A General Framework for Adaptive Anomaly Detection with Evolving Connectionist Systems

Yihua Liao* V. Rao Vemuri*† Alejandro Pasos†

Abstract

Anomaly detection techniques hold great potential for combating novel intrusions, malicious insiders and fast spreading worms. A widely acknowledged challenge in anomaly detection is how to accurately model a subject's normal behavior in the presence of concept drift. This paper presents a new adaptive anomaly detection framework that aims to adapt to normal behavior changes while still recognizing anomalies. Evolving connectionist systems are employed to learn a subject's behavior in an online, adaptive fashion without *a priori* knowledge of the underlying data distributions.

1 Introduction

As one of the two general approaches to intrusion detection, anomaly detection has been under intensive study for the last two decades [1]. Unlike the alternative approach, misuse detection, which generates an alarm when a known attack signature is matched, anomaly detection tries to identify activities that deviate from the normal behavior of the monitored system, network or users. Anomaly detection techniques hold great potential for combating novel intrusions and malicious insiders as well as fast and wildly spreading worms.

A number of anomaly detection methodologies have been developed over the years. These techniques can be further categorized as *generative* or *discriminative* approaches. A generative approach (e.g., [2–4]) builds a model solely based on the set of normal training examples and evaluates each testing instance to see how well it fits the model. A discriminative approach (e.g., [5]), on the other hand, attempts to learn the distinction between the normal and abnormal classes. Both normal and attack examples are used in training for discriminative approaches. Generative approaches are more realistic due to the fact that attack examples are usually very rare in practice.

Regardless of the approach used, most previous work did not address the issue of *concept drift*, which refers to the process that a subject's normal behavior patterns change over time in a continuous manner. In a practical environment, however, system and network activities as well as user behavior can change for bona fide reasons. Modeling a subject's normal behavior in the presence of concept drift is a challenging task because the underlying data distribution is not known *a priori*, unexpected changes may happen at any time, and therefore the normal behavior may not be strictly predictable in the long term. The key to this difficult problem is adaptive learning. An effective anomaly detection system should be capable of adapting to normal behavior changes while still recognizing anomalous activities. Otherwise, large amount of false alarms would be generated if the model failed to change adaptively to accommodate the new patterns [6]. A seemingly obvious solution is to update the training corpus with each new set of audit data and re-build the normal behavior model. However, it is not computationally feasible for most existing methods (e.g., [5] [7] [8]) because they are expensive to generate a model and not suitable for incremental, adaptive learning.

In this paper, we present a general adaptive anomaly detection framework that is applicable to host-based and network-based intrusion detection without *a priori* knowledge of the underlying data distributions. Our framework employs unsupervised evolving connectionist systems to learn system, network or user behavior in an online, adaptive fashion. The evolving connectionist systems are designed for modeling evolving processes [9]. They are stable enough to retain patterns learned from previously observed data while being flexible enough to learn new patterns from new incoming data. The structure of the intelligent connectionist systems evolves in time. New input data, including new features and new patterns, are naturally accommodated through efficient local element tuning.

2 Adaptive Anomaly Detection Framework

In addressing the problem of adaptive anomaly detection two fundamental questions arise: (a) How to generate a model or profile that can concisely describe a subject's normal behavior, and more importantly, can it be updated efficiently to accommodate new behavior patterns? (b) How to select instances to update the model without introducing noise and incorporating abnormal

*Department of Computer Science, University of California, Davis, One Shields Ave, Davis, CA 95616, USA

†Department of Applied Science, University of California, Davis, One Shields Ave, Davis,CA 95616, USA

patterns as normal? Our adaptive anomaly detection framework addresses these issues through the use of online unsupervised learning methods, under the assumption that normal instances cluster together in the input space, whereas the anomalous activities correspond to outliers that lie in sparse regions of the input space. Our framework is general in that the underlying clustering method can be any online unsupervised evolving connectionist system and it can be used for different types of audit data. Without loss of generality, we assume the audit data that is continuously fed into the adaptive anomaly detection system has been transformed into a stream of input vectors after pre-processing, where the input features describe the monitored subject's behavior.

Adaptive learning and evolving connectionist systems are an active area of artificial intelligence research. Evolving connectionist systems are artificial neural networks that resemble the human cognitive information processing models. They operate continuously in time and adapt their structure and functionality through a continuous interaction with the environment [9]. Due to their self-organizing and adaptive nature, they provide powerful tools for modeling evolving processes and knowledge discovery. They can learn in unsupervised, supervised or reinforcement learning modes. The online unsupervised evolving connectionist systems provide one-pass clustering of an input data stream, where there is no predefined number of different clusters that the data belong to.

A simplified diagram of an evolving connectionist system for online unsupervised learning is given in Figure 1 (some systems may have an additional fuzzy input layer, which represents the fuzzy quantization of the original inputs with the use of membership functions [10]). A typical unsupervised evolving connectionist system consists of two layers of nodes: an *input* layer that reads the input vectors into the system continuously, and a *pattern* layer (or cluster layer) representing previously learned patterns. Each pattern node corresponds to a cluster in the input space. Each cluster is in turn represented by a weight vector. Then the subject's normal behavior profile is conveniently described as a set of weight vectors that represent the clustering of the previous audit data.

A distance measure has to be defined to measure the mismatch between a new instance (i.e., a new input vector) and existing patterns. Based on the distance measure, the system either assigns an input vector to one of the existing patterns and updates the pattern weight vector to accommodate the new input, or otherwise creates a new pattern node for the input. The details of clustering vary with different evolving connectionist systems.

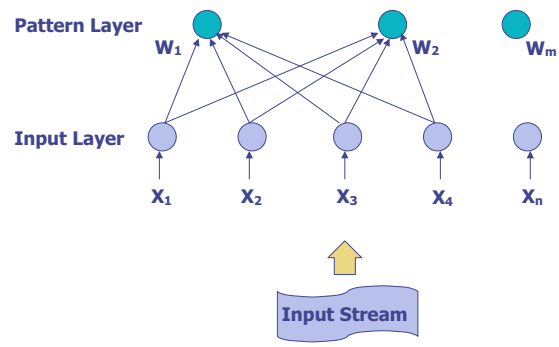

Figure 1: A simplified diagram of an evolving connectionist system for unsupervised learning.

In order to reduce the risk of false alarms (classifying normal instances as abnormal), we define three states of behavior patterns (i.e., the pattern nodes of the evolving connectionist system): *normal*, *uncertain* and *anomalous*. Accordingly, each instance is labeled as either *normal*, *uncertain* or *anomalous*. In addition, the alarm is differentiated into two levels: *Level 1 alarm* and *Level 2 alarm*, representing different degrees of anomaly. As illustrated in Figure 2, a new instance is assigned to one of the existing normal patterns and labeled *normal* if the similarity between the input vector and the normal pattern is above a threshold (the *vigilance* parameter). Otherwise, it is *uncertain*. The *uncertain* instance is either assigned to one of the existing *uncertain* patterns if it is close enough to that *uncertain* pattern, or becomes the only member of a new *uncertain* pattern. A *Level 1 alarm* is triggered whenever a new *uncertain* pattern is created as the new instance is different from all the learned patterns and thus deserves special attention. At this point, some preliminary security measures need to be taken. However, one can not draw a final conclusion yet. The new instance can be truly anomalous or merely the beginning of a new normal behavior pattern, which will be determined by the subsequent instances. After the processing of a certain number (the N_{watch} parameter) of the subsequent instances in the same manner, if the number of members of an *uncertain* pattern reaches a threshold value (the Min_{count} parameter), the *uncertain* pattern becomes a *normal* pattern and the labels of all its members are changed from *uncertain* to *normal*. This indicates that a new behavior pattern has been developed and incorporated into the subject's normal behavior profile as enough instances have shown the same pattern. On the other hand, after N_{watch} subsequent instances, any *uncertain* pattern with less than Min_{count} members will be destroyed and all its members are la-

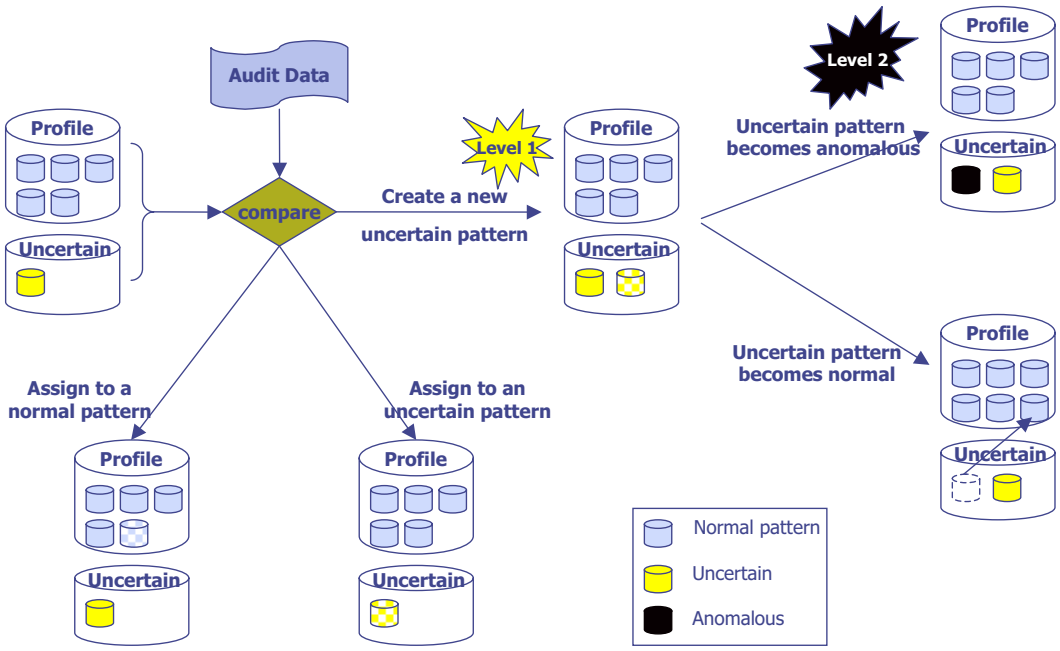

Figure 2: Adaptive anomaly detection framework.

beled *anomalous*. This will make sure that anomalous patterns, corresponding to the sparse regions in the input space, will not be included into the normal profile. A *Level 2 alarm* is issued when an instance is labeled *anomalous* and further response actions are expected.

The main tunable parameters of an adaptive anomaly detection system are summarized as follows:

- *Vigilance*. This threshold controls the degree of mismatch between new instances and existing patterns that the system can tolerate.
- *Learning rate*. It determines how fast the system should adapt to a new instance when it is assigned to a pattern.
- N_{watch}. It is the period that the system will wait before making a decision on a newly created *uncertain* pattern.
- Min_{count}, the minimum number of members that an *uncertain* pattern should have in order to be recognized as a *normal* pattern.

Our framework does not require *a priori* knowledge of the number of input features. When a new input feature is presented, the system simply adds a new input node to the input layer and connections from this newly created input node to the existing pattern nodes. This can be very important when the features that describe a subject's behavior grow over time and can't be foreseen in a dynamic environment. Similarly, accommodation of a new pattern is efficiently realized by creating a new pattern node and adding connections from input nodes to this new pattern node. The rest of the structure remains the same.

With the framework, the learned normal profile is expressed as a set of weight vectors representing the coordinates of the cluster centers in the input space. These weight vectors can be interpreted as a knowledge presentation that can be used to describe the subject's behavior patterns, and thus they can facilitate understanding of the subject's behavior. The weight vectors are stored in the long term memory of the connectionist systems. Since new instances are compared to all previously learned patterns, recurring activities would be recognized easily.

3 Related Work

Among the few adaptive anomaly detection systems, NIDES [11] is probably the best known. The statistical component of NIDES generates user profiles that are constantly aged by multiplying them by an exponential decay factors. This method of aging creates a moving time window for the profile data, so that the new behavior is only compared to the most recently observed behaviors that fall into the time window. One drawback of NIDES is that a user's recurring behavior can cause

frequent and unnecessary updates of the profile, let alone its complex statistical model.

Teng et al. [12] used inductively generated sequential patterns to perform adaptive real time anomaly detection. Lane and Brodley [13] proposed an nearest neighbor classifiers based online learning scheme and examined the issues of incremental updating of system parameters and instance selection. Mixture models were employed in [14] and [15] to generate adaptive probabilistic models and detect anomalies within a dataset. Cannady [16] demonstrated the use of a reinforcement learning method that uses feedback from the protected system. Fan [17] used ensembles of classification models to adapt existing models in order to detect newly established patterns. Hossain and Bridges [18] proposed a fuzzy association rule mining architecture for adaptive anomaly detection.

Compared to previous statistical or rule-learning based adaptive anomaly detection systems, our framework does not require *a priori* knowledge of the underlying data distributions. Through the use of evolving connectionist systems, it provides efficient adaptation to new patterns in a dynamic environment. Unlike other neural networks that have been applied to intrusion detection (e.g., [3] [19]) as "black boxes", our evolving connectionist systems can provide knowledge to "explain" the learned normal behavior patterns.

Our approach also falls into the category of unsupervised anomaly detection [20–22] as it does not require the knowledge of data labels. However, our algorithms assign each instance into a cluster in an online, adaptive mode. No distinction between training and testing has to be made. Therefore the period of system initialization during which all behaviors are assumed normal is not necessary.

4 Discussion

There are certain limitations with our adaptive anomaly detection framework. Our approach assumes that the number of normal instances vastly outnumbers the number of anomalies, and the anomalous activities appear as outliers in the data. This approach would miss the attacks or masquerades if the underlying assumptions do not hold. Furthermore, it is possible that one can deliberately cover his malicious activities by slowly changing his behavior patterns without triggering a *level2 alarm*. However, a *level1 alarm* is issued whenever a new pattern is being formed. It is then the security officer's responsibility to identify the user's intent in order to distinguish malicious from non-malicious anomalies, which is beyond the scope of this paper.

5 Conclusion

This paper has presented a new adaptive anomaly detection framework, based on the use of evolving connectionist systems, to address the issue of concept drift. A subject's normal behavior patterns are discovered by online, adaptive learning.

Future work is in progress to explore automated determination of the parameters and empirically test our adaptive anomaly detection framework using evolving connectionist systems such as Fuzzy Adaptive Resonance Theory (ART) [23] and evolving self-organizing maps.

Acknowledgment

This work is supported by the AFOSR grant F49620-01-1-0327 to the Center for Digital Security of the University of California, Davis.

References

[1] J. McHugh, "Intrusion and intrusion detection," *International Journal of Information Security*, vol. 1, pp. 14–35, 2001.

[2] H. Javitz and A. Valdes, "The nides statistical component description and justification," Computer Science Laboratory, SRI International, Menlo Park, CA, Tech. Rep., 1993.

[3] H. Debar, M. Becker, and D. Siboni, "A neural network component for an intrusion detection system," in *Proceedings of IEEE Symposium on Security and Privacy*, Oakland, CA, May 1992, pp. 240–250.

[4] Y. Liao and V. R. Vemuri, "Use of k-nearest neighbor classifier for intrusion detection," *Computers & Security*, vol. 21, no. 5, pp. 439–448, Oct. 2002.

[5] W. Hu, Y. Liao, and V. R. Vemuri, "Robust support vector machines for anomaly detection in computer security," in *Proceedings of the 2003 International Conference on Machine Learning and Applications (ICMLA'03)*, Los Angeles, CA, June 2003.

[6] R. A. Maxion and T. N. Townsend, "Masquerade detection using truncated command lines," in *Proceedings of International Conference on Dependable Systems & Networks*, Washington DC, June 2002, pp. 219–228.

[7] W. Lee, S. Stolfo, and P. Chan, "Learning patterns from unix process execution traces for intrusion detection," in *Proceedings of the AAAI97 workshop on AI methods in Fraud and risk management*, July 1997, pp. 50–56.

[8] C. Warrender, S. Forrest, and B. Pearlmutter, "Detecting intrusions using system calls: Alternative data models," in *Proceedings of IEEE Symposium on Security and Privacy*, Oakland, CA, May 1999, pp. 133–145.

[9] N. Kasabov, *Evolving Connectionist Systems: Methods and Applications in Bioinformatics, Brain Study and Intelligence Machines.* Springer, 2002.

[10] J.-S. R. Jang, C.-T. Sun, and E. Mizutani, *Neuro-Fuzzy and Soft Computing: A computational approach to learning and machine intelligence.* Prentice Hall, 1997.

[11] T. F. Lunt, "Detecting intruders in computer systems," in *Proceedings of Conference on Auditing and Computer Technology*, 1993. [Online]. Available: http://www.sdl.sri.com/nides/index5.html

[12] H. S. Teng, K. Chen, and S. C. Lu, "Adaptive real-time anomaly detection using inductively generated sequential patterns," in *Proceedings of IEEE Symposium on Security and Privacy*, Oakland, CA, May 1990, pp. 278–284.

[13] T. Lane and C. Brodley, "Approaches to online learning and conceptual drift for user identification in computer security," ECE and the COAST Laboratory, Purdue University, Tech. Rep. Coast TR 98-12, 1998.

[14] E. Eskin, M. Miller, Z.-D. Zhong, G. Yi, W.-A. Lee, and S. Stolfo, "Adaptive model generation for intrusion detection systems," in *Proceedings of the ACM-CCS Workshop on Intrusion Detection and Prevention*, Athens, Greece, 2000.

[15] K. Yamanishi, J. Takeuchi, and G. Williams, "On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms," in *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Boston, MA, Aug. 2000, pp. 320–324.

[16] J. Cannady, "Next generation intrusion detection: Autonomous reinforcement learning of network attacks," in *Proceedings of the 23rd National Information Systems Security Conference*, 2000.

[17] W. Fan, "Cost-sensitive, scalable and adaptive learning using ensemble-based methods," Ph.D. dissertation, Columbia University, Feb. 2001.

[18] M. Hossain and S. M. Bridges, "A framework for an adaptive intrusion detection system with data mining," in *Proceedings of the 13th Annual Canadian Information Technology Security Symposium*, Ottawa, Canada, June 2001.

[19] A. K. Ghosh and A. Schwartzbard, "A study in using neural networks for anomaly and misuse detection," in *Proceedings of the Eighth USENIX Security Symposium*, 1999.

[20] E. Eskin, A. Arnold, M. Prerau, L. Portnoy, and S. Stolfo, "A geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data," in *Applications of Data Mining in Computer Security*, D. Barbara and S. Jajodia, Eds. Kluwer, 2002.

[21] A. Lazarevic, L. Ertoz, V. Kumar, A. Ozgur, and J. Srivastava, "A comparative study of anomaly detection schemes in network intrusion detection," in *Proceedings of the 3rd SIAM International Conference on Data Mining*, San Francisco, California, 2003.

[22] A. Valdes, "Detecting novel scans through pattern anomaly detection," in *Proceedings of DARPA Information Survivability Conference and Exposition*, Washington, DC, April 2003, pp. 140–151.

[23] G. A. Carpenter, S. Grossberg, and D. B. Rosen, "Fuzzy art: Fast stable learning and categorization of analog patterns by an adaptive resonance system," *Neural Networks*, vol. 4, pp. 759–771, 1991.