A Behavior Based Intrusion Detection System Using Machine Learning Algorithms

Murat OĞUZ *muratoguz@live.com*
Institute of Science
Electrical and Computer Engineering
Melikşah University
Talas-Kayseri, 38280, Turkey

İhsan Ömür BUCAK *iobucak@meliksah.edu.tr*
Faculty of Engineering and Architecture
Department of Electrical and Electronics Engineering
Melikşah University
Talas-Kayseri, 38280, Turkey

Abstract

Humans are consistently referred to as the weakest link in information security. Human factors such as individual differences, cognitive abilities and personality traits can impact on behavior and play a significant role in information security. The purpose of this study is to identify, describe and classify the human factors affecting Information Security and develop a model to reduce the risk of insider misuse and assess the use and performance of the best-suited artificial intelligence techniques in detection of misuse. More specifically, this study provides a comprehensive view of the human related information security risks and threats, classification study of the human related threats in information security, a methodology developed to reduce the risk of human related threats by detecting insider misuse by a behavior-based intrusion detection system using machine learning algorithms, and the comparison of the numerical experiments for analysis of this approach. Specifically, by using the machine learning algorithm with the best learning performance, the detection rates of the attack types defined in the organized five dimensional human threats taxonomy were determined. Lastly, the possible human factors affecting information security as linked to the detection rates were sorted upon the evaluation of the taxonomy.

Keywords: Human Factors, Information Security, Taxonomy, Classification, Behavior-based Intrusion Detection.

1. INTRODUCTION

Recently published annual reports of enterprise security companies consistently show that employees are very often the cause of the most significant and costly security breaches and this has motivated our study [1, 2]. Organizations that value their information need to protect it from threat sources that exploit vulnerabilities in information systems. Although attacks originating from outside threat sources, such as hacking attempts or viruses, have gained a lot of publicity, the more risky attacks come from inside [3, 4].

In this study, the human factors affecting information security in organizations were identified and classified. A most up-to-date taxonomy about human threats was introduced. A model to reduce the risk of insider misuse was developed and the performances of machine learning algorithms in detection of misuse were measured. Next, the detection rates of Probe, Denial of Service, User to Root, and Remote to Local attack types defined in the human threats taxonomy were also determined. Furthermore, these detection rates were mapped to the human factors through the evaluation of the taxonomy. More specifically, this study provides the following: a comprehensive view of the human related information security risks and threats, classification study of the human related threats in information security, a methodology developed to reduce the risk of human related threats by detecting insider misuse by behavior-based intrusion detection systems using machine learning algorithms, and the comparison of the numerical experiments for analysis of this approach.

Lastly, by using the machine learning algorithm with the best learning performance, the possible human factors affecting information security as linked to the detection rates were sorted upon the evaluation of the taxonomy.

Rest of this paper is organized as follows: In Section 2, the human factors in information security is briefly mentioned, and a taxonomy and classification study about the human threats are given. Section 3 explains the methodology and related issues. In Section 4, different machine learning algorithms are presented and their results including a comparison will be discussed. Finally, conclusions and future work are given in Section 5 and Section 6.

2. CLASSIFICATION STUDY

Primarily, the human factors in information security breaches were researched, and two major questions came up at this stage. What types of human factors cause what kind of information security breaches? The purpose of asking this question is to identify information security breaches and human factors, and highlight the link in between. The second question is if there is a classification, taxonomy or a study already published about this subject? The purpose of asking this question is to get to know that how the researchers approach the subject historically and where we are standing. The intrusion detection systems research community has developed various approaches for classifying intrusion incidents. Human related threats are considered a special case of an intrusive activity. This taxonomy study covers a more recent and up-to-date taxonomy effort with increased dimensions and features. We certainly believe that the level of the effort in this taxonomy study brings it to the more advanced and general level that can be used for several objectives built upon it.

2.1 Impact Level

After looking through well-known and recent taxonomies in the literature, Padayachee's and Stanton's taxonomies were examined. The Padayachee's taxonomy was found very theoretical mostly related with motivation but it was advanced [5]. On the other hand, Stanton et al. developed very practical and improvable taxonomy but it was basic [6]. At this point, a practical and advanced taxonomy study that can combine their main strength came forward as a result.

Firstly, the impact level was added to the new taxonomy effort as the third dimension as it was related with the risk of the behavior. If one can define the risk of behavior, precautions must be taken to reduce or avoid it. If the risk is low or the cost of treatment is not cost effective for the organization, it can be ignored [7]. Impact level comes from consideration of three compromises as follows:

1. Confidentiality: The potential impact if the information is seen by those who should not see it.
2. Integrity: The potential impact if the accuracy or completeness of the information is compromised.
3. Availability: The potential impact if the information becomes inaccessible.

We get the impact levels from National Vulnerability Database (NVD), Common Vulnerability Scoring System (CVSS) and Common Vulnerabilities and Exposures (CVE) [8-10]. Adding the impact level on a three level taxonomy enables us to know the risk level of the behavior so that we can accept it or ignore it but how can we detect it? Another dimensional need came out with the motivation of this question. None of the previously mentioned taxonomies are oriented towards detection of insider Information Technology (IT) misuse in terms of considering how we would approach the task of monitoring activities to determine where problems may be apparent.

2.2 Threat Layer

In determining the means to link classification to the method of detection, it is considered appropriate to classify human behavior as based on the level of the system at which they might be detected. The basis for this is that different types of behaviors can occur at varying layers of the system. With this form of classification in mind, the concept can be illustrated using a variety of recognized insider activities, and then considering the different layers at which they may be detected. The examples of the incidents concerned are considered in the sub-sections that follow. These consider what could be monitored, and how this could be used to detect, control and restrict misuse-related behavior. This layer comprises the following layers to be inspected further:

1. Network-layer: Mostly misuse activities are using network services or they are related to the network services, therefore several type of misuse can be detected by monitoring the traffic at network-layer.
2. System-layer**:** Some misuse activities may occur at the system layer on the host systems by running applications like malwares or hack tools. Endpoint agent would need to be installed on the system for detecting system-layer misuse.

2.3 Threat Type

Once the monitoring level is identified, the next question arises on how to detect the actions which endanger the integrity, confidentiality or availability of a resource as an effort to provide a solution to existing security issues. This can be done by intrusion detection systems (IDS).

There are a large variety of attack types [11]. A good taxonomy makes it possible to classify individual attacks into sharing common properties [12]. One widely used taxonomy [13] divides attacks into groups sharing common practices: Probes, Denial of Service (DoS), User to Root (U2R) and Remote to Local (R2L). Threat type is important. The threats need to be detected to manage them. Hence, the threat type was added to the taxonomy as another attribute and the five dimensional human threats taxonomy was obtained as shown in Table 1. Bucak discussed all levels of the taxonomy work in detail [14].

Human threats taxonomy is tailored to the needs of automated human threat prediction. Once it is compared with the similar taxonomies in the literature, it is observed that the others use lesser dimensions for detection as seen in Table 2. The Neumann-Parker taxonomy classifies intrusions into nine categories, which describe the type of the attacks [15]. Cheswick-Bellovin taxonomy classifies attacks into seven categories [16]. Tuglular taxonomy classifies misuse incident in three dimensions such as misuse type, response level and misuse layer [17]. Magklaras-Furnell taxonomy classifies misuse in three basic dimensions: abuser system role and expertise, reason of misuse and system impact [18]. Stanton and colleagues taxonomy classifies security behavior by using two-factor taxonomy, where the two factors are intentionality and technical expertise [6]. Padayachee taxonomy classifies information security behavior by extrinsic and intrinsic motivations and their sub-dimensions like intention and impact level [5].

3. METHODOLOGY

Human threats were classified. However, they need to be detected to prevent them. There are many approaches which use data mining algorithms to detect intrusions. Network based detection is one of the mechanism to accurately distinguish insider behavior from the normal behavior. Anomaly detection has become up-to-date topic because of the weakness of signature-based IDSs in detecting novel or unknown attacks.

3.1 Why KDD Data Set?

KDD is the most widely used data set for the evaluation of anomaly detection methods. This data set is prepared by Stolfo et al. [19] and is built based on the data captured in DARPA IDS evaluation program [20]. DARPA is about 4 gigabytes of compressed raw data of 7 weeks of network traffic, which can be processed into about 5 million connection records, each with about

100 bytes. The two weeks of test data have around 2 million connection records. The simulated attacks fall in one of the following four categories: Probes, DoS, U2R and R2L [21].

Expertise	Intention	Impact Level	Threat Layer	Attack Type	Title
High	Malicious	High	OS	User to Root (U2R)	Intentional destruction
High	Malicious	Medium	Network	Denial of Service (DoS)	Man in the middle
Low	Malicious	High	Network	Denial of Service (DoS)	Resource exhaustion
Low	Malicious	Medium	OS	Remote to Local (R2L)	Stealing privilege
High	Neutral	High	Network	Probe	Dangerous tinkering
High	Neutral	Medium	Network	Probe	Accidentally allowing
Low	Neutral	High	OS	User to Root (U2R)	Naive mistakes
Low	Neutral	Medium	OS	User to Root (U2R)	Personal usage
High	Beneficial	High	OS	Remote to Local (R2L)	Aware assurance
High	Beneficial	Medium	Network	User to Root (U2R)	Paying attention
Low	Beneficial	High	OS	Remote to Local (R2L)	Basic hygiene
Low	Beneficial	Medium	Network	Remote to Local (R2L)	Awareness

TABLE 1: Human Threats Taxonomy.

Study	Expertise	Intention	Impact Level	Threat Layer	Attack Type
Bucak-Oguz	✓	✓	✓	✓	✓
Neumann-Parker	-	-	-	-	✓
Cheswick-Bellovin	-	-	-	-	✓
Tuglular	-	-	✓	✓	✓
Magklaras-Furnell	✓	✓	✓	-	-
Stanton and Colleagues	✓	✓	-	-	-
Padayachee	-	✓	✓	-	-

TABLE 2: Comparison of the Taxonomies.

Finding a suitable dataset to test machine learning algorithms is a difficult task. DARPA's military network attack simulation dataset from the KDD Challenge Cup [22] was used as a representation of network traffic captured in a real network environment. KDD is the most widely used as one of the few publicly available data sets for the evaluation of anomaly detection systems. KDD dataset is still in use today because it contains a variety of well-known misuse behavior that are similar to attacks seen today and it also contains novel attacks.

3.2 Data Set Reorganization and Preparation

One of the most important deficiencies in the KDD data set is the huge number of redundant records, which causes the learning algorithms to be biased towards the frequent records, and thus prevent them from learning infrequent records, which are usually more harmful to networks such as U2R and R2L attacks [21]. In addition, the existence of these repeated records in the test set will cause the evaluation results to be biased by the methods which have better detection rates on the frequent records [21]. To solve this issue, all the repeated records in the entire KDD train and test set were removed, and only one copy of each record was maintained.

The reorganized KDD data set has still a very large number of records. In KDD Cup, due to hardware requirements they used %10 of training set. The size of the data set was reduced because of the same reason; therefore, 126135 records from the reorganized KDD training data set were selected and named as improved KDD training data set. This was done homogeneously by considering the attack type ratio. Holdout validation was used and the data ratio was taken 80% for training and %20 for testing according to the findings of Kearns [23]. Furthermore, 25184 records from the reorganized KDD test data set were set homogeneously and an improved KDD test data set was created.

4. NUMERICAL EXPERIMENTS

In this study, various supervised machine learning algorithms were chosen in order to establish benchmark measurements for each type of algorithm which are currently used by the machine learning community. Weka [24] software was decided to be used for evaluation because it contains a collection of visualization tools and algorithms for data analysis and predictive modeling together with graphical user interfaces for easy access to this functionality. It is also well-suited for developing new machine learning schemes [20]. The experiments described below were performed on a computer with Intel Xeon E5504 2.00 GHz CPU and 8 Gigabytes memory. The evaluation of the various algorithms investigates the performance of each individual learning algorithm, when applied to the data set [25-32].

4.1 ZeroR

The ZeroR function was trained against the improved KDD data set, and can be evaluated through a confusion matrix. There are two classes as normal and anomaly. Considering all classes in a row by column matrix, this will produce a 2x2 matrix. The confusion matrix produced is shown in Table 3. The mean ‘Area Under the Curve (AUC)’ is 0.5 and the ‘Receiver Operating Characteristic (ROC)’ curve for ZeroR is shown in Figure 1 where the x-axis attributes false positive (FP) rate and the y-axis attributes true positive (TP) rate.

a	b	classified as
16080	0	a = normal
9104	0	b= anomaly

TABLE 3: ZeroR Confusion Matrix.

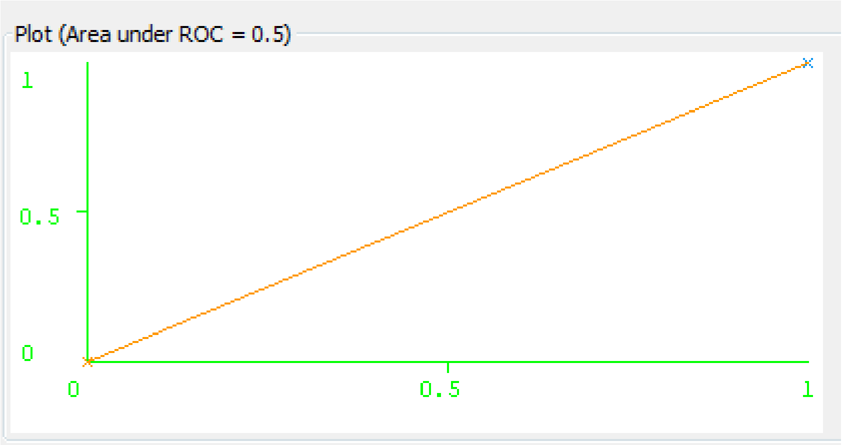

FIGURE 1: ZeroR ROC Curve for Normal Class.

The ZeroR algorithm, as simple as it is, does provide us with a worst-case baseline; using the testing set provided demonstrates that blindly selecting the class based on the majority of seen classes in the training set can be worse than guessing, for example when the majority of training classes becomes the minority in the test data set.

4.2 Naive Bayes

Naive Bayes algorithm was run against data set using the default settings required such that it compares the frequency across all attributes to the class without assuming any conditional probabilities between classes, in other words, complete independence. It took 3 seconds to build, and 0.84 seconds to run. The confusion matrix produced is shown in Table 4.

a	B	classified as
15184	896	a = normal
1856	7248	b= anomaly

TABLE 4: Naive Bayes Confusion Matrix.

Naïve Bayes algorithm produced a 0.891 average TP rate, and a 0.15 FP rate. The average precision and recall measurements were 0.891. The mean AUC was 0.946 and the ROC curve for Naïve Bayes is shown in Figure 2. The overall accuracy and mean AUC values of Naïve Bayes algorithm is much higher than the previously seen ZeroR implementation.

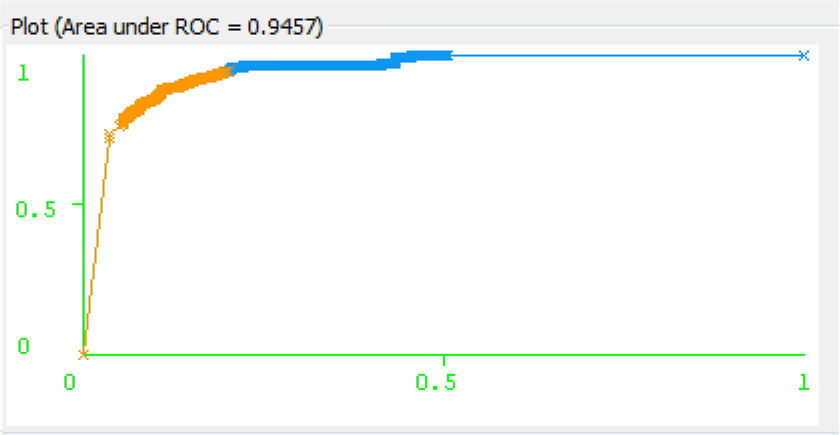

FIGURE 2: Naive Bayes ROC Curve for Normal Class.

4.3 J48 Decision Tree

J48 tree is an open source version of C4.5, and was run with the confidence factor set to 0.25, and the minimum number of instances per leaf set to 2.

a	B	classified as
15936	144	a = normal
672	8432	b= anomaly

TABLE 5: J48 Confusion Matrix.

Considering the confusion matrix in Table 5, the following performance metrics were produced: the algorithm using the dataset produced a 0.968 average TP rate, and a 0.05 average FP rate. The average precision and recall measurements were 0.968. The mean AUC was 0.958 and the ROC curve for J48 are shown in Figure 3. The values are much higher than the Naive Bayes results.

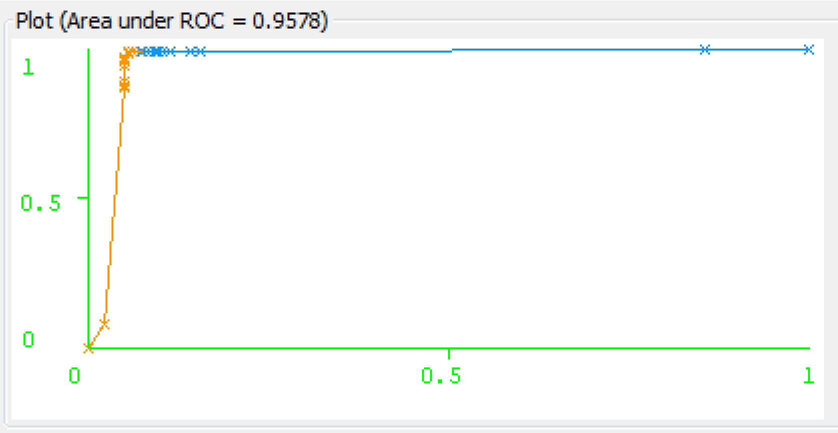

FIGURE 3: J48 ROC Curve for Normal Class.

4.4 Logistic Regression

Logistic Regression is a standard implementation of the sigmoid function training through gradient descent. The Logistic Regression algorithm was run against the improved KDD data set.

a	B	classified as
15264	816	a = normal
928	8176	b= anomaly

TABLE 6: Logistic Regression Confusion Matrix.

Considering the confusion matrix in Table 6, the algorithm using the dataset produced a 0.931 average TP rate, and a 0.083 average FP rate. The average precision and recall measurements were 0.931. The mean AUC was 0.938 and the ROC curve for Logistic Regression is shown in Figure 4. The values are higher than the Naive Bayes results but lower than the J48 values.

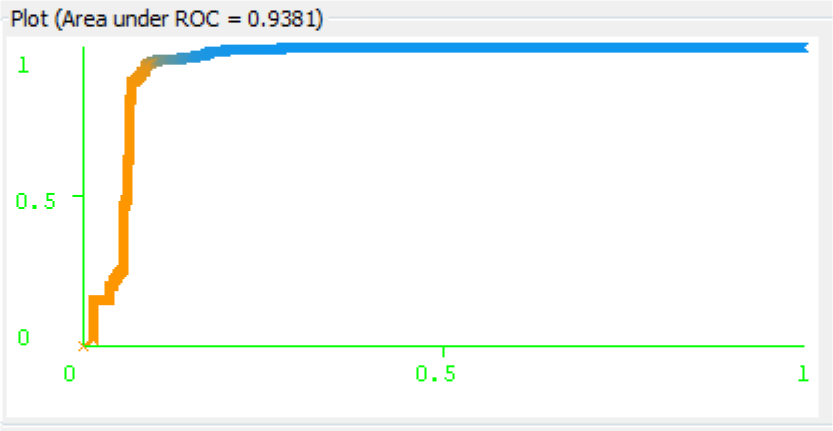

FIGURE 4: Logistic Regression ROC Curve for Normal Class.

4.5 K-Nearest Neighbor (K-NN)

Evaluation of the instance-based learning algorithm (or K-Nearest Neighbor as it is commonly called) was performed. In this experiment, the independent variable *k*, being the number of grouped points involved in the voting process, was initialized to default value 1 for initial training and testing, with no distance weighting for penalization. The K-NN algorithm was run against the standard dataset.

a	b	classified as
15824	256	a = normal
752	8352	b= anomaly

TABLE 7: K-NN Confusion Matrix.

Considering the confusion matrix in Table 7, the algorithm using the dataset produced a 0.96 average TP rate, and a 0.058 average FP rate. The average precision and recall measurements were 0.96. The mean AUC was 0.95 and the ROC curve for K-NN is shown in Figure 5. The values are higher than the Naive Bayes and Logistic Regression results but lower than the J48 values. The K-NN training and testing times are slightly different from the previously seen classifiers.

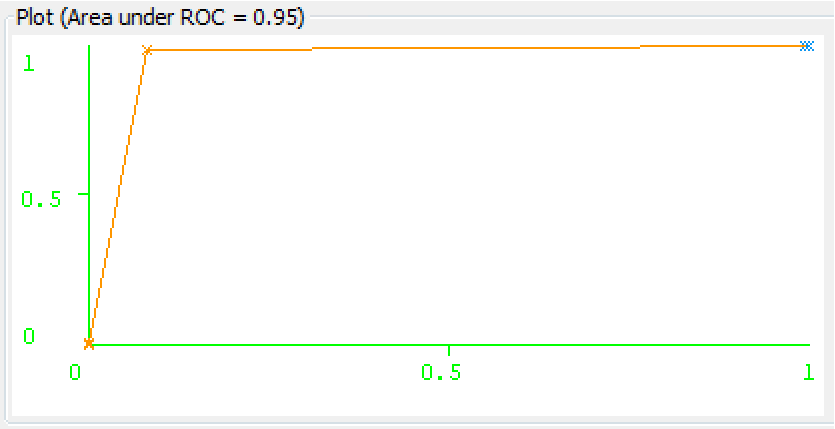

FIGURE 5: K-NN ROC Curve for Normal Class.

4.6 Support Vector Machines (SVMs)

The Support Vector Machines, based on the popular Sequential Minimal Optimization (SMO) algorithm was used for the experiments. The SMO algorithm was run against the dataset.

a	b	classified as
15520	560	a = normal
1200	7904	b= anomaly

TABLE 8: SMO Confusion Matrix.

Considering the confusion matrix in Table 8, the algorithm, using the standard dataset, produced a 0.93 average TP rate, and a 0.097 average FP rate. The average precision and recall measurements were 0.93. The mean AUC was 0.917 and the ROC curve for SMO is shown in Figure 6. The values are higher than the Naive Bayes and closed to Logistic Regression results but lower than the J48 and K-NN values. The SMO training and testing times are much higher than the previously seen classifiers.

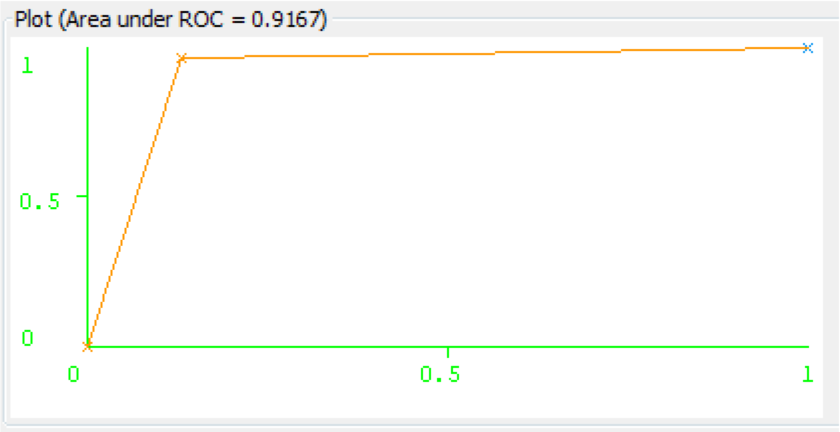

FIGURE 6: SMO ROC Curve for Normal Class.

4.7 Artificial Neural Networks

Multilayer Perceptron (MLP) algorithm is a standard implementation of the sigmoid perceptron using Least Mean Squared cost calculation, and gradient descent back-propagation. The single Multilayer Perceptron algorithm was run against the standard dataset.

a	b	classified as
15296	784	a = normal
880	8224	b= anomaly

TABLE 9: MLP Confusion Matrix.

Considering the confusion matrix in Table 9, the MLP algorithm, using the dataset, produced a 0.934 average TP rate, and a 0.079 average FP rate. The average Precision and Recall measurements were 0.934. The mean AUC was 0.951 and the ROC curve for MLP is shown in Figure 7. The values are higher than the Naive Bayes, Logistic Regression and SMO but lower than the J48 and K-NN values. It can be seen that the MLP network has a significant degradation in training time.

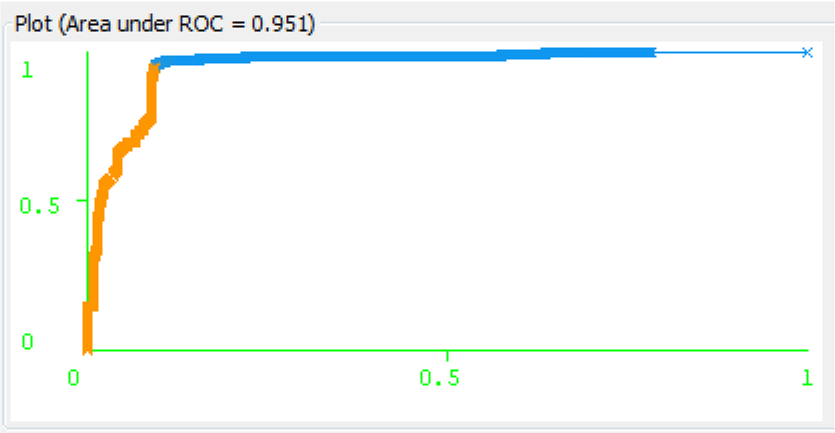

FIGURE 7: MLP ROC Curve for Normal Class.

4.8 Genetic Programming

Genetic Programming (GP) was run against the dataset where elite size set to 0.5, population size and new population size set to 100. The GP algorithm was run against the dataset.

a	b	classified as
15600	480	a = normal
2032	7072	b= anomaly

TABLE 10: GP Confusion Matrix.

Considering the confusion matrix in Table 10, the GP algorithm, using the dataset, produced a 0.901 average TP rate, and a 0.154 average FP rate. The average precision and recall measurements were 0.903 and 0.901 respectively. The mean AUC was 0.873 and the ROC curve for GP is shown in Figure 8. The values are higher than the Naive Bayes, but lower than the other classifier values.

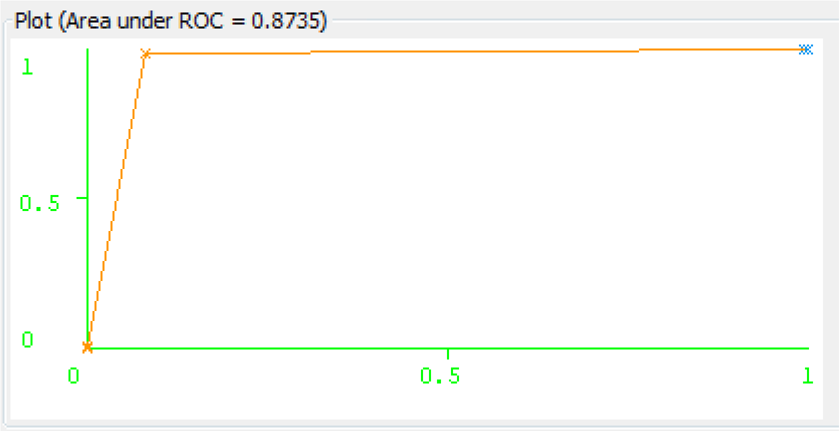

FIGURE 8: GP ROC Curve for Normal Class.

4.9 Comparison of the Results

The training times in seconds for each of the algorithms are illustrated in Figure 9, where iBK indicates K-NN.

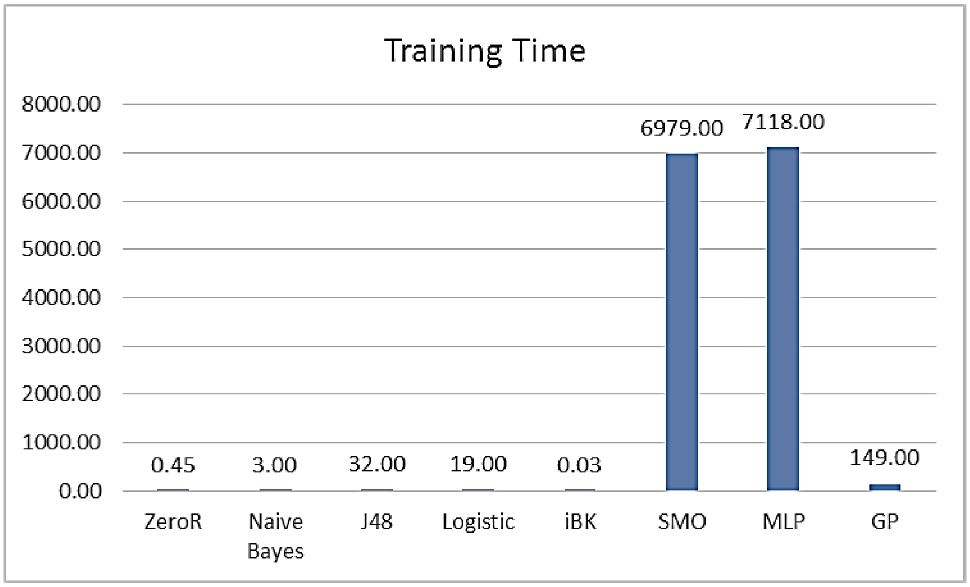

FIGURE 9: Training Times in Seconds.

The algorithm training times are widely different. As expected, the more complex algorithms such as artificial neural networks and support vector machines require more time to train than less complex algorithms such as decision trees or logistic regression. Based on the results of our experiments, the K-NN learner had the fastest learner build time (0.03 seconds). The testing times in seconds for each of the algorithms are illustrated in Figure 10.

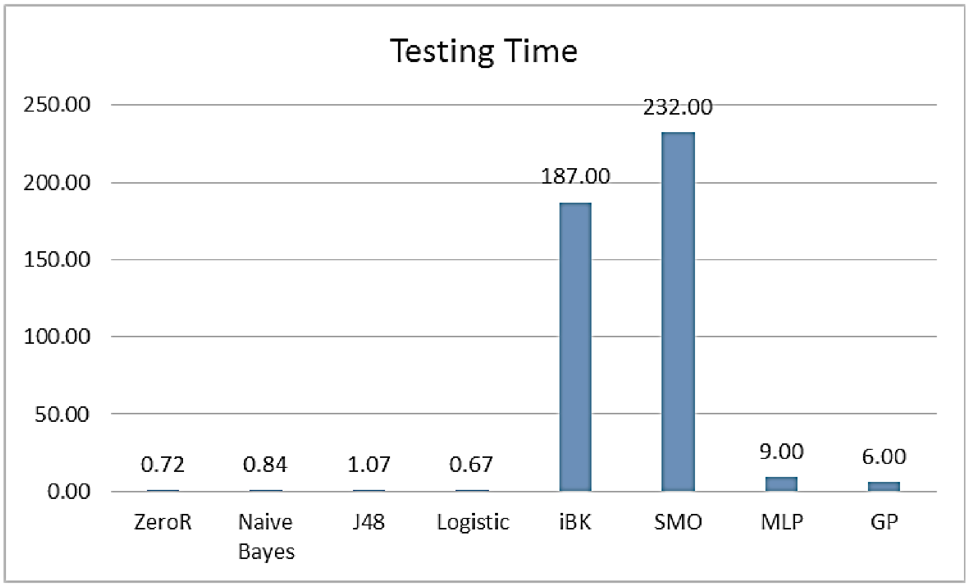

FIGURE 10: Testing Time in Seconds.

As with the training times, the algorithm testing times are also widely distributed. The previously more expensive learning algorithms, due to their quick access (decision tree) and propagation (neural network), are somewhat faster. Based on the results of our experiments, the Logistic Regression learner had the fastest learner test time (0.67 seconds). The ROC AUC values for each of the algorithms values are illustrated in Figure 11.

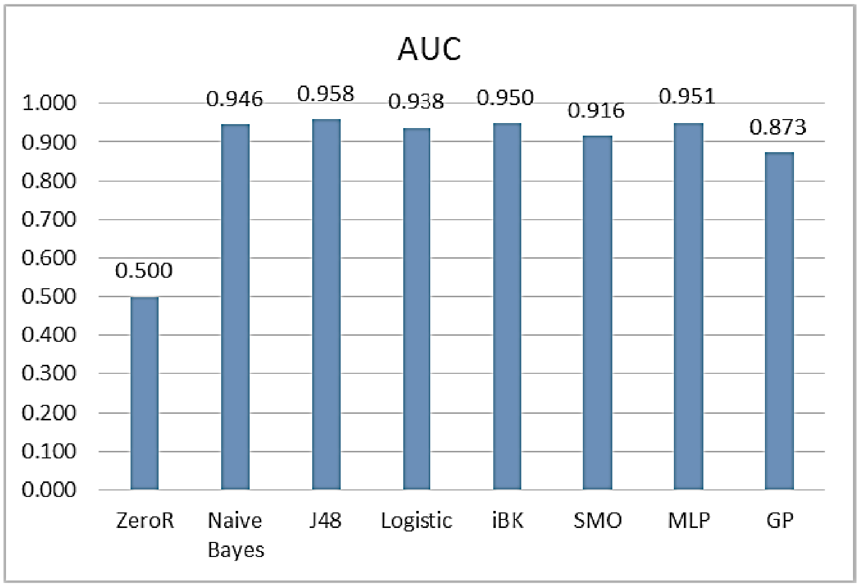

FIGURE 11: AUC Values.

The difference in accuracy between algorithms is much less apparent than the differences previously seen in the training and testing times. Based on the results of our experiments, the J48 classifier had the highest AUC (0.958). ZeroR, as expected, is the worst-case lower bound against which all others are compared.

4.10 Detection Rate of the Human Threats

In this study, a human threats taxonomy was created to contribute to the establishment of a behavior based detection model. The human behaviors were classified and the attack types of these behaviors were defined in order to define the risk and to measure the detection rate of those threats individually. In this human threats taxonomy, the attacks were divided into four classes: Probe, U2R, R2L, and DoS. According to the experimental results, J48 algorithm had the highest accuracy in anomaly detection, therefore J48 algorithm was used for measurement of detection rates. In this section, by running the J48 algorithm, the detection performances of the attacks were measured by types which were defined in the enriched taxonomy. The mean AUC, TP and FP rate values for the each type are shown in Table 10.

According to the results in Table 10, DoS attacks such as Man in the middle and Resource exhaustion threats ended up with the best detection rates above the average, and Probe type attacks such as Dangerous tinkering and Accidentally allowing threats ended up with better detection rates above the average. U2R types of attacks such as Intentional destruction, Naive mistakes, Personal usage and Paying attention threats had lower detection rates below the average. Lastly, R2L attacks such as Stealing privilege, Aware assurance, Basic hygiene, and Awareness threats had the lowest detection rates. The results show that detecting U2R and R2L attacks is more difficult than detecting DoS and Probe types of attacks. The reason behind this lies in the fact that most of the machine learning algorithms offer better level of classification rates for DoS and Probe attacks; because these type of attacks create a great deal of connections in a short period of time, but R2L and U2R attack types do not form a sequential pattern unlike DoS and Probe attacks. Therefore, the detection of R2L and U2R attacks by any classifier is a difficult task.

Expertise	Intention	Impact Level	Threat Layer	Attack Type	Title	AUC	TP Rate	FP Rate
High	Malicious	High	OS	User to Root (U2R)	Intentional destruction	0.909	0.915	0.107
High	Malicious	Medium	Network	Denial of Service (DoS)	Man in the middle	0.971	0.979	0.049
Low	Malicious	High	Network	Denial of Service (DoS)	Resource exhaustion	0.971	0.979	0.049
Low	Malicious	Medium	OS	Remote to Local (R2L)	Stealing Privilege	0.874	0.892	0.141
High	Neutral	High	Network	Probe	Dangerous tinkering	0.968	0.972	0.058
High	Neutral	Medium	Network	Probe	Accidentally allowing	0.968	0.972	0.058
Low	Neutral	High	OS	User to Root (U2R)	Naive mistakes	0.909	0.915	0.107
Low	Neutral	Medium	OS	User to Root (U2R)	Personal usage	0.909	0.915	0.107
High	Beneficial	High	OS	Remote to Local (R2L)	Aware assurance	0.874	0.892	0.141
High	Beneficial	Medium	Network	User to Root (U2R)	Paying attention	0.909	0.915	1.107
Low	Beneficial	High	OS	Remote to Local (R2L)	Basic hygiene	0.874	0.892	0.141
Low	Beneficial	Medium	Network	Remote to Local (R2L)	Awareness	0.874	0.892	0.141

TABLE 10: Detection Rates of the Human Threats.

5. CONCLUSION

A suitable taxonomy of human threat factors was introduced as based on threats associated to legitimate user actions. The impact level of attacks and the attack types were described. Examples about the human threats were given. The taxonomy has been tailored to the needs of automated human threat prediction. The establishment of this classification scheme paved the way for the construction of a suitable human threat prediction. Taxonomy has been an important milestone for this study, because it has enhanced the ability to examine the problem in a more systematic way and eventually contributed to the establishment of an insider threat prediction model.

A standard sample of machine learning algorithms, from the simplistic Zero-R and Naive Bayes, to the more sophisticated and expensive Support Vector Machine, Artificial Neural Network and Genetic Programming has been reviewed. Based on the results of our experiments, the J48 classifier had the highest AUC (0.958), and the K-Nearest Neighbor learner had the fastest learner build time (0.03 seconds), and the Logistic Regression learner had the fastest learner test time (0.67 seconds). The experiment observation has been described in terms of classification performance, or accuracy. Classification performance in this context has been measured in terms of true-positive, false-positive, true-negative, and false-negative. These dependent measurements have typically been performed in terms of percentage of data sets identified over all possible sets to be identified correctly.

The absolute times are highly dependent on the implementation platform, however the relative times are a strong indicator of relative algorithm performance. These metrics were analyzed according to their contribution for each algorithm. In practice, the training time is not as important as the testing time. Model creation in this case would be performed offline using already labeled data whereas testing would be on-line and produces an events-per-second metric that is a major consideration in the intrusion detection industry. If the best algorithm is desired to be chosen for intrusion detection, the accuracy and testing time values of the algorithms shall be compared. In this context, J48 appeared to be the best algorithm for detection when these values were compared. Because J48 had the highest accuracy and J48 had very fast test time which was very close to the fastest time. One of the reasons may be its becoming a decision tree algorithm. To understand why a decision tree algorithm had the best score, one can look through the decision tree properties [33].

The results also show that detecting U2R and R2L attacks is more difficult than detecting DoS and Probe types of attacks. The reason behind this lies in the fact that most of the machine learning algorithms offer a better level of classification rate for DoS and Probe attacks because these type of attacks create a good deal of connections in a short period of time but R2L and U2R attack types do not form a sequential pattern unlike DoS and Probe attacks therefore the detection of R2L and U2R attacks by any classifier is a difficult task.

In conclusion, based on the proposed system, the most important factors to identify human related threats in information security have been presented. The proposed system which used data including user-related and application-related factors have been tested against machine learning algorithms to determine whether or not those factors were effective and viable to profile user behavior.

The data set chosen has aimed to represent as many variations of user activity as possible such that user behavior with different roles captured within the system can be used effectively to reduce the risk of human related threats and, as the result, to show a higher detection and lower false alarm rates. At the same time, this will result in an accurately and reliably increased IDS.

6. FUTURE WORK

This study involved the application of a broad range of machine learning algorithms for the purpose of anomaly detection. These algorithms require an offline training phase, but the testing phase requires much less time and future work could investigate how well it can be adapted to performing online. The main difficulties in adapting these techniques for practical use are the difficulties involved in acquiring labeled training data and in investigating how the training on this dataset can be useful in classifying real datasets.

This study used the default settings for most of the algorithms that we tested. Every algorithm is unique, and will perform differently depending on the dataset. Some of the algorithms tested had very few options to consider when implementing them but some of the more complex algorithms have more parameters that are tunable. Considering the very large parameter space available to the more complex algorithms, future research could be performed into optimal methods of finding the right parameters for each algorithm combination in order to further increase the performance.

7. REFERENCES

[1] Verizon. “The 2013 Data Breach Investigations Report.” Internet: http://www.verizonenterprise.com/resources/reports/rp_data-breach-investigations-report-2013_en_xg.pdf, June 20, 2013 [Feb. 18, 2014].

[2] Checkpoint. “Checkpoint 2013 Security Report.” Internet: http://sc1.checkpoint.com/documents/security-report/, Jan. 20, 2013 [May 12, 2014].

[3] E.E. Schultz. “A framework for understanding and predicting insider attacks.” Proceedings of Computers and Security, 2002, pp. 526-531.

[4] W.H. Baker, C.D. Hylender and J.A. Valentine. “2008 Data Breach Investigations Report.” Internet: http://www.verizonenterprise.com/resources/security/databreachreport.pdf, Oct. 20 2008 [May 23 2013].

[5] K. Padayachee. “Taxonomy of compliant information security behavior.” Computers and Security, vol. 31, pp. 673-680, Jul. 2012.

[6] J.M. Stanton, K.R. Stam, P. Mastrangelo and J. Jolton. “Analysis of end user security behaviors.” Computers and Security, vol. 24, pp. 124-133, Mar. 2004.

[7] ISO/IEC 17799. “Information Technology-Security Techniques-Code of practice for information security management.” Internet: http://www.iso.org/iso/catalogue_detail?csnumber=39612, June 15, 2005 [Oct.10, 2013].

[8] NVD. “National Vulnerability Database.” Internet: http://nvd.nist.gov, Jan. 18, 2014 [Jan. 20, 2014].

[9] CVSS, “Common Vulnerability Scoring System.” Internet: http://www.first.org/cvss, Mar. 19, 2014 [Mar. 21, 2014].

[10] CVE. “Common Vulnerabilities and Exposures.” Internet: http://cve.mitre.org, Mar. 20, 2014 [Mar. 21, 2014].

[11] Fyodor. “Fyodor’s Exploit Word.” Internet: http://insecure.org/sploits.html, May 18, 2013 [Apr. 2, 2014].

[12] S. Mukkamala, A. Sung and A. Abraham. “Intrusion detection using ensemble of soft computing and hard computing paradigms.” Journal of Network and Computer Applications, vol. 28, pp. 167-182, 2005.

[13] K. Kendall. A database of computer attacks for the evaluation of intrusion detection systems, Master’s Thesis, MIT, 1999.

[14] I.O. Bucak. “An Extended Human Threats Taxonomy To Identify Information Security Breaches,” in Proc. Advances in Computing, Electronics and Communication (ACEC 2015), pp. 31-36, 2015.

[15] P.G. Neumann and D. B. Parker. "A Summary of Computer Misuse Techniques." Proceedings of the 12th National Computer Security Conference, 1989, pp. 396-407.

[16] W.R. Cheswick and S. M. Bellovin. Firewalls and Internet Security: Repelling the Wily Hacker. Boston, MA: Addison-Wesley, 1994, pp. 159-166.

[17] T. Tuglular. "A preliminary Structural Approach to Insider Computer Misuse Incidents." EICAR 2000 Best Paper Proceedings, pp. 105-125, Jan. 2000.

[18] G.B. Magklaras and S. M. Furnell. "Insider Threat Prediction Tool: Evaluating the probability of IT misuse." Computers & Security, vol. 21, pp. 62-73, Feb. 2002.

[19] Stolfo, W. Fan, W. Lee, A. Prodromius and P.K. Chan. “Cost-based modeling for fraud and intrusion detection: Results from the jam project.” DARPA Information Survivability Conference and Exposition (DISCEX), 2000, pp. 130-144.

[20] R.P. Lippmann and R.K. Cunningham. “Improving Intrusion Detection Performance Using Keyword Selection and Neural Networks.” Computer Networks, vol. 34, pp. 597-602, 2000.

[21] M. Tavallaee, E. Bagheri, W. Lu and A. Ghorbani. "A Detailed Analysis of the KDD CUP 99 Data Set," in Proc. Computational Intelligence in Security and Defense Applications (CISDA), 2009, pp. 53-58.

[22] KDD Cup. KDD Cup 99 task description. Internet: http://kdd.ics.uci.edu/databases/kddcup99/task.html, Nov. 30, 1999, [Jan. 29, 2014].

[23] M. Kearns. "A bound on the error of cross validation using the approximation and estimation rates, with consequences for the training-test split." Neural Computation, vol. 9, pp. 1143-1161, 1997.

[24] M. Hall, E. Frank, G. Holmes, B. Pfabringer, P. Reutermann and I.H. Witten. "The Weka data mining software: an update." SIGKDD Explorations, vol. 11, pp. 10-18, 2009.

[25] K. Xu, Z.L. Zhang and S. Bhattacharyya. "Profiling internet backbone traffic: behavior models and applications." in Proc. SIGCOMM, 2015, pp. 169-180.

[26] K.C. Khor, C.Y. Ting and S.P. Amnuaisuk. "From feature selection to building of Bayesian classifiers: A network intrusion detection perspective." American Journal of Applied Sciences, vol. 6, pp. 1949-1960, 2009.

[27] K.M. Faraoun and A. Boukelif. "A. Neural networks learning improvement using the k-means clustering algorithm to detect network intrusions." International Journal of Computational Intelligence, vol. 3, pp. 161-168, 2006.

[28] I. Chairunnisa, I. Lukas and H.D. Widiputra. "Clustering base intrusion detection for network profiling using k-means, ecm and k-nearest neighbor algorithms." Konferensi Nasional Sistem dan Informatika, 2009, pp. 247-251.

[29] A. Ali, A. Saleh and T. Ramdan. "Multilayer perceptrons networks for an intelligent adaptive intrusion detection system." International Journal of Computer Science and Network Security, vol. 10, pp. 275-279, 2010.

[30] L. Vatisekhovich. "Intrusion detection in TCP/IP networks using immune systems paradigm and neural network detectors." XI International PhD Workshop, 2009.

[31] D.A. Zilberbrand. "Efficient Hybrid Algorithms for Plan Recognition and Detection of Suspicious and Anomalous Behavior." Ph.D. Thesis, Bar-Ilan University, Ramat-Gan, 2009.

[32] J. R. Koza. Genetic Programming. Massachusetts: MIT Press, 1992, pp. 17-120.

[33] S. J. Russell and P. Norvig. Artificial Intelligence: A Modern Approach, 2nd Ed. New Jersey: Englewood Cliffs, 2003, pp. 653-663.