Hybrid Packet Filtering for overcoming DDoS Attacks against AMI components

Hasty Atashzar[1]
Iran Energy Efficiency Organization[1](SABA)
Tehran, Iran
Atashzar@saba.org.ir[1]

Hadi Modaghegh
Iran Energy Efficiency Organization[1](SABA)
Tehran, Iran
modaghegh@saba.org.ir

Abstract— Advanced Metering Infrastructures have great potential to improve the stability and reliability of the electric power grid and will become key tools to empower consumers in the energy market. However, if they are not based upon a secure system architecture, they could in fact become one of the grid's most significant liabilities, due to their expected pervasive deployment.
One of the most significant cyber attacks which threaten the availability of AMI components is DDoS attack.
This paper concentrates on applying machine learning techniques to defence against DDOS attacks in AMI system.
Accordingly, an effective classifier, PCNN, is proposed. PCNN classifier consists of two major parts: Principal Component Analysis (PCA) for feature extraction and MLP Neural Network (NN) for packet classification. Experimental dataset generated by Spirnet Avalanche in our Lab has been employed to evaluate and examine the proposed classifier. Results demonstrate that PCNN classifier is able to classify the incoming traffic with less than 7% false acceptance rate and less than 4% false rejection rate at the edge of the victim network.

I. INTRODUCTION

Advanced Metering Infrastructures have great potential to improve the stability and reliability of the electric power grid and will become key tools to empower consumers in the energy market. However, if they are not based upon a secure system architecture, they could in fact become one of the grid's most significant liabilities, due to their expected pervasive deployment [1,2].
One of the most significant cyber attacks which threaten the availability of AMI components is Distributed Denial of Service, DDoS, attack [3,4,5,6].

DDoS attacks are one of the most critical security issues on the Internet. DDOS attacks were first seen in June 1998 [7,8]. These attacks typically exhaust the network bandwidth, capacity of processing or network stack resources and disturb the network connectivity to the victims.

Generally, there are three main approaches to defend against DDoS attacks; DDoS prevention, DDoS detection and DDoS traffic filtering. This paper concentrates on classifying incoming traffic (while an attack has been launched) which is the main part of third approach. There are two categories of traffic classification approaches at the victim site, signature-detection and anomaly-detection based methods [9,10,11,12].

In this paper, based on anomaly-detection viewpoint, an efficient classifier, PCNN, has been proposed which consists of two major parts. At first, an efficient traffic feature is extracted by applying PCA to the initial traffic data (which has been previously gathered from the packet header fields). Then an Artificial Neural Network has been employed to classify and filter the incoming traffic under DDoS attack.

PCNN classifier performance is directly influenced by its position in the network. So it is important to implement the defensive system at the right place. Although, traceback and source-end methods defence against DDoS close to the sources of the attack, a solution closed to the victim is more practical due to the availability of AMI network management information. So, PCNN classifier should be implemented at the edge of the protected network which is the Centeral system in our architecture.

The rest of the paper is organized as fallows. Section 2 describes the proposed method for classifying network traffics under DDoS attack. Section 3 explains how PCA and ANN have been applied to cooperate effectively. Section 4 presents the collected data and the experimental results. Section 5 concludes the paper, and expresses the open fields of research in the proposed framework, in order to be improved in future.

II. PROPOSED METHOD

The proposed framework, PCNN classifier, is composed of three phases which will be explained in this section; Training phase, feature extraction phase and traffic filtering phase. These parts cooperate effectively to classify the network traffics into malicious packets and normal packets during an attack.

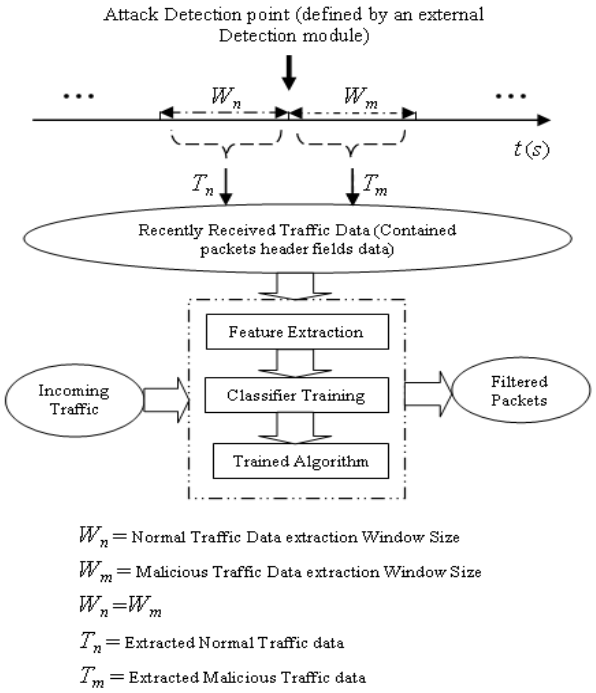

Figure. 1. An Overview of the proposed framework

A. Training Phase

As shown in Figure 1, whenever an attack is detected (which is done by an external detection module), salient values of packets header fields from recently received traffic in a window before the detection point, W_n, and a window after the detection point, which is named W_m in figure 1 , are captured. The initial training traffic data is called T_n (which is stand for captured data in W_n). It will be applied as normal traffic training data set and T_m (which is stand for captured data in W_m), will be applied as malicious traffic training data set. However, in real networks, T_n and T_m are not pure and contains some percentages of impurity. In the other words, T_m may contain legitimate packets data as well as T_n may contain attack packets data. If α represents the ratio of impurity in T_m, and β represents the ratio of impurity in T_n then T_m and T_n are defined as

$$T_m = (1-\alpha)t_a + \alpha t_l \quad (1)$$

$$T_n = (1-\beta)t_l + \beta t_a \quad (2)$$

Where t_a stands for pure attack packets data and t_l stands for pure legitimate packets data.

As mentioned before, an ANN has been used to classify and filter the incoming traffic under DDoS attack. Although ANN learning is robust to errors in training data, in this case, the experimental results (which will be presented in section 4) show that if the impurity of training data sets rises, ANN will not able to be trained well. Therefore in order to overcome this problem and diminish the effects of impurity, PCA algorithm has been applied to the training data sets before training. So during training stage, one half of the data has been used to train the classifier and the other half has been used to evaluate the training results.

B. Feature Extraction Phase

Although there is no agreement on what kind of traffic should be viewed as "abnormal", the traditional knowledge is that the traffic generated by DDoS attacks usually exhibits some specific characteristics [4]. So, in this framework various components of traffic have been collected to indicate traffic behavior in DDoS domain distinctively

TABLE 1. Extracted Header fields to make traffic feature vector

Packet Type	Field 1	Field 2	Field 3	Field 4	Field 5
IP header	TLenn	ID	SrcIPadd	DstIPadd	---
TCP header	SrcPort	DstPort	flags	WinSize	TCPlen
UDP header	SrcPort	DstPort	UDPlen	---	---

For this purpose, at first, the extraction module has not focused on a single set of features but extracted salient values from IP header (OSI Layer3) fields and TCP/UDP header (OSI Layer4) fields. Table 1 gives an overview over the collected traffic data. So through this stage, high dimensional traffic data sets are captured. Although extracting various prominent values from packets header intensify the traffic features, high dimensionality of captured data sets poses many serious problems during classification procedure. Moreover, as mentioned before, in real networks, samples collected during an attack contain percentages of normal packets. So traffic data sets captured through this stage are both high dimensional and impure.

To overcome these problems PCA have been applied to the collected traffic data sets. As the result of this process, the dimension size of traffic data sets has been reduced and the impurity has been filtered out. So efficient traffic features have finally been extracted.

III. PCA & ANN

Principal Component Analysis (PCA) is a procedure for recognizing patterns beyond data by compressing the data to emphasize their similarities and differences. Principal component analysis conventionally is carried out on the symmetric covariance matrix or on the symmetric correlation matrix [13].

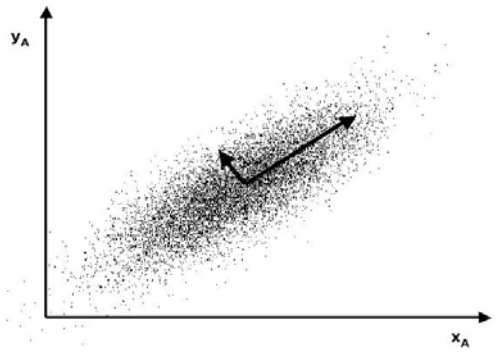

Figure.2. Applying PCA, principal dimensions of data has been recognized [13]

In fact, PCA is a statistical linear transformation that transforms the data to a new coordinate system such that the greatest eigenvalue of the data covariance matrix, lies on the first coordinate (called the first principal component), the second greatest eigenvalue lies on the second coordinate, and so on. Figure 2 lucidly shows that how PCA extracts the main axes on which data exhibits the most variability [13].

To apply this process to DDoS area, at first during training stage, a database of traffic data should be defined. For this purpose, Initial database, which contains j samples of traffic, was generated by extracting the packet header fields as shown in Table 1. So each sample represents a packet characterized by its extracted header fields.

In order to filter out the initial traffic data impurity and reduce the dimension of traffic data and training time, PCA was applied to the matrix form of Initial database; matrix I and Training database was generated. Matrix I is a $j \times k$ matrix which "j" is the number of packets collected during training stage and "k" denotes the number of extracted header fields shown in Table 1. So Training database represents packets expressed by new features that are more appropriate to train the classifier. A simple three-layer feed-forward neural network with sigmoid transfer function has been used to classify the traffic.

In order to train the classifier, Training database, generated by PCA, was fed into the ANN input layer. The number of the input neurons was equal to the number of the features extracted by PCA. A single output neuron was used to indicate the possibility of a packet being malicious or normal. The sigmoid function was used to map the output of each computation back into the range [0, 1]. The ANN was trained to output a value of 1 for malicious packet and a value of 0 for normal packet. At the end of this process, the fully trained algorithm would be able to classify the incoming attacked traffic.

During the classification stage, at first, a string of initial traffic data shown in Table 1 was collected from each packet header fields. Then the eigen vectors produced by PCA in the training stage, was applied to the vector form of each packet data string, and its efficient features was extracted. Finally a string of extracted features for each packet was fed into the fully trained algorithm to be classified as a normal packet or a malicious one. Figure 3 gives a general view of the proposed classifier.

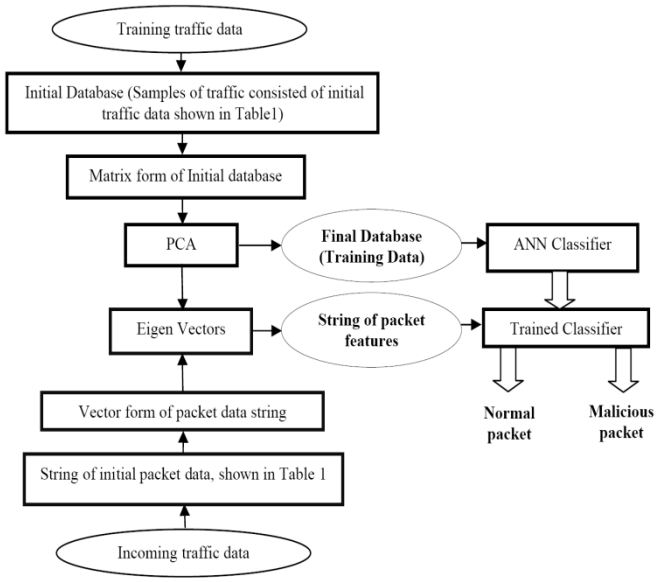

Figure. 3. General view of PCNN classifier

IV. EXPERIMENTAL RESULTS

The IDS evaluation appliance, Spirnet Avalanche 290 was applied for the evaluation of the proposed method. Various tests were carried out, and following sections show the achievements [14].

Spirent Avalanche datasets contains thousands of attacks (including DDoS, Worms, E-mail attacks, Viruses/Trojans/Malware, VoIP attacks, Application penetration, Evaded Attacks/Fragmentation, Port Scanning/Port Corruption, Buffer Overflows/Protocol Exploitation, Create millions of flooding attacks per second) to evaluate intrusion detection systems.

In order to evaluate the proposed classifier, 200 simulated clients were created with Avalanche (in IP address range: 192.168.42.25-192.168.42.225) as attackers. A real server with CAS applications (Windows server 2008 at IP address: 192.168.42.21) was used as a victim. To create training and test data set, Normal traffic and DDoS attacks were emulated with avalanche.

Avalanche contains several DDoS attack profiles, In this case JETCAST_SRVR_DOS, BMPTIFF_DOS and LIBTIFF_DOS profiles were applied for the evaluation [6]. Traffic packets captured with Wireshark and required fields (shown in Table 1) were extracted.

One half of the extracted data was used for training the classifier and the other half was used for evaluating the trained classifier. To optimize the training time, each database contained 1000 malicious packets and 1000 normal packets due to window size of 0.3(s).

Then using these samples, various tests were performed. Several experiments were carried out to evaluate the

efficiency of PCNN classifier and a number of experiments were performed to find out the practical training time for ANN (which resulted in window size of 0.3(s)). The results of these experiments are exhibited in the following sections.

A. Evaluation of PCNN efficiency

As mentioned, in a real-world scenario the defensive system has to deal with impure data sets. So to understand the effects of this impurity on the classification results, dirty data sets were produced. In these data sets, α is considered 30%, and β varies from 5% to 40% which refers to 5 attack scenarios.

At first, PCNN ability was evaluated while PCA had not been applied. Then, PCA was applied and the same experiments were carried out. Figures 4 to 7 show the results of these experiments.

Figure 4 shows the results of the experiments while PCA has not applied. According to the figure, with the pure data sets there was no false accepts and no false rejects. But when some impurity were added, False Acceptance Rate (FAR) and False Rejection Rate (FRR) were increased considerably, while, due to high error rate, the classifier could not practically distinguish between malicious packets and normal packets when α is equal to 20%.

To defeat the impurity effects on the classification results and to reduce the dimension of traffic data, PCA was applied to the training data sets and several experiments were carried. During these experiments numbers of reduced dimensions and the effect of PCA on the classification results were assessed.

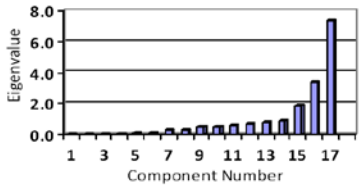

Figure. 4. Eigenvalues of a training data set while α is equal to 40% and β is considered 30%he proposed framework

Figure 4 shows the eigenvalues of an impure training data sets covariance matrix. According to the figure, six first eigenvalues are significantly smaller than the others. So to extract the main axes of traffic data, six first eigenvectors related to the six first eigenvalues are eliminated and the 11 left eigenvectors construct the main directions of traffic data.

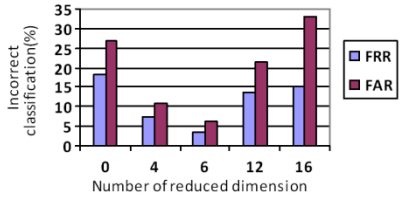

Figure. 5. False Acceptance (FAR) and false Rejection Rates (FRR) while α is equal to 40% and β is considered 30%

To investigate the effect of the number of eliminated eigenvectors on the classification results, various experiments have been performed. Figure 5 shows the results. As it is presented in Figure 5, when six dimensions of traffic data has been reduced, PCNN classifier provides the best results as FAR is less than 8% and FRR is less than 5%. In compare of previous results (which acquired while PCA has not been applied), FAR has improved about 87% and FRR has improved about 92%. Consequently one can be derived that, under the circumstance in which window size is 0.3(s) and dimension of initial traffic data (which was collected from packets header fields) is 17, PCNN classifier provides the best outcome by reducing 6 dimension of initial traffic data.

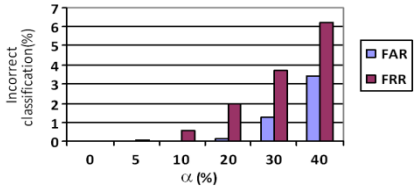

Figure. 6. FAR and FRR while α is equal to 40% and β is considered 30%

To exhibit the PCA effect on the classifier ability, several experiments were carried out. The experiments evaluated the performance of PCNN classifier via various percent of impurity in training traffic data. According to results shown in Figure 5, six dimensions of traffic data had been reduced in all the experiments. The results have been shown in Figure6.

Figure 6 shows that PCNN classifier can be effectively applied in the presence of dirty training data and so it will be able to deal with the real attack scenario (when the baseline traffic contains malicious packets and attack traffic contains normal packets). As expected, by raising the impurity, the classification ability is slightly decreased and some malicious packets were accepted while some normal packets will be dropped. As shown in Figure 6, while α is equal to 40% and β is considered 30%, FAR is less than 7% and FRR is less than 4%.

B. Evaluation of window size

All of the experiments performed to evaluate the proposed framework, have been carried out under special circumstance in which the window size is 0.3(s) and baseline traffic contains 30% of malicious packets and attack traffics (according to various attack scenario intention) contain wide range of normal packets percentage.

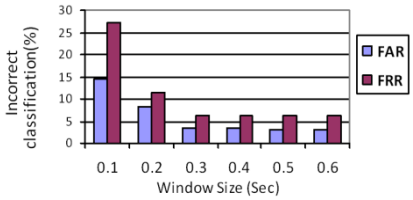

Figure. 8. False Acceptance (FAR) and false Rejection Rates (FRR) with various window size while α is equal to 40% and β is considered 30%

In fact these experiments have been performed while the external detection module has detected the attack when the malicious packets occupy 30% of baseline traffic in the last 0.3(s) before the detection point.

Since, the performance of the proposed framework, PCNN classifier, is influenced by processing time, minimizing this time would cause better performance. One of the conceivable methods of decreasing the processing time is to decrease the window size to the minimum possible value. According to this point of view, several experiments were performed. The experiments evaluated PCNN classifier via different values of window size. Figure 7 shows the results. As shown in Figure 7 when the window size is less than 0.3(s) the classification performance will decrease and FAR and FRR will exceed the desirable results, while the system perform effectively with the window size of 0.3 (s) or higher. So the minimum possible window size is 0.3(s). However the minimum possible window size depends on the attack detection point, and if the detection point changes, the minimum possible size of window will change too. In this paper, all of the experiments have been performed with a certain detection point.

V. CONCLUSION

Securing AMI Systems against DDoS attacks is still a complicated problem. Unless trackback and source-end schemes become universally established, a defence close to the victims (Data Concentrators and Centeral Systems) is the most practical solution. In this paper an effective classifier, PCNN, based on Principle Component Analysis and Artificial Neural Networks was proposed to filter out the malicious data packet after attack detection. Instead of identifying a single metric to classify DDoS attack traffics, data from different levels of the network stack was collected. To reduce the dimension of the collected traffic data and extract more effective features, PCA was applied. When the resources utilization levels exceed the normal thresholds, traffic features generated by PCA were fed into the ANN to train the classifier. If the training process was successful then it could be used to classify the incoming traffics and filter out the malicious packets.

Experimental results show, the proposed framework, PCNN classifier, was able to classify the traffic packets with less than 7% false acceptance rate and less than 4% false rejection rate.These errors are lower than 10% which are acceptable error due to the nature of AMI systems.

Future works can concentrate on variety of concepts related to the framework proposed in this paper. One of them is finding the relation between window size and attack detection point. The other area for future works is applying other machine learning techniques. Although ANNs was used as the machine learning algorithm of our choice, the proposed framework could easily be extensible with different algorithms. Future works may be concentrated on extending this solution to use multiple machine learning/intelligent algorithms.

REFERENCES

[1] S. Borenstein, M. Jaske, and A. Rosenfeld. Dynamic pricing, advanced metering and demand response in electricity markets. Center for the Study of Energy Markets, Oct. 31, 2002.

[2]S. McLaughlin, D. Podkuiko, and P. McDaniel, \Energy Theft in the Advanced Metering
Infrastructure," in Proceedings of 4th International Conference on Critical Information Infrastructures Security, 2009.

[3] European Network and Information Security Agency (ENISA), “Smart Grid Security Recommendations for Europe and Member States”, 01-07-2012.

[4] National Institute of Standards and Technology (NIST). NISTIR 7628: Guidelines for Smart Grid Cyber Security. Smart Grid Interoperability Panel–Cyber Security Working Group (SGIP–CSWG). 2010.

[5]F. Cleveland, BCyber security issues for advanced metering infrastructure (AMI),[in Proc. Power Energy Soc. Gen. MeetingVConv. Delivery Electr. Energy 21st Century, Apr. 2008, DOI: 10.1109/PES.2008.4596535.

[6]X. Wang and P. Yi, “Security framework for wireless communications in smart distribution grid,” Smart Grid, IEEE Transactions on, vol. 2, no. 4, pp. 809 –818, dec. 2011.

[7]I. Shin, J. H. Huh, C. Lee, and D. M. Nicol, “A Monitoring Architecture for Smart Meter Mesh Networks in the Smart Grid,” (submitted for review), 2011.

[8]D. Jin, C. Lee, D. Nicol, I. Shin, and H. Zhu, Simulation-based Study of Distributed Denial-of-Service Attacks in Advanced Metering Infrastructure, INFORMS Annual Meeting, Charlotte, NC, USA, November 2011.

[9]A. K. Ghosh, A. Schwartzbard, A study in Using Neural Networks for Anomaly and Misuse Detection, *Proc. 8th USENIX Security Symposium* 1999.

[10]R. Berthier, W. Sanders, and H. Khurana, “Intrusion detection for advanced metering infrastructures: Requirements and architectural directions,” in Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on, oct. 2010, pp. 350 –355.

[11] T. Alpcan and T. Basar, Network Security: A Decision and Game Theoretic Approach. Cambridge, U.K.: Cambridge Univ. Press, 2011.

[12] J. P. Hespanha, P. Naghshtabrizi, and Y. Xu, BA survey of recent results in networked control systems,[Proc. IEEE, vol. 95, no. 1, pp. 138–162, Jan. 2007.

[13] I. T. Jollife Principal Components Analysis. Heidelberg: Springer; 2002.

[14]] Avalanche, Available: http://www.spirent.com/Products/Avalanche/Avalanche_Latest_Release, retrieved: Mar. 2013