Experimental Analysis of Random Forest, K-Nearest Neighbor and Support Vector Machine Anomaly Detection

Abstract

The goal of this study is to perform experimental analysis of anomaly intrusion detection techniques. Intrusion detection is used to discover attacks against computers and networks. There are several intrusion detection techniques that are in use, in this project we focus on anomaly detection techniques. Anomaly detection technique is a new approach to detect abnormal behavior or unauthorised access in the network which can prevent intentional and unintentional network intrusion by attackers. In the anomaly detection technique, there are supervised and unsupervised method. Here we focus on supervised anomaly detection techniques which are Support Vector Machine (SVM), Random Forest (RF) and K-Nearest Neighbor (KNN) and made a comparison between then using the "UNSW-NB 15" dataset.

1. INTRODUCTION

With the tremendous growth of the internet, networked computer systems are playing an increasingly more critical role in our society. The exquisite benefits of the internet also give rise to a darker side. The range of computer attacks has rapidly increase, within last few decades, new threats are created regularly with the aid of people and corporations that attack and misuse personal systems increasing the risk of intrusions and cyber-attacks to a critical level [1].

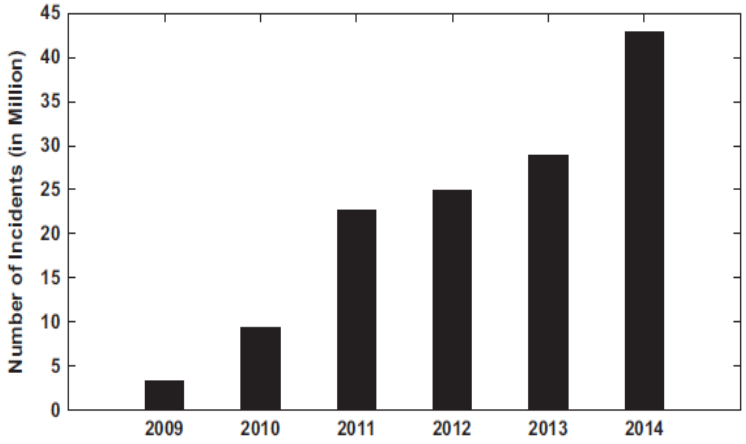

Figure 1: Growth of information security incidents [2].

Network intrusion is harming datacenters and servers by intruding in on confidential documents and by corrupting the system as well as the data from machines. The financial implication of such intrusions is immense and it's a world-wide problem. Internet service providers need to be alert to the situation using various intrusion detection techniques to make the internet a safer place.

Though there is an impressive development in Intrusion detection. Intrusion detection in network traffic is a complicated study. Network traffic is a necessary tool for network traffic estimation and traffic management system. In a specific time, the attacks can be detected using past network traffic to find any intrusive behavior in the network traffic pattern [2].

The most commonly used method for detecting cyber-attacks are based on signature-based detection techniques, but they are only able to detect previously known attack that match an existing signature and require the signature database to be manually updated regularly to be able

to catch new attacks. The focus has now shifted to the intrusion detection techniques that used data mining to overcome the limitations of the former [3].

There are two categories of data mining-based intrusion detection techniques; misused detection and anomaly detection.

The misuse detection techniques use an already labeled data instance to train a learning algorithm, each instance in the data set already has a label on which the algorithm learns the relationship between the input and output. The algorithms can classify the intrusions into categories such as attack or not attack, the requirement here is that the data including the attacks must be appropriately labeled for the model to be created [3].

The biggest advantage of these over the signature-based intrusion detection techniques is that the models of misuse are created automatically and are more sophisticated and accurate in detecting intrusions. The only problem arises when an attack whose instances have not been learned occur, in such a case it will not be able to identify it [3].

The other techniques for datamining called anomaly detection can build models of normal data and identify changes from the normal pattern of the data. The biggest advantage of these techniques is that they can identify new intrusions by observing a deviation from normal behavior. Once the system is trained, when it is given a new piece of test data it will attempt to identify if it is a normal or anomalous data. However, the chances of a false alarm in this technique are relatively high since it may see normal data with a pattern it has not seen before as an anomaly and thus flag them as an intrusion [3].

The main challenges of anomaly detection can be listed as follows:

- It is so hard to differentiate between the actual anomaly and normal dataset because dataset possesses noise [2].
- Scarcity in finding generic anomaly detection technique, such as we can not use an intrusion detection technique in a wired network to a wireless network [2].
- Behavior in network tends to change time to time and can be dissimilar with the propagation of time, therefore, present intrusion detection cannot be used in the future. Intruders are getting smarter, to keep up more appropriate featured detection techniques need to be developed to prevent further intrusion [2].
- Scarcity of publicly available labeled dataset in network anomaly detection [2].

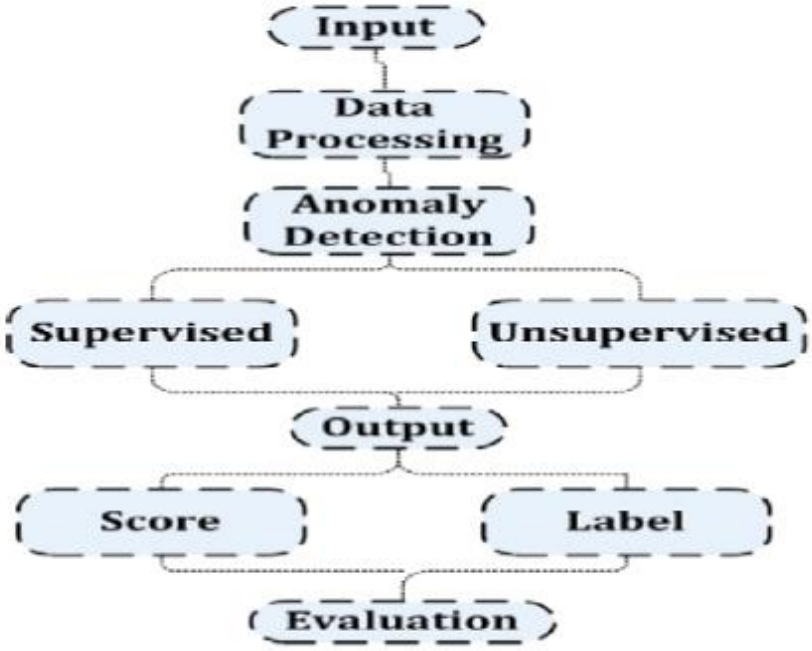

Figure 2: Generic framework for network anomaly detection [2]

Network anomaly is increasing in popularity and favorite research area of this century, there are several literatures on anomaly detection techniques. In this project we aim to verify some of these claims and compare some of the widely used supervised anomaly intrusion detection techniques.

Our focus is on Support Vector Machine (SVM), K-Nearest neighbor (KNN) and Random Forest (RF), even though there are several papers on hybrid implementations of these algorithm, we will analyze them individually. To accomplish our project, we used Weka Machine Learning program because of the vast number of algorithms and simplicity of its implementation. We used the UNSW-NB 15 data set. The aim of using this dataset is that we have an even distribution of attack and non-attack traffic and it is large enough to assume that the learned model will be able to generalize well when presented with unknown test data.

This project is organized as follows: in section 2 we introduced different intrusion detection techniques, while in section 3 we discuss WEKA and the UNSW-NB 15 dataset. Finally, section 4 provides details about the experimental followed by conclusion in section 5.

2. Different anomaly intrusion detection techniques

Anomaly detection is required to identify if a change from the established normal pattern has occurred and to flag this as an outlier. It tries to determine whether deviation from the established normal usage patterns can be flagged as intrusions. There several anomaly detection techniques based on various machine learning algorithms.

Some methods use supervised machine learning algorithms such as KNN, SVM, random forest, Logistical Regression, Naïve Bayes etc. In supervised machine learning methods, a model is developed by giving it a training dataset which has both input and output. The learning (training) task is to compute the approximate distance between the input and output examples to create a classifier model. When the model is created using the training datasets it can then be used to classify unknown data.

Others use unsupervised algorithms such as K means, in which the input is given without the corresponding output, which the algorithm must identify by itself. Efforts have also been made to combine both supervised and unsupervised algorithms to further enhance performance of the models.

K-Nearest Neighbour

One of the simplest machine learning classification technique is K-nearest neighbor (k-NN) which can help to classify samples. The distance between different points of the input vectors are calculated and based on that unlabeled points are assigned to the class nearest to its neighbors. One of the most important parameters in K-NN is the value of K and it will greatly influence the performance of the model. The choice of the K value can be tricky as it will influence both the time and accuracy. If K is too large it may take a very long time to classify the samples and if it is too small accuracy may be affected. In either case it's a complex task to come up with the optimal value of K [1].

There are two categories of KNN anomaly detection schemes; Density based, and Distance based. The Density based scheme is best suited for processing unevenly dense training data where applications require a multilevel detection such as criminal activities in electronic commerce [4]. While Distance based anomaly detection schemes are best suited for a unified

notion of anomaly. A data y is an anomaly in a dataset X if less than K data are within distance d from y.

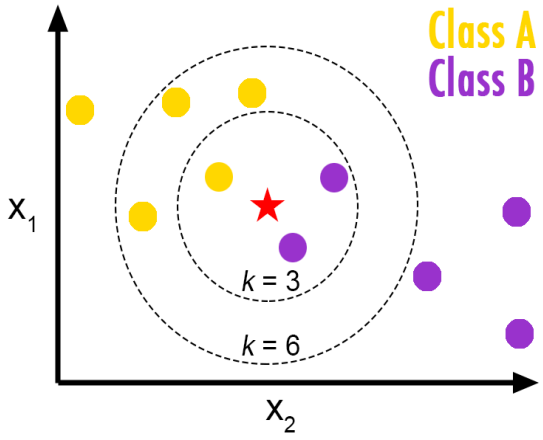

Figure 3: K- Nearest Neighbor [7]

There are many distance-based techniques: In [5] they proposed a distance-based anomaly detection technique which is based on the distance of a point from its Kth nearest neighbor. Each point is ranked based on its distance to the Kth nearest neighbor and the top n points of this ranking are determined to be anomaly. A partition-based algorithm for mining outliers is used to first partition the input data sets into disjoint subsets. A partition is removed as soon as it is determined that it does not contain anomaly resulting in substantial savings in computation [4]. Another approach in [6] tries to combine the kNN-classifier with the k-means clustering. Using K-means clustering a number of clusters are established, a triangular area is made from the two cluster centers and one data point. The sum of the triangular area of each data point is used as a new data, and K-NN with cross validation is then used to detect anomaly on the new data. Even though this approach may produce a better result than KNN alone it has a very high resource consumption [4].

Support Vector Machine

SVM first maps the input vector into a higher dimensional feature space and derives a hyperplane that maximizes the separating margin between the positive and negative classes (It separate a set of training vectors which belong to two different classes). The separating hyper-

plane is determined by support vectors and is extremely robust to outliers. The standard SVM algorithm is a supervised learning technique, which requires labeled data to create a classification rule [2]. However, it can also be adapted as an unsupervised learning algorithm in which it separates the entire set of training data from its origin. The SVM also provides a user specified parameter called penalty factor which allows for a tradeoff between the number of misclassified samples and the width of a decision boundary [2].

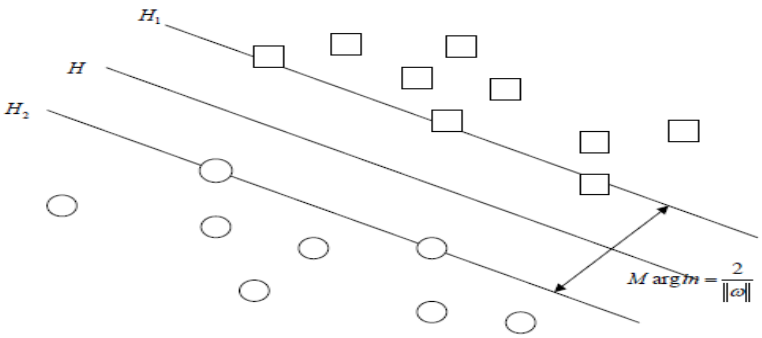

Figure 4: SVM Margin Maximization [8]

From figure 4, the Square and circle are two types of data, H represents the classification line, and H1 and H2 are both parallel to H. The distance between H1 and H2 is the sorting interval which separates the datasets into classes. Support vector machine tries to ensure that the classification is accurate by maximizing the distance between H1 and H2 [8].
In another scenario, the RSVM (Robust SVM) incorporates the averaging technique in the form of a class centre to make the decision surface smoother and automatically control regularization. In addition, the number and the quantity of support vectors in the RSVM is significantly less than the standard SVM which results in a reduced run time [9]. In [8] the paper combines both unsupervised fuzzy C-means clustering (FCM) and supervised Support Vector Machine (SVM) for industrial intrusion detection. FCM is used to first classify unlabeled data and then SVM is used to further classify the data, improving the classification accuracy without need to know class label in advance [8].

Random Forest

Random Forest is an ensemble learner method. It uses an extension to the bagging approach in which each classifier is built individually by working with a bootstrap sample of the input data. In Random Forest, a decision at a node split is made from a randomly selected number of features. This random feature selection helps it to efficiently handle many features and reduces the interdependence between feature attributes.

Random Forest is mostly suited for efficient handling of non-linear classification and handling data imbalances in different classes in a large dataset and considerably faster than other methods [10].

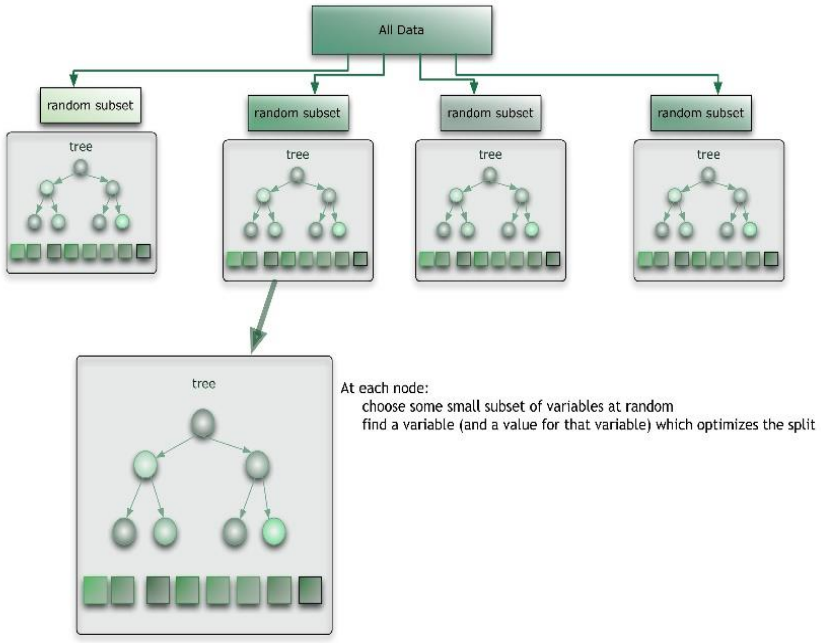

Figure 5: Random Forest [11]

There is no single best algorithm that achieves highest performance in all scenarios, so an algorithm can perform well in once scenario and may perform poorly on another scenario and vice versa. Over all according to the study in [12] Random Forest and SVM perform amongst the best while KNN performs significantly less. We aim to investigate this in our experiments discussed in the next sections.

Table 1: Normalized scores for each learning algorithm by metric (average over eleven problems) [12].

MODEL	CAL	ACC	FSC	LFT	ROC	APR	BEP	RMS	MXE	MEAN	OPT-SEL
BST-DT	PLT	.843*	.779	**.939**	**.963**	**.938**	.929*	**.880**	**.896**	**.896**	**.917**
RF	PLT	.872*	.805	.934*	.957	.931	**.930**	.851	.858	.892	.898
BAG-DT	–	.846	.781	.938*	.962*	.937*	.918	.845	.872	.887*	.899
BST-DT	ISO	.826*	.860*	.929*	.952	.921	.925*	.854	.815	.885	.917*
RF	–	**.872**	.790	.934*	.957	.931	**.930**	.829	.830	.884	.890
BAG-DT	PLT	.841	.774	.938*	.962*	.937*	.918	.836	.852	.882	.895
RF	ISO	.861*	**.861**	.923	.946	.910	.925	.836	.776	.880	.895
BAG-DT	ISO	.826	.843*	.933*	.954	.921	.915	.832	.791	.877	.894
SVM	PLT	.824	.760	.895	.938	.898	.913	.831	.836	.862	.880
ANN	–	.803	.762	.910	.936	.892	.899	.811	.821	.854	.885
SVM	ISO	.813	.836*	.892	.925	.882	.911	.814	.744	.852	.882
ANN	PLT	.815	.748	.910	.936	.892	.899	.783	.785	.846	.875
ANN	ISO	.803	.836	.908	.924	.876	.891	.777	.718	.842	.884
BST-DT	–	.834*	.816	**.939**	**.963**	**.938**	.929*	.598	.605	.828	.851
KNN	PLT	.757	.707	.889	.918	.872	.872	.742	.764	.815	.837
KNN	–	.756	.728	.889	.918	.872	.872	.729	.718	.810	.830
KNN	ISO	.755	.758	.882	.907	.854	.869	.738	.706	.809	.844
BST-STMP	PLT	.724	.651	.876	.908	.853	.845	.716	.754	.791	.808
SVM	–	.817	.804	.895	.938	.899	.913	.514	.467	.781	.810
BST-STMP	ISO	.709	.744	.873	.899	.835	.840	.695	.646	.780	.810
BST-STMP	–	.741	.684	.876	.908	.853	.845	.394	.382	.710	.726
DT	ISO	.648	.654	.818	.838	.756	.778	.590	.589	.709	.774
DT	–	.647	.639	.824	.843	.762	.777	.562	.607	.708	.763
DT	PLT	.651	.618	.824	.843	.762	.777	.575	.594	.706	.761
LR	–	.636	.545	.823	.852	.743	.734	.620	.645	.700	.710
LR	ISO	.627	.567	.818	.847	.735	.742	.608	.589	.692	.703
LR	PLT	.630	.500	.823	.852	.743	.734	.593	.604	.685	.695
NB	ISO	.579	.468	.779	.820	.727	.733	.572	.555	.654	.661
NB	PLT	.576	.448	.780	.824	.738	.735	.537	.559	.650	.654
NB	–	.496	.562	.781	.825	.738	.735	.347	-.633	.481	.489

3. Weka Machine Learning Tool

There are several machine Learning tools, however in this project we will be using Weka. Weka is one of the famous machine learning tools, it was developed by professionals of University of Waikato, New Zealand in 1997 and was issued on the General Public License [13].

Weka uses a Graphical User Interface (GUI) to help study the information in datasets. It can perform various calculations using up to 49 data preprocessing ,3 association rules, and up to 76 order classifications [13].

Some of the modules of Weka include Knowledge Flow used for running machine learning test and experimenter which is used for analysis and comparison of various testing. It uses dataset formats such as .csv, .arf to extract relevant information from a crude one [13].

Figure 6: Weka machine learning interface.

Dataset

The raw network packets of the UNSW-NB 15 data set were created in the Cyber Range Lab of the Australian Centre for Cyber Security (ACCS) using the IXIA PerfectStorm tool to generate a normal and attack behavior traffic. Tcpdump tool was used to capture 100 GB of raw network traffic [UNSW-NB15]. In this dataset there are nine types of attacks namely; Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode and Worms [14].
Th Argus, Bro-IDS tools was used, and twelve models were developed to generate totally 49 features with the class label.
The dataset is partitioned into two sets a training set and a testing set. The UNSW_NB15_training-set.csv file contains 82,332 records of training set while the UNSW_NB15_testing-set contains 175,341 records of the testing set [14].

The dataset we have used here contains nine type of attacks. The brief description of each type is mentioned below:
(1) Fuzzers: This is type of attack where the attacker put massive amount of random data to a program or an operating system or in network until it crashes and then try to find out security loophole of it.

(2) Analysis: These are different type of intrusion techniques mainly to intrude through various computer port scanning such as email ports (spams), web-scripts ports etc.
(3) Backdoor: This technique used to bypass authentication of system, by securing unauthorized remote access to a device. Here when a hacker tries to breach the computer with remote access, which are said to be the backdoor and the attackers attack the computer without letting knowing the system of this backdoor intrusion. Or we can describe it as a disguised point of entry.
(4) DoS: DoS attack is a well-known attack. It is basically creating unnecessary fake traffic in the network so that know other traffic or the real traffic can not enter in to the network. A server can go under DoS attack if attacker puts so many page requests basically from engine to replicate the requests and make the memory of the server extremely busy. So, the real user never gets the access into the server.
(5) Exploit: This is kind of a trap set by attackers where a host falls into trap by making it walk through a sequence of instructions that can lead them to expose their vulnerability.
(6) Generic: A Block-cipher has a block and key size and a generic attack is the type of attack where without considering the structure of the block cypher, the attack works against all block cipher using a hash function.
(7) Reconnaissance: When intruder try to gather information about the vulnerability of a network or of a computer to take its security controls, we call it reconnaissance attack. there is serval way of reconnaissance, such as trojan, fishing, malicious mail link or website link, free application software like antivirus
(8) Shellcode: here the attacker intrudes through a slice of the short code using a shell to gain access and ultimately to compromise the machine.
(9) Worm: worm a type of virus that can spread from computer to computer by itself. In majority of the time, it uses a computer network and creates its own existence and wait for any security failure near target computer to access it [14].
The dataset was labeled using two attributes; the label attribute uses 0 for normal traffic and 1 for an attack while the attack_cat represents all the 9 category of attacks and the normal.

Table 2 : A part of UNSW-NB15 data set Distrution [14]

Category	Training set	Testing set
Normal	56,000	37,000
Analysis	2,000	677
Backdoor	1,746	583
DoS	12,264	4089
Exploits	33,393	11,132
Fuzzers	18,184	6,062
Generic	40,000	18,871
Reconnaissance	10,491	3,496
Shellcode	1,133	378
Worms	130	44
Total Records	175,341	82,332

4. EXPERIMENTAL ANALYSIS

In this project we are focussing on the analysing various anomaly detection techniques. We will be using the UNSW-NB 15 dataset discussed earlier to compare especially Support Vector Machines (SVM), K-Nearest Neighbor and Random Forest for anomaly detection.
Some of the major factors that affect the performance of a model are:
Model overfitting, Class unbalance, Missing values, Features choice for the model.

Since the UNSW-NB data set has both a training data set of about 82332, and a testing dataset of about 175,341 records, we have enough data to train and also evaluate the model to ensure there is little to no overfitting and also the ratio of attack and normal traffic is pretty evenly distributed avoiding the class unbalance issue. Finally, what makes this dataset stand out for out experiments is the fact that there are no missing values in the training/testing set and the number of features is not too large to accomplish our experiment within a reasonable time frame.

Feature Selection:
Optimal feature selection is vital for the success of the model, there are several methods to perform feature selection such as filter method, wrapper method and embedded method, variants of which are available in Weka. We tried some of these methods and used ten selected features in the preliminary phase and realized that the model's performance was below acceptable during

testing i.e. below 25% accuracy. Each of the methods selected different set of features, thus we decided to use the entire feature set to train and test the model. This was computationally an expensive process and took considerable time to train and evaluate.

Comparing KNN, SVM, Random Forest Models using Weka Explorer Module

We first used the Weka Explorer module to create Machine Learning models for each of the three algorithms and compared their Correctly/Incorrectly Classified, Kappa Statistics, Mean Absolute Error (MAE)/Root Mean Squared Error (RMSE), Confusion Matrix and F-Measure.

Table 3: Correctly and Incorrectly Instance Classification

Algorithms	Correctly Classified Instances	Incorrectly Classified Instances
Support Vector Machines	82319	13
K-Nearest Neighbor	82332	0
Random Forest	82332	0

In our observation from the table 3 we can see that all the three Algorithm have a high degree of accuracy in classifying the data but SVM, is observed to perform with lesser accuracy as compared to K-Nearest Neighbor and Random Forest.

Table 4: Kappa Statistics

Algorithm	Kappa Statistics
Support Vector Machines	0.9997
K Nearest Neighbor	1
Random Forest	1

Kappa is the chance-corrected measure of agreement between the classifications and the true classes. It is calculated by taking the attribute expected by chance away from the observed attribute and dividing by the maximum possible value of the attribute. A value greater that 0 means that the classifier is doing better than chance which is desired. With our results in Table 4,

both K-Nearest Neighbor and Random Forest perform better than Support Vector Machines (SVM).

Table 5: Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE)

Algorithm	Mean Absolute Error	Root Mean Squared Error
Support Vector Machines	0.0002	0.0126
K Nearest Neighbor	0	0
Random Forest	0.0005	0.0031

The Mean Absolute Error and the Root Mean Squared Error are used to evaluate the performance of the model. The smaller the RMSE the better the model. From Table 5, K-Nearest Neighbor performs best while SVM again performs worst of the three.

Table 6: Accuracy Measurement

Algorithm	TP Rate	FP Rate	Precision	Recall	F-Measure
Support Vector Machines	1	0	1	1	1
K Nearest Neighbor	1	0	1	1	1
Random Forest	1	0	1	1	1

The accuracy, precision, recall and F measure of the models is obtained using the confusion Matrix. The confusion Matrix which identifies the number of correct and incorrect predictions is computed by the model in comparison to the actual outcomes in the data. It is made of an NxN matrix where N is the number of classes.

Table 7: Confusion Matrix

	Correctly Classified	Incorrectly Classified
Selected	True Positive	False Positive
Not selected	False Negative	True Negative

$$Accuracy = \frac{True\ Positive + True\ Negative}{True\ Positive + True\ Negative + False\ Negative + False\ Positive}$$

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$F\ Meausre = \frac{2 * Precision * Recall}{Precision + Recall}$$

In our SVM, KNN and Random Forest model, as seen in Table 6, all three have the same accuracy, precision, recall and F score.

Comparing KNN, SVM, Random Forest using Weka Experimenter Module

The model based on the training dataset and testing dataset was giving us a very close outcome of about 100% when we used separate models in the Weka Explorer module, we decided to use the Experimenter Module to compare the three-machine learning algorithms, to see if the results maybe different. Both training and testing datasets are used to compare the three algorithms.

Percent Correct:

Table 8: Comparison of percent correct

Dataset	KNN	SVM	Random Forest
UNSW_NB15_training-set	100	76.22	100
UNSW_NB15_testing-set	100	71.21	100

This time it could clearly be seen that SVM had the worst performance of the three, on the training dataset it classified 76.22% correctly while on the testing dataset it classified 71.21% correctly.

Root Mean Squared Error:

Table 9: Comparison of Root Mean Squared Error

Dataset	KNN	SVM	Random Forest
UNSW_NB15_training-set	0	0.49	0.01
UNSW_NB15_testing-set	0	0.54	0.01

The lower the RMSE the better the model. As in Table 9, it was observed that SVM had the highest RMSE of 0.49 for training dataset and 0.54 for testing dataset and KNN again had the lowest RMSE of 0 and random forest RMSE 0.01, thus KNN performs best and SVM the worst.

F Measure:

Table 10: Comparison of F measure

Dataset	KNN	SVM	Random Forest
UNSW_NB15_training-set	1.00	0.79	1.00
UNSW_NB15_testing-set	1.00	0.18	1.00

From Table 10, we see that the F Measure of SVM is the lowest while both KNN and Random Forest have the same F-score.

5. Conclusion

Even though there is no single algorithm that may perform optimally for every situation. In our case when using the Weka Explorer and the Experimenter modules, we get close to the same performance for all the three anomaly detection algorithms. From our analysis, it's clear that KNN performs best while SVM performs worst in all scenarios even though SVM is computationally most resource intensive. However, the scenario may change if we perform feature selection and use a different dataset. Due to time constraint we were not able to analyze the algorithms using other optimizations of the algorithms to evaluate their performance.

Reference

[1] Chih-Fong Tsai, Yu-Feng Hsu, Chia-Ying Lin, Wei-Yang Lin, Intrusion detection by machine learning: A review, Expert Systems with Applications, Volume 36, Issue 10, 2009, Pages 11994-12000, ISSN 0957-4174.

[2] Mohiuddin Ahmed, Abdun Naser Mahmood, Jiankun Hu,A survey of network anomaly detection techniques,Journal of Network and Computer Applications, Volume 60,2016,Pages 19-31,ISSN 1084-8045.

[3] A. Lazarevic, L. Ertoz, V. Kumar, A. Ozgur, and J. Srivastava, “A Comparative Study of Anomaly Detection Schemes in Network Intrusion Detection,” in Proceedings of the 2003 SIAM International Conference on Data Mining, 0 vols., Society for Industrial and Applied Mathematics, 2003, pp. 25–36.

[4] M. Xie, J. Hu, S. Han and H. Chen, "Scalable Hypergrid k-NN-Based Online Anomaly Detection in Wireless Sensor Networks," in *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 8, pp. 1661-1670, Aug. 2013.

[5] S. Ramaswamy, R. Rastogi, and K. Shim, “Efficient Algorithms for Mining Outliers from Large Data Sets,” ACM SIGMOD Record, vol. 29, no. 2, pp. 427-438, June 2000.

[6] C.-F. Tsai and C.-Y. Lin, “A Triangle Area Based Nearest Neighbors Approach to Intrusion Detection,” Pattern Recognition, vol. 43, no. 1, pp. 222-229, Jan. 2010

[7] https://dslytics.wordpress.com/2017/11/16/classification-series-5-k-nearest-neighbors-knn.

[8] Shang, Wenli & Cui, Junrong & Song, Chunhe & Zhao, Jianming & Zeng, Peng. (2018). Research on Industrial Control Anomaly Detection Based on FCM and SVM. 218-222.

[9] HuW, Liao Y, Vemuri VR. Robust anomaly detection using support vector machines. In: Proceedings of the international conference on machine learning; 2003.

[10] M. S. Alam and S. T. Vuong, "Random Forest Classification for Detecting Android Malware," 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, Beijing, 2013, pp. 663-669.

[11] http://blog.citizennet.com/blog/2012/11/10/random-forests-ensembles-and-performance-metrics

[12] R. Caruana and A. Niculescu-Mizil, “An Empirical Comparison of Supervised Learning Algorithms,” in Proceedings of the 23rd International Conference on Machine Learning, New York, NY, USA, 2006, pp. 161–168.

[13] A. K. Pandey, D. S. Rajpoot and D. S. Rajpoot, "A comparative study of classification techniques by utilizing WEKA," *2016 International Conference on Signal Processing and Communication (ICSC)*, Noida, 2016, pp. 219-224

[14] N. Moustafa and J. Slay, “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, vol. 25, no. 1–3, pp. 18–31, Apr. 2016.