CLASSIFICATION OF FIREWALL LOG FILES USING SUPERVISED MACHINE LEARNING METHODS

SRI DANALAKSMI. P

(20PCS008)

Project Report Submitted

In partial fulfillment of the requirements for the Award of

Master of Science in Computer Science

DEPARTMENT OF COMPUTER SCIENCE

AVINASHILINGAM INSTITUTE FOR HOME SCIENCE AND

HIGHER EDUCATION FOR WOMEN

COIMBATORE – 641043

MAY – 2022

1. INTRODUCTION

A firewall is a security access management point that controls access to computer networks and ensures safe network connectivity. A network firewall is a system or set of systems that uses pre-configured rules or filters to regulate access between different networks, an assured network and an unassured network. The outcomes of firewall rules can be audited, verified, and evaluated via monitoring. The analysing and classifying the firewall, checks and decides the packets to pass it or not. It can improve security purpose even more by allowing based on the required protocols. Firewall rules specifies the different types of network traffic which are permitted or not permitted. A firewall rule can be used to block the network traffic coming from the public Internet to private computer (inbound) or traffic coming from private computer to the public Internet (outbound). A rule can be deployed in both set of traffics at the identical time.

The survey's findings show that network engineering teams are devoting more time and effort to firewall maintenance, and that their duties are becoming more difficult. The majority of these chores, according to over 45 percent of respondents, are still done by hand. It's challenging to keep up with everything since most teams are dealing with a multi-vendor environment with inherent complexity.

1.1 MOTIVATION AND JUSTIFICATION

In generation of thousands of firewall logs per day, classifying the log files may help to observe the files and reduce the risk of threats. Thus, this project has its own space and necessity to be developed.

1.2 PROBLEM STATEMENT

To analyse and classify the firewall logs in order to handle the traffic during network observance to check that each data packet arrives and also to decide whether or not to pass it.

1.3 OBJECTIVE

To design a methodology to analyse and classify the firewall logs using different machine learning classifiers based on the action in their activities to apprehend the logs, and the performance of the model is estimated using different metrics.

2. ABSTRACT

A firewall retains traffic entering and departing the domain it was supposed to protect. Some firewalls may provide information about the source and type of traffic entering the environment. A firewall's policy must be enhanced with a successful logging capability in order to be successful. The logging feature keeps track of how the firewall handlesdifferent sorts of traffic. Organizations can use the logs to find out things like Source IP addresses and destination IP addresses, protocols, and port numbers. Monitoring and analyzing log files can assist IT businesses improve the end-user reliability of their systems. Log files may consists ofmalicious texts, strings that tricks the users to hack the information. In generation of number of firewall logs per day, classifying the log files may help to observe more efficient, the number of unnecessary attributes can be minimized with the help of classification, resulting in a more efficient performance. The project title is ‘Classification of firewall log files using supervised machine learning methods’, the main intent of this project is to analyze and classify firewall logs which may consists of source port, destination port, bytes sent and received, etc., It checks thateach data packet arrives on both sides of the firewall, it then decides whether or not to pass it.Firewalls can improve security even more by allowing quite well control over which system functions and processes have access to networking resources. The process starts with data collection followed by pre-processing techniques and main features to be selected to build a framework using supervised machine learning algorithms. In classification problems, the selectionof appropriate and relevant dataset features plays a critical role. The feature selection approaches to improve the accuracy of classification system using Weka tool. Different classification techniques like Support Vector Machine, Naïve Bayes, Logistic Regression and K-Nearest Neighbor were adopted and their performance were analyzed.

KEYWORDS: Classification, Firewall, Log files, Network Security, Protocols, Supervised learning.

3. METHODOLOGY

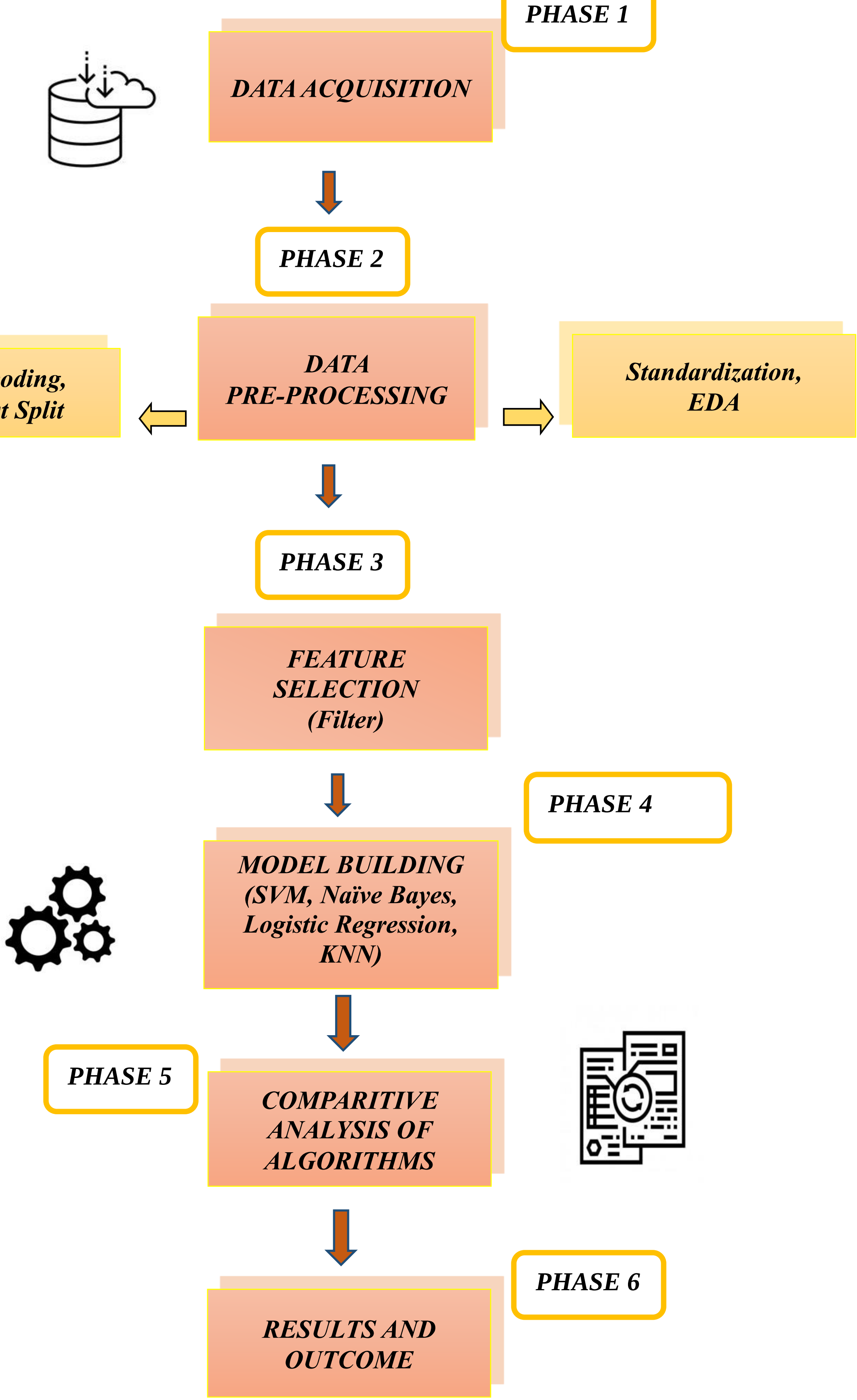


FIG. 3.1 METHODOLOGY OVERVIEW

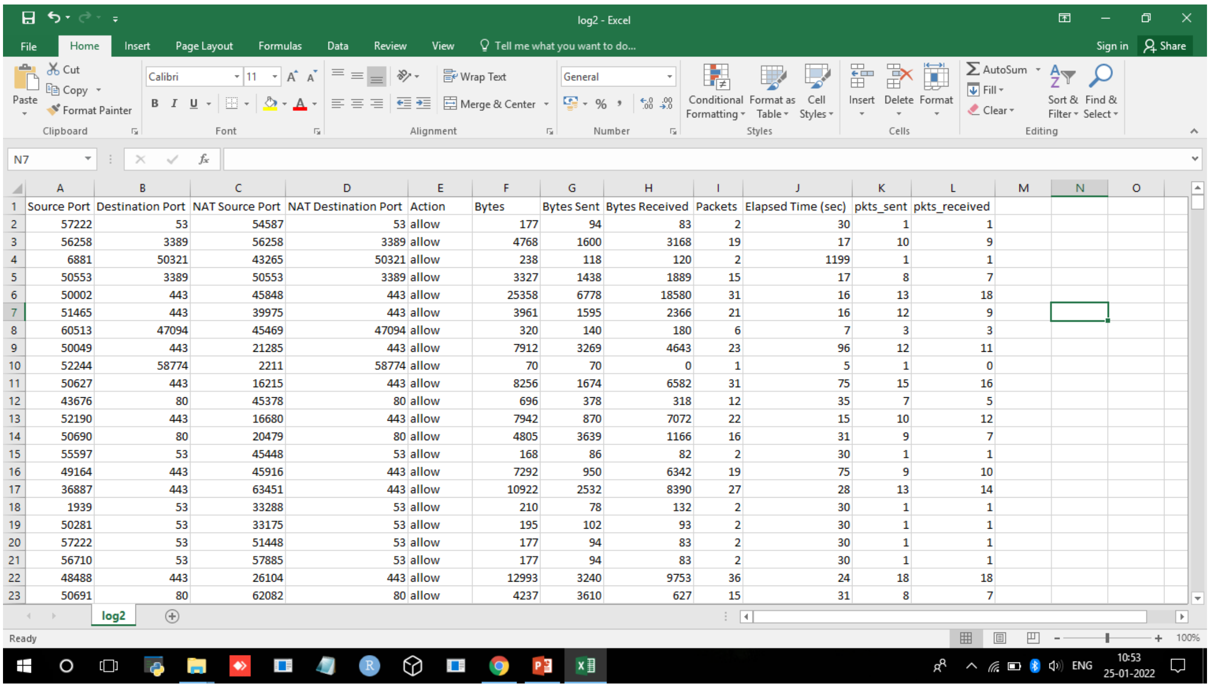
The above figure 3.1 represents the overall flow of the project, The Methodology starts with Data Acquisition followed by different pre-processing techniques and main features to be selected using feature selection using weka tool filter methods, then to build a framework using supervised machine learning algorithms

4. RESULTS AND DISCUSSION

4.1 PHASE 1 - DATA ACQUISITION

The process of acquiring data from relevant sources before it is saved, cleaned, pre-processed, and used in other processes is referred to as "data acquisition." It is the process of acquiring critical business information, converting it into the proper business form, and loading it into the relevant system.

PREVIEW OF A DATASET



Source Port	Destination Port	NAT Source Port	NAT Destination Port	Action	Bytes	Bytes Sent	Bytes Received	Packets	Elapsed Time (sec)	pkts_sent	pkts_received
57222	53	54587	53	allow	177	94	83	2	30	1	1
56258	3389	56258	3389	allow	4768	1600	3168	19	17	10	9
6881	50321	43265	50321	allow	238	118	120	2	1199	1	1
50553	3389	50553	3389	allow	3327	1438	1889	15	17	8	7
50002	443	45848	443	allow	25358	6778	18580	31	16	13	18
51465	443	39975	443	allow	3961	1595	2366	21	16	12	9
60513	47094	45469	47094	allow	320	140	180	6	7	3	3
50049	443	21285	443	allow	7912	3269	4643	23	96	12	11
52244	58774	2211	58774	allow	70	70	0	1	5	1	0
50627	443	16215	443	allow	8256	1674	6582	31	75	15	16
43676	80	45378	80	allow	696	378	318	12	35	7	5
52190	443	16680	443	allow	7942	870	7072	22	15	10	12
50690	80	20479	80	allow	4805	3639	1166	16	31	9	7
55597	53	45448	53	allow	168	86	82	2	30	1	1
49164	443	45916	443	allow	7292	950	6342	19	75	9	10
36887	443	63451	443	allow	10922	2532	8390	27	28	13	14
1939	53	33288	53	allow	210	78	132	2	30	1	1
50281	53	33175	53	allow	195	102	93	2	30	1	1
57222	53	51448	53	allow	177	94	83	2	30	1	1
56710	53	57885	53	allow	177	94	83	2	30	1	1
48488	443	26104	443	allow	12993	3240	9753	36	24	18	18
50691	80	62082	80	allow	4237	3610	627	15	31	8	7

FIG. 4.1 DATASET PREVIEW

The above figure represents the preview of a dataset which includes all the twelve attributes. There are 65533 records and 12 features in total. The Class is ‘Action feature’. So, there are 4 classes in total. They are allow, action, drop and reset-both classes.

4.2 PHASE 2 - DATA PRE-PROCESSING

Data preparation is a major process in Machine Learning, which improves data integrity and makes it easier to extract useful cognizance from the dataset. The first stage in generating a analytics paradigm is preparing the data.

4.2.1 Label Encoding

The process of converting labels into numeric format so that machineries can read them is known as labelling encoding.

FIG. 4.2.1 LABEL ENCODED FOR THE CLASS - ACTION

The above figure represents that the label has been encoded, for the Action class based on the firewall rules the packets decides to pass it or not.

4.2.2 Train and Test Split

The training set is a segment of a dataset that is used to train a machine learning model. A test set, on the other hand, is a subset of the dataset used to evaluate the machine learning model. The ML model uses the test set to predict outcomes.

```
In [9]: # Divide that data into train and test split
        from sklearn.model_selection import train_test_split

        # Splitting the dataset into dependant and independant feature

        X = df.drop(["Action"],axis =1)
        y = df["Action"]

        # Splitting the dataset into train and test sets: 80-20 split

        from sklearn.model_selection import train_test_split

        X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 42)
        X_train.shape, y_train.shape, X_test.shape, y_test.shape

Out[9]: ((52425, 11), (52425,), (13107, 11), (13107,))
```

FIG. 4.2.2 TRAIN AND TEST SPLIT

The figure 4.2.2 represents the training and testing data ratio to use for building a model.

4.2.3 Standardization

Feature scaling is required by machine learning approaches that determine data distances. Standardization is used here.

```
[[ 0.00410758 -0.54874891 -0.65779368 ...  0.03860065 -0.008746
  -0.02077207]
 [ 0.75810782 -0.54864056 -0.87840622 ... -0.2140437  -0.01182699
  -0.02577167]
 [ 0.00410758 -0.56987603  1.35874068 ... -0.11687279 -0.01126681
  -0.02452177]
 ...
 [-0.88185253 -0.54874891 -0.5945663  ... -0.15898018 -0.00790573
  -0.02077207]
 [ 0.48389612 -0.54874891  0.21012511 ... -0.16545824 -0.00902609
  -0.02202197]
 [ 0.46672101 -0.56987603  2.08230243 ... -0.11363376 -0.01182699
  -0.02535504]]
[[ 0.12708658  2.90223129 -0.87840622 ... -0.2140437  -0.01182699
  -0.02577167]
 [ 0.03767115 -0.54874891 -0.55015226 ...  0.07422998 -0.00958627
  -0.02202197]
 [ 0.96335688 -0.56987603  0.35808083 ... -0.11687279 -0.01182699
  -0.02535504]
```

FIG. 4.2.3 FEATURE SCALING USING STANDARDIZATION

The above figure represents, Feature scaling using the method - Standardization.

4.2.4 Exploratory Data Analysis

EDA provides support including, improving data comprehension, recognizing different patterns in data and clarifying the problem statement.

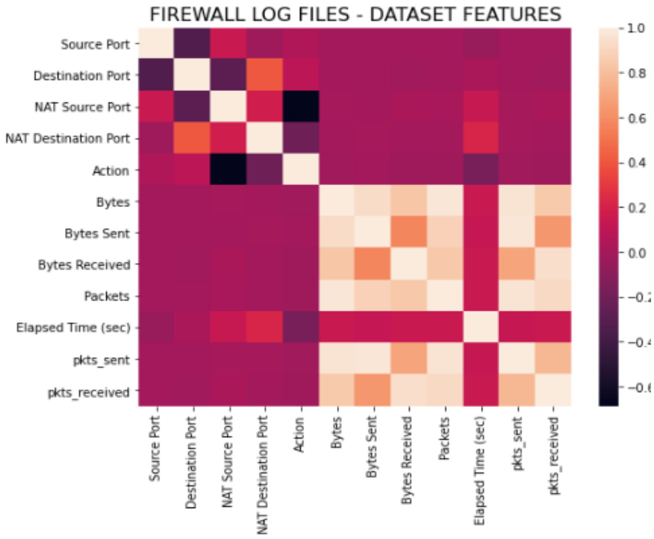


FIG. 4.2.4 HEAT MAP FOR ALL FEATURES

The above figure represents the heat map which all the 12 features involved in the dataset.

4.3 PHASE 3 - FEATURE SELECTION USING WEKA TOOL (FILTER)

FEATURE SELECTION USING WEKA TOOL

WEKA has an automated feature selection tool. There are various techniques present in weka tool. From that, Search method and Attribute Evaluator used in this project are:

- **Ranker +InfoGainAttributeEval**
- **Ranker +CorrelationAttributeEval**

The Ranker

Individual evaluations are used to rank attributes. When combined with attribute evaluators, it's a strong option (ReliefF, GainRatio, Entropy etc).

- **InfoGainAttributeEval**

Measures the information gain with respect to the class to determine the value of an attribute.

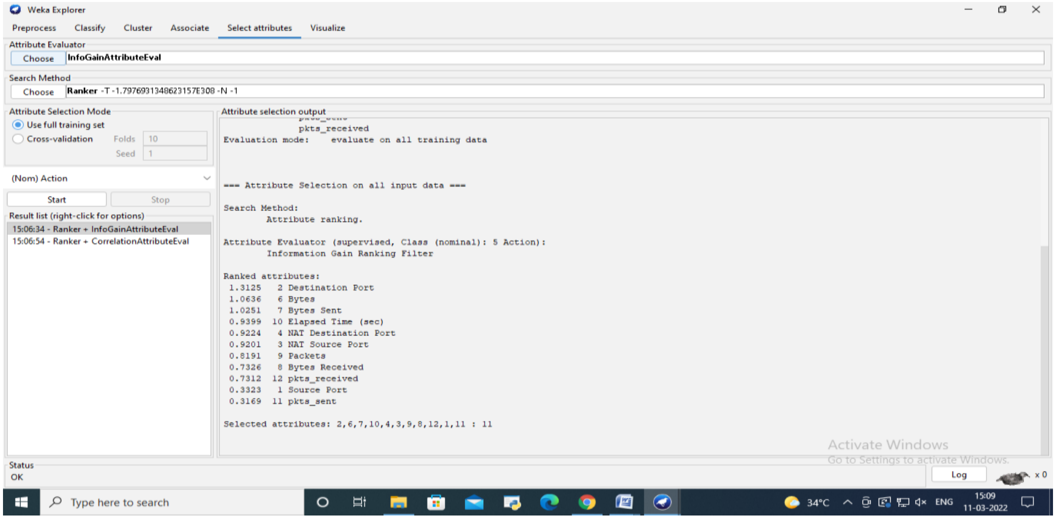


FIG. 4.3.1 FEATURE SELECTION – InfoGainAttributeEval

The above figure represents the combination of **Ranker +InfoGainAttributeEval** to select the features.

CorrelationAttributeEval

Measures the correlation (Pearson's) between an attribute and the class to determine its value to select a best feature.

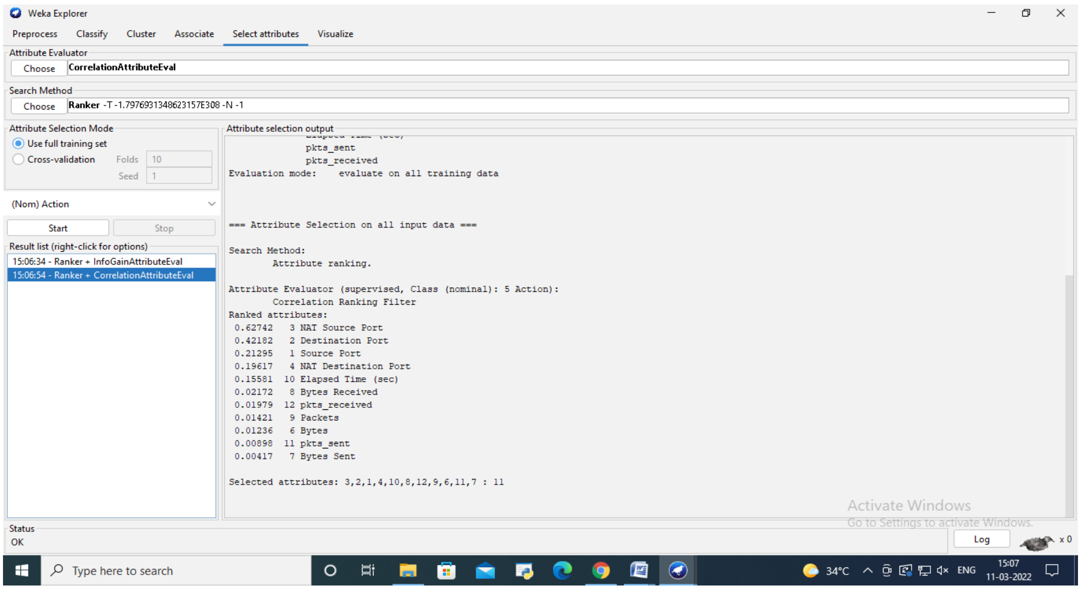


FIG. 4.3.2 FEATURE SELECTION – CorrelationAttributeEval

The features **Source Port, Destination Port, NAT Source Port, NAT Destination Port, and Bytes** were chosen as they provided the best accuracy when compared to other features.

4.4 PHASE 4 - MODEL BUILDING

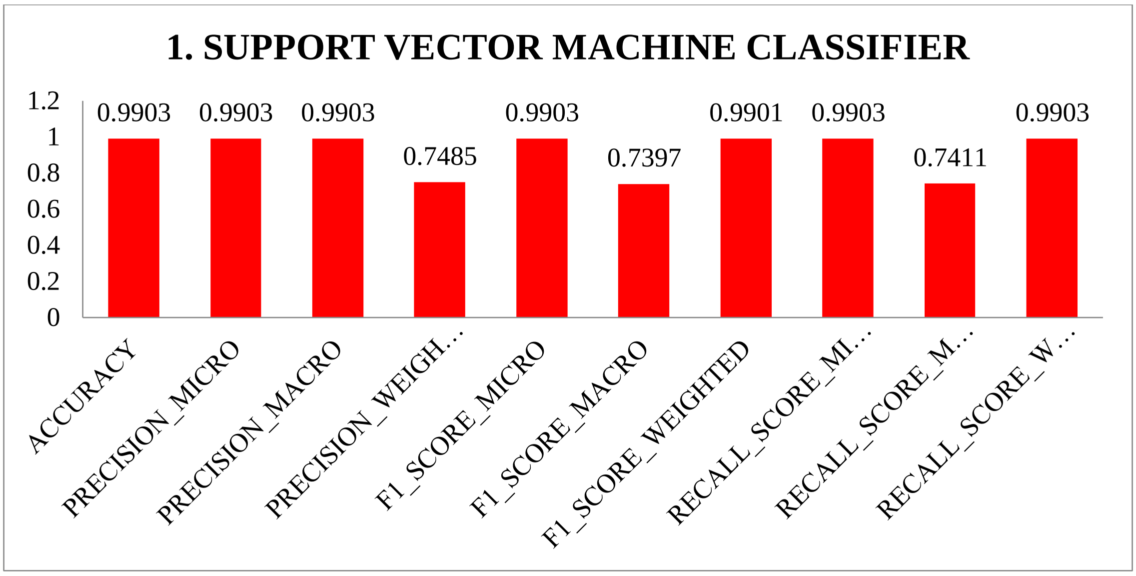


FIG. 4.4.1 SVM CLASSIFIER COMPARISON

The above figure gives the final results of comparison of SVM based on their accuracy, precision micro, macro and weighted, f1-score micro, macro and weighted, recall score micro, macro and weighted.

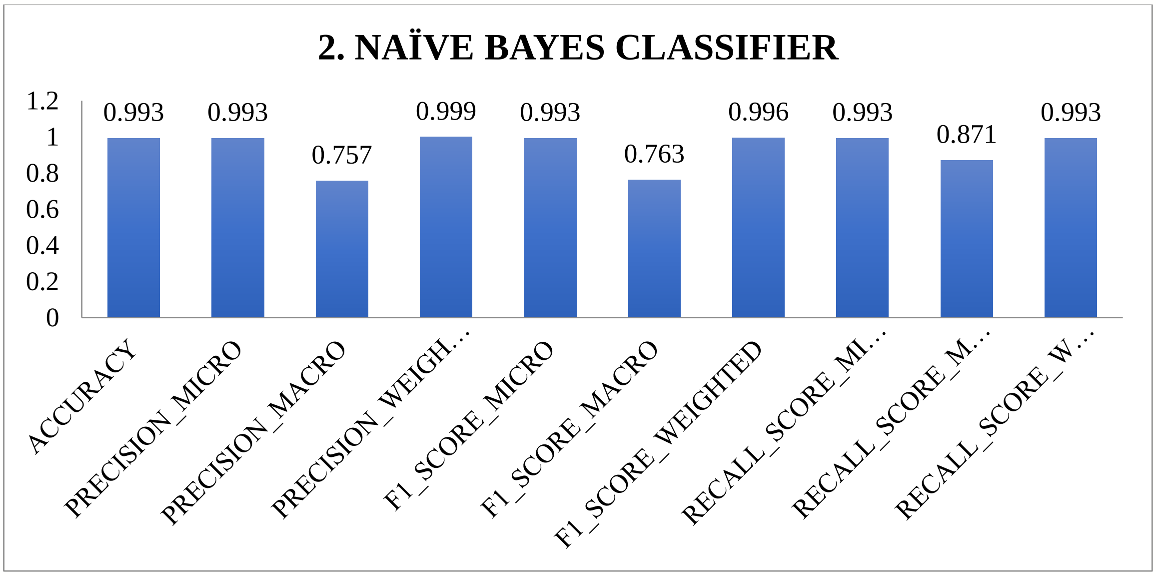


FIG. 4.4.2 NAÏVE BAYES CLASSIFIER COMPARISON

The above figure gives the final results of comparison of Naïve Bayes Classifier based on their accuracy, precision micro, macro and weighted, f1-score micro, macro and weighted, recall score micro, macro and weighted.

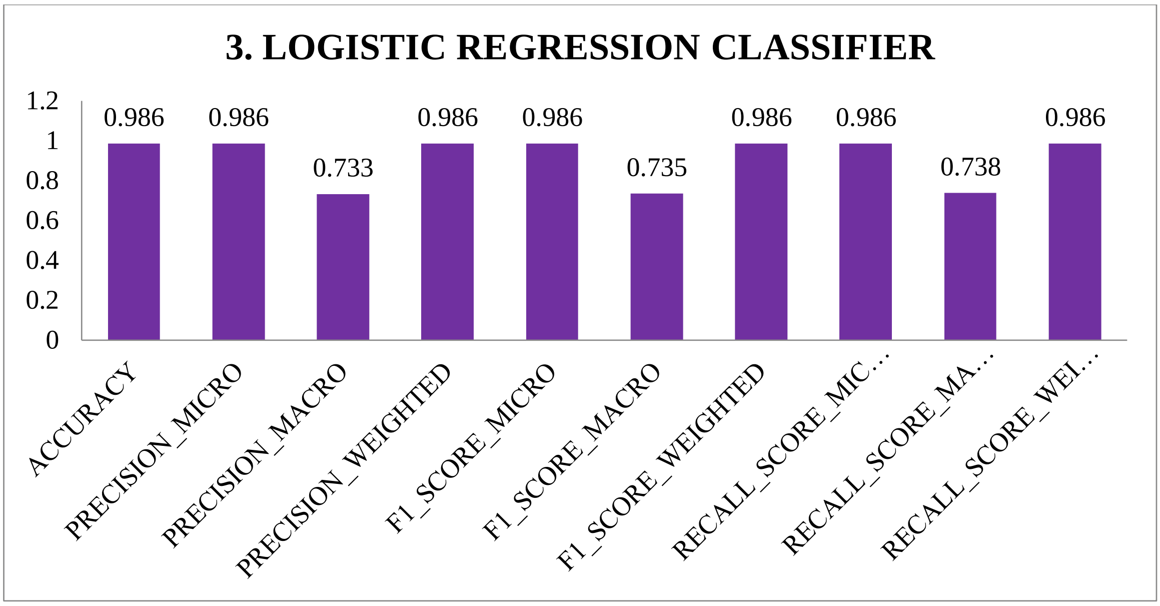


FIG. 4.4.3 LOGISTIC REGRESSION CLASSIFIER COMPARISON

The above figure gives the final results of comparison of Logistic Regression Classifier based on their accuracy, precision micro, macro and weighted, f1-score micro, macro and weighted, recall score micro, macro and weighted.

FIG. 4.4.4 KNN CLASSIFIER COMPARISON

The above figure gives the final results of comparison of KNN Classifier based on their accuracy, precision micro, macro and weighted, f1-score micro, macro and weighted, recall score micro, macro and weighted.

PERFORMANCE EVALUATION

TRAIN ACCURACY COMPARISON

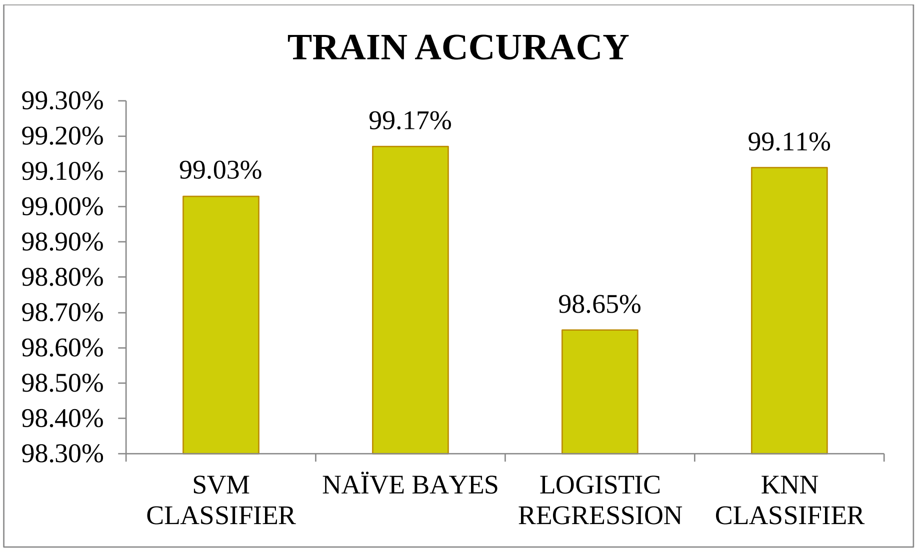


FIG. 4.4.5 TRAIN ACCURACY COMPARISON

The above figure gives the results of performance evaluation of SVM, Naïve Bayes, Logistic Regression and KNN Classifier based on their train accuracy.

PRECISION ACCURACY MICRO – COMPARISON

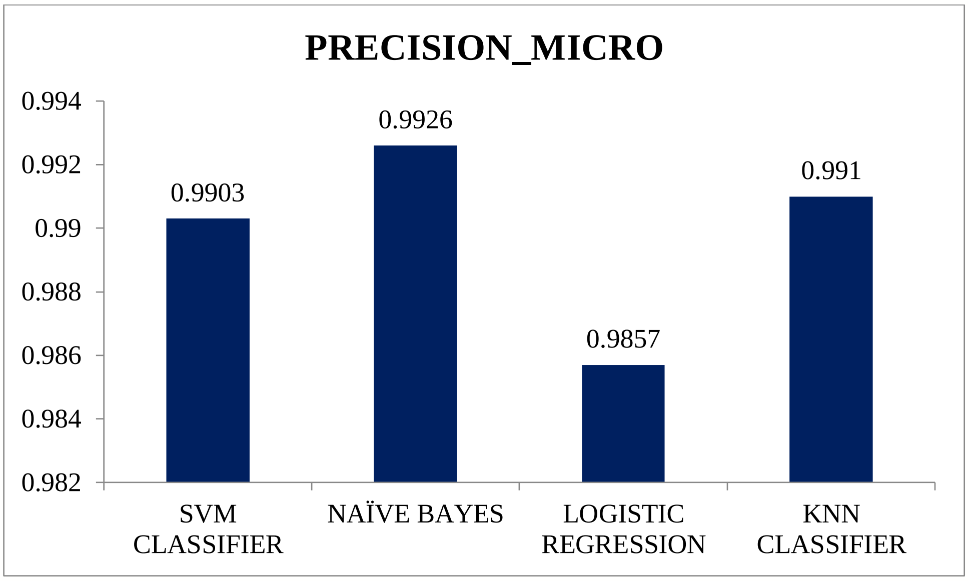


FIG. 4.4.6 PRECISION ACCURACY MICRO – COMPARISON

The above figure gives the results of performance evaluation of SVM, Naïve Bayes, Logistic Regression and KNN Classifier based on their Precision Micro.

PRECISION ACCURACY MACRO – COMPARISON

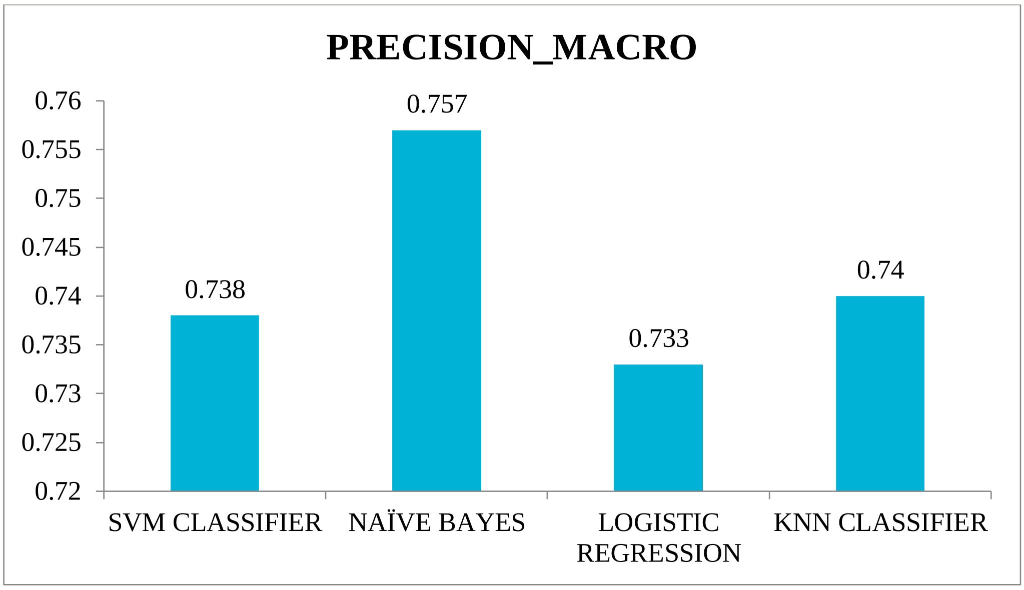


FIG. 4.4.7 PRECISION ACCURACY MACRO – COMPARISON

The above figure gives the results of performance evaluation of SVM, Naïve Bayes, Logistic Regression and KNN Classifier based on their Precision Macro.

PRECISION ACCURACY WEIGHTED – COMPARISON

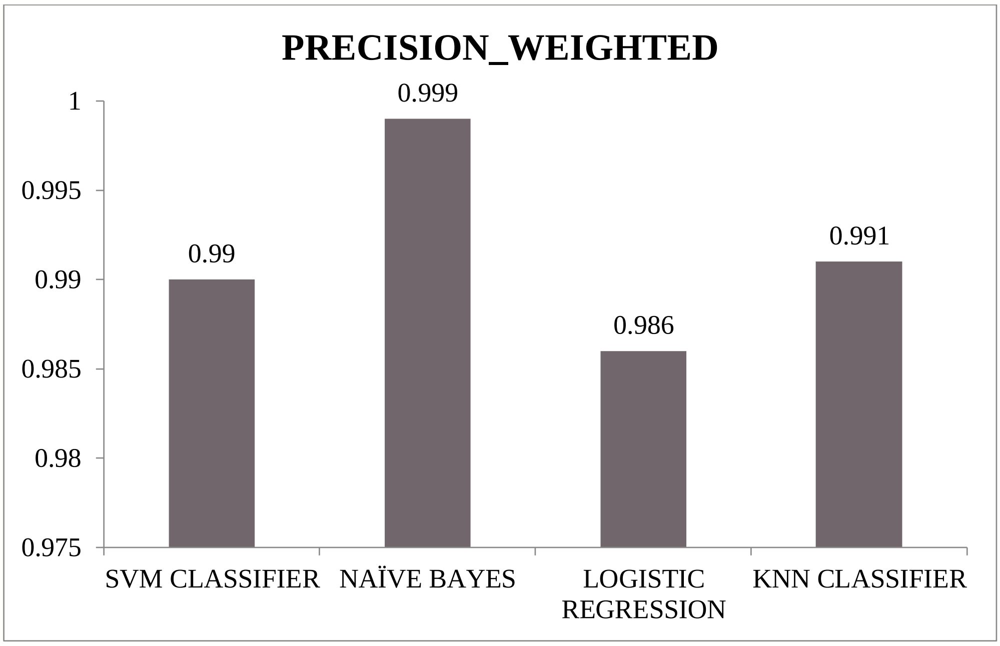


FIG. 4.4.8 PRECISION ACCURACY WEIGHTED – COMPARISON

The above figure gives the results of performance evaluation of SVM, Naïve Bayes, Logistic Regression and KNN Classifier based on their Precision Weigjted.

F1 SCORE ACCURACY MICRO – COMPARISON

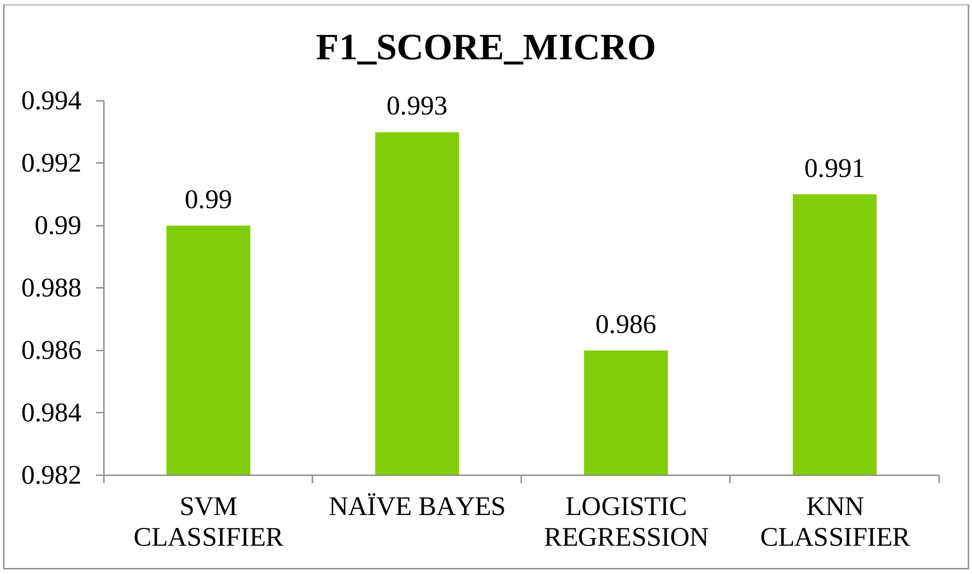


FIG. 4.4.9 F1 SCORE ACCURACY MICRO – COMPARISON

The above figure gives the results of performance evaluation of SVM, Naïve Bayes, Logistic Regression and KNN Classifier based on their f1 score Micro.

F1 SCORE ACCURACY MACRO – COMPARISON

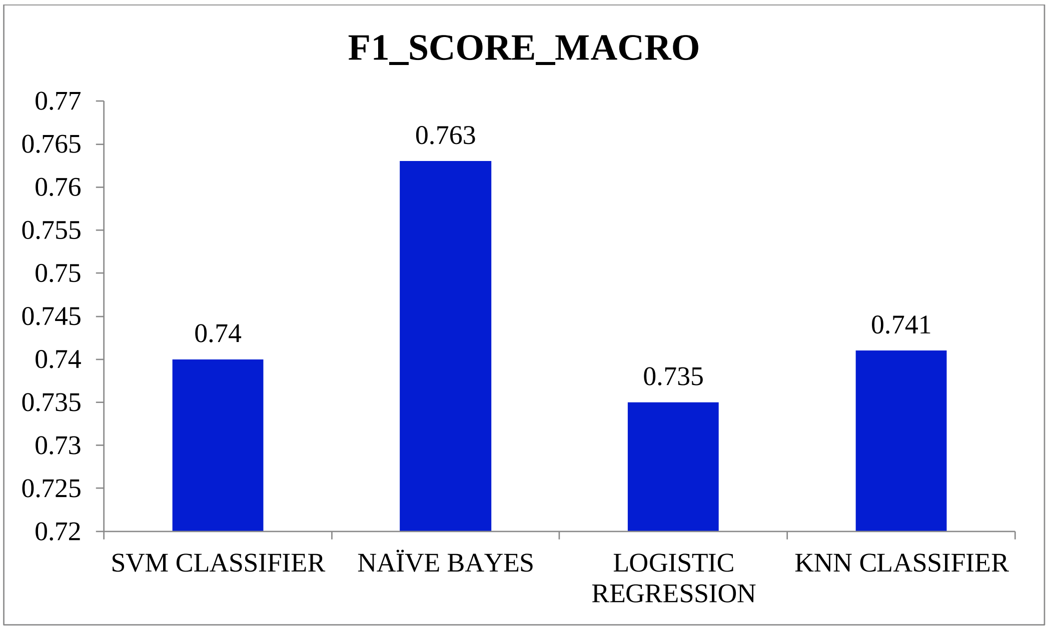


FIG. 4.4.10 F1 SCORE ACCURACY MACRO – COMPARISON

The above figure gives the results of performance evaluation of SVM, Naïve Bayes, Logistic Regression and KNN Classifier based on their f1 score Macro.

F1 SCORE ACCURACY WEIGHTED – COMPARISON

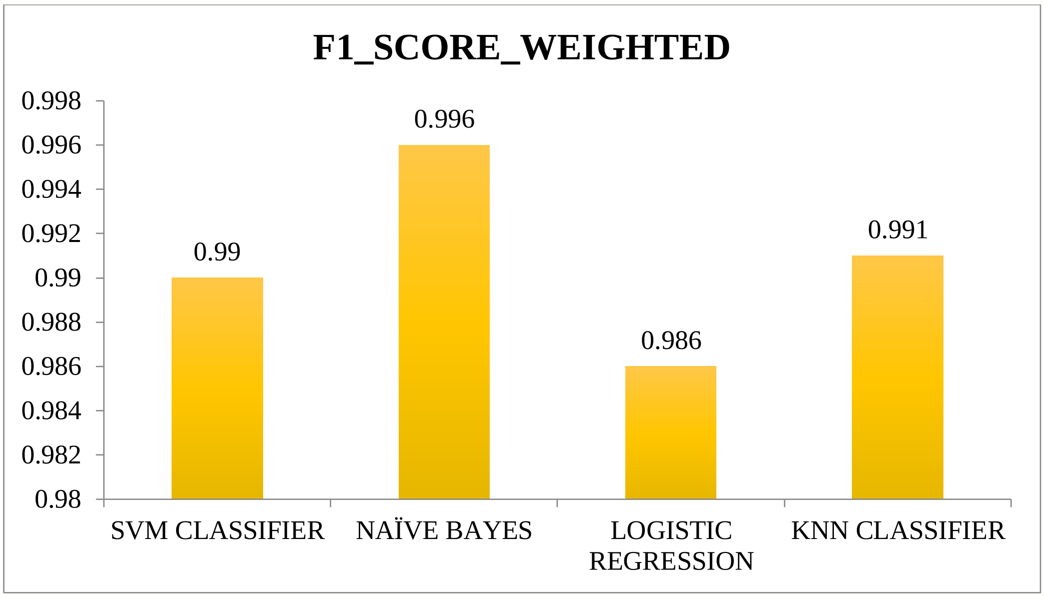


FIG. 4.4.11 F1 SCORE ACCURACY WEIGHTED – COMPARISON

The above figure gives the results of performance evaluation of SVM, Naïve Bayes, Logistic Regression and KNN Classifier based on their f1 score Weighted.

RECALL SCORE ACCURACY MICRO – COMPARISON

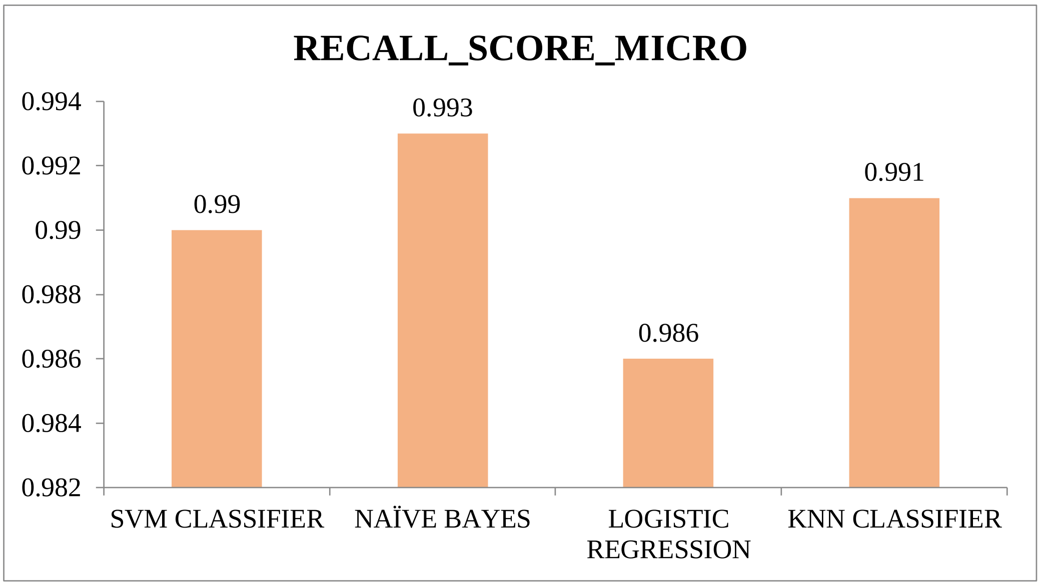


FIG. 4.4.12 RECALL SCORE ACCURACY MICRO – COMPARISON

The above figure gives the results of performance evaluation of SVM, Naïve Bayes, Logistic Regression and KNN Classifier based on their Recall score Micro.

RECALL SCORE ACCURACY MACRO – COMPARISON

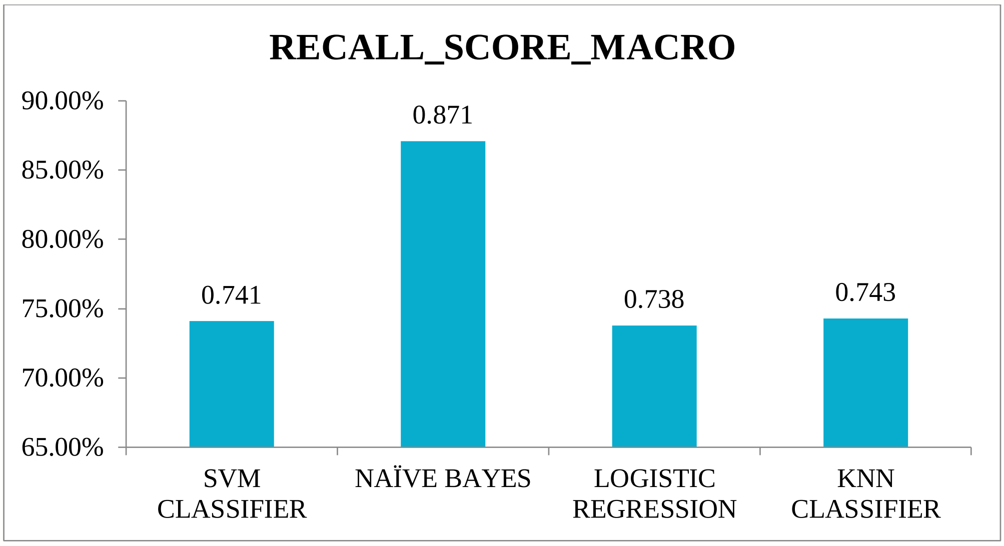


FIG. 4.4.13 RECALL SCORE ACCURACY MACRO – COMPARISON

The above figure gives the results of performance evaluation of SVM, Naïve Bayes, Logistic Regression and KNN Classifier based on their Recall score Macro.

RECALL SCORE ACCURACY WEIGHTED – COMPARISON

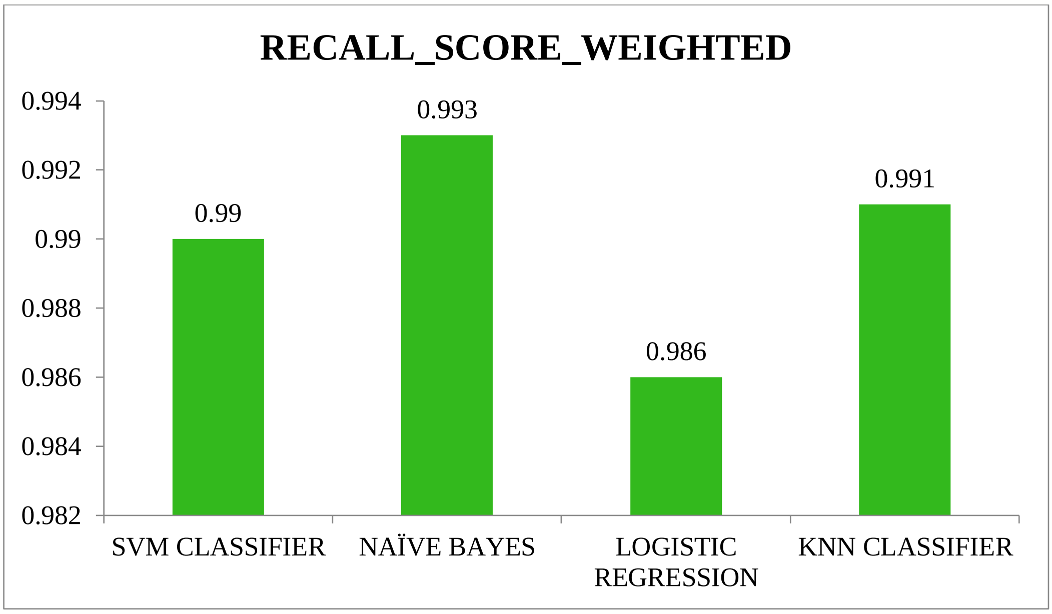


FIG. 4.4.14 RECALL SCORE ACCURACY WEIGHTED – COMPARISON

The above figure gives the results of performance evaluation of SVM, Naïve Bayes, Logistic Regression and KNN Classifier based on their Recall score Weighted.

4.5 PHASE 5 - COMPARITIVE ANALYSIS OF ALGORITHMS

The Naive Bayes method performed better than the other classification methods such as SVM, Logistic Regression and KNN Classifier in the model. With 99.26% accuracy, the Naive Bayes classifier was found to have the highest Accuracy value.

MODEL	ACCURACY
SVM CLASSIFIER	99.03%
NAÏVE BAYES	**99.26%**
LOGISTIC REGRESSION	98.70%
KNN CLASSIFIER	99.10%

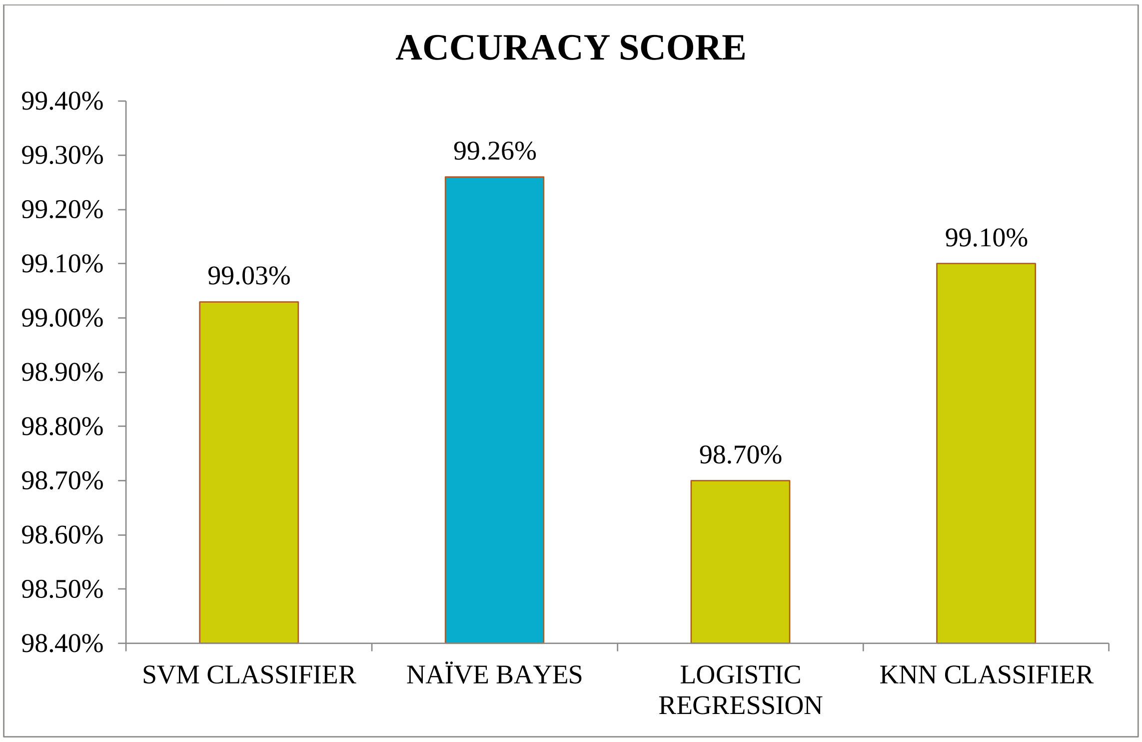


FIG. 4.5.1 COMPARITIVE ANALYSIS OF ALGORITHMS

The above figure gives the final results of comparison of different algorithms, based on their overall accuracy level.

5. CONCLUSION

The firewall is the most crucial elements of a network, and there should be no contradiction in the security policies employed, because to do so would expose the network to security risks. So, all the people should be aware of the risks employed in all the components. Here, considered only the 5 distinct features: Action (Allow, Deny, Drop, Reset-Both) Source Port, Destination Port, NAT Source Port, NAT Destination Port, Bytes. The Naive Bayes method performed well. With 99.26% accuracy, the Naive Bayes classifier was found to have the highest Accuracy value. Further the model can be developed using other different algorithms which can give more accuracy in terms of selected features.

6. REFERENCES

[1] AL-Behadili, H. (2021). Decision Tree for Multiclass Classification of Firewall Access. *International Journal of Intelligent Engineering and Systems*, *14*(3), 294–302. https://doi.org/10.22266/ijies2021.0630.25

[2] Allagi, S., & Rachh, R. (2019). Analysis of Network log data using Machine Learning. *2019 IEEE 5th International Conference for Convergence in Technology (I2CT)*. https://doi.org/10.1109/i2ct45611.2019.9033737

[3] As-Suhbani, H. E., & Khamitkar, S. (2019). Classification of Firewall Logs Using Supervised Machine Learning Algorithms. *International Journal of Computer Sciences and Engineering*, *7*(8), 301–304. https://doi.org/10.26438/ijcse/v7i8.301304

[4] Ertam, F., & Kaya, M. (2018). Classification of firewall log files with multiclass support vector machine. *2018 6th International Symposium on Digital Forensic and Security (ISDFS)*. https://doi.org/10.1109/isdfs.2018.8355382

[5] Hommes, S., State, R., & Engel, T. (2012). A distance-based method to detect anomalous attributes in log files. *2012 IEEE Network Operations and Management Symposium*. https://doi.org/10.1109/noms.2012.6211940

[6] Kamiya, K., Aoki, K., Nakata, K., Sato, T., Kurakami, H., &Tanikawa, M. (2015). The method of detecting malware-infected hosts analyzing firewall and proxy logs. *2015 10th Asia-Pacific Symposium on Information and Telecommunication Technologies (APSITT)*. https://doi.org/10.1109/apsitt.2015.7217113

[7] Kowalski, K., & Beheshti, M. (2006). Analysis of Log Files Intersections for Security Enhancement. *Third International Conference on Information Technology: New Generations (ITNG'06)*.https://doi.org/10.1109/itng.2006.32

[8] Nimbalkar, P., Mulwad, V., Puranik, N., Joshi, A., &Finin, T. (2016). Semantic Interpretation of Structured Log Files. *2016 IEEE 17th International Conference on Information Reuse and Integration (IRI)*. https://doi.org/10.1109/iri.2016.81

[9] Sharma, D., Wason, V., &Johri, P. (2021). Optimized Classification of Firewall Log Data using Heterogeneous Ensemble Techniques. *2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*. https://doi.org/10.1109/icacite51222.2021.9404732

[10] Winding, R., Wright, T., &Chapple, M. (2006). System Anomaly Detection: Mining Firewall Logs. *2006 Securecomm and Workshops*. https://doi.org/10.1109/seccomw.2006.359572

WEBSITE REFERENCES

- https://www.researchgate.net/publication/335826729_Classification_of_Firewall_Logs_Using_Supervised_Machine_Learning_Algorithms
- https://www.exabeam.com/siem-guide/siem-concepts/firewall-logs
- https://medium.com/mlearning-ai/routing-network-traffic-based-on-firewall-logs-using-machine-learning-dc5e5c8c6bb3
- http://www.ijstr.org/final-print/feb2020/Discovering-Anomalous-Rules-In-Firewall-Logs-Using-Data-Mining-And-Machine-Learning-Classifiers.pdf
- https://www.kdnuggets.com/2017/02/machine-learning-driven-firewall.html
- https://towardsdatascience.com/how-data-science-could-make-cybersecurity-troubleshooting-easier-firewall-logs-analysis-591e4832f7e6
- https://www.loganalysis.org/firewall-logging/
- https://repositorio-aberto.up.pt/bitstream/10216/128588/2/412447.pdf