A Study on IDS Based CMAC Neuron Network to Improve the Attack Detection Rate

Trong-Minh Hoang[1(✉)] and Trang-Linh Le Thi[2]

[1] Posts and Telecoms Institute of Technology, Ha Noi, Vietnam
hoangtrongminh@ptit.edu.vn
[2] Electric Power University, Ha Noi, Vietnam

Abstract. The massive growth of the Internet of Things has brought a lot of attractive benefits because it is going to have a positive impact on life and work through many applications. Besides its advantages, the adoption of massive applications also points the door for attackers to gain cyberattacks on the system. Hence, needed solutions to detect attacks from the edge of the network must be considered to reduce the pressure on the computing elements in core networks. Therefore, approximate approaches to low computational complexity in an Intrusion Detection System (IDS) are being studied to favor limited-resource devices. In this study, a novel IDS based intelligent computation is proposed, the Cerebellar Model Articulation Controller (CMAC) neuron network is chosen to tailor various hardware edge devices. Moreover, to approach edge processing, a feature selection reduction scheme is proposed to reduce the time complexity of the training phase while keeping reasonable accuracy. The experimental results are compared to other previous studies in the same input conditions to high-light the proposed advantages.

Keywords: Security · IDS · Neuron network · Machine learning · Dataset

1 Introduction

Today, Internet of things (IoT) applications are being developed in mass to meet the needs of automation for the economic, medical, or agricultural industries. Communication network architectures force the processing hierarchy to move closer to the end-users for faster processing and reduce the pressure on the core network infrastructure. Therefore, intelligent computing solutions and information processing at the edge of the network are considered key technologies for new generation networks such as 5G and 6G [1].

Security for the IoT network is becoming more and more important as the number of devices increases rapidly with a variety of devices. Attacks can come from any device and lead to serious damage to network infrastructure. Moreover, attacks based on smart IoT devices can cause rich vulnerable issues. Hence, IDS based on intelligent methodologies is needed for such scenarios.

These solutions are intended to deal with the intelligence of attacks in open environments. Historical data streams are identified and regionally sorted to confirm current

N.-S. Vo et al. (Eds.): INISCOM 2021, LNICST 379, pp. 504–511, 2021.
https://doi.org/10.1007/978-3-030-77424-0_39

unusual events. In it, neural networks are trained and classified for threshold values. However, the time and algorithmic complexity of AI solutions is always a big challenge [2]. Therefore, finding a solution suitable for edge network computing characteristics is a research direction recently. Along with this approach, this study provides an intelligent edge processing solution to deal with attacks with a feature reduction method while still ensuring the accuracy of the model. The results proving the validity of the proposal presented in the paper have been compared with the previous research results. The proposed model gives high accuracy and less feature number than the recent proposals on the UNSW-NB15 dataset. The paper structure is organized as follows: The next section presents related work; Sect. 3 briefs the primary and base principles; The detailed proposal is illustrated in Sect. 4; in the last section, our conclusions and future works are presented.

2 Related Work

In recent years, the need to compute and manage huge IoT devices has become imperative to adapt to massive application growth. Therefore, cloud computing systems have been decentralized towards the edge of the network. Along with that, security systems also transfer some functions to different cloud layers to ensure the safety and efficiency of the whole system [3]. To early prevent attacks from IoT devices and reduce compute load to the processing center, IDS intrusion detection systems are migrated to the network edge to detect and predict device or traffic anomalies [4, 5].

IDS systems were developed to identify attacks and to avoid attacks if possible. Using the effectiveness of machine learning and artificial intelligence strategies, IDS becomes smarter and gets more accurate in dealing with its decision that comes decision tree techniques, fuzzy logic techniques, support vector machines, or neural networks are applied [6, 7].

In deployment scenarios at the edge of the network, IDS systems face new challenges including fast response times and the high dimension of data sizes. That is also the biggest obstacle of traditional ML algorithms and needs to be carefully considered nowadays. Therefore, the approach to finding new algorithms is very urgent in the goal of IoT network protection. With certain advantages of CMAC which are fast calculation and easy deployment on hardware [8], the proposed IDS based on CMAC is a deal to tailor with edge computing scenarios. Moreover, to reduce the computational complexity when the input data is a high dimension, feature selection techniques are selected as a preprocessing step for the intelligent algorithms of the IDS system. This solution eliminates redundancy and improves system IDS performance [9]. Different from the previous approaches, a solution proposed in this study to reduce the number of traits for decision trees according to the Gini index is proposed. Experimental results on the UWNB data set show that the proposed solution gives more accurate results with some properties less than the previous solutions.

3 Premier

a. **The feature selection approach**
The purpose of feature selection is to find a subset of attributes from the original set sufficient to represent the data. Among wrapper, filter, and embedded approaches we use the filter approach to get efficient data based on the statistics expressed by the Gini index [10]. Gini index is used to determine which feature/attribute gives us the maximum information about a class. For the dataset X which contains n class, the *Gini* index is determined by the formula [10],

$$i(X) = 1 - \sum_{i=1}^{n} p_i^2, \tag{1}$$

where p_i- the probability of an object being classified to a particular class.

After splitting of dataset X with A selection features into two subsets X_1 and X_2 with some records respectively N_1 and N_2. The *Gini* index is determined by the formula,

$$i_A(X) = \frac{N_1}{N} i(X_1) + \frac{N_2}{N} i(X_2), \tag{2}$$

where $i(X_1) = 1 - \sum_{i=1}^{n} p_i^2(X_1)$, $p_i(X_1)$ *is* the probability of an object being classified to a particular class in the dataset X_1, $i(X_2) = 1 - \sum_{i=1}^{n} p_i^2(X_2)$, $p_i(X_2)$ is the probability of an object being classified to a particular class in the dataset X_2.

The feature is considered the best if $\Delta i(A) = i(X) - i_A(X)$ has reached the maximum value.

In this study, the feature is selected by using the random forest algorithm over the Gini impurity index, which attributes with the highest Gini impurity will be the most important value.

b. **The CMAC proprieties**
The structure of CMAC was originally composed of two inter-layers (a, p) mappings illustrated in Fig. 1. The mathematical formulation to express the relationship between input vectors and output vector of CMAC model and multilayer perceptron (MLP) model is the same. However, CMAC has several interesting different features as below.

- The input vectors are only accepted integer values;
- Only output for multiple inputs;
- The parameter p is the key parameter that determines network performance through memory capacity and convergence time.

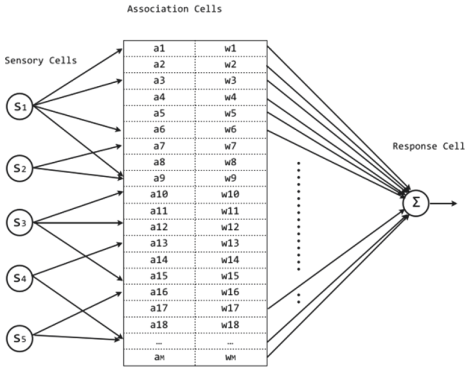

Fig. 1. The CMAC structure

4 The Proposed Model and Experimental Validations

a. **The proposed model**
To build and validate our proposal, the UNSW-NB dataset 15 is used in this study [11]. The main characteristics of the dataset include 44 attributes for each attack type, 2,540,044 records stored in four CSV files. The percentages of the number of attack types are shown in the second column of Table 1.

Table 1. Components of attack types in the UNSW-NB15 dataset

Attack types	Num of records
Normal	1959775
Reconnaissance	13357
Backdoor	1983
DoS	5665
Exploits	27599
Analysis	2184
Fuzzers	21795
Worms	171
Shellcode	1511
Generic	25378

The input vectors of the CMAC neural network only accept integer values so they must first be quantized. For quantization, it is necessary to determine the maximum and minimum values of each attribute. These values are described in Table 2.

The maximum value in order to quantize for each feature applied to CMAC neural network is: 17, 257, 257, 1025, 9, 65. The set of values of 6 input vector features is:

$X = \{x^{(1)} = \overline{1, 17}; x^{(2)} = \overline{1, 257}; x^{(3)} = \overline{1, 257}; x^{(4)} = \overline{1, 1025}; x^{(5)} = \overline{1, 9}; x^{(6)} = \overline{1, 65}\}$.

The values of the 6 input vector features are quantized according to formula (1).

The learning process of the CMAC neural network depends on the value of the general parameter p, only receiving the $p = 2, 4, 6, 8, 16, 32$. Furthermore, the accuracy depends on the threshold ϑ. The number of training steps is 10 000 000.

Table 2. The minimum and maximum value of each feature

№	Feature	Minimum value	Maximum value
1	Service	21	33
2	Sttl	0	255
3	Dttl	0	252
4	Smeansz	24	1504
5	Ct_state_ttl	0	6
6	Ct_srv_dst	1	62

The training process will be done when the input vector is quantized, the number of DoS attack records is 4412, the non-DoS attack is 126485 (accounting for 80% of the number of DoS attack records and the DoS attack of UNSW-NB dataset 15).

When the training process is done, the testing process will be carried out. The data of the testing process are also quantized like the training process. The number of DoS attacks and non-DoS attack records that will be tested is 1103 and 31616 respectively (accounting for 20% of the DoS attack records and non-DoS attack records of the UNSW-NB 15 dataset). Each record will be labeled:

- 1 - when the record is a DoS attack,
- 0 - when the record is a non-DoS attack.

The process of testing the CMAC neural network is performed with different threshold values from 0.1 to 0.9 with a jump of 0.01. When comparing the obtained results, the number of identified attacks reached the highest result when the threshold value is 0.57, the percentages of the identities for a DoS attack and a non-DoS attack are 86.13% and 85.13% respectively.

b. **The validation**

The two networks used for comparison are MLP and Support Vector Machines (SVM). SVM is another kind of machine learning technique. Based on the principles of linear classification, SVM creates a hyper-plane to maximize the distance between two layers.

The training process of the MLP network and SVM were implemented in a MATLAB environment, by using the application package Neural Network Toolbox. The

learning process of the CMAC neural network is done in Visual Studio 2013 and/or the programming language is C++.

During the training process, the SVM network uses 2 functions: Gaussian Radial Basis Function (RBF) and polynomial to select the best results. The experimental results are listed in Table 3.

Table 3. Test results

Type of neural network	Parameters				
	Methodology	The number of layers and neurons in each layer	Threshold value	Identification rate of DoS attack, %	Identification rate of non-DoS attack, %
NN CMAC	NN CMAC		0.566	86.49	85.1
MLP	Trainlm	30–20-10–1	0.72	85.31	84.71
SVM	Rbf	-	-	56.3	89.45

c. **The proposed feature selection to apply CMAC neural network**
The selection of features by using the random forest algorithm to determine the Gini impurity index [10], in which the feature with the highest Gini impurity will be the most important. By applying the above method, the results obtained 9 features with the highest Gini impurity index are Proto, Service, Sttl, Dttl, Synack, Smeansz, Ct_srv_src, Ct_state_ttl, Ct_srv_dst.

The records from the UNSW-NB 15 data set with the 9 selected above features were added to the MLP network to compare the results with the MLP network by using 42 features. If the result by using 9 features is higher or equal to the result by using 42 attributes, it will reduce 1 feature. The reduction process of a feature will be finished when the obtained result by applying the number of feature drops is lower than the result by using 42 features.

The training process MLP network was performed on 4.412 DoS attack records and 12.6485 non-DoS attack records (accounting for 80% of the number of records of the UNSW-NB15 data set). The input MLP will use 42 features of all types of attacks and the used algorithms are trainlm, traingdx, trainscg, trainbfg. During the learning and testing process we used 3 layers (15–10-1, 30–20-1, 50–30-1, 100–50-1, 100–100-1, 150–100-1, 200- 100–1, 200–150-1) and 4 layers (30–20-10–1). The threshold for classification will run from 0.1 to 0.9 with a jump of 0.01. The testing process was performed on 1.103 DoS attack records and 31.616 non-DoS attack records (accounting for 20% of records from the UNSW-NB 15 dataset). After testing all the cases, the highest classification accuracy for DoS and non-DoS attacks are 85.31% and 85.71% respectively. The proposed feature selection algorithm is illustrated below.

The proposed feature selection algorithm

Begin

1. $i := [1,...9]$, (*i:* the number of features in the attribute set *F*)
2. $j := [1,...c_9^i]$, (*j*: numerical order in *i*)
3. $F = \{F_{ij}\}$, (F: the attribute set)
4. For $i = 9 \rightarrow 1$ do
5. Set threshold [0.1 – 0.9] and step parameter [0.01]
6. Compare $B_{ij} <> A_{42}$ (B_{ij} : experimental value, A_{42} : original MLP value)
7. If $B_{ij} < A_{42}$, go to the end
8. If $B_{ij} \geq A_{42}$, go to next step
9. Select a feature set with the highest accuracy
10. $i := i - 1$, return to step 4.

End

The above results show that, when using the input vector with 6 features, the result is no worse than when using 42 features. Therefore, 6 features: Service, Sttl, Dttl, Smeansz, Ct_state_ttl, Ct_srv_dst were selected to include in the CMAC neural network used to identify DoS attack on UNSW NB 15 dataset.

5 Conclusion

The aim of improving IDS system performance for edge computations is an interesting problem in currently researched problem because of its topicality. The research results show that IDS based CMAC is an effective tool to detect attacks. To overcome the limitation of the CMAC as the input of the network is limited by the number of features, we proposed a novel method based on the combination of the MLP network and the Random forest for reducing input features that also degraded its complexity. Experimental results show that when reducing the number of features from 42 to 6 for DoS attack on dataset UNSW NB 15, the NN CMAC gave higher results than MLP and SVM networks. In the next research, we will present the NN CMAC test on other real attack dataset and will propose a multi-expert system in which the component is neural networks.

References

1. Letaief, K.B., Chen, W., Shi, Y., Zhang, J., Zhang, Y.-J.A.: The roadmap to 6G AI-empowered wireless networks. IEEE Commun. Mag. **57**, 84–90 (2019)
2. Fang, H., Qi, A., Wang, X.: Fast authentication and progressive authorization in large-scale IoT: how to leverage AI for security enhancement. IEEE Netw. **34**(3), 24–29 (2020)
3. Cao, K., Liu, Y., Meng, G., Sun, Q.: An overview on edge computing research. IEEE Access **8**, 85714–85728 (2020)

4. Mudgerikar, A., Sharma, P., Bertino, E.: Edge-based intrusion detection for IoT devices. ACM Trans. Manage. Inf. Syst. **11**(4), 21 (2020). Article 18 https://doi.org/10.1145/3382159
5. Almogren, A.S.: Intrusion detection in edge-of-things computing. J. Parallel Distrib. Comput. (2019). https://doi.org/10.1016/j.jpdc.2019.12.008
6. Nguyen, V.-T., Nguyen, T.-X., Hoang, T.-M., Vu, N.-L.: A new anomaly traffic detection based on fuzzy logic approach in wireless sensor networks. In: Proceedings of the Tenth International Symposium on Information and Communication Technology (SoICT 2019), pp. 205–209. Association for Computing Machinery, New York (2019). https://doi.org/10.1145/3368926.3369714
7. Zhang, H., Wu, C.Q., Gao, S., Wang, Z., Xu, Y., Liu, Y.: An effective deep learning-based scheme for network intrusion detection. In: 24th International Conference on Pattern Recognition (ICPR), pp. 682–687. IEEE (2018)
8. Xing, F.: A Historical Review of Forty Years of Research on CMAC. In ArXiv, abs/1702.02277 (2017)
9. Moustafa, N., Slay, J.: A hybrid feature selection for network intrusion detection systems: Central points. In arXiv preprint. arXiv:1707.05505 (2017)
10. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001)
11. https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/. Accessed 18 Oct 2020.