Intrusion Detection Using Machine Learning Algorithms

Adithya R, Advaith AJ, G Sabarinath, Tharun S Kumar
Computer Science and Artificial Intelligence Engineering
Amrita Vishwa Vidhyapeetham, Amritapuri
amenu4aie20005@am.students.amrita.edu, amenu4aie20006@am.students.amrita.edu, amenu4aie20026@am.students.amrita.edu, amenu4aie20071@am.students.amrita.edu

***Abstract*— Since the beginning of the 21st century, the world has witnessed rapid growth in almost all the sectors like agriculture, industry, transportation etc. Computer and internet usage is also on the rise, as they play a crucial part in the development of all other industries. By the starting of last decade, every one of us also started using the mobile phone internet and all for our day. And by the end of the decade, the commencement of covid, made us totally dependent on the internet for various things. We now live in a technologically advanced world. This has also led to cyber network attacks, and a lot of us are becoming victims of this. This is when an Intrusion Detection System, which can provide additional security, comes into play. Various ML techniques have gained a good interest in these tasks. In this research paper, our aim is to implement some of the ML techniques like SVM, KNN, K-means in the KDDCup99 Dataset and find out their performances.**

Keywords*— Intrusion Detection, Binary class, Multiclass, k-means, KNN, SVM, kddcup99

I. Introduction

During the current situation of Covid-19, the usage of the internet and social media are hiked to a large extend. With everything becoming digital including works, education system, shopping, social media, entertainment etc., the cases of network attacks are also increased a lot. Thousands of hackers emerge and daily millions of cases are reported in various parts of the world. Users' privacy is getting disclosed and much of their amounts of money are being robbed and even many are facing life-threatening situations. Therefore, it has become very necessary that to provide a very strong, reliable, and less complex security system that can defend against these types of Cyber-attacks and viruses.

For Defending against various attacks many types of computer security measures have been researched and introduced in the past few years, which include various types of antivirus software, windows firewall, cryptography and IDS (Intrusion Detection Systems). An IDS is basically a system or software or a tool that can identify the presence of intrusions and attacks in a well-protected system. Two types of Intrusion Detection Systems are in general. The First one is the Signature-based IDS which will create a predefined model based on the currently known attack types. The drawback of this is that the system will fail in identifying the new types of attacks that are previously unknown to it. The Second type of IDS is anomaly-based, this understands the behaviour of normal packets and compares them with the newly coming packets. If there is any violation of fluctuation found it will report as an intrusion. Hence this method can effectively be used for the detection of unknown types of attacks. But the drawback in this method is that even if the newly coming packet is not an attack and if it has some difference in the behaviour, then also the system will classify it as an attack, which means there are huge chances for false-positive detections.

In this research, we study the performance of such ML techniques such as SVM (Support Vector Machine), KNN (K Nearest Neighbor) and the K-means Clustering, KDDCup99 is the dataset we choose for this research. Various preprocessing techniques like data filtering, Interpolation, normalization, label encoder, PCA (principal component analysis) have been used to make the dataset clean and efficient for our machine learning models. KNN works by computing the distances between a query and all of the data's examples, selecting the K examples that are closest to the query, and then voting for the most frequent label. The SVM algorithm divides the data into classes by drawing a line or hyperplane. The K-means clustering method aims to generate clusters out of comparable elements. The number of groups is denoted by the letter K. The main objective of this research is to find the performance analysis of the three machine learning models (KNN, SVM, K-means) in Intrusion Detection.

II. Literature Review

2.1 Research on Intrusion Detection Based on Improved Combination Of K-Means and Multi-level SVM

To increase detection efficiency, Zhang Xiaofeng, et al., 2017 designed an approach that first uses improved K-means to divide the data to be detected into distinct clusters and mark it as normal or abnormal, and then uses multi-level SVM to categorize the clusters designated as abnormal. The second technique works by selecting only one data at a time, calculating the nearest cluster using Euclidean distance, and moving the cluster in the data's direction, successfully solving the problem of the algorithm being in the local optimum. The clusters are then transferred to an SVM to achieve the best optimum results after processing the dataset through the modified K-means algorithm.

2.2 Intrusion Detection System using Support Vector Machine

J. Jha, Leena Ragha, and colleagues (2013) used the NSL-KDD Dataset to undertake a study that combined the Information Gain Ratio (IGR) and the K-mean algorithm to SVM for intrusion detection. Using a support vector machine, this study gives a complete methodology for selecting the optimal set of NSL-KDD dataset features that efficiently define normal traffic and distinguish it from aberrant traffic. This dataset has 41 features, with the last column referring to the network attack. Due to its good generalization nature and ability to overcome the curse of dimensionality, the Support Vector Machine is a prominent tool for anomaly detection. It can also give real-time detection capability. Mapping string values to numerical values, scaling attribute values between [-1,1], and mapping 0 to normal data and 1 to anomaly data are all used in the dataset preparation. This study employs a hybrid feature selection strategy that incorporates filter and wrapper models. Using the Information Gain Ratio (IGR) measure, features are sorted in decreasing order. The reason for choosing a smaller dataset is to lower the amount of time it takes to train the SVM classifier.

2.3 Support Vector Machine for Network Intrusion and Cyber-Attack Detection

K. Ghanem, F. J. Aparicio-Navarro, et al., 2017 compared the performance of their anomaly-based Intrusion Detection System (IDS) with the popular machine learning algorithm Support Vector Machine (SVM) in order to see if their IDS could be supplemented with the SVM-based IDS to improve the system's overall accuracy. Adding a second line of defense could help to eliminate false alerts. Because of its strong generalization and ability to overcome the challenge of dimensionality, SVM has become the most prominent Machine Learning (ML) technique used for intrusion detection. The performance of one-class and two-class SVMs is compared in this research utilizing linear and non-linear Radial Basis Function (RBF) forms. An SVM's purpose is to generate a hyperplane that optimizes the training data's margin while minimizing the complexity and risk of overfitting. This assessment analysis sheds light on the SVM algorithms' detecting capabilities. The results show that linear two-class SVM and linear one-class SVM outperform linear two-class SVM. As a result, the adoption of two-class and one-class linear SVM could potentially enhance the performance of their anomaly-based IDS.

2.4 Anomaly Based Intrusion Detection in Industrial Data with SVM

There are two types of intrusion detection: signature-based and anomaly-based. Prior knowledge of signatures is required. Anomaly-based establishes a profile based on usual activity, and any deviation is recorded as an intrusion. The goal of a support vector machine is to establish a hyperplane between data sets to indicate which class they belong to, and then train the machine to understand data structure and map it to the appropriate class label. Using the one-against-one approach, multi-class SVM is also an option for intrusion detection. The feature extracting module of this one-class SVM-based detection model accepts raw data as input and extracts predicted features to generate formatted data. Detection rate, precision, recall, and F-value are often used measures for evaluating performance. When compared to PNN and C-SVM, one class SVM gets greater detection rates and averages better performance in terms of accuracy recall and F value.

III. Dataset

Our research's main objective is in selecting the best features from the KDDCup99 dataset that can highly contribute to the identification of attacks, in other words, that can help in identifying whether a network is attacked or not. The KDD Cup 1999 is the data set we use for this project, which contains 41 features that help us in the identification process. Some of the important features are attack types and the protocol that each network use. Some of the features are of zero values and they will not contribute much to our research. The dataset is primarily classified binary class which is nothing but which tells us if a network is attacked to no. Secondly, it is also classified as Multiclass, which give us information like out of five types i.e., Denial of Service (DOS), Probe, Remote to Local(R2L), User to Root(U2R) and Normal (Not attacked) what type is a particular network is about. Out of these 41 features, 7 of them are symbolic. These symbolic features should be encoded into numerical values in order to make it easy for our ML model to work with. So, we use a label encoder. The dates should also be scaled to a proper range because of the varying resolution that they come in. The scaled data should be within a range of [0,1]. Out of our total dataset, 9711 (43.08%) of data are normal. And the remaining 12,833(56.92%) were all attacked. Out of the 12,833 attacked these are further classified into four types of attacks. 7417(32.90%) are of DOS type attacks, 2928(12.99%) are of R2L type attacks, 2421(10.74%) are of Probe type attacks and the remaining 67(0.30%) are of U2R type attacks. Various preprocessing techniques have been done before modelling with our ML algorithms in order to improve the accuracy of predictions.

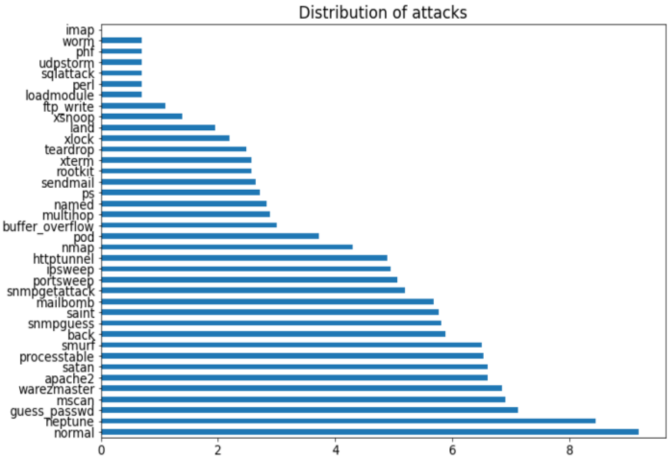


Fig. 1. Distribution of different types of attacks

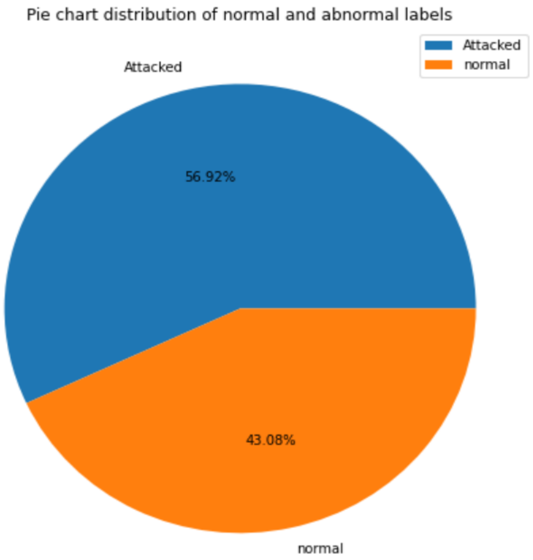


Fig. 2. Distribution of binary class target values

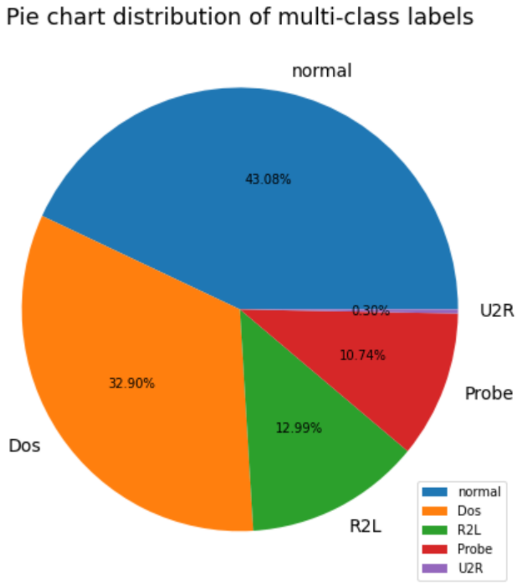


Fig. 3. Distribution of multiclass target values

IV. Methodology

Before giving the dataset to the ML algorithms, we need to preprocess them for getting the best results. The preprocessing is done to remove the unwanted or insignificant columns, fill the missing values in the dataset using interpolation, and then Normalize our dataset. Then we classified our dataset into two categories namely the binary class and the multiclass, in binary class classification the attacks were encoded to 0 and normal (not attacked) were encoded to 1. In multi-class classification, the attacks were further classified into four types and encoded to values (0,3,2,4) respectively for (DOS, R2L, Probe, U2R). After this, the Dataset is being split to train and test with a ratio of 80:20. Then feature selection is done using PCA (Principal Component Analysis). The result of PCA gives the independent reduced features known as Principal Components. Then later on the preprocessed dataset is given to Different machine Learning Models namely SVM, KNN and K-means clustering. Then the performance of each algorithm is closely measured using different evaluation parameters like Accuracy, precision, recall and F1-score. The Model that gives the higher values for the parameters is selected as the best model

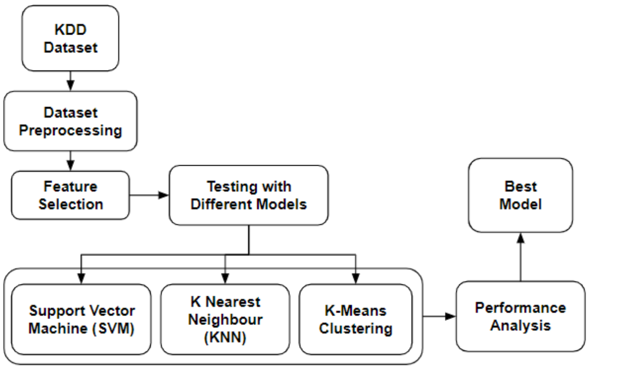


Fig. 4. Block Diagram

A. *Pre-processing Techniques*

Preprocessing is carried out to prepare data in datasets for Machine Learning models. The first step is to filter the data (data filtering), this step is done in order to remove the unnecessary features from the dataset. Some of the features in the dataset contains all zero values only and they will not contribute much to increasing the accuracy of our ML model. Converting categorical values to numerical values is the second stage in Preprocessing. Our machine learning model is not capable of working with the string values. So, we much convert those strings to numerical form in order to make it work. Label Encoder is used to convert the data. The label encoder function searches a feature for distinct strings and assigns numeric values in alphabetical order. The third technique used in preprocessing is an interpolation. This function fills the missing values of the dataset. Lastly, normalization is done in order to scale the data which helps in reducing the computational complexity while handling a large range of values. The min-max scalar is the normalization technique used here that scales down the values to between 0 and 1

Interpolation: $$x_{i+1} = x_i + \frac{x_j - x_i}{j-i} \qquad (1)$$

where x_i and x_j are the previous and next available values, i and j are the indices of them.

Normalization: $$x = \frac{x - x_{min}}{x_{max} - x_{min}} \qquad (2)$$

x is the value to be normalized, x_{max} and x_{min} are the maximum and minimum value of x respectively.

B. *Feature Selection*

After cleaning and filtering the data, the following step is to pick the features that have a higher variance than the target values, reducing the dataset's dimensionality. A feature reduction approach called Principal Component Analysis was used to pick the target feature (PCA).

i. Consider the dataset to be an ‘$n \times m$’ matrix where n is the observations and m is the features in the dataset.

ii. $A[n \times n] = X^T X$ which is called the covariance matrix

iii. The characteristic equation is $A - \lambda I = 0$. By doing $|A - \lambda I|$ we get 'λ' value, substituting in characteristic equation we get eigenvector $[W(m \times m)]$ matrix. (3)

$$W = \begin{bmatrix} \uparrow & \uparrow & & \uparrow \\ w_1 & w_2 & \rightarrow & w_m \\ \downarrow & \downarrow & & \downarrow \end{bmatrix}_{m\times m} \qquad (3)$$

iv. After finding the eigenvector variance across each direction is found.

v. On doing $W \times X$ we get new linearly transformed matrix T.

vi. Dimension of T is n*m, same as the dimension of X, but we want a reduced matrix. Therefore, set a threshold value for the variance, to select a reduced W matrix. (4)

vii. On doing $W \times X$ we get new linearly transformed matrix T with dimension $n \times r$. (5)

$$W = \begin{bmatrix} \uparrow & \uparrow & & \uparrow \\ w_1 & w_2 & \rightarrow & w_m \\ \downarrow & \downarrow & & \downarrow \end{bmatrix}_{m\times m} \qquad (4)$$

$$\underbrace{\qquad\quad}_{w_r}$$

$$\underbrace{T_r}_{n\times r} = \underbrace{X}_{n\times m}\ \underbrace{W_r}_{m\times r} \qquad (5)$$

A total of 26 features were obtained after data filtering and cleaning. Using the train test split function, data was split into 80 percent and 20 percent training and testing percentages, respectively. The data collection is then passed via PCA for feature reduction, with a variance of 0.95 in this project. Following PCA, components that sum up to 0.95 variance are chosen.

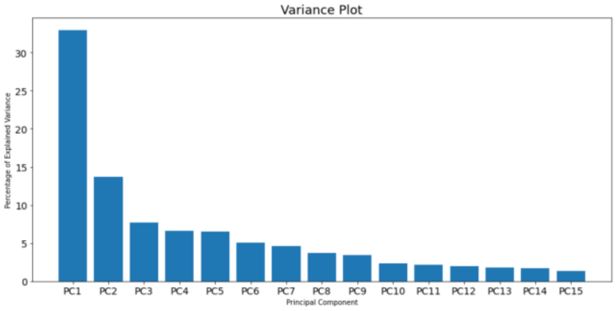


Fig. 5. PCA *Variance* plot

After this the preprocessed Dataset is given to the different machine learning alorithms to work with.

K-Nearest Neighbour (KNN)

KNN is a learning model that is based on instances. The Euclidean distance is used to calculate distance. K is selected by plotting the graph between error rate and K value. The point at which the K value has the minimum error rate will be selected. In our research the value selected for k is 3 since it is having the least error rate.

$$d(p,q) = \sqrt{\textstyle\sum_{i=1}^{n}(q_i - p_i)^2} \qquad (6)$$

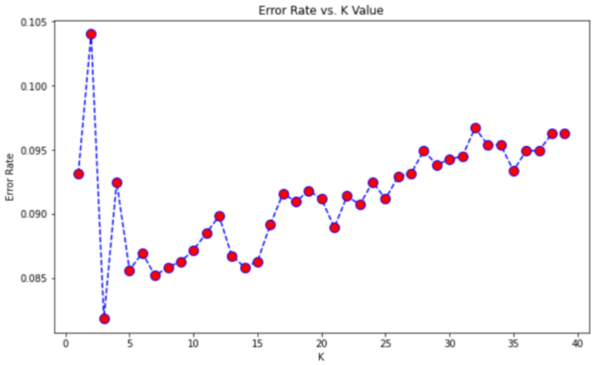


Fig. 6. Error Rate vs k value plot for Binary KNN

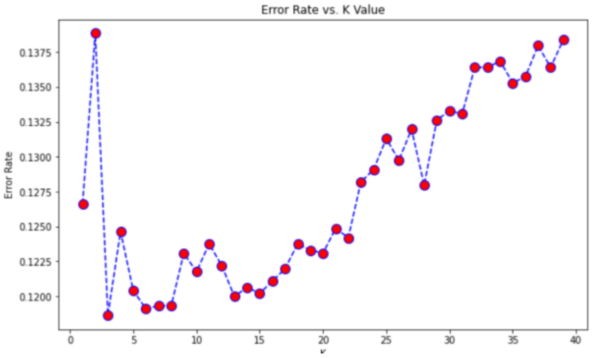


Fig. 7. Error Rate vs k value plot for Multiclass KNN

C. Support Vector Machine (SVM)

SVM identifies the best separating hyperplane that optimizes the training data margin while minimizing complexity. Linear or non-linear SVMs will have to be utilized depending on the data's distinguishability. By projecting the data into a highly dimensional feature space, an SVM that uses kernel functions provides an effective alternative technique for changing the non-linear approach into a linear one (Linear, Polynomial, Radial Basis Function). The data is classified by SVM into the appropriate attack type.

D. K-Means Clustering

Choose how many clusters you want to find in the data. To get the best suited value for k, the elbow function method is used. Calculate the distance between the points and the original clusters, then allocate each point to the clusters that are closest to it. Calculate each cluster's mean and compare the distance between the mean and the data points. Add up the variation inside each cluster to determine the grade of clustering. Change the distinct points you've chosen until you locate the one with the lowest variance sum. Carry out the same procedure with different 'k' values. The plot of 'k' vs. Within-Cluster-Sum-of-Squares (WCSS) was obtained. The sum of squares of the distances between each data point in each cluster and its respective centroids is WCSS. The value of ‘k’ at which the WCSS does not go down quickly is the optimum ‘k’ value.

E. *Evaluation Parameters*

The confusion matrix and evaluation parameters are used to evaluate the classification models SVM and KNN. The precision, recall, and f1-score evaluation parameters are computed from the obtained confusion matrix as shown in Tab. 1.

Precision: It is the ratio of the count of correct predictions of a value to the total prediction of that value.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \qquad (7)$$

Recall: It is the ratio of the count of correct predictions of a value to the actual count of that value.

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \qquad (8)$$

F1 Score: It is the harmonic mean of precision and recall values.

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \qquad (9)$$

V. RESULTS AND DISCUSSION

A. *KNN*

By looking at the train test accuracy graph (Fig. 8) we can clearly see that the train accuracy is getting reduced after k = 3in binary classification. Also, the change in test accuracy is very small when compared to the test accuracy. We obtain the maximum accuracy for the test data when k =3. Similar for the case of the Multiclass also (Fig. 9)

The train and test accuracies obtained for the binary class are 95.59% and 91.67% respectively. Also, for the multi class the obtained train test accuracy is 94.19% and 90.40% respectively. The values for the other evaluation parameters like precision recall and f1-score are given the table1.

In a confusion matrix, the principal diagonal values show the correctly predicted values, while the other diagonal values reveal the incorrectly predicted values.
The count 2165 corresponds to successfully expected zeros, while the count 3002 corresponds to correctly predicted ones. The incorrectly predicted ones and zeros are represented by the counts 297 and 172, respectively. (Fig. 10)

The count 1785, 2214, 570, 526 corresponds to the correctly predicted zero, one, two, three and four target classes. The sum of all the values of the confusion matrix corresponds to the total count of test samples. (Fig. 11)

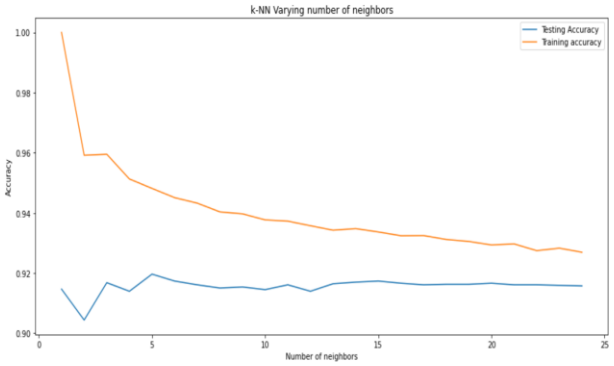


Fig. 8. Train vs Test accuracy for different values of k for binary- class KNN

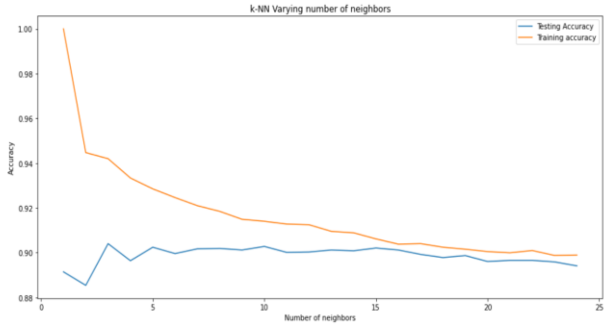


Fig. 9. Train vs Test accuracy for different values of k for Multi- class KNN

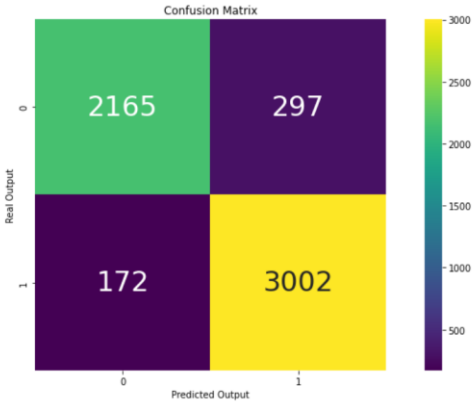


Fig. 10. Confusion matrix of binary class KNN

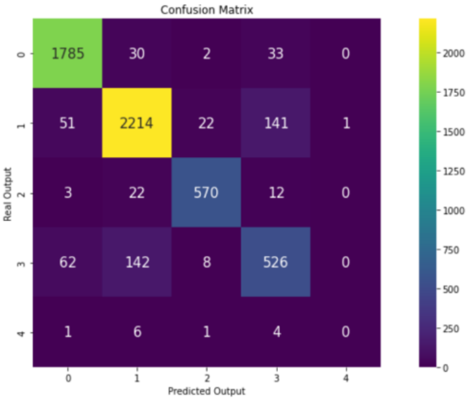


Fig. 11. Confusion matrix of multiclass KNN

B. *SVM*

The model got overfit and incapable of reliably predicting from the test set as the value of gamma was gradually increased up to 100. (100 percent training accuracy and 51 percent testing accuracy). Similarly, increasing the C parameter to 10000 resulted in a model that was 99 percent accurate for test data and 91 percent accurate for training data.

Finally, the values of C and gamma are determined by trial and error to be C=1000 and gamma = 0.1, which produces the most accurate results. Training received 95.91 percent of accuracy, while tests received 93.23 percent of accuracy. Tab. 1 shows the precision recall and f1-values generated from the model. The value 1821 corresponds to the correctly expected zero, while the count 2435 corresponds to the correctly anticipated one in Fig. 12. The incorrectly anticipated ones and zeros are represented by the counts 172 and 81, respectively. The numbers 1371, 1848, 472, 513 in Fig. 1 correspond to the zero, one, two, three, and four target classes, respectively.

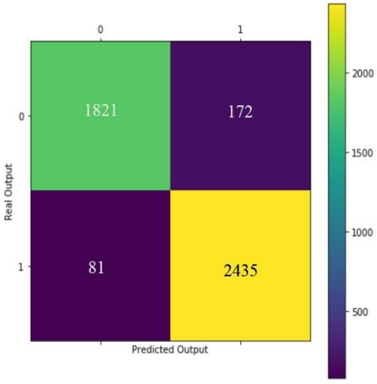


Fig. 12. Confusion matrix of binary class SVM

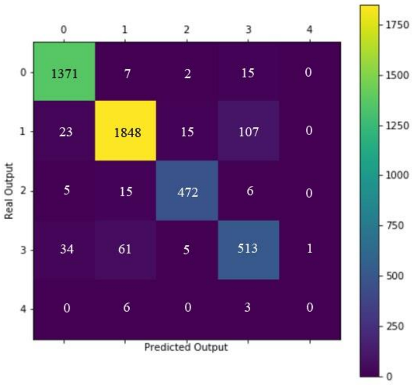


Fig. 13. Confusion matrix of multi class SVM

C. *K-Means Clustering*

Unlike KNN and SVM, K-means Clustering is not a classification technique; rather, the data is clustered by the algorithm based on their similarity. As a result, the accuracy should be lower than SVM and KNN. A procedure known as the elbow method is used to find the value of K. The procedure is performed over a variety of K values in this technique, and the average score for all clusters is calculated and plotted. From the resulting plot Fig. 15, the optimum K value is the one corresponding to the breaking point in the graph, i.e., 2 here. The parameters used in the function like maximum iteration, tolerance, n_init (number of time algorithms run with different centroid values), etc. are taken with reference to [6]. The testing and training accuracy of the K-means model is 70.83% and 70.08%. In Fig. 14 the count 1950 corresponds to the correctly predicted zero and 1244 corresponds to correctly predicted one. The count 43 and 1272 shows the wrongly predicted one and zero target class, respectively.

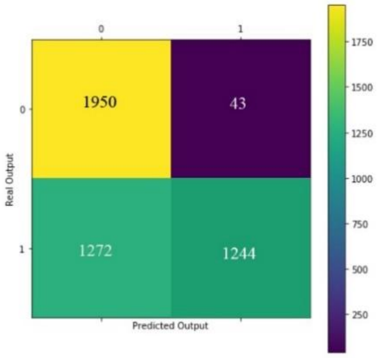


Fig. 14. Confusion matrix of K-Means Clustering

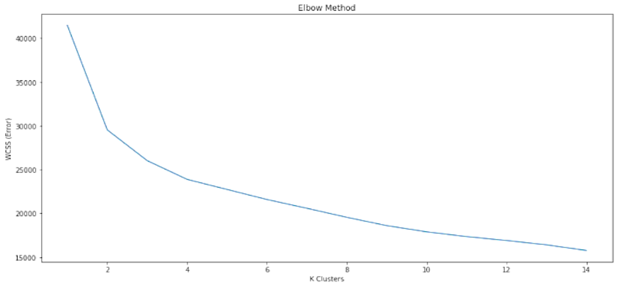
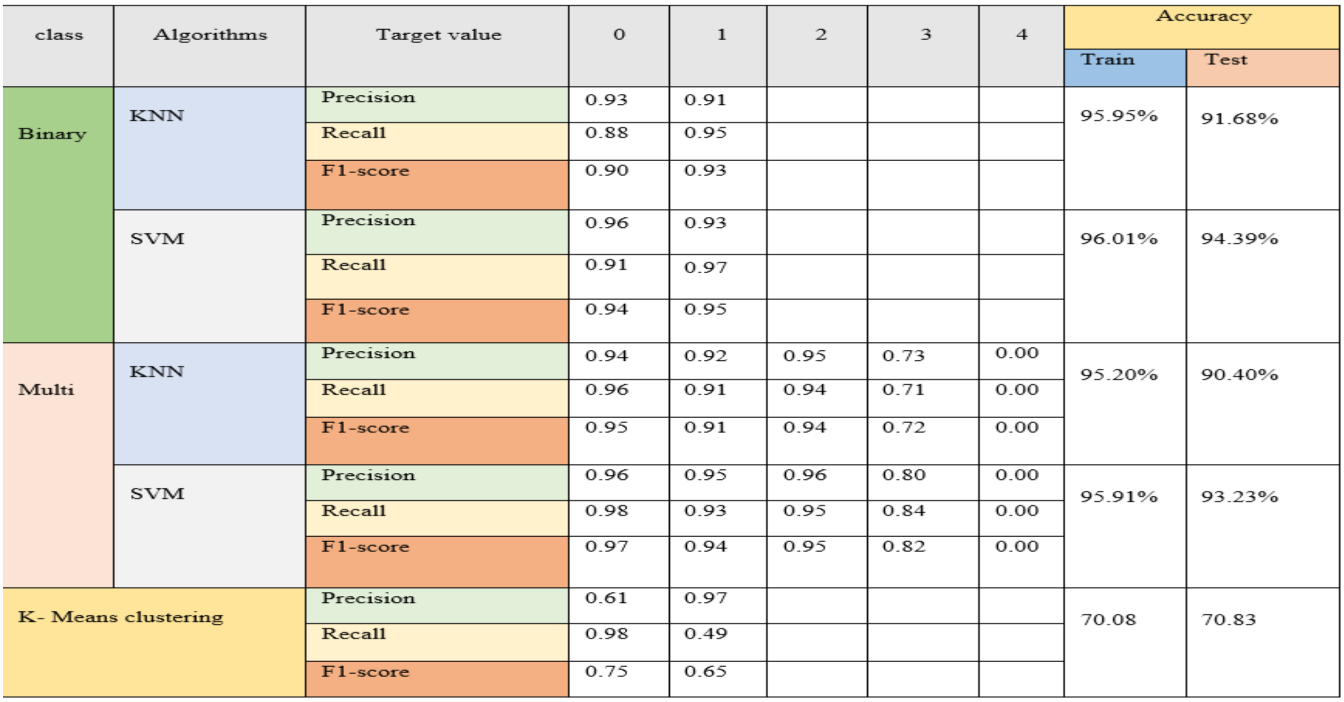


Fig. 15. Elbow Method Plot

Tab. 1. Evaluation Parameter values for different algorithm



class	Algorithms	Target value	0	1	2	3	4	Accuracy	
								Train	Test
Binary	KNN	Precision	0.93	0.91				95.95%	91.68%
		Recall	0.88	0.95					
		F1-score	0.90	0.93					
	SVM	Precision	0.96	0.93				96.01%	94.39%
		Recall	0.91	0.97					
		F1-score	0.94	0.95					
Multi	KNN	Precision	0.94	0.92	0.95	0.73	0.00	95.20%	90.40%
		Recall	0.96	0.91	0.94	0.71	0.00		
		F1-score	0.95	0.91	0.94	0.72	0.00		
	SVM	Precision	0.96	0.95	0.96	0.80	0.00	95.91%	93.23%
		Recall	0.98	0.93	0.95	0.84	0.00		
		F1-score	0.97	0.94	0.95	0.82	0.00		
K- Means clustering		Precision	0.61	0.97				70.08	70.83
		Recall	0.98	0.49					
		F1-score	0.75	0.65					

Tab. 1 summarizes the evaluation parameters of three different ML algorithms, K-means clustering, KNN and SVM. From the table it is very clear that the SVM gives the best result whereas K-means is having very poor performance.

VI. Conclusion

In this research inorder to identify network intrusion, we offer a novel detection model based on Non-Linear SVM. The results of multiple experiments on the KDD-CUP99 dataset show that on both binary and multiclass assaults, it outperforms KNN and K-Means Clustering. Precision, recall, F1-Score, and accuracy metrics were used to assess the model's efficiency. SVM consistently produced promising results in predicting the type of network assault. Furthermore, because the detection model is based on supervised learning, it would be difficult for the model to correctly predict an attack in a scenario where the network model encounters an unexpected environment. The model would become more robust and capable of handling real-time applications if this issue could be resolved. We'll put this on the back burner for now.

VII. References

[1] Z. Xiaofeng and H. Xiaohong, "Research on intrusion detection based on improved combination of K-means and multi-level SVM," 2017 IEEE 17th International Conference on Communication Technology (ICCT), Chengdu, 2017, on pp. 2042-2045, doi: 10.1109/ICCT.2017.8359987.

[2] Jayshree Jha and Leena Ragha. Article: Intrusion Detection System using Support Vector Machine. *IJAIS Proceedings on International Conference and workshop on Advanced Computing 2013* ICWAC(3):25-30, June 2013.

[3] K. Ghanem, F. J. Aparicio-Navarro, K. G. Kyriakopoulos, S. Lambotharan and J. A. Chambers, "Support Vector Machine for Network Intrusion and Cyber-Attack Detection," 2017 Sensor Signal Processing for Defence Conference (SSPD), London, 2017, on pp. 1-5, doi: 10.1109/SSPD.2017.8233268.

[4] M. Zhang, B. Xu and J. Gong, "An Anomaly Detection Model Based on One-Class SVM to Detect Network Intrusions," 2015 11th International Conference on Mobile Ad-hoc and Sensor Networks (MSN), Shenzhen, 2015, pp. 102-107, doi: 10.1109/MSN.2015.40.

[5] S. D. D. Anton, S. Sinha, H. D. Schotten, “Anomaly-based Intrusion Detection in Industrial Data with SVM and Random Forest.” 2019 27th International Conference on Software, Telecommunications and Computer Networks (SoftCOM). On arXiv:1907.10374 [cs.CR]

[6] Siddiqui, Mohammad Khubeb & Muhammad, Shams. (2013). Analysis of KDD CUP 99 Dataset using Clustering based Data Mining. International Journal of Database Theory and Application. 6. 23-34. 10.14257/ijdta.2013.6.5.03.

[7] Nazeer, K.A., & Sebastian, M. (2009). Improving the Accuracy and Efficiency of the k-means Clustering Algorithm.