Universitas Bina Insan, Indonesia
[2]Faculty of Computer Science, Universitas Sriwijaya, Indonesia
[3]Faculty of Computing, Universiti Teknologi Malaysia, Malaysia
[4]College of Computer Science, Albaha University, Saudi Arabia

Abstract
The Internet of Things (IoT) enables the interaction of physical systems connected to the internet network, resulting in the generation of extensive data traffic with high dimensions. While IoT applications offer benefits and convenience to users, network security remains uncertain. One example is vulnerability to cyber-attacks, such as botnets targeting consumers' IoT devices. In the realm of network security analysis, dealing with high-dimensional data poses distinct challenges for researchers. These challenges include the curse of dimensionality, which can complicate feature definitions; predominantly unordered datasets; combinations of clusters; and exponential data growth. In this study, we applied feature reduction using the Linear Discriminant Analysis (LDA) method to minimize features on the IoT network to detect botnet. The reduction process is carried out on the N-BaIoT dataset which has 115 features reduced to 2 features. Performing feature reduction with detection systems has become more effective and efficient. Experimental result showed that the application of LDA combined with machine learning on the classification Decision Tree method was able to detect with accuracy that reached 100% in 98.58s with only two features.

Keywords:
Dimensionality reduction;
IoT;
LDA;

Article History:
Received: May 16, 2023
Revised: July 23, 2023
Accepted: July 28, 2023
Published: February 2, 2024

Corresponding Author:
Deris Stiawan
Faculty of Computer Science,
Universitas Sriwijaya, Indonesia
Email: deris@unsri.ac.id

INTRODUCTION

The Internet of Things (IoT) enables the interaction of physical systems connected to the internet network, resulting in the generation of extensive data traffic with high dimensions [1]. While IoT applications offer benefits and convenience to users, network security remains uncertain. One example is vulnerability to cyber-attacks, such as botnets targeting consumers' IoT devices [2]. In the realm of network security analysis, dealing with high-dimensional data poses distinct challenges for researchers. These challenges include the curse of dimensionality, which can complicate feature definitions; predominantly unordered datasets; combinations of clusters; and exponential data growth [3]. Additionally, the substantial dimensions of data traffic can impact the performance of machine learning during the data analysis process [4].

Set of data with high scalability may have useless and unrelated features that tend to disguise the main features, which in turn decrease the data analysis performance such as classification accuracy [5]. On the other hand, further ineffective dimension reduction will jeopardize efficiency of machine learning for pattern recognition and increase the workload of data analysis process [6]. One of the methods that commonly used, effective and efficient to reduce the number of data attributes is the dimension reduction method [7]. The dimension reduction method is able to project data that have high dimensions into low-dimension data while preserving the original information [8]. In addition,

it is very efficient in memory space required for data storage [9].

Many studies have utilized dimensionality reduction method for data analysis on attack detection [10]–[12], however, most of the methods are unsuccessful in utilizing lower dimension scale, because data dimension reduction does not necessarily increase the classification [13]. This research work contributes towards an analysis of IoT botnet dataset scalability reduction. The best data dimensional obtained from the experiment are used as selected features for the detection system. The level of effectiveness and efficiency is measured through detection accuracy level as well as detection time. The measurement results are compared to a detection system without the selected features. Moreover, the experiment results are also compared to existing detection systems. Metrics used for comparison include: accuracy, execution time, detection rate, false positive rate (FPR), false negative rate (FNR), sensitivity, specificity, and precision.

The rest of this paper is divided into five sections as follows. In Section 2, we present related works on dimensionality reduction method on IoT botnet detection. Section 3 describes the dataset, LDA, classification algorithm, evaluation performance, and analysis tools. Section 4 presents the results of the experimental analysis and a comparison with other works. Section 5 presents the Discussion. Section 6 presents the conclusions and further work on dimensionality reduction method in the IoT botnet.

RELATED WORK

The development of IoT technology has increased the research need to develop effective security protection from an attack. The attack may caused by network traffic of heterogeneous IoT devices, which generates high-scale data, thus retaining the chance of attack [14]. Few researchers have used the dimensionality reduction method to detect IoT botnet. The first step to creating botnet attack protection toward an IoT network that has high-scale data traffic is to use dimensional reduction. Bahsi et al. [15] performed feature reduction on an N-BaIoT dataset [16] from 115 features to 2, 3, and 10 features using the Fisher Score method. The proposed method uses fewer features to reach a high accuracy level. A slight decrease in accuracy was observed when using the classification Decision Tree method while applying fewer features, although still above 98% for each feature. With k-NN, there is an increase in accuracy for each lower number of features, reaching up to 98.05% accuracy.

With the same N-BaIoT dataset, Nomm and Bahsi [17] performed feature reduction of 3, 5, and 10 features by using comparative unsupervised method, i.e.: entropy, Hopkins statistics and variance, then continue with the SVM classification method and isolation forest. The authors reported that the entropy-SVM method increased the accuracy up to 93.15% when fewer features were used. The result is opposite to variance-SVM and Hopkins-SVM method, which significantly lowered the accuracy, whereas with the Hopkins-isolation forest, the decrement was not significant but had low accuracy. Different results were obtained for the variance-isolation forest and entropy-isolation forest methods, which have variant accuracy when using low features.

Using the same N-BaIoT dataset, Liu et al. [18] used a triangle area map-based multivariate correlation analysis algorithm (TAM-based MCA) method to reduce features into 23 dimensions. Using the convolutional neural network (CNN) method, their approach offered very high accuracy, up to 99.57%. Alqahtani et al. [19] optimized performance classification GXGBoost by reducing data into three features using Fisher Score. In the experiment, IoT Botnet detection using the N-BaIoT dataset was effective and efficient with an accuracy of 99.96%.

In another study with the IoT network intrusion detection dataset, Desai et al. [20] used dimensionality reduction method and optimized the classification function on IoT botnet detection. Principal component analysis (PCA) method was chosen by the authors to reduce the data dimension. IoT network intrusion detection dataset, which has 115 features, was reduced to 10, 15, and 20 features, and then are classified using multi-classification. The results from the experiment showed an accuracy level reaching 99.97% using Random Forest classifier. This result is superior to the Decision Tree and SVM classification methods.

Besides using the N-BaIoT dataset, there are also studies that have used the dimensionality reduction method on IoT botnet with another dataset. Alshamkhany et al. [21] reduced data dimension using the PCA method and machine learning. Their experiments with the UNSW-NB15 dataset [22] and Bot-IoT [23], which were classified using the machine learning SVM-RBF method, achieved a very high accuracy result of 99.9%. The PCA method was used to reduce number of features from 43 into 20 features. Popoola et al. [24] reduced features in the Bot-IoT dataset to six features using the long short-term memory autoencoder (LAE) method. The method showed that classification of deep

bidirectional long short-term memory performed well.

METHODS

Dataset

Experiments in this work uses the N-BaIoT dataset [16], which was extracted into CSV format by using a statistics method [25]. The N-BaIoT dataset was selected due to its high data dimension that demands more computational power; it encompasses a large number of features, necessitating the removal of unnecessary ones; and finally, reducing the data dimensions is crucial to achieving better performance [17]. In this work, only 20% of the data was used by randomly selecting from each dataset file. Data distribution of sets that were completely used is shown in Table 1. The dataset is created from a network representing an IoT system in real world. The IoT system consists of nine IoT devices, i.e.: four security cameras, one baby monitor camera, two doorbells, one thermostat, and one webcam). Botnet attacks were injected into the network. The dataset has a total of 7062606 records with 115 features. Moreover, to facilitate result comparison, the medBIoT dataset was also employed [26]. The medBioT dataset surpasses the N-BaIoT dataset in terms of traffic volume. It comprises data gathered from both physical and virtual IoT devices, totaling 83 devices. Within this dataset, there is one category of benign traffic and three types of attack traffic (bashlite, mirai, and torii), amounting to a total of 17,845,567 data records. In contrast, this research utilizes approximately 15% of the medBioT dataset, resulting in a total of 2,728,266 data records, which is nearly twice the number of N-BaIoT datasets used.

A detailed breakdown of the medBIoT dataset's distribution can be found in Table 2. Data variables (attributes) with different scales were standardized using *StandardScaler* to reduce dimensions, which was continued by training and then testing the classification model.

Table 1. Distribution of N-BaIoT dataset

Device	File	Label	Total Data
Four security cameras, baby monitors, two doorbells, thermostats, and webcams	Benign	Benign	111179
	Combo		103030
	Junk		52158
	Scan	Bashlite	51022
	Tcp		171969
	Udp		192873
	Ack		128764
	Scan		107596
	Syn	Mirai	146660
	Udp		246001
	Udpplain		104660
	Total		1415912

Table 2. Distribution of medBIoT dataset

Device	New Label Feature	File	Total Data
Twenty Lock, twenty two switch, twenty fan, twenty one light	Benign	bashlite_leg_fan	209715
		bashlite_leg_light	195363
		bashlite_leg_lock	209715
		bashlite_leg_switch	209715
		mirai_leg_fan	58620
		mirai_leg_light	52769
		mirai_leg_lock	52836
		mirai_leg_switch	43911
		torii_leg_fan	15389
		torii_leg_light	4376
		torii_leg_lock	2181
		torii_leg_raspberry1	25910
		torii_leg_raspberry2	943
		torii_leg_switch	6380
	Bashlite	bashlite_mal_CC_fan	37770
		bashlite_mal_CC_light	33457
		bashlite_mal_CC_lock	31428
		bashlite_mal_CC_switch	22845
		bashlite_mal_spread_fan	206221
		bashlite_mal_spread_light	180027
		bashlite_mal_spread_lock	152360
		bashlite_mal_spread_switch	177722
	Mirai	mirai_mal_CC_fan	593329
		mirai_mal_CC_light	18636
		mirai_mal_CC_lock	11357
		mirai_mal_CC_switch	21205
		mirai_mal_spread_fan	23966
		mirai_mal_spread_light	14248
		mirai_mal_spread_lock	18590
		mirai_mal_spread_switch	32527
	Torii	torii_mal_fan	27496
		torii_mal_light	70
		torii_mal_lock	33
		torii_mal_raspberry1	9245
		torii_mal_ raspberry2	528
		torii_mal_switch	27383
Total			2728266

Random data separation was performed to split the dataset into a training set, having 70% of the data, and a testing set, having 30% of the data.

LDA

The LDA technique projects the original data matrix to a lower dimension space. To reach this goal, three steps must be performed. The first step is to calculate the distance between different classes, which is called variants between classes and matrix between classes. The second step is calculating the distance between mean and sample from each class, which is called variant within class or matrix within-class. In the third step, a lower-dimension

room is created to maximize the variant between classes and minimize the variant within classes [27]. LDA is performed to obtain appropriate training data by giving new space because its reduction technique is created by maximizing the distance of class [28].

LDA reaches transformation linear optimal *W*, which reduces the distance within classes and extends simultaneously the distance between classes. Criteria *J* (XW), which is being maximized, as presented in (1).

$$J(\mathrm{XW}) = -LDA(XW) = -\frac{W^T S_B W}{W^T S_w W} \quad (1)$$

where SB is the between-class matrix and SW is the within-class matrix and determined by (2) and (3).

$$S_B = \sum_C (\mu_C - \bar{x})(\mu_C - \bar{x})^T \quad (2)$$

$$S_W = \sum_C \sum_{\chi_i \epsilon C} (\chi_i - \mu_C)(\chi_i - \mu_C)^T \quad (3)$$

LDA reduction process on the N-BaIoT dataset:

1. Determine the number of classes in the dataset. The N-BaIoT dataset has 3 classes
2. Determine the maximum amount of data reduction that can be used in the LDA method according to the statement of Tsymbal et al. [29], for LDA-based dimensional reduction, the maximum feature reduction is the total number of classes –1. Thus, since the N-BaIoT dataset has three classes, the maximum feature reduction is (3–1=2) features.
3. The LDA reduction process on the N-BaIoT dataset which has a total of 115 features/columns is transformed into a 2-column matrix.

The advantage of LDA compared to other techniques in dimension reduction is in the process of reducing dimensions while maintaining the global structure of the data so that distances in low-dimensional structures are found to be significant [29]. In addition to the level of accuracy resulting from LDA data reduction, the test accuracy is better than PCA [30].

Classification Algorithm

Machine learning applications have great potential for botnet classification, which is mainly not in the IoT environment [31]. Hossain et al. [32] stated that classification using machine learning is effective in botnet detection. In this study, we used a classification Decision Tree (DT) algorithm, and then compared it with several other classification algorithms, such as AdaBoost (AB), K-nearest neighbor (k-NN), Random Forest (RF), and Gradient Boosting (GB).

Decision Tree

Decision Tree is a predictive model that maps observations on data [33]. Tariq and Baig [34] used the Decision Tree classification method in detecting botnet. The results in better level detection compared to other approaches with the same heterogeneity level in testing, with accuracy reaches 94.8%. Mata et al. [35] applied feature selection using a Decision Tree in botnet detection, yielding very efficient results with an average time detection of 0.78 microseconds.

K-nearest neighbor

K-NN classifier is used to classified unlabelled observation group into class from the similarity of label. Observation characteristic which grouped are used to train and testing. By using parameter k in determining how many selected neighbour. Correct selection k will impacted significant toward performance diagnostic KNN algorithm. K reduce impact of variants that caused by random false, but this will risk to ignore the small pattern but important. Key to select correct k value is to reach balanced between overfitting and underfitting [36][37]. In detecting HTTP botnet Dollah et al [38] using k-NN classification algorithm. Proposed method able to classified HTTP Botnet in network traffic with average accuracy 92,93%.

Random Forest

Random forest consists of Decision Tree. Random operation is introduced in the creation process, including selecting sample subsets and feature subsets to guarantee the independence of each Decision Tree, increase classification accuracy, and obtains enhanced generalization ability. Random operation in random forests significantly improves classification performance. Given that the process of each Decision Tree is very fast, parallelization in creating a random forest can be made, which improves classification speed [39][40]. Hoang and Nguyen [41] detected botnets using a machine learning technique, i.e.: random forest method for effective botnet detection with accuracy reached 90%. Moubayed et al. [42] optimized the random forest method in botnet detection using the hyperparameter method, resulting in a genetic algorithm that had good framework effectiveness in detecting botnet attacks from bad hosts.

Adaboost

AdaBoost algorithm is a learning algorithm ensemble that consists of sub-classifiers used to overcome the weakness of bad classification from each sub-classifier [43]. Javed et al. [44] used Adaboost to detect botnet attacks on

network traffic using Adaboost to achieve a true positive rate that reached 99.7%. This methodology was efficient in detecting botnet.

Gradient Boosting

Gradient boosting is an algorithm like boosting, which is used for regression. Boosting algorithm combines weak learning, that is, learning that is slightly better than random, with performing reduplication from learning [45]. Ongun et al. [46] used machine learning to detect botnets from network traffic by using a gradient boosting classification method to reach better detection accuracy, even in imbalance data scenarios.

Evaluation Performance

The performance of botnet detection was evaluated using a confusion matrix table, as shown in Table 3 [47]. The confusion matrix for botnet detection consists of the following.

1. TP (true positive) is the correct number of actual data that are predicted to be normal.
2. FP (false positive) is number actual data normal which predict as botnet attack.
3. FN (false negative) is the number of actual data attacks that false predict as normal.
4. TN (true negative) is number actual data normal which false predict as botnet attack.

Table 3. Confusion Matrix

		Predicted class	
		Positive (P)	**Negative (N)**
Actual class	True (T)	TP	FP
	False (F)	FN	TN

The definitions in the confusion matrix, which was mostly used in calculating the classification matrix, are as follows.

1. Accuracy is the ratio between the total data classified correctly and the total sample.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (4)$$

2. Precision is the ratio between negative sample classified correctly with total sample negative prediction.

$$Precision = \frac{TP}{TP+FP} \quad (5)$$

3. Sensitivity is the ratio between a positive sample classified correctly and a total sample.

$$Sensitivity = \frac{TP}{TP+FN} \quad (6)$$

4. Specificity is the ratio between negative samples classified correctly with total sample.

$$Specificity = \frac{TN}{TN+FP} \quad (7)$$

5. False positive rate is ratio between negative samples which false classified with total sample.

$$False\ possitive\ rate = \frac{FP}{TN+FP} \quad (8)$$

6. False negative rate is ratio between positive sample which false classified with total sample

$$False\ negative\ rate = \frac{FN}{FN+TP} \quad (9)$$

Experimental Setup

This research compares the system detection of IoT botnets on machine learning with and without the LDA dimensionality reduction method. In this study, we use the N-BaIoT dataset CSV version [16]. Figure 1 shows the proposed framework.

Figure 1 shows all performance evaluations, namely accuracy, precision, sensitivity, specificity, FPR, FNR, and execution time in this experiment which was carried out resulting from the training data set.

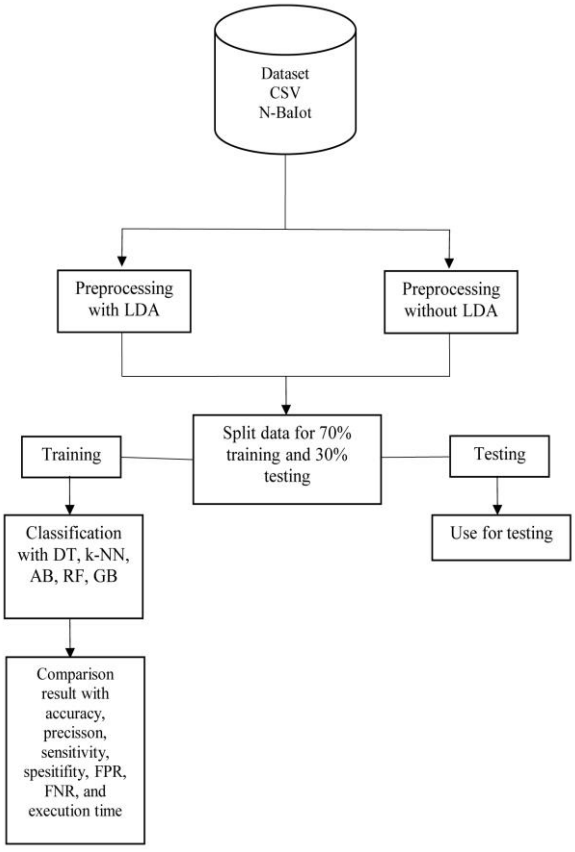

Figure 1. Proposed research framework

Algorithm 1 : Reduction data LDA
Input : Dataset N-BaIoT. // Matrix A
Output : Accuracy, precision, sensitivity, specificity, FPR, FNR, and execution time
Stage 1 :
1. Construct the matrices
2. S_B as in (2). // reduce between-class scatter
3. S_w as in (3). // reduce within-class scatter
4. Compute the T eigenvectors, with the corresponding eigenvalues sorted in nondecreasing order
5. J ⟵ XW, where W = [W_1, ….. , W_T]. // optimal transformation
6. A^L ⟵ J^T A. // reduce represention

Stage 2 :
1. Split data training 70% and data testing 30%
2. Classification data training with DT, AB, GB, K-NN, and RF

Figure 2. Dimensionality reduction LDA

The data reduction process with LDA is described in pseudocode as presented in Figure 2.

Hardware and Software Setup

In this experiment, we perform a simulation using a computer with a specification Intel core i7 processor 9th gen, 16 GB DDR4 RAM, 512 GB SSD, and NVIDIA GTX1660 Ti GPU. The operating system was Windows 10, and Python 3.7.4 was used for the analysis.

RESULTS AND DISCUSSION

Experiment Result

We evaluate the performance of dimension reduction by comparing the results with and without the LDA method using five classification algorithms. The performance was measured using eight metrics, i.e.: accuracy, precision, sensitivity, specificity, FPR, FNR, and execution time. The performance of accuracy results with LDA for each class is presented in Table 4 DT also has the highest value in detecting Bashlite, Benign, and Mirai, each with value of 1. The average performance of Classification with LDA is presented in Table 5. DT has the highest accuracy that reaches 100%. DT also shows superior precision value and sensitivity, high specificity and low FPR but with value equal to that of RF, although its FNR value was lower than that of RF. The highest FPR value is achieved by k-NN, i.e.: 6.255. The highest value for FNR is achieved by RF, which reaches 1.2841. Significantly, DT has better performance than the other classification methods.

Table 4. Accuracy Each Class with LDA

Detection	k-NN	DT	RF	AB	GB
Bashlite	0.9997	1	1	0.9849	0.9956
Benign	0.9919	1	0.9999	0.9803	0.9901
Mirai	0.9981	1	1	0.9941	0.9949

Table 5. Performance Metric with LDA

Metric	k-NN	DT	RF	AB	GB
Accuracy	99.82	100	99.99	98.93	99.48
Precision	0.9984	1	0.9999	0.9908	0.9948
Sensitivity	0.995	1	0.9999	0.9852	0.9939
Specificity	0.9999	1	1	0.9992	0.9993
FPR	6.255	0	0	0.0008	0.0007
FNR	0.005	0	1.2841	0.0148	0.006

Table 6. Accuracy Each Class without LDA

Detection	k-NN	DT	RF	AB	GB
Bashlite	0,9999	1	1	0,9998	0,9999
Benign	0,9990	1	1	0,9998	0,9994
Mirai	0,9999	1	1	0,9996	0,9999

Table 7. Performance Metric without LDA

Metric	k-NN	DT	RF	AB	GB
Accuracy	99,99	100	100	99,97	99,99
Precision	0,9998	1	1	0,9997	0,9999
Sensitivity	0,9993	1	1	0,9998	0,9994
Specificity	0,9999	1	1	0,9999	0,9999
FPR	1,2511	0	0	8,2619	5,0043
FNR	0,0007	0	0	0,0002	0,0006

The performance of the accuracy results without LDA for each class is presented in Table 6. DT and RF also have the highest value in detecting Bashlite, Benign, and Mirai, each with the value of 1. The average performance of the results of classification without LDA is shown in Table 7. The best performance result is DT equal to RF for each measurement value. DT and RF were both superior compared to other classification methods. Next, k-NN, AB, and GB have highest FPR, with the highest value for AB reaches up to 8.2619.

Result Analysis

Experimental results showed that the proposed LDA dimension reduction functions well. Implementation of dimensional reduction speeds up the botnet detection process. Nevertheless, the use of low scale dimension of data slightly decreases the classification accuracy of AB, k-NN and GB. DT and RF are not affected.

The precession values of AB and GB decrease significantly, while k-NN decrease slightly. DT and RF are again not affected. Next, for sensitivity and specificity values, DT and RF still are not affected, while k-NN is not affected

only for specificity value. In contrast, the values for AB and GB relatively decrease.

The experimental results clearly show that the use of the LDA method on data dimension reduction gives an impact on IoT botnet detection, as shown in Table 8. DT and RF show the highest accuracy and stability, with or without using the LDA method, with accuracy level reaches 100%. k-NN, AB, and GB, show a slight decrease in accuracy level when using the LDA.

The use of the LDA method also has an impact on the precision value. Table 9 shows a comparison between using the LDA method and without LDA. The dimension reduction with LDA does not have an impact on DT and RF, which has a stable value of 1. However, k-NN, AB, and GB experience a slight decrease in precision.

We further evaluate the sensitivity of the proposed method. Table 10 shows a comparison of the sensitivity values when the LDA method was used and without LDA. DT and RF has stable sensitivity values with and without the LDA method, while k-NN, AB, and GB show a decrease with the LDA method.

The results of the performance evaluation of specificity values are shown in Table 11. The table shows a summary of the specificity values that impacted by the use of LDA method. On k-NN, DT, and RF the specificity values are stable and are not impacted. In contrast, AB and GB experience a decrease in specificity value when using LDA method.

Table 8. Comparison of accuracy values

Method	Without LDA	With LDA
k-NN	99.99	99.82
DT	100	100
RF	100	100
AB	99.97	98.93
GB	99.99	99.48

Table 9. Comparison of precision values

Method	Without LDA	With LDA
k-NN	0.9998	0.9984
DT	1	1
RF	1	1
AB	0.9997	0.9908
GB	0.9999	0.9948

Table 10. Comparison of sensitivity values

Method	Without LDA	With LDA
k-NN	0.9993	0.995
DT	1	1
RF	1	1
AB	0.9998	0.9852
GB	0.9994	0.9939

Table 11. Comparison of specificity values

Method	Without LDA	With LDA
k-NN	0.9999	0.9999
DT	1	1
RF	1	1
AB	0.9999	0.9992
GB	0.9999	0.9993

Table 12. Comparison of FPR values

Method	Without LDA	With LDA
k-NN	1.2511	6.255
DT	0	0
RF	0	0
AB	8.2619	0.0008
GB	5.0043	0.0007

Table 13. Comparison of FNR values

Method	Without LDA	With LDA
k-NN	0.0007	0.005
DT	0	0
RF	0	0
AB	0.0002	0.0148
GB	0.0006	0.006

FPR performance was evaluated with similar condition, as shown in Table 12. Two classification methods show decreased FPR values. Nevertheless, k-NN shows improvement with the LDA method, and exhibit significant increase, which reaches 6.255. AB and GB show an opposite trend, with a significant decrease in FPR, whereas DT and RF remained at 0.

Performance evaluation results on FNR are displayed in Table 13. While FNR DT and RF still on 0 whether using LDA or without LDA, k-NN and GB show decrement FPR values. This is opposite to AB which has slightly increment FPR value.

The use of LDA for dimensional reduction overall has positive impact on execution time as shown in Table 14. The times to execute the classification using k-NN, DT, RF, AB, and GB classifiers decrease. Executing classification using k-NN without LDA requires 30908.87 seconds and decrease drastically to 73.95 seconds when incorporating LDA dimensional reduction. Execution time of DT decreases almost double, while for AB and GB, the execution times are faster significantly when incorporating the LDA dimensional reduction. The fastest processing time for classification is achieved by k-NN, which only needs 73.95 seconds.

The experimental results show that the performance of each classification model has good results. Then validation was carried out to detect overfitting problems using K-fold cross-validation [48]. In Intrusion Detection System

(IDS) research cross-validation has been widely used, such as for validating the KNN, NB, SVM, and RF classification models in detecting DDoS attacks [49], validating the LSTM deep learning model to detect different types of attacks between R2L and U2R [50], and validating the convolution neural network model for anomaly attack detection [51]. In this experiment's validation, a value of k=10 is utilized for each classification model. In each iteration, the sampled data used will be shuffled, and then each subset will contain an equal number of samples [52]. The results of the performance evaluation with cross-validation are presented in Table 15.

Comparison with other datasets and other work

To determine the effectiveness of the use of the proposed LDA method, we compare it with 100% dataset N-BaIoT, and other datasets on the DT classification method and previous research works that use the same dataset and also implement dimensional reduction methods. Here, only results of implementation of lower dimensional data were considered. Results of the comparison of other datasets are shown in Table 16 and comparisons of other work are shown in Table 17.

Table 14. Comparison of execution time

Method	Without LDA	With LDA
k-NN	30908.87 s	73.95 s
DT	163.75 s	98.58 s
RF	675.74 s	270.36 s
AB	1143.27 s	289.11 s
GB	5404.97 s	665.13 s

Table 15 Evaluation with cross-validation

Method	Average Accuracy (%)	Error (%)
k-NN	99.76	0.0001
DT	**100**	**0.0001**
RF	**100**	**0.0001**
AB	98.95	0.0006
GB	99.48	0.0003

Table 16. Comparison results with other datasets

Metric	20% of N-BaIoT	100% of N-BaIoT	30% of MedBIoT
Accuracy	100	100	100
Precision	1	1	1
Sensitivity	1	1	1
Specificity	1	1	1
FPR	0	0	0
FNR	0	0	0

Table 17. Comparison results with other works

Ref & (Year)	Method	No. of Feature	Accuracy
[15]	Fisher Score + DT	2	98.43
[17]	Entropy + SVM	3	93.15
[19]	Fisher Score + XGBoost	3	99.96
This Work	**LDA + DT**	**2**	**100**

Discussion

We have presented detection systems with a high accuracy level and low FPR level for the identification of IoT botnets. Classification models used in the proposed system without LDA shows that DT and RF had the highest accuracy level, reaching 100%. With LDA, only DT that remains stable, while RF shows a slight decrease in accuracy. A comparison of the classification methods without using the dimensionality reduction LDA method reveals that DT has a stable accuracy of 100%, whereas the other classification methods experience decreased accuracy with LDA. Overall, the achieved levels of accuracy show that the use of the dimensionality reduction LDA method was very effective and efficient for IoT botnet detection for classification. DT generates more accurate results than other classifiers, i.e.; AB, k-NN, RF, and GB.

A detection system is better when it achieves a high accuracy level and a very low FPR value. Models with a high accuracy level and high FPR cannot be used. With or without LDA, DT and RF both show an FPR value of 0. Combining dimensionality reduction LDA with DT and RF generates high accuracy level. Thus, this fact shows that the proposed system has a good performance because a more accurate classifier was built when a lower FPR was generated.

Precision shows the reliability of the detection model in the classified sample as positive. DT and RF have an excellent ability to classify samples as positive, with a value of 1, whether using LDA or without LDA. This is different from k-NN, AB, and GB, which have no decrement of precision value when using the LDA.

Specificity represents how much correct data are predicted by detection system. DT and RF have a good ability to classify samples as positive, with or without LDA. This result is different for AB and GB, which show a decrease in the specificity value with LDA. k-NN has a flat specificity value with or without LDA.

FNR, which is the critical level when facing the detection model, is reflected from model sensitivity. DT and RF have the highest sensitivity levels with or without LDA, and their FNR values

were 0. k-NN sensitivity value increases when LDA was used, while the FNR value decreases. The reverse was the case for AB and GB, which has a decrease in the sensitivity value when LDA was used, with an increase in FNR value. Thus, in term of sensitivity value of 1 and FNR of 0, DT and RF using LDA are the best models for the IoT botnet detection system, because the models will cover all chances of detecting botnets.

The efficiency of the detection system is observed by the speed of the execution time. During the experiments of IoT botnet detection, we observe an increase in execution speed, which was significant for k-NN, DT, RF, AB, and GB classifiers with or without LDA. If we consider the high accuracy level and lowest FPR, then DT has the fastest time of execution, as it only requires 98.58s.

Compared to other studies of the same theme that use N-BaIoT dataset and with dimensionality reduction LDA method, the proposed system in this paper shows the highest accuracy, and DT classification had the highest score, i.e.: 100%. Bahsi et al. [15] use the Fisher Score dimensionality reduction method to reduce data dimension into two features, and in detecting botnet by using the DT classification method, their accuracy level reaches 98.43%. Nomm et al. [17] use the entropy method to reduce data dimension into three features, while SVM is used for its detection process, reaching only 93.15% accuracy level. Alqahtani et al. [19] also use Fisher Score to reduce data into three features, and select the XGBoost classification method to detect botnet, with the accuracy level reaching 99.96%.

The effectiveness of the detection system can be observed at the level of its accuracy. This accuracy is indicated by the use of the number of features. Without the LDA method with 115 features compared to using the LDA method, which had only two features, the accuracy level of our models remains the same. This fact suggests that the reduction in the number of features used in the IoT botnet detection system was very effective. Compared to previous studies, the proposed system is more effective in detecting IoT botnets, which is indicated by a higher level of accuracy.

CONCLUSION

The LDA dimensionality reduction method has been implemented and used to detecting IoT botnets effectively and efficiently. We showed that detection system with a very low feature numbers can reach a very high accuracy level, and those fewer features can boost up time execution as well. We observed that combining the LDA method and DT and RF classifiers in IoT Botnet detection system provides the best accuracy of 100% with a FPR value of 0. Detection times for DT and RF are 98.58 seconds and 270.36 seconds, respectively.

We propose that future studies should investigate the efficiency level of using LDA method from the perspective of energy consumption and memory used. We also consider extending the framework of the research for detecting botnet in real time fashion, using balanced data, which can boost execution time and maximize the accuracy.

REFERENCES

[1] M. S. Mahdavinejad, M. Rezvan, M. Barekatain, P. Adibi, P. Barnaghi, and A. P. Sheth, “Machine learning for internet of things data analysis: a survey,” *Digital Communications and Networks*, vol. 4, no. 3, pp. 161–175, 2018, doi: 10.1016/j.dcan.2017.10.002.

[2] K. Somasundaram and K. Selvam, “IOT – Attacks and Challenges,” *International Journal of Engineering Research & Technology*, vol. 8, no. 9, pp. 9–12, 2018, doi: 10.31873/ijetr.8.9.67.

[3] X. S. Yang, S. Lee, S. Lee, and N. Theera-Umpon, “Information Analysis of High-Dimensional Data and Applications,” *Mathematical Problems in Engineering.*, vol. 2015, no. ii, pp. 2–4, 2015, doi: 10.1155/2015/126740.

[4] A. Ullah, F. H. Khan, U. Qamar, and S. Bashir, “Dimensionality reduction approaches and evolving challenges in high dimensional data,” *ACM International Conference Proceeding Series*, pp. 1–8, 2017, doi: 10.1145/3109761.3158407.

[5] J. Wang, S. Yue, X. Yu, and Y. Wang, “An efficient data reduction method and its application to cluster analysis,” *Neurocomputing*, vol. 238, pp. 234–244, 2017, doi: 10.1016/j.neucom.2017.01.059.

[6] Z. Cheng and Z. Lu, “A novel efficient feature dimensionality reduction method and its application in engineering,” *Complexity*, vol. 2018, pp. 1-14 2018, doi: 10.1155/2018/2879640.

[7] T. Zhang and B. Yang, “Dimension reduction for big data,” *Statistics and its Interface*, vol. 11, no. 2, pp. 295–306, 2018, doi: 10.4310/SII.2018.v11.n2.a7.

[8] J. Yan *et al.*, “Effective and efficient dimensionality reduction for large-scale and streaming data preprocessing,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 3, pp. 320–332,

2006, doi: 10.1109/TKDE.2006.45.
[9] S. I. Popoola, B. Adebisi, R. Ande, M. Hammoudeh, and A. A. Atayero, "Memory-efficient deep learning for botnet attack detection in iot networks," *Electronics*, vol. 10, no. 9, pp. 1–18, 2021, doi: 10.3390/electronics10091104.
[10] V. V. Platonov and P. O. Semenov, "Dimension reduction in network attacks detection systems," *Nonlinear Phenomena in Complex Systems*, vol. 17, no. 3, pp. 284–289, Mar. 2014.
[11] H. I. Alsaadi, R. M. Almuttairi, O. Bayat, and O. N. Ucani, "Computational intelligence algorithms to handle dimensionality reduction for enhancing intrusion detection system," *Journal of Information Science and Engineering*, vol. 36, no. 2, pp. 293–308, Jan. 2020, doi: 10.6688/JISE. 202003_36(2).0009.
[12] S. H. Abbas, "Ids Feature Reduction Using Two," *International Journal of Civil Engineering and Technology* vol. 8, no. 3, pp. 468–478, Mar. 2017.
[13] J.Kittler, "Feature selection and extraction," in *Handbook of Pattern and Image Proc.*, 1st ed., San Diego, USA, 1986, pp. 59–83, doi: 10.1002/0470854774.ch9.
[14] N. Koroniotis, N. Moustafa, and E. Sitnikova, "Forensics and Deep Learning Mechanisms for Botnets in Internet of Things: A Survey of Challenges and Solutions," *IEEE Access*, vol. 7, pp. 61764–61785, 2019, doi: 10.1109/ACCESS.2019.2916717.
[15] H. Bahsi, S. Nomm, and F. B. La Torre, "Dimensionality Reduction for Machine Learning Based IoT Botnet Detection," in *Proc. 2018 15th International Conference on Control, Automation, Robotics and Vision, ICARCV*, 2018, pp. 1857–1862, doi: 10.1109/ICARCV.2018.8581205.
[16] Y. Meidan *et al.*, "N-BaIoT-Network-based detection of IoT botnet attacks using deep autoencoders," *IEEE Pervasive Computing*, vol. 17, no. 3, pp. 12–22, Sep. 2018, doi: 10.1109/MPRV.2018.03367731.
[17] S. Nomm and H. Bahsi, "Unsupervised Anomaly Based Botnet Detection in IoT Networks," in *Proc.- 17th IEEE International Conference on Machine Learning and Applications, ICMLA*, 2019, pp. 1048–1053, doi: 10.1109/ICMLA.2018.00171.
[18] J. Liu, S. Liu, and S. Zhang, "Detection of IoT botnet based on deep learning," in *Chinese Control Conference, CCC*, 2019, vol. 2019–July, no. 1, pp. 8381–8385, doi: 10.23919/ChiCC.2019.8866088.
[19] M. Alqahtani, H. Mathkour, and M. M. Ben Ismail, "IoT botnet attack detection based on optimized extreme gradient boosting and feature selection," *Sensors (Switzerland)*, vol. 20, no. 21, pp. 1–21, 2020, doi: 10.3390/s20216336.
[20] M. G. Desai, Y. Shi, and K. Suo, "IoT Bonet and Network Intrusion Detection using Dimensionality Reduction and Supervised Machine Learning," *2020 11th IEEE Annu. Ubiquitous Comput. Electron. Mob. Commun. Conf. UEMCON 2020*, pp. 0316–0322, 2020, doi: 10.1109/UEMCON51285.2020.9298146.
[21] M. Alshamkhany, W. Alshamkhany, M. Mansour, M. Khan, S. Dhou, and F. Aloul, "Botnet Attack Detection using Machine Learning," in *Proc. 14th International Conference on Innovations in Information Technology, IIT*, 2020, no. November, pp. 203–208, doi: 10.1109/IIT50501.2020.9299061.
[22] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," *Proc. Mil. Commun. Inf. Syst. Conf. MilCIS*, pp. 1-6 2015, doi: 10.1109/MilCIS.2015.7348942.
[23] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset," *Future Generation Computer Systems*, vol. 100, pp. 779–796, 2019, doi: 10.1016/j.future.2019.05.041.
[24] S. I. Popoola, B. Adebisi, M. Hammoudeh, G. Gui, and H. Gacanin, "Hybrid Deep Learning for Botnet Attack Detection in the Internet-of-Things Networks," *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4944–4956, Mar. 2021, doi: 10.1109/JIOT.2020.3034156.
[25] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, "Kitsune: An ensemble of autoencoders for online network intrusion detection," *arXiv*, no. February, pp. 18–21, 2018, doi: 10.48550/arxiv.1802.09089.
[26] A. Guerra-Manzanares, J. Medina-Galindo, H. Bahsi, and S. Nõmm, "MedBIoT: Generation of an IoT botnet dataset in a medium-sized IoT network," *ICISSP 2020 - Proc. 6th Int. Conf. Inf. Syst. Secur. Priv.*, pp. 207–218, 2020, doi: 10.5220/0009187802070218.
[27] A. Tharwat, T. Gaber, A. Ibrahim, and A. E. Hassanien, "Linear discriminant analysis: A detailed tutorial," *AI Community*, vol. 30, no. 2, pp. 169–190, 2017, doi: 10.3233/AIC-170729.
[28] M. A. Salam, A. T. Azar, M. S. Elgendy, and

K. M. Fouad, “The Effect of Different Dimensionality Reduction Techniques on Machine Learning Overfitting Problem,” *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 4, pp. 641–655, 2021, doi: 10.14569/IJACSA.2021.0120480.

[29] F. J. H. Heras and G. G. de Polavieja, “Supervised dimensionality reduction by a Linear Discriminant Analysis on pre-trained CNN features,” pp. 1-12 2020, doi: 10.48550/arxiv.2006.12127.

[30] Z. Guo and Y. Zhang, “A Similar Distribution Discriminant Analysis with Orthogonal and Nearly Statistically Uncorrelated Characteristics,” *Mathematical Problems in Engineering*, vol. 2019, pp. 1-10, 2019, doi: 10.1155/2019/3145973.

[31] S. Lee, A. Abdullah, N. Jhanjhi, and S. Kok, “Classification of botnet attacks in IoT smart factory using honeypot combined with machine learning,” *PeerJ Computer Science*, vol. 7, pp. 1–23, 2021, doi: 10.7717/PEERJ-CS.350.

[32] M. I. Hossain, S. Eshrak, M. J. Auvik, S. F. Nasim, R. Rab, and A. Rahman, “Efficient Feature Selection for Detecting Botnets Based on Network Traffic and Behavior Analysis,” in *Proc. 7th International Conference on Networking, Systems and Security*, 2020, pp. 56–62, doi: 10.1145/3428363.3428378.

[33] A. Bijalwan, N. Chand, E. S. Pilli, and C. Rama Krishna, “Botnet analysis using ensemble classifier,” *Perspectives in Science*, vol. 8, pp. 502–504, 2016, doi: 10.1016/j.pisc.2016.05.008.

[34] F. Tariq and S. Baig, “Machine Learning Based Botnet Detection in Software Defined Networks,” *International Journal of Security and Its Applications*, vol. 11, no. 11, pp. 1–12, 2017, doi: 10.14257/ijsia.2017.11.11.01.

[35] J. Velasco-Mata, V. González-Castro, E. Fidalgo, and E. Alegre, “Efficient Detection of Botnet Traffic by features selection and Decision Trees,” *arXiv*, pp. 1-20, 2021, doi: 10.48550/arxiv.2107.02896.

[36] Z. Zhang, “Introduction to machine learning: K-nearest neighbors,” *Annals of Translational Medicine,* vol. 4, no. 11, pp. 1–7, 2016, doi: 10.21037/atm.2016.03.37.

[37] Y. E. Wella et al.," Service quality dealer identification: the optimization of K-Means clustering," *SINERGI,* vol. 27, no.3, pp. 433-442, 2023, doi: 10.22441/sinergi.2023.3.014

[38] R. F. M. Dollah, M. A. Faizal, F. Arif, M. Z. Mas’ud, and L. K. Xin, “Machine learning for HTTP botnet detection using classifier algorithms,” *Journal of Telecommunication, Electronic and Computer Engineering.*, vol. 10, no. 1–7, pp. 27–30, 2018.

[39] A. Parmar, R. Katariya, and V. Patel, “A Review on Random Forest: An Ensemble Classifier,” *Lect. Notes Data Eng. Commun. Technol.*, vol. 26, pp. 758–763, 2019, doi: 10.1007/978-3-030-03146-6_86.

[40] S. Amalia, I. Deborah, and I. N. Yulita, “Comparative analysis of classification algorithm: Random Forest, SPAARC, and MLP for airlines customer satisfaction,” *SINERGI*, vol. 26, no. 2, p. 213, 2022, doi: 10.22441/sinergi.2022.2.010.

[41] X. D. Hoang and Q. C. Nguyen, “Botnet detection based on machine learning techniques using DNS query data,” *Future Internet*, vol. 10, no. 5, pp. 1–11, 2018, doi: 10.3390/FI10050043.

[42] A. Moubayed, M. N. Injadat, and A. Shami, “Optimized Random Forest Model for Botnet Detection Based on DNS Queries,” in *Proc. International Conference on Microelectronics, ICM*, 2020, vol. 2020–Decem, pp. 1-4, doi: 10.1109/ICM50269.2020.9331819.

[43] S. Chen, B. Shen, X. Wang, and S. J. Yoo, “A strong machine learning classifier and decision stumps based hybrid adaboost classification algorithm for cognitive radios,” *Sensors (Switzerland)*, vol. 19, no. 23, pp. 1-15, 2019, doi: 10.3390/s19235077.

[44] A. Rehman Javed, Z. Jalil, S. Atif Moqurrab, S. Abbas, and X. Liu, “Ensemble Adaboost classifier for accurate and fast detection of botnet attacks in connected vehicles,” *Transactions on Emerging Telecommunications Technologies*, no. June, pp. 1–18, 2020, doi: 10.1002/ett.4088.

[45] C. Bentéjac, A. Csörgő, and G. Martínez-Muñoz, *A comparative analysis of gradient boosting algorithms*, vol. 54, no. 3. Springer Netherlands, pp. 1937-1967, 2021, doi: 10.1007/s10462-020-09896-5.

[46] T. Ongun, T. Sakharaov, S. Boboila, A. Oprea, and T. Eliassi-Rad, “On Designing Machine Learning Models for Malicious Network Traffic Classification,” *arXiv*, pp. 1-9, 2019, doi: 10.48550/arxiv.1907.04846.

[47] A. Tharwat, “Classification assessment methods,” *Applied Computing and Informatics*, vol. 17, no. 1, pp. 168–192, 2021, doi: 10.1016/j.aci.2018.08.003.

[48] H. Shafique, A. A. Shah, M. A. Qureshi, and M. K. Ehsan, “Machine Learning Empowered Efficient Intrusion Detection Framework,” *VFAST Transactions on Software Engineering*, vol. 10, no. 2, pp. 27–

35, 2022.
[49] M. Aamir and S. M. A. Zaidi, “DDoS attack detection with feature engineering and machine learning: the framework and performance evaluation,” *International Journal of Information Security*, vol. 18, no. 6, pp. 761–785, 2019, doi: 10.1007/s10207-019-00434-1.
[50] Y. Imrana, Y. Xiang, L. Ali, and Z. Abdul-Rauf, “A bidirectional LSTM deep learning approach for intrusion detection,” *Expert Systems with Applications*, vol. 185, no. June 2020, p. 115524, 2021, doi: 10.1016/j.eswa.2021.115524.
[51] Z. Wang, Z. Li, D. He, and S. Chan, “A lightweight approach for network intrusion detection in industrial cyber-physical systems based on knowledge distillation and deep metric learning,” *Expert Systems with Applications* vol. 206, p. 117671, 2022, doi: doi: 10.1016/j.eswa.2022.117671.
[52] M. Artur, “Review the performance of the Bernoulli Naïve Bayes Classifier in Intrusion Detection Systems using Recursive Feature Elimination with Cross-validated selection of the best number of features,” *Procedia Computer Science*, vol. 190, no. 2019, pp. 564–570, 2021, doi: 10.1016/j.procs.2021.06.066.