A Comparative Study of Hidden Markov Model and Support Vector Machine in Anomaly Intrusion Detection

Article · September 2013

DOI: 10.20533/jitst.2046.3723.2013.0023

Nasser Abouzakhar
University of Hertfordshire

A Comparative Study of Hidden Markov Model and Support Vector Machine in Anomaly Intrusion Detection

Ruchi Jain, Nasser S. Abouzakhar
School of Computer Science School of Computer Science
University of Hertfordshire, Hatfield, UK University of Hertfordshire, Hatfield, UK

Abstract

This paper aims to analyse the performance of Hidden Markov Model (HMM) and Support Vector Machine (SVM) for anomaly intrusion detection. These techniques discriminate between normal and abnormal behaviour of network traffic. The specific focus of this study is to investigate and identify distinguishable TCP services that comprise of both normal and abnormal types of TCP packets, using J48 decision tree algorithm. The publicly available KDD Cup 1999 dataset has been used in training and evaluation of such techniques. Experimental results demonstrate that the HMM is able to classify network traffic with approximately 76% to 99% accuracy while SVM classifies it with approximately 80% to 99% accuracy.

Keywords-Hidden Markov Model, Support Vector Machine, Distinguishable TCP Services, Anomaly Intrusion Detection

1. Introduction

The increase in the number of interconnected networks to the Internet has led to an increase in unlimited security threats and violations. As a shared resource computer networks and communication links allow unauthorized users to gain access to private information and critical resources of organizations. Therefore, information security has become a major concern to various businesses and organizations and requires an intelligent security system that can automatically detect the intrusions. An Intrusion Detection System (IDS) [5] has become popular tool for observing patterns of activities in user accounts and detect malicious behavior. Anomaly detection approach [2] is a key element of intrusion detection that attempts to evaluate the behavior of a user or system and consider intrusive or irregular activities as some deviation from normal patterns.

The HMM technique [14] has been attempted by Joshi and Phoha [6] for classifying the TCP network traffic as an attack or normal. They have taken only 12.195% of the total 41 features of the KDD Cup 1999 dataset [19] for developing anomaly intrusion detection model. The model verified that the TCP session is a normal or having anomaly with 79% accuracy. In our previous work [5], we have proposed an HMM based anomaly intrusion detection, and confirmed it's effectiveness. In this paper, the SVM technique [9], [10], [11], [12], [15], [21] is proposed for binary classification of each distinguishable TCP service. The Sequential Minimal

Optimization (SMO) algorithm [20], an implementation of Waikato Environment for Knowledge Analysis (Weka) toolkit [22], is used that supports SVM. Two Kernel functions [9], [15] including Polynomial and Radial Basis Function (RBF) are used for mapping the dataset into higher dimensional spaces. A comparison between HMM and SVM, in terms of the classification results, is presented. The remainder of this paper is organized as follows; Section 2 provides a brief explanation of the concepts of HMM. Section 3 briefly presents the basic principal of SVM Section 4 describes the methodology for designing Anomaly Intrusion Detection Model that classifies network traffic as an attack or normal using HMM and SVM. Extensive experiments based on the KDD Cup 1999 dataset are given in detail in Section 5. Finally, concluding remarks with discussions are showed in Section 6.

2. Hidden markov model

An HMM is a double embedded stochastic process consisting of an underlying stochastic process that is hidden (not observable), but can only be observed through another set of stochastic process that produces the sequence of observations [14]. HMM based applications are widely used in several different areas such as speech recognition, bioinformatics, and genomics. Srivastava et al. [18] have used HMM for identifying credit card fraud by showing that the credit card transactions with high probability is deemed by proficient HMM as normal behavior or fraudulent. Cho and Park [1] have proposed an efficient anomaly based IDS by demonstrating privilege transition flow of data by using HMM. Ourston et al. [13] present an approach to identify complex Internet attacks by using HMM. Hoang et al. [4] introduce a new method that processes sequences of system calls for anomaly detection by employing HMM technique. Lane [7] has used HMM to model human behavior. Once human behavior is correctly modeled then deviations can be detected by analyzing behavior patterns since an attacker and genuine user are not expected to have similarities in behaviors.

An HMM is characterized by $\lambda=(A,B,\pi)$, and the elements of HMM are briefly described here as follows [6], [18]: (1) N is the number of states in the model. The set of states are denoted as $S=\{S_1,S_2,S_3,..S_N\}$, where S_i, $i=1,2,3,\ldots,N$ is an individual state. The state at time t is denoted by q_t, (2) M is the number of distinct observation symbols per state. The observation symbols correspond to the physical output of the system being modeled, and the set of symbol $V=\{V_1,V_2,V_3,\ldots V_M\}$,where V_i, $i=1,2,3,\ldots,M$ is an individual symbol, (3) The state transition probability distribution is denoted by $A=[a_{ij}]$, (4) The observation symbol probability distribution in state j is denoted by $B=[b_j(k)]$, (5) The initial state probability vector is symbolized as $\pi=[\pi_i]$, and (6) A random sequence $O=\{O_1,O_2,\ldots,O_T\}$, where each observation O_t is one of the symbols from V, and T is the number of observations in the sequence.

HMM is not completely evaluated in this paper; we refer the reader to read [14] for more details.

3. Support vector machine

The SVM is a supervised learning method that deals with two-class problems, known as binary classification problems – in which the data is separated by a hyperplane defined by a number of support vectors. Support vectors are subsets of training data used to define the boundary between the two classes [15].

SVM has became the method of choice to solve difficult classification problems in a wide range of application domains such as data mining, speech emotion recognition, text categorization, and computer vision. Wang et. al [21] have proposed a new fuzzy SVM for discriminating good creditors from the bad ones by evaluating the credit score of each applicant with different memberships. Osowaski et. al [12] effectively introduced reliable heartbeat recognition by using SVM with two preprocessing methods: higher order statistics (HOS) and Hermite characterization of QRS complex of the registered electrocardiogram (ECG) waveform, for utilizing the least mean square method to optimize the weights of the weighted voting integrating scheme. Olivo et. al [11] used SVM for detecting email phishing attacks. Min and Lee [9] have applied SVM with optimal parameter values of kernel function to the bankruptcy prediction problem.

4. Methodology

This section explained a proposed methodology for anomaly intrusion detection. For experiments, the KDD Cup 1999 network traffic dataset have employed that provided by Lee and Stolfo [8]. The dataset is categorized into three divisions: training dataset, 10% of training dataset and test (corrected) dataset. It contains 41 features that represent a sequence of TCP session labelled as either normal or a member of one of the 39 attack classes in the dataset. Each attack type is grouped into four classes: Denial of Service (DoS), Probing, User to Root (U2R) and Root to Local (R2L). Based on the experiments of Shyu et al. [17], the attack types are carried out in this paper as one attack group for detecting any connections that is not normal. Among 41 features, a service feature of the dataset is analysed to assess the robustness of models.

Table 1. List of distinguishable TCP services

No.	Service Names	Type	Number of Attack TCP Packets	Number of Normal TCP Packets
1	auth	Attack/Normal	1054	2328
2	domain_u	Attack/Normal	9	57773
3	ecr_i	Attack/Normal	2808204	3456
4	finger	Attack/Normal	1874	5017
5	ftp_data	Attack/Normal	2604	38093
6	ftp	Attack/Normal	1393	3821
7	http	Attack/Normal	4045	619046
8	imap4	Attack/Normal	1066	3
9	other	Attack/Normal	16133	56520
10	private	Attack/Normal	1026978	73853
11	shell	Attack/Normal	1046	5
12	smtp	Attack/Normal	1183	95371
13	ssh	Attack/Normal	1068	7
14	telnet	Attack/Normal	2050	2227
15	urp_i	Attack/Normal	3	5375
16	X11	Attack/Normal	6	129

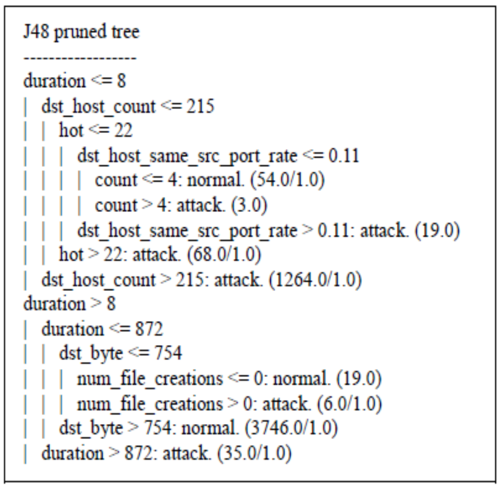
A service feature of the training dataset consists of 70 types of discrete values that denote network service on the destination [19]; these services are selected for identifying TCP packets containing both normal and an attack types. The J48 decision tree algorithm [22] of Weka is applied over each TCP service in the training dataset for obtaining decision trees. We have analysed decision tree of 16 TCP services that are considered as distinguishable TCP services containing both normal and an attack types of TCP packets. Table 1 lists the distinguishable TCP services including values, types and number of TCP packets for both normal and an attack types. Use of HMM and SVM are explained in the following sections 4.1 and 4.2 respectively.

4.1. Use of HMM in anomaly intrusion detection

There are 16 decision trees obtained for distinguishable TCP services, see Table 1. Each TCP service is associated with one decision tree that comprises of features and their values. A decision tree of ftp service is only briefly described in this section for explaining the process of selecting features and classification of their values for HMM based anomaly intrusion detection, followed by training and evaluation procedures and anomaly detection phase.

4.1.1. Feature selection and classification of their values

The decision tree of ftp service, shown in Figure 1, contains 5 levels of split on features. Each feature consists of some values that represent branches. There are total 9 numbers of leaves that described the type of TCP packet either normal or an attack.



```
J48 pruned tree
------------------
duration <= 8
|   dst_host_count <= 215
|   |   hot <= 22
|   |   |   dst_host_same_src_port_rate <= 0.11
|   |   |   |   count <= 4: normal. (54.0/1.0)
|   |   |   |   count > 4: attack. (3.0)
|   |   |   dst_host_same_src_port_rate > 0.11: attack. (19.0)
|   |   hot > 22: attack. (68.0/1.0)
|   dst_host_count > 215: attack. (1264.0/1.0)
duration > 8
|   duration <= 872
|   |   dst_byte <= 754
|   |   |   num_file_creations <= 0: normal. (19.0)
|   |   |   num_file_creations > 0: attack. (6.0/1.0)
|   |   dst_byte > 754: normal. (3746.0/1.0)
|   duration > 872: attack. (35.0/1.0)
```

Figure 1. Decision tree of 'ftp' service

As shown in Figure 1, the decision tree of ftp service is pruned and retains only those features and their values that provide statistically significant predictive power to model. The decision whether the TCP packet type is normal or an attack is based on the combined features and their values, as seen in Figure 1. We treat all the features and their values separate for reducing dimensionality of the dataset to improve the performance of model.

The values of all the features create a TCP session of the dataset; hence the feature values make a TCP session a normal or an attack one. These values represent observations that can be taken as observation sequences by classifying them with appropriate observation symbol numbers [6]. For generating observation symbols, we use Urn and Ball model of [14]. In this paper, each state has carried out to be a specific urn and the number of observation symbols per state determined the number of coloured balls exists in a particular urn. Since there are 16 distinguishable TCP services, each of the TCP service containing different types of features and their values are selected based on the decision tree.

Based on the decision tree of ftp service, showed in Figure 1, there are 7 features: (1) duration, (2) dst_host_count, (3) host, (4) dst_host_same_src_port_rate, (5) count, (6) dst_byte, and (7) num_file_creations. Among 7 features, duration feature has 4 variations of values thus, it is divided into two parts, namely, duration_1 and duration_2 and these two parts are considered as two features. The values of each feature is divided into 2 sets for reducing the infinitely large number of observation symbols per state, each set corresponds to the particular colour of the ball and the number of distinct observation symbols per state. Thus, the number of distinct observation symbols per state has 2 values in the model. Table 2 presents the actual values of selected features of one of the TCP sessions of the dataset along distinct observation symbol numbers of the corresponding values of a TCP session, given a ftp service.

Table 2. Distinct observation symbol values for 'ftp' service

Features	Value 1	Value 2
duration_1	<=8	>8
duration_2	<=872	>872
dst_host_count	<=215	>215
hot	<=22	>22
dst_host_same_src_port_rate	<=0.11	>0.11
count	<=4	>4
dst_bytes	<=754	>754
num_file_creations	<=0	>0
	Observation symbol number for value 1 and value 2	
	1	2

We give the ranges of values of each observation symbol number of selected features: (1) Observation symbol 1 has deputed to the values that are "<=8" for duration_1, "<=872" for duration_2, "<=215" for dst_host_count, "<=22" value for host, "<=0.11" for dst_host_same_src_port_rate, "<=4" for count,

"<=754" for dst_bytes, and "<=0" for num_file_creations, and (2) Observation symbol 2 has deputed to the values that are ">8" for duration_1, ">872" for duration_2, ">215" for dst_host_count, ">22" value for hot, ">0.11" for dst_host_same_src_port_rate, ">4" for count, ">754" for dst_bytes, and ">0" for num_file_creations.

The above described values have adapted as observation sequence $O=O_1,O_2,O_3,O_4,O_5,O_6,O_7,O_8$ and classified with their observation symbol numbers according to Table 2. Since the feature values represent TCP session, the observation sequences can also be classified as TCP session with type normal or an attack and divided into two types: (1) a known observation sequence that is generated from the training dataset and used for training model, and (2) a unknown observation sequence that is generated from the test dataset and used for evaluating trained model. The lengths of observation sequences are between 2 to 16 (dependent upon the number of features and their values obtained by a decision tree of a particular distinguishable TCP service). Once the observation sequences of type normal or an attack are generated, the next step is to set up the HMM parameters for training and evaluating model, explained in the following section.

4.1.2. Training and evaluation procedures

This section describes the training procedure of each TCP session of the dataset followed by initial estimation of HMM parameters, and evaluation of trained model. We use HMM tool kit of R project [16] for modifying and implementing both training and evaluation algorithms, described below.

1. Training procedure

A practical, but fundamental issue is to resolve the problem of determining the best model $\lambda=(A , B , \pi)$ that fits the known observation sequence O or maximises the conditional probability of the O given model. The most common approaches for solving learning problem are Baum-Welch Training (BWT) algorithm [1], [6], [14] and Viterbi Training (VT) algorithm (an alternative approach for model parameters estimation) [3].

The proposed approach consists of two types of model: (1) HMM for "normal" for modelling normal behaviour λ_1, and (2) HMM for an "attack" for modelling attack behaviour λ_2; these models are then assigned to each distinguishable TCP service. Initial values of HMM parameters π , A , and B are carried out to be uniformly distributed [6], [18] for each distinguishable TCP service. We present the number of hidden states N, that are 3: state S_1 , state S_2 , and state S_3. The number of observation symbols M is 2 for states S_1, S_2 and S_3. For each distinguishable TCP service, the initial values of π , A , and B for each TCP session are initialized to be the same for both HMM for "normal" and HMM for an "attack" models. Table 3 and 4 indicate the corresponding initial values of π, and A for each distinguishable TCP service.

Table 3. Initial state probability (parameter π_i of HMM) for one of the TCP session of training dataset for each distinguishable TCP service

States	Initial state probability value (π_i)
S1	0.3881
S2	0.3108
S3	0.3011

Table 4. State transition probability (parameter 'A' of HMM) for one of the TCP session of training dataset for each distinguishable TCP service

States	S1	S2	S3
S1	0.6132	0.1054	0.2814
S2	0.2681	0.5937	0.1382
S3	0.6138	0.2181	0.1681

Each distinguishable TCP service consists of HMM in which every state of the model can be reached in a single step from every other state. We given names for the states as S_1, S_2 and S_3. A model of ftp service is briefly explained in Figure 2; the connecting link indicates the state transition probability A for each state, and $O_1,O_2,O_3,O_4,O_5,O_6,O_7,O_8$ are observations (represent features) that are observable from each state. Among 16 distinguishable TCP service we describe the training process for ftp service in the following sub sections.

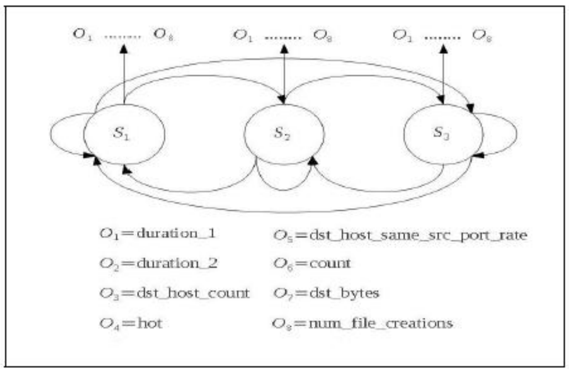


Figure 2. HMM based anomaly intrusion detection model of ftp service

a) Baum-Welch training algorithm

BWT algorithm [1], [6], [14] is used for estimating HMM parameters. For each iteration it considers all possible state paths in model of type normal or an attack for each known observation sequence $O{=}O_1{,}O_2{,}O_3{,}O_4{,}O_5{,}O_6{,}O_7{,}\text{and}\,O_8$ of type normal or an attack in terms of updating the estimated number of counts for each transition and emission. It modifies HMM parameters to get a new set of parameters until a point is reached where the sample likelihood is locally maximal using equations (1), (2), and (3):

- Re-estimating initial state probability
$$\overline{\pi_i} = \gamma_1(i) \qquad (1)$$
- Re-estimating state transition probability
$$\overline{a_{ij}} = \frac{\Sigma_{t=1}^{T-1} \varepsilon_t(i,j)}{\Sigma_{t=1}^{T-1} \gamma_t(i)} \qquad (2)$$
- Re-estimating observation symbol probability
$$\overline{B_j(V_k)} = \frac{\Sigma_{t=1,\ \text{where}O_t=Vk}^{T} \gamma(j)}{\Sigma_{t=1}^{T} \gamma_t(j)} \qquad (3)$$

Table 5 indicates the corresponding trained values of A parameter of HMM for an “attack” model for ftp service respectively.

Table 5. Trained state transition probability A for each TCP session of ftp service for HMM for an “attack” model

States	S1	S2	S3
S1	0.3341136	0.3330136	0.3328728
S2	0.3339677	0.3330734	0.3329588
S3	0.3339496	0.3330808	0.3329695

b) Viterbi training algorithm

VT algorithm [3] is an alternate approach for model parameter estimations. The most probable state path Q* for each unknown observation sequence $O{=}O_1{,}O_2{,}O_3{,}O_4{,}O_5{,}O_6{,}O_7{,}\text{and}\,O_8$ of type normal or an attack of the test dataset is calculated using Viterbi decoding [14]. This path is then used for estimating the counts of the number of transactions and symbol emissions to re-calculate the parameters using equations (4) and (5) respectively:

- Re-estimating state transition probability
$$\overline{a_{ij}} = \frac{\Sigma_{t=1}^{T-1} X_t(i) X_{t+1}(j)}{\Sigma_{t=1}^{T-1} X_t(i)} \qquad (4)$$
- Re-estimating observation symbol probability
$$\overline{b_j(V_k)} = \frac{\Sigma_{t=1_{t:O_t=Vk}}^{T} X_t(j)}{\Sigma_{t=1}^{T} X_t(j)} \qquad (5)$$

2. Evaluation procedure

For each distinguishable TCP service a problem of calculating the probability of the unknown observation sequence P=(O|λ) , where the model λ of type normal or an attack and the observation sequence of type normal or an attack $O{=}O_1{,}O_2{,}O_3{,}O_4{,}O_5{,}O_6{,}O_7{,}\text{and}\,O_8$ are given already can be solved using Forward algorithm and Backward algorithm [6], [14]. We describe the evaluation process for ftp service in the following sub sections.

a) Forward algorithm

The forward variable αt (i) [1] demonstrates probability of the partial unknown observation sequence $O{=}O_1{,}O_2{,}O_3{,}O_4{,}O_5{,}O_6{,}O_7{,}\text{and}\,O_8$, and state Si at time t , given the model λ . The main stages that are involved in Forward Procedure are described using equations (6), (7), and (8):

- Initialization
$$\alpha_t(i){=}\pi_i b_i(O_1) \quad \text{where} \quad 1{\le}i{\le}N \qquad (6)$$
- Induction
$$\alpha_{t+1}(j){=}\left[\Sigma_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(O_{t+1}) \qquad (7)$$
$$\text{where} \quad 1{\le}t{\le}T-1; \quad 1{\le}j{\le}N$$
- Termination
$$P(O|\lambda){=}\Sigma_{i=1}^{N} \alpha_T(i) \qquad (8)$$

b) Backward algorithm

The Backward variable $\beta_t(i)$ [6] defines the conditional probability of the partial observation sequence from O_{t+1} to the end, given state S_i at time t and the model λ . The main stages that are involved in Backward algorithm are described using equations (9), and (10):

- Initialization
$$B_T(i){=}1 \quad \text{where} \quad 1{\le}i{\le}N \qquad (9)$$
- Induction
$$B_t(i){=}\Sigma_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j) \qquad (10)$$
$$\text{where} \quad t{=}T-1, T-2, \ldots .1 \quad \text{and} \quad 1{\le}i{\le}N$$

4.1.3. Anomaly detection

In anomaly detection phase, we give discrete observation symbol values of the following features: (1) duration_1, (2) duration_2, (3) dst_host_count, (4) hot, (5) dst_host_same_src_port_rate, (6) count, (7) dst_bytes, and (8) num_file_creations that correspond to the unknown observation sequence of $O=O_1,O_2,O_3,O_4,O_5,O_6,O_7,\text{and } O_8$ of type normal or an attack. The Forward and Backward algorithms compute the value of given unknown observation sequence O and model λ_1 (where i=2 and represents number of models in the trained dataset). The maximum likelihood principle [18] is applied over calculated probabilities to find which model λ_1 shows the highest likelihood [6], thus:

$$i=argmax\left(PO|\lambda_i\right) \qquad (11)$$

The anomaly detection phase (recognition algorithms) verifies the TCP packet as normal or an attack by appointing the HMM of type normal or an attack.

4.2. Use of SVM in anomaly intrusion detection

For each distinguishable TCP service, the Weka toolkit [22] has been used for finding the optimal linear hyperplane in the feature or attribute space that maximally separates the two target classes; normal and attack, of KDD Cup 1999 dataset [19]. The SMO algorithm [20] of Weka that supports SVM is employed in the process of training and evaluation. This classifier uses the full training dataset and all the features present in the dataset for building a model.

Figure 3 shows optimal linear hyperplane with +1 indicating feature values of type attack, and –1 indicating feature values of type normal. The SVM endeavours to place a linear boundary between the normal and attack classes and feature values this line in such a way that the margin, the amount of space between the dotted lines, is maximized. The nearest feature values on the hyperplane define the margin that known as support vectors (green hexagons and red diamonds). Support vectors consist all the information that are essential for defining the classifier. The learning ability of SVM is independent of the numbers of features. A good introduction theory can be found in [10], [15], [20]. The hyperplane that separates the normal and attack classes is defined as:

$$w^T x+b=0 \qquad (12)$$

where $W=(w_1, w_2, w_3, .., w_n)$ are weight vectors for n features $A=(a_1, a_2, a_3, .., a_n)$, b is a scalar, and $X=(x_1, x_2, x_3, .., x_n)$ are the values of features.

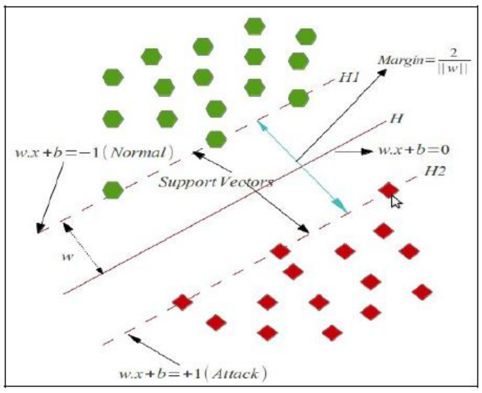


Figure 3. The optimal linear hyperplane

A function $f:\Re^n \rightarrow \{\pm 1\}$ for a classification problem can estimated by employing training dataset. Lets denote the $Class\,Attack = A$ with $x \in Z$, $y=1$ and $Class\,Normal = B$ with $X \in B$, $y=-1$. Suppose there are N training feature values $\{(x_1, y_1), (x_2, y_2), (x_3, y_3), ..., (x_N, y_N)\}$, where $x_i \in \Re^n$ and $y_i \in \{\pm 1\}$. Consider a hyperplane defined by (w, b) is written as:

$$w^T x+b \geq +1 \qquad for\,all\, x \in Z \qquad (13)$$

$$w^T x+b \leq -1 \qquad for\,all\, x \in B \qquad (14)$$

The classification of a new object x is done with

$$f_{w,b}(x)=sign(w^T x+b) \qquad (15)$$

The equation (13) and (14) are combined to give the following:

$$y(w^T x+b) \geq 1 \qquad for\,all\, x \in A \cup B \qquad (16)$$

The maximal margin classifier optimizes equation (16) by separating the data with the maximal margin hyperplane. The learning problem has been formulated as $1/2 * ||w||^2$ subject to the constraints of linear separability. The optimization is a Quadric Programming (QP) problem:

$$Minimize_{w,b}: \Phi(w)=\frac{1}{2}||w||^2 \qquad (17)$$

$$s.t.\, y(w^T x+b) \geq 1. \qquad (18)$$

To solve the QP problem, the Lagrange multiplier method is presented for transforming it into dual problem. The original QP problem is solved by the maximum of the following dual problem:

$$L(A)=\Sigma_{i=1}^{n} \Sigma_{j=1}^{n} A_i A_j y_i y_j < x_i, x_j > \qquad (19)$$

$$s.t.\, \Sigma_{j=1}^{n} A_j y_j = 0 \qquad (20)$$

where A is a Lagrange multiplier. The optimal discriminant function is expressed as:

$$f_{w,b}(x)=sign(\Sigma_i^n A_i y_i (x_i . x)+b) \qquad (21)$$

In applications, two parameters are decided; the kernel function and the margin that separates the data. There are 2 SVM kernel functions [9], [15] used for classification problem at the training time, described below.

- ***RBF:*** The RBF can be defined as;

$$K(a,b)=\exp\left(\frac{||a-b||^2}{2\sigma^2}\right) \qquad (22)$$

where, σ is a constant.

- ***Polynomial Kernel:*** The Polynomial kernel is a non-stationary kernel. Polynomial kernels are well suited for problems where all the training data is normalized.

$$K(a,b)=(a^T b+c)^d, d=1,2,.. \qquad (23)$$

Adjustable parameters are the constant term c and the polynomial degree d.

5. Experiments and Results

The evaluation metrics uses the following steps to evaluate the performance of HMM and SVM based anomaly intrusion detection models [2]:

• True Positive (TP): A TP occurs when actual class of test instance is positive and classified correctly as positive.

• False Positive (FP): A FP occurs when actual class of test instance is negative and classified incorrectly as positive.

• True Negative (TN): A TN occurs when actual class of test instance is negative and classified correctly as negative.

• False Negative (FN): A FN occurs when actual class of test instance is positive and classified incorrectly as negative.

• Detection rate: It is the proportion of the total number of predictions that has been detected accurately.

$$Detection\,rate = \frac{TP+TN}{TP+FN+TN+FP}$$

- Error rate: It is the proportion of the total number of predictions that are incorrect.

$$Error\ Rate = 1-Accuracy$$

- Precision: It is the proportion of the accuracy that describes a prediction of a specific class such as positive or negative.

$$Precision\,for\,positive = \frac{TP}{TP+FP}$$

$$Precision\,for\,negative = \frac{TN}{TN+FN}$$

- Recall: It is the proportion of actual positive or negative instances that are correctly predicted as positive or negative instances.

$$Recall\,for\,positive = \frac{TP}{TP+FN}$$

$$Recall\,for\,negtive = \frac{TN}{TN+FP}$$

- F-measure: It is a measure of combined precision and recall.

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

The 16 distinguishable TCP services are analysed in terms of detection rate, error rate, precision, recall and f-measure. Out of 16 TCP services, the classification results of 6 TCP services are presented in this section due to their evaluation of high positives rates and low negative rates, and the remaining 10 services are discarded. Table 6 shows the comparison of standard evaluation terms for HMM.

Table 6: Comparison of standard evaluation terms for HMM

Service	Detection rate	Error rate	Class	Preci-sion	Recall	F-measure
auth I	0.96	0.04	Attack	0.901	1	0.948
			Normal	1	0.936	0.967
auth II	0.88	0.12	Attack	0.918	0.823	0.868
			Normal	0.865	0.939	0.900
ecr_i	0.99	0.01	Attack	0.999	0.999	0.999
			Normal	0.815	0.972	0.886
finger	0.96	0.04	Attack	0.914	1	0.955
			Normal	1	0.945	0.972
http	0.98	0.02	Attack	0.848	0.872	0.860
			Normal	0.993	0.992	0.993
private	0.88	0.12	Attack	0.883	0.981	0.930
			Normal	0.915	0.604	0.728
telnet I	0.80	0.20	Attack	0.798	0.996	0.886
			Normal	0.931	0.169	0.287
telnet II	0.76	0.24	Attack	0.756	0.998	0.860
			Normal	0.977	0.150	0.261
telnet III	0.94	0.06	Attack	0.966	0.980	0.973
			Normal	0.568	0.427	0.49

Due to large memory usage in SVM, the TCP services ecr_i and private's datasets were partitioned into 10 subsets that were then individually used as a "full training set" using both Polynomial and RBF kernels. The rest of the services did not have memory problems so the datasets were trained without portioning. Table 7 presents the comparison of standard evaluation terms for SVM.

Table 7. Comparison of standard evaluation terms for SVM using Polynomial kernel and RBF kernel

Service	Detection rate	Error rate	Class	Precis-ion	Recall	F-measure
auth	0.95	0.05	Attack	0.982	0.902	0.940
			Normal	0.936	0.989	0.962
ecr_i	0.99	0.01	Attack	1	1	1
			Normal	0.7	0.821	0.755
finger	0.98	0.02	Attack	1	0.966	0.983
			Normal	0.978	1	0.989
http	0.99	0.01	Attack	0.974	0.919	0.946
			Normal	0.996	0.999	0.997
other	0.80	0.20	Attack	0.974	0.815	0.888
			Normal	0.168	0.631	0.266
private	0.86	0.14	Attack	0.984	0.846	0.910
			Normal	0.540	0.931	0.68

As statistical results indicate, average detection rate for HMM and SVM are 90.55 and 92.83, respectively. Average false alarm rate obtained was 0.09 and 0.07 in HMM and SVM respectively. As the results show, SVM performs almost similar in detecting accuracy rate and false alarm rate to HMM. The Receiver Operating Characteristics (ROC) curves [16] are then employed for representing relation between True Positive Rate (TPR) and False Positive Rate (FPR) of distinguishable TCP services for HMM only, see Figure 4. The TPR is TP divided by the total number of positives, which are TP + FN. The FPR is FP divided by the total number of negatives, FP + TN [2].

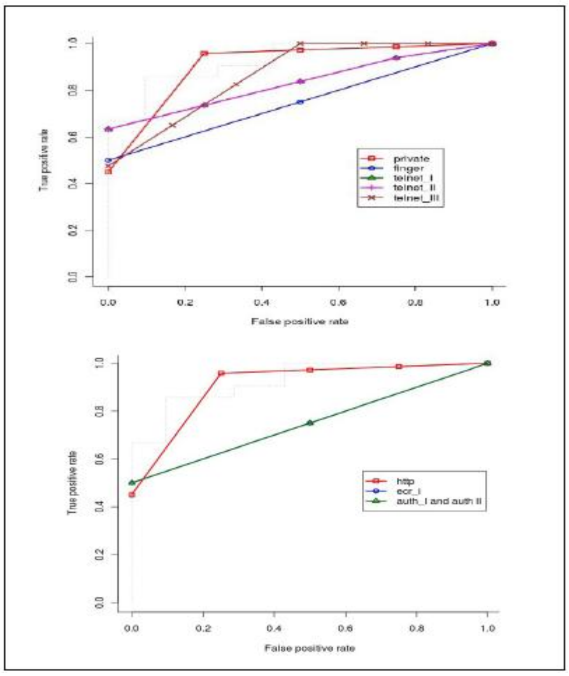


Figure 4. ROC Curves of private, finger, telnet, http, ecr_i and auth services for HMM

6. Conclusions

This paper presents a comparative study of HMM and SVM in anomaly intrusion detection. A new method of identifying distinguishable TCP services using J48 decision tree algorithm is introduced. In the HMM based anomaly intrusion detection, each of distinguishable TCP service is associated with a decision tree that consists of features and their values presented in the KDD Cup 1999 dataset. These features are then selected for training HMM model for each TCP session of the dataset using BWT and VT algorithms. Evaluation of trained model is performed with Forward and Backward algorithms. In the SVM case, SMO algorithm of Weka is used with kernel functions at the time of training and evaluation of SVM model of each TCP service.
Both HMM and SVM methods show similar results with SVM showing marginally improved anomaly detection. However, SVM based training requires all the training data set to be applied at a time. This will preclude large training dataset. HMM based training is per TCP session and there are no such restrictions.

Acknowledgements

The authors wish to acknowledge Dr. Peter lane of University of Hertfordshire with help of Weka Tools, Romil Jain of Santa Clara University with background of HMM and SVM.

References

[1] Cho, B.S. and Park, 1.H. (2003), 'Efficient anomaly detection by modeling privilege flows using hidden Markov model', Computers and Security, 22(1), pp. 45-55.

[2] Carter, E. (2002), Intrusion detection systems, Cisco Press, Indianapolis.

[3] Gernot, A.F. (2008), Markov Models for pattern recognition: From Theory to Applications, Springer, Heidelberg.

[4] Hoang, XD., Hu, 1. and Bertok, P. (2003), 'A multilayer model for anomaly intrusion detection using program sequences of system calls', The 11th IEEE International Conference on Networks, pp. 531-536.

[5] Jain, R. and Abouzakhar, N.S. (2012), 'Hidden Markov Model Based Anomaly Intrusion detection', The 7th International Conference for Internet Technology and Secured Transactions, pp. 528-533.

[6] Joshi, S.S. and Phoha, V.V. (2005), 'Investigating Hidden Markov Models Capabilities in Anomaly Detection', In proceedings of 43rd ACM Southeast Conference, 1, pp. 98-103.

[7] Lane, T. (1999), 'Hidden Markov Models for Human/Computer Interface Modeling, In Computer

Engineering. In Proceedings of the IJCAI-99 Workshop on Learning About Users, pp. 35-44.

[8] Lee, W. and Stolfo, SJ. (2000) 'A Framework for Constructing Features and Models for Intrusion Detection Systems', ACM Transactions on Information and System Security, 3(4), pp. 227-261.

[9] Min, J.H. and Lee, Y.C. (2005), 'Bankruptcy prediction using support vector machine with optimal choice of kernel function parameters', Expert Systems with Applications, pp. 603-614.

[10] Mulay, S.A., Devale, P.R. and Gajre, G.V. (2010), 'Intrusion Detection System using Support Vector Machine and Decision Tree', International Journal of Computer Applications, 3(3), pp. 40-43.

[11] Olivo, C.K., Santin, A.O. and Oliveria, L.S. (2011), 'Obtaining the threat model for e-mail phishing', Applied Soft computing, pp. 1-8.

[12] Osowski, S., Hoai, L.T. and Markiewicz, T. (2004), 'Support Vector Machine-Based Expert System for Reliable Heartbeat recognition', IEEE Transactions on Biomedical Engineering, 51(4), 2004, pp. 582-589.

[13] Ourston, D., Matzner, S., Stump, W. and Hopkins, B. (2002) 'Application of Hidden Markov Models to Detecting Multi-stage Network Attacks', In proceedings of the 36th Hawaii International Conference on System Sciences, 9, pp. 334-344.

[14] Rabiner, L.R. (1989) 'A tutorial on hidden Markov models and selected applications in speech recognition', In proceedings of the IEEE, 77(2), pp. 257 -286.

[15] Raju, E. and Sravanthi K. (2013), 'Network intrusion detection using Support Vector Machines', International Journal of Computer Science and Management Research, 2(1), pp 1314-1319.

[16] R project, The R Project for Statistical Computing, Available at: <http://www.r-project.org/> [Accessed March 6, 2012].

[17] Shyu, M.L., Sarinnapakorn, K., Kuruppu-Appuhamilage, I., Chen, S.c., Chang, L.w. and Goldring, 1. (2005) 'Handling Nominal Features in Anomaly Intrusion Detection Problems', The 15th International Workshop on Research Issues on Data Engineering (RIDE), in conjunction with The 21st International Conference on Data Engineering, pp. 55-62.

[18] Srivastava, A, Kundu, A, Sural, S. and Majumdar, A.K. (2008), 'Credit Card Fraud Detection Using Hidden Markov Model', IEEE Transactions on Dependable and Secure Computing, 5(1), pp. 37-47.

[19] The UCI KDD Archive, KDD cup 1999 data, http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html [Accessed December 16, 2011].

[20] Vapnik, V.N. (1995), The Nature of Statistical Learning Theory, 1st ed., Springer-Verlag, New York.

[21] Wang, Y., Wang, S. and Lai, K.K. (2005), 'A new fuzzy support vector machine to evaluate credit risk', IEEE Transactions on Fuzzy Systems, 13(6), pp. 820-831.

[22] Weka, Data Mining Software in Java, Available at: <http://www.cs.waikato.ac.nz/ml/weka/> [Accessed December 15, 2011].