Multilayer Intrusion Detection System Using Conditional Random Fields

Anuja S. Mukane, Bharati P. Vasgi

Abstract—A number of methods and frameworks have been proposed and many systems have been built to detect intrusions since 1980s. Intrusion Detection Systems are now an essential component in the overall network and data security arsenal. With the rapid advancement in the network technologies including higher bandwidths and ease of connectivity of wireless and mobile devices, the focus of intrusion detection has shifted from simple signature matching approaches to detecting attacks based on analyzing contextual information which may be specific to individual networks and applications. As a result, anomaly and hybrid intrusion detection approaches have gained significance. Intrusion detection faces a number of challenges: an intrusion detection system must reliably detect malicious activities in a network and must perform efficiently to cope with large amount of network traffic. In this paper, an attempt has been made to adress Accuracy and Efficiency of the system by using Conditional Random Fields and Multilayer Approach. It is demonstrated that high attack detection accuracy can be achieved by using Conditional Random Fields with Multilayer Approach. Inrusion detection tests are conducted for individual layer as well as integrated Multilayer IDS using KDD 99 test data. Test results show that the present system performs better than the other well-known methods such as decision trees and the naive Bayes.

Index Terms— Attack Patterns, CRF, Intrusion Detection, KDD 99 Cup Data, Multilayer, Training, Testing.

1 Introduction

In order to launch an attack, an attacker often follows a sequence of events. The events in such a sequence are highly correlated and long range dependencies exist between them. Further, in order to prevent detection, the attacker can also hide the individual events within a large number of normal events. As a result, considering the events in isolation affects classification and results in a large number of false alarms. Additionally, the individual events themselves are vector quantities and consist of multiple features which are monitored continuously. These features are also highly correlated and must not be analyzed in isolation. In order to operate in high speed networks, present anomaly based systems consider the events individually, thereby, discarding any correlation between the sequential events. In cases when the present systems consider a sequence of events, they monitor only one feature, ignoring others, which results in a poor model. Hence, an efficient intrusion detection frameworks and methods which consider a sequence of events and analyzemultiple features without assuming any independence among the features are used.

In the present system, Conditional Random Fields with multilayer approach used to build Intrusion Detection System that is effective in detecting a wide variety of attacks. In the present work CRF approach is effectively used for attack pattern recognition. Layered Framework for building intrusion detection systems has been introduced which can detect a wide variety of attacks reliably and efficiently when compared to the traditional network intrusion detection systems. In the present layered framework, a number of subsystems separately trained with KDD'99 training data and sequentially arranged sub systems in order to decrease the number of false alarms and increase the attack detection coverage.

Layered intrusion detection system with four class of attacks such as Probe, DoS, R2L and U2R as a separate layer integrated sequentially are shown in Fig.1. In this system, individual layer is trained with KDD training data with manually selected feature for each layer and attack patterns are identified for each layer using CRF approach. During testing using KDD test data each layer blocks the attacks corresponding to their class and passes the normal data to the next layer making the system more efficient.

2 Description of KDD cup 1999 Data

KDD cup 1999 Intrusion Detection data-set is a version of the 1998 DARPA intrusion detection evaluation program, prepared and managed by the MIT Lincoln Labs. The data set contains about five million connection records as the training data and about two million connection records as the test data. In our experiments, the ten percent of the total training data and ten percent of the test data (with corrected labels) which are provided separately are used. This leads to 494,020 training and 311, 029 test instances.

Each record in the data set represents a connection between two IP addresses, starting and ending at some well defined times with a well defined protocol. Further, with 41 different features, every record represents a separate connection and, hence in the experiments, every record is considered to be independent of every other record.

The training data is either labeled as normal or as one of the 24 different kinds of at-tack. All of the 24 attacks can be grouped into one of the four classes; Probe, Denial of Service

- *Anuja S. Mukane is currently pursuing masters degree program in information technology, in Sinhgad College of Engineering, Pune, Pune University, India. E-mail: anumukane@gmail.com*
- *Bharati P. Vasgi is currently working as Assistant Professor in Sinhgad College of Engineering, Pune, Pune University, India. E-mail: bharativasgi@gmail.com*

(DoS), unauthorized access from a remote machine or Remote to Local (R2L) and unauthorized access to root or User to Root (U2R). Similarly the test data is also labeled as either normal or as one of the attacks belonging to the four attack classes. It is important to note that the test data includes specific attacks which are not present in the training data. This makes the intrusion detection task more realistic.

2.1 Attacks in KDD Training andTesting Data Set

Training data includes the following attacks: back, buffer over flow, ftp write , guess-passwd , imap, ipsweep, land, loadmodule, multihop, neptune, nmap, perl, phf, pod, portsweep , rootkit, satan, smurf, spy, teardrop, warezclient, warezmaster.

Test data includes specific attacks which are not present in the training data are: snmpgetattack ,mailbomb , snmpguess , mscan , apache2, httptunnel , pro-cesstable , xterm , saint.

2.2 Some Records in KDD99 Data Set

Two nos. of data records/ instances are shown below as an example with 41 feature values in each record.

15, tcp, smtp, SF, 1855, 335, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0.00, 0.00, 0.00, 0.00, 1.00, 0.00, 0.00, 233, 143, 0.61, 0.02, 0.00, 0.00, 0.00, 0.00, 0.01,0.01, normal.

0, tcp, private, S0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 237, 8, 1.00,1.00, 0.00, 0.00, 0.03, 0.07, 0.00, 255, 8, 0.03, 0.07, 0.00, 0.00, 1.00, 1.00, 0.00, 0.00,neptune.

3 Layered Approach for Intrusion Detection

The Layered Intrusion Detection System represents a sequential Layered Approach and is based on ensuring availability, confidentiality, and integrity of data and (or) services over a network. Fig.1 gives a generic representation of the framework. The goal of using a layered model is to reduce computation and the overall time required to detect anomalous events. The time required to detect an intrusive event is significant and can be reduced by eliminating the communication overhead among different layers. This can be achieved by making the layers autonomous and self sufficient to block an attack without the need of a central decision maker.

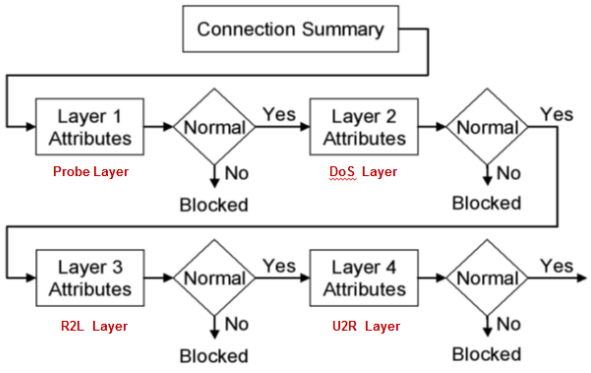

Fig. 1 IDS Layer Representation

Every layer in the LIDS framework is trained separately and then deployed sequentially. Four layers are defined which correspond to the four attack groups mentioned in the data set. They are Probe layer, DoS layer, R2L layer, and U2R layer. Each layer is then separately trained with a small set of relevant features. Feature selection is significant for Layered Approach and discussed in the next section. In order to make the layers independent, some features may be present in more than one layer. The goal is to improve the speed of operation of the system. Hence, it achieved by implementing the LIDS and selecting a small set of features for every layer rather than using all the 41 features. This results in significant performance improvement during both the training and the testing of the system.

4 CRF Approach

KDD 1999 data set described in section II is used in the present work. Conventional methods, such as decision trees and naive Bayes, are known to perform well in such an environment; however, they assume observation features to be independent. Conditional random fields is used which can capture the correlations among different features in the data and hence perform better when compared with other methods. The KDD 1999 data set represents multiple features, a total of 41, for every session in relational form with only one label for the entire record. Fig.2 represents how conditional random fields can be used for detecting network intrusions. To manage complexity and improve systems performance, integration of the layered approach with the Conditional Random Fields is done to build a single system which is more efficient and more effective.

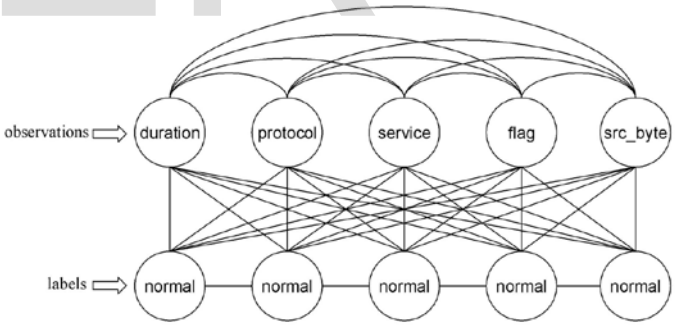

Fig.2 Conditional random fields for detecting network intrusions

In the Fig.2, observation features `duration', `protocol', `service', `flag' and `source bytes' are used to discriminate between attack and normal events. The features take some possible value for every connection which are then used to determine the most likely sequence of labels. During training, feature weights are learnt and during testing, features are evaluated for the given observation which is then labeled accordingly. It is evident from the Fig.2 that every input feature is connected to every label which indicates that all the features in an observation determine the final labeling of the entire sequence. Thus, a conditional random field can model dependencies among different features in an observation. Present intrusion detection systems do not consider such relationships. Our first goal is to improve the attack detection accuracy.

Conditional random fields improve the attack detection accuracy particularly for the U2R attacks. They are also effective in detecting the Probe, R2L and the DoS attacks.

However, in this project work the CRF approach is used to find out the attack patterns from the KDD'99 training data set and subsequently the same are used during testing on test data set. In particular, conditional probability i.e. probability of attack as well as probability of normal are calculated for given set of feature values from training data set and accordingly attack patterns are identified.

5 Feature Selection

Given the data, four layers are selected corresponding to the four attack groups (Probe, DoS, R2L, and U2R). Attacks belonging to different classes are different and, hence for better attack detection, it becomes necessary to consider them separately. As a result, in layered system, every layer is trained separately to optimally detect a single class of attack. Therefore different features are selected for different layers based upon the type of attack the layer is trained to detect.

The Probe layer is optimally trained to detect only the Probe attacks. Hence, only the Probe attacks and the normal instances from the audit data is used to train this layer. Other layers can be trained similarly. Note that, different features are selected to train different layers in our framework. Hence, domain knowledge is used to select features for all the four attack classes. Approach for selecting features for every layer and why some features were chosen over others is explained below[1].

5.1 Probe Layer

Probe attacks are aimed at acquiring information about the tar-get network from a source which is often external to the network. Hence, basic connection level features such as the `duration of connection' and `source bytes' are significant; while features like `number of file creations' and `number of files accessed' are not expected to provide information for detecting Probe attacks. e. g. ipsweep, nmap, portsweep, satan. Features selected for Probe layer are shown in Table 1.

TABLE 1
FEATURE SELECTED FOR PROBE LAYER

Feature Number	Feature Name
1	duration
2	protocol_type
3	pervice
4	flag
5	source_bytes

5.2 DoS Layer

DoS attacks are meant to prevent the target from providing ser-vice(s) to its users by flooding the network with illegitimate requests. Hence, to detect attacks at the DoS layer; network traffic features such as the `percentage of connections having same destination host and same service' and packet level features such as the `source bytes' and `percentage of packets with errors' are significant. To detect DoS attacks, it may not be important to know whether a user is `logged in or not' and hence, such features are not considered in the DoS layer. e. g. back, neptune, pod, smurf, teardrop, land. Features selected for DoS layer are shown in Table 2.

TABLE 2
FEATURE SELECTED FOR DOS LAYER

Feature Number	Feature Name
1	duration
2	protocol_type
4	flag
5	source_bytes
23	count
34	dst_host_same_srv_rate
38	dst_host_serror_rate
39	dst_host_srv_serror_rate
40	dst_host_rerror_rate

5.3 R2L Layer

R2L attacks are one of the mostdifficult attacks to detect as they involve both, the network level and the host level features. Hence, to detect R2L attacks, both the network level features such as the `duration of connection', `service requested' and the host level features such as the `number of failed login attempts are selected among others. e. g. ftp write, guess passwd, imap, phf, multihop, spy, warezclient, warezmaster. Features selected for R2L layer are shown in Table 3.

TABLE 3
FEATURE SELECTED FOR R2L LAYER

Feature Number	Feature Name
1	duration
2	Protocol_type
3	Service
4	Flag
5	source_bytes
10	Hot
11	Num_failed_logins
12	Logged_in
13	Num_compromised
17	Num_files_creations
18	Num_shells
19	Num_access_files
21	Is_host_login
22	Is_guest_login

5.3 U2R Layer

U2R attacks involve the semantic details which are very difficult to capture at an early stage at the network level. Such attacks are often content based and target an application. Hence for detecting U2R attacks, features such as `number of file creations', `number of shell prompts invoked' are selected,

while features such as `protocol' and `source bytes are ignored. e.g. buffer over flow, loadmodule, perl, rootkit. Features selected for U2R layer are shown in Table 4.

TABLE 4
FEATURE SELECTED FOR U2R LAYER

Feature Number	Feature Name
10	Hot
13	Num_compromised
14	Root_shell
16	Num_root
17	Num_files_creations
18	Num_shells
19	Num_access_files
21	Is_host_login

From all the 41 features in the KDD 1999 data set, only five features for Probe layer, nine features for DoS layer, 14 features for R2L layer and eight features for U2R layer are selected manually[1]. Since every layer in the framework is independent, feature sets for all the four layers are not disjoint.

6 SINGLE LAYER INTRUSION DETECTION SYSTEM FRAMEWORK

As explained above, for four attack classes four independent layer IDS models are formed, separately, with feature selection. Single layer IDS Framework for Probe Layer is shown in Fig. 3. Each layer is trained separately to optimally detect a single class of attack. Therefore different features are selected for different layers based upon the type of attack the layer is trained to detect. For example, for probe layer, training of the system is done with Probe attacks and normal audit patterns only from 10% KDD 99 train data set. In this paper, during the training, basically the respective layer attack patterns are formulated using CRF approach and further incorporated in the individual layer IDS. Similarly, the test data is divided into four classes. Experiments are performed separately for all the five attack classes by randomly selecting data corresponding to that particular attack class and normal data only. For example, to detect Probe attacks, testing of the system is done with Probe attacks and normal audit patterns only from 10% KDD 99 test data set.

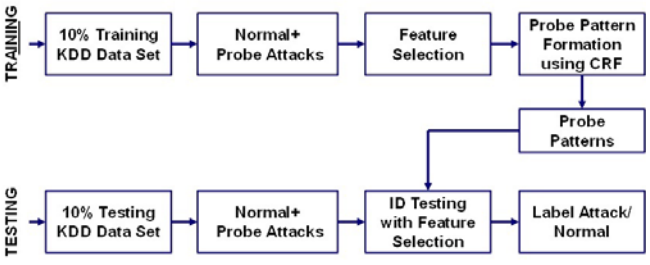

Fig. 3 Single layer IDS Framework (Probe Layer)

7 INTEGRATED MULTILAYER INTRUSION DETECTION SYSTEM FRAMEWORK

Now, layered framework can be integrated with the conditional random fields to build an effective and an efficient network intrusion detection system. Given the four different attack classes in the KDD 1999 data, a four layer system where every layer corresponds to a single attack class is implemented. The four layers are arranged in a sequence as represented in Fig.4.

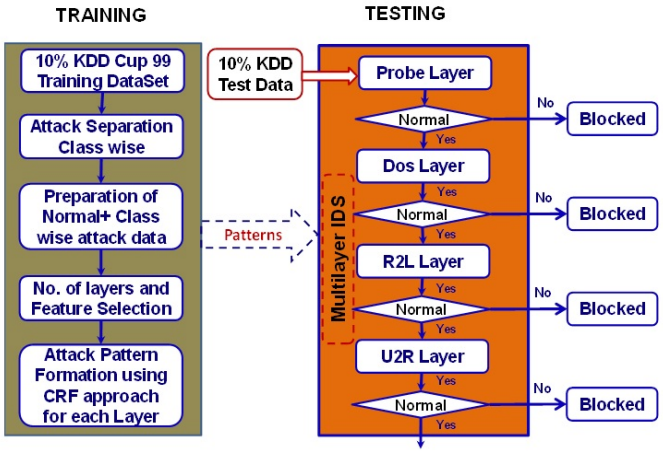

Fig. 4 Integrated Multilayer IDS Framework with CRF Approach

In the system, every layer is trained separately with the normal instances and with the attack instances belonging to a single attack class. The layers are then arranged one after the other in a sequence as shown in Fig.4. However, during testing, all the audit patterns (irrespective of their attack class, which is unknown) are passed into the system starting from the first layer. If the layer detects the instance as an attack, the system labels the instance as a Probe attack and initiates the response mechanism; otherwise it passes the instance to the next layer. Same process is repeated at every layer until either an instance is detected as an attack or it reaches the last layer where the instance is labeled as normal if no attack is detected.

8 ALGORITHM FOR MULTILAYER IDS

The algorithm to integrate the layered intrusion detection framework with conditional random fields approach is given below.

Algorithm 1 : Training

1. Select the number of layers, n, for the complete system.
2. Separately perform features selection for each layer.
3. Form a pattern using conditional random fields for each layer using the features selected from Step 2.
4. Plug in the models sequentially such that only the connections labeled as normal are passed to the next layer.

Algorithm 2 : Testing

1. For each (next) test instance perform Steps 2 through 5.
2. Test the instance and label it either as attack or normal.
3. If the instance is labeled as attack, block it and identify it as an attack represented by the layer name at which it is detected and go to Step 1. Else pass the sequenceto the next layer.

4. If the current layer is not the last layer in the system, test the instance and go to Step 3. Else go to Step 5.
5. Test the instance and label it either as normal or as an attack. If the instance is labeled as an attack, block it and identify it as an attack corresponding to the layer name.

9 Experimental Tests and Results

Intrusion detection tests are conducted on both KDD 99 corrected test dataset. Various layerwise intrusion detection tests such as probe layer test, DoS layer test, R2L test and U2R test and also integrated multilayer test have been carried out on KDD 99 test data.

The results of the above test cases are given in terms of Precision, Recall, and F-Value. However, Precision, Recall, and F-Value are not dependent on the size of the training and the test samples. They are defined as follows:

$$Precision = \frac{Number\ of\ TP}{(Number\ of\ TP + Number\ of\ FP)}$$

$$\text{Recall} = \frac{\text{Number of TP}}{(\text{Number of TP} + \text{Number of FN})}$$

$$\text{F} - \text{Value} = \frac{(1+\beta^2) \times \text{Recall } \times \text{Precision})}{\beta^2 \times (\text{Recall} + \text{Precision})}$$

Where, TP- True Positive, FP- False Positive, FN- False Negative and β corresponds to the relative importance of precision versus recall and is ususally set to 1.

Test details such as unit to test i.e. attack types, Assumption, test data, Steps to Execute, expected results actual achieved results for all test cases are explained in subsequent paragraphs.

9.1 Probe Layer Test

Unit to Test: Probe attacks
Assumption: Probe Layer is trained with feature selected for probe Layer and probe instances and normal instances contained in KDD99 Training set and will detect probe attacks.
Test Data: KDD99 Test data(Corrected) which contains Probe attack audit patterns and normal patterns.
Steps to Execute: Pass the probe audit patterns and normal patterns to probe layer model and compare it with the patterns created by CRF for probe attacks.
Expected Result: All probe attacks should be detected and blocked by Probe Layer.
Actual Result: Probe attacks detected with precision: 91.75%, recall: 98.77% & F-value: 95.13% for KDD99 corrected test dataset.
Pass/Fail: Pass.

9.2 DoS Layer Test

Unit to Test: DoS attacks
Assumption: DoS Layer is trained with feature selected for DoS Layer and DoS in-stances and normal instances contained in KDD99 Training set and will detect DoS attacks.
Test Data: KDD99 Test data (Corrected) which contains DoS attack audit patterns and normal patterns.
Steps to Execute: Pass the DoS audit patterns and normal patterns to dos layer model and compare it with the patterns created by CRF for DoS attacks.
Expected Result: All dos attacks should be detected and blocked by DoS Layer.
Actual Result: DoS attacks detected with precision: 99.75%, recall: 90.92% & F-value: 95.12% for KDD99 corrected test dataset.
Pass/Fail: Pass

9.3 R2L Layer Test

Unit to Test: R2L attacks
Assumption: R2L Layer is trained with feature selected for R2L Layer and R2L in-stances and normal instances contained in KDD99 Training set and will detect R2Lattacks.
Test Data: KDD99 Test data(Corrected) which contains R2L attack audit patterns and normal patterns.
Steps to Execute: Pass the R2L audit patterns and normal patterns to R2L layer model and compare it with the patterns created by CRF for R2L attacks.
Expected Result: All R2L attacks should be detected and blocked by R2L Layer.
Actual Result: R2L attacks detected with precision: 100.00%, recall: 42.28% & F-value: 59.43% for KDD99 corrected test dataset.
Pass/Fail: Pass.

9.4 U2R Layer Test

Unit to Test: U2R attacks
Assumption: U2R Layer is trained with feature selected for U2R Layer and U2R in-stances and normal instances contained in KDD99 Training set and will detect U2R attacks.
Test Data: KDD99 Test data(Corrected) which contains U2R attack audit patterns and normal patterns.
Steps to Execute: Pass the U2R audit patterns and normal patterns to U2R layer model and compare it with the patterns created by CRF for U2R attacks.
Expected Result: All U2R attacks should be detected and blocked by U2R Layer.
Actual Result: U2R attacks detected with precision: 100.00 %, recall: 20.51% & F-value: 34.04% for KDD99 corrected test dataset.
Pass/Fail: Pass.

9.5 Multilayer Test

Unit to Test: All attacks layer wise
Assumption: All Layers are trained with feature selected with respective of attack class a layer is trained to detect. Each layer should detect the attack.
Test Data: KDD99 Test data (corrected) with irrespective of attack class.
Steps to Execute: Pass the all audit patterns starting from first layer model and compare it with the patterns created by CRF for each layer. Detected instances blocked and pass normal instance to next layer.
Expected Result: All Layers detect attack and block attack with respective to attack class it is trained and pass only the normal instances.
Actual Result: Integrated system have precision: 99.71%, recall:

93.51 % & F-value: 96.51 % for KDD99 corrected test dataset.
Pass/Fail: Pass

10 Comparison with Other Approaches

The results of the experiments carried out in the present work using layered CRF approach with feature selection are compared with the results reported with other methods [1] such as Layered Nave Bayes and Layered Decision Trees. The results in terms of Precision, Recall and F-value for each layer (with feature selection) along with other approaches are tabulated in the Tables from 5 to 8 for individual layers.

From the comparision it is found that most of the present methods for intrusion detection fail to reliably detect R2L and U2R attacks. However, high attack detection accuracy can be achieved by using Conditional Random Fields with Multilayer Approach.

TABLE 5

COMPARISON OF RESULTS FOR PROBE LAYER

Results	Precision (%)	Recall (%)	F-Value (%)
Layered CRF	91.75	98.75	95.13
Layered Naives Bayes	73.23	19.57	31.22
Layered Decision Tree	87.04	97.41	91.93

TABLE 6

COMPARISON OF RESULTS FOR DOS LAYER

Results	Precision (%)	Recall (%)	F-Value (%)
Layered CRF	99.75	90.92	95.12
Layered Naives Bayes	99.39	97.00	98.19
Layered Decision Tree	99.90	97.10	98.50

TABLE 7

COMPARISON OF RESULTS FOR R2L LAYER

Results	Precision (%)	Recall (%)	F-Value (%)
Layered CRF	100.00	42.28	59.43
Layered Naives Bayes	81.81	06.47	11.98
Layered Decision Tree	85.48	10.39	18.43

TABLE 8

COMPARISON OF RESULTS FOR U2R LAYER

Results	Precision (%)	Recall (%)	F-Value (%)
Layered CRF	100.00	20.51	34.04
Layered Naives Bayes	35.48	55.12	41.97
Layered Decision Tree	51.00	38.20	43.70

11 Conclusion

Multilayer inintrusion detection systems employing CRF approach developed is presented in this paper. In this paper, an attempt has been made to address the Accuracy and Efficiency for building robust and efficient intrusion detection systems. Experimental results show that CRFs are very effective in improving the attack detection rate and decreasing the FAR. Having a low FAR is very important for any intrusion detection system. Further, feature selection and implementing the Multilayer approach expected to significantly reduce the time required to train and test the model. The test results of the present approach is compared with some well-known methods and found that most of the present methods for intrusion detection fail to reliably detect R2L and U2R attacks. Finally, our system has the advantage that the number of layers can be increased or decreased depending upon the environment in which the system is deployed, giving flexibility to the network administrators.

References

[1] Kapil Kumar Gupta, BaikunthNath, RamamohanaraoKotagiri, "Layered Approach Using Conditional Random Fields for Intrusion Detection", IEEE Transactions on Dependable and Secure Computing (vol. 7 no.1), pp. 35-49, 2010.

[2] W. Lee and S. Stolfo, "Data Mining Approaches for Intrusion Detection," Proc. Seventh USENIX Security Symp. (Security' 98), pp. 79-94, 1998.

[3] H. Shah, J. Undercoffer, and A. Joshi, "Fuzzy Clustering for Intrusion Detection," Proc. 12th IEEE Intl Conf. Fuzzy Systems (FUZZ-IEEE 03), vol. 2, pp. 1274-1278, 2003.

[4] C. Kruegel, D. Mutz, W. Robertson, and F. Valeur, "Bayesian Event Classiffication for Intrusion Detection," Proc. 19th Ann. Computer Security Applications Conf. (ACSAC 03), pp. 14-23, 2003.

[5] N.B. Amor, S. Benferhat, and Z. Elouedi, "Naive Bayes vs. Decision Trees in In-trusion Detection Systems," Proc. ACM Symp. Applied Computing (SAC 04), pp. 420-424, 2004.

[6] H. Debar, M. Becke, and D. Siboni, "A Neural Network Component for an Intrusion Detection System," Proc. IEEE Symp. Research in Security and Privacy (RSP 92), pp. 240-250, 1992.

[7] K.K. Gupta, B. Nath, and R. Kotagiri, "Conditional Random Fields for Intrusion Detection," Proc. 21st Intl Conf. Advanced Information Networking and Applications Workshops (AINAW 07), pp. 203-208, 2007.

[8] J. Lafferty, A. McCallum, and F. Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data," Proc. 18th Intl Conf. Machine Learning (ICML 01), pp. 282-289, 2001.

[9] A. McCallum, D. Freitag, and F. Pereira, "Maximum Entropy Markov Models for Information Extraction and Segmentation," Proc. 17th Intl Conf. Machine Learning (ICML 00), pp. 591-598, 2000.

[10] D.S. Kim and J.S. Park, "Network-Based Intrusion Detection with Support Vector Machines," Proc. Information Networking, Networking Technologies for Enhanced Internet Services Intl Conf. (ICOIN 03), pp. 747-756, 2003.

[11] Y. Gu, A. McCallum, and D. Towsley, "Detecting Anomalies in Network Traffic Using Maximum Entropy Estimation," Proc. Internet Measurement Conf. (IMC05), pp. 345-350, USENIX Assoc., 2005.

[12] *Autonomous Agents for Intrusion Detection*, http://www.cerias.purdue.edu/research/aafid/, 2010.

[13] *Overview of Attack Trends*, http://www.cert.org/archive/pdf/ attack trends.pdf, 2002.

[14] *Probabilistic Agent Based Intrusion Detection*, http://www.cse.sc.edu/research/isl/agentIDS.shtml,2010.

[15] KDD Cup 1999 Intrusion Detection Data, http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, 2010

[16] *SANS Institute Intrusion Detection* FAQ, http://www.sans.org/ resources/idfaq/,2010.