# Applying CMAC-Based On-Line Learning to Intrusion Detection

James Cannady
Ph.D. Candidate
School of Computer and Information Sciences
Nova Southeastern University
Fort Lauderdale, FL 33314

cannadyj@scis.nova.edu

## Abstract

*The timely and accurate detection of computer and network system intrusions has always been an elusive goal for system administrators and information security researchers. Existing intrusion detection approaches require either manual coding of new attacks in expert systems or the complete retraining of a neural network to improve analysis or learn new attacks. This paper presents a new approach to applying adaptive neural networks to intrusion detection that is capable of autonomously learning new attacks rapidly by a modified reinforcement learning method that uses feedback from the protected system.*

Keywords: Intrusion detection, CMAC, denial of service attacks.

## Introduction

Because of the increasing dependence which companies and government agencies have on their computer networks the importance of protecting these systems from attack is critical. A single intrusion of a computer network can result in the loss, unauthorized utilization, or modification of large amounts of data and cause users to question the reliability of all of the information on the network. There are numerous methods of responding to a network intrusion, but they all require the accurate and timely identification of the attack. The individual creativity of attackers, the wide range of computer hardware and operating systems, and the ever-changing nature of the overall threat to targeted systems have contributed to the difficulty in effectively identifying intrusions. There are currently two primary approaches to detecting intrusions. Anomaly detection involves identifying activities that vary from established patterns for users, or groups of users. The technique typically involves the creation of knowledge bases that contain the profiles of the monitored activities. The second general approach to intrusion detection, misuse detection, involves the comparison of a user's activities with the known behaviors of attackers attempting to penetrate a system. While anomaly detection typically utilizes threshold monitoring to indicate when a certain established metric has been reached, misuse detection techniques frequently utilize a rule-based expert systems. When applied to misuse detection, the rules become scenarios for network attacks. Unfortunately, since expert systems have no capability for autonomous learning they require frequent updates by a system administrator to remain current. When a new form of attack is identified the signature must be manually encoded as a rule in the expert system for it to be identified in the network stream and these updates may be ignored or performed infrequently by the administrator. Rule-based systems also suffer from a lack of flexibility in the rule-to-audit record representation. Slight variations in an attack sequence can affect the activity-rule comparison to a degree that the attack is not detected by the intrusion detection mechanism.

While intrusion detection systems (IDS) generally rely on expert systems to identify attacks a limited amount of research has been conducted on the application of neural networks to address the inherent weaknesses in rule-based approaches. Debar (1992) and Fox (1990) proposed neural networks as alternatives to the statistical analysis component of anomaly detection systems. These neural networks identify the typical characteristics of system users and identify statistically significant variations from the user's established behavior. Cannady (1998) demonstrated the use of multi-level perceptron/SOM hybrid neural networks in the identification of computer attacks and Bonifacio (1998) demonstrated the use of a neural network in an integrated detection system. However, each of these approaches required the complete retraining of the neural networks to learn new attacks.

This paper presents the results of a research effort that investigated the application of a cerebellar model articulation controller (CMAC) neural network (Albus, 1975) in the detection of denial-of-service (DoS) attacks. An attacker carries out a DoS attack by making a computer resource inoperative, by taking up so much of a shared resource that none of the resource is left for other users, or by degrading the resource so that it is less valuable to users. The attacker can flood the system with repeated requests for meaningless processes, continually sending the host garbage data, causing the host to initiate a re-boot, or other similar actions. Each of these attacks reduces or eliminates the availability of computer resources. In a ping flood attack the host machine is rapidly sent ECHO requests by an attacker. The response to each of these requests limits the amount of available system memory for other processes. As the number of successive requests is sent the protected host may slow to a stop as it attempts to manage the increased activity. A similar type of attack, known as a UDP Packet Storm attack, relies on a rapid succession of UDP packets to overwhelm the host.

There were three objectives of the research effort described in this paper:

1. Determine if a CMAC neural network could recognize activity that represented a denial of service attack.
2. Evaluate the ability of a CMAC neural network to recognize new patterns through generalization.
3. Test the ability of a CMAC neural network to autonomously learn new attacks on-line.

While each of these capabilities provide advances over most existing approaches to intrusion detection, the ability to autonomously learn new attack patterns without manual updating or retraining was the primary focus of the work. The inability of existing systems to autonomously identify new attacks increases the long-term cost of the systems due to the requirement for dedicated personnel to identify and implement the necessary updates. However, more significant is the fact that the lack of autonomous learning by existing approaches results in an IDS that is only as current as the most recent update and therefore becomes progressively more ineffective over time.

## CMAC-based On-line Learning Approach

The CMAC algorithm was selected for this approach primarily because of the capability for on-line learning. While the prior work on neural network-based intrusion detection demonstrated the ability to accurately identify network attacks, those efforts were unable to demonstrate the ability to improve analyses without completely retraining the neural network. Like the original CMAC proposed by Albus the neural network that was used in this effort utilized a binary kernel function. The CMAC also used 5000 memory cells that were randomly mapped from the generalized input vectors. The input vectors were ordered sets of normalized floating point numbers that represented types of Ethernet network packets, (e.g., ping, telnet, FTP, etc.). The packets were collected from a live network stream using a Linux-based network sniffer prior to being classified. The classified packet types were arranged in the order that they were received into data vectors consisting of 10 elements each. Like other standard implementations of the CMAC algorithm the single output was a weighted sum of the memory cells addressed by the generalized vectors.

The first phase of the research involved training the CMAC with "normal" and simulated ping flood attack network activity for nine iterations followed by a single test iteration. Three hundred data vectors were used, of which fifty represented an increasing and subsequently decreasing denial of service attack. Traditional least mean square (LMS) learning algorithm was used to update the CMAC weights ($W$) based on multiplying the difference in the desired output ($O_d$) and actual output ($O_a$) by a positive learning factor ($ß$):

$$w_{i+1} = w_i + ß\,(O_d - O_a)$$

This CMAC was designed to utilize feedback from the protected host ($S$) to produce an output that represented the probability of an attack. The feedback from the protected host was in the range 0.0 (system stopped) to 1.0 (system optimal). Each feedback value was based on the combined effect of the input packets on a variety of system state indicators, (e.g., CPU load, available memory, network load, etc.). The output ranged from 0.0 (no attack) to .99 (definite attack) and should be inversely proportional to $S$. While receiving normal network data the state of the protected host should be nominal, (e.g., 0.75 - 0.99), and the corresponding CMAC output should be small, (e.g., 0.0

– 0.25). However, during a DoS attack the state of the protected host becomes degraded as the system reacts to the flood of packets and $S$ is reduced. Since the input to the CMAC was limited to the network events and system feedback the LMS learning algorithm was modified to incorporate the feedback by using the inverse of the state of the protected host $(1 – S)$ as the desired output from the CMAC:

$$w_{i+1} = w_i + ß((1-S) - O_a)$$

The standard LMS learning algorithm was further modified to respond more effectively in a network environment. The inverse of the host state, $(1 – S)$ was used in place of a constant learning factor to increase the flexibility of the learning rate:

$$w_{i+1} = w_i + (1-S)((1-S) - O_a)$$

While the typical use of a constant learning factor in LMS algorithm implementations will usually settle to an acceptable error level, the use of a single learning factor prevents the system from varying the rate at which new information is learned. The ability to vary the learning factor may offer the advantage of allowing a system to increase or decrease learning rates in response to circumstances in a dynamic environment. By using $(1 – S)$ as the learning factor the CMAC developed in this work was designed to learn faster when the state of the protected host was degraded and at a lower rate when the state was nominal. This results in an adaptive learning rate that responds to the current level of potential threat to the protected host.

The second phase of the work involved testing the ability of the CMAC to generalize sufficiently to identify and learn a new (*a priori*) attack pattern. While the first phase of the evaluation had used a ping flood attack as part of the network data the second phase used data vectors that represented a UDP Packet Storm attack. These attacks are sufficiently dissimilar to provide an accurate evaluation of the on-line learning ability of the CMAC. Five different UDP Packet Storm vectors of varying severity were used as input. The CMAC was evaluated based on the initial presentation of each of the five vectors prior to receiving host feedback and then on a second presentation of the same vectors after the host had provided feedback.

## Results

The first phase of the evaluations tested the ability of the CMAC to identify patterns of network activity (both normal and attacks). While the CMAC was provided with the same data vectors for nine training iterations consistent results were noted from the second test through the ninth. This indicated that the CMAC was achieving optimal learning during the second presentation of each series of data.

The first test used a constant learning factor of 0.75. While several tests were conducted with learning factors in the range 0.0 – 1.0, the use of the learning factor of 0.75 was the most accurate for the static learning implementations. The test resulted in an average error of 3.24%. This error rate is significantly better than the average error rate of 15% in commercial IDSs, (Bonifacio, 1998). Figure 1 shows the desired and actual (CMAC) response (neural network output) to each of the 300 network events.

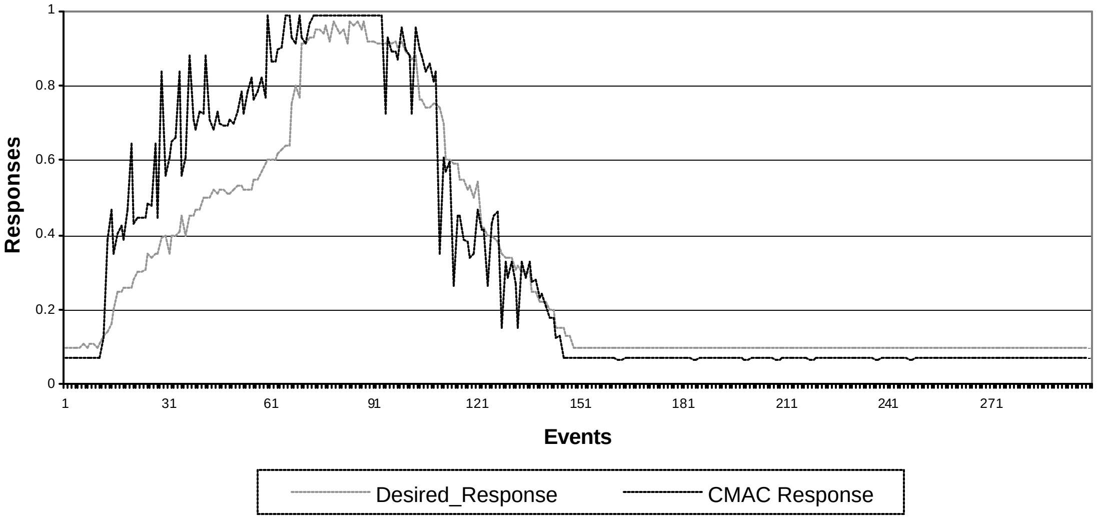

**Figure 1: Static Learning Factor Results**

The subsequent test of the CMAC using the same input vectors and an adaptive learning factor based on the state of the host resulted in an average error of 0.12%, (Figure 2). The results of this test demonstrate the improved accuracy of the adaptive learning factor approach in identifying network activity when compared to the static learning approach.

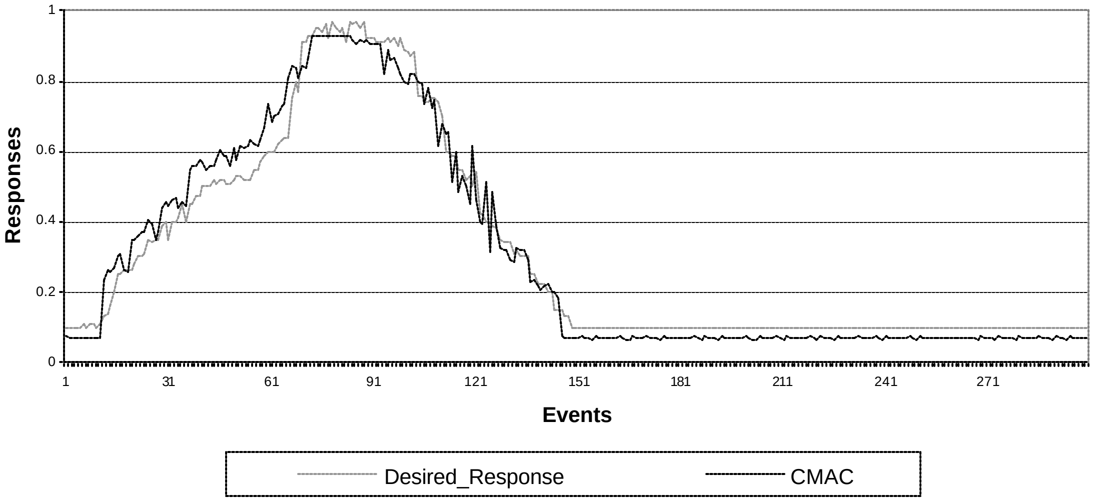

**Figure 2: Adaptive Learning Factor Results**

After completing the tests of the pattern recognition capability of the CMAC the second phase tests were conducted using the adaptive learning factor approach with five different UDP Packet Storm attacks of varying severity. The initial presentation of the five new attacks resulted in an average error of 15.2%, (Figure 3). This average error of the CMAC based on an initial presentation of each attack is similar to results obtained from exis ting intrusion detection approaches that have been thoroughly trained to recognize the activity.

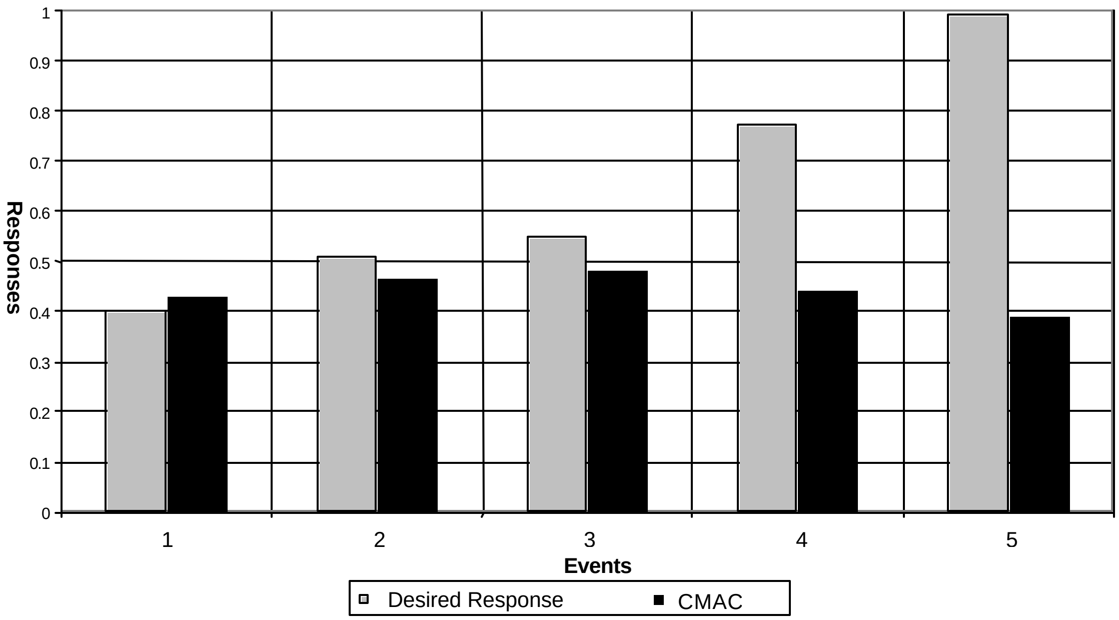

**Figure 3: Initial results to *a priori* attacks**

After the initial test of the CMAC with the *a priori* attacks the state of the protected host after receiving the attacks was provided to the CMAC as feedback. A subsequent presentation of the *a priori* attacks resulted in an average error of the CMAC output of 0.4%, (Figure 4). This test demonstrated that the CMAC approach was able to autonomously identify network attack with extreme accuracy on the second presentation of each attack.

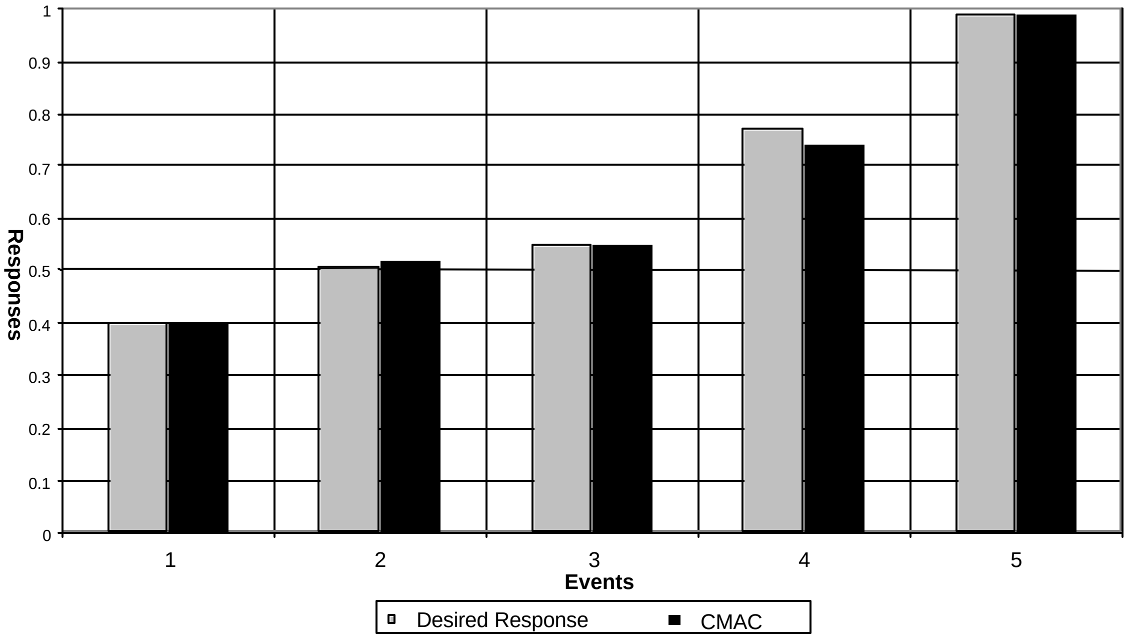

**Figure 4: Final results to *a priori* attacks**

## Conclusions

Research and development of IDSs has been ongoing since the early 1980's and the challenges faced by designers increase as the targeted systems because more diverse and complex. The results demonstrate the potential for a powerful new analysis component of a complete IDS that would be capable of identifying *priori* and *a priori* denial of service attack patterns. Based on the results of the tests that were conducted on this approach there were several significant advances in the detection of network attacks:

- On-line learning of attack patterns – The approach has demonstrated the ability to rapidly learn new attack patterns without the complete retraining required in other neural network approaches. This is a significant advantage that could allow the IDS to continually improve its analytical ability without the requirement for external updates.

- Rapid learning of data – The CMAC was able to accurately identify the data vectors after only a single training iteration. This is a significant improvement over other neural network approaches that may require thousands of training iterations to accurately learn patterns of data.

- Extremely accurate in identifying *priori* attack patterns – The use of the dynamic learning factor resulted in an average error of 0.12%, compared with an average error of 15% in existing IDSs. Because other information security components rely on the accurate detection of computer attacks the ability to accurately identify network events could greatly enhance the overall security of computer systems.

- Immediate identification of *a priori* attacks - The approach has demonstrated the ability to effectively identify potential attacks during the initial presentation prior to receiving feedback from the protected host. While the error in the response was higher than during subsequent presentations of the pattern after feedback had been received, the average error rate of 15.2% is consistent with normal results from existing IDSs. In addition, the ability of this approach to utilize generalization to provide some indication of attack is an advantage over expert system approaches that require an exact match to coded patterns to provide an alert.

- Adaptive learning algorithm– The use of an adaptive learning factor, based on the current state of the protected host, provides the ability to rapidly learn new attacks, thereby significantly reducing learning time in periods when rapid attack identification is required.

The results of the tests of this approach shows significant promise, and our future work will involve the application of this approach to other complex forms of attacks which are typically addressed through misuse detection. We are also developing a full-scale integrated intrusion detection and response system that will incorporate the CMAC-based approach as the analytical component.

## References

Albus, J.S. (1975, September). A New Approach to Control: The Cerebellar Model Articulation Controller (CMAC). Transactions of the ASME.

Bonifacio, J.M, Cansian, A.M., de Carvalho, A., & Moreira, E. (1998). Neural Networks Applied in Intrusion Detection. In Proceedings of the International Joint Conference on Neural Networks.

Cannady, J. (1998). Applying Neural Networks to Misuse Detection. In Proceedings of the 21st National Information Systems Security Conference.

Debar, H. & Dorizzi, B. (1992). An Application of a Recurrent Network to an Intrusion Detection System. In Proceedings of the International Joint Conference on Neural Networks.

Fox, Kevin L., Henning, Rhonda R., & Reed, Jonathan H. (1990). A Neural Network Approach Towards Intrusion Detection. In Proceedings of the 13th National Computer Security Conference.