iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** XII **Month of publication:** December 2021

DOI: https://doi.org/10.22214/ijraset.2021.39640

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 9 Issue XII Dec 2021- Available at www.ijraset.com

Detection of Network Attacks using Machine Learning: A New Approach

Avinash R. Sonule[1], Mukesh Kalla[2], Amit Jain[3], D. S. Chouhan[4]
[1, 2, 3, 4]*Department of Computer Science & Engineering, Sir Padampat Singhania University (SPSU), Udaipur-313601, Rajasthan, India*

Abstract: ***The Cyber-attacks become the most important security problems in the today's world. With the increase in use of computing resources connected to the Internet like computers, mobiles, sensors, IoTs in networks, Big Data, Web Applications/Server, Clouds and other computing resources, hackers and malicious users are planning new ways of network intrusions. Many techniques have been developed to detect these intrusions which are based on data mining and machine learning methods. These intrusions detection techniques have been applied on various IDS datasets. UNSW-NB15 is the latest dataset. This data set contains different modern attack types and wide varieties of real normal activities. In this paper, we compare Naïve Bays algorithm with proposed probability based supervised machine learning algorithms using reduced UNSW NB15 dataset.***

Keywords: UNSW NB-15, Machine Learning, Naïve Bayes, All to Single (AS) features probability Algorithm.

I. INTRODUCTION

The increase in the number of devices connected to the Internet has resulted in a number of useful solutions in different fields such as agriculture, health care, commerce, IT and other industry. Such a huge increase in demand for connectivity has challenged the traditional network architectures. The networks can be accessed using a number of ways and this becomes a threat to the network. To overcome this problem, the system will be able to predict any type of threat to the network keep it secure. Thus, our system can predict attacks even before they happen in order to warn the users of the potential threat that may affect them.

Cyber security is a broad field of research, and the detection of malicious activities on the network is among the oldest and most common problems. However, intrusion detection is mostly reactive and responses to specific patterns or observed anomalies. The intuitive next step is taking a proactive approach, in which there is a need to preemptively infer the upcoming malicious activities so that we could react to such events before they cause any harm. Research efforts and progress in predictions and forecasting in cyber security are not as prominent as attack detection. However, it is gaining more attention, and a breakthrough in this field would benefit the whole discipline of cyber security.

Currently, most of the organizations rely on traditional security options in order to secure their data. But this has led to attackers exploiting their security systems. In order to provide better security, organizations must be aware of the threats they face, in order to tackle those threats. Our paper aims to provide network attack prediction in two categories..

Intrusion Detection Systems (IDS)[1][2] is a device or software application that monitors network and the system for suspicious activities and warns the system or network administrator. There are Host based IDS and Network based IDS. A Host based Intrusion Detection System keeps track of individual host machine and gives notice to the user if suspicious activities like deleting or modifying a system file, undesired configuration changes, unnecessary system calls sequence are found. Generally, a Network based Intrusion Detection System(NIDS)[3] is kept at network points like a gateway or routers to detect the intrusions over the network.

A NIDS monitors and detects network-attack patterns over networking environments and protect computing resources against malicious activities. At high level, IDS can be categorized by the detection mechanism used by it. These IDSes are : i) misuse detection, ii) anomaly detection and iii) hybrid detection. Misuse detection techniques have been used to detect known attacks while the Anomaly detection techniques have been used to detect unknown attacks.

Machine Learning (ML) can be used for all the three types of detection techniques. Machine learning is subclass of Artificial Intelligence (AI) which are used in computers. A machine learning models have two parts: training and testing. By using training data samples as a input, learning algorithm learn the features in the training. In the testing, the learning algorithm predicts the unknown data.

©IJRASET: All Rights are Reserved

Machine learning algorithms are applied on different network attack datasets with or without feature selection approaches. Supervised learning algorithms build a mathematical model of a set of data which contains both the inputs and the desired outputs. The data is known as training data, and consists of a set of training examples. Each training example has one or more inputs and a desired output. It is also known as a supervisory signal. Unsupervised learning algorithms take a set of data that contains only inputs, and find pattern in the data, such as grouping or clustering of data points. The algorithms therefore learn from test data that has not been labeled, classified or categorized. Instead of responding to feedback, unsupervised learning algorithms identify commonalities in the data and react based on the presence or absence of such commonalities in each new piece of data.

To provide network attack prediction based on historical attack data. Providing accurate results on what type of attack may happen based on multiple factors. The rest of the paper is organized as follows: section II gives related work for probability based algorithms . Section III gives in detail of the Naïve Bayes drawbacks with examples and UNSW NB 15 dataset[4][5]. In section IV, proposed methods with All to Single (AS) features probability Algorithm with examples **a**re given. Section V gives comparisons and results. Finally, section VI gives future direction and concludes the work.

II. RELATED WORK

Many researchers have used different Machine Learning algorithms on different Datasets. Priya et al.[6] have done survey on different machine learning algorithms applied on various datasets The different machine learning algorithms have applied on] UNSW NB 15 dataset. The following some researchers have used Naïve Bays algorithm on UNSW NB 15 dataset.

Moustafa et al. [7] suggested an approach which reduce the irrelevant features set which then used with machine learning methods to detect intrusion. An NIDS architecture is then used for anomaly intrusion detection and misuse intrusion detection. NIDS takes the input from the UNSW-NB15 dataset and then computes the center points of attribute values which is the most frequent value. All these center points are given to the Apriori algorithm as an input to reduce processing time. This Apriori algorithm finds out the highly ranked attributes /features using the correlation of the two or more attributes. The filtered dataset which consists of the selected features feed to the detection engine. They applied three ML algorithms on UNSW-NB15. d Naive Bayes (NB) gives 79.5% accuracy and 23.5% FAR.

Bhamare et al. [8] presented the machine learning approach to detect the cyber-attack. They have used different ML algorithms on UNSW-NB15 dataset. This has comprehensive representation of modern attack which give real attack scenarios. Misuse detection techniques such as LR, NB, DT and Support Vector Machine use 3 different kernels such as Polynomial, Linear, RBF are applied on Dataset. NB gives an accuracy of 73.8%, with RBF kernel gives accuracy 70.15% , poly function based NB gives FPR 7.3%

Anwer et al.[9] proposed framework for efficient network anomaly detection using different machine learning classifiers. The feature selection framework applies five different strategies for features selection. The aim of this framework is to select the minimum number of features that gives the highest accuracy. UNSW-NB15 dataset is used in the experimental results to evaluate the proposed framework. J48 and Naïve Bayes algorithms are used as classifiers. The experimental results obtained show that, the best strategy is by using 18 features from the GR ranking method and applying J48 as a classifier getting an accuracy of 88% and a speedup factor of 2.

Moustafa et al.[10] proposed an ensemble intrusion detection technique to reduce malicious events particularly botnet attacks against DNS, HTTP and MQTT protocols utilized in IoT networks. From these protocols new statistical flow features are obtained based on an analysis of their potential properties. Then, ensemble learning method named AdaBoost is developed using Decision Tree (DT), Naive Bayes (NB) and Artificial Neural Network (ANN) machine learning techniques. AdaBoost evaluates the effect of these features and detect malicious events effectively.

The UNSW-NB15 with simulated IoT sensors' data are used to extract the proposed features and evaluate the ensemble technique. The proposed ensemble technique provides a higher detection rate and a lower false positive rate compared with each classification technique included in the framework. The simplest feature selection method Correlation Coefficient (CC) is used to compute the strength degree between some features. Using the DNS data source of the UNSW-NB15 dataset, the accuracy and DR of the ensemble method achieved 99.54% and 98.93%, respectively, while the FPR produces 1.38%, which outperforms the performance of the DT, NB and ANN techniques. HTTP data source of the UNSW-NB15 dataset, the accuracy and DR of the ensemble method achieved is 98.97%, 97.02% and FPR 2.58%. The DT technique produces a 95.32% accuracy, 94.15% DR and 5.22% FPR, and then the ANN technique achieves a 92.61% accuracy, 91.48% DR and 7.87% FPR. Lastly, the NB technique achieves an accuracy rate of 91.17%, 90.78% DR and 8.25% FPR.

©IJRASET: All Rights are Reserved

Beloucha et al.[11] proposed a framework which evaluates the performance of four classification algorithms; SVM, Naive Bayes, Decision Tree and Random Forest using Apache Spark for intrusion detection in network traffic. Apache Spark a big data processing tool. Using UNSW-NB15 Naive Bayes and SVM have almost same sensitivity with values 92.46% and 92.13% .They found that specificity for the Random Forest and Decision Tree based schemes are almost same with 97.75% and 97.10% respectively. However, specificity for SVM based scheme is about 91.15%. Naive Bayes provides lowest Specificity. the accuracy of the Naive Bayes based scheme is lower among the all schemes with 74.19%.

Nawir et al.[12] proposed Network Intrusion Detection System using machine learning algorithms for binary classification. They used three types of ML algorithms from Bayesian's family in WEKA tools. They are Average One Dependence Estimator (AODE), Bayesian Network (BN), and Naive Bayes (NB).

The performance these classifiers measured in term of classification rate and processing time for classifier model to classify the data instances of UNSW-NB15 dataset. The parameters of these classifiers set to default as in WEKA and using tenfold cross validation to validate the training set before the model been tested. It is found that AODE is processing fast for network anomaly detection system compared to other two classifiers with accuracy 94.37% with training time 4.13s. BN algorithm gives the accuracy 92.70% and time taken is 4.17s. Naive Bayes algorithm required small amount of time but its accuracy is not comparable to AODE and BN algorithms

III. NAIVE BAYES CLASSIFIER AND DATASET

Naive Bayes classifier is based on the Bayesian learning method and it is found to be useful in many applications. It is called "naive" because it is based on the simplifying assumption that attribute values are conditionally independent of each other. It is applied to the learning task where each instance x can be described by a conjunction of attributes and where the target function f(x) can take any of the value from some finite set V (a set of target values).

It estimates the posterior probabilities of observing a class label from a set of normal class and anomaly class labels. For a given test instance, Class label with largest posterior is chosen as the predicted class.

Naive Bayes classifier achieves a fast speed of detection and is simpler than other classifiers. However, it makes an assumption that features are independent of each other.

This independent relation assumption may not hold true in detecting various types of attacks. For example, in the publicly available UNSW NB 15 intrusion detection dataset, the features are highly dependent on each other.

Bayes theorem provides a way of calculating the posterior probability, **P(c|x), from P(c), P(x), and P(x|c)**. Naive Bayes classifier assume that the effect of the value of a predictor (x) on a given class (c) is independent of the values of other predictors. ... P(c) is the prior probability of class.

The conditional probability can be calculated using the joint probability, although it would be intractable. Bayes Theorem provides a principled way for calculating the conditional probability. The simple form of the calculation for Bayes Theorem is as follows: **P(A|B) = P(B|A) * P(A) / P(B)**

The conditional probability can be calculated using the joint probability, although it would be intractable. Bayes Theorem provides a principled way for calculating the conditional probability. The simple form of the calculation for Bayes Theorem is as follows: **P(A|B) = P(B|A) * P(A) / P(B)**

The main limitation of Naive Bayes is the assumption of independent predictor features. Naive Bayes implicitly assumes that all the attributes are mutually independent. In real life, it's almost impossible that we get a set of predictors that are completely independent or one another.

UNSW-NB15 Dataset: The existing datasets do not represent the modern network traffic with different attack scenarios. The cyber security research group at the Australian Centre for Cyber Security (ACCS) and other researchers of this domain around the globe took this as a challenge. The raw network packets of the UNSW-NB15 dataset[6] was created by the IXIA PerfectStorm tool in the Cyber Range Lab of ACCS for generating a hybrid of real modern normal activities and synthetic contemporary attack behaviors. The Argus, Bro-IDS tools are used with twelve algorithms to generate total 49 features with the class label. For this paper we have used reduced (few tuples) of UNSW-NB15 dataset with 4 attributes and Class Label. The column having discrete values are and binary class labels are taken. The reduced UNSW NB15 dataset with 20 tuples is divided into training and testing datasets.

Table 1: Training Dataset

Sr No	proto used	service used	ct_srv_src	ct_src_ltm	Class
1	UDP	No	3	1	Attack
2	UDP	Yes	3	1	Attack
3	UDP	No	2	1	Normal
4	UDP	No	1	3	Normal
5	TCP	No	1	2	Normal
6	TCP	Yes	1	2	Attack
7	TCP	Yes	2	2	Normal
8	UDP	No	3	3	Attack
9	TCP	No	3	2	Normal
10	TCP	No	1	3	Normal
11	TCP	Yes	3	3	Normal
12	UDP	Yes	2	3	Normal
13	TCP	No	2	1	Normal
14	UDP	Yes	1	3	Attack

Table 2:Tesing Dataset

Sr No	proto used	service used	ct_srv_src	ct_src_ltm	Class
1	UDP	Yes	2	2	Normal
2	UDP	No	1	1	Normal
3	TCP	No	2	3	Attack
4	UDP	No	1	3	Normal
5	UDP	Yes	3	2	Attack
6	UDP	Yes	3	3	Attack

IV. PROPOSED METHODS

The reduced UNSW NB 15 dataset as shown in Table 1 with 4 attributes/features proto(Flow feature), service(Basic Feature), ct_srv_src (Connection features)(No. of connections that contain the same service (http, ftp, ssh, dns ..,else (-)) and source address (Source IP address) in 100 connections according to the last time (The content size of the data transferred from the server's http service) , ct_src_ltm(No. of connections of the same source address (Source IP address) in 100 connections according to the last time (The content size of the data transferred from the server's http service).) and class with some discrete values are used. The features proto, service, ct_srv_src , ct_src_ltm are Transaction protocol used, whether any services(dns/http/etc) used or not,, respectively. The binary class labels are Attack and Normal. The reduced UNSW NB15 dataset is divided into training and testing in 70 and 30 % respectively. Table 1 and Table 2 show the training and testing datasets respectively.

Preparation of Training/Testing Dataset: Every test sample is compared/tested with every tuple of training dataset for the $2^{\text{no of attributes}}$ -1 times . It means every rows/tuple of the training and dataset are converted to $2^{\text{no of attributes}}$ -1 sub rows or sub tuples. Every attribute values of sample is compared with respective attribute of training dataset. Initially row is created with all attribute values. Next rows are created with n-1 attributes. This is done till the number of attributes are reduced one attribute. If there are 4 attributes then 2^4 -1=15 rows are created. 1 row with all 4 attributes values, 4 rows with any three attribute values, 6 rows with any two attribute values, 4 rows with one attribute values.

The total number of rows ${}^4C_4 + {}^4C_3 + {}^4C_2 + {}^4C_1 = 1 + 4 + 6 + 4 = 15$ rows .

$$\sum_{r=n}^{1} {}^nC_r$$

Where n is the total number of attributes/columns

${}^4C_4=1$, ${}^4C_3=4$, ${}^4C_2=6$, ${}^4C_1=4$,,, $\sum {}^nC_r$ => 1+4+6+4 =15 rows

©IJRASET: All Rights are Reserved

Test sample with 4 attributes values shown in table 3 is converted $2^{\text{no of attributes}}$ -1= 2^4 -1=15 sub rows while training and testing as shown in table 4. The tuple with 3 attributes is converted to $2^{\text{no of attributes}}$ -1= 2^3 -1=7 and so on.

Table 3: Sample tuple

Sr No	proto used	service used	ct_srv_src	ct_src_ltm	Class
1	UDP	Yes	2	2	?

Table 4: Tuple 1 with all sub tuples

Sr No	proto used	service used	ct_srv_src	ct_src_ltm	Class
1	UDP	No	3	1	Attack
2	UDP	No	3		Attack
3	UDP	No		1	Attack
4	UDP		3	1	Attack
5		No	3	1	Attack
6	UDP	No			Attack
7	UDP		3		Attack
8	UDP			1	Attack
9		No	3		Attack
10		No		1	Attack
11			3	1	Attack
12	UDP				Attack
13		No			Attack
14			3		Attack
15				1	Attack

A. AS Algorithm 1

1) Set class variable count values to zeros. Scan the dataset tuple by tuple and match all the n attribute values of sample tuple with tuples of training dataset. If all n attributes values matched, increase the count of respective class variables .Mark the tuples of training dataset where all/n/maximum attributes values of sample tuples are matched with the attribute values of training dataset.

2) Find probabilities of the classes from the marked tuples i.e $P(C_1)$ and $P(C_2)$

3) Find the Class with highest probability

4) Assign class for sample tuple which have highest probability

5) If probability are the same for all classes REPEAT Step 1 to 4 for n-1 (or maximum -1) attributes and break)/(till single attribute).

Note: Only marked tuples are considered for calculation. In NB ,all tuples are considered for calculation.

To give the equal chances to all classes and all attributes rigorous conditional probabilities can be used. The conditional probability of class for n attributes with n-1 attributes.

B. AS Algorithm 2

1) Mark the tuples of dataset where all/n/maximum attributes values of sample tuples are matched with the attribute values of training dataset

2) Find probabilities of the classes for marked tuples i.e $P(C_1)$ and $P(C_2)$

3) REPEAT Step 1 to 2 for n-1 (or maximum -1) attributes till single attribute.

4) Find the(rigorous conditional probabilities) product of all probabilities for the respective classes.

5) Assign the Class to the sample tuple which have highest product.

Note: Only marked tuples are considered for calculation. In NB are tuples are considered for calculation that is disadvantage of NB which gives pure accuracy.

P(C1)= P(for n attributes) x P(for n-1 attributes) x P(for n-2 attributes) …P(1 attribute)
P(C2)= P(for n attributes) x P(for n-1 attributes) x P(for n-2 attributes) …P(1 attribute)
An unseen sample X= <UDP, yes,2, 2, >

By Naïve Bayes

An unseen sample X= < UDP, Yes, 2, 2 >

P(X|Normal).P(Normal) = P(UDP |N).P(Yes |N).P(2|N).P(2 |N).P(N)

=3/9 X 3/9 X 4/9 X 3/9 X 9/14

=0.01058

P(X|Attack).P(Attack) = P(UDP |A).P(Yes |A).P(2|A).P(2 |A)

=4/5 X 3/5 X 0 X 1/5 X 5/14 = 0

The instance X= < UDP, Yes, 2, 2 >will be classified as Normal.

By All to Sigle features probability Algorithm 2
An unseen sample X= <UDP, yes, 2, 2, >

Table 5

Sr No	proto used	service used	ct_srv_src	ct_src_ltm	Class
1	UDP	No	3	1	Attack
2	UDP	Yes	3	1	Attack
3	UDP	No	2	1	Normal
4	UDP	No	1	3	Normal
5	TCP	No	1	2	Normal
6	TCP	Yes	1	2	Attack
7	TCP	Yes	2	2	Normal
8	UDP	No	3	3	Attack
9	TCP	No	3	2	Normal
10	TCP	No	1	3	Normal
11	TCP	Yes	3	3	Normal
12	UDP	Yes	2	3	Normal
13	TCP	No	2	1	Normal
14	UDP	Yes	1	3	Attack

For 3 Attributes as shown in table 5

Class Probabilities

P(Normal=2/2

P(Attack)=0

For 2 Attributes Class Probabilities as shown in table 6

P(Normal)=3/6

P(Attack)=3/6

Table 6

Sr No	proto used	service used	ct_srv_src	ct_src_ltm	Class
1	UDP	No	3	1	Attack
2	UDP	Yes	3	1	Attack
3	UDP	No	2	1	Normal
4	UDP	No	1	3	Normal
5	TCP	No	1	2	Normal
6	TCP	Yes	1	2	Attack
7	TCP	Yes	2	2	Normal
8	UDP	No	3	3	Attack
9	TCP	No	3	2	Normal
10	TCP	No	1	3	Normal
11	TCP	Yes	3	3	Normal
12	UDP	Yes	2	3	Normal
13	TCP	No	2	1	Normal
14	UDP	Yes	1	3	Attack

Table 7

Sr No	proto used	service used	ct_srv_src	ct_src_ltm	Class
1	UDP	No	3	1	Attack
2	UDP	Yes	3	1	Attack
3	UDP	No	2	1	Normal
4	UDP	No	1	3	Normal
5	TCP	No	1	2	Normal
6	TCP	Yes	1	2	Attack
7	TCP	Yes	2	2	Normal
8	UDP	No	3	3	Attack
9	TCP	No	3	2	Normal
10	TCP	No	1	3	Normal
11	TCP	Yes	3	3	Normal
12	UDP	Yes	2	3	Normal
13	TCP	No	2	1	Normal
14	UDP	Yes	1	3	Attack

For Single Attribute Class Probabilities as shown in Table 7.

P(Normal)=8/13

P(Attack)=5/13

Taking all Probabilities together

P(Normal)= 8/13.3/6.2/2=48/78

P(Attack)= 5/13. 3/6.0=0

The instance X= < UDP, Yes, 2, 2 > will be classified as Normal.

Example 2 : An unseen sample X= < UDP, No, 1, 1,>

By Naïve Bayes Algorithm
An unseen sample X= < UDP, No, 1, 1,>

P(X|Normal).P(Normal) = P(UDP |N).P(No |N).P(1|N).P(1|N).P(N)
=3/9 X 2/9 X 3/9 X 6/9 X 9/14
=0.010582

P(X|Attack).P(Attack) = P(UDP |A).P(No |A).P(1|A).P(1 |A).P(A)
=2/5 X 2/5 X 4/5 X 2/5 X 5/14
=0.018286

The instance X= < UDP, No, 1, 1,> will be classified as Attack.

By All to Sigle feature probability Algorithm

Table 8

Sr No	proto used	service used	ct_srv_src	ct_src_ltm	Class
1	UDP	No	3	1	Attack
2	UDP	Yes	3	1	Attack
3	UDP	No	2	1	Normal
4	UDP	No	1	3	Normal
5	TCP	No	1	2	Normal
6	TCP	Yes	1	2	Attack
7	TCP	Yes	2	2	Normal
8	UDP	No	3	3	Attack
9	TCP	No	3	2	Normal
10	TCP	No	1	3	Normal
11	TCP	Yes	3	3	Normal
12	UDP	Yes	2	3	Normal
13	TCP	No	2	1	Normal
14	UDP	Yes	1	3	Attack

For 3 Attributes Class Probabilities as shown Table 8.
P(Normal)=2/3
P(Attack)=1/3

Table 9

Sr No	proto used	service used	ct_srv_src	ct_src_ltm	Class
1	UDP	No	3	1	Attack
2	UDP	Yes	3	1	Attack
3	UDP	No	2	1	Normal
4	UDP	No	1	3	Normal
5	TCP	No	1	2	Normal
6	TCP	Yes	1	2	Attack
7	TCP	Yes	2	2	Normal
8	UDP	No	3	3	Attack
9	TCP	No	3	2	Normal
10	TCP	No	1	3	Normal
11	TCP	Yes	3	3	Normal
12	UDP	Yes	2	3	Normal
13	TCP	No	2	1	Normal
14	UDP	Yes	1	3	Attack

For 2 Attributes Class Probabilities as shown in Table 9.

P(Normal)=5/9

P(Attack)=4/9

For Single Attributes Class Probabilities as shown in Table 10.

P(Normal)=7/12

P(Attack)==5/12

Taking all Probabilities together

P(Normal) = 2/3. 5/9. 7/12 = 70/324

P(Attack) = 1/3. 4/9. 5/12.= 20/324

The instance X= < UDP, No, 1, 1,> will be classified as Normal

Table 10

Sr No	proto used	service used	ct_srv_src	ct_src_ltm	Class
1	UDP	No	3	1	Attack
2	UDP	Yes	3	1	Attack
3	UDP	No	2	1	Normal
4	UDP	No	1	3	Normal
5	TCP	No	1	2	Normal
6	TCP	Yes	1	2	Attack
7	TCP	Yes	2	2	Normal
8	UDP	No	3	3	Attack
9	TCP	No	3	2	Normal
10	TCP	No	1	3	Normal
11	TCP	Yes	3	3	Normal
12	UDP	Yes	2	3	Normal
13	TCP	No	2	1	Normal
14	UDP	Yes	1	3	Attack

V. COMPARISONS & RESULTS

Only 12/13 marked tuples considered by AS and all 14 by NB. No need to consider those tuples where not a single attribute value matching. NB is calculating probabilities of all classes for all tuples. Both the AS features probability algorithms are giving the following confusion matrix shown in Table 11. The Confusion matrix for Naïve Bayes algorithm is shown in Table 11. The accuracy for NB is 66,67%.

Table 11. Confusion matrix for ASfpa

6	Class1 : Normal Predicted	Class 2: Attack Predicted	
Class1: Normal Actual	TP=3	FN=0	3
Class2: Attack Actual	FP=0	TN=3	3
	3	3	6

Table 11. Confusion matrix for Naïve Bayes

6	Class1 : Normal Predicted	Class 2: Attack Predicted	
Class1: Normal Actual	TP=2	FN=1	3
Class2: Attack Actual	FP=1	TN=2	3
	4	2	6

.

By NB

Accuracy=(TP+TN)/(TP+TN+FP+FN)=

(2+2)/(4+2+0+0)=5/6=0.6667

By ASfpa

Accuracy=(TP+TN)/(TP+TN+FP+FN)=

(4+2)/(4+2+0+0)=6/6=1

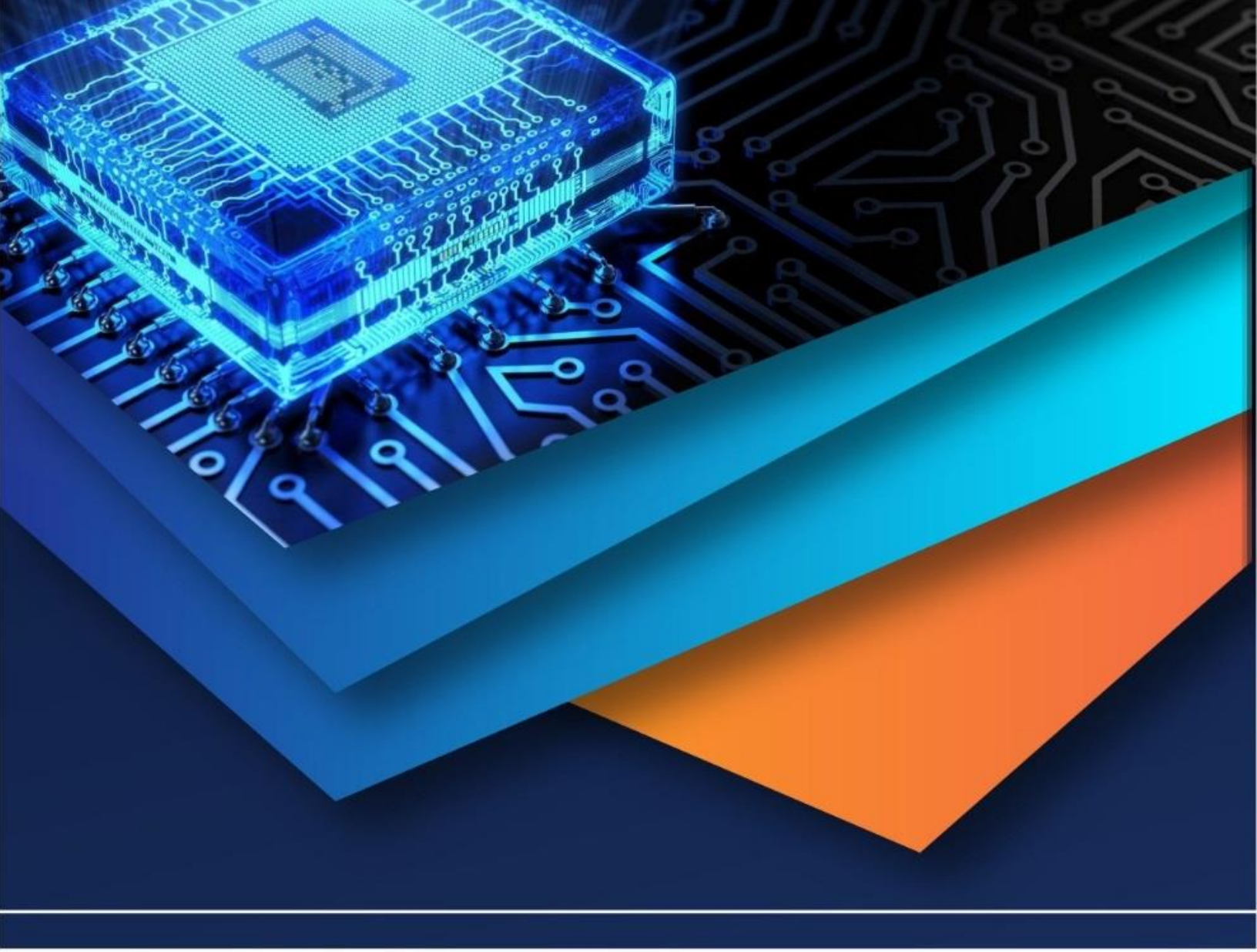
VI. CONCLUSION AND FUTURE WORK

In this paper Naïve Bays and Proposed All to Single feature rigorous Conditional probability applied on reduced UNSW NB15 dataset. We have used small dataset to concentrate more on algorithm. The proposed algorithm is giving good accuracy. This algorithm can be applied on any dataset of any size.

REFERENCES

[1] Heady R., Luger G., Maccabe A., Servilla M.:The architecture of a network level intrusion detection system, Tech. rep., Computer Science Department, University of New Mexico, New Mexico, (1990)
[2] Stefan A.: Intrusion detection systems: A survey and taxonomy, Technical report, Vol. 99, (2000).
[3] Vigna G., Kemmerer R. A. : Netstat: A network-based intrusion detection system, in Journal of Computer Security. Citeseer, (1999)
[4] Moustafa N., Slay J.:Unsw-nb15: A comprehensive data set for network intrusion detection systems (unsw-nb15 network data set), in Military Communications and Information Systems Conference (MilCIS), Canberra, Australia, pp. 1–6,(2015)
[5] Moustaf N, Slay J.:The evaluation of network anomaly detection systems: Statistical analysis of the unsw-nb15 data set and the comparison with the kdd99 data set, Information Security Journal: A Global Perspective, in press. ids (2015)
[6] Mishra P., Varadharajan V., Tupakula U., Pilli E.S.:A Detailed Investigation and Analysis of using Machine Learning Techniques for Intrusion Detection, IEEE Communications Surveys & Tutorials(2018)
[7] Moustafa N., Slay J.:A hybrid feature selection for network intrusion detection systems: Central points, pp. 1–10,(2015)
[8] Bhamare D. , Salman T., Samaka M. , Erbad A., R. Jain:Feasibility of supervised machine learning for cloud security, in International Conference on Information Science and Security (ICISS). IEEE, 1–5(2016)
[9] Anwer H.M., Farouk M., Abdel-Hamid A.: A Framework for Efficient Network Anomaly Intrusion Detection with Features Selection, 9th International Conference on Information and Communication Systems (ICICS) 2018, IEEE page no.157,(2018)
[10] Moustafa N., Turnbull B., Choo K.R.: An Ensemble Intrusion Detection Technique based on proposed Statistical Flow Features for Protecting Network Traffic of Internet of Things, IEEE Internet of Things Journal (2018)
[11] Beloucha M., Hadaja S.E.. Idhammadb M.: Performance evaluation of intrusion detection based on machine learning using Apache Spark, The First International Conference On Intelligent Computing in Data Sciences Performance, Procedia Computer Science 127 (2018) 1-6, Elsevier(2018)
[12] Nawir M., Amir A., Lynn O.B., Yaakob N., Ahmad R.B.: Performances of Machine Learning Algorithms for Binary Classification of Network Anomaly Detection System, 1st International Conference on Big Data and Cloud Computing (ICoBiC) 2017 IOP Publishing, IOP Conf. Series: Journal of Physics (2017)

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)