Performance of Machine Learning Using Preprocessing and Classification for Intrusion Detection System

Thazin Tun
Cisco Lab, UCSY
University of Computer Studies, Yangon
Yangon, Myanmar
thazintun@ucsy.edu.mm

Khaing Khaing Wai
Cisco Lab, UCSY
University of Computer Studies, Yangon
Yangon, Myanmar
khaingkhaingwai@ucsy.edu.mm

Myint Soe Khaing
Cisco Lab, UCSY
University of Computer Studies, Yangon
Yangon, Myanmar
myintsoekhaing@ucsy.edu.mm

***Abstract*— Extracting valuable information from data is a challenging task. Many times, because of massive, redundant, incorrect and noisy outcomes, an analyst will end up with an incorrect classifier. It can also be due to misinterpreting the results and using incorrect procedures for a particular scenario. In our research, we discussed Naïve Bayes, one of the principal approaches to data mining. We did a comparative study of this approach as well. WEKA, which is open source software, was the instrument we used. The datasets used in the NSL-KDD dataset are KDDTrain+.arff and the datasets used in the DARPA dataset are DARPAWeek3-1.arff. ARFF is an abbreviation for Attribute Relation File Format, the simple dataset format that has been adopted by WEKA. The training set sample size is the amount of attributes that are present in the dataset and the amount of documents. Classification models shall be evaluated on the basis of the amount of class labels included in the dataset, the accuracy, the quantity and duration of the legislation created, the error rate and the standing status of the classification. The results show that, based on the amount of experiments we have performed, Naïve Bayes provides better accuracy with discretization than without discretization.**

Keywords—Intrusion Detection System (IDS), Machine Learning (ML), Naïve Bayes (NB), NSL-KDD Dataset, DARPA Dataset

I. INTRODUCTION

It is almost difficult to make precious decisions by manually reviewing current databases and archives that have enormous data. Data mining has made it possible to efficiently process data from many different components. It is easier to categorize the data now. A significant feature of data mining techniques is the use of machine learning approaches in decision support systems and the automation of certain tasks, such as classification , regression, etc.

Data mining is defined as discovering hidden patterns / data from wide raw data sets. To obtain these models and patterns which are present in the data, we use mathematical analysis. It has become increasingly important to obtain useful data knowledge and a deep insight into the data because of the large size of the transaction system database and many other out-of-home outlets. Data mining has made dealing with data much easier than it was a couple of years ago. Different types of information can now be classified, the data summarized and the relationship between different data determined due to the amount of techniques.

But if the content is obsolete, boring, noisy and inconsistent, then there is no question that a very challenging job is the training process. If the training process isn't sufficient, the volume of data will be misclassified. False positive and negative results are generated by misinterpreting the data, which would eventually lead us to incorrect conclusions. One of the most common data mining assignments is classification. We have therefore attempted to equate Naïve Bayes with discretization in this paper and Naïve Bayes without discretization technique.

The tool we use is WEKA, an open source project that is very useful for experiments in data mining and machine learning. WEKA supports various approaches, such as data mining, pre-processing, clustering, sorting, visualization and feature selection, extraction, etc. The data supplied must be available in a single ARFF file and should have attributes that are somehow related. There may be .arff/.csv extensions to the file. Classification models shall be assessed on the basis of the amount of class labels included in the dataset, the precision, the quantity and length of the produced rules, the rate of error and the standard deviation.

II. RELATED WORK

Deep learning algorithms are currently being researched for artificial neural networks and decision tree classifiers. Abdullah et al [1] play out a relative report between various openly accessible information mining and information revelation approaches and programming bundles. In that paper, WEKA, Green, Tanagra, and KNIME were used to see the classification effectiveness of different datasets, including the Car Evaluation dataset. So, the WEKA toolkit is better than the Orange, Tanagra, and KNIME tools, if we compare them. They suggested a method of discretization based on the cluster group in conjunction with the characteristics of rough sets and back-propagation of the artificial neural network. They concluded that the system used as a back-propagation network post-information recognition device is more fault-tolerant and much more capable of interference.

Another research using multiple rule-based classifiers to construct predictive models for the identification of dengue epidemic Decision Tree, Rough Set Classifier, Naive Bayes, and Associative Classifier is an example of rule-based classifiers. The findings concluded that in terms of classification accuracy and ROC (receiver operating characteristic) importance, the multiple classifiers were comparable to single classifiers[2]. Better prediction outcomes can be achieved compared to nonlinear regression by using the neural network model [3].

Each instance of a dataset is taken by a classifier function and mapped by prediction to a distinct class. A binary classifier assigns the network events to either a normal event class or a malicious event class in the case of intrusion detection, while a multiclass classifier further assigns DoS, probe, U2R or R2L classes to the malicious event class. The forms are characterized as follows—

- Denial of Service (DoS): Blocks/restricts computer networks and systems.
- Probe: Attacker probes for vulnerabilities in a network. These can lead to attacks later
- Remote to Local (R2L): Intruder has remote unauthorized access to a system
- User to Root (U2R): Intruder who has user access later tries to access admin or root privilege

Similar to many other data mining techniques, designing the optimum classifiers includes two important tasks: choosing the input attribute from a theoretically large collection of possible attributes in a given dataset and selecting the model based on the chosen one. It is difficult to choose the right attributes, but for the sake of efficient processing speed, it must be done to minimize amount of attributes and remove outdated, redundant and noisy data for the sake of predictive accuracy.. A multi-class classifier G needs to map the feature space with A attributes to C classes on a dataset D consisting of instances of {E1, E2,..., Ei,..., Et}.

III. PROPOSED MACHINE LEARNING

The proposed machine learning uses the NSL-KDD dataset for intrusion detection through supervised training methods to construct the classification models. It is composed of the steps shown in Fig 1.

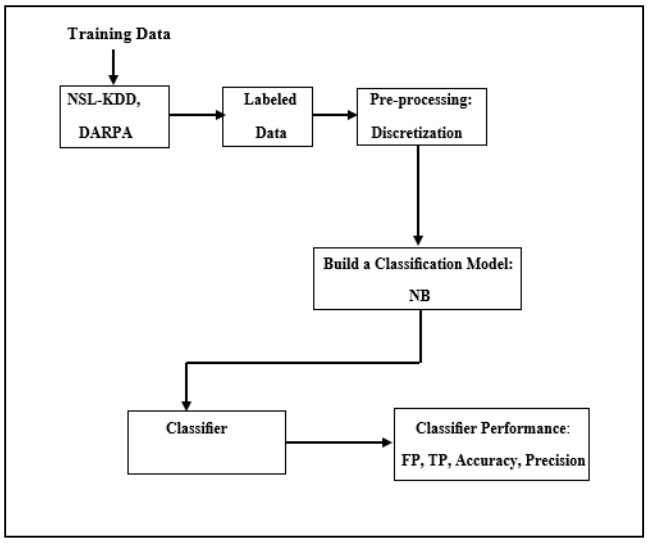

Fig. 1. Proposed Model

A. Naive Bayes

The basic Bayes' Theorem is the basis of this classifier. On classification tasks, it can achieve reasonably good efficiency. The Naive Bayes classifier greatly simplifies learning by assuming, given the class variable, that the characteristics are independent. More formally, the following equation describes this classifier [4].

Where a function vector is denoted by X =), (and Cj, j = 1, 2, ..., N denotes potential class labels. Calculating conditional probabilities P (XJ Ci) and previous probabilities consists of the training method for classifier learning. Here, the training examples which fall into the ci class are computed by counting and then splitting the resulting count by the size of the training set. Similarly, conditional probabilities are determined by simply examining the frequency distribution of function XJ within the training subset that is known as class ci. To define a class-unknown test vector, the posterior probability of each class is calculated, provided the characteristic values present in the test vector are present, and the test vector is assigned to the highest probability class.

B. Discretization Method

The method of transforming a feature's constantly domain into a nominal domain with a finite amount of values is discretization. For certain classifiers, front-end discretization may be appropriate if their algorithms cannot handle constantly characteristics by design. In addition, earlier studies showed that discretization improves the accuracy of classifiers, especially in larger datasets, including Naïve Bayes classifiers [5]. In the last two decades, numerous studies have explored discretization methods to decide how to group constantly values, how to place cut points on the constantly scale, and how many intervals should be used to produce data sets [6] [7] [5] [8].

IV. NSL-KDD DATASET

In the efficiency of the system for intrusion detection, proper data set collection plays a major role. The widely used IDS training and research dataset is KDD-Cup. There are 41 characteristics that are divided into basic characteristics, material and traffic. Based on data from DARPA'98, the KDD-Cup was constructed. An enhanced version of the KDD-Cup dataset that does not suffer from KDD-Cup deficiencies is NSL-KDD. Special characteristics for which NSL-KDD has been chosen over KDD-Cup[9] are given below.

- It does not include duplicate train documents, so the classifiers are not skewed against more frequent documents..
- In the proposed test sets, there are no duplicate documents; thus, learner output is not skewed by way of the techniques that have better detection rates on frequent documents.
- Amount of documents selected from each category of difficulty levels is reversely proportional to the percentage of documents in the original KDD data collection. As a consequence, the classification rates of various methods of machine learning vary in a wider range, making it more effective to provide a precise assessment of different methods for learning.
- Amount of documents in the train and test sets is fair, making it economical to run the experiments on the entire set without the need to pick a small portion randomly. Consequently, the findings of the review of various research papers would be coherent and comparable.

In the dataset, there are about 490,000 single link documents without redundancy. There are 41 attributes and one class attribute in each relational record. Class attributes link labels with exactly one type of specific attack as a standard or attack. But for analysis research, Train+ is used from the NSL-KDD dataset, which is about 125973 documents with class attributes as a normal or a special form of attack. The form of the attack includes the following attacks: DOS, Investigate, R2L and U2R [10].

V. DARPA DATASET

In terms of assessing the efficiency of intrusion detection systems, the MIT Lincoln Laboratory IDS assessment approach is a realistic solution that has made a significant contribution to scientific progress in this area. Many have criticized and found the DARPA IDS assessment dataset as a somewhat obsolete dataset, unable to handle the current trend in attacks. The MIT-DARPA dataset was used to train and test Intrusion Detection Systems' performance. The network traffic was documented in tcp dump format and given for review, including the entire payload of each packet. Data for weeks one and three were used for the training of the PHAD and ALAD anomaly detectors and as test data for weeks four and five. There were 190 attributes of 57 attacks that included 37 Probes, 63 DoS attacks, 53 R2L attacks, 37 U2R/Data attacks in the DARPA 1999 test data. For analysis research, DARPAWeek3-1 is used from the DARPA dataset, which is about 500,000 documents and 23 attributes.

VI. RESEARCH METHOD

The first thing we did was data pre-processing to boost the accuracy of the dataset. In data preprocessing, many processes are involved, such as integration, collection, washing, reduction, and transformation. There are no missing values in any of the data sets, but by using those intervals, we have completed the data preprocessing. We have used four different types of datasets in this analytical analysis, which are KDDTrain+.arff. Multivariates are the properties of the data collection. Categorical, integer and true are the attribute characteristics. The survey the framework used in this informative assessment, step by step, is listed below:

A. Data Pre-processing

In order to improve the accuracy of a dataset, data pre-processing is essentially carried out so that we can obtain clean data that can actually be useful for modeling [11]. For data preprocessing, there are a variety of techniques,

- Data cleaning
- Data integration
- Data transformation
- Data reduction.

The gold of the information preprocessing step is to make information reasonable for the motivations behind exploration. It additionally improves the nature of information and better matches a particular information mining procedure or device.

B. Data Cleaning

Different types of noises and irregularities present in the data are removed at this point, thereby further enhancing the quality of the data. Basically, by correcting irregularities, filling out the missing values, identifying or removing outliers, and softening noise, etc., data cleaning is achieved by cleaning data. We also refined the results by using certain intervals. During data preprocessing, data documents that are inconsistent and do not contribute to the prediction results are removed and WEKA has removed instances that had missing values with the data cleaning tool [12].

C. Data Reduction

The initial data is typically really large and huge. The method of data reduction is the step taken to decrease the amount of initial data without interfering with the initial data 's integrity. Therefore, a reduced set of data is given. This is accompanied by the process of discretization that splits a constantly attribute into a set of intervals of attributes and thus actually reduces the amount of values. The goal of this step is to provide a simple data processing technique.

D. Data Transformation

This processes the data into a form more suitable for data mining.

E. Classification Model

We used a single classifier from Naïve Bayes. The tool used is WEKA for analysis. To test the accuracy of the dataset, we made changes to parameters such as the confidence factor, but our results were not affected.

F. Testing

The split factor we used for training effects in our experiment. The performance assessment of classification models is performed using the parameters below:

- Accuracy
- Rules
- Standard deviation

VII. EXPERIMENTS ANALYSIS AND RESULTS

The next step is the implementation of the testing stage process after the creation of the training models. In a classification algorithm, there are many evaluation metrics that can be used. The confusion matrices for each machine learning classifier were created in this paper. This includes critical data on current and expected performance groups. The output metrics below are also measured:

True negative (TN): Valid documents are correctly known as usual documents.

False negative (FN): The percentage of attacks from incorrect documents as a daily record.

True Positive (TP): This value is used to correctly identify attack packets as attacks:

$$TP = TP / TP + FN \quad (1)$$

False Positive (FP): This value is an incorrect classification judgment where, by increasing the FP value, the normal packet marked as attack increases the computation time; on the other hand, increasing the FN value is assumed to be less than detrimental.

$$FP = FP / TN + FN \quad (2)$$

Precision: One of the main indicators for performance. That is the total amount of documents correctly identified as threats, broken down by the complete amount of documents identified as threats. The precision can be dictated by the condition below:

$$P = TP / TP + FP \quad (3)$$

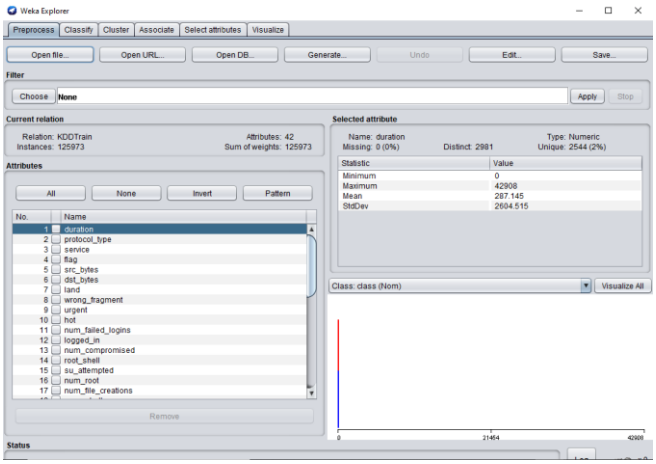

Fig. 2. Preprocessing of KDDTrain+ data set

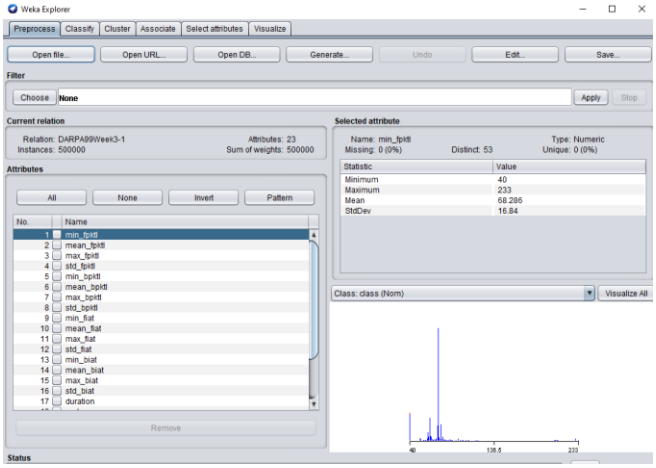

Fig. 3. Preprocessing of DARPAWeek3-1 data set

Figure 2 and 3 shows that the data preprocessing of KDDTrain+.arff and DARPAWeek3-1.arff with WEKA tools.

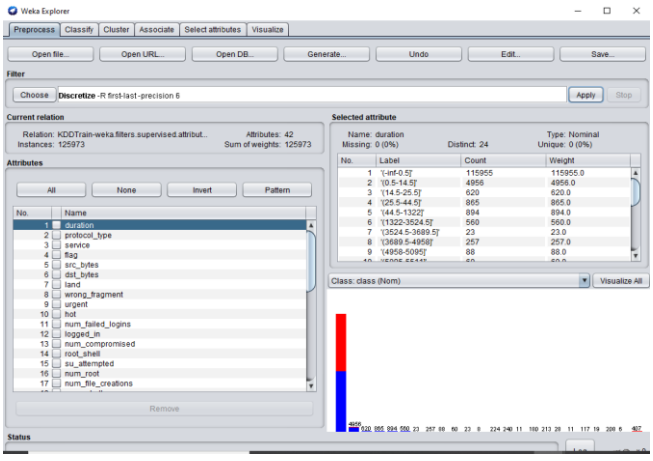

Fig. 4. Discretization of KDDTrain+ data set

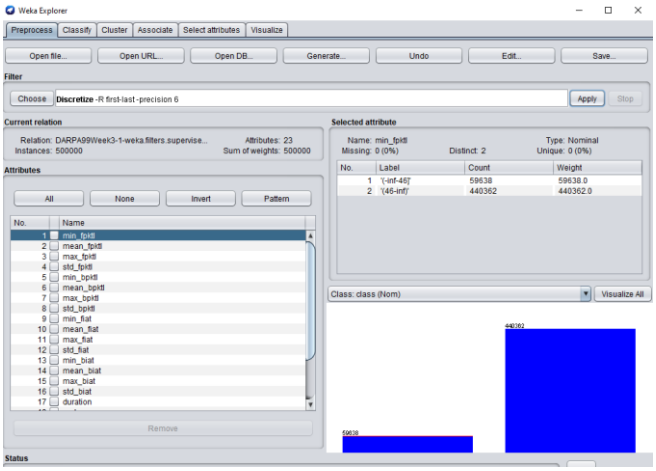

Fig. 5. Discretization of DARPAWeek3-1 data set

Figure 4 and 5 shows that Discretization of KDDTrain+.arff and DARPAWeek3-1.arff with WEKA tools.

```
Time taken to build model: 0.04 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      122353               97.1264 %
Incorrectly Classified Instances      3620                2.8736 %
Kappa statistic                          0.9421
Mean absolute error                      0.0318
Root mean squared error                  0.1612
Relative absolute error                  6.3994 %
Root relative squared error             32.3153 %
Total Number of Instances           125973
```

Fig. 6. Naïve Bayes on KDDTrain+.arff with discretization

```
Time taken to build model: 0.91 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      113858               90.3829 %
Incorrectly Classified Instances     12115                9.6171 %
Kappa statistic                          0.806
Mean absolute error                      0.0965
Root mean squared error                  0.3058
Relative absolute error                 19.3947 %
Root relative squared error             61.3067 %
Total Number of Instances           125973
```

Fig. 7. Naïve Bayes on KDDTrain+.arff without discretization

We showed the outcome of the KDDTrain+.arff dataset with discretization and without discretization in Figure 6 and Figure 7. The findings indicate that, with discretization, Naïve Bayes is stronger than Naïve Bayes without discretization.

```
Time taken to build model: 0.18 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      499149               99.8298 %
Incorrectly Classified Instances       851                0.1702 %
Kappa statistic                          0.8591
Mean absolute error                      0.0018
Root mean squared error                  0.0384
Relative absolute error                 17.0805 %
Root relative squared error             53.2114 %
Total Number of Instances           500000
```

Fig. 8. Naïve Bayes on DARPAWeek3-1.arff with discretization

```
Time taken to build model: 2.56 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      496866               99.3732 %
Incorrectly Classified Instances      3134                0.6268 %
Kappa statistic                          0.6225
Mean absolute error                      0.0063
Root mean squared error                  0.0792
Relative absolute error                 60.2502 %
Root relative squared error            109.7631 %
Total Number of Instances           500000
```

Fig. 9. Naïve Bayes on DARPAWeek3-1.arff without discretization

We showed the outcome of the DARPAWeek3-1.arff dataset with discretization and without discretization in Figure 8 and Figure 9. The findings indicate that, with discretization, Naïve Bayes is stronger than Naïve Bayes without discretization.

Using the Weka tool, the suggested model intrusion detection system approach is implemented to compare with existing distinct classification algorithms. To test the proposed model, different dataset were used. The output of the proposed model is shown in Table 1. It analyzed that the correct instance classification is 122353 and 3620 instances are misclassifications out of 125973 instances for KDDTrain+ dataset. In addition, it analyzed that the correct instance classification is 499149 and 551 instances are misclassifications out of 500000 instances for DARPAWeek3-1 dataset.

TABLE I. RESULTS REVIEW OF PROPOSED MODEL

Performance	Proposed model with KDDTrain+ dataset	Proposed model with ARPAWeek3-1 dataset
Time	0.04 seconds	0.18 seconds
Correctly Classified Instances	122353	499149
Incorrectly Classified Instances	3620	551
Total Amount of Instances	125973	500000

Compared to the proposed model, Table 2 shows the performance of various dataset with proposed model. It is noted that the proposed model outperformed better than without the discretization algorithms.

TABLE II. EFFECT OF PROPOSED MODEL WITH VARIOUS DATASET

Classifier	Dataset	FP	TP	Accuracy	Precision
Naïve Bayes	KDDTrain+	0.032	0.971	97.1264	0.972
Naïve Bayes	DARPA Week3-1	0.001	0.998	99.8298	0.999

VIII. CONCLUSION

The efficiency of the classifiers for machine learning in this paper: Naive Bayes with discretization and without discretization. The basis of all the experiments were the NSL-KDD and DARPA datasets. 125973 and 500000 instances of documents were extracted in the experiments as training data for the selected machine learning classifiers to construct training models. The FP, TP and Accuracy have been increased and the design time has been decreased. In order to validate the proposed model, different performance metrics were used. 97.13 percent is the accuracy of the proposed model for NSL-KDD dataset and 99.83 percent is the accuracy of the proposed model for the DARPA dataset. A comparative study of outcomes is provided between the proposed model and various current models. After evaluating the Naïve Bayes algorithms for both NSL-KDD and DARPA datasets, we can conclude that better results are always given with discretization on Naïve Bayes than without discretization on Naïve Bayes. We may also compare the outcomes depending on the scale of the dataset for future work. Similarly, other variables not mentioned here could be the basis for the study of other researchers.

REFERENCES

[1] Ioannis, "A Comparable Study employing WEKA Clustering/Classification Algorithms for Web Page Classification," Dpt of Inf. And Comm. Systems Engineering, 2011, pp.235-239.

[2] Bakar, Z. Kefli et al. "Predictive Models For Dengue Outbreak Using Multiple Rulebase Classifiers," Electrical Engineering and Informatics (ICEEI), International Conference on 17-19 July 2011, pp. 1-6.

[3] Nor Azura Husin and Naornie Salim. "A Comparative Study For Back Propagation Neural Network And Nonlinear Regression Models For Predicting Dengue Outbreak," Jurnal Teknologi Maklumat Bil 4, Dec 2008. Pp. 97-112.

[4] Witten, I.H., and Frank, E. Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (The Morgan Kaufmann Series in Data Management Systems). 2005.

[5] Liu, H., Hussain, F., Tan, C. L., & Dash, M. (2002). Discretization: An enabling technique. Data Mining and Knowledge Discovery, 6(4), 393–423. http:// dx.doi.org/10.1023/a:1016304305535.

[6] R., J. Dougherty, & M. Sahami. (In the year 1995). Discretization of constantly, supervised and unsupervised functions. In Proceedings of the 12th International Conference (pp. 194-202) Machine Learning.

[7] Fayyad, U., & Irani, K. (In the 1993 year). In constantly-valued attribute classification learning, multi-interval discretization. In Proceedings of the 13th International Artificial Intelligence Joint Conference (pp. 1022-1029).

[8] Yang, Y., & Webb, G.I. and G.I. G.I., and G.I., and the year 2002. For Naïve-Bayes classifiers, a comparative study of discretization processes. In PKAW 2002 Proceedings, The Pacific Rim Knowledge Acquisition Workshop 2002 (pp. 159-173).

[9] K. C. Ting and Khor Somnuk. "A Feature Selection Approach to Network Intrusion Detection System." IEEE, 2009.

[10] "Determining the DOS attack feature collection, Ms Pooja Bhoria, Dr. Kanwal Garg, IJARCSSE, Volume 3, Issue 5, 2013".

[11] Tarmizi, NoorDianaAhmadetal, "DataPreprocessing:CaseStu dyon Dengue Dataset," Universiti Kebangsaan Malaysia, 2011.

[12] M. G. Al-Naymat, Alkasassbeh, A. B. Hassanat, and M. Almseidin, International Journal of Advanced Computer Science & Applications, vol., "Detecting distributed denial of service attacks using data mining techniques," 1, no. 7, pp. 445-436.