Conditional Random Fields for Intrusion Detection

Kapil Kumar Gupta, Baikunth Nath (Senior Member IEEE), Kotagiri Ramamohanarao
Department of Computer Science and Software Engineering
National ICT Australia
The University of Melbourne, Australia
{kgupta, bnath, rao}@csse.unimelb.edu.au

Abstract

An Intrusion Detection System is now an inevitable part of any computer network. With the ever increasing number and diverse type of attacks, including new and previously unseen attacks, the effectiveness of an Intrusion Detection System is often subjected to testing. The use of such systems have greatly reduced the threat level, however, the networks and hence the data and services offered by them are far away from the state when they can be considered as secure. In this paper we propose and experimentally validate the use and robustness of 'Conditional Random Fields,' for the task of Intrusion Detection. We show, experimentally, that the Conditional Random Fields, can be very effective in detecting intrusions when compared with the previously known techniques.

1. Introduction

Today, in any computer network, the task of Intrusion Detection is of prime significance and a challenge for the network administrators and security personnel. The problem further deteriorates when the attackers come up with more and more and previously unseen attacks even when the current systems are unable to detect all the existing attacks with acceptable reliability [15]. Thus the need to develop more accurate and reliable systems is inevitable.

In this paper we propose and evaluate the use of the Conditional Random Fields [17], a novel technique for the task of Intrusion Detection, and experimentally show that they have higher detection accuracy when compared to any other technique for the same task. We show that the Conditional Random Fields perform better than other methods and offer features which are inherent to the task of Intrusion Detection. We present a better and an effective analysis engine (the core) which can be plugged into any Intrusion Detection System. Our system is a Hybrid system as it builds a model based on both the normal and the anomalous data and hence has the advantages of both the Signature based and the Behaviour based systems. Further, our system can be used as a stand alone system monitoring an entire Network or a single Host or even a single Application running on a particular host.

Currently available Intrusion Detection Systems are either Signature based or Behaviour based. The Signature based systems build a model based on the available knowledge of the attacks and hence extract out signatures which are used to build a classifier to detect same or similar patterns when deployed online. This is also known as Misuse Detection. Complementary to these are the Behaviour based systems which build a model based on the available knowledge of the normal use of the system. Once deployed, they classify any significant deviation from the learnt behaviour as attack. This is also called as Anomaly Detection. The Signature based systems are inept in detecting novel intrusions while the Behaviour based systems suffer from high false alarm rate [7]. The location or deployment of the Intrusion Detection Systems is also significant. Based on their location the Intrusion Detection Systems are either Network based, Host based or Application based, where each of these have their specific advantages and disadvantages. The difference between the three lies in the data that is used as the input to the analysis engine of the Intrusion Detection System. For a Network based Intrusion Detection System, this data is generally the network traffic logs while for the Host based and Application based systems it is the system logs and application logs respectively [7]. The Network based Intrusion Detection Systems work with limited information and suffer when Encryption and Network Address Translation are used, though they are easy to manage and deploy in any sized network. Further, these systems have to deal with the large amount of network traffic and hence they need to be very efficient. The Host based and Application based systems, on the other hand, are difficult to deploy and manage on every node in a large network and are themselves victims of attacks, however, they can work with the information and application level semantics. Other type

21st International Conference on
Advanced Information Networking and Applications Workshops (AINAW'07)
0-7695-2847-3/07 $20.00 © 2007 IEEE

of systems include Distributed Intrusion Detection Systems, Agent based systems and Collaborative Intrusion Detection Systems [8]. As a result of the inherent weaknesses of a single Intrusion Detection System, a number of frameworks have been proposed which describe the collaborative use of Network based and Host based systems [31]. Similarly, systems which employ both Signature based and Behaviour based techniques are discussed in [11]. Another framework in [14] discusses the advantages of Layered Approach for network security.

The rest of the paper is organized as follows; we discuss the related work in Section 2. We then discuss the Conditional Random fields in Section 3 followed by the experimental details and results in Section 4. We finally conclude and provide directions for future work in Section 5.

2. Related Work

The field of Intrusion Detection and Network Security is not new and a number of methods have been proposed and a number of systems have been built to detect intrusions. We now briefly discuss some of the techniques with regards to the task of Intrusion Detection.

Data mining approaches, which includes Association Rules and Frequent Episodes, are based on building classifiers by discovering relevant patterns of program and user behaviour [18], [19]. These approaches can deal with symbolic data and the features can be defined in the form of packet and connection details. However, mining of features is limited to entry level of the packet and also requires the number of attributes to be large and the records to be sparsely populated; otherwise they tend to produce very large number of rules which increases the complexity [5]. Clustering of data has been applied extensively for Intrusion Detection using various clustering methods including k-means, fuzzy c-means and many others [22], [25]. However, one of the main drawbacks of clustering techniques is that it is based on calculating the numeric distance between the observations and hence the attributes of the observations must be numeric. Symbolic attributes can not be easily used which results in inaccuracy. Further, they are unable to capture the relationship between different attributes of a single record. Naive Bayes classifiers are also proposed in [6], however, they make very strict independence assumption between the attributes [27]. In [16] the Bayesian network is discussed for Intrusion Detection. Such systems tend to be attack specific as they build a decision network based on special features of each attack. Thus, the size of the Bayesian network increases rapidly as the number of features and the type of attacks modeled increases. Zhuowei et. al. model the statistical properties in sequences of system calls [33]. However, they only model the sequence of system calls and ignore other details. Hidden Markov Models have also been used for Intrusion Detection. [29], [10], [28] describes their use for modeling the normal sequence of system calls [12] of a privileged process, which can then be used to detect anomalous traces. However, modeling the system calls alone may not always provide accurate classification as various connection level features are ignored. Further, Hidden Markov Models are generative models and fail to model long range dependencies between the observations.

Decision Trees [6] have also been used for Intrusion Detection. The Decision Trees method selects the best attribute for each decision node during the construction of the tree based on some well defined criteria. One such criterion is to use the gain ratio as used in C4.5. Decision Trees can be easily used for building the Misuse Detection Systems, but, it is very difficult to construct Anomaly Detection System. [24], [21], [32] discuss the use of Artificial Neural Networks for Network Intrusion Detection. Though, the Neural Networks can work effectively with noisy data, they require large amount of data during training and it is often hard to select the best possible Neural Network architecture. Support Vector Machines which maps real valued input feature vector to higher dimensional feature space through nonlinear mapping have been used for detecting intrusions in [21]. The Support Vector Machines provide real time detection capability and can deal with large dimensionality of data. However, they are used effectively for binary-class classification only. Along with these, other techniques for detecting intrusion includes the use of Genetic Algorithms, Autonomous Agents for Intrusion Detection [1] and Probabilistic Agent based Intrusion Detection [4].

The most closely related work to our work is of Lee et al. [18], [19] and [20]. They, however, consider a Data Mining approach and resort to mining the Association Rules and finding the Frequent Episodes, thus calculating the Support and Confidence separately. Instead, in our work we define features based on the related attributes alone as well as features based on previous labeling and related attributes in an observation and perform Sequence Labeling via the Conditional Random Fields to label each attribute of the observation before finally labeling each observation instance. This setting is sufficient to model the correlation between different attributes of the observation. The authors in [13] describes the use of Maximum Entropy principle for detecting anomalies in the network traffic. The key difference between their approach and ours is that they use only the normal data during training and build a baseline system i.e. a Behaviour based system while we train our system with both normal and anomalous data and use sequence labeling and classification during testing. Secondly, they fail to model long range dependencies in the observations, which can be easily represented in our model.

21st International Conference on
Advanced Information Networking and Applications Workshops (AINAW'07)
0-7695-2847-3/07 $20.00 © 2007 IEEE

3. Conditional Random Fields

Conditional Random Fields are undirected graphical models used for the task of sequence tagging. The difference between the Conditional Random Fields (CRF) and other graphical models such as the Hidden Markov Models (HMM) is that the HMM, being generative, models the joint distribution $p(y, x)$ whereas the CRF are discriminative models directly modeling the conditional distribution $p(y|x)$ which is the distribution of interest. Similar to HMM, the Naive Bayes are also generative and model the joint distribution. Modeling the joint distribution for the task of classification and sequence labeling has two disadvantages. First, it is not the distribution that is of interest, since the observations are completely visible and the interest is in finding the correct class for the visible observation which is the same as the conditional distribution $p(y|x)$. Secondly, inferring the conditional probability $p(y|x)$ from the modeled joint distribution, using the Bayes rule, requires the marginal distribution $p(x)$. To calculate this marginal distribution is difficult as the amount of training data is often limited and the observation x contains highly dependent features which are difficult to model and hence strong independence assumptions are made to simplify the task. This results in reduced accuracy [26]. Thus, the Conditional Random Fields simply try to predict y given the $x's$. This allows them to model arbitrary features in different attributes in x [9]. Conditional Random Fields also avoid the observation bias and the label bias problem which is present in other discriminative models, such as the Maximum Entropy Markov Models. The Maximum Entropy Markov Models use per-state exponential model for the conditional probabilities of the next state given the current state and the observation sequence [23] while the Conditional Random Fields have a single exponential model for the joint probability of the entire sequence of labels given the observation sequence, thus avoiding the label bias problem [17].

Given X and Y, the random variables over data sequence to be labeled and corresponding label sequences, let $G = (V, E)$ be a graph such that $Y = (Y_v)_{v \in V}$ is represented by the vertices of the graph G, then, (X, Y) is a Conditional Random Field, when conditioned on X, the random variables Y_v obey the Markov property with respect to the graph: $p(Y_v|X, Y_w, w \neq v) = p(Y_v|X, Y_w, w \sim v)$, where $w \sim v$ means that w and v are neighbors in G, [17], i.e. a CRF is a random field globally conditioned on X. For a simple sequence modeling, as in our case, the joint distribution over the label sequence Y given X has the form:

$$p_\theta(y|x) \propto \exp\Big(\sum_{e \in E,k} \lambda_k f_k(e, y|_e, x) + \sum_{v \in V,k} \mu_k g_k(v, y|_v, x)\Big) \tag{1}$$

where x is the data sequence, y is a label sequence, and $y|_s$ is the set of components of y associated with the vertices in subgraph S. Also, the features f_k and g_k are assumed to be given and fixed [17]. The parameter estimation problem is to find the parameters $\theta = (\lambda_1, \lambda_2, ...; \mu_1, \mu_2, ...)$ from the training data $D = (x^i, y^i)_{i=1}^N$ with the empirical distribution $\tilde{p}(x, y)$ [17]. The graphical structure of a Conditional Random Fields can be represented as shown in Figure 1

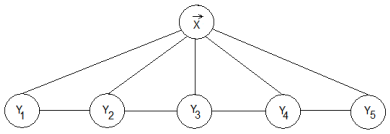

Figure 1. Graphical Representation of a Conditional Random Field

where $\overrightarrow{X}$ represents a sequence of length five, in this case, and each attribute of $\overrightarrow{X}$ is correspondingly labeled as Y_i.

The task of Intrusion Detection can now be compared to many problems in Machine Learning, Natural Language Processing and Bio-Informatics. The Conditional Random Fields have been proven to be very successful in such tasks, as they make no unwarranted assumptions about the data, and once trained they also appear to be very efficient and robust. The task of Intrusion Detection, however, has some major requirements. It has to be an online task and there is no knowledge available for the future observations. Further, once deployed, the system has to deal with large amount of data and thus it must be able to perform fast enough to be effective. Conditional Random Fields satisfy all of these requirements and once the model has been trained and deployed, they are very fast in labeling the data as either normal or as attack. The complexity of a Conditional Random Field is quadratic with respect to the number of labels. This is problematic when the number of labels is large, such as in the language tasks, but in our case we have only two labels; normal and attack. We observe, that the training of a Conditional Random Field is expensive but once trained their performance is comparable to that of the Decision Trees and Naive Bayes classifiers. Thus our system is very efficient and can be used online. As discussed in Section 2, the current work does not consider the relationships among the attributes in the observations. They either consider only one attribute, such as in the system call modeling, or assume conditional independence among the attributes. However, if we can model these relations suitably, the system is bound to perform better. As we will show from our experimental results, the Conditional Random Fields can be effectively used to model such complex relationships among the attributes of an observation without compromising the accuracy and efficiency of classification. We first perform the sequence labeling using a Conditional Random Field where the system considers every record as a separate sequence of attributes and labels each attribute of this sequence to

give the final label for each record. We then perform experiments with the Decision Trees and Naive Bayes classifier and compare the results with the Conditional Random Fields.

4. Experiments

In this section we describe the data-set used, the details of our experimental procedure, the results and comparison of our results. We give the Precision, Recall and F-Value together with the classification accuracy for all our experiments. Accuracy is simply the ratio of the total number of correct classifications to the total number of records and thus depends on the sample size whereas Precision, Recall and F-measure are not dependent on the size of training and testing sample and are defined as follows:

$$Precision = \frac{TP}{TP+FP}$$

$$Recall = \frac{TP}{TP+FN}$$

$$FMeasure = \frac{(1+\beta^2)*Recall*Precision}{\beta^2*(Recall+Precision)}$$

where TP, FP and FN are the number of True Positives, False Positives and False Negatives respectively and β corresponds to relative importance of precision vs. recall and is usually set to a value of 1.

4.1. Description of Data

We use the benchmark KDD cup 1999 Intrusion Detection data-set for our experiments [3]. The KDD cup 1999 data contains about five million connection records as the training data and about two million connection records as the test data. In our experiments we have used the ten percent of the total training data which is provided separately. This leads to four hundred ninety four thousand and twenty connection records in total. 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 each record is represented by forty one different attributes. Each record is considered as independent of any other record. The training data is either labeled as normal or as one of the twenty four different kinds of attacks. These twenty four attacks can be grouped into four classes: Denial of Service (DoS), Probing, R2L (unauthorized access from a remote machine) or U2R (unauthorized access to root).

4.2. Methodology and Experiments Performed

In our experiments with Conditional Random Fields we use the Conditional Random Field toolkit, CRF++ [2]. Further, we used Weka tool [30] to perform the experiments with Decision Trees and Naive Bayes classifier. First we relabel and separate the entire data-set into five classes; normal, DoS, Probe, R2L and U2R depending upon the initial label of each record. We can do this because each record in the data is independent of any other record. We then randomly divide each set except DoS into two approximately equal parts i.e. from normal set we create n-training and n-testing, from Probe set we create p-training and p-testing, from R2L set we create r-training and r-testing and from U2R set we create u-training and u-testing. Since the number of records in the DoS group is very large, we divide the DoS set into five sets randomly. We use only two of these and ignore the remaining three and call the two sets as d-training and d-testing. Additionally, we create two random sets of about 300 records each from the normal set and call them as n-small-training and n-small-testing. We use the same set of data for all our experiments.

4.3. Experimental Results

The objective of our experiments is to test how accurately the different models can identify individual class of attack when it is mixed with the normal class only i.e. the attacks here belong to only one attack group.

First we mix n-training with d-training and n-testing with d-testing to get the training and testing data to detect DoS attacks from the normal data. Table 1 shows the results for this experiment.

Table 1. Normal - DoS

	Accuracy (%)	Precision (%)	Recall (%)	F-Value (%)
Conditional Random Fields	99.99	99.99	99.99	99.99
Decision Tree	99.99	99.99	99.99	99.99
Naive Bayes	98.97	99.05	99.28	99.16

To detect Probe attacks, we mix n-training with p-training and n-testing with p-testing to get the training and testing data. Table 2 shows the results for this experiment.

Table 2. Normal - Probe

	Accuracy (%)	Precision (%)	Recall (%)	F-Value (%)
Conditional Random Fields	99.94	99.50	99.06	99.28
Decision Tree	99.96	99.75	99.21	99.48
Naive Bayes	91.95	32.37	93.37	48.07

To detect R2L attacks, we mix n-training with r-training and n-testing with r-testing to get the training and testing data. Table 3 shows the results for this experiment.

Table 3. Normal - R2L

	Accuracy (%)	Precision (%)	Recall (%)	F-Value (%)
Conditional Random Fields	99.96	99.62	96.37	97.97
Decision Tree	99.91	98.11	94.01	96.01
Naive Bayes	98.68	44.63	74.59	55.84

Finally, we mix n-small-training with u-training and n-small-testing with u-testing to get the training and testing data to detect U2R attacks from the normal data. It should be noted that, for the U2R group, we have used the reduced set of normal data as the number of records in the U2R set is very small and hence, the normal records tend to dominate in the model and bias the model parameters towards normal, thus decreasing the classification capability of the classifier. Further, we found that reducing the sample size for the normal data did not affect the classification accuracy of the remaining normal data and the results were similar to those with the reduced sample size. Table 4 shows the results for this experiment.

Table 4. Normal - U2R

	Accuracy (%)	Precision (%)	Recall (%)	F-Value (%)
Conditional Random Fields	99.08	100.00	86.36	92.68
Decision Tree	96.92	75.00	81.82	78.26
Naive Bayes	94.46	55.88	86.36	67.86

It is evident from our experiments that the Conditional Random Fields are far better than the Naive Bayes for the detection of all the four attack groups. When compared with the Decision Trees we find that the Conditional Random Fields have much higher F-value for the R2L and U2R type of attacks. The domain knowledge suggests that in order to detect the R2L and U2R attacks, relationship among different attributes is of prime importance and this can be modeled using the Conditional Random Fields and hence they give better results. The Decision Trees, however, are equally good for detecting the DoS attacks and are slightly better in case of probes.

4.4. Discussion

The KDD cup 1999 Intrusion Detection data is one of the widely used data set for the task of Intrusion Detection. It is worth mentioning that the data-set is dominated by the denial of service attacks. The proportion of the R2L and the U2R groups of attacks is very small. However, it still remains the preferred data-set. Further, detecting the Denial of Service attacks and the Probing attacks in this data-set is trivial and high accuracy is achieved by most of the current known techniques. Detecting the R2L and the U2R attacks is challenging. The prime reason for this is that they do not follow any specific pattern which can be modeled easily. It is in the last two groups wherein most of the Intrusion Detection techniques fail to provide adequate results. Our experimental results show that the Conditional Random Fields perform significantly better than the other compared techniques in these two attack groups as well. The better accuracy of Conditional Random Fields is attributed to the fact that they label every attribute, of each instance, from the best possible labels in such a way that the overall probability of all the labels in the entire sequence is maximum. As a result they find the best possible tag for the entire instance and the results show that they are better suited for our task. This also justifies our motive of considering a partially relational data, the KDD cup 1999 data, as a sequential data with respect to the different attributes of each record. We call the KDD cup 1999 data as a partially relation data set because all the forty one attributes in each instance of the data-set can be grouped into three sets, basic features of individual TCP connection, content features suggested from the domain knowledge and the traffic features, which form a logical sequence.

5. Conclusion and Future Work

In this paper, we have shown that the Conditional Random Fields can be effectively used for the task of Intrusion Detection and the experimental results shows that they outperform the existing techniques. We also discussed how the sequence labeling tools such as the Conditional Random Fields can be easily and effectively used for labeling relational data. As part of our future work, we plan to test the performance of the Conditional Random Fields on new unseen attacks by integrating our Layered Approach [14] with this work.

References

[1] Autonomous agents for intrusion detection. Last assessed: July 12, 2006. http://www.cerias.purdue.edu/research/aafid/.

[2] Crf++: Yet another crf toolkit. Last Accessed: August 31, 2006. http://chasen.org/˜taku/software/CRF++/.

[3] Kdd cup 1999 intrusion detection data. Last assessed: July 02, 2006. http://kdd.ics.uci.edu//databases/kddcup99.

[4] Probabilistic agent based approach for intrusion detection. Last assessed: July 06, 2006. http://www.cse.sc.edu/research/isl/agentIDS.shtml.

[5] T. Abraham. Iddm: Intrusion detection using data mining techniques. Last assessed: July 06, 2006. http://www.dsto.defence.gov.au/publications/2345/DSTO-GD-0286.pdf.

[6] N. B. Amor, S. Benferhat, and Z. Elouedi. Naive bayes vs decision trees in intrusion detection systems. In *Proceedings of the ACM symposium on Applied computing*, pages 420–424. ACM Press, 2004.

[7] R. Bace and P. Mell. *Intrusion Detection Systems*. Gaithersburg, MD : Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology, 2001.

[8] M. Bishop. *Computer Security Art and Science*. Addison-Wesley, 2003.

[9] T. G. Dietterich. Machine learning for sequential data: A review. In *Proceedings of the Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition*, pages 15–30. Lecture Notes in Computer Science, Springer-Verlag, No. (2396), 2002.

[10] Y. Du, H. Wang, and Y. Pang. A hidden markov models-based anomaly intrusion detection method. In *Fifth World Congress on Intelligent Control and Automation (WCICA)*, pages 4348–4351. IEEE Press, vol(5), 2004.

[11] L. Ertoz, A. Lazarevic, E. Eilertson, P.-N. Tan, P. Dokas, V. Kumar, and J. Srivastava. Protecting against cyber threats in networked information systems. In *Proceedings of SPIE; Battlespace Digitization and Network Centric Systems III*, pages 51–56, 2003.

[12] S. Forrest, S. A. Hofmeyr, A. Somayaji, and T. A. Longstaff. A sense of self for Unix processes. In *Proceedinges of the IEEE Symposium on Research in Security and Privacy*, pages 120–128. IEEE Computer Society Press, 1996.

[13] Y. Gu, A. McCallum, and D. Towsley. Detecting anomalies in network traffic using maximum entropy estimation. In *Proceedings of the Internet Measurement Conference 2005*, page 345350, 2005.

[14] K. K. Gupta, B. Nath, and K. Ramamohanarao. Network security framework. *International Journal of Computer Science and Network Security*, 6(7B):151–157, 2006.

[15] K. K. Gupta, B. Nath, K. Ramamohanarao, and A. Kazi. Attacking confidentiality: An agent based approach. In *Proceedings of IEEE International Conference on Intelligence and Security Informatics*, pages 285–296. Lecture Notes in Computer Science, Springer Verlag, vol(3975), 2006.

[16] C. Kruegel, D. Mutz, W. Robertson, and F. Valeur. Bayesian event classification for intrusion detection. In *19th Annual Computer Security Applications Conference*, pages 14–23. IEEE Computer Society, 2003.

[17] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of Eighteenth International Conference on Machine Learning, ICML*, pages 282–289, 2001.

[18] W. Lee and S. Stolfo. Data mining approaches for intrusion detection. In *Proceedings of the 7th USENIX Security Symposium*, pages 79–94, 1998.

[19] W. Lee, S. Stolfo, and K. Mok. Mining audit data to build intrusion detection models. In *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining*, pages 66–72. AAAI Press, 1998.

[20] W. Lee, S. J. Stolfo, and K. W. Mok. A data mining framework for building intrusion detection model. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 120–132. IEEE Press, 1999.

[21] S. Mukkamala, G. Janoski, and A. Sung. Intrusion detection using neural networks and support vector machines. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, pages 1702–1707. IEEE Press, 2002.

[22] L. Portnoy, E. Eskin, and S. Stolfo. Intrusion detection with unlabeled data using clustering. In *ACM Workshop on Data Mining Applied to Security (DMSA)*. ACM Press, 2001. Last Accessed: Septemebr 18, 2006. http://www1.cs.columbia.edu/ids/publications/cluster-ccsdmsa01.pdf.

[23] A. Ratnaparkhi. A maximum entropy model for part-of-speech tagging. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 133–142. Association for Computational Linguistics, 1996.

[24] J. Ryan, M. J. Lin, and R. Miikkulainen. Intrusion detection with neural networks. In *Advances in Neural Information Processing Systems*. MIT Press, 1998. Last Accessed: Septemebr 18, 2006. http://nn.cs.utexas.edu/downloads/papers/ryan.intrusion.pdf.

[25] H. Shah, J. Undercoffer, and A. Joshi. Fuzzy clustering for intrusion detection. In *The 12th IEEE International Conference on Fuzzy Systems*, pages 1274–1278. IEEE Press, vol(2), 2003.

[26] C. Sutton and A. McCallum. *Introduction to Statistical Relational Learning: An Introduction to Conditional Random Fields for Relational Learning*. MIT Press, 2006. Last Accessed: Septemebr 18, 2006. http://www.cs.umass.edu/˜mccallum/papers/crf-tutorial.pdf.

[27] A. Valdes and K. Skinner. Adaptive, model-based monitoring for cyber attack detection. In *Recent Advances in Intrusion Detection (RAID)*, pages 80–92. Lecture Notes in Computer Science, Springer-Verlag, number (1907), 2000.

[28] W. Wang, X. H. Guan, and X. L. Zhang. Modeling program behaviors by hidden markov models for intrusion detection. In *Proceedings of International Conference on Machine Learning and Cybernetics*, pages 2830–2835. IEEE Press, vol(5), 2004.

[29] C. Warrender, S. Forrest, and B. Pearlmutter. Detecting intrusions using system calls: alternative data models. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 133–145. IEEE Press, 1999.

[30] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.

[31] Y.-S. Wu, B. Foo, Y. Mei, and S. Bagchi. Collaborative intrusion detection system (cids): A framework for accurate and efficient ids. In *Proceedings of the 19th Annual Computer Security Applications Conference (ACSAC)*, pages 234–244, 2003.

[32] Z. Zhang, J. Li, C. Manikopoulos, J. Jorgenson, and J. Ucles. Hide: a hierarchical network intrusion detection system using statistical preprocessing and neural network classification. In *Proceedings of the IEEE Workshop on Information Assurance and Security United States Military Academy*, pages 85–90, 2001.

[33] L. Zhuowei, A. Das, and S. Nandi. Utilizing statistical characteristics of n-grams for intrusion detection. In *Proceedings. 2003 International Conference on Cyberworlds*, pages 486–493. IEEE Press, 2003.