# Evaluating Features Selection on NSL-KDD Data-Set to Train a Support Vector Machine-Based Intrusion Detection System

Luis Alfredo Álvarez Almeida
*Universidad Tecnologica de Bolivar*
Cartagena, Colombia
luisalvarezalmeida@gmail.com

Juan Carlos Martinez Santos
*Universidad Tecnologica de Bolivar*
Cartagena, Colombia
jcmartinezs@utb.edu.co

***Abstract*—The integrity of information and services is one of the more evident concerns in the world of global information security, due to the fact that it has economic repercussions on the digital industry. For this reason, big companies spend a lot of money on systems that protect them against cyber-attacks like Denial of Service attacks. In this article, we will use all the attributes of the data-set NSL-KDD to train and test a Support Vector Machine model. This model will then be applied to a method of feature selection to obtain the most relevant attributes within the aforementioned data-set and train the model again. The main goal is comparing the results obtained in both instances of training and validate which was more efficient.**

***Index Terms*—Machine Learning, Data set , Classification Model, Dos Attacks, Support Vector Machine, Feature Selection**

## I. Introduction

The networks computer has many applications in the real word, for instance: business data processing, education and learning, collaboration, widespread data acquisition and entertainment. One disadvantage of networks protocol communication is the flexibility of them for allowing that intruders launch attacks [2]. This attacks are more devastating every day. For this reason, the possibility that appear novel each day is highly

Companies that offer their services through the Internet are vulnerable cyber-attacks, putting at risk the availability of their services to their target audience. Denial of Service attacks are one the most popular on the internet. The effects cause by these attacks can limit the bandwidth of web servers, limiting the number of users that can access any given service. In other words, they produce the Denial of Services [8].

Nowadays, there are systems that have mechanisms that protects companies against the different threats on the web, like Intrusion Detection Systems (IDS). Their main task is to detect or recognize behavior patterns of users [on the web]. IDS can be implemented with two focuses: based on signatures and based on anomalies.The development [Body] of this article will be focused on designing an Intrusion Detection System based on the detection of anomalies using automated learning techniques that are capable of taking advantage of the quality of the attributes provided by the data-set NSL-KDD [1] and the attributes obtained in the preprocessing phase applied to said data-set.The main objective is to use the algorithms of the Support Vector Machines to train and validate the accuracy of the classification of the selected model. The method proposed in this project is based on the Knowledge Discovery in Databases (KDD) technique. This approach permits analyzing and processing large amounts of information, in order to identify relevant patterns in a selected data-set. This methodology suggests the development of various phases to obtain ideal results, like: Selection, preprocessing and transformation of data, and the evaluation of the results obtained during the implementation of the model [4].

### A. DDOS Attacks

They are one of the most popular threats in the world of global information security. The Main goal of these attacks is to prevent a legitimate user from accessing services offered by companies on the internet [11]. To perform them, the attackers have to recruit hundreds of hosts that are controlled through malware introduced into their systems. This group of devices is known as a Botnet (zombies or bots or slave agents) [6]. There are various motivations for the attackers to perform DDoS, among them being: Financial/economical gain, revenge, ideological beliefs, the possibility of an intellectual challenge and cyberwarfare [12].

### B. Intrusion Detection System

Due to the increase in cybernetic network attacks, researchers have proposed a mechanism to control such attacks, as they represent a financial threat to companies offering services on the Internet. Among the different mechanisms that exist to neutralize this type of attacks are the intrusion detection systems (IDS). Intrusion Detection Systems (IDS) are a tool that allows the identification of malicious activities within the traffic of a network [5].

978-1-7281-1614-3/19/$31.00 ©2019 IEEE

There are two main types of Intrusion Detection System (IDS): Signature Based IDS (SBIDS) and Anomaly Based IDS (ABIDS) [10].

SBIDS is well known, as it stops misuse, known attack signatures are stored in a rule file. When analyzing network traffic with this tool, anomalous events are compared with the signature database to find matches and generate an alert. The main drawback with this method is that our SBIDS will only be able to identify attacks that are stored in the signature database [5].

The number of interested researchers in ABIDS has increased significantly in the last few years, because this mechanism has the power to identify novel and strange attacks using machine learning algorithms [7].

ABIDS have two important advantages over signature based intrusion detection systems. Firstly, the ability to identify known attacks, for instance "zero day" attacks. Anomaly detection systems have the power to take as reference the normal network working. Finally, that the aforementioned profiles of normal activity are customized for every system, application and/or network. [10]

### C. Support Vector Machine

This method was proposed in 1998 by Vladimir Vapnik. Support Vector Machines (SVM) aim to find the better hyper-plane that is able to separate the data-set into different classes, taking into account that the better hyper-plane found is the one that allows it to trace the distance or margin maximum between hyper-plane and data point [3]. The main objective is to reduce the generalization of error.In this method it is not necessary to reduce the number of features, in order to avoid over-fitting. It is one of the reasons that SVM is a technique that is used a lot, because its performance is not affected by the number of selected features. Therefore, so it is ideal for working with great quantities of data. SVM allows it to adapt the hyper-planes to the data-set depending on the need, using Kernel functions. During the training phase, the user can provide the Kernel functions (Linear, Polynomial, Gaussian, RBF, etc.) required to adjust the model. Each of the Kernels that are selected will define its own support vectors that will be crucial to perform the classification [9].

## II. Dataset

The data-set we will use for this experiment is known as NSL-KDD [1]. This is an improved version of the KDDcup99 [2] data-set, which contains many irregularities like: redundant data, empty registries, duplicated registries, etc. The attacks contained in the NSL-KDD data-set are divided into:

1) DOS: denial of service
2) R2L: unauthorized access from a remote machine
3) U2R: unauthorized access to local super user (root)
4) Probing: surveillance and other probing

The attributes of the NSL-KDD data-set are defined in the following Table I, which describes each one of the features of the data-set. In this article, we only focus in records of DOS attacks.The experiment has two stages. First, preprocessing data. Second, evaluation and validation of model results.

TABLE I
Features of NSL KDD Dataset and Their Types and Numbers [9]

| Type | Features with their numbers |
|---|---|
| Categorical | protocol_type(1), service(2), flag(4) |
| Binary | land(7), logged_in(12),root_shell(14), su_attempted(15),is_host_login(21), is_guest_login)22) |
| Numeric | duration(1), src_bytes(5), dst_bytes(6), wrong_fragment(8), urgent(9), hot(10), num_failed_logins(11), num_compromised(13), num_root(16), num_file_creations(17), num_shells(18), num_access_files(19), num_outbound_cmds(20), count(23) srv_count(24), serror_rate(25), srv_serror_rate(26), rerror_rate(27), srv_rerror_rate(28), same_srv_rate(29) diff_srv_rate(30), srv_diff_host_rate(31), dst_host_count(32), dst_host_srv_count(33), dst_host_same_srv_rate(34), dst_host_diff_srv_rate(35),dst_host_same_src_port_rate(36), dst_host_srv_diff_host_rate(37), dst_host_serror_rate(38), dst_host_srv_serror_rate(39), dst_host_rerror_rate(40), dst_host_srv_ rerror_rate(41) |

## III. Methodology

To develop this experiment, the model will be evaluated in two ways. First, once the transformation and normalization data is realized in all data set and second, when the transformation, normalization, and reduction data have been applied, as showed in Figure 1. Furthermore, the evaluation of the model will be carried out in two instances. The main idea is to compared the result obtained from both evaluations to choose the better process.

### A. Process 1

In this phase, it is necessary to carry out a suitable transformation and normalization of the data-set. This will make the model more efficient and the classification percent will be relevant.

*1) Transformation Data:* In this stage, the get_dummies function of pandas provided by python will be implemented to convert categorical variables into dummy/indicator variables. Once the data transformation is performed, the data set becomes bigger than before, increasing from 42 to 115 columns, as show on the Table II. Because, when the transformation was performed 73 new columns were created.

*2) Normalization Data:* In this part of process all attributes will formatted to the same scale. Here, we will use the Scikit-learn [3], which provides the Min-Max method of standardization.

*3) Evaluation of the Model:* Now, we will use the attributes normalized and defined above in the Table II, to evaluate the Polynomial, RBF and Gaussian kernels of the Support Vector Machine to validate the precision of each one. Also, the Cross Validation test will be implemented. In this test, the training

[1]https://www.unb.ca/cic/datasets/nsl.html

[2]http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[3]https://scikit-learn.org/

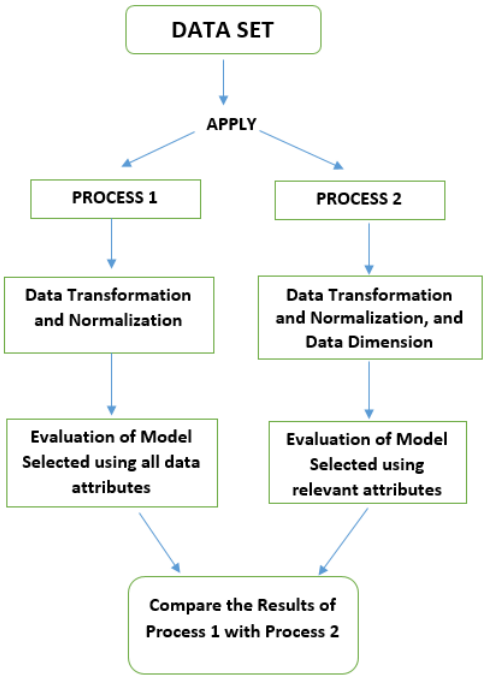


Fig. 1. Phases of Methodology

set is split into k smaller sets, in this occasion k = 10. This is necessary to train the model and identify the over-fitting in time.

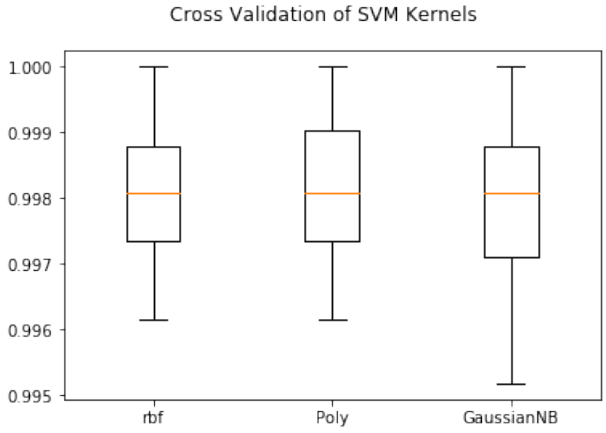


Fig. 2. Cross Validation for each of the kernels

The results show that when the model is tested using all features of data the phenomenon called over-fitting occurs. as shown in the Figure 2 which represent a box plot of Cross Validation for each kernels of model. The above can be verified in the Table III. Because, the values of Cross Validations is too close to one, for example: RBF = 0.998068, Polynomial = 0.998165 and Gaussian = 0.997972.

### B. Process 2

At this stage, we have already implemented data transformation and standardization. Now, we will apply the next step:

TABLE II
TRANSFORMATION FEATURES APPLIED TO DATA SET

| Type | Attributes |
|---|---|
| Binary | duration, src_bytes, dst_bytes, land, wrong_fragment, urgent, hot, num_failed_logins, logged_in, num_compromised, root_shell, su_attempted, num_root, num_file_creations, num_shells, num_access_files, num_outbound_cmds, is_host_login, is_guest_login, count, srv_count, serror_rate, srv_serror_rate, rerror_rate, srv_rerror_rate, same_srv_rate, diff_srv_rate, srv_diff_host_rate, dst_host_count, dst_host_srv_count, dst_host_same_srv_rate, dst_host_diff_srv_rate, dst_host_same_src_port_rat, dst_host_srv_diff_host_rate, dst_host_serror_rate, dst_host_srv_serror_rate, dst_host_rerror_rate, dst_host_srv_rerror_rate, type, k, enco_icmp, enco_tcp, enco_udp, enco_IRC, enco_X11, enco_Z39_50, enco_auth, enco_bgp, enco_courier, enco_csnet_ns, enco_ctf, enco_daytime, enco_discard, enco_domain, enco_domain_u, enco_echo, enco_eco_i, enco_ecr_i, enco_efs, enco_exec, enco_finger, enco_ftp, enco_ftp_data, enco_gopher, enco_hostnames, enco_http, enco_http_443, enco_imap4, enco_iso_tsap, enco_klogin, enco_kshell, enco_ldap, enco_link, enco_login, enco_mtp, enco_name, enco_netbios_dgm, enco_netbios_ns, enco_netbios_ssn, enco_netstat, enco_nnsp, enco_nntp, enco_ntp_u, enco_other, enco_pop_2, enco_pop_3, enco_printer, enco_private, enco_remote_job, enco_rje, enco_shell, enco_smtp, enco_sql_net, enco_ssh, enco_sunrpc, enco_supdup, enco_systat, enco_telnet, enco_tftp_u, enco_tim_i, enco_time, enco_urp_i, enco_uucp, enco_uucp_path, enco_vmnet, enco_whois, enco_REJ, enco_RSTO, enco_RSTOS0, enco_RSTR, enco_S0, enco_S1, enco_S2, enco_S3, enco_SF. |

TABLE III
MODELS PERFORMANCE USING CLASSIFICATION METRICS

| RBF | Polynomial | Gaussian | Metrics |
|---|---|---|---|
| 0.996275 | 0.997647 | 0.996078 | Accuracy |
| 0.996154 | 0.998899 | 0.995607 | Precision |
| 0.993425 | 0.994521 | 0.993425 | Recall |
| 0.998068 | 0.998165 | 0.997972 | Cross Validation |

Dimensionality Reduction of Features and Evaluation of the Model.

*1) Dimensionality Reduction of Features:* : To carry out the reduction of features the $VarianceThreshold$ function provided by Scikit-learn python library will implemented. This function makes it easier to evaluate the $p - value$ parameter defined in Equation 1 to select the most relevant features in all data-sets. This method considers that Boolean features are Bernoulli random variables, and the variance of them is defined by:

$$VAR(X) = p * (1 - p) \tag{1}$$

Since the goal is to select the best features for the model, we iterated the parameter 'p' between a value of 0.6 and 0.9 to get the Table IV. This table has four important metrics: accuracy, precision, recall and finally cross-validation score that will help us to validate the system. In this part of the process, we will be able to identify the appropriate p-value for selecting the best attributes.

TABLE IV
RESULTS OF CHANGING P-VALUE PARAMETER

| $p$ **VALUE** | | | | |
|---|---|---|---|---|
| 0.6 | 0.7 | 0.8 | 0.9 | Metrics |
| 0.820000 | 0.823922 | 0.921961 | 0.982745 | Accuracy |
| 0.687623 | 0.694466 | 0.954043 | 0.978087 | Precision |
| 0.934330 | 0.939873 | 0.837162 | 0.974960 | Recall |
| 0.825926 | 0.823995 | 0.926876 | 0.980777 | Cross Validation |

The achieved result shows us that when the Variance method iterated with p equal to 0.9 the model could suffer an over-fitting, because the score obtained was 0.980777, and it is very close from 1 which is a perfect score. Thus, the model could miss classify the new entries. Therefore to reduce the risk of over-fitting we selected $p$ equal to 0.8, because the measurements obtained with each of the metrics, show a better behavior of the model and It's a little further from 1. Thus, the possibility of over-fitting is lower. The attributes obtained iterating with $p - value$ equal to 0.8 are showed in the Table V.

TABLE V
TABLE OF FEATURES SELECTED [2]

| **No** | **Attribute Name** | **Description** |
|---|---|---|
| 1 | Logged_in | Login Status : 1 if successfully logged in; 0 otherwise |
| 2 | Serror_rate | The percentage of connections that have activated the flag (4) s0, s1, s2 or s3, among the connections aggregated in count (23) |
| 3 | Srv_serror_rate | The percentage of connections that have activated the flag (4) s0, s1, s2 or s3, among the connections aggregated in srv_count (24) |
| 4 | Same_srv_rate | The percentage of connections that were to the same service, among the connections aggregated in count (23) |
| 5 | Dst_host_srv_count | Number of connections having the same port number |
| 6 | Dst_host_same_srv_rate | The percentage of connections that were to the same service, among the connections aggregated in dst_host_count (32) |
| 7 | Dst_host_rerror_rate | The percentage of connections that have activated the flag (4) REJ, among the connections aggregated in dst_host_count (32) |
| 8 | Dst_host_srv_rerror_rate | The percentage of connections that have activated the flag (4) REJ, among the connections aggregated in dst_host_srv_c ount (33) |
| 9 | enco_host_name | Destination network service used |
| 10 | enco_printer | Destination network service used |
| 11 | enco_S3 | Destination network service used |

*2) Evaluation of Model:* The evaluation of model will be realized taking account the features obtained in the Table V and the result are showed in the Table VI. In this part of process each model was evaluated using classification metrics, such as: Accuracy, Precision and recall. The results observed are satisfactory. Because, Unlike the result observed in the Table III, the rating of each mean decreases. Moreover, The cross validation test was realized using each one of kernels of SVM chosen (RBF, Polynomial and Gaussian). It is a suitable technique. In process, the training data set was splitting in k = 10 parts. The aim of this, is training the model selecting random partition of data and then test the model. Once, the model was tested using the cross validations test, a new vector mean for each kernel is generated. The value of this vector help us to identify over-fitting in the process. Thus, selecting a suitable model score it would be easier for the model classify the new entries correctly and also, this will prevent choose the model that has perfect score. The Figure 3 shows a box plot of the cross validation performance.can be noticed the Kernels with better performance are: RBF and Gaussian.

TABLE VI
MODELS PERFORMANCE USING CLASSIFICATION METRICS

| **Models** | | | | |
|---|---|---|---|---|
| No. | RBF | Polynomial | GaussianNB | Metrics |
| 0 | 0.922745 | 0.916078 | 0.912157 | Accuracy |
| 1 | 0.959538 | 0.956329 | 0.952866 | Precision |
| 2 | 0.818630 | 0.808021 | 0.800000 | Recall |
| 2 | 0.918565 | 0.915474 | 0.919435 | Cross Validation |

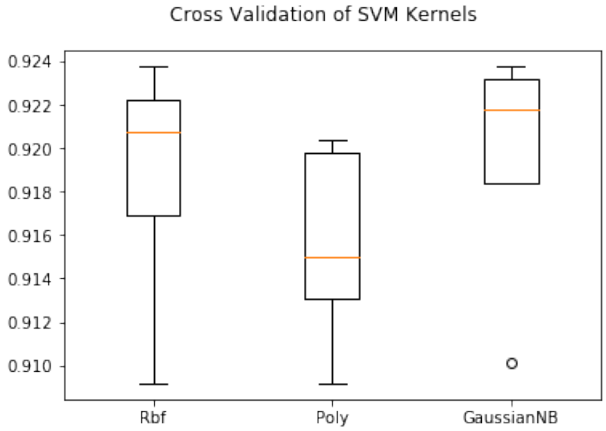


Fig. 3. Box plot: Cross Validation

## IV. CONCLUSIONS

The availability of services is the principal concern of companies. Because, each day, new and novel attacks are arriving at the internet. Denial of Service attacks are a powerful and danger weapon to break down the any system. For this, is convenient has a robust and secure Intrusion Detection Systems anomaly-based that can be able to detect this kind of attacks. In this experiment, we can see how the model was gradually evolved the results of precision due to three fundamentals stage implemented in each process. The preprocessing data phase is very important in this process. Because, the model archive classify better and it get better results. Moreover, is necessary Taking in account the SVM model work better when the data has been normalized and transformed to numerical values.

Also, it is important to highlight that when the most relevant attributes of the data set are chosen, a significant improvement is produced, such as: minimization of execution times and high classification accuracy. Also it was possible to reduce the number of attributes from 115 to only 11 features. Which will reduce computational expense. Since, if the algorithm works with all attributes of the data it can generalize the knowledge of the model highly and it would only recognize behaviors similar to the data supplied in the training. Finally, we can say that that the first process promised a lot but this produced very over-adjusted results. Therefore, once the model is tested it can fail. However, the second process allowed to choose the features that are most useful to train the model. Another important point is that the GuassianNB kernel is ideal to build a Intrusion Detection System, as described in Fig. 3.

## REFERENCES


[1] Canadian Institute for Cybersecurity. NSL-KDD Dataset, 1999.

[2] L Dhanabal and SP Shantharajah. A study on nsl-kdd dataset for intrusion detection system based on classification algorithms. *International Journal of Advanced Research in Computer and Communication Engineering*, 4(6):446–452, 2015.

[3] Khalid Fakieh. Survey on ddos attacks prevention and detection in cloud. *International Journal of Applied Information Systems*, 12 2016.

[4] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. The kdd process for extracting useful knowledge from volumes of data. *Communications of the ACM*, 39(11):27–34, 1996.

[5] Manasi Gyanchandani, JL Rana, and RN Yadav. Taxonomy of anomaly based intrusion detection system: a review. *International Journal of Scientific and Research Publications*, 2(12):1–13, 2012.

[6] Parneet Kaur, Manish Kumar, and Abhinav Bhandari. A review of detection approaches for distributed denial of service attacks. *Systems Science & Control Engineering*, 5(1):301–320, January 2017.

[7] Markos Markou and Sameer Singh. Novelty detection: a reviewpart 2:: neural network based approaches. *Signal processing*, 83(12):2499–2521, 2003.

[8] Nisharani Meti, DG Narayan, and VP Baligar. Detection of distributed denial of service attacks using machine learning algorithms in software defined networks. In *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 1366–1371. IEEE, 2017.

[9] Mohammad Reza Parsaei, Samaneh Miri Rostami, and Reza Javidan. A hybrid data mining approach for intrusion detection on imbalanced nsl-kdd dataset. *International Journal of Advanced Computer Science and Applications*, 7(6):20–25, 2016.

[10] Animesh Patcha and Jung-Min Park. An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer networks*, 51(12):3448–3470, 2007.

[11] NARESH BODDULA2 SRINIVAS KALIME1. A study on detection of distributed denial of service attacks using machine learning techniques. *International Journal of Research*, page 10, 2018.

[12] Saman Taghavi Zargar, James Joshi, and David Tipper. A Survey of Defense Mechanisms Against Distributed Denial of Service (DDoS) Flooding Attacks. *IEEE Communications Surveys & Tutorials*, 15(4):2046–2069, 2013.