INTEK Jurnal Penelitian
Vol. 9, No. 2, pp. 98-104, Oktober 2022

Brute-force Detection Using Ensemble Classification

Ekky Kharismadhany [1,*,a] , Maretha Ruswiansari [1,*,b] , Tri Harsono [1,c]
[1] Computer Engineering Major, PENS, Surabaya, Indonesia
[a] kharisma1770@ce.student.pens.ac.id, [b]maretha@pens.ac.id, [c]trison@pens.ac.id

***Abstract*—Traditional brute-force is a dictionary-based attack that tries to unlock an authentication process in service. This type of brute force can be applied in web and SSH services, and brute-force XSS injects JavaScript code. In this paper, we explore four types of ensemble classifiers using CIC-CSE-IDS 2018 to determine which yields the highest accuracy, recall, precision, and F1 in detecting three types of brute force. The first step of the research is to normalise the dataset with the tanH operator. The second step is to train the single classifier to determine three types of single classifiers combined as ensemble classifiers. The last step is predicting and comparing the results of four ensemble classifiers. The stacking algorithm achieves the best test result that reaches 94.87%, 99.94%, 98.82%, and 99.37% for accuracy, precision, recall, and F1, respectively.**

Index Terms—brute-force; ensemble classifier; SMOTE

I. Introduction

A brute force has various ways to manifest itself. The Open Web Application Project (OWASP) describes that an attacker may use a dictionary attack (with or without mutation) or a traditional brute-force attack (with given classes of characters, e.g., alphanumeric case (in)sensitive). However, the attacker has a predetermined value, like a dictionary consisting of words obtained from various sources, like a content management system or software such as dirBuster. Considering a given method, the number of tries, the efficiency of the system which conducts the attack, and the estimated efficiency of the attacked system can calculate how long it will take to submit all chosen predetermined values.

The secure shell (SSH) is a cheap, software-based solution for keeping prying eyes away from the data on a network. The SSH protocols cover authentication, encryption, and the integrity of data transmitted over the network. The system administrator or developer uses SSH features to do secure logins, file copying, and secure invocation of remote commands on a remote host [1]. SSH Brute force attacks try to access a remote cloud service or machine by performing an authentication attempt on a remote machine. The process continues until the username and password are matched. The attacker would use an application, such as hydra, for this process. The other type of brute force which uses the exact mechanism is a web-targeted brute force attack.

XSS (Cross-Site Scripting) Brute force is another type of brute force that inject code into a computer. This code can get users' personal information and send it back to the attacker. There are two types of XSS brute force. Persistent XSS brute force can be done by writing a script designed to run when a user visits the injected page and sends the information taken from the victim's computer to the attacker's server [2]. Non-persistent XSS brute force is usually done by sending the victim an unalarming message through their email or a regular website. Suppose the user clicks the link inside these messages to execute the script.

The Kaspersky report in 2021 shows that brute force attack is the highest initial attack vector with the variation of attack duration between hours and months, followed by vulnerability attacks and malicious email [3]. This initial attack vector is followed by ransomware, which has the highest impact, data leakage, and money theft. In the same study, Kaspersky also found that 51% of attack detection is done after the attack was started, and remediation duration is done in weeks or months, representing 37% of the total remediation duration category.

Many developments have been made in machine learning applications to detect brute force attacks in recent years. To improve the detection rate before an attack, a machine learning technique was used to help detect the initial attack vector. This paper used CIC-CSE-IDS 2018 as a dataset and standard single classifier methods such as decision trees, naive Bayes, random forest, support vector

DOI : http://dx.doi.org/10.31963/intek.v9i2.3350

machine, and KNN. By combining three single classifier methods, this paper combined these methods into four ensemble classifier methods, majority voting, bagging, stacking, and boosting. This paper compared the results of these ensemble classifiers.

II. Research Methodology

A. Proposed Approach

The development approach to making a machine learning model was consisted of four phases. The four phases were dataset preparation, feature engineering, ensemble classifier training, and classification metric. The pre-processing data stage consists of invalid and empty data removal, feature normalization, and feature selection on the dataset. This study used two feature selection methods, chi-square, and Spearman correlation. Because of the dataset had imbalanced target class frequency, this study used SMOTE technique to balance the frequency. The goal of this phase was to prepare datasets for further processing. In the data training phase, common single classifier methods such as decision tree, support vector machine, naive Bayes, and random forest. This study compared these methods' metrics, choosed the three best methods, and combined these three methods to make an ensemble classifier. This study used four ensemble classifier methods: majority voting, bagging, stacking, and boosting for the ensemble classifier method. The last phase of the proposed approach was to test the model. This study used four metrics, accuracy, recall, precision and F1. These metrics used to determine the model's performance to detect brute force.

B. *Experiment Procedure*

1. *Dataset Preparation*

This study uses CSE-CIC-IDS 2018 dataset to build the model. This dataset was a collaborative project conducted by Communication Security Establishment and the Canadian Institute of Cybersecurity. The dataset had 77 features and 11 types of network activity [8]. Table 1 shows each network activity type's percentage and the total number of rows. As shown in the table, the dataset has an imbalanced network type, with 77.55% of the dataset having a benign network type, and the smallest is SQL injection with 0.001% parts of the dataset.

Table 1. The Percentage and Total Dataset's Features Label

NO	TYPE	PERCETAGE	TOTAL
1	Benign	77.55%	4883142
2	DDOS attack-HOIC	10.93%	686012
3	Bot	4.56%	286191
4	FTP-Brute force	3.08%	193360
5	SSH-Brute force	2.98%	187589
6	DoS attacks-GoldenEye	0.66%	41508
7	DoS attacks-Slowloris	0.17%	10990
8	DDOS attack-LOIC-UDP	0.27%	1730
9	Brute Force -Web	0.009%	611
10	Brute Force - XSS	0.003%	230
11	SQL Injection	0.001%	87

Before the dataset can be processed further, this study checks its property for invalid values such as infinity and NaN values. This action was needed to ensure the program did not crash when it encountered these values. This research split the dataset into two parts: training and testing set. The training set had 80% of the total dataset, and the rest of the dataset was allocated to the testing set.

2. *Feature Engineering*

As shown in Table 1, the dataset is imbalanced. This study used the synthetic minority over-sampling technique (SMOTE) to balance the dataset. SMOTE is a dataset manipulation technique which under samples the majority class and oversamples the minority class. This technique uses a k-nearest neighbour algorithm to generate data rows of the minority class synthetically. This study assumed a class was a minority if the class has less than 100.000 rows. The usage goals of SMOTE was to improve the classifier's performance, as shown in Ramezankhani A. and Azizi F.'s study [9,10].

This study used the tanH operator to scale the dataset into 0 – 1 range values. TanH operator has been reported by Nandakumar, Ross, and Jain in Thangasamy S. and Latha L. studies has robust and efficient performance [11].

$$S'_k = \frac{1}{2} * \left(tanh \frac{0.01 * S_k - u_g h}{\sigma_g h} + 1 \right) \quad (1)$$

DOI : http://dx.doi.org/10.31963/intek.v9i2.3350

Where S'k is the product of the operation, the μ and σ are the row's mean and standard deviation respectively. As the dataset has many features, this study used features selection techniques such as chi-square and Spearman correlation techniques to reduce the number of features. This feature selection technique was also implemented in Fitni, Q.R.S and Ramli K, showing that chi-square has faster learning time and Spearman correlation has higher learning metric results [12].

3. *Ensemble Classifier*

The processed data set was used to train a single classifier. There were five single classifiers used in this study. The five single classifiers were decision tree, support vector machine, k-nearest neighbour, naive Bayes, and random forest. The four single classifiers were trained using a dataset that had been processed using Spearman and chi-square techniques. The training results of the single classifiers were compared to get the three single classifiers that have the best performance metrics. The feature selection technique that yields the highest metric was also used for ensemble classifiers training. The three best single classifiers were made into ensemble classifiers. This study used four ensemble classification algorithms: majority voting, boosting, bagging, and stacking.

4. *Classification Metrics*

This study used common metric to determine the performance of the model. These metrics were the following:

- *Accuracy*

The percentage of samples which correctly classified compared to total samples.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (2)$$

- *Recall*

The proportion of all x categories samples that eventually correctly classified as x categories. This metric reflects the ability of the classifier to detect anomalies.

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

- *Precision*

The ratio between correctly classified categories and falsely classified categories.

$$Precision = \frac{TP}{TP+FP} \quad (4)$$

- *F1*

The F1 score is a ratio between recall and precision. F1 metric also shows the relationship between false positives and false negatives observed in the experiment.

$$F1 = \frac{2*precision*recall}{precision+recall} \quad (5)$$

- Area Under Curve Test

The metric is a probabilistic curve between the true positive rate and false positive rate at some threshold value. This metric also shows how machine learning differentiates the classification target class. The interpretation of the area under curve graph is as follows:

  - The average value of sensitivity for all possible values of specificity
  - The average value of specificity for each possible value of sensitivity.

- Data Retention Test

The application would be tested on how the application can handle the data sent by the sniffer. The formula used for the testing:

$$dataRetention = \frac{TotalDataSaved}{TotalDataSent} * 100\% \quad (6)$$

III. Results and Discussion

This study used kaggle online environment in the experiment. The python programming language was used alongside sklearn, NumPy, and pandas library.

A. Dataset Preparation

The CSE-CIC-IDS 2018 dataset contained packet capture (pcap), logs, labels, and the comma-separated-value (CSV) files from the dataset's source. The normal (benign) and attack- type network activity was distributed

DOI : http://dx.doi.org/10.31963/intek.v9i2.3350

inside several CSV files. This study used CSV files to train and test the model. Inside the dataset, this study found that some missing and infinite-value columns need to be deleted. This removal goal was to avoid the program throwing an error. Table 2 shows the amount of infinite-value and missing-value inside the dataset.

Table 2. Missing and Infinity Values

Data Type	Total
Missing Value	17079
Infinity Value	20280

The dataset was applied to the label encoding operation. The label encoding operation was an operation that was used to change a categorical feature into a numerical feature. This study applied this operation to the 'Label' feature. As the feature has 11 unique values, this study also maps it into 11 different integer values. Table 3 shows the categorical feature with its corresponding integer value.

Table 3. Categorical Value Mapping to Numerical Mapping

Category	Numerical
Benign	1
DDOS attack-HOIC	2
Bot	3
FTP-Brute force	4
SSH-Brute force	5
DoS attacks-GoldenEye	6
DoS attacks-Slowloris	7
DDOS attack-LOIC-UDP	8
Brute Force -Web	9
Brute Force - XSS	10
SQL Injection	11

B. Feature Engineering

The cleansed dataset will be a subject of the feature engineering process. The feature engineering process aimed to scale and reduce features inside the dataset. The dataset features were selected with two methods, chi-square and Spearman's coefficient ranking correlation. Chi-square's test of independence used two hypotheses to determine which features have a strong relationship with label features. The following hypothesis was used to determine the connection between the tested and response features:

- H0 = tested feature is connected to response feature
- H1 = tested feature is not connected to response feature

The hypothesis was tested using an alpha value of 95%. If the test failed to reject the null hypothesis, the corresponding feature would be deleted from the dataset. Ten features were considered duplicates of other features; thus, those were deleted from the training and testing set. Table 4 shows ten features that are deleted from the training and testing set:

Table 4. The Removed Feature from Dataset Using Chi-Square Technique

No	Feature Name	No	Feature Name
1	Bwd Blk Rate Avg	6	CWE Flag Count
2	Bwd Byts/b Avg	7	Fwd Blk Rate Avg
3	Bwd PSH Flags	8	Fwd Byts/b Avg
4	Bwd Pkts/b Avg	9	Fwd Pkts/b Avg
5	Bwd URG Flags	10	Fwd URG Flag

Spearman correlation ranking selected highly correlated features. This study deleted features with robust correlation based on features' correlation coefficient. Table 5 shows the coefficient value and its corresponding description.

Table 5. Spearman Coefficient and Description

No	Value	Description
1	0.0 – 0.19	Very Weak
2	0.2 – 0.39	Weak
3	0.4 – 0.59	Medium
4	0.6 – 0.79	Strong
5	0.8 – 1.0	Very Strong

There are 45 features deleted from the training and testing set by using spearman correlation ranking correlation.

C. Ensemble Classifier Training and Metric Evaluation

This study compareed the average results metrics of five single classifier, to determine which feature selection techniques to use for ensemble classifier training. As a

DOI : http://dx.doi.org/10.31963/intek.v9i2.3350

reference, this study also included the average results metrics trained with the data without feature selection. Table 6 shows the average metric value for each single classifier algorithm.

Table 6. The Comparison of Single Classifier Metrics

Algorithm	No Feature Selection	Chi Square	Spearman
Decision Tree	97.17%	100%	96.61%
Naive Bayes	34.12%	33.98%	61.07%
SVM	76.60%	78.94%	78.99%
Random Forest	97.77%	97.82%	97.77%
KNN	98%	99.94%	98.82%

The chi-square performs well, with the highest being 100% average performance metric using the decision tree classifier. Other classifiers such as support vector machine, random forest, and k-nearest neighbour also perform better when the training data was being processed with chi-square feature selection. The only exception was naive Bayes with 61.07% while using Spearman feature selection.

The ensemble classifier training and evaluation phase used the chi-square feature selection technique, decision tree, random forest, and KNN as single classifier components. Four ensemble classifier algorithms were evaluated to determine which algorithms have the highest metrics. Table 7 shows the frequency of each target label inside the processed database.

Table 8 shows brute force average detection metrics for each ensemble classifier. The stacking ensemble algorithm had the highest average metric with 94.87%, 99.94%, 98.82%, and 99.37% in accuracy, precision, recall, and F1, respectively. This metric showed that the stacking ensemble algorithm de- livers accurate results and that most of the predicted class was correctly predicted. Other ensemble classifier algorithms such as majority boosting, boosting, and bagging were also perform well, with an average metric around 95%.

Table 7. Categorical Feature to Numerical Mapping

No	Category	Training Set	Testing Set
1	Benign	80020	19980
2	FTP-Brute Force	80104	19896
3	SSH-Brute Force	80127	19873
4	DdoS Attack-HOIC	79763	20237
5	Bot	80120	19880
6	DoS Attack -GoldenEye	80033	19967
7	DoS Attack-Slowloris	80004	19996
8	DdoS attack LOIC-UDP	79934	20066
9	Bruteforce - Web	80034	19966
10	Bruteforce - XSS	79835	20165
11	SQL Injection	80026	19974

Table 8. The Ensemble Classifier Metric's Result Chi Square Technique

Algorithm	Accuracy	Precision	Recall	F1
Majority Voting	97.17%	94.76%	94.94%	94.84%
Boosting	93.33%	95.67%	92.23%	94.49%
Bagging	94.60%	94.80%	94.63%	94.7%
Stacking	94.87%	99.94%	98.82%	99.37%

The area under curve test conducted to determine stacking algorithm's capability to separate each label correctly. Figure 1 shows area under curve test result of stacking algorithm.

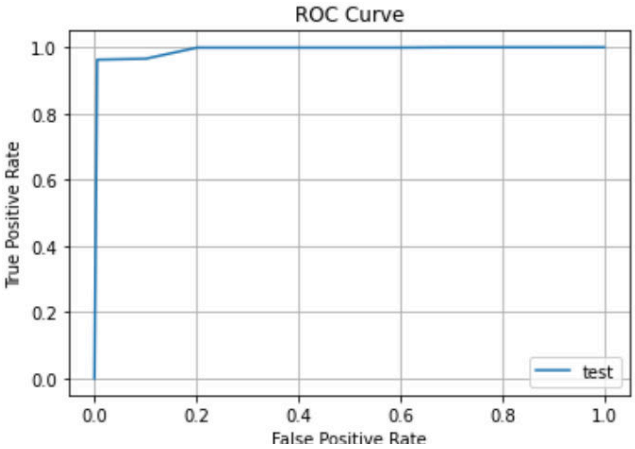

Figure 1. Area Under Curve Test Result

DOI : http://dx.doi.org/10.31963/intek.v9i2.3350

Figure 1 shows that Area Under Curve result reaching ~1 or 100%. This shows that model capability to separate each label is good.

Data retention test was conducted by sending sniffed data from application sniffer. The data sent to a message broker (Apache Kafka) and consumed by the web server. Then, the data received by the web server. The following is total data being sent from application sniffer:

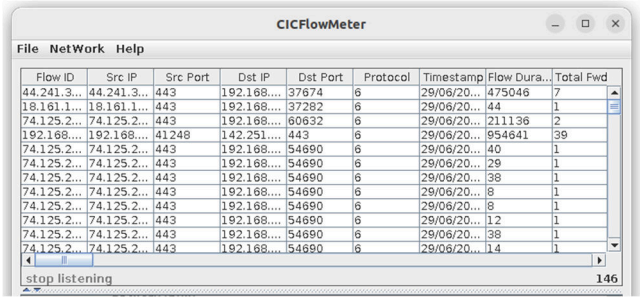

Figure 2. Total Data Sent by Application Sniffer

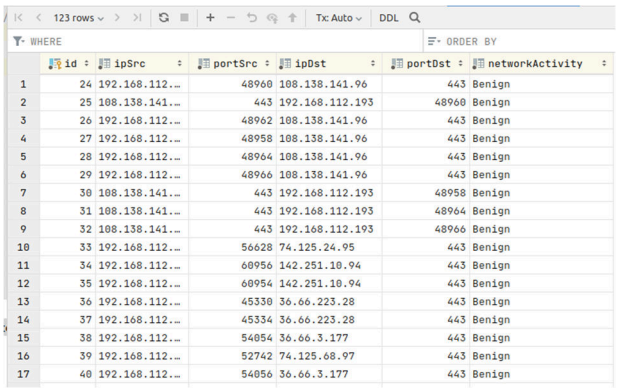

Figure 3. Total Data Saved in Database

Figure 2 shows that there are 146 data are being publish to Apache Kafka. From the Apache Kafka, the data were consumed by the web service and saved into its database. This is the total data successfully saved into the database. Figure 3 shows that 123 data rows being saved by database. Thus, the retention rate calculated with (5) formula is:

$$dataRetention = \frac{123}{146} * 100\% = 84.24\%$$

The application also had feature to show the approximate location of the source and destination IP address as shown at figure 4. Figure 4 shows that the destination IP was from Surabaya Indonesia and Destination IP was from Maine, United States of America.


Figure 4. The Approximate Location of the Source and Destination IP Address

Stacking algorithm was also tested against other dataset which usually used to train model related to cyber-attack. The study uses NSL-KDD dataset to verify the robustness of the proposed study. Table 9 shows the training and testing set used to train and test the stacking model.

Table 9. Training and Testing Set (NSL KDD)

Name	Total
Training Set	13600
Testing Set	3400

Before the training set used to train the model, it will process through chi-square technique and its value would be normalized using tanH operator. Table 10 shows the performance result of stacking algorithm trained with NSL KDD dataset.

Table 10. The Performance Result of Stacking Algorithm Trained with NSL-KDD Dataset

Algorithm	Accuracy	Precision	Recall	F1
Stacking	99.2%	97.61%	98.59%	98.09%

Table 10 shows that although stacking algorithm trained with other dataset, it retained its robustness by 99.2%, 97,61%, 98.59%, and 98.08% in accuracy, precision, recall, and F1, respectively.

IV. Conclusion

1. Feature selection technique such as chi-square and spearman can be used to reduce dataset dimension.

DOI : http://dx.doi.org/10.31963/intek.v9i2.3350

2. The process of building machine learning model comprised in several steps, such as data normalization using tanH operator, and train, of each of single classifier was using sklearn library. From the training, the decision tree algorithm reached the highest metric score with 100% and 99.61% average using chi-square and spearman technique respectively.

3. The performance of an ensemble classifier is reliant on the performance of its individual classifiers. For instance, the stacking algorithm's performance may be inferior to that of a decision tree due to the impact of other individual classifiers on the prediction process.

4. The quantity comparison between two study showed that stacking algorithm was outperforming the other studies by 2 – 10%.

5. The area under curve test showed that stacking algorithm can separate each of label class accurately with the 99.16% score.

References

[1] *D. Barrett, R. Silverman and R. Byrnes, SSH, the secure shell. Sebastopol, CA: O'Reilly Media, Inc., 2011.*

[2] *P. Hope and B. Walther, Web Security Testing Cookbook: Systematic Techniques to Find Problems Fast, 1st ed. Sebastopol, CA: O'Reilly Media, 2008.*

[3] *Incident Response Analyst Report", media.kasperskycontenthub.com, 2022. [Online]. Available: https://media.kasperskycontenthub.com/wp-content/uploads/sites/43/2021/09/13085018/Incident-Response-Analyst-Report-eng-2021.pdf. [Accessed: 11- Jun- 2022].*

[4] *J. Luxemburk, K. Hynek and T. Čejka," Detection of HTTPS Brute-Force Attacks with Packet-Level Feature Set," 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC), 2021, pp. 0114-0122, doi: 10.1109/CCWC51732.2021.9375998.*

[5] *S. Wanjau, G. Wambugu and G. Kamau," SSH-Brute Force Attack Detection Model based on Deep Learning", International Journal of Computer Applications Technology and Research, vol. 10, no. 01, pp. 42-50, 2021. Available: 10.7753/ijcatr1001.1008 [Accessed 11 June 2022].*

[6] *M. M. Najafabadi, T. M. Khoshgoftaar, C. Kemp, N. Seliya and R. Zuech," Machine Learning for Detecting Brute Force Attacks at the Network Level," 2014 IEEE International Conference on Bioinformatics and Bioengineering, 2014, pp. 379-385, doi: 10.1109/BIBE.2014.73.*

[7] *" IDS 2018 — Datasets — Research — Canadian Institute for Cybersecurity — UNB", Unb.ca, 2022. [Online]. Available: https://www.unb.ca/cic/datasets/ids-2018.html. [Accessed: 17 May- 2022].*

[8] *A. Ramezankhani, O. Pournik, J. Shahrabi, F. Azizi, F. Hadaegh and D. Khalili," The Impact of Oversampling with SMOTE on the Performance of 3 Classifiers in Prediction of Type 2 Diabetes", Medical Decision Making, vol. 36, no. 1, pp. 137-144, 2014. Available: 10.1177/0272989x14560647.*

[9] *M. D. Hossain, H. Ochiai, F. Doudou and Y. Kadobayashi," SSH and FTP brute-force Attacks Detection in Computer Networks: LSTM and Machine Learning Approaches," 2020 5th International Conference on Computer and Communication Systems (ICCCS), 2020, pp. 491-497, doi: 10.1109/ICCCS49078.2020.9118459.*

[10] *N. Chawla, K. Bowyer, L. Hall and W. Kegelmeyer," SMOTE: Synthetic Minority Over-sampling Technique", Journal of Artificial Intelligence Research, vol. 16, pp. 321-357, 2002. Available: 10.1613/jair.953.*

[11] *A. Jain, K. Nandakumar and A. Ross," Score normalization in multimodal biometric systems", Pattern Recognition, vol. 38, no. 12, pp. 2270-2285, 2005. Available: 10.1016/j.patcog.2005.01.012.*

[12] *Q. R. S. Fitni and K. Ramli," Implementation of Ensemble Learning and Feature Selection for Performance Improvements in Anomaly-Based Intrusion Detection Systems," 2020 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT), 2020, pp. 118-124, doi: 10.1109/IAICT50021.2020.9172014.*

DOI : http://dx.doi.org/10.31963/intek.v9i2.3350