Algorithm Specifications

Authors: Devi Parikh, Liu Li-Hao, Tsuhan Chen

Department of Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA, USA

Date: 09/15/2005

Table of Contents:

1. Introduction

Pattern recognition techniques will be used to distinguish between normal and abnormal traffic patterns. If abnormal, the specific type of attack will also be identified. Neural networks will be the building blocks of the classification system.

The two key points that distinguish the proposed algorithm from its counterparts are its Data Fusion capabilities and its Adaptability.

1.1. Data Fusion:

Conventionally it is known that the information about a given traffic pattern being normal or malicious can be obtained from multiple sources such as the traffic statistics, the content information and other basic information about the connection. The proposed algorithm will take advantage of this fact, and attempt to build a classification system that is capable of combining information from multiple sources to make a more informed decision.

1.2. Adaptability:

In real scenarios, the traffic patterns change over time and a classification system that is capable of adapting to these changes is desirable. The proposed algorithm will also attempt to incorporate this adaptability in the classification system. So if a new type of attack becomes known, the system would detect it as a previously unseen pattern, and with some involvement from an external expert, would be able to update its functioning so as to be able to detect this attack type in the future.

2. Algorithm

2.1. Overview

The entire classification system would be divided into several sub-systems (one sub-system per class). For example, if the classification system is initially trained to recognize k different classes (1 normal class, and k-1 different types of attacks), the classification system would initially have k sub-systems. Each of these k sub-systems has just one response – positive or negative for the class that it represents.

Each of these sub-systems would have different subordinate systems (one sub-ordinate system per available source of information). So if there are f different sources of information (SI) available that provide relevant information that can help identify an attack and distinguish between the different types of attacks, each of the k sub-systems would have f subordinate systems, one corresponding to each of the f sources of information. All f subordinate systems in one of the k sub-systems try and identify the same class. However, they use different sources of information to perform this task. This is where the classification system has its ability to perform data fusion. Hence there are a total of k*f subordinate systems.

Each of these subordinate systems would be built out of n different artificial neural networks (NN). These n different neural networks in each of the k*f subordinate systems would do an identical task, and use the same source of information. However, they would have slightly different characteristics. Research shows that combining slightly different classifiers provides better classification than having just one strong classifier.

Hence, the entire classification system would comprise of k*f*n artificial neural networks.

The layout of the entire classification system is shown in Figure 1. For demonstration purposes, k is taken as 3 (3 different initial classes – 1 normal class, 2 different types of attacks: attack1 and attack2), f is taken as 3 (3 different sources of information SI) and n is taken as 2 (2 neural networks NN per class per source of information). Detailed specification about the algorithm and classification system design is provided next.

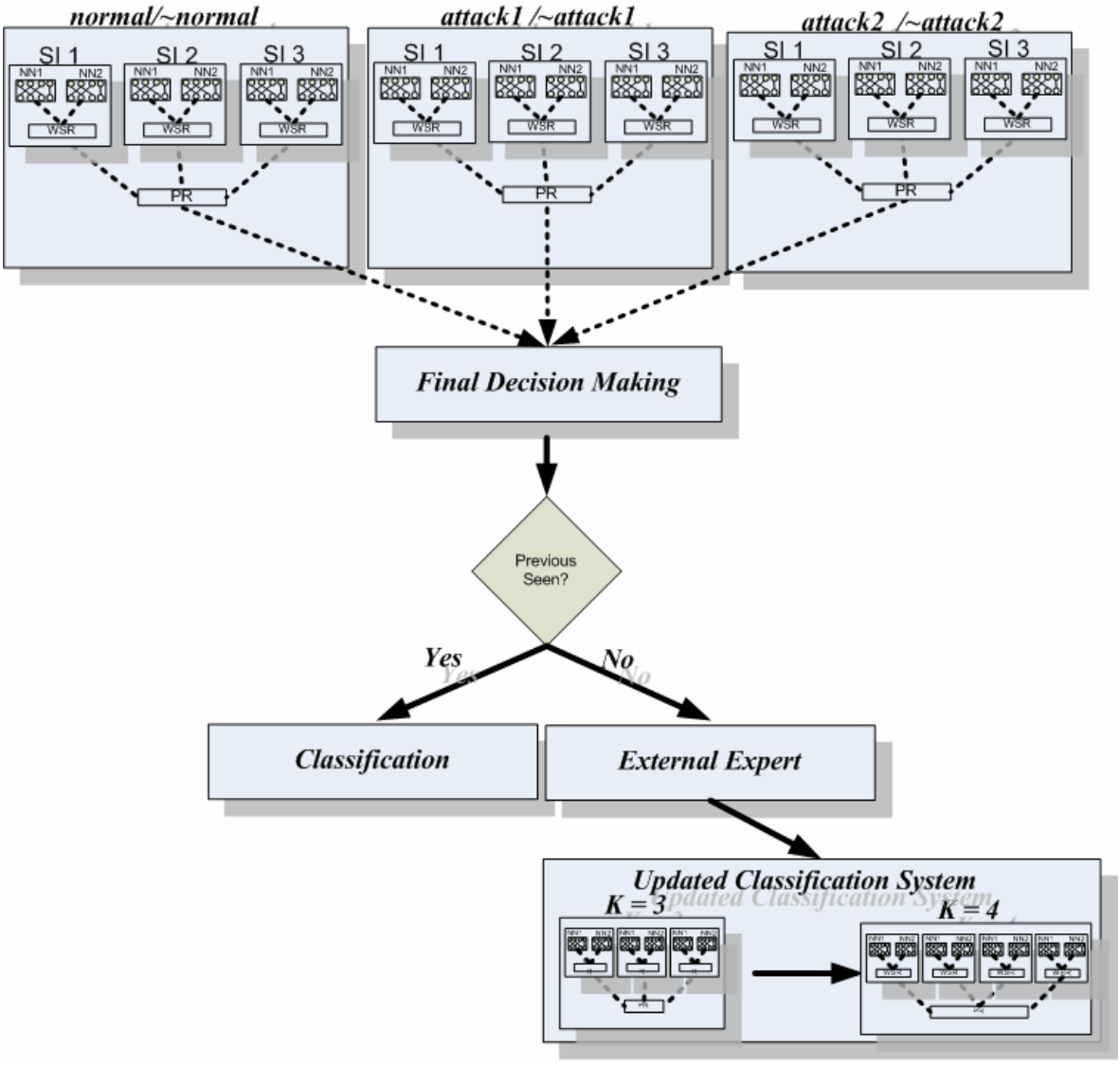

Figure 1: Layout of Classification System

There are two different phases involved in building the classification system

2.2. Training and Validation (with Data Fusion)

The entire available data would be divided into training, validation and testing. The first k*f (out of the total n*k*f) neural networks will be trained on the training data. They will be validated on the validation data and their performances will be recorded. The neural networks are accepted only if their performance on the validation data is above 50%. If not, the neural network is rejected and a new neural network is generated. A weight will be assigned to each of the accepted neural networks proportional to their performance on the validation data. The second k*f set of neural networks will be generated on a slightly different subset of the training data. This is what introduces the difference in the characteristics of the neural networks as indicated earlier. Their performance will be evaluated on the validation data. Only if the combined performance of these neural networks with their respective preceding neural networks (generated above) is better than the performance of the first set of neural networks individually, and if the performance of the second set of neural networks on the validation data is above 50%, the second network is stored. Else, it is replaced with a new network. Once the second set of neural networks are generated and accepted, they are assigned weights by the same scheme. This is repeated till all n sets of k*f neural networks are generated. This completes the training (and validation) phase.

The algorithm for the training and validation phase is summarized below:

```
Training and Validation
Input: Data split into Training and Validation
Output: Trained classification system

For each class (initially three classes)
        For each feature set ( basic, traffic, content)
                Loop1: Train 1st  NN of 2/3 of training data (random selected)
                        Validation performance = V1
                        While V1<0.5
                                Goto Loop1
                        End
                        Calculate weight
                Loop2: Train 2nd  NN of 2/3 of training data (random selected)
                        Validation performance = V2
                        While V2<0.5
                                Goto Loop2
                        End
                        Calculate weight
                        Validation of two networks combined = V12
                        While V12<V1
                                Goto Loop2
                        End
        End
End
```

Algorithm for Training and Validation

2.3. Testing (with Adaptability)

When a test instance is presented to the classification system, all neural networks evaluate their outputs for that instance (note: the neural networks would use only those feature sets of the provided instance that they were trained on). The n different neural networks in each of the k*f subordinate systems would be combined using the weighted sum rule (WSR). Having done this, the outputs of the f different subordinate systems in each of the k sub-systems would be combined using the product rule (PR). So now we have the k different outputs (all positive or negative for each of the k classes).

If exactly one of the k outputs is positive, that class is picked as the final decision. If more than one output are positive, the class with the highest confidence is picked. If none of the outputs are positive, it means that the provided instance is of a previously unseen class. An expert is alerted and is required to provide a label to this class. Once this is done, the system has k+1 classes to distinguish among, and this is done by adding a (k+1)st sub-system with n*f neural networks in a fashion similar to that explained above. This is where the system shows is capabilities to adapt.

The algorithm for the testing phase is summarized below:

Testing
Input: Testing dataset
Output: Classification decision (and possibly an updated classification system)

For each instance
    Combine both neural networks within 9 groups (3x3) – Weighted Sum Rule
    Combine 3 decisions (for each class) – Product Rule
    If one of 3 decisions is positive
        Pick corresponding class as decision
    Elseif more than one are positive
        Pick class with highest confidence
    Elseif all 3 decisions are negative
        Raise an alert
        Expert involvement to provide label
        Update system by adding a 4^{th} group (with 2x3 neural networks)
    End
End

Algorithm for Testing

3. Data

Initially, the DARPA database collected by MIT in 1998 will be used to simulate the above specified scenario and establish the effectiveness of the proposed algorithm. There are four different types of attacks in the database (total five classes). Initially only three of these classes may be used (k = 3) and the other two classes may be added during the testing phase to establish the capability of the algorithm to adapt to changing traffic scenarios. Three different sources of information are present in the database (f = 3). For each of the k*f subordinate-systems, 2 neural

networks (n = 2) may be generated. Multilayer perceptron neural networks will be used initially. These parameters and choices may vary during implementation.

4. Summary

To summarize, the novelty of the proposed algorithm lies in its capabilities to fuse information from multiple sources and also adapt itself to changing environments. It uses a multiple classifiers approach to incorporate these capabilities.