Hidden Markov Model Based Anomaly Intrusion Detection

Ruchi Jain
Department of Computer Science
University of Hertfordshire
Hatfield, UK

Nasser S. Abouzakhar
Department of Computer Science
University of Hertfordshire
Hatfield, UK

Abstract—**This paper aims to investigate and identify distinguishable TCP services, that comprise of both attack and normal types of TCP packets, using J48 decision tree algorithm. A predictive model capable of discriminating between normal and abnormal behavior of network traffic is developed by integrating Hidden Markov Model (HMM) technique with anomaly intrusion detection approach for each distinguishable TCP service. The model has been trained for each TCP session of the KDD Cup 1999 dataset using Baum-Welch training (BWT) and Viterbi training (VT) algorithms. Evaluation of the developed HMM model is performed using Forward and Backward algorithms. Results show that the proposed model is able to classify network traffic with approximately 76% to 99% accuracy. The overall performance of model is measured using standard evaluation method ROC curves.**

Keywords-Hidden Markov Model, Distinguishable TCP services, Anomaly intrusion detection

I. 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) [1] has become popular tool for observing the behavior pattern of an intruder and detects unauthorized activities in user accounts. Anomaly detection approach [3] 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 applied to anomaly detection by Joshi and Phoha [9] 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 and the model verified that the TCP session is a normal or having anomaly with 79% accuracy.

In this paper, we introduce a novel approach for identifying distinguishable TCP services using J48 decision tree algorithm an implementation of Waikato Environment for Knowledge Analysis (Weka) [5]. Each distinguishable TCP service is associated with a decision tree containing features and their values present in KDD Cup 1999 dataset [19]; these features are then selected for developing HMM based anomaly intrusion detection model. Training for developed model has performed using BWT algorithm [4], [14] and VT algorithm [6]. Performance of trained model is then evaluated using Forward algorithm and Backward algorithm [9].

The rest of the paper is organized as follows: Section II provides a brief explanation of the concepts of HMM. Section III describes the proposed methodology for designing an HMM based anomaly intrusion detection model that classifies network traffic as an attack or normal. It provides the description of the KDD Cup 1999 dataset, distinguishable TCP services, feature selection and classification of their values, training and evaluation processes, and anomaly detection phase of the model. Detailed experimental results are explained in Section IV. Finally, some concluding remarks with discussions are showed in Section V.

II. HMM BACKGROUND

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 [4] 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. [8] introduce a new method that processes sequences of system calls for anomaly detection by employing HMM technique. Lane [10] 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 [9],

[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, $i=1,2,3,...,N$ 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 $V=\{V_1,V_2,V_3,...V_M\}$, where $V_i, i=1,2,3,...,M$ 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 $O=\{O_1,O_2,...,O_T\}$, 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.

III. USE OF HMM IN ANOMALY INTRUSION DETECTION

This section introduces a use of HMM for anomaly intrusion detection. For experiments, we have employed KDD Cup 1999 network traffic dataset provided by Lee and Stolfo [11]. 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) [2]. 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 model, described in the following section.

A. Identify Distinguishable TCP Services

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 is applied over each TCP service in the training dataset for obtaining decision trees [5]. Each TCP service is associated with one decision tree that comprises of features and their values. 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. These distinguishable TCP services are then selected for classifying more than one feature and their value. Table I lists the distinguishable TCP services including values, types and number of TCP packets for both normal and an attack types.

TABLE I. 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

There are 16 decision trees obtained for distinguishable TCP services; a decision tree of ftp service is only briefly described in Section B for explaining the process of selecting features and classification of their values.

B. Feature Selection and Classification of their Values

The decision tree of ftp service, shown in Fig. 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 Fig. 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, refer to read [7] for understanding IF THEN classification rules. 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 [9]. 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 Fig. 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 II 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 II. 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 II. 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.

C. *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 [15] for modifying and implementing both training and evaluation algorithms, described in sub sections 1 and 2.

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 BWT algorithm [4], [14] and VT algorithm (an alternative approach for model parameters estimation) [6].

The proposed approach consists of two types of model: (1) HMM for "normal" for modelling normal behavior λ_1, and (2) HMM for an "attack" for modelling attack behavior 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 [11], [20] 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 III and IV

indicate the corresponding initial values of π, and A for each distinguishable TCP service.

TABLE III. INITIAL STATE PROBABILITY (PARAMETER 'Π' 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 IV. 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 Fig. 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.

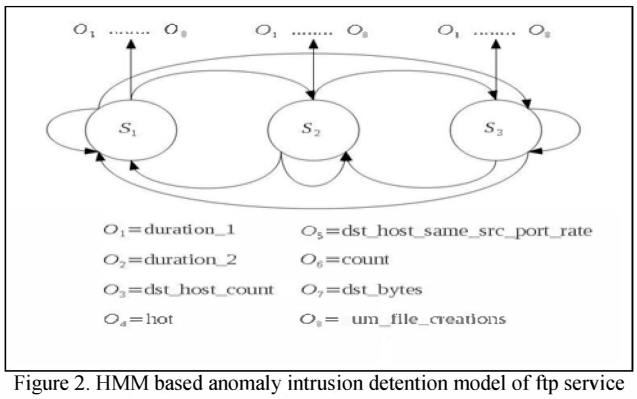

Figure 2. HMM based anomaly intrusion detention model of ftp service

Among 16 distinguishable TCP service we describe the training process for ftp service in the following sub sections.

a) Baum-Welch training algorithm

BWT algorithm [4], [9], [14] is a 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 $O=O_1, O_2, O_3, O_4, O_5, O_6, O_7$, 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

$$\overline{\pi_i} = \gamma_1(i) \quad (1)$$

- Re-estimating state transition probability

$$\overline{a_{ij}} = \frac{\Sigma_{t=1}^{T-1} \varepsilon_t(i,j)}{\Sigma_{t=1}^{T-1} \gamma_t(i)} \quad (2)$$

- Re-estimating observation symbol probability

$$\overline{B_j(V_k)} = \frac{\Sigma_{t=1,\ where O_t = Vk}^{T} \gamma(j)}{\Sigma_{t=1}^{T} \gamma_t(j)} \quad (3)$$

Table V indicates the corresponding trained values of A parameter of HMM for an "attack" model for ftp service.

TABLE V. 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 [6] is an alternate approach for model parameter estimations. The most probable state path Q^* for each unknown observation sequence $O=O_1, O_2, O_3, O_4, O_5, O_6, O_7$, 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

$$\overline{a_{ij}} = \frac{\Sigma_{t=1}^{T-1} X_t(i) X_{t+1}(j)}{\Sigma_{t=1}^{T-1} X_t(i)} \quad (4)$$

- Re-estimating observation symbol probability

$$\overline{b_j(V_k)} = \frac{\Sigma_{t=1_{t:O_t=Vk}}^{T} X_t(j)}{\Sigma_{t=1}^{T} X_t(j)} \quad (5)$$

2) Evaluation Procedure

For each distinguishable TCP service a problem of calculating the probability of the unknown observation

sequence $P=(O|\lambda)$, where the model λ of type normal or an attack and the observation sequence of type normal or an attack $O=O_1,O_2,O_3,O_4,O_5,O_6,O_7,O_8$ are given already, can be solved using Forward algorithm and Backward algorithm [9], [14]. We describe the evaluation process for ftp service in the following sub sections.

a) Forward algorithm

The forward variable $\alpha_t(i)$ [4] 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 S_i 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

$$\alpha_1(i)=\pi_i b_i(O_1) \text{ where } 1\le i\le N \quad (6)$$

- Induction

$$\alpha_{t+1}(j)=\left[\Sigma_{i=1}^{N}\alpha_t(i)a_{ij}\right]b_j(O_{t+1}) \quad (7)$$
$$where \quad 1\le t\le T-1;\ 1\le j\le N$$

- Termination

$$P(O|\lambda)=\Sigma_{i=1}^{N}\alpha_T(i) \quad (8)$$

b) Backward algorithm

The Backward variable $\beta_t(i)$ [9] 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

$$B_T(i)=1 \quad \text{where} \quad 1\le i\le N \quad (9)$$

- Induction

$$B_t(i)=\Sigma_{j=1}^{N}a_{ij}b_j(O_{t+1})\beta_{t+1}(j) \quad (10)$$
$$where \quad t=T-1,T-2,T-3\ldots.1 \quad \text{and} \quad 1\le i\le N$$

D. Anomaly Detection Phase

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 $O_1,O_2,O_3,O_4,O_5,O_6,O_7,\text{and}O_8$ of type normal or an attack. The Forward and Backward algorithms compute the value of given unknown observation sequence O and model λ_i (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_i shows the highest likelihood [9], thus:

$$i=argmax\left(P(O|\lambda_i)\right) \quad (11)$$

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.

IV. EXPERIMENTS AND RESULTS

The performance of HMM based anomaly intrusion detection model is evaluated in terms of overall detection accuracy rate, error rate, precision, recall and f-measure [16]. The overall detection accuracy rate is the fraction of the total number of predictions that has been detected accurately. The overall error rate represents the fraction of the total number of predictions that are incorrect. We have analyzed the above mentioned terms for 16 distinguishable TCP services from which the classification results of 6 TCP services are presented in table VI.

TABLE VI. COMPARISON OF STANDARD EVALUATION TERMS

Service	Detection rate	Error rate	Class	Precision	Recall	F-measure
auth 1	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	0.487

The Receiver Operating Characteristic (ROC) [15] is used for representing relation between True Positive Rate (TPR) and False Positive Rate (FPR) for distinguishable TCP services (Table VI), see Fig. 3. The TPR presents the fraction of actual attack instances that are correctly predicted as attack instances, whereas the FPR accounts for the fraction of actual normal instances that are incorrectly predicted as an attack instances [1].

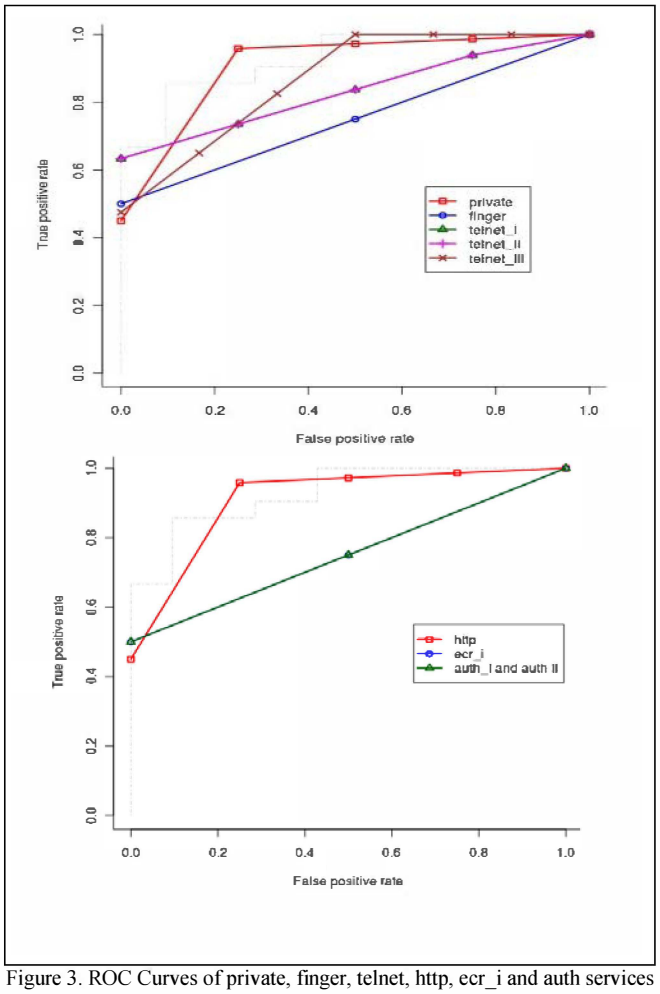

Figure 3. ROC Curves of private, finger, telnet, http, ecr_i and auth services

V. CONCLUSIONS AND DISCUSSIONS

This paper presents a new method of identifying distinguishable TCP services using J48 decision tree algorithm. 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 based anomaly intrusion detection model for each TCP session of the dataset using BWT and VT algorithms. Evaluation of trained model is performed with Forward and Backward algorithms. The results show that the HMM technique can be applied to anomaly intrusion detection using number of features and their values present in each of decision tree. The proposed model leads to a substantial reduction of the error rates with 1% to 14% and a significant increase in detection rates with 76% to 99% accuracy.

ACKNOWLEDGMENT

The authors wish to acknowledge Dr. Peter lane of UOH with help of WEKA Tools and Romil Jain with background of HMM.

REFERENCES

[1] Abouzakhar, N.S., Chen, H. and Christianson, B., 2011. An Enhanced Fuzzy ARM Approach for Intrusion Detection. International Journal of Digital Crime and Forensics, 3(2), pp. 41-61.

[2] Altwaijry, H. and Algarny, S., 2011. Multi-Layer Bayesian Based Intrusion Detection System. In proceedings of the World Congress on Engineering and Computer Science, II.

[3] Carter, E., 2002. *Intrusion detecetion systems*. Indianapolis, IN: Cisco Press.

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

[5] Frank, E., Hall, M., Holmes, G., Kirkby, R., Bernhard, P., Witten, I.H. and Trigg, L. 2006. Weka: A Machine Learning Workbench for Data Mining. The Data Mining and Knowledge Discovery Handbook, pp. 1305-1314.

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

[7] Halpin, T., 2009. Selected Readings on Database Technologies and Applications. Hershey: IGI Global.

[8] Hoang, X.D., Hu, J. and Bertok, P., 2003. A multi-layer model for anomaly intrusion detection using program sequences of system calls. The 11th IEEE International Conference on Networks, pp. 531 –536.

[9] 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.

[10] 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.

[11] Lee, W. and Stolfo, S.J., 2000. A Framework for Constructing Features and Models for Intrusion Detection Systems. ACM Transactions on Information and System Security, 3(4), pp. 227–261.

[12] Olson, D.L. and Delen, D., 2008. Advanced data mining techniques. Heidelberg: Springer.

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

[16] Singh, S., Tu, H., Donat, W., Pattipati, K. and Willett, P., 2009. Anomaly detection via feature-aided tracking and hidden Markov models. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, 39(1), pp. 144–159.

[17] Shyu, M.L., Sarinnapakorn, K., Kuruppu-Appuhamilage, I., Chen, S.C., Chang, L.W. and Goldring, T., 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. [online] Available at: <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html> [Accessed December 16, 2011].