ISSN 2347 - 3983
Volume 8. No. 1.1, 2020
International Journal of Emerging Trends in Engineering Research
Available Online at http://www.warse.org/IJETER/static/pdf/file/ijeter3381.12020.pdf
https://doi.org/10.30534/ijeter/2020/3381.12020

A Smart Network Intrusion Detection System based on Network Data Analyzer and Support Vector Machine

Osanaiye S. Babatunde[1], Abdul Rahim Ahmad[2], Salama A. Mostafa[3], Cik Feresa Mohd Foozy[3], Bashar Ahmed Khalaf[4], Ali Hussein Fadel[5] and Palaniappan Shamala[6]

[1]Computer Science Department, Federal University Lokoja, Nigeria, tunsphen@yahoo.com
[2]College of Computer Science and Informatics, Universiti Tenaga Nasional, Selangor, 43009, Malaysia, abdrahim@uniten.edu.my
[3]Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, Johor, 86400, Malaysia, salama@uthm.edu.my, feresa@uthm.edu.my
[4]College of Basic Education, University of Diyala, 32001, Diyala, Iraq, basharalzubaidy60@gmail.com
[5]Department of Computer Science, University of Diyala, 32001, Diyala, Iraq, Alifoxpro2012@gmail.com
[6]Faculty Science Computer and Mathematics, Universiti Teknologi MARA (UiTM), Segamat, Johor, Malaysia, shamalap@uitm.edu.my

ABSTRACT
Because of the critical interest for viable IDS in networks security, the researchers are trying to recognize enhanced methods. This work shows how the KDD dataset is exceptionally helpful for testing distinctive DDoS classifiers. Conclusively, there are two principal ways to reduce the classification complexity and improve the DDoS attack detection accuracy by using nonlinear Support Vector Machine (SVM)s: (1) reducing the number of support vectors; (2) simplifying the classification process for special kernels. This paper proposes a Smart Intrusion Detection System (SIDS) that integrates a Network Data Analyzer (NDA) and SVM to reduce the computation iterations needed by the SVM by eliminating the presumed attack types before performing the classification process. Reduction in data can also serve as a way to increase speed and reduce time in computations. Also, it enhances performance evaluation as 3 types of attack are easier to evaluate than 4 types especially where the 4th type is dominant in the analyzed datasets (the case of DDoS attack being about 79% of the total dataset). As experimented, the proposed Smart Intrusion Detection System method has shown a way in dataset reduction by simply eliminating the DDOS attack types with the same amount of data as compared to Batch 2. Batch 1 serves as a control experiment as indicated by its good performance evaluation measurements.

Key words: DDoS attack, Intrusion Detection System (IDS), Support Vector Machine (SVM), and Network Data Analyzer (NDA)**.**

1. INTRODUCTION

An Intrusion Detection System (IDS) could be a product/software as well as equipment/hardware that screens organized traffic for intrusion detection, for example, suspicious exercises and cautions the framework or system administrator. Sometimes, the IDS may likewise react to abnormal or malignant traffic by starting a predetermined action, for example, stopping the client or source IP address from gaining access to the network [1]. The general IDS framework is crucial to consider how it connects and associates with its environments/surroundings. The user behaviors, including interlopers, from whom input comes, are considered as part of the external environment. The intrusion detection process begins with figuring out what is to be identified and subsequently results in a choice being made [2].

Furthermore, Fig. 1 shows the general architecture of the IDS. It is consisting of a number of different components with specific goals in collectively detecting and analyzing suspicious traffic. The components are as follows: Sensors, web mining, Buffering and Decision Making. The sensors are employed to monitor the incoming traffics. Whereas, Web Mining is a technique that has been applied to analyze the traffics all the time to set the threshold. Then, based on the analysis of the Web mining technique, the traffics which classified more than the threshold will forward to the decision making methods and the buffering. Also, the traffics which notified less than the threshold will pass directly to the webserver [3].

There are two limits: total security and total access. The nearest to a totally secure machine is one that is unplugged from the network but such an isolated system is useless in this state. Despite what might be expected, a machine with total access is highly usable but at the same time impractical due to network dangers [4]. In this way, every connection or IT infrastructural relationship needs to choose for itself was between the two boundaries of absolute security and all-out access. A strategy needs to verbalize this, and afterwards,

characterize how that will be implemented; and everything that is done for the sake of security, at that point, must implement that arrangement consistently. A risk or threat is an undesirable (purposeful or unintentional) occasion that may result in damage to an asset. In addition, a threat is abusing a known defenselessness (vulnerability) or prior observable fault/weakness [5]. However, Figure 1 shows the general architecture of the IDS.

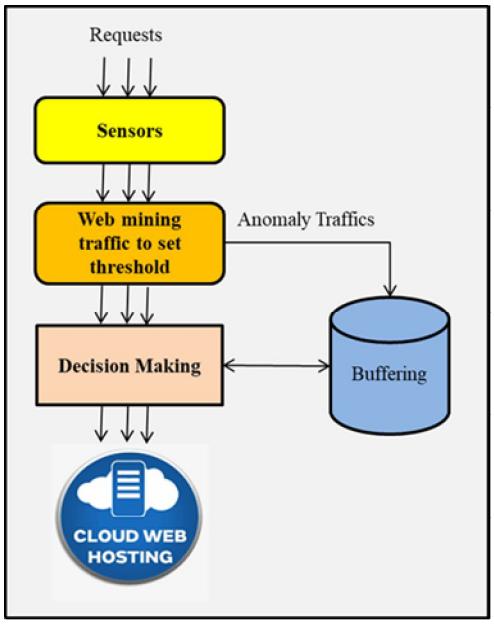

Figure 1: General IDS Architecture [3]

SIDS scours and screens organize the traffic for suspicious activities and caution the framework or system overseer. Now and again, the SIDS may likewise react to peculiar or malicious traffic by taking an action, for example, denying the client or source IP address access to the system. When somebody finds new computer security weakness or vulnerability, a horde of hackers begins thumping at the entry points of networks worldwide to check whether they can undermine the protections in place [6]. Numerous locales utilize a mix of firewalls on border routers and host-based packet filters and wrappers to secure themselves. However, imagine a scenario in which the weakness is in the specific instrument that is utilized to protect an administration. In what capacity can network administrator realize that their machines are enduring an onslaught as well as have been endangered? The most ideal approach to stop the menace is to utilize IDS [7].

The aim of this study is to meet the challenges of effective classification. Consequently, the system is designed to defend against DDoS attack. The NDA and SVM are included in the SIDS to classify the DDoS in such a way that increases the accuracy rate and reduces false alarm rate. The efficiency of this model is tested and measured using Knowledge Discovery and Data Mining (KDD) Cup 99 dataset, which was prepared to test and evaluate the performance of the proposed defence systems.

The following section presents the related work of using different types of artificial intelligence techniques for the SIDS development. Section 3 describes the testing dataset and its general properties. Section 4 presents the design and implementation of the SIDS based on SVM. Section 5 presents the results and the conclusion is in Section 6.

2. RELATED WORK

Many machine learning approaches and Artificial intelligence methods such as heuristic techniques, for example, Genetic Algorithm (GA) and ANN are utilized in IDS picking up its capacity to learning and improvement, which makes them progressively precise and productive in confronting the expanding number of erratic attacks. The GA and ANN joined method to give the IDS additional execution and precision [8]. Pradhan et al. [9], considered the client activities as a parameter in irregularity recognition utilizing a backpropagation in their performance. This work is extremely encouraging. The back-propagation NN had a classification rate of 100%. The identification rate was 88% on attacks when all is said and done whether known or obscure attacks. The principal benefits of this study are the base measure of training data that requires giving great aftereffects of the classification of the traffic.

Lately, an enhancement option of ANN is proposed known as Multi-Layer Perceptron (MLP) ANN. The MLP technique made ANN IDS strategies increasingly exact and productive as far as discovery and ordinary correspondence is concerned. The MLP-ANN strategy indicates discovery results in much superior to customary techniques. MLP conquers the restriction of recognition of low-frequency attacks. What's more, MLP-ANN IDS can characterize the sort of attacks and arrange them. This element enables the framework to predefine activities against comparative future attacks [10]. In the classifiers choice model exhibited by Nguyen and Choi [11], they removed 49,596 occurrences of KDD dataset and considered a lot of classifiers under control condition.

Lahre et al. [12] exhibited distinctive ways to deal with or manage KDD dataset, directed, and unsupervised strategies reenacted utilizing MATLAB, and scientists test regulated and unsupervised methods with fluffy tenets to distinguish the execution of the proposed framework. Breiman [13], concentrated on arbitrary woodland and how is it consolidated between trees indicators and the researcher proposed mistake in the irregular backwoods as a limit number of trees in the timberland. The informational collections (datasets) which are utilized for training and testing the machine learning model give the establishment to the proposed inductive NIDS. To a substantial degree, the

characterization execution of the last theory is affected by the nature of the training sets. These collections of information or datasets are currently analyzed in detail. Their structure is investigated and the planning is looked into once more.

Support Vector Machines (SVM) in most ways have turned out to be a standout amongst the most prominent methods for anomaly intrusion detection since it has great generalization nature and has shown the good performance of its capacity to overcome the scourge or challenge of dimensionality [8]. Although, a few enhancements have been done however the number of measurements (sample size) or the number of dimensions still influences the execution of SVM-based classifiers [3]. Another problem is that each element of information (data feature) is similarly dealt with in SVM i.e. features are treated with the same importance. Prior, Hamed T. et al. [14], utilized a calculation that uses the rough set hypothesis to rank features and ascertain feature weights in improving SVM Learning with Weighted Features.

Additionally, Hu et al. [15] displayed another methodology, in view of robust vector machines (RSVMs) which successfully addresses the problem of over-fitting presented by the noise in the training dataset which settles on the decision surface smoother and consequently controls the measure of regularization. In genuine intrusion detection datasets, numerous elements of information also known as features are repetitive or less critical [16]. Notwithstanding the methodology utilized, most strategies presently being used depend on the assumption that the training models utilized by the intrusion identifier are untainted and trustable, i.e., the labels of the training samples are completely correct. In practice, however, the informational collections acquired from true frameworks (review trails, application logs, network packet sniffing) hardly fulfil this presumption. Above all else, it is not in every case simple to acquire clean information. The training samples might be mislabeled because of the fuzzy limit among typical and odd behaviors.

3. METHODS AND MATERIALS

3.1 Testing Dataset

This area presents the structure of the training dataset utilized for the one-class SVM. Also, the planning of the training dataset is secured since the one-class SVM isn't prepared with the training data in its original frame. The dataset utilized for preparing was taken from the UCI KDD Archive (KDD Cup, 1999) by MIT Lincoln Labs (1998) [3]. This dataset was a DARPA Intrusion Detection Evaluation Program produced in 1998 and overseen by MIT Lincoln Labs for research in intrusion detection thorough review and assessment. The 1999 KDD intrusion detection challenge utilizes a variant of this dataset. As twisted from Lincoln Labs set up a domain to get nine weeks of crude TCP dump information for a Local Area Network (LAN) reenacting a run of the U.S. Airforce LAN. They worked the LAN as though it were a genuine Air Force condition, however, peppered and attacked it with numerous attacks [3].

The set of reviewed/evaluated information of intrusions was recreated in a military system condition. The crude datasets of training data were around 4 gigabytes of compacted double TCP dump information of system traffic from seven weeks of system traffic, processed into around five million association records, and test data of two million association records. An association or a connection is a grouping of TCP packets beginning and ending at characterized times, with information streams to and from a source IP address to an objective IP address under a characterized protocol. Associations were named as either typical or as an attack, with one explicit attack type. The training and test data were utilized to acquire a suitable model for exact intrusion recognition [17], [18]. To the extent the data is concerned, at first, the set was given unlabeled, for example, no class has attributes. Subsequent to the KDD cup, classes were attributed, and the dataset was made open to evaluate the precision of different models. In this paper, about 500K and 312K of the dataset are exclusively used for training and testing respectively [3]. The KDD Cup99 Dataset is used in this work for the following reasons:

- The KDD Cup 1999 dataset is utilized when benchmarking intrusion detection problems. It was created basically for Intrusion Detection by recreating possible attack types any network system can encounter.
- The dataset contains attack types grouped into four categories. Each category has 41 attributes representing different features and is labelled for ease of recognition.
- It is large enough to test for redundancy and can easily be normalized.
- Finally, SVM is a computational Artificial Intelligence technique that works with numeric data. This dataset is more numeric than alpha-numeric and the non-numeric values have alternate numeric representatives such as port numbers i.e. HTTP can be replaced with port number 80.

3.2 THE SIDS MODEL

It is interesting that all attack types have features (see Table 1) used in recognizing their characteristics and hence in their detection [19]. Exploiting this area has allowed certain attack types to be selected and analyzed without having anything to do with the others. In KDD'99 10% corrected Training dataset which is labelled allows another means of detection where the labels are looked out for. These techniques are employed in the SIDS and SVM implementation to sort out the DDoS attack types [3], [20]. The raw data is fed into the system and SIDS sorts out the detected DDoS attack types

(and drops them out of the whole dataset), the rest is transformed into the SVM format, the transformed data is scaled and fed into the SVM for classification.

However, the proposed methods consist of several stapes as shown in Fig. 2. In the first step, the web mining function is employed to monitor and analyze the incoming traffics to set the threshold. Subsequently, the incoming traffics will forward to the threshold to determine whether the incoming traffics are attack traffics or not, in case of the traffics more than the threshold it means that it is attacked traffics and it will send to the buffer to save the IP address which sends the attack traffic for the permanent block. Whereas, in case of the incoming traffics is less than the threshold this means the traffics identified as normal traffic, then it will forward to the buffer to save the IP address and pass to the webserver. Figure 2 Shows the architecture of the SIDS

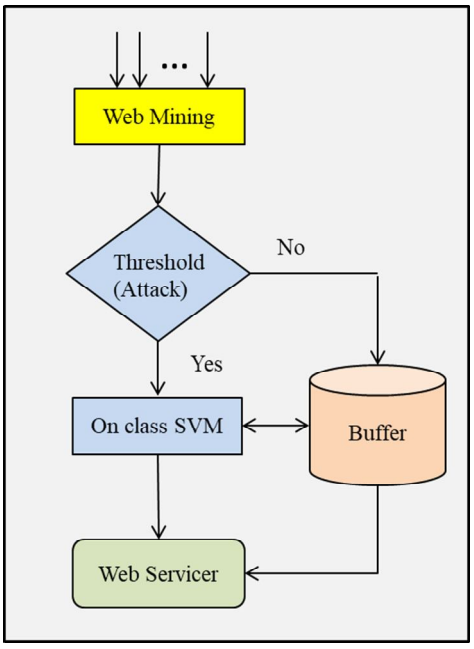

Figure 2: The architecture of the SIDS

In addition, a screen report is generated after the sorting and transformation is complete indicating the type and number of DDoS attack types detected and removed; the total number of the whole dataset before sorting and transformation and finally, the total number of attack types removed. Algorithm 1 presents the operational steps of the DDoS attack identification model.

Algorithm 1: The DDoS attack identification model

step1: prepare Raw KDD 1999 dataset;
step2: sort and Transformation of Dataset to SVM;
step3: perform normalization (scaling) to the dataset;
step4: set the training and testing files;
step5: classify the data by SVM;
step6: verify the accuracy of the results;

The DDoS attack types are removed and as such, the whole dataset is reduced such that the time needed for a total transformation, scaling and classification is significantly reduced since DDoS attack types make about 79% of the whole dataset; leaving the other three types of attack namely R2L, U2R and Probing for classification. However, in the course of this project assorting of these attack types (DDoS) are done using the second technique i.e. using the attack type labels since feature selection is out of the scope of this research and the dataset employed is a labelled one.

Table 1: KDD dataset features [3], [7].

Attack	Record	Class	#	Attack	Record	Class
Land	6	Dos	#	stan	1829	Probe
neptune	20750	Dos	#	spy	1	R2L
smurf	1327	Dos	#	phf	3	R2L
pod	87	Dos	#	imap	6	R2L
back	502	Dos	#	nmap	743	Probe
teardrop	437	Dos	#	ftp	4	R2L

As mentioned earlier, in going through the datasets, all the DOS types of attacks are counted and the number of times each occurred is calculated and at the same time are dropped (as shown in Table 2) from the rest of the dataset. Then, the other types of attacks (U2R, R2L and Probing) are fed into the SVM for classification. At this stage, the occurrence per duration (the frequency of occurrence of such type of attack in a specific time frame or window) is not taken into consideration since the SIDS is still running on offline datasets. It is then implemented in C++ programming language along with the Data Processing Program as one application. Figure 3 shows some of the considered DDoS attack types with dataset affiliation.

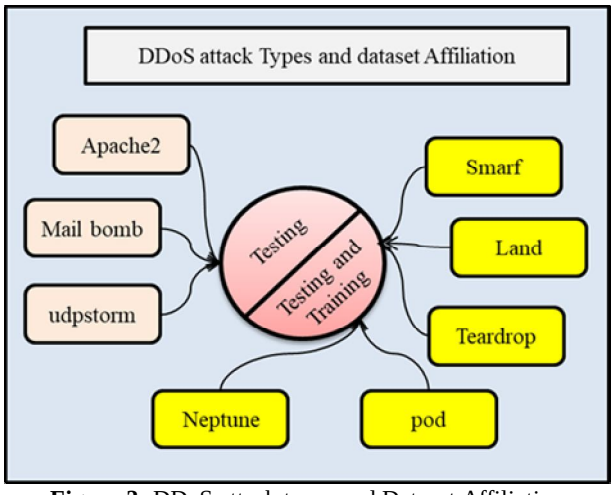

Figure 3: DDoS attack types and Dataset Affiliation

4. THE IMPLEMENTATION OF THE SIDS

In practical application, it scans through the Network payload and analyses the packet headers looking for source and destination IP addresses. It is noteworthy that various

organizations have the different threshold set for their systems in responding to requests i.e. the number of requests that are handled by the system, usually not exceeding the actual request the systems can handle. Requests from all addresses are evaluated looking for the number of times these requests are sent in a particular time frame (determined by the organization's network policy). If these requests are close to the threshold/limit or more than the set threshold, the SIDS produces a report (this is also done in a particular time frame containing the Source and Destination IP addresses, the number of request per time frame including the Normal Organisation's Network Request Threshold or Limit Value) directed at the Network Administrator. It is now left for the Network Administrator to thwart requests from such source if it is deemed malicious. Furthermore, the SIDS can be given the privilege to drop such Network Frames or packets. Limitations can be in the form of an Organization's Policy which usually changes without alterations to the entire network system and at this point, the old policy is still implemented which leaves the whole system vulnerable.

4.1. DATA PRE-PROCESSING

The quality or nature of a model depends to a substantial degree on the nature of the information used to produce (train) it. A huge amount of the time spent in some random information or data mining venture is committed to the preparation of data. The data must be cautiously examined, purged, and changed, and proper algorithm data preparation techniques must be utilized. Data mining can just reveal designs effectively present in the data; the objective dataset must be sufficiently extensive to contain these examples while staying sufficiently succinct to be mined in an adequate time span [18]. A typical hotspot for data is a data mart or warehouse in which The UCI KDD Archive is an instance [21]. The objective set is then cleaned. Cleaning evacuates the perceptions with noise and missing information. The spotless data is reduced into feature vectors, one vector for every perception. A feature or component vector is an abridged variant of the crude data perception. The feature or element vectors are put into two sets, the "training set" and the "test set". The training set is utilized to "train" the data mining algorithm(s), whereas the test set is utilized to check the exactness of any examples discovered. After the datasets have been preprocessed, the following steps are carried out with details. Algorithm 2 presents the main SVM operational steps.

Algorithm 2: The SVM operational steps
step1: apply the RBF kernel;
step2: use cross-validation for selecting best cost, c and gamma, γ parameters;
step3: use C and γ to train the whole training set;
step4: test (predict) using the prepared test set;

4.2. DATA PROCESSING

Preprocessing is a procedure used in converting raw data into machine input. SVM necessitates that every datum occurrence is presented as a vector of a real number [6], [7], [8], [10], [20]. Subsequently, if there are unmitigated qualities, we initially need to change over crude data collection into the numeric dataset. In the data pre-processing we changed over the whole data index into SVMs format; it implies changing an overall characteristic of the record into real numbers. The procedure of data preparation is additionally made complex by the way that any data, to which a model is applied, regardless of whether for testing or scoring, must experience similar processes as the data used to train the model. Below are the steps for transforming the KDD 1999 dataset into an SVM format and further training and testing; SVM classifies only numeric values and every nominal value or character has to be transformed into its corresponding numeric value. Interestingly, KDD '99 dataset contains characters and strings of words such as TCP, ICMP, SMURF, RSTO, etc. as indicated in the table below [21]. During transformation, data types (service, protocol, flag, etc.) are replaced with their corresponding port numbers (see Table 2), normal classes with -1 and all attack type classes with +1.

Table 2: KDD '99 10% corrected dataset sample [21]

0,tcp,http,SF,181,5450,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,8,8,0.0,0.00,0.00,0.00,1.00,0.00,0.00,9,9,1.00,0.00,0.11,0.00,0.0, 0.0, 0, 0.0.0, normal.
0,tcp,http,SF,239,486,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,8,8,0. 00,0.00,0.00,0.00,1.00,0.00,0.00,19,19,1.00,0.00,0.05,0.00.00,0.00,0.00,0.00,normal.
0,tcp,http,SF,235,1337,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,8,8,0.00,0.00,0.00,0.00,1.00,0.00,0.00,29,29,1.00,0.00,0.03,0.00,0.00,0.00,0.00,normal.

In addition, redundant values e.g. zero, separation commas were also eliminated since they give no information and as such an unintentional feature selection was done. The transformed data starts with an index and the value, then a space and the next index till the whole attribute for that particular line has been exhausted (see Table 3 for a sample of KDD '99 dataset and Table 5 for the sample of the transformed dataset).

Table 3: KDD '99 10% corrected dataset sample [21]

-1 0:3 1:80 2:10 3:181 4:5450 10:1 21:8 22:8 27:1 30:1 31:9 32:1 34:0.11
-1 0:3 1:80 2:10 3:239 4:486 10:1 21:8 22:8 27:1 30:19 31:19 32:1 34:0.05
-1 0:3 1:80 2:10 3:235 4:1337 10:1 21:8 22:8 27:1 30:29 31:29 32:1 34:0.03

5. RESULTS AND DESCUSION

From data preprocessing, 3 batches of training sets and a batch of the testing set. In terms of training time, Batches 1 and 2 were identical (4mins 2s and 4min 20s respectively) for the classifier generation by the SVM. This could be attributed to the size of the dataset which is the same. For Batch 3 however, the training time was reduced considerably by more than half of the training time taken by Batches 1 and 2 (1mins 58s). The main critical inadequacy in the KDD informational collection is the gigantic number of excess records. Investigating KDD train and test sets, it is discovered that about 78% and 75% of the records are copied and intrinsic in them respectively. This expansive measure of repetitive records in the train set will make learning calculations one-sided towards the more occurring records, and in this way keep it from learning rare records which are generally more dangerous to systems, for example, U2R attacks. In SVM learning, this is called overfitting of data features. The presence of these rehashed records in the test set, then again, will cause the assessment results to be one-sided by the strategies which have better recognition rates on the more occurring records. Removing the redundancies or duplicates helps set a checkpoint for comparison of the proposed frameworks. Perfect data usually have no pattern during classification because one dataset is different from another and another and this eliminates bias by the methods which have better detection rates on the frequent records.

Implementing the performance evaluation, it is easier to determine how well an algorithm or a framework implements its classification. Microsoft Excel served instrumental in determining the comparison between the test set that is classified and the result of the classification (the output result of SVM). It is noteworthy that the comparison is done using the (+1, -1 or 0, 1) label on the dataset which can be found at the first entity on a scaled dataset (both training and test set) and the SVM output after classification (also in +1, -1 and 0, 1 as the case may be). If (+1, -1) is used for training set and test set during scaling, the output will be a property of (+1 or -1). This is also applicable to datasets that have been scaled (0, 1), the output after classification is a property of (0s and 1s). This is illustrated in Table 6; it shows a scaled dataset in the range of (+1, -1) and the corresponding output after classification in the (+1 and -1) property. The numbers in the red boxes are the Variables to be compared to determine the performance metrics. The steps involved in using the Microsoft Excel package for comparison are as follows:

The features in the boxes are taken out and put on different columns in a new spreadsheet in the Excel program. This is illustrated in Figure 3. In so doing, the primary concerns are on the determinants of an instance of a dataset as an attack (+1) or as benign (-1) and compared with the classified output as intrusive or legitimate (+1 or -1).

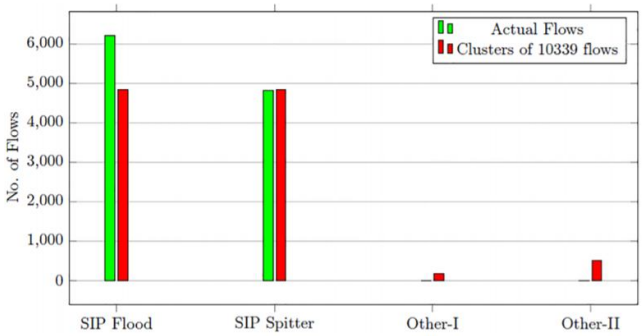

Figure 4: Alignment of instance indicator and corresponding classification result by the SVM

From Figure 4, the data are then sorted accordingly in ascending or descending order. Here, there are 4 combinations of (1 and -1) as shown: (1, 1), (1, -1), (-1, 1) and (-1, -1). These are then related to the performance metrics and the numbers of times these combinations occur are recorded for the performance measurement evaluation. Table 4 shows the relationship between performance metrics and number combinations of instance determinant and classified result.In arranging the combinations in descending order, no matter where a certain combination occurs is placed in the same group as its kind and in turn aids counting. From Figure 3, (-1, -1) occurs thrice, (-1, 1) occurs once, (1, -1) occurs twice and (1, 1) happens 4 times. Therefore:

True Negative, *tn* (-1, -1) = 3;
True Positive, *tp* (1, 1) = 3;
False Negative, *fn* (1, -1) = 2; and
False Positive, *fp* (-1, 1) = 1.

Table 4: The evaluation of the results

Matrix	Explanation and Meaning	Representation
TP	An attack and alarm were raised: +1 indicates that there was an attack and hence an alarm was raised meaning that particular instance was classified correctly as indicated by +1.	(+1, +1) or (1, 1)
TN	No attack and no alarm were raised: the instance is indicated as legitimate (-1) and a (-1) is a correct prediction	(-1, -1)
FP	No attack but the alarm is raised: The instance is labelled (-1) meaning it is a legitimate action but the SVM classified it as intrusive as shown by (+1) in the classified result output.	(-1, +1) or (-1, 1)
FN	An attack occurred but no alarm is raised: This also indicates a false classification where the instance is an intrusive action (+1) but it was misclassified to be benign by the SVM as indicated (-1) in the classified result output.	(+1, -1) or (1, -1)

In the same manner, the procedure is applied to Batch 1, Batch 2 and Batch 3 and the result is presented in Table 7

Table 7: Batches and Performance Metrics

Batch	Tp	TN	FP	FN	Total
1	39.70%	58.43%	1.29%	0.58%	99.42%
2	36.53%	48.60%	8.58%	6.29%	93.71%
3	56.06%	37.73%	3.12%	3.04%	96.91

In this work, False Alarm represents the total number of incorrect alarmed raised i.e. False Alarm Rate = False Positive Rate (simply put as False Alarm as shown in (1)). It is indicated that a wrong prediction or classification of a set is False Alarm and as such; False Alarm Rate = Miss Rate (False Negative Rate) + False Alarm (False Positive Rate).

$$\text{Accuracy} = \frac{fn}{tp+fn} + \frac{fp}{tn+fp} \quad (1)$$

In a simpler term, False Alarm Rate is the same as the Error Rate. In this research, False Alarm is taken to be the total number of the wrong classification by an algorithm of a dataset. False Alarm Rate and Error Rate will be used interchangeably in the course of this analysis. The outcome of the performance measurement using the metrics is detailed in Table 5.

Table 5: The performance measurement of the 3 batches

Batch	**Accuracy**	**FAR**	**FPR**	**FNR**
1	98.13 %	1.87%	2.16 %	1.44%
2	85.13%	14.87%	15.01%	14.69%
3	93.79%	6.21 %	7.64%	5.22%

From the result, it can be deduced that the removal of DDoS attack types by SIDS gave a reduction in time and as such, increased efficiency in classification in terms of classifier or model generation in a way that the final dataset to be classified contains fewer instances. Accuracy as determined from the evaluation as indicated by Table 8 which shows that Batch 1 has a high accuracy of 98.13 % which can be attributed to lack of redundancy and also based on the property that the data was perfect and devoid of duplicates. In reality, network data are not perfect but random. In Batch 2, the accuracy is 85.13% and this is owed to the randomness of the data. Batch 3 is also a random dataset. However, SIDS was applied to treat it of DDoS attack types leaving it with fewer instances in the dataset for classification. The accuracy of 93.79% can imply that intrusions in small volumes of data can easily be detected than when they are in large volumes. This accuracy term denotes that the classifier was able to classify all the instances in the test set (perfect) and on the randomly selected dataset, SVM classified 85.13% correctly leaving 14.87% as misclassified instances. Also in the SIDS prepared datasets, instances of data were accurately classified to the tune of 93.79% leaving 6.21% as the falsely classified dataset instances.

In as much as Batch 1 has better performance measurement compared to the other two, the focus is basically on Batch 2 and 3 whose degree of randomness distinguishes them from Batch 1. From Table 6.4, it can be seen that Batch 2 has a False Positive Rate of 15.01% whereas Batch 3 is about half; 7.64%. The deduction here is that the datasets are random but Batch 2 has 4 attack types with duplicates whereas in Batch 3, only 3 attack types are in view (DDoS has been removed by SIDS) and as DDoS makes about 79% of the data, classification is biased towards a particular type of attack (in Batch 2). This deficiency (Bias: underfitting of other attack types and overfitting of DDoS attack types) is removed by SIDS in Batch 3. This is also applicable to the difference in False Negative Rate (Batch 2, 14.69% and Batch 3, 5.22%) and hence, the False Alarm Rate (Batch 2, 14.87% and Batch 3, 6.21%) as False Alarm Rate is subject to False Positive Rate and False Negative Rate.

5. CONCLUSION

From the result, it can be deduced that the removal of DDoS attack types by proposed method gave a reduction in time and as such, increased efficiency in classification in terms of classifier or model generation in a way that the final dataset to be classified contains fewer instances. Accuracy as determined from the evaluation as indicated by Table 8 which shows that Batch 1 has a high accuracy of 98.13 % which can be attributed to lack of redundancy and also based on the property that the data was perfect and devoid of duplicates. In reality, network data are not perfect but random. In Batch 2, the accuracy is 85.13% and this is owed to the randomness of the data. Batch 3 is also a random dataset. However, SIDS was applied to treat DDoS attack types leaving it with fewer instances in the dataset for classification. The accuracy of 93.79% can imply that intrusions in small volumes of data can easily be detected than when they are in large volumes. This accuracy term denotes that the classifier was able to classify all the instances in the test set (perfect) and on the randomly selected dataset, SVM classified 85.13% correctly leaving 14.87% as misclassified instances. Also in the SIDS prepared datasets, instances of data were accurately classified to the tune of 93.79% leaving 6.21% as the falsely classified dataset instances.

ACKNOWLEDGEMENT

This research is supported by Universiti Tun Hussein Onn Malaysia under Tier 1 Grant Scheme Vot H237 and Vot H101

REFERENCES

1. S. Aljawarneh, M. Aldwairi, & M. B. Yassein. **Anomaly-based intrusion detection system through feature selection analysis and building hybrid**

efficient model. *Journal of Computational Science*, 25, 152-160, 2018.

2. P. R, Varma, V.V. Kumari, and S. S. Kumar. **A Survey of Feature Selection Techniques in Intrusion Detection System: A Soft Computing Perspective.** *In Progress in Computing, Analytics and Networking* (pp. 785-793). Springer, Singapore, 2018.
3. B. A. Khalaf, S. A. Mostafa, A. Mustapha, M. A., Mohammed, & W. M. Abduallah, **Comprehensive review of artificial intelligence and statistical approaches in distributed denial of service attack and defense methods**. *IEEE Access*, 7, 51691-51713, 2019.
4. R. Roman, J. Lopez, and M. Mambo. **Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges.** *Future Generation Computer Systems*, 78, pp.680-698, 2018.
5. B. A. Khalaf, S. A. Mostafa, A. Mustapha, & N. Abdullah. **An adaptive model for detection and prevention of DDoS and flash crowd flooding attacks.** *In 2018 International Symposium on Agent, Multi-Agent Systems and Robotics (ISAMSR)*, (pp. 1-6), Malaysia, 2018.
6. S. A., Mostafa, A., Mustapha, P., Shamala, O. I., Obaid, & B. A. Khalaf. **Social networking mobile apps framework for organizing and facilitating charitable and voluntary activities in Malaysia**. *Bulletin of Electrical Engineering and Informatics*, 9(2), 827-833. 2020.
7. B. A. Khalaf, S. A. Mostafaa, A. Mustapha, , A. Ismaila, , M. A. Mahmoudb, M. A. Jubaira, & M. H. Hassana, A **Simulation Study of Syn Flood Attack in Cloud Computing Environment**. *AUS Journal*, 1-10, 2019.
8. I. Ahmad, M.,Basheri, M. J. Iqbal, & A. Rahim, **Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection**. *IEEE access*, 6, 33789-33795, 2018.
9. M. Pradhan, S. K. Pradhan, & S. K. Sahu. **Anomaly detection using artificial neural network**. *International Journal of Engineering Sciences & Emerging Technologies*, 2(1), 29-36, 2012.
10. A. A. Heidari, H. Faris, I. Aljarah, & S. Mirjalili. **An efficient hybrid multilayer perceptron neural network with grasshopper optimization**. *Soft Computing*, 23(17), 7941-7958, 2019.
11. H. A. Nguyen, & D. Choi. **Application of data mining to network intrusion detection: classifier selection model. In Asia-Pacific Network Operations and Management Symposium** *Springer, Berlin, Heidelberg*, (pp. 399-408), 2008.
12. M. K. Lahre, M. T. Dhar, D. Suresh, K. Kashyap, & P. Agrawal. **Analyze different approaches for IDS using KDD 99 data set.** *International Journal on Recent and Innovation Trends in Computing and Communication*, 1(8), 645-651, 2013.
13. Breiman, Leo. "**Random forests.**" *Machine learning* 45.1 5-32, 2001.
14. T. Hamed, J.B. Ernst, and S.C. Kremer. **A survey and taxonomy of classifiers of intrusion detection systems**. *In Computer and network security essentials Springer, Cham*, pp. 21-39, 2018.
15. W. Hu, Y., Liao & V. R. Vemuri. **Robust anomaly detection using support vector machines.** *In Proceedings of the international conference on machine learning*, pp. 282-289, 2003.
16. K. Selvakumar, M., Karuppiah, L., SaiRamesh, S. H., Islam, M. M., Hassan, G., Fortino, & K. K. R. Choo, **Intelligent temporal classification and fuzzy rough set-based feature selection algorithm for intrusion detection system in WSNs**. *Information Sciences*, 497, 77-90, 2019.
17. R., Bala & R. Nagpal. **A REVIEW ON KDD CUP99 AND NSL-KDD DATASET.** *International Journal of Advanced Research in Computer Science*, 10(2), 64. 2019.
18. H. A., Ismael, J. M., Abbas, S. A., Mostafa, & A. H. Fadel. **An enhanced fireworks algorithm to generate prime key for multiple users in fingerprinting domain**. *Bulletin of Electrical Engineering and Informatics*, 10(1). 2020.
19. A. J. Mohammed, M. H. Arif, A. A. Ali, **A multilayer perceptron artificial neural network approach for improving the accuracy of intrusion detection systems**, *International Journal of Artificial Intelligence*, v. 9, n. 4, 2020.
20. N. M. Sahib, A. H. Fadel, & N. S. Ahmed. **Improved RC4 Algorithm Based on Multi-Chaotic Maps.** *Research Journal of Applied Sciences, Engineering and Technology*, 15(1), 1-6, 2018.
21. O. Yavanoglu & M. Aydos. **A review on cyber security datasets for machine learning algorithms**. **In 2017 IEEE International Conference on Big Data (Big Data)** (pp. 2186-2193), 2017.