Detection of Intrusion using Layered Approach with Conditional Random Fields

SURESH BABU.CH[#], SATISH[*], T.RAJESH[#]

[#]*II M.TECH, Department of Computer Science & Engineering, ASR College of Engineering, JNTUK, Tanuku, A.P.*

[*]*HOD & Assistant Professor of CSE, ASR College of Engineering, JNTUK, Tanuku, A.P.*

Under esteemed guidance of

[#]*Assistant Professor of CSE, ASR College of Engineering, JNTUK, Tanuku, A.P.*

Abstract— 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 the large amount of network traffic. In this paper, we address these two issues of Accuracy and Efficiency using Conditional Random Fields and Layered Approach. We demonstrate that high attack detection accuracy can be achieved by using Conditional Random Fields and high efficiency by implementing the Layered Approach. Experimental results on the benchmark KDD ’99 intrusion data set show that our proposed system based on Layered Conditional Random Fields outperforms other well-known methods such as the decision trees and the naive Bayes. The improvement in attack detection accuracy is very high, particularly, for the U2R attacks (34.8 percent improvement) and the R2L attacks (34.5 percent improvement). Statistical Tests also demonstrate higher confidence in detection accuracy for our method. Finally, we show that our system is robust and is able to handle noisy data without compromising performance.

Keywords— Intrusion detection, Layered Approach, Conditional Random Fields, network security, decision trees, naive Bayes.

I. INTRODUCTION

Intrusion detection as defined by the SysAdmin, Audit, Networking and Security (SANS) Institute is the art of detecting inappropriate, inaccurate, or anomalous activity [6]. Today, intrusion detection is one of the high priority and challenging tasks for network administrators and security professionals. More sophisticated security tools mean that the Attackers come up with newer and more advanced penetration methods to defeat the installed security systems [4] and [5]. Thus, there is a need to safeguard the networks from known vulnerabilities and at the same time take steps to detect new and unseen, but possible, system abuses by developing more reliable and efficient intrusion detection systems. Any intrusion detection system has some inherent requirements. Its prime purpose is to detect as many attacks as possible with minimum number of false alarms, i.e., the system must be accurate in detecting attacks. However, an accurate system that cannot handle large amount of network traffic and is slow in decision making will not fulfill the purpose of an intrusion detection system. We desire a system that detects most of the attacks, gives very few false alarms, copes with large amount of data, and is fast enough to make real-time decisions.

Intrusion detection started in around 1980s after the influential paper from Anderson [6]. Intrusion detection systems are classified as network based, host based, or application based depending on their mode of deployment and data used for analysis. Additionally, intrusion detection systems 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.

The rest of this paper is organized as follows: In Section 2, we discuss the related work with emphasis on various methods and frameworks used for intrusion detection. We describe the use of Conditional Random Fields (CRFs) for intrusion detection in Section 3 and the Layered Approach in Section 4. We then describe how to integrate the Layered Approach and the CRFs in Section 5. In Section 6, we give our experimental results and compare our method with other approaches that are known to perform well. We observe that our proposed system, Layered CRFs, performs significantly better than other systems. We study the robustness of our method in Section 7 by introducing noise in the system. We discuss feature selection in Section 8 and draw conclusions in Section 9.

II. RELATED WORK

The field of intrusion detection and network security has been around since late 1980s. Since then, a number of methods and frameworks have been proposed and many systems have been built to detect intrusions. Various techniques such as association rules, clustering, naive Bayes classifier, support vector machines, genetic algorithms, artificial neural networks, and others have been applied to detect intrusions. In this section, we briefly discuss these techniques and frameworks.

We compare the Layered Approach with the works in [2], [4], and [5]. The authors describe the combination of “strong” classifiers using stacking, where the decision tress, naive

Bayes, and a number of other classification methods are used as base classifiers. The authors show that the output from these classifiers can be combined to generate a better classifier rather than selecting the best one. In the authors use a combination of “weak” classifiers. The individual classification power of weak classifiers is slightly better than random guessing. The authors show that a number of such classifiers when combined using simple majority voting mechanism, provide good classification. In the authors apply a combination of anomaly and misuse detectors for better qualification of analyzed events. However, our work is not based upon classifier combination. Combination of classifiers is expensive with regard to the processing time and decision making. The purpose of classifier combination is to improve accuracy. Rather, our system is based upon serial layering of multiple hybrid detectors. From our experiments in Section 6, we show that the Layered CRFs perform better than individual classifiers and they are more efficient and accurate than a system based on classifier combination. The results from individual classifiers at a layer are not combined at any later stage in the Layered Approach, and hence, an attack can be blocked at the layer where it is detected. There is no communication overhead among the layers and the central decision-maker. In addition, since the layers are independent they can be trained separately and deployed at critical locations in a network depending upon the specific requirements of a network. Using a stacked system will not give us the advantage of reduced processing when an attack is detected at the initial layers in the sequential model.

In this paper, we show the effectiveness of CRFs for intrusion detection. Motivated by our results, we perform detailed analysis and show that CRFs are a strong candidate for building robust intrusion detection systems. We then show that high efficiency can be achieved by implementing the Layered Approach. Finally, we integrate the Layered Approach and the CRFs to develop a system that is accurate and performs efficiently.

III. CONDITIONAL RANDOM FIELDS FOR INTRUSION DETECTION

Conditional models are probabilistic systems that are used to model the conditional distribution over a set of random variables. Such models have been extensively used in the natural language processing tasks. Conditional models offer a better framework as they do not make any unwarranted assumptions on the observations and can be used to model rich overlapping features among the visible observations. Maxent classifiers, maximum entropy Markov models and CRFs are such conditional models. The advantage of CRFs is that they are undirected and are, thus, free from the Label Bias and the Observation Bias. The simplest conditional classifier is the Maxent classifier based upon maximum entropy classification, which estimates the conditional distribution of every class given the observations. The training data is used to constrain this conditional distribution while ensuring maximum entropy and hence maximum uniformity. We now give a brief description of the CRFs, which is motivated from the work in [6].

A. MOTIVATING EXAMPLE

The data analyzed by the intrusion detection system for classification often has a number of features that are highly correlated and complex relationships exist between them. For example, when classifying network connections as either normal or as attack, a system may consider features such as “logged in” and “number of file creations.” When these features are analyzed individually, they do not provide any information that can aid in detecting attacks.

However, when these features are analyzed together, they can provide meaningful information, which can be helpful for the classification task. Taking another example, the connection level feature such as the “service invoked” at the destination provides some information about the class label (in case an attacker sends request to a service that is not available). This information becomes more concrete and aids in classification when analyzed with other features such as “protocol type” and “amount of data transferred” between source and destination (in case the client connects to an available service such as the ftp and performs data transfer). These relationships, between different features in the observed data, if considered during classification can significantly decrease classification error. The CRFs do not consider features to be independent and hence perform better when compared with other methods.

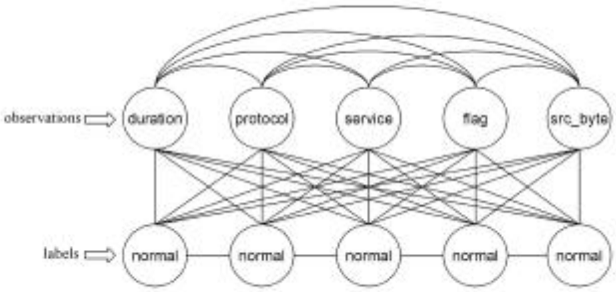


Fig. 1. Graphical representation of a CRF

The data set used in our experiments represents features of every session in relational form with only one label for the entire record. In this case, using a conditional model would result in simple maximum entropy.

However, we represent the data in the form of a sequence and assign a label to every feature in the sequence using the first-order Markov assumption instead of assigning a single label to the entire observation. Though, this increases the complexity but it also increases the attack detection accuracy. Each record represents a separate connection, and hence, we consider every record as a separate sequence. 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 duration, protocol, service, flag, and src_bytes 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. As it is evident from the figure, 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.

As we will show from our experimental results, the 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.

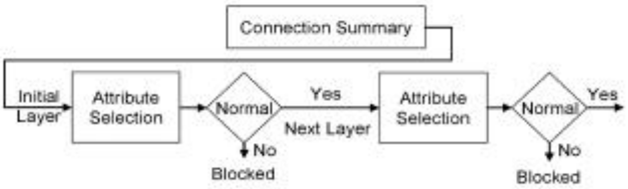


Fig.2. Layered Representation.

Our first goal is to improve the attack detection accuracy we first compare the accuracy of CRFs for detecting attacks with other methods in Section 6. We consider all the 41 features in the data set for each of the four attack groups separately. As we shall observe, the CRFs outperform other methods for detecting "Unauthorized access to Root" (U2R) attacks. They are also effective in detecting the Probe, "Remote to Local" (R2L), and "Denial of Service" (DoS) attacks. However, CRFs can be expensive during training and testing. For a simple linear chain structure, the time complexity for training a CRF is OðT L2NI Þ, where T is the length of the sequence, L is the number of labels, N is the number of training instances, and I is the number of iterations. During inference, the Viterbi algorithm is employed, which has a complexity of OðT L2Þ. The quadratic complexity is significant when the number of labels is large as in language tasks. However, for intrusion detection, there are only two labels "normal" and "attack," and thus, the system is very efficient. We further improve the overall system performance by using the Layered Approach, which decreases T, i.e., the length of the sequence. The Layered Approach is described next.

IV LAYERED APPROACH FOR INTRUSION DETECTION

We now describe the Layer-based Intrusion Detection System (LIDS) in detail. The LIDS draws its motivation from what we call as the Airport Security model, where a number of security checks are performed one after the other in a sequence. Similar to this model, the LIDS represents a sequential Layered Approach and is based on ensuring availability, confidentiality, and integrity of data and (or) services over a network. Fig. 2 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. Every layer in the LIDS framework is trained separately and then deployed sequentially. We define four layers that 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 layers essentially act as filters that block any anomalous connection, thereby eliminating the need of further processing at subsequent layers enabling quick response to intrusion. The effect of such a sequence of layers is that the anomalous events are identified and blocked as soon as they are detected.

Our second goal is to improve the speed of operation of the system. Hence, we implement the LIDS and select 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. In many situations, there is a trade-off between efficiency and accuracy of the system and there can be various avenues to improve system performance. Methods such as naive Bayes assume independence among the observed data. This certainly increases system efficiency, but it may severely affect the accuracy. To balance this trade-off, we use the CRFs that are more accurate, though expensive, but we implement the Layered Approach 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.

V. INTEGRATING LAYERED APPROACH WITH CONDITIONAL RANDOM FIELD

In Section 1, we discussed two main requirements for an intrusion detection system; accuracy of detection and efficiency in operation. As discussed in Sections 3 and 4, respectively, the CRFs can be effective in improving the attack detection accuracy by reducing the number of false alarms, while the Layered Approach can be implemented to improve the overall system efficiency. Hence, a natural choice is to integrate them to build a single system that is accurate in detecting attacks and efficient in operation. Given the data, we first select four layers corresponding to the four attack groups (Probe, DoS, R2L, and U2R) and perform feature selection for each layer, which is described next.

A. FEATURE SELECTION

Ideally, we would like to perform feature selection automatically. However, as will be discussed later in Section 8, the methods for automatic feature selection were not found to be effective. In this section, we describe 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 category. 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.

1) Probe Layer

The probe attacks are aimed at acquiring information about the target network from a source that 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 files creations" and "number of files accessed" are not expected to provide information for detecting probes.

2) DoS Layer

The DoS attacks are meant to force the target to stop the service(s) that is (are) provided by flooding it with illegitimate requests. Hence, 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" are significant. To detect DoS attacks, it may not be important to know whether a user is "logged in or not."

3) R2L Layer

The R2L attacks are one of the most difficult to detect as they involve the network level and the host level features. We therefore 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.

4) U2R Layer

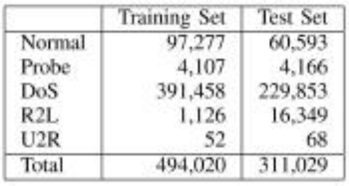
The U2R attacks involve the semantic details that are very difficult to capture at an early stage. Such attacks are often content based and target an application. Hence, for U2R attacks, we selected features such as "number of file creations" and "number of shell prompts invoked," while we ignored features such as "protocol" and "source bytes." We used domain knowledge together with the practical significance and the feasibility of each feature before selecting it for a particular layer. Thus, from the total 41 features, we selected only 5 features for Probe layer, 9 features for DoS layer, 14 features for R2L layer, and 8 features for U2R layer. Since each layer is independent of every other layer, the feature set for the layers is not disjoint. The selected features for all the four layers are presented in Appendix A. We then use the CRFs for attack detection as discussed in Section 3. However, the difference is that we use only the selected features for each layer rather than using all the 41 features. We now give the algorithm for integrating CRFs with the Layered Approach.

Algorithm
Training

Step 1: Select the number of layers, n, for the complete system.
Step 2: Separately perform features selection for each layer.
Step 3: Train a separate model with CRFs for each layer using the features selected from Step 2.
Step 4: Plug in the trained models sequentially such that only the connections labeled as normal are passed to the next layer.

Testing
Step 5: For each (next) test instance perform Steps 6 through 9.
Step 6: Test the instance and label it either as attack or normal.
Step 7: 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 5. Else pass the sequence to the next layer.
Step 8: If the current layer is not the last layer in the system,
Test the instance and go to Step 7. Else go to Step 9.
Step 9: 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.

TABLE 1
Data Set



	Training Set	Test Set
Normal	97,277	60,593
Probe	4,107	4,166
DoS	391,458	229,853
R2L	1,126	16,349
U2R	52	68
Total	494,020	311,029

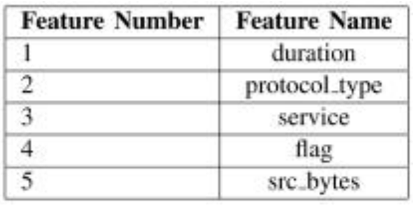
Our final goal is to improve both the attack detection accuracy and the efficiency of the system. Hence, we integrate the CRFs and the Layered Approach to build a single system. We perform detailed experiments and show that our integrated system has dual advantage. First, as expected, the efficiency of the system increases significantly. Second, since we select significant features for each layer, the accuracy of the system further increases. This is because all the 41 features are not required for detecting attacks belonging to a particular attack group. Using more features than required can result in fitting irregularities in the data, which has a negative effect on the attack detection accuracy of the system.

VI CONCLUSION

In this paper, we have addressed the dual problem of Accuracy and Efficiency for building robust and efficient intrusion detection systems. Our experimental results in Section 6 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 Layered Approach significantly reduce the time required to train and test the model. Even though we used a relational data set for our experiments, we showed that the sequence labeling methods such as the CRFs can be very effective in detecting attacks and they outperform other methods that are known to work well with the relational data. We compared our approach with some well-known methods and found that most of the present methods for intrusion detection fail to reliably detect R2L and U2R attacks, while our integrated system can effectively and efficiently detect such attacks giving an improvement of 34.5 percent for the R2L and 34.8 percent for the U2R attacks. We also discussed how our system is implemented in real life. Our system can help in identifying an attack once it is detected at a particular layer, which expedites the intrusion response mechanism, thus minimizing the impact of an attack. We showed that our system is robust to noise and performs better than any other compared system even when the training data is noisy. 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. The areas for future research include the use of our method for extracting features that can aid in the development of signatures for signature-based systems. The signature-based systems can be deployed at the periphery of a network to filter out attacks that are frequent and previously known, leaving the detection of new unknown attacks for anomaly and hybrid systems. Sequence analysis methods such as the CRFs when applied to relational data give us the opportunity to employ the Layered Approach, as shown in this paper. This can further be extended to implement pipelining of layers in multicore processors, which is likely to result in very high performance.

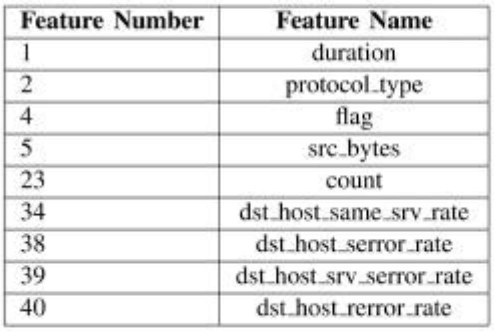
FEATURE SELECTION

1 Feature Selected for Probe Layer



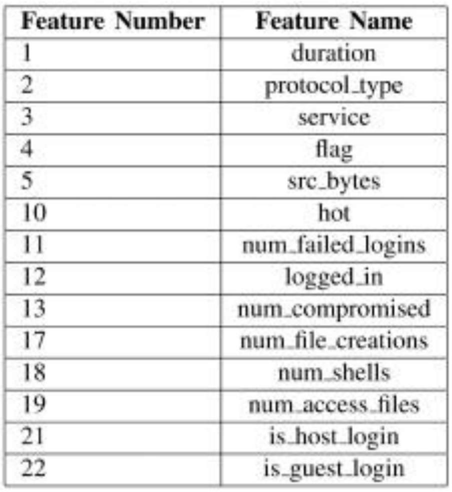
Feature Number	Feature Name
1	duration
2	protocol_type
3	service
4	flag
5	src_bytes

2 Features Selected for DoS Layer



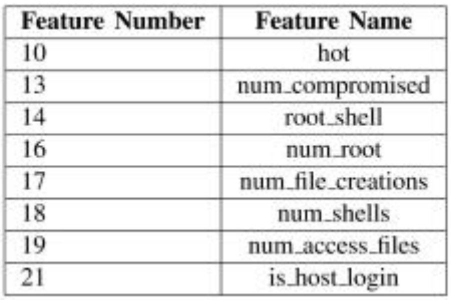
Feature Number	Feature Name
1	duration
2	protocol_type
4	flag
5	src_bytes
23	count
34	dst_host_same_srv_rate
38	dst_host_serror_rate
39	dst_host_srv_serror_rate
40	dst_host_rerror_rate

3 Features Selected for R2L Layer



Feature Number	Feature Name
1	duration
2	protocol_type
3	service
4	flag
5	src_bytes
10	hot
11	num_failed_logins
12	logged_in
13	num_compromised
17	num_file_creations
18	num_shells
19	num_access_files
21	is_host_login
22	is_guest_login

4 Features Selected for U2R Layer



Feature Number	Feature Name
10	hot
13	num_compromised
14	root_shell
16	num_root
17	num_file_creations
18	num_shells
19	num_access_files
21	is_host_login

REFERENCES

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

[2] CRF++: Yet Another CRF Toolkit, http://crfpp.sourceforge.net/, 2010.

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

[4] Overview of Attack Trends, http://www.cert.org/archive/pdf/attack_trends.pdf, 2002.

[5] Probabilistic Agent Based Intrusion Detection, http://www.cse.sc.

[6] 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