ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol.2 , No.1, Pages : 01-06 (2013)
Special Issue of ICACSE 2013 - Held on 7-8 January, 2013 in Lords Institute of Engineering and Technology, Hyderabad

Intrusion Detection Using Pipelining of Layers with Conditional Random Fields in Multicore Processors

Veerraju Gampala[1], Srilakshmi Inuganti[2], Satish Muppidi[3]
1Assistant Professor, Department of Information Technology, GMRIT, Rajam, Andhra Pradesh, India, veerraju.g@gmrit.org
2Assistant Professor, Department of Information Technology, GMRIT, Rajam, Andhra Pradesh, India, srilakshmi.i@gmrit.org
3Assistant Professor, Department of Information Technology, GMRIT, Rajam, Andhra Pradesh, India, satish.m@gmrit.org

Abstract: An intrusion detection system is a predictable element of any computer network system. Now a day's intrusion detection faces a number of key challenging issues. The challenges like detect malicious activities and the large amount of network traffic. In this paper these two issues are addressed using conditional random fields (CRFs) and pipelining of layers in multicore processors to get accuracy, efficiency and high performance. We demonstrate that high attack detection using CRFs, high efficiency using layered concept and high performance using pipelining of layers approach. Our proposed system performs well when compare to other intrusion detection systems like naïve bayes and decision trees. Our proposed system will detect attacks like U2R attack, R2L attack, Probe attack and DoS attack with very high accuracy. Our method is practically implemented and observed the results of four type attacks.

Keywords: Intrusion detection, Pipelining of layers, Multicore processors, Conditional random fields, Naïve bayes, and Decision trees.

INTRODUCTION

Intrusion detection started in around 1980s after the influential paper from Anderson. Intrusion detection is the art and science of sensing when a system or network is being used inappropriately or without authorization. An intrusion-detection system (IDS) monitors system and network resources and activities and, using information gathered from these sources, notifies the authorities when it identifies a possible intrusion.

For the purpose of dealing with IT, there are three main types of IDS:

Network intrusion detection system (NIDS)

Is an independent platform that identifies intrusions by examining network traffic and monitors multiple hosts, developed in 1986 by Pete R. Network intrusion detection systems gain access to network traffic by connecting to a network hub, network switch configured for port mirroring, or network tap. In a NIDS, sensors are located at choke points in the network to be monitored, often in the demilitarized zone (DMZ) or at network borders. Sensors capture all network traffic and analyze the content of individual packets for malicious traffic. An example of a NIDS is Snort.

Host-based intrusion detection system (HIDS)

It consists of an agent on a host that identifies intrusions by analyzing system calls, application logs, file-system modifications (binaries, password files, capability databases, Access control lists, etc.) and other host activities and state. In a HIDS, sensors usually consist of a software agent. Some application-based IDS are also part of this category. Examples of HIDS are Tripwire and OSSEC.

Stack-based intrusion detection system (SIDS)

This type of system consists of an evolution to the HIDS systems. The packets are examined as they go through the TCP/IP stack and, therefore, it is not necessary for them to work with the network interface in promiscuous mode. This fact makes its implementation to be dependent on the Operating System that is being used.

Intrusion detection system can also be classified as signature based or anomaly based depending upon the attack detection method. The signature-based systems are trained by extracting specific patterns (or signatures) from previously known attacks while the anomaly-based systems learn from the normal data collected when there is no anomalous activity.

Another approach for detecting intrusions is to consider both the normal and the known anomalous patterns for training a system and then performing classification on the test data. Such a system includes the advantages of both the signature-based and the anomaly-based systems and is known as the Hybrid System. Hybrid systems can be very efficient, subject to the classification method used, and can also be used to label unseen or new instances as they assign one of the known classes to every test instance. This is possible because during training the system learns features from all the classes.

ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol.2 , No.1, Pages : 01-06 (2013)
Special Issue of ICACSE 2013 - Held on 7-8 January, 2013 in Lords Institute of Engineering and Technology, Hyderabad

In this paper we propose and evaluate the use of the conditional Random fields and Pipelining of Layer concepts in multicore processors for the task of intrusion detection. We show that the conditional Random Fields and pipelining of layers perform better than other methods and offer features which are inherent to the task of Intrusion Detection.

RELATED WORK

The area of Intrusion Detection system and Network Security is not new and a number of techniques 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 system. a variety of techniques such as association rules, clustering, naive Bayes classifier, support vector machines, genetic algorithms, and others have been applied to detect intrusions. In this section, we briefly discuss these techniques and frameworks.

A. **Data clustering methods**: Data clustering methods such as the k-means and the fuzzy c-means have also been applied extensively for intrusion detection system. The main drawback of the clustering technique is that it is based on calculating numeric distance between the observations, and hence, the observations must be numeric.

B. **Data mining approach**: Data mining approaches for intrusion detection consist of association rules and frequent episodes, which are based on structure classifiers by discovering relevant patterns of program and user behavior. Association rules and frequent episodes are used to learn the record patterns that express user behavior. These methods can deal with symbolic data, and the features can be defined in the form of packet and connection details.

C. **Pattern recognition method:** Naive Bayes classifiers have also been used for intrusion detection. However, they make strict independence assumption between the features in an observation resulting in lower attack detection accuracy when the features are correlated, which is often the case for intrusion detection. Bayesian network can also be used for intrusion detection. However, they tend to be attack specific and build a decision network based on special characteristics of individual attacks. Thus, the size of a Bayesian network increases rapidly as the number of features and the type of attacks modeled by a Bayesian network increases.

D. **Decision trees:** Decision trees have also been used for intrusion detection. The decision trees select the best features for each decision node during the construction of the tree based on some well-defined criteria. One such criterion is to use the information gain ratio, which is used in C4.5. Decision trees generally have very high speed of operation and high attack detection accuracy.

E. **Genetic algorithm:** Other approaches for detecting intrusion include the use of genetic algorithm and autonomous and probabilistic agents for intrusion detection. These methods are generally aimed at developing a distributed intrusion detection system.

CONDITIONAL RANDOM FIELDS

Conditional random fields (CRFs) are a class of statistical modeling method often applied in pattern recognition and machine learning, where they are used for structured prediction. Whereas an ordinary classifier predicts a label for a single sample without regard to "neighboring" samples, a CRF can take context into account; e.g., the linear chain CRF popular in natural language processing predicts sequences of labels for sequences of input samples.

CRFs are a type of discriminative undirected probabilistic graphical model. It is used to encode known relationships between observations and construct consistent interpretations. It is often used for labeling or parsing of sequential data, such as natural language text or biological sequences and in computer vision. Specifically, CRFs find applications in shallow parsing, named entity recognition and gene finding, among other tasks, being an alternative to the related hidden Markov models. In computer vision, CRFs are often used for object recognition and image segmentation.

Lafferty, McCallum and Pereira define a CRF on observations X and random variables Y as follows: Let G= (V, E) be a graph such that $Y= (Y_v)_{v\in V,}$ so that Y is indexed by the vertices of G. Then (X, Y) is a conditional random field in case, when conditioned on X, the random variables Y_v obey the Markov property with respect to the graph: $Y= (Y_v)_{v\in V}P(Y_v|X, Y_{W,} w\neq v) = P(Y_v|X, Y_{W,} w \approx v)$where $w \approx v$ means that w and v are neighbors in G. i.e a CRF is a random field globally conditioned on X. What this means is that a CRF is a partially directed graphical model whose nodes can be divided into exactly two disjoint sets X and Y, the observed and output variables, respectively; the conditional distribution P(Y |X) is then modeled. For a simple sequence modeling, in our proposed system, the joint distribution over label sequence Y given X has the following form:

ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol.2 , No.1, Pages : 01-06 (2013)
Special Issue of ICACSE 2013 - Held on 7-8 January, 2013 in Lords Institute of Engineering and Technology, Hyderabad

$P\theta(y|x) \alpha \exp (\sum_{e\in E,k} \lambda_k f_k(e,y|_e,x)+ \sum_{v\in V,k} \mu_k g_k(v,y|_v,x))$

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 sub graph S. Also, the features f_k and g_k are assumed to be given and fixed. The parameter estimation problem is to find the parameters $\theta = (\lambda_1, \lambda_2,...; \mu_1, \mu_2, ...)$ from the training data $D = (x^i, y^i)^N_{i=1}$ with the empirical distribution $p\square(x, y)$. The graphical structure of a Conditional Random Fields can be represented as shown in Fig 1.

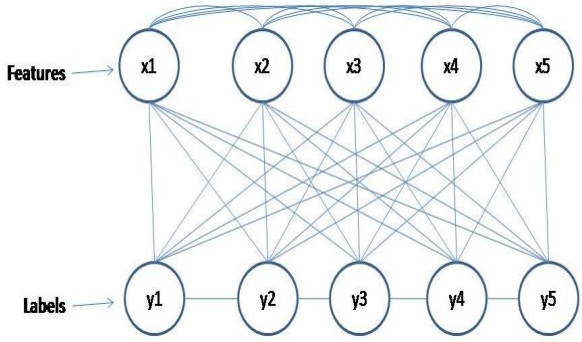

Fig 1:. Graphical Representation of a Conditional Random Field

where x1,x2,x3,x4,x5 represents observations of a sequence of length five, in this case, and each feature of observation(xi) is correspondingly labeled as yi. For example a feature fk might be true if observation x2 is "service=ftp", label y1 is "normal" and label y2 is "normal". Similarly a feature gk be true if the observation x2 is "protocol=tcp" and label y2 is "attack". We aim to model the relationships among features of individual connections using a CRF, as shown in Fig 1. In the figure, features such as x1, x2, x3, x4, and x5 take some possible value for every connection. During training, feature weights are learnt, and during testing, features are evaluated for the given observation, which is then labeled accordingly.

Every label is connected to every input feature, which indicates that all the features in an observation help in labeling, and thus, a CRF can model dependencies among the features in an observation. Present intrusion detection systems do not consider such relationships among the features in the observations. They either consider only one feature, such as in the case of system call modeling, or assume conditional independence among different features in the observation as in the case of a naive Bayes classifier. CRFs can effectively model such relationships among different features of an observation resulting in higher attack detection accuracy. Another advantage of using CRFs is that every element in the sequence is labeled such that the probability of the entire labeling is maximized, i.e., all the features in the observation collectively determine the final labels. Hence, even if some data is missing, the observation sequence can still be labeled with less number of features. The task of intrusion detection can be compared to many problems in machine learning, natural language processing, and bioinformatics. The CRFs have proven to be very successful in such tasks, as they do not make any unwarranted assumptions about the data. Hence, we explore the suitability of CRFs for intrusion detection.

PIPELINING OF LAYERS FOR INTRUSION DETECTION

Layered security model

Layered security model is a sequential model in which number of security checks are performed one after other in sequence. Sequential Layered Approach and is based on ensuring availability, confidentiality, and integrity of data and services over a network. The objective of using a layered security 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 independent to block an attack without the need of a central decision-maker. In model every layer is trained separately and then deployed successively. We define four layers that correspond to the four attack groups. They are Probe layer, DoS layer, R2L layer, and U2R layer. Every layer is then separately trained with a small set of relevant features. Feature selection is significant for Layered model. Layers act as filter that identifies anomalies connection and blocks it without further processing.

To improve the speed of our intrusion detection system, we implemented layered model assigned a small set of features to each layer rather than assigning all features to one layer. Performance is improved during training and testing of our system. We implement the Layered model to improve overall system performance. The performance of our proposed system, Layered CRFs, is comparable to that of the decision trees and the naive Bayes, and our system has higher attack detection accuracy. Fig 2 represents this layered security model.

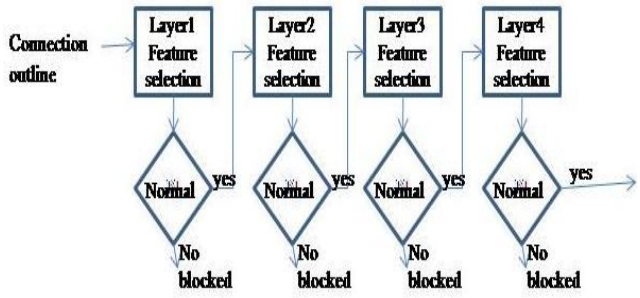

Fig 2: Layered security model.

Pipelining

Pipelining is used by virtually all modern multicore processors to enhance performance by overlapping the

ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol.2 , No.1, Pages : 01-06 (2013)
Special Issue of ICACSE 2013 - Held on 7-8 January, 2013 in Lords Institute of Engineering and Technology, Hyderabad

execution of instructions. Pipeline: is an implementation technique where multiple instructions are overlapped in execution. Pipeline stage: The computer pipeline is to divide instruction processing into stages. Each stage completes a part of an instruction and loads a new part in parallel. The pipeline is divided into segments and each segment can execute it operation concurrently with the other segments. Once a segment completes an operation, it passes the result to the next segment in the pipeline and fetches the next operations from the preceding segment. Pipelining does not decrease the time for individual instruction execution. Instead, it increases instruction throughput. The throughput of the instruction pipeline is determined by how often an instruction exits the pipeline.

Proposed Pipelined Layered Security Model (PLSM)

The layered security model improves the intrusion detection system performance by finding attacks in four layers. If the attack is found in initial layer then it is blocked otherwise test instance is passed to next layer. If the test instance is passed through all the four layers then it indicates there is no attack. In this technique the test instance is passed sequentially through all the layers. Only one test instance is checked at a time. Next instance has to wait until the previous instance completes its checking through all layers. Even it is finding attacks with good performance it causes delay for checking number of instances at a time even some of the layers are free state to check the instance. To overcome this drawback and to get high performance of our intrusion detection system, we proposed to apply pipelining concept to layered security model in multicore processors, it is termed pipelined layered security model. Fig 3. represents our proposed model.

In this PLSM, testinstance1 is passed to layer1; it will test for attacks in cycle1 if attacks not found then it will send to layer2 in cycle2. At this time (cycle2) layer1 is in Free State, so testinstance2 is passed to layer1 for testing. In cycle2 layer1 will test testinstance2 and layer2 will test testinstance1 for attacks, if attacks not found then these instances are passed to next layer for testing. In cycle3 layer1 is Free State, so testinstanc3 is passed to layer1 for testing. In the next cycle testinstance4 is passed to layer1 which is Free State. In cycle4 testinstance1 is in layer4, testinstance2 is in layer3, testinstance3 is in layer2 and testinstance4 is in layer1 for testing. After completion of cycle4, testinstance1 is tested in all four layers it will decide whether testinstance1 is blocked or not depending upon the attacks. The results of testinstance2, testinstance3 and testinstance4 will get after completion of cycle5, cycle6 and cycle7 respectively. By using this proposed model it will improves throughput and latency. Time required for a testinstance propagating through the pipeline is nothing but latency. The numbers of testinstances that are completed testing for attacks per unit of time is nothing but throughput.

Throughput = number of testinstances completed / total time clock cycles

For example if we assume total time cycles 7 then

Throughput of PLSM = 4/7

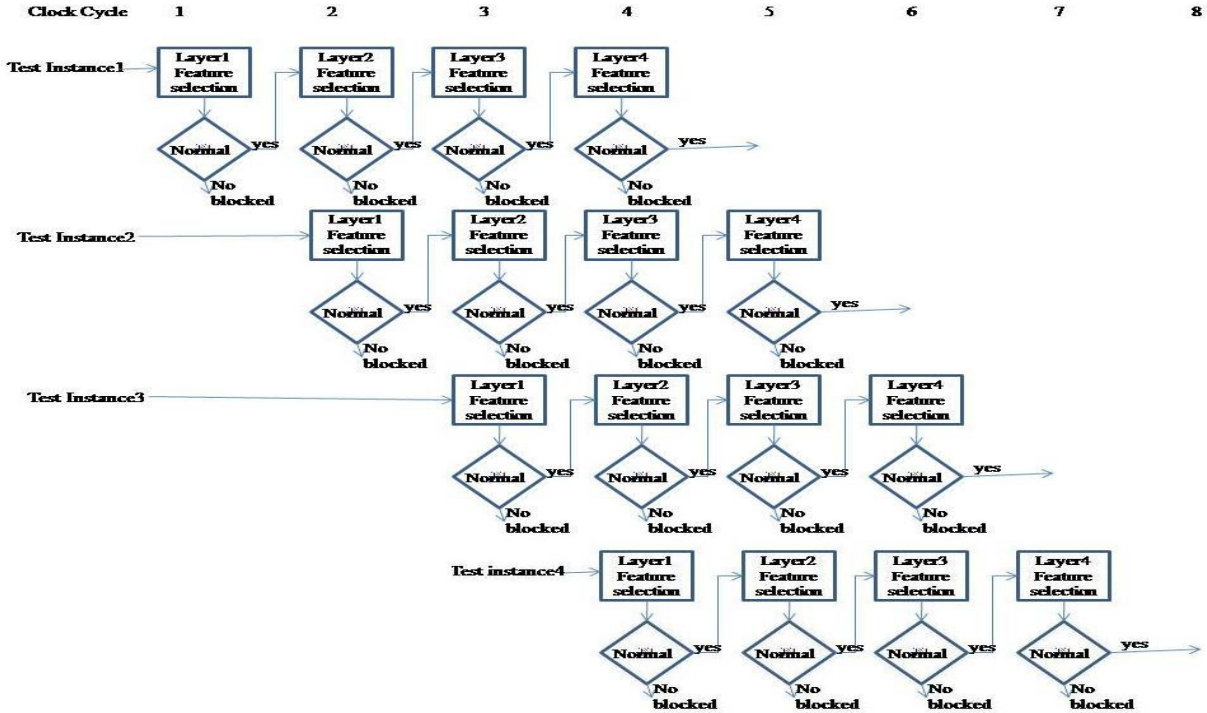

Fig 3:.Representation of pipelined layered security model (PLSM)

Without using PLSM to complete testing for 4 testinstances it will take 16 time clock cycles but with PLSM intrusion detection system it will take only 7 time clock cycles. That's why our proposed model exhibits high performance in multicore processors.

ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol.2 , No.1, Pages : 01-06 (2013)
Special Issue of ICACSE 2013 - Held on 7-8 January, 2013 in Lords Institute of Engineering and Technology, Hyderabad

FEATURE SELECTIONS FOR EACH LAYER

We first select four layers corresponding to the four attack groups those are Probe, DoS, R2L, and U2R; and perform feature selection for each layer. We illustrate our approach for selecting features for every layer and why some features were chosen over others. In our system, every layer is separately trained to detect a single type of attack group. We observe that the attack groups are different in their impact, and hence, it becomes necessary to treat them differently. Hence, we select features for each layer based upon the type of attacks that the layer is trained to detect.

Denial of Service Layer

Denial of Service attack (DoS) is an attack in which the attacker makes some computing or memory resource too busy or too full to handle legitimate requests, or denies legitimate users access to a machine. Therefore, for the DoS layer, 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" is significant. To detect DoS attacks, it may not be important to know whether a user is "logged in or not."

Probe Layer

Probing attack is an attempt to gather information about a network of computers for the apparent purpose of circumventing its security controls. Therefore, essential connection level features such as the "duration of connection" and "source bytes" are significant while features like "number of files creations" and "number of files accessed" are not expected to provide information for detecting probes.

User to Remote Layer

User to Remote attack (U2R) is a class of exploit in which the attacker starts out with access to a normal user account on the system (perhaps gained by sniffing passwords, a dictionary attack, or social engineering) and is able to exploit some vulnerability to gain root access to the system. Therefore, for U2R attacks, we selected features such as "number of file creations" and "number of shell prompts invoked," while we ignored features such as "protocol" and "source bytes."

Remote to Local Layer

Remote to Local attacks (R2L) occurs when an attacker who has the ability to send packets to a machine over a network but who does not have an account on that machine exploits some vulnerability to gain local access as a user of that machine. Therefore, for R2L attacks we selected both the network level features such as the "duration of connection" and "service requested" and the host level features such as the "number of failed login attempts" among others for detecting R2L attacks. Since each layer is independent of every other layer, the feature set for the layers is not disjoint.

ALGORITHM FOR ASSIMILATING CRF WITH PLSM

Our intrusion detection system should provide two main requirements accuracy and efficiency. With the PLSM model it will improve the overall system efficiency. CRFs will improve the attack detection accuracy. Therefore by assimilating both CRFs and PLSM model into a single system model which is accurate in detecting attacks and efficient of operation.

Algorithm for Training Layers

To train layers in our proposed system we have to follow the following steps.

1. Select the number of layers in our system.
2. Perform feature selection for each layer separately which is discussed in section 5
3. Prepare a separate model with CRFs for each layer using the features selected in step 2.
4. Block in training models linearly such that only the instances labeled as normal are passed to next layer.

Algorithm for Testing Instances

To test connection test instances which are passed through layers has to follow the following steps.

1. For each test instance perform following steps.
2. Test the instance and label it as either attack or normal.
3. If the instance is labeled as attack then block it and identify the type of attack which is represented by layer name at which it is detected and go to step 1. Otherwise pass the instance to next layer follow the next step 4 and if the current layer is layer 1which is Free State assign layer to another multicore processor and go to step 1.
4. If the current layer is not the last layer in the system then test the instance and go to step 3. Otherwise go to step 5.
5. Test the instance and label it either attack or normal. If it is labeled as an attack, block it and identify the attack which is corresponding with the layer name otherwise there is no attack in the connection test instance.

With the above two algorithms, we improves both attack detection accuracy and the efficiency of the system. Therefore we assimilate both CRFs and PLSM into single system.

International Journal of Advanced Trends in Computer Science and Engineering, Vol.2 , No.1, Pages : 01-06 (2013)
Special Issue of ICACSE 2013 - Held on 7-8 January, 2013 in Lords Institute of Engineering and Technology, Hyderabad

CONCLUSION

In this paper, we discussed about layered security model to improve attack detection accurately and CRFs are assimilated to layered security model to improve overall efficiency of the system. We applied pipelining concept to layered security model (termed PLSM) to get high performance intrusion detection system in multicore processors. With this PLSM model, it improves latency and throughput of intrusion detection system when compare to the layered security model. When compared to our model with other well known methods for intrusion detection, our assimilated pipelined model detecting all attacks such as Dos, probe, R2L and U2R attacks with effectively, efficiently and accurately.

REFERENCES

[1] Autonomous Agents for Intrusion Detection, http://www.cerias. purdue.edu/research/aafid/, 2010.
[2] Probabilistic Agent Based Intrusion Detection, http://www.cse.sc.edu/research/isl/agentIDS.shtml, 2010.
[3] T. Abraham, IDDM: Intrusion Detection Using Data Mining Techniques, http://www.dsto.defence./gov.au/publications/ 2345/DSTO-GD-0286.pdf, 2008.
[4] N.B. Amor, S. Benferhat, and Z. Elouedi, "Naive Bayes vs. Decision Trees in Intrusion Detection Systems," Proc. ACM Symp. Applied Computing (SAC '04), pp. 420-424, and 2004.
[5] 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.
[6] P. Dokas, L. Ertoz, A. Lazarevic, J. Srivastava, and P.-N. Tan, "Data Mining for Network Intrusion Detection," Proc. NSF Workshop Next Generation Data Mining (NGDM '02), pp. 21-30, 2002.
[7] K.K. Gupta, B. Nath, and R. Kotagiri, "Network Security Framework," Int'l J. Computer Science and Network Security, vol. 6, no. 7B, pp. 151-157, 2006.
[8] K.K. Gupta, B. Nath, and R. Kotagiri, "Conditional Random Fields for Intrusion Detection," Proc. 21st Int'l Conf. Advanced Information Networking and Applications Workshops (AINAW '07), pp. 203-208, 2007.
[9] K.K. Gupta, B. Nath, R. Kotagiri, and A. Kazi, "Attacking Confidentiality: An Agent Based Approach," Proc. IEEE Int'l Conf. Intelligence and Security Informatics (ISI '06), vol. 3975, pp. 285-296, 2006.
[10] J. Lafferty, A. McCallum, and F. Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data," Proc. 18th Int'l Conf. Machine Learning (ICML '01), pp. 282-289, 2001.
[11] W. Lee and S. Stolfo, "Data Mining Approaches for Intrusion Detection," Proc. Seventh USENIX Security Symp. (Security '98), pp. 79-94, 1998.
[12] W. Lee, S. Stolfo, and K. Mok, "Mining Audit Data to Build Intrusion Detection Models," Proc. Fourth Int'l Conf. Knowledge Discovery and Data Mining (KDD '98), pp. 66-72, 1998.
[13] W. Lee, S. Stolfo, and K. Mok, "A Data Mining Framework for Building Intrusion Detection Model," Proc. IEEE Symp. Security and Privacy (SP '99), pp. 120-132, 1999.
[14] A. McCallum, "Efficiently Inducing Features of Conditional Random Fields," Proc. 19th Ann. Conf. Uncertainty in Artificial Intelligence (UAI '03), pp. 403-410, 2003.