Attack Prediction in a Network using Support Vector Machine

Himanshu Londhe
Department of Computer Science and Electrical Engineering
University of Maryland
Baltimore County
email: hlondhe1@umbc.edu

***Abstract*—A lot of systems that are built these days have the need to be secure. This includes network security and all systems should be able to cope with attacks from the outside. Cyber threats detection is a major component in security which is provided to organizations. In this paper, it is described how to predict an attack by classification of incoming traffic data provided at the 'IEEE BigData 2019 cup: Suspicious Network Event Recognition.'.[1]. The approach here is to use Support Vector Machine to classify whether the incoming traffic, based on the tags is an attack or not**

***Keywords— Cyber Security, Support Vector Machine, Machine Learning, Data analytics, BigData, Data mining*.**

I. INTRODUCTION

The data mining challenged was arranged by IEEE where contestants were provided with a dataset containing network traffic data. The task is to detect truly suspicious events and false positives [confusion matrix link]. The goal is to notify the Security team whether the incoming unseen traffic is an attack or not given the training data. Here we assume that the training data is accurate.

Out of the various classification algorithms, support vector machines which were developed in 1963 by Vladimir Vapnik is well suited for this type of challenge as it provides great accuracy in classification. This is a supervised learning classification algorithm and is polynomial in nature. It is well suited for handling big data-sets [2].

The aim of our classifier is to correctly predict which traffic data is an attack and to notify the SOC team at SoD[1].

II. BACKGROUND WORK

A. Choosing a classifier and why Support Vector Machine

For this challenge algorithms like for Support Vector Machines, Random Forest, k-Nearest Neighbor (KNN), Naïve Bayes can be used to classify the data set given. Choosing the best algorithm considering all the tradeoffs offs is a difficult task.

The study done by Phan Thanh Noi et al compares SVM, knn and Random forest by classifying the sentinel-2 imagery [3]. The study concludes that the accuracy for svm is the highest among the three which was called the 'overall accuracy'. SVM provides high accuracy even when the data-set is of large size and can be implemented in linear time complexity [3].

Work done by Jayshree Jha et al in their paper concludes that since network attacks have increased these days due to increase in number of systems a requirement for an Intrusion Detecting System is necessary [4]. As the volume of data is very large, we cannot detect such attacks manually. Thus, there is a need for a machine learning algorithm which can, in real time detect incoming attacks. Along with other classifiers, the author believes that support vector machines are one of the best algorithms to implement to classify anomalous behavior in a network. Furthermore, it also illustrates on how to choose the best features along with dimensionality reduction to choose for classifying since Support Vector Machine (SVM) is a binary classifier. A data set with the garbage columns removed will help the classifier run faster and more accurately. The proposed system by the paper is shown in the figure below:

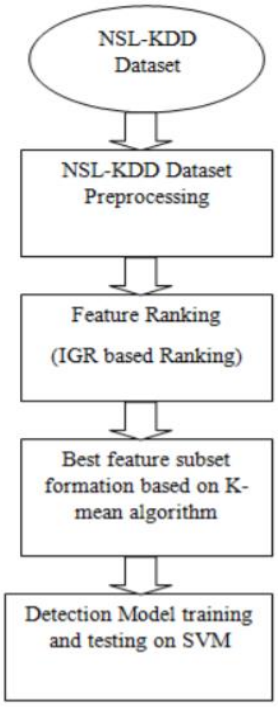

Figure 1: Intrusion Detection System Architecture Proposed in the paper [4].

The research paper on an enhanced SVM done by Himanshu Londhe et al [2] classifies the HIGGS data set provided on University of California, Irvine's Machine Learning Repository. This study chooses the best features by brute force classifying the possible combinations of the reduced data set size. When the max score is achieved for the features, they are chosen by the classifier to run the algorithm with the full data set. But the accuracy with smaller data sets does not necessarily co relate with the full-size data set. The study used SVM for classifying the dataset with sizes of 10000,100000, and 500000 and has 18 features. The training data set provided for the IEEE challenge is of size 32295 and has 36 columns. Since SVM is scalable to this extent as shown in [2], it is very well suited for this type of classification work. The table below shows classification accuracy and time with respect to the size of the data set [2].

Table I

Execution Timing [2]

Data Size	Spiral SVM	SVM	Percentage Increase (Average Increase in time and accuracy)
10,000	0.226/87.8%	0.227/74.7%	8.96%
100,000	5.996/89.69%	6.108/75.63%	10.21%
500,000	47.339/89.7%	51.243/75.64%	13.09%

Here, we can see that the classification accuracy of SVM remains consistent with increase in data set size.

The study done by Mulay Snehal et al also investigates the use of Support Vector Machines in Intrusion Detection Systems [5]. They show that Support Vector Machine which is primarily a binary classifier can be used to do multi-class classification using decision trees.

For a data set like the one given for the challenge which has a large number of features we can use some of the techniques used in the above-mentioned paper [5].

III. PROPERTIES OF SUPPORT VECTOR MACHINES

Support Vector Machines uses this equation to classify the given data. The algorithm is given data in the form of $(\vec{x}_1, y_1)...(\vec{x}_n, y_n)$. Where y_i are either 1 or −1, each indicating the class to which the point x_i belongs. We want to find the "maximum-margin hyperplane" that divides the group of points x_i for which $y_i = 1$ and also for 0. This is done by using Lagrange's Multiplier [2]. The Equation comes out as:

$$f(\vec{w}, b) = [\frac{1}{n}\sum_{i=1}^{n}\max(0, 1 - y_i(w.x_i - b))] + \lambda||w||^2$$

Where '*f*' is the classifier function and vector '*w*' is the width vector [6].

- Support vector machine is primarily a binary classifier. It searches for the closest points which it calls the "support vectors" [2].
- Once it has found the closest points, the SVM draws a line connecting them.
- When the algorithm has two points, it connects them to draw or create a line on the plot.
- This is often called as the hyperplane that separates the positive and negatively marked values.
- Anything above or below the line gets classified in two categories respectively.
- In the case of our given data, the classifier will classify the data into category as 'notified' or 'not notified' which are denoted by '1' and '0'.

So finally, our challenge here is to:

- Figure out how to reduce the dimensions in the data set and choose attributes to classify with.
- Select how much percent data to train and test on.
- Perform classification using different kernels of SVM.
- Calculate accuracy scores and performance metrics and maximize accuracy in classifying the data set.

IV. THE APPORACH

The SVM library provided scikit-learn gives five types of kernels which are 'linear', 'polynomial', 'radial basis function (rbf)', 'sigmoid', and 'precomputed' [7].

The figure below shows how SVM performs by using these different kernels [7].

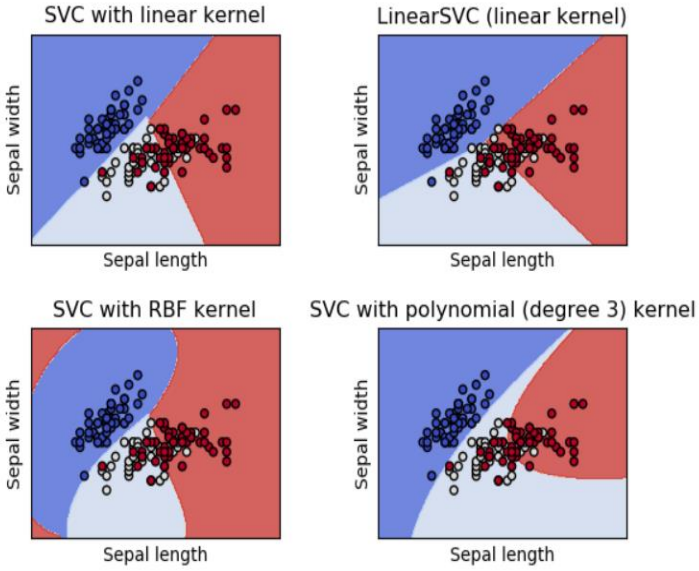

Figure 2: Classification example with various kernel [7].

The steps included in the proposed classification model include preprocessing the data and scaling it. Doing principle component analysis and then implementing the svm algorithm using sklearn libraries. The figure below shows the flow for how the model will use the data to classify the data.

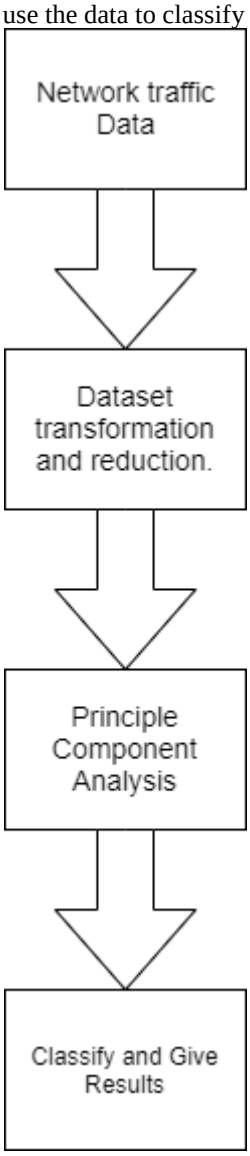

Figure 3: Flow of the Proposed SVM model

A. *Data Splitting*

We have decided to do an 80-20 split on the training data provided. That is, we will split the 80% of our training data to train the model and the remainder of the 20% to test our data. Therefore, we will be removing the 'notified' label from our testing data to check accuracy.

B. *Scaling and Priciple Component Analysis*

As mentioned before, it is a challenge for us to determine which features in the data set are vital for getting valid results.

By doing Principle Component Analysis (PCA), we can extract dominant patterns from the dataset and set a complexity score to all the features analyzed. In the initial analysis, the algorithm looks for outliers and strong groupings in the plots, indicating that the data matrix perhaps should be "polished" or whether disjoint modeling is the proper course. For plotting purposes, two or three principal components are usually sufficient, but for modeling purposes the number of significant components should be properly determined [8].

Thus, by using PCA, we can drastically reduce the number of features used in the actual classification phase. The resulting time in execution is significantly lower for the SVM when data was preprocessed using PCA. The following table shows the time it took for the linear SVM kernel to classify the same data set with and without doing PCA.

Table 2

Run Time of SVM with and without PCA

	With PCA	Without PCA
Time taken for execution	5 min 13 secs	2 hours 18 mins 39sec

Since, we have reduced the classification time drastically using PCA, we can now perform the classification with useful features only.

V. Performace Measures

A. *Confusion Matrix*

For measuring the performance of a classification algorithm results we rely on calculating the confusion matrix. It comprises of four values that are:

- True Positives: Things that are correct and were correctly identified. This value should be maximum.
- True Negatives: Things that were wrong and were correctly identified as wrong. This value should be maximum
- False Positives: Things that were wrong but were identified as correct. Value should as low as possible
- False Negatives: Things that were true but were identified as wrong. This value too should be minimum.

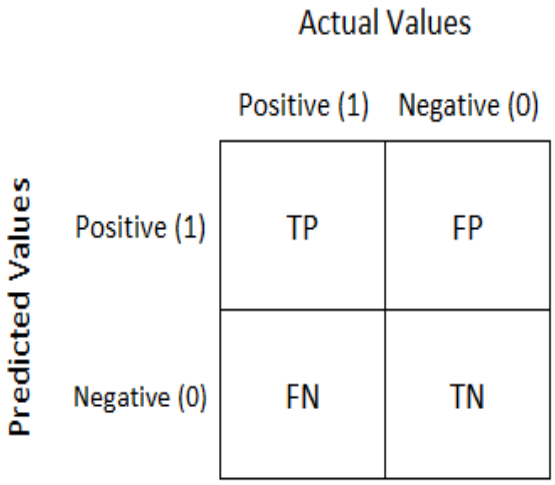

Figure 4: Confusion Matrix.

B. Derived Values

The values from the confusion matrix are further used to calculate accuracy, precision, recall and F1 score. These are all defined as following:

- Accuracy: It is defined as Accuracy = (True Positive + True Negative) / (Positives + Negatives).
- Precision: It is defined as Precision = True Positive / (True Positives + False Positives).
- Recall: It is defined as Recall = True Positives / (True Positives + False Negatives)
- F1 score: It is defined as F1 = 2True Positives / (2*True Positives + False Positives + False Negatives)

The concept of F1 was introduced to report the overall performance of the classifiers. Having only high accuracy or recall or precision is not enough. So, considering all the trade-offs between recall, precision and accuracy, F1 score combines these performance metrices to give a balanced score out of 1. The higher the F1, better the performance is of the system.

VI. Results

For computing results, we have considered four different kernels for SVM. Each kernel gives different results with little variation in the accuracy but a lot of changes in the other derived performance metrices. By observing all the performances, we can determine which kernel of SVM is best suited for this specific application.

Table 3

Linear SVM Confusion Matrix

	True Positive	True Negative
Predicted Positive	7410	0
Predicted Negative	476	0

Accuracy = 93.96%

Precision = 1.000

Recall = 0.9396

F1 = 0.9103

Table 4

Polynomial SVM Confusion Matrix

	True Positive	True Negative
Predicted Positive	7389	21
Predicted Negative	465	11

Accuracy = 93.84%

Precision = 0.9972

Recall = 0.9408

F1 = 0.9123

Table 5

RBF SVM Confusion Matrix

	True Positive	True Negative
Predicted Positive	7409	1
Predicted Negative	472	4

Accuracy = 94.00%

Precision = 0.9999

Recall = 0.9401

F1 = 0.9115

Table 6
Sigmoid SVM Confusion Matrix

	True Positive	True Negative
Predicted Positive	7210	200
Predicted Negative	435	41

Accuracy = 91.93%

Precision = 0.9730

Recall = 0.9431

F1 = 0.9069

From the above results, we see that all the SVM kernels yield almost the same accuracy and precision. All the models have a good recall value. The task at hand is then to choose the kernel which has the best average score, that is the one with the highest F1 score, which is highest for the polynomial SVM kernel. Even so, the true negative values that should be as high as possible is very low in all the models. That could be the case as we consider that the data set has a lot less notified as attack events.

This model thus gives us an accuracy of 93.84% and an F1 score of 0.9123.

VII. Conclusion

Thus, for any future attacks that will occur over the network the SVM classifier will almost correctly identify which one is truly an attack or isn't. The polynomial kernel in SVM is best suited for the challenge at hand. In the wake of rising number of network attacks there is a need of a state-of-the-art Intrusion Detection System which uses machine learning algorithms. The SVM algorithm helps achieve this goal by one step.

References

[1] IEEE BigData 2019 Cup: Suspicious Network Event Recognition: https://knowledgepit.ml/suspicious-network-event-recognition/

[2] Enhanced Support Vector Machine with Speed Up and Reduced Sensitivity,Harshal Chaudhari, Himanshu Londhe, Nachiket Namjoshi, Rahul Kolhatkar, Sandeep Chaware.*"Enhanced Support Vector Machine with Speed Up and Reduced Sensitivity"*, Volume 6, Issue XII, International Journal for Research in Applied Science and Engineering Technology (IJRASET) Page No: 502-506.

[3] Thanh Noi P, Kappas M. Comparison of Random Forest, k-Nearest Neighbor, and Support Vector Machine Classifiers for Land Cover Classification Using Sentinel-2 Imagery. *Sensors (Basel)*. 2017;18(1):18. Published 2017 Dec 22. doi:10.3390/s18010018.

[4] 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. .

[5] Mulay, Snehal & Devale, P.R. & Garje, Goraksh. (2010). Intrusion Detection System Using Support Vector Machine and Decision Tree. International Journal of Computer Applications. 3. 10.5120/758-993.

[6] Support Vector Machines (SVM) URL: https://en.wikipedia.org/wiki/Support_vector_machine.

[7] Scikit-learn: Machine Learning in Python Pedregosa, F. and Varoquaux, G. and Gramfort, A. and Michel, V. and Thirion, B. and Grisel, O. and Blondel, M. and Prettenhofer, P. and Weiss, R. and Dubourg, V. and Vanderplas, J. and Passos, A. and Cournapeau, D. and Brucher, M. and Perrot, M. and Duchesnay, E. p. 2825--2830. 2011

[8] Svante Wold, Kim Esbensen, Paul Geladi,Principal component analysis, Chemometrics and Intelligent Laboratory Systems,Volume 2, Issues 1–3,1987