Comparative Analysis of Machine Learning Methods in Anomaly-based Intrusion Detection

W.D.Y.N. Piyasinghe, K.E.G.A.P. Kadurugasyaya, K.P.D.H.M. Karunatissa,
S.L.P. Yasakethu, R.G. Ragel
Faculty of Engineering
University of Peradeniya
Peradeniya,Sri Lanka
{ yniluka, anuradhe.prabhath, harsha.mshan, lasithkethu }@gmail.com, roshanr@pdn.ac.lk

***Abstract* - Intrusions can put computer systems to a vulnerable state by compromising overall security. A lot of research studies have been done in anomaly-based intrusion detection methods using machine learning based techniques. Most of such methods can only detect known attacks while some of these algorithms have the capability of identifying novel attacks. But still there is an imbalance between the extensive amount of research on this domain and operational deployments of such systems due to the high false positive rate. In our study, we propose a comparative performance analysis of supervised, semi-supervised and unsupervised learning methods for the first time in intrusion detection domain by using several learning algorithms.**

Keywords – Anomaly-based intrusion detection, Supervised Learning, Semi-Supervised Learning, Unsupervised Learning

I. INTRODUCTION

Cyber intrusion is any unauthorized access to a network, which makes connected computers vulnerable by compromising network security and stability. Intrusion detection techniques are classified according to detection principles as misuse/signature detection, anomaly detection, and hybrid detection [1]. Signature detection compares similarities between events and signatures of known attacks. Therefore, it has lower false positive rate when detecting known attacks but fails in detecting new attacks. Anomaly detection builds a model of usage patterns describing the typical behavior of the data at the initial phase. Then new events are compared with this model, and deviations are detected [2] and flagged as attacks. This approach has the advantage of detecting previously unknown attacks but suffers from a high rate of false alarms. Hybrid detection is a combination of anomaly-based and signature-based detection to overcome drawbacks of the above mentioned methods.

Machine learning techniques are essential in building classifier models and detecting intrusions in anomaly detection systems. Supervised machine learning methods need labeled training data set for the above task, but this kind of data is difficult to obtain in real network environments since it is a time-consuming procedure. Also, there is no possible way to guarantee that the labeled data covers all possible attacks in network environment due to the dynamic behavior of network traffic [1]. Unsupervised learning methods can overcome drawbacks of supervised learning methods since there is no need for a labeled training dataset. Previous research works show that the supervised learning has high false positive rate if unknown attacks are present in the testing dataset. Unsupervised methods do not show a significant difference between known and unknown attacks but have low performance [3]. To overcome the limitations of the above methods, the recent intrusion detection research is more focused towards semi-supervised learning methods. Semi-supervised learning uses a small amount of labeled data with a large number of unlabeled data for building classifiers [4] [5].

The objective of this study is to compare the performance of supervised, semi-supervised and unsupervised machine learning techniques in anomaly based intrusion detection. For this study, we will use algorithms which have shown promising results in previous research. Further, we will select the most efficient algorithm out of the selected algorithms considering the detection accuracy and complexity. Our experiment is based on KDD Cup 1999 dataset [6]. Receiver operator characteristic (ROC) curves are used for comparing results after algorithms have been tested in a common testing framework.

II. RELATED WORK

Laskov et al. [1] have done an evaluation of several supervised and unsupervised algorithms using KDD Cup 1999 dataset. The experiment was done in two scenarios. In the first scenario, both training and testing data come from the same data distribution. Therefore only known attacks were included in the testing dataset. In the second scenario some of the attacks in the testing data are not presented in training dataset (unknown attacks for testing). Results showed that the supervised learning methods outperform unsupervised learning methods if testing data contains known attacks. The accuracy of supervised methods has a lower value if testing data includes unknown attacks. Unsupervised methods have similar performance when detecting both known/unknown

attacks. Accuracy values were alike to the performance of detecting unknown attacks in supervised learning methods. Also, non-linear methods (such as Support Vector Machines (SVM) and rule-based methods) have the best performance among selected supervised learning methods.

Zhang et al. [7] have done a study using random forest algorithm to detect intrusions without using attack-free data in the training phase. They built a framework to analyze the performance of random forest in misuse, anomaly, and hybrid detection systems and especially used outlier detection of random forest algorithm in anomaly detection framework. Sampling techniques such as cross-validation are used to increase the detection rate. The results showed that their approach achieved higher detection rate when the false positive rate is low, compared to previous unsupervised anomaly detection methods. The results also demonstrated a decrease in the detection performance when the amount of attack data is increased in the testing dataset.

Another research [8] with more focus towards comparing performances of supervised methods in intrusion detection is done against KDD Cup 1999 dataset. Support vector machine (SVM) and Neural Network (NN) algorithms are used for the experiment and performance were compared using different measures such as accuracy and false positive rate in the optimum level of those algorithms. Since data is not distributed evenly in the KDD dataset, they applied different normal data proportions for training and testing phases and obtained one average value for comparison. It has shown that SVM has superior results than NN for accuracy and false positive rate.

But there is no previous study done in comparative analyze of the performance of supervised, semi-supervised and unsupervised machine learning techniques in anomaly based intrusion detection. We focus on addressing this in our research.

III. BACKGROUND: MACHINE LEARNING METHODS

Machine learning algorithms used for our experiment are described in the following section.

A. Supervised Learning Algorithms

Support Vector Machine (SVM)

A Support Vector Machine can be used for both classification and regression purposes. The goal of SVM is to find a hyperplane that would leave the maximum distance between data points from two classes. As shown in Fig 1 hyperplane can be written as the set of points x satisfying $w^Tx + b = 0$, where the vector w is a normal vector perpendicular to the hyperplane and b is the offset of the hyperplane $w^Tx + b = 0$ from the original point along the direction of w.

Both nonlinear and linear SVM map the original feature space to a higher-dimensional feature space where the training set is separable by using kernel functions. Selecting a kernel function and tuning parameters for SVM are still trial-and-error procedure.

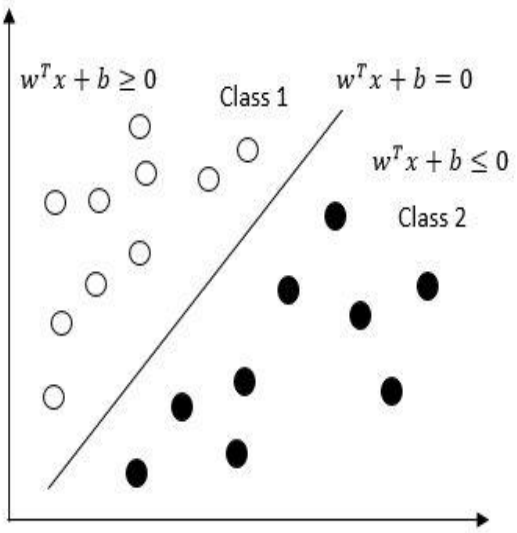

Fig. 1 Hyperplane through linearly separable classes

K-Nearest Neighbors (KNN)

K-Nearest Neighbours algorithm is a non-parametric method, which can be used for both classification and regression. Input for the algorithm is k closet training data points in the feature space. In classification, the output is a class membership, which is obtained by a majority vote of its neighbours, with the object being assigned to the class most common among its k nearest neighbours. Different distance functions are used to calculate the distance from a particular data point to its neighbours. The optimal value of k depends upon the characteristics of data. In general, larger values of k reduce the effect of noise on the classification, but make boundaries between classes less distinct. Heuristic techniques are used to find an adequate k value.

B. Semi-Supervised Learning Algorithms

One-class Support Vector Machine

One-class SVM needs to have data related to only one class for building a strong decision boundary which differentiates a particular data class from other data type. This algorithm obtains a spherical boundary, in feature space, around the data. The volume of this hypersphere is minimized, to minimize the effect of incorporating outliers in the solution. The resulting hypersphere is characterized by a center and a radius R>0 as distance from the center to (any support vector on) the boundary, of which the volume will be minimized. As shown in Fig 2, this hypersphere is used as classification

boundary for identify new data as similar or different to the training set.

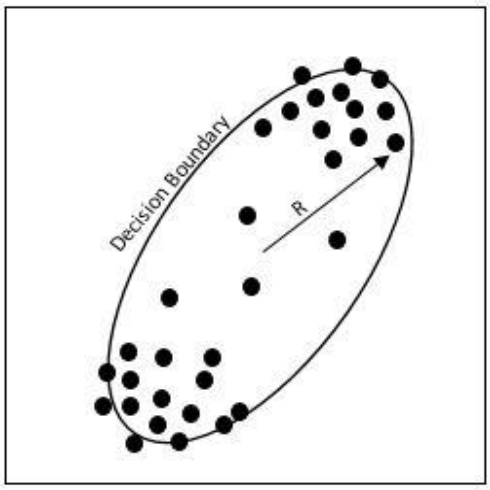

Fig. 2 Hypersphere defines the decision boundary for outliers

YATSI

YATSI classifier uses both labeled and unlabeled data in a two-stage setup for training. In the first stage, a standard classifier is trained on the available labeled training data. In the second stage, the model generated from the learning data is used to pre label all the unlabeled training data. Then these pre-labeled data are used together with labeled original training data in a weighted nearest neighbor algorithm for constantly improvement of the classifier.

C. Unsupervised Learning Algorithms

k-Means

k-Means clustering partitions the given data points into k clusters, in which each data point is more similar to its cluster centroid than to the other cluster centroids. k-Means initially creates k clusters from a set of objects so that the members of a group are more similar than in between groups. Then new objects are assigned to the cluster which is closer to the centroid using different distance metrics. The process is repeated until cluster centroids do not change. Accuracy is depended on distance metrics and cluster number k for partitioning. No evaluation method can guarantee optimal value for k.

Expectation Maximization (EM)

The EM algorithm is an iterative procedure to compute the Maximum Likelihood (ML) estimate in the presence of missing or hidden data. Each iteration of the EM algorithm consists of two processes, The E-step and the M-step. In E-step, the missing data are estimated given the observed data and current estimate of the model parameters. This is achieved using the conditional expectation, explaining the choice of terminology. In the M-step, the likelihood function is maximized under the assumption that the missing data are known. The estimate of the missing data from the E-step is used for the actual missing data. Steps are repeated until convergence.

IV. INTRUSION DATASET AND DATA PREPROCESSING

Our experiment is based on KDD Cup 1999 dataset [6], which was built using data captured in DARPA'98 IDS evaluation programme. In this dataset, a total of nine weeks raw Transmission Control Protocol (TCP) dump data was split into a seven-week training data set and remaining two weeks were turned into test data [9]. It consists of nearly 4.9 million single connection vectors which contain 41 features and labelled as normal or as an attack by specifying attack name [10]. Attacks fall into one of following categories: DoS (Denial of service), R2L (Remote to Local) – unauthorized access to a local machine from a remote system, U2R (User to Root) – Root level privileges for a normal user and Probing. There is a total of 24 attack types contain in the dataset [11]. Due to difficulties in processing such a large amount of data, we have used a kddcup_data_10_percent dataset [6] which only consists of nearly 0.49 million records.

Some characteristic of this dataset eventually leads to problems in applying machine learning algorithms for training. Certain attributes have nominal values which have to be converted into numeric representation before feeding into some algorithms. First, each nominal feature which has n possible classes was converted into binary representation by mapping i^{th} class value of the feature to j^{th} component as 1 and other positions as 0. Then the binary representation was converted into corresponding decimal value.

$$b(i) = (0,..,1,..,j,.); i \in n, j = \begin{cases} 1; if\ i^{th} class = j^{th} component \\ 0; if\ j^{th} class \neq j^{th} component \end{cases} \quad (01)$$

The dataset also has to be normalized for making all attributes proportion with one another. This will eliminate the influence of one feature over another since most of the features in the dataset have uneven distribution [9]. We used standard score formula for data normalization.

$$n(x_i) = \frac{x_i - \mu}{\sigma}\ ;\ \mu = Mean, \sigma = StandardDeviation \quad (02)$$

The dataset also has an irregular class distribution [9], which means an imbalance between normal and attack data. If classification algorithm is not robust enough for supervised learning, this imbalanced behavior leads classifier's more bias towards the majority class [12]. Therefore, the overall intrusion detection accuracy might be compromised.

Also, previous works suggest that all 41 attributes are important for identifying attacks and therefore reducing dataset based on features will result in low probability of recognizing certain attack types [13].

By considering above mentioned criteria, we build 4 types of datasets using KDD data for our experiment.

1. Training dataset for supervised learning
2. Imbalanced training dataset for unsupervised learning
3. Training dataset for semi-supervised learning
4. Testing datasets

Testing datasets were prepared to analyze the detection accuracy of previously known attacks and unknown attacks. The known attack dataset contained all the 14 different attack types which are already contained in the training datasets whereas unknown dataset contained 8 more novel attacks.

IV. Implementation

Weka machine learning environment [14] was used with some external Java based libraries for the implementation purposes.Weka has built in capability to generate receiver operating characteristic (ROC) curves for defined threshold values. Data pre-processing methods were implemented using the Java programming language.

A. Classifier models in supervised and semi-supervised learning

Separate datasets were prepared for training supervised algorithms and semi-supervised algorithms due to different requirement in Weka for the above two learning methods. Classifier models obtained by optimizing parameters of each algorithm for highest classification accuracy. 10 fold cross-validation method also used to avoid over-fitting of the classifier to training dataset.

B. Labeled clusters in unsupervised learning

Since unlabeled data is fed into algorithms in unsupervised learning, there is no explicit method to determine whether resulting clusters belongs to normal or attacks unless imbalanced dataset is used for training [18]. Thus, we labeled clusters implicitly by using the imbalanced dataset in the training phase and then testing data categorized according to the shortest Euclidian distance to any labeled cluster.

V. Results and Evaluation.

Precision (03), Recall (04) and F-Score (05) values for intrusion detection were formulated using the confusion matrix shown in Table I.

TABLE I
CONFUSION MATRIX FOR INTRUSION DETECTION

	Classified as Normal	Classified as Attack
Actual Normal	TP	FP
Actual Attack	FN	TN

TP – Classified as Normal while they actually were Normal

TN – Classified as Attack while they actually were Attack

FP – Classified as Attack while they actually were Normal

FN – Classified as Normal while they actually were Attack

$$Precision = \frac{TP}{TP + FP} \quad (03)$$

$$Recall = \frac{TP}{TP + FN} \quad (04)$$

$$F - Score = 2.\frac{Precision.Recall}{Precision + Recall} \quad (05)$$

Also performance of each machine learning method was evaluated by plotting the receiver operating characteristic (ROC) curve. It draws true positive rate of detection (TPR) against the false positive rate of detection (FPR) for various threshold values. Total area under the curve summarizes the overall detection accuracy of the intrusion detector [19]. Hence it can be used as a performance measuring parameter.

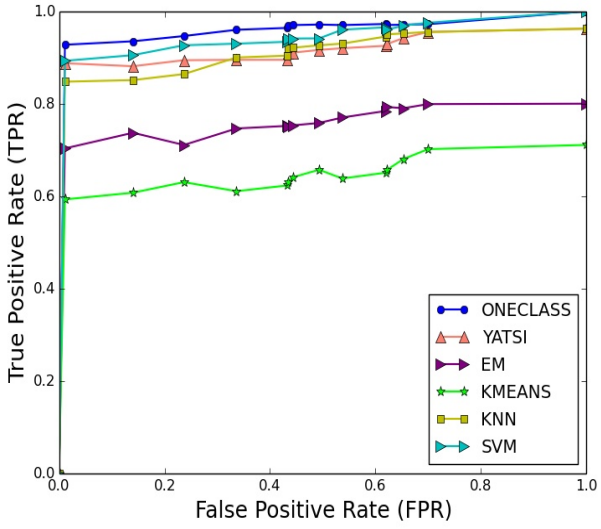

Fig. 3 ROC-curves obtained with the different machine learning methods on testing datasets with known attacks.

As shown in Fig 3, One-class SVM has the highest accuracy followed by SVM, YATSI and KNN algorithms

when detecting known attacks. But EM and k-Means algorithms have significant lower variation of detection accuracy. It depicts that both supervised (SVM, KNN), and semi-supervised (One-class SVM, YATSI) learning methods show very high accuracy than unsupervised methods (k-Means, EM) for detecting known attacks. Table II shows Precision, Recall and F-Score values for detecting known attacks by different algorithms.

TABLE II
PERFORMANCE METRICS FOR DETECTING KNOWN ATTACKS

Algorithm	Precision (%)	Recall (%)	F-Score (%)
SVM	92.8	90.7	91.7
KNN	96.2	91.7	93.9
EM	89.4	53.9	67.3
KMeans	92.9	50.1	65.1
one-class SVM	92.8	99.8	96.3
YATSI	95.0	96.9	95.9

According to Fig 4 unsupervised learning methods (k-Means, EM) show the highest accuracy for detecting unknown attacks followed by semi-supervised methods (One-class SVM, YATSI) in the second. Supervised learning methods (SVM, KNN) are not suitable in this scenario due to very low accuracy rates. The main advantage of unsupervised learning is that there is no need of labelled training data to build clusters. Variations in attributes distribution in the dataset can be used to build different clusters without using the class label. Hence, there is no significant accuracy variation for known or unknown testing data. But the quality of the training dataset directly affects the accuracy. Table III shows Precision, Recall and F-Score values for detecting unknown attacks by different algorithms.

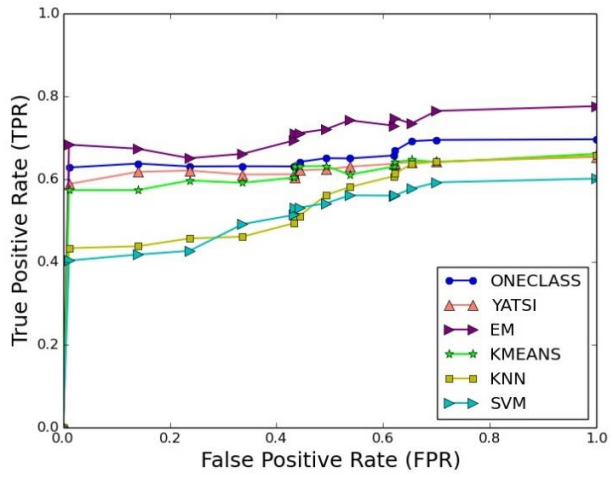

Fig. 4 ROC-curves obtained with the different machine learning methods on testing datasets with unknown attacks.

TABLE III
PERFORMANCE METRICS FOR DETECTING UNKNOWN ATTACKS

Algorithm	Precision (%)	Recall (%)	F-Score (%)
SVM	79.1	44.9	57.3
KNN	68.2	39.0	49.6
EM	81.1	69.7	75.0
KMeans	91.8	53.6	67.7
1-class SVM	89.3	59.6	71.5
YATSI	84.5	51.0	63.6

Table IV shows total detection accuracy of algorithms for known and unknown attacks. It shows that One-class SVM algorithm (semi-supervised) has highest ROC area when detecting known attacks while having a reasonable value for detecting unknown attacks. One-class SVM algorithm is trained using a dataset contain only one label. In this scenario, a dataset only containing normal data was used to build a classifier model. One-class SVM has the capability to build strong boundaries about standard data which will be used to classify other class types as anomalies.

TABLE IV
ROC AREA OF DIFFERENT ALGORITHMS

Testing Dataset	Algorithm	ROC Area(Accuracy) (%)
Known Data	SVM	95.1
	KNN	93.1
	EM	78.4
	KMeans	68.0
	One-class SVM	97.9
	YATSI	97.0
Unknown Data	SVM	54.1
	KNN	58.3
	EM	71.1
	KMeans	62.4
	One-class SVM	65.1
	YATSI	63.7

In supervised learning (SVM, KNN), the classification model is built based on different class labels of the dataset. If classifier's parameters are tuned into an optimum level, then the model has the capability of classifying known testing data with very high accuracy using previous knowledge. But accuracy rate drastically reduces for unknown test data since the supervised model does not have a prior knowledge about untrained attack types. Therefore supervised learning algorithms show high accuracy in detecting known attacks while have very low accuracy in detecting novel attacks.

V. Conclusions

In this study, we have presented an evaluation of different supervised, semi-supervised and unsupervised learning methods in intrusion detection. It demonstrates that semi-supervised and supervised learning methods have higher performance than unsupervised methods when test data does not contain unknown attacks. The detection rate of supervised methods significantly drops when testing for unknown attacks. The accuracy of unsupervised learning is not reduced by unknown attacks and it has very similar performance to semi-supervised learning methods. Semi-supervised learning methods show promising results for any type of test data. Experiment suggests that the semi-supervised learning methods show promising results in intrusion detection domain than both supervised and unsupervised learning methods. Also, One-class SVM semi-supervised algorithm has the highest overall accuracy of the experimented machine learning methods.

References

[1] Pavel Laskov, Patrick Dussel, Christin Schafer and Konrad Rieck, "Learning intrusion detection:supervised or unsupervised?," Image Analysis and Processing – ICIAP 2005, vol. 3617, pp. 50 - 57, 2005.

[2] Gustavo Nascimento, Miguel Correia, "Anomaly-based Intrusion Detection in Software as a Service," DSNW '11 Proceedings of the 2011 IEEE/IFIP 41st International Conference on Dependable Systems and Networks Workshops, pp. 19-24, 2011.

[3] C.Chen,Yunchao Gong,Yingjie Tian, "Semi-supervised learning methods for network intrusion detection," in Systems, Man and Cybernetics, 2008. SMC 2008. IEEE International Conference on , Singapore, 2008.

[4] Jyoti Haweliya,Bhawna Nigam, "Network Intrusion Detection using Semi Supervised Support Vector Machine," International Journal of Computer Applications , vol. 85, 2014.

[5] X. Zhu, "Semi Supervised Learning Literature Survey," Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005.

[6] "KDD Cup 1999 Data," 28 October 1999. [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html. [Accessed 01 April 2016].

[7] Jiong Zhang, Mohammad Zulkernine, Anwar Haque, "Random-Forests-Based Network Intrusion Detection Systems," IEEE Transactions on Systems, Man, and Cybernetics, vol. 38, no. 5, pp. 649 - 659 , 2008.

[8] Hua TANG,Zhuolin CAO, "Machine Learning-based Intrusion Detection Algorithms," Journal of Computational Information Systems, vol. 5, no. 6, pp. 1825-1831, 2009.

[9] Bertrand Portier, Jerome Froment-Curtil, "Data Mining Techniques for Intrusion Detection," in the University of Texas, Austin, 2000.

[10] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu and Ali A. Ghorbani, "A Detailed Analysis of the KDD CUP 99 Data Set," in Computational Intelligence for Security and Defense Applications, 2009. CISDA 2009. IEEE Symposium, Ottawa, 2009.

[11] L. Portnoy, "Intrusion detection with unlabeled data using clustering," Columbia University.

[12] Sumeet Dua,Xian Du, "Modeling Data with Skewed Class Distributions to Handle Rare Event Detection," in Data Mining and Machine Learning in Cybersecurity, Auerbach, 2011, p. 52.

[13] H. Gunes Kayacik, A. Nur Zincir-Heywood, Malcolm I. Heywood, "Selecting Features for Intrusion Detection:A Feature Relevance Analysis on KDD 99 Intrusion Detection Datasets," in In Proceedings of the third annual conference on privacy, security and trust, 2005.

[14] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten, "The WEKA Data Mining Software: An Update," SIGKDD Explorations, vol. 11, no. 1, 2009.

[15] T. Subbulakshmi1,A. Ramamoorthi,Dr. S. Mercy Shalinie, "Ensemble design for intrusion detection systems," International Journal of Computer science & Information Technology(IJCSIT), vol. 1, no. 1, 2009.

[16] R. Kohavi, "Wrappers for Performance Enhancement and Oblivious Decision Graphs," Thesis (Ph. D.) - Stanford University, 1995.

[17] Kurt Driessens, Peter Reutemann, Bernhard Pfahringer, Claire Leschi, "Using weighted nearest neighbor to benefit from unlabeled data," Advances in Knowledge Discovery and Data Mining, vol. 3918, pp. 60 - 69, 2006.

[18] Sumeet Dua, Xian Du, "Clustering-Based Anomaly Detection," in Data Mining and Machine Learning in Cybersecurity, Auerbach, 2011, p. 102.

[19] R.A Maxion, R.R Roberts, "Proper Use of ROC Curves in Intrusion/Anomaly Detection," 2004.

[20] Y Wang, J. Wong, A. Miner, "Anomaly intrusion detection using one class SVM," Information Assurance Workshop, pp. 358-364, 2004.