Representing the Abnormal Web Access for Advertisement Counting Based on the Rule-Extraction Mechanism of Neural Networks

Jen-Cheng Chen[1], Maiga Chang[2], Ko-Kang Chu[3]

[1] Dept. of Electronic Engineering, Chung Yuan Christian University,
32023 Chung-Li, Taiwan
clement1972@gmail.com
[2] National Science & Technology Program for e-Learning in Taiwan, Nat'l Central Univ.,
32099 Chung-Li, Taiwan
maiga@ms2.hinet.net
http://maiga.dnsalias.org/maiga/resume_e.htm
[3] Dept. of Electronic Engineering, Chung Yuan Christian University,
32023 Chung-Li, Taiwan
kirk@ms2.hinet.net

Abstract. This paper is a continuous research based on the research results in the past two years. At first, an Accuracy Counter was developed for the Internet advertiser to assure the reasonableness and fairness of Information stored in publisher-site databases on WWW. After a year running, the neural network with the hybrid training mechanism replaced the rule-based filter for detecting the cheating model of abnormal access behaviors which are performed by users in the conflicting environment suchlike the Internet Advertisement Market. However, the results which produced by neural networks are usually not readable to hu-man administrators. This paper tries to develop a rule-extraction mechanism for neural networks in order to represent a more human readability rule set for administrators. With the rule-extraction mechanism, the administrator will be able to discover the cheating patterns of either the advertisers or the publishers. Moreover, this mechanism should also be a useful tool for profiling the intruder behaviors in the network security domain.

1 Introduction

Web counters are usually used for recording the number of times a webpage has been visited. The value of a web counter could represent the importance of the page on which the counter is located. There are many companies that provide counter services. If we use the counter service of such a company, we can put a counter on our home-page, and the value of the counter will be stored in the database of the company. When the value of a counter is explicitly shown on a webpage, we call it a "visible type" counter. An "invisible type" counter, on the other hand, is one that does not have its value shown in the webpage on which the counter is located.

In the past, researches found that CPC (Cost Per Click) and CTR (Click Through Rates) are both fair and the easiest Internet advertisement pricing model [10]. Since then, whenever an advertiser rents a place to put its advertisement, the publisher would install an "invisible type" counter for each ad. and charge the renter according to the counts. As the name suggests, such counters are invisible, and users do not know their values. Since advertisers could not have idea about how many visitors had been browsed their advertisement already and the publishers who need higher counter value to charge higher price, there would be always some cheaters exist in such kind of conflicting environment. Therefore, an Accuracy Counter (AC) was developed to prevent the reasonableness and fairness of the counter's value [3].

After one-year experiment, neural network was used to replace the rule-based detection module [4]. Neural networks are usually using for detecting the possible malicious behaviors in network administration fields [7]. The major characteristic of neural networks is trainable, and the rules of the trained neural networks are stored by the weights [5]. Since the weights are numerical format, the weights can not be applied to a traditional rule-based expert system directly. Therefore, it needs a method of rule extraction to extract rules from neural networks [2][11]. Then, the extracted rules can be used as the knowledge base of expert system.

In Section 3, each neuron is modeled in order to construct boolean rules. Section 4 develops the algorithms for extracting rules from a neural network with the Bound Decomposition Tree (BDT). A neural network is constructed as the experiment and the rules obtained by the rule extraction algorithm are discussed in Section 5. Section 6 gives a simple conclusion and possible future works.

2 Accuracy Counter and Rule-Extraction

In general, a user will tend not to repeatedly access the same webpage within a short time interval. Therefore, when the same computer accesses the same webpage again in a very short time, it is feasible to consider the connection attempt as "unreasonable". The Accuracy Counter (AC) system is designed to block increments of the counter's value under such circumstances. Detail analysis of AC could be found in [3]. Fig. 1 demonstrates the architecture of the AC.

The original AC system took a rule-based detection module in the stage 4 of Fig. 1, however, the rule-based system for administrators who want to detect the abnormal access behaviors is not enough. Therefore, a neural-network-based hybrid training mechanism was proposed to replace the original rule-based detection module [4]. Although the neural-network-based AC system performed well, it still can not make the administrator feels comfortable due to the 'black box' effects of the neural net-works.

Some methods of rule extraction from a neural network had been presented and classified to two categories, the decompositional and the pedagogical algorithms [11]. The pedagogical rule extraction algorithm aims to extract rules that map inputs directly into outputs. Some typical algorithms of this category are

"VIA", "rule-extraction-as-learning" and "RULENEG".[7] Most of pedagogical algorithms are not effective when the size of the neural network increases, such as any real world problem. In order to improve the efficiency, decompositional algorithms focus on heuristically searching and extracting rules in neurons of neural networks individually. The existing decompositional algorithms include "KT"[6], "Subset", "MofN"[12], "RULEX"[1] , "Setiono"[9], etc.

This paper applies the analysis of hypercubes for the hyperplane of a neuron to propose an algorithm (Bound Decomposition Tree, BDT) to extract its positive, negative and uncertain Boolean rules, thus find the overall rules of a neural network. From the experiment [2], it showed that the BDT is quite simple and efficient. After rule extracting, the simpler rules will be got from neural networks and they will be stored in the knowledge base. The administrator could either understand or retrain the neural network according these human readable rules.

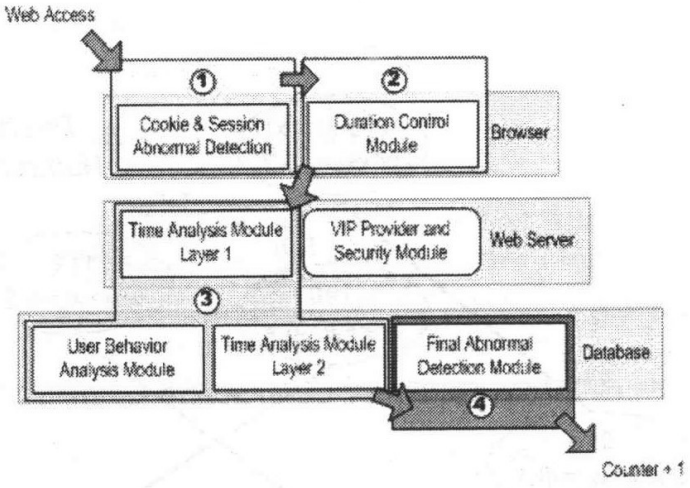


Fig. 1. The AC system is a three-tier secured counter.

3 Rule-Extraction Mechanism

A typical neuron used in this paper is defined as

$$y = f(\mathbf{w}^T\mathbf{x} + w_0) = f(\sum_{j \in N} w_j x_j + w_0) \tag{1}$$

in which N is the number of neuron inputs, $\mathbf{w} = [w_1, w_2, ..., w_n]^T$ is the weight vector of connection weights, w_0 is the bias and $\mathbf{x} = [x_1, x_2, ..., x_n]^T$ is the inputs of the neuron. The threshold function $f : R[0, 1]$ is monotonically increasing and may be linear, piecewise linear, hard-limited and sigmoid, as defined in a conventional neural net-work textbook. The weight vector and its threshold can be collected as a set of augmented weights $\mathbf{w}^* = [\mathbf{w}, w_0]^T$, corresponding to the

augmented input $\mathbf{x}^* = [\mathbf{x}, 1]^T$. When the neuron inputs are binary, $\mathbf{x} = [x_j] \in \Xi = \{0,1\}^N$, it is called binary neuron.

In essence, the outputs of a binary neuron are linearly separable and can divide the N-dimensional space (R^N) into two halfspaces as Fig.2 shows. Therefore, the input states can also be classified into positive state set $\Xi_{p,\mathbf{w}^*}$ and negative state set $\Xi_{n,\mathbf{w}^*}$.
Take the input states in Fig. 2 as example :

$$\Xi_{p,\mathbf{w}^*} = \{110, 111, 100\} = \{11-, 1-0\} \tag{2}$$

and

$$\Xi_{n,\mathbf{w}^*} = \{010, 000, 011, 001, 101\} = \{0--, -10\} \tag{3}$$

in which '-' is called don't care state, meaning this state can be either '0' or '1' from the viewpoint of boolean logic. For example, 1-0 corresponds to $x_1\overline{x_3}$ and {11-, 1-0} corresponds to $x_1x_2 + x_1\overline{x_3}$.

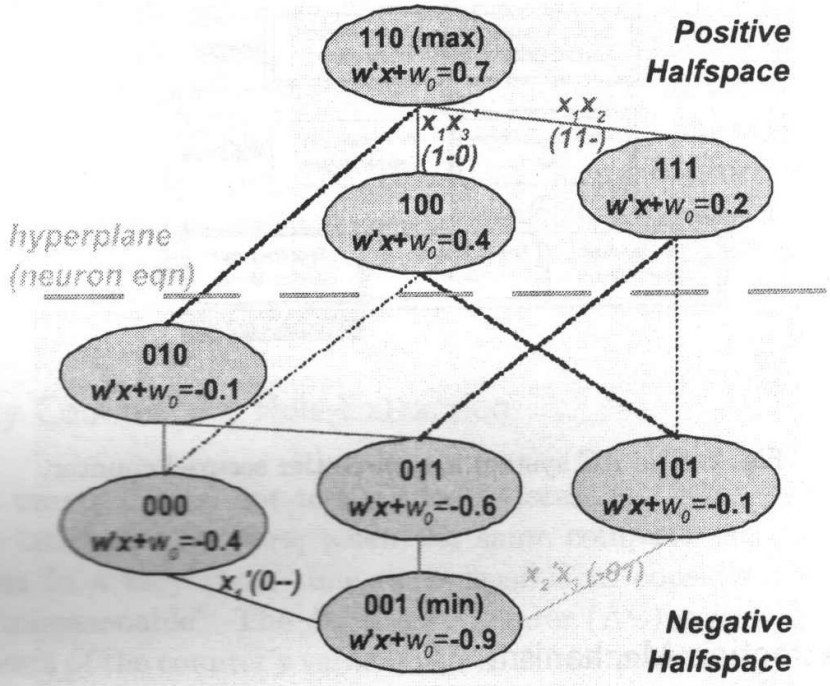


Fig. 2. The outputs of a binary neuron are linearly separable. According to the output, different input states can be divided to two halfspaces.

4 Rule Extraction Algorithm

For the example in Fig. 2, the initial bounds between the negative halfspace and the positive halfspace are -1.05 (001) and 0.75 (110). The initial bounds can be indicated as $bounds(\Psi = \text{'- - -'}) = [-1.05, 0.75]$ and $boundvar(\Psi = \text{'- - -'}) = 1.8$.

Algorithm (Boolean Rules for a Single Neuron)

1. Given a set of weights and its threshold $\mathbf{w}^* = [\mathbf{w}, w_0]$ in Eq.(1) for a single neuron, and a rule margin ?.
2. Sort these weights $\mathbf{w} = [w_j]$ by their absolute values to obtain an index sequence $IS = \{j = argmax_{j \in N} |w_j|\}$.
3. Set layer number $N_L = 0$, $\Psi = \Xi_{\mathbf{w}^*}$, $bound(\Psi)$, and $boundvar(\Psi)$.
4. Let $N_L = N_L + 1$, j = the first index j in IS, and IS = IS {j}.
5. decompose each uncertain hypercube, $\Psi = positivecube(\ldots) \cup negativecube(\ldots)$, in the deepest layer into positive hypercube and negative hypercube.
6. For each decomposed hypercubes, calculate the $bounds = [lbound(\ldots), ubound(\ldots)]$.
7. $boundvar(\ldots) = boundvar(\Psi) - |w_j|$.
8. If all new hypercubes are positive or negative, or the sum of the rest weights less than ?, then stop; otherwise go to step 4.
9. To transform all the calculated hypercubes into the positive rule $R_{p,\mathbf{w}^*}$, the negative rule $R_{n,\mathbf{w}^*}$, and the uncertain rule $R_{u,\mathbf{w}^*}$.

5 Experiment and Results

To better understand the effects of the rule extraction that we proposed in this paper, a neural network is constructed first. The training data come from the database of Accuracy Counter during March 2004 to June 2004, and there are total 78,072 records. The structure of the neural network is a 2-layers BPN (5x1) with random initial weights. The 2-layers structure includes one input layer and one output layer.

In the experiment, there are five input elements (N=5), they are

x1 : Does the visitor stays over 30 seconds? (0 for No, 1 for Yes)

x2 : Is there a Virtual IP (VIP) which stored in the cookie? (0 for No, 1 for Yes)

x3 : Is there any Real IP (RIP) belongs to the same class B network with the visitor's RIP already? (0 for No, 1 for Yes)

x4 : Are the visits at regular intervals time difference? Which means that is the time difference between $(t_n - t_{n-1})$ and $(t_{n-1} - t_{n-2})$ larger than 10 seconds? (0 for No, 1 for Yes)

x5 : Does the visitor comes to visit the webpage after 30 minutes from his/her last visit? (0 for No, 1 for Yes)

After training, the weights of the 2-layers BPN are

$[w_1, w_2, w_3, w_4, w_5, w_0] = [72.50, -2.81, -2.80, 72.48, 72.47, -182.61]$.

The rules extracted from the well-trained 2-layers BPN are

$R = (x_1' + x_2' x_3 x_5' + x_4' + x_5')' = (x_1' + x_4' + x_5')' = x_1 x_4 x_5$

= The visit would be reasonable if the same visitor who comes to the web-page over 30 minutes from his/her last visit (x_1) **AND** stays over than 30 seconds (x_4) **AND** his/her visits are not at regular intervals (x_5).

The above experiment result shows that the neural network can effectively learn the model of AC. And then BDT can present the trained model with few

logic rule. There-fore, the web administrator can understand the mean of trained AC model, and can confirm if the rules conform to the experience.

The final architecture of the neural network-based abnormal access behavior detection module with the rule-extraction mechanism is shown in Fig. 3.

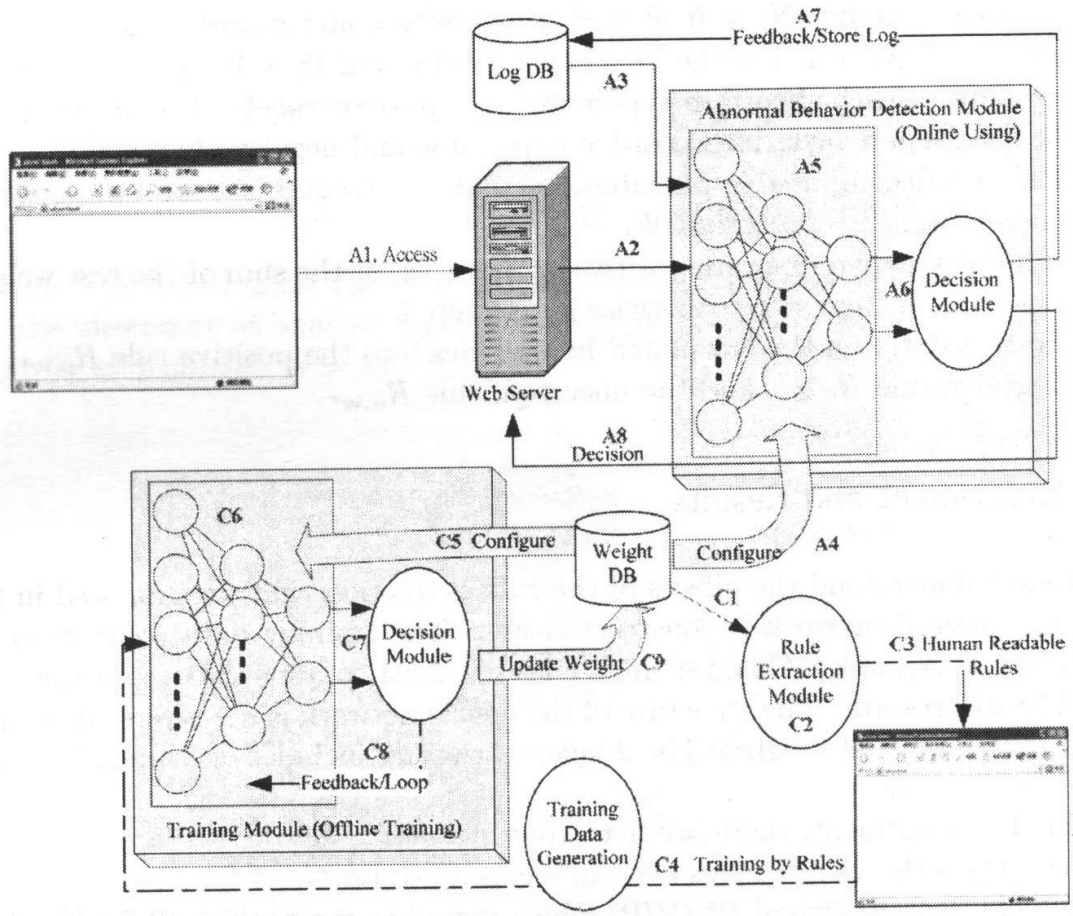


Fig. 3. Using the results of the rule extraction mechanism to represent and retrain the neural network on the Internet environment for system administrator

6 Conclusions

Based on the analysis of the hyperplane, this paper proposed an algorithm to extract the positive and negative, even uncertain rules of a neuron at the same time. After the rules are extracted from a well-trained neural network, the administrator then could get more clear understanding about the cheater. Moreover, the administrator can input the rules that he/she thinks important and make the neural network retraining. Although the proposed algorithm could extract rules from a neural network with lower complexity, the system still needs to be enhanced. For example, the extracted rules are Boolean logic format; hence, the input values can not be continuous.

References

1. Andrews, R., Geva, S., Rule extraction from local cluster neural nets, Neurocomputing 47 (2002) 1-20
2. Chen, J.C., Heh, J.S., Chang, M., Designing a Decompositional Rule Extraction Algorithm for Neural Networks, In 3th International Symposium on Neural Networks (ISNN 2006), Chengdu, China, May 28-31 (2006)
3. Chu, K.-K., Chang, M., Hsia, Y.-T., Designing a Mechanism to Assure the Reasonableness and Fairness of Information Stored in Databases on WWW, In the Proceedings of the IEEE International Conference on Systems, Man and Cybernetics 2003 (IEEE SMC 2003), Wash-ington DC, USA, Oct. 5-8, 2003 Vol. 3 (2003) 2302-2307
4. Chu, K.-K., Chang, M., Hsia, Y.-T., A Hybrid Training Mechanism for Applying Neural Networks to Web-based Applications, In the Proceedings of the IEEE International Confer-ence on Systems, Man and Cybernetics 2004 (IEEE SMC 2004), Hague, Netherlands, Octo-ber 10-13 (2004) 3543-3547
5. d'Avila Garcez, A.S., Broda, K., Gabbay, D.M., Symbolic knowledge extraction from trained neural networks: A sound approach, Artificial Intelligence 125 (2001) 155-207
6. Fu, L.-M., Rule generation from neural networks, IEEE Transactions on System, Man, and Cybernetics 28 (1994) 1114-1124
7. Planquart, J.-P., Application of Neural Networks to Intrusion Detection, SANS Institute (2001). Retrieved from http://www.sans.org/rr/papers/30/336.pdf
8. Pop, E., Hayward, R., Diedrich, A., RULENEG: extracting rules from a trained neural net-work using stepwise negation, Technical Report, Neurocomputing Research Centre, Queen-sland University of Technology (1994)
9. Setiono, R., Extracting rules from neural networks by pruning and hidden-unit splitting, Neural Computation 9 (1997) 205-225
10. Sweeney, T., Click-Through Rates on online ads are declining, but true believers say there's no place like the Web to build the brand, InternetWeek, Manhasset 832 (2000) PG.57
11. Tickle, A.B., Andrews, R., Golea, M., Diederich, J., The truth will come to light: directions and challenges in extracting the knowledge embedded within trained artificial neural net-works, IEEE Transactions on Neural Networks 9 (1998) 1057-1068
12. Towell, G. G., Shavlik, J.W., Knowledge-based artificial neural networks, Artificial Intelli-gence 70 (1994) 119-165