

Dimensional Feature Reduction for Detecting Botnet Activities

Muhammad Aidiel Rachman Putra*, Tohari Ahmad*, Dandy Pramana Hostiadi**

*Department of Informatics, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia

** Department of Informatics, Institut Teknologi dan Bisnis STIKOM Bali, Bali, Indonesia

6025211020@mhs.its.ac.id, tohari@if.its.ac.id, dandy@stikom-bali.ac.id

Abstract— Rising number of devices linked to the internet has made computer network security crucial. Those devices may be compromised, forming botnets, one of the most severe threats to network security due to their unique characteristics. An in-depth analysis of various processes, including feature extraction, is required to develop a botnet detection model with reliable performance. In this system, feature extraction is one of feature engineering, which is part of the data pre-processing. To find the best approach, we analyze the impact of feature extraction using dimensional reduction with four techniques: Principal Component Analysis, Truncate Singular Value Decomposition, Factor Analysis, and Fast Independent Component Analysis. The feature extraction results are brought to the classification stage to analyze their impact using several machine learning algorithms such as k-NN, Decision Tree, Random Forest, Naive Bayes, and Logistic Regression. Using the CTU-13, NCC-1, and NCC-2 datasets, it is found that dimensional reduction is suitable with k-NN but not recommended for a tree-based machine learning algorithm.

Keywords— botnet detection, intrusion detection system, network infrastructure, network security, dimensional reduction

I. INTRODUCTION

As an advanced threat, a botnet may infect computers with malicious code, forming a communication network with a botmaster as the center [1]–[3]. Botnets were connected in a centralized at the beginning but have evolved into decentralized networks [4], [5]. A hybrid botnet architecture has been developed, combining numerous architectures with the concept of layering [6], where their advancement makes them even more challenging to detect.

In line with the botnet evolution, various detection models have been introduced in multiple approaches, such as machine learning [7], [8], neural networks [1], graph-based clustering [9], and the newest is a combination of various approaches with a hybrid technique [4]. Generally, a more reliable detection model comes from a proper pre-processing method.

Pre-processing in machine learning involves some stages, including data cleansing, data transformation, and feature engineering. This feature engineering focuses on processing feature attributes from datasets, whose steps are creation, transformation, selection, and extraction. Extraction reduces the feature size so an algorithm can process it without changing the original correlation or relevant information [7]. The most

dimensional reduction techniques used to facilitate feature extraction are Principal Component Analysis (PCA) [10], Singular Value Decomposition (SVD) [7], Factor Analysis (FA) [11], and Fast Independent Component Analysis (Fast ICA) [12].

This research takes those dimensional reduction techniques in the feature engineering phase to develop a botnet detection system. The goal is to find the impact of dimensional reduction on machine learning detection models. In addition, an ideal combination can be found by implementing several dimensional reduction techniques and machine learning algorithms such as k-Nearest Neighbor (k-NN), Decision Tree (DT), Random Forest (RF), Naïve Bayes (NB), and Logistic Regression (LR).

The structure of this paper begins with related works in section II. The proposed method is described in section III. Section IV explains the experimental results and analysis. Finally, the conclusion is described in Section V.

II. RELATED WORK

In general, the performance of a botnet detection model also comes from choosing the correct pre-processing technique. Some previous studies give much attention to the pre-processing phase. Joshi et al. [1] use a fuzzy logic approach as a basis for feature engineering and feature selection with feature participant analysis before the classification process with an Artificial Neural Network (ANN). Feature engineering is also implemented by Putra et al. [8] with a one-hot encoding technique to identify Botnet Attack Communication Pattern Behavior. Velasco-Mata et al. [13] take two feature selection strategies, Information Reinforcement and Gini Importance, which generate several pre-selected subsets of five, six, and seven features. In other research, Yang et al. [14] detect mobile botnets by implementing a multi-level feature extraction concept to extract info from traffic levels.

Other studies often implement dimensional reduction techniques in the feature extraction phase. Joy and Hasan [15] conduct a comparative test of the feature selection method with Minimum Redundancy and Maximum Relevance (mRMR), feature extraction with PCA, and the PCA-mRMR hybrid model for Image classification with Support Vector Machine (SVM). PCA is also used in image feature extraction in gene-culture studies [16] to analyze the impact on image processing.

As a result, feature extraction with PCA significantly improves the quality of training data processing.

Al-Ezzi et al. [7] use SVD to extract features in EEG signal data. Extraction is performed in the main components of EEG data to remove background noise and reduce dimensions to improve system performance. SVD can also be combined with the CNN algorithm in a hybrid technique to extract features and classify diabetic retinopathy data [17]. Jadhav et al. [18] consider SVD in the feature extraction process for the segmentation and classification of retinal images. PCA further reduces SVD features before being applied to the k-NN and SVM classification models. This research proves that PCA can also improve the classification performance of machine learning algorithms.

III. PROPOSED METHOD

This research proposes a feature extraction technique in the botnet detection model using several dimensional reduction methods. It is to analyze their impact on improving performance. The model consists of four main processes, pre-processing, dimensional reduction, cosine similarity measurement, and classification, as shown in Fig. 1.

A. Pre-Processing

This step is an initial procedure to eliminate data issues that may interfere with the following process. Some problems with the data include data format, dirty data, and features that are less relevant to participate in the classification process. The pre-processing stage starts with data cleansing to remove network traffic records dominated by null values.

The second sub-process of this pre-processing stage is data transformation, aiming to convert object-type data into numeric. The *SrcAddr*, *DstAddr*, *State*, and *Proto* are several features that need to transform in bidirectional network flow data. Data transformation begins with converting the IPv4 format in the *SrcAddr* and *DstAddr* features into integers. The *State* and *Proto* features are transformed according to a predefined integer taken from the python dictionary, the same as [8].

The pre-processing ends with the feature selection, which is done manually. The reduced features are irrelevant to participate in the classification process. In addition, several

features with object type and a high number of null values are also reduced. The features included in the following process are *Dur*, *Proto*, *SrcAddr*, *Sport*, *DstAddr*, *Dport*, *State*, *TotPkts*, *TotBytes*, and *Label*.

B. Dimensional Reduction

In this phase, ten of 11 features from the pre-processing stage are extracted into a new feature. One feature not included in the extraction process is the Label because of its function as the target feature to be used during training and later evaluation. The extraction process uses several dimensional reduction techniques: Truncate SVD, PCA, Fast ICA, and FA. The implementation of these four techniques on a dataset containing network traffic can be described as follows:

- Truncate SVD: Factorization of the network traffic data set (D) with the number of records j and the number of features is k denoted by (1).

$$D_{j \times k} = U_{j \times j} \Sigma_{j \times k} V_{k \times k}^T \quad (1)$$

Let the singular value of matrix D be Σ and the diagonal entries denoted as σ_i , whereas ls_i and rs_i is part of left-singular (U), and right-singular (V). Singular values of D can be composed as (2).

$$D = \begin{bmatrix} ls_{1,1} & \dots & ls_{1,k} \\ \vdots & \ddots & \vdots \\ ls_{j,1} & \dots & ls_{j,k} \end{bmatrix} \begin{bmatrix} \sigma_{1,1} & \dots & 0 \\ 0 & \ddots & 0 \\ 0 & \dots & \sigma_{j,k} \end{bmatrix} \begin{bmatrix} rs_{1,1} & \dots & rs_{1,k} \\ \vdots & \ddots & \vdots \\ rs_{j,1} & \dots & rs_{j,k} \end{bmatrix} \quad (2)$$

- PCA: if Trf_i represents the i^{th} record of the dataset with the number of records being j and $feat_a$ is the a^{th} feature out of 10 in the dataset, then the variance for each feature can be formulated as (3).

$$var(feat_a) = \frac{\sum_{i=1}^j (Trf(feat_a)_i - \overline{Trf(feat_a)})^2}{(j-1)} \quad (3)$$

While the calculation of the covariance of $feat_a$ and $feat_b$, where $a \neq b$ and $O = Trf(feat_a)$; $P = Trf(feat_b)$ is denoted as:

$$cov(feat_a, feat_b) = \frac{\sum_{i=1}^j (O_i - \bar{O})(P_i - \bar{P})}{(j-1)} \quad (4)$$

- Fast ICA: as in [12], the negative entropy approximation method in FastICA is denoted as (5), where the mean value is denoted as $E\{\cdot\}$; $G(\cdot)$ is a non-linear function; $W^T \bar{X}$ and v are the output and gaussian random variable with zero mean and unit variance, respectively.

$$J(W) = [E\{G(W^T \bar{X})\} - E\{G(v)\}]^2 \quad (5)$$

- FA: The common multivariate statistical approach [19]. This technique turns some dependent features into other features known as factors, where the first factor contains the primary information from the original dataset. In general, FA serves three purposes: (1) Reduce factor

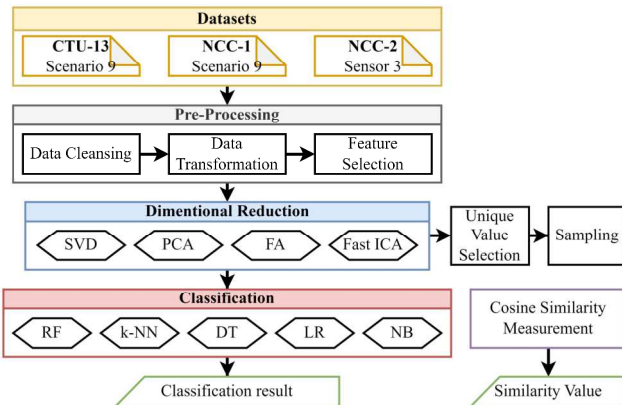


Figure 1. Proposed method

dimensions. (2) Calculate factor weights and (3) Calculate weighted summary factor scores.

Implementing several dimensional reduction techniques uses a quite popular library in the Python programming language, Skicit Learn. All parameters follow the default values set by the library.

C. Similarity Analysis

Normal activities dominate most network traffic data. In this phase, the normal traffic is separated from the botnet by analyzing the Label feature, values containing the keyword 'botnet' will be considered botnet traffic, and the other is normal. Measuring the similarity between the two groups requires in-depth analysis to make the calculation results fair. For this purpose, several approaches to balance the proportion of normal and botnet traffic are implemented: Random Undersampling, Random Oversampling, Oversampling with 0 value, and Unique Value Selection.

- Random under-sampling is a process to reduce the amount of normal traffic by selecting randomly according to the amount of botnet traffic.
- Random over-sampling increases the number of records from botnet traffic. Sampling network traffic records is done randomly from existing data.
- Over-sampling with a 0 value adds the number of records from botnet traffic with a value of 0.
- Unique value selection selects botnets and normal traffic unique values before the over-sampling or under-sampling technique.

Furthermore, botnet and normal traffic data are collected to calculate the similarity value, which is done by using the cosine similarity formula. If $nTrf$ is normal traffic in a collection of normal traffic ($NTrf$) and botnet traffic is $bTrf$ in the collection of botnet traffic ($BTrf$), similarity measurement can be expressed in (6), where $i = 1, 2, 3, 4, \dots, j$; j is number of Trf in $BTrf$ or $NTrf$.

$$Cos(BTrf, NTrf) = \frac{\sum_{x=0}^j bTrf \cdot nTrf}{\sqrt{\sum_{i=1}^j bTrf^2} \sqrt{\sum_{i=1}^j nTrf^2}} \quad (6)$$

D. Classification

This process receives input from network traffic records with two features, namely the Label feature as the target and the extracted features using dimensional reduction techniques. The classification is performed using machine learning algorithms DT, RF, LR, NB, and k-NN. This process begins with data splitting to divide the data into training as much as 80% of the total data and data testing, which uses all input data. The training results in the form of learning models are used in the classification process with several algorithms. The output of this process is a collection of normal and botnet activities, whose classification process is illustrated in Fig. 2.

IV. RESULT AND DISCUSSION

This section discusses the results of testing using the Python 3.10 programming language. The tests are carried out on a personal computer with Intel Core i7-9700F 3.00GHz

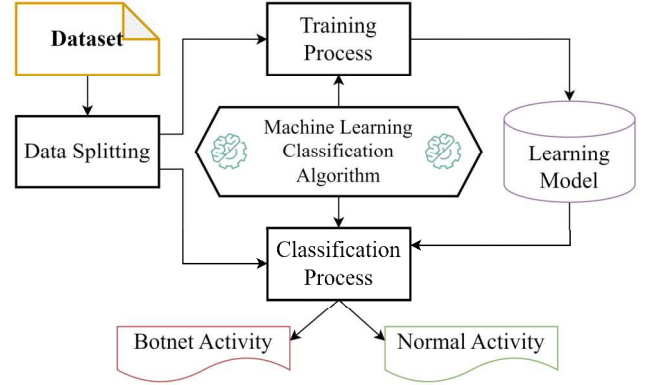


Figure 2. Classification process

processor specifications, 256GB SSD Storage, and 16GB RAM using the CTU-13 [20] scenario 9, NCC-1 [21] scenario 9, and NCC-2 [22] sensor 3 datasets. These sub-datasets are chosen because they have more than one bot and relatively large network traffic.

A. Comparison of Dimensional Reduction Techniques

Dimensional reduction produces a new feature with a single value. Botnets and normal traffic can be compared using a similarity measurement approach to show how effective this dimensional reduction is. Similarity measurement produces values that are in the range of between -1 and 1. If the similarity value is 0, then botnet traffic and normal traffic have no similarities. The higher the similarity measurement value, the more similar the botnet and normal traffic are. Meanwhile, the value below 0 indicates that the botnet and normal traffic are opposite. In other words, the dimensional reduction model gets better if the similarity measurement value is closer to -1.

Testing results on the NCC-2 dataset show that from all the dimensional reduction methods using different sampling processes, the PCA technique obtains the lowest value of -0.0043, which is processed through the Random Oversampling process without selecting unique values. The PCA technique obtained the optimal value due to its characteristics of finding correlations between features. The results of similarity on the NCC-2 dataset with several dimensional reduction methods are provided in Table 1.

Table 2 shows the similarity measurement value of each dimensional reduction method on the NCC-1 dataset. Truncate SVD is a dimensional reduction method that produces the highest average similarity value of 0.7701. The highest similarity value of 0.9635 is obtained by the Truncate SVD method, which goes through the Random Undersampling process. The low performance of the Truncate SVD technique is caused by the non-linear characteristic of traffic data, while this model is more suitable for linear data.

In testing results using the CTU-13 dataset, the lowest similarity value is -0.0106 with the PCA technique, which goes through the Random Undersampling process on selected unique values data. Even though the lowest similarity value is found obtained by the PCA technique, on average, it is not better than Fast ICA. It is because the values obtained from

TABLE 1. SIMILARITY MEASUREMENT ON NCC-2 DATASET

Dataset	Unique Selection		Sampling Rule		Similarity Measurement			
	<i>Botnet</i>	<i>Normal</i>	<i>Oversampling</i>	<i>Undersampling</i>	<i>PCA</i>	<i>Truncate SVD</i>	<i>Fast ICA</i>	<i>FA</i>
NCC-2 Sensor 3	No	No	No	Random	-0.0038	0.9640	-0.0038	0.0001
	No	No	Random	No	-0.0043	0.9636	-0.0042	-0.0001
	No	No	with 0	No	0.0024	0.2738	0.0024	0.0008
	No	Yes	No	Random	-0.0006	0.9608	0.0031	0.0000
	No	Yes	Random	No	0.0038	0.9609	0.0040	0.0001
	No	Yes	with 0	No	0.0029	0.3132	0.0035	0.0006
	Yes	Yes	No	Random	0.0086	0.9609	0.0043	0.0003
	Yes	Yes	Random	No	0.0026	0.9616	0.0027	0.0000
	Yes	Yes	with 0	No	0.0015	0.1304	0.0008	0.0005
Average					0.0015	0.7210	0.0014	0.0002
Minimum Value					-0.0043	0.1304	-0.0042	-0.0001
Maximum Value					0.0086	0.9640	0.0043	0.0008

TABLE 2. SIMILARITY MEASUREMENT ON NCC-1 DATASET

Dataset	Unique Selection		Sampling Rule		Similarity Measurement			
	<i>Botnet</i>	<i>Normal</i>	<i>Oversampling</i>	<i>Undersampling</i>	<i>PCA</i>	<i>Truncate SVD</i>	<i>Fast ICA</i>	<i>FA</i>
NCC-1 Scenario 9	No	No	No	Random	-0.0131	0.9635	-0.0090	-0.0006
	No	No	Random	No	-0.0113	0.9633	-0.0114	-0.0003
	No	No	with 0	No	-0.0053	0.3884	-0.0053	-0.0004
	No	Yes	No	Random	-0.0025	0.9612	-0.0058	-0.0004
	No	Yes	Random	No	-0.0043	0.9612	-0.0044	-0.0003
	No	Yes	with 0	No	-0.0032	0.5407	-0.0036	-0.0015
	Yes	Yes	No	Random	0.0013	0.9616	-0.0019	0.0005
	Yes	Yes	Random	No	-0.0036	0.9613	-0.0037	0.0001
	Yes	Yes	with 0	No	-0.0010	0.2295	-0.0014	-0.0003
Average					-0.0048	0.7701	-0.0052	-0.0003
Minimum Value					-0.0131	0.2295	-0.0114	-0.0015
Maximum Value					0.0013	0.9635	-0.0014	0.0005

PCA tend to be low in oversampling data with a value of 0. In contrast, these values do not correspond to the average value resulting from dimensional reduction with the PCA technique. Table 3 presents all results of calculating similarity values between botnets and normal traffic using several dimensional reduction methods on the CTU-13 dataset.

In general, Fast ICA is the most recommended dimensional reduction method. In all three datasets, its average similarity value is mostly lower than in the other methods. On the other hand, Truncate SVD is a dimensional reduction method considered unsuitable for botnet datasets because of its dependence on linear data. The average similarity value of Truncate SVD is the highest in each dataset.

B. Detection Result

The proposed method has tested the extraction results using the dimensional reduction technique on a machine learning-based detection model. The goal is to find the impact of feature

extraction with the dimensional reduction method. Tests are carried out using five machine-learning classification algorithms: DT, RF, NB, k-NN, and LR, and the detection results are compared to the performance with and without feature extraction. Accuracy, precision, and recall are used as the basis for analyzing the detection results.

Evaluation of the performance of the detection model on the NCC-2, NCC-1, and CTU-13 datasets is shown in Tables 4, 5, and 6. Overall, the model with feature extraction works well when combined with the k-NN algorithm. On the other hand, tree-based algorithms are not recommended to go through the feature extraction process on the data. Feature extraction using the dimensional reduction method creates a single feature, causing the model formed by a tree-based algorithm to have fewer complex rules. Consequently, it decreases the detection performance as well. The NB and LR algorithms still require more in-depth analysis, although, in general, their performance is quite good when combined with feature extraction.

TABLE 3. SIMILARITY MEASUREMENT ON CTU-13 DATASET

Dataset	Unique Selection		Sampling Rule		Similarity Measurement			
	Botnet	Normal	Oversampling	Undersampling	PCA	Truncate SVD	Fast ICA	FA
CTU-13 Scenario 9	No	No	No	Random	-0.0050	0.9754	-0.0074	-0.0009
	No	No	Random	No	-0.0064	0.9755	-0.0065	-0.0023
	No	No	with 0	No	-0.0032	0.3007	-0.0032	-0.0043
	No	Yes	No	Random	-0.0081	0.9726	-0.0091	-0.0028
	No	Yes	Random	No	-0.0095	0.9729	-0.0093	-0.0030
	No	Yes	with 0	No	-0.0039	0.3398	-0.0059	-0.0037
	Yes	Yes	No	Random	-0.0106	0.9709	-0.0060	-0.0052
	Yes	Yes	Random	No	-0.0094	0.9705	-0.0097	-0.0030
Average					-0.0067	0.7543	-0.0070	-0.0033
Minimum Value					-0.0106	0.3007	-0.0097	-0.0052
Maximum Value					-0.0032	0.9755	-0.0032	-0.0009

TABLE 4. DETECTION RESULT ON NCC-2 DATASET

Alg.	Result	Dimensional Reduction Method (%)				
		Not Use	PCA	SVD	Analysis	Fast ICA
DT	Acc	99.98	98.78	98.31	98.22	96.32
	Prec	99.80	94.39	92.80	97.50	79.63
	Rec	99.87	89.14	84.19	78.50	68.93
RF	Acc	99.98	98.73	98.34	99.72	96.52
	Prec	99.93	94.07	93.12	97.82	81.68
	Rec	99.80	88.77	84.36	98.45	69.60
NB	Acc	52.51	91.44	92.41	91.25	91.45
	Prec	13.35	10.42	0	9.23	10.45
	Rec	96.08	1.73	0	1.77	1.72
k-NN	Acc	99.65	99.71	99.67	99.50	99.70
	Prec	97.32	98.16	97.85	96.91	98.11
	Rec	98.03	97.96	97.84	96.45	97.98
LR	Acc	92.43	74.96	92.43	92.43	92.43
	Prec	~	15.30	0	~	~
	Rec	0	50.94	0	0	0

V. CONCLUSION

This research examines a botnet detection model using the feature extraction approach with the dimensional reduction method: PCA, Truncate SVD, Fast ICA, and FA. The extracted features are brought into the detection model with five machine-learning classification algorithms: DT, RF, NB, k-NN, and LR. The effectiveness of the dimensional reduction method is tested by comparing botnet and non-botnet data using cosine similarity calculations. Meanwhile, performance evaluation parameters of accuracy, precision, and recall are used to analyze the impact of feature extraction on the detection model.

The similarity measurement result shows that Fast ICA is the most effective dimensional reduction method. On the other hand, Truncate SVD has yet to deliver satisfactory results. Thus, feature extraction works well on the model with the k-NN algorithm, with the evaluation results on the NCC-2 dataset getting the highest accuracy value of 99.71% and the highest precision of 98.16%. As for testing the NCC-1 model dataset

TABLE 5. DETECTION RESULT ON NCC-1 DATASET

Alg.	Result	Dimensional Reduction Method (%)				
		Not Use	PCA	SVD	Analysis	Fast ICA
DT	Acc	99.97	98.46	97.75	99.47	96.15
	Prec	99.89	96.83	92.90	99.20	91.53
	Rec	99.89	91.99	90.82	96.99	79.89
RF	Acc	99.98	98.38	97.55	99.71	96.35
	Prec	99.96	94.38	95.14	99.00	91.79
	Rec	99.89	94.04	86.94	98.91	81.12
NB	Acc	43.43	87.91	85.52	88.13	87.91
	Prec	19.79	67.74	5.67	68.19	67.74
	Rec	99.76	25.85	0.23	28.32	25.85
k-NN	Acc	99.51	99.60	99.54	99.33	99.60
	Prec	98.12	98.72	98.48	97.94	98.70
	Rec	98.36	98.41	98.21	97.25	98.42
LR	Acc	85.94	73.35	86.01	86.02	86.02
	Prec	34.36	26.76	0	~	~
	Rec	0.57	52.17	0	0	0

with feature extraction, it obtained the highest score at 99.60% accuracy, 98.72% precision, and 98.42% recall. In testing with the CTU-13 model dataset with feature extraction and the k-NN algorithm also obtained the highest accuracy value of 99.29%, the highest precision of 96.87%, and 95.08% recall.

The experiment can conclude that k-NN+PCA is the best combination for botnet detection using bidirectional network flow data, with an average performance of 99.53% accuracy, 97.92% precision and 97.15% recall. In contrast, the dimensional reduction technique is unsuitable for tree-based machine learning algorithms like DT and RF because a single extracted feature forms a model with less complex rules and has an impact on decreasing the detection performance. Models with NB and LR algorithms perform well when combined with feature extraction but still require more in-depth analysis to improve detection performance.

In the future, the opportunities for research on feature extraction will be quite significant. For example, analyzing the most effective number of extracted features suitable for the

TABLE 6. DETECTION RESULT ON CTU-13 DATASET

Alg.	Result	Dimensional Reduction Method (%)				
		Not Use	PCA	SVD	Analys is	Fast ICA
DT	Acc	99.84	98.97	94.48	94.44	91.93
	Prec	99.03	90.66	95.97	94.79	73.43
	Rec	99.15	98.56	39.36	39.48	13.99
RF	Acc	99.88	99.05	94.50	99.61	91.97
	Prec	99.20	92.40	96.36	97.94	75.26
	Rec	99.44	97.34	39.40	97.67	14.05
NB	Acc	39.29	91.14	91.14	91.14	91.14
	Prec	12.70	~	~	~	~
	Rec	99.63	0	0	0	0
k-NN	Acc	99.06	99.29	99.03	98.94	99.28
	Prec	94.82	96.87	95.61	95.80	96.85
	Rec	94.57	95.08	93.31	92.10	95.01
LR	Acc	91.14	34.57	91.14	91.14	91.14
	Prec	~	10.33	0	~	~
	Rec	0	83.08	0	0	0

detection model and the best feature combinations brought to the extraction process. In addition, future research will be carried out by adding sample datasets to assist in a more profound analysis.

ACKNOWLEDGMENT

This work has been supported by The Institut Teknologi Sepuluh Nopember (ITS) and PMDSU Scholarship from The Ministry of Education, Culture, Research and Technology, The Republic of Indonesia.

REFERENCES

- [1] C. Joshi, R. K. Ranjan, and V. Bharti, "A Fuzzy Logic based feature engineering approach for Botnet detection using ANN," *J. King Saud Univ. - Comput. Inf. Sci.*, 2021, doi: 10.1016/j.jksuci.2021.06.018.
- [2] K. S. Huancayo Ramos, M. A. Sotelo Monge, and J. Maestre Vidal, "Benchmark-Based Reference Model for Evaluating Botnet Detection Tools Driven by Traffic-Flow Analytics," *Sensors*, vol. 20, no. 16, 2020, doi: 10.3390/s20164501.
- [3] W. Wang, Y. Shang, Y. He, Y. Li, and J. Liu, "BotMark: Automated botnet detection with hybrid analysis of flow-based and graph-based traffic behaviors," *Inf. Sci. (Ny)*, vol. 511, pp. 284–296, 2020, doi: 10.1016/j.ins.2019.09.024.
- [4] D. P. Hostiadi and T. Ahmad, "Hybrid model for bot group activity detection using similarity and correlation approaches based on network traffic flows analysis," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 7, pp. 4219–4232, 2022, doi: 10.1016/J.JKSUCI.2022.05.004.
- [5] R. U. Khan, X. Zhang, R. Kumar, A. Sharif, N. A. Golilarz, and M. Alazab, "An adaptive multi-layer botnet detection technique using machine learning classifiers," *Appl. Sci.*, vol. 9, no. 11, 2019, doi: 10.3390/app9112375.
- [6] I. Apostol, A.-D. Tica, and V.-V. Patriciu, "Design and implementation of a novel hybrid botnet," in *2022 14th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, 2022, pp. 1–6, doi: 10.1109/ECAI54874.2022.9847442.
- [7] A. Al-Ezzi, N. Kamel, A. Al-Shargabi, N. Yahya, I. Faye, and M. I. Al-Hiyali, "SVD-Based Feature Extraction Technique for the Improvement of Effective Connectivity Detection," in *2021 International Conference on Intelligent Cybernetics Technology and Applications, ICICyTA 2021*, 2021, pp. 52–57, doi: 10.1109/ICICyTA53712.2021.9689141.
- [8] M. A. R. Putra, T. Ahmad, and D. P. Hostiadi, "Analysis of Botnet Attack Communication Pattern Behavior on Computer Networks," *Int. J. Intell. Eng. Syst.*, vol. 15, no. 4, 2022, doi: <https://doi.org/10.22266/ijies2022.0831.48>.
- [9] S. Chowdhury, M. Khanzadeh, R. Akula, F. Zhang, S. Zhang, H. Meda, M. Marufuzzaman, and L. Bian, "Botnet detection using graph-based feature clustering," *J. Big Data*, vol. 4, p. 14, 2017, doi: 10.1186/s40537-017-0074-7.
- [10] Y. Ono and Y. Mitani, "Evaluation of feature extraction methods with ensemble learning for breast cancer classification," *LifeTech 2022 - 2022 IEEE 4th Glob. Conf. Life Sci. Technol.*, pp. 194–195, 2022, doi: 10.1109/LifeTech53646.2022.9754789.
- [11] B. Zhao, M. O. Ulfarsson, J. R. Sveinsson, and J. Chanussot, "Mixtures of Factor Analyzers and Deep Mixtures of Factor Analyzers Dimensionality Reduction Algorithms for Hyperspectral Images Classification," in *International Geoscience and Remote Sensing Symposium (IGARSS)*, 2019, pp. 891–894, doi: 10.1109/IGARSS.2019.8898002.
- [12] J. Liu, H. Song, H. Sun, and H. Zhao, "High-Precision Identification of Power Quality Disturbances under Strong Noise Environment Based on FastICA and Random Forest," *IEEE Trans. Ind. Informatics*, vol. 17, no. 1, pp. 377–387, 2021, doi: 10.1109/TII.2020.2966223.
- [13] J. Velasco-Mata, V. González-Castro, E. F. Fernández, and E. Alegre, "Efficient Detection of Botnet Traffic by Features Selection and Decision Trees," *IEEE Access*, vol. 9, pp. 120567–120579, 2021, doi: 10.1109/ACCESS.2021.3108222.
- [14] M. Yang and Q. Wen, "A multi-level feature extraction technique to detect mobile botnet," in *2016 2nd IEEE International Conference on Computer and Communications (ICCC)*, 2016, pp. 2495–2498, doi: 10.1109/CompComm.2016.7925148.
- [15] A. A. Joy and M. A. M. Hasan, "A Hybrid Approach of Feature Selection and Feature Extraction for Hyperspectral Image classification," in *2019 International Conference on Computer, Communication, Chemical, Materials and Electronic Engineering (IC4ME2)*, 2019, pp. 1–4, doi: 10.1109/IC4ME247184.2019.9036617.
- [16] C. Wu, G. Qi, H. Zhao, and Z. Chen, "Feature Extraction of Cultural Gene Image Based on PCA Method," in *2020 International Conference on Computer Engineering and Application (ICCEA)*, 2020, pp. 860–863, doi: 10.1109/ICCEA50009.2020.00189.
- [17] M. Nahiduzzaman, M. R. Islam, S. M. R. Islam, M. O. F. Goni, M. S. Anower, and K.-S. Kwak, "Hybrid CNN-SVD Based Prominent Feature Extraction and Selection for Grading Diabetic Retinopathy Using Extreme Learning Machine Algorithm," *IEEE Access*, vol. 9, pp. 152261–152274, 2021, doi: 10.1109/ACCESS.2021.3125791.
- [18] A. S. Jadhav, R. V. Pawar, and P. B. Patil, "Segmentation and Classification of Retina Images using SVD Features," in *2021 5th International Conference on Electrical, Electronics, Communication, Computer Technologies and Optimization Techniques (ICEECCOT)*, 2021, pp. 712–716, doi: 10.1109/ICEECCOT52851.2021.9707985.
- [19] L. Gong, S. Xie, Y. Zhang, M. Wang, and X. Wang, "Hybrid feature selection method based on feature subset and factor analysis," *IEEE Access*, vol. PP, p. 1, 2022, doi: 10.1109/ACCESS.2022.3222812.
- [20] S. García, M. Grill, J. Stiborek, and A. Zunino, "An empirical comparison of botnet detection methods," *Comput. Secur.*, vol. 45, pp. 100–123, 2014, doi: <https://doi.org/10.1016/j.cose.2014.05.011>.
- [21] D. P. Hostiadi and T. Ahmad, "Dataset for Botnet group activity with adaptive generator," *Data Br.*, vol. 38, 2021, doi: <https://doi.org/10.17632/4vftxh97m8.1>.
- [22] M. A. R. Putra, D. P. Hostiadi, and T. Ahmad, "Botnet dataset with simultaneous attack activity," *Data Br.*, vol. 45, p. 108628, Dec. 2022, doi: 10.1016/J.DIB.2022.108628.