Training a multi-criteria decision system and application to the detection of PHP webshells

Alexandre Croix Thibault Debatty Wim Mees

Cyber Defence Lab, Royal Military Academy, Brussels, Belgium

Introduction

The growth of the amount of information and usage of multi-criteria decision systems makes, data-fusion techniques, and aggregation functions especially more and more important. These techniques are used in a wide field of applications and are related to a difficult problem: how to determine parameters for a known function.

Several approaches have been developed to determine these parameters. Some of them used a large dataset of examples, and the parameters are deduced from these examples.

We worked on a method to determine parameters for the WOWA function. WOWA, for Weighted Ordered Weighted Averaging, is an aggregation operator that is a generalization of the Ordered Weighted Averaging (OWA) and the Weighted Mean (WM). Concretely, WOWA merges a set of numerical data in a single number thanks to two weighting vectors: one for the weighted mean (w) and the other for the OWA operator (p). The WOWA operator combines the advantages of both of them. The weighted mean, weights the information sources, and the OWA gives importance to the data according to their scores. The WOWA operator takes three vectors as arguments to produce a single numerical result. The expression could be written : $output = WOWA(w, p, data)$.

Learning

There are different methods to learn aggregation function parameters. One of these approaches is based on Genetic Algorithms (GA). A GA is an iterative procedure that maintains a population of n individuals ($P(t) = (x_1^t, ..., x_n^t)$ at iteration t). An element of the population is called "chromosome" and is a potential solution of the problem. Each *chromosome* is composed of characteristics named "genes". In this case, a *gene* is a single weight (value between 0 and 1) and a *chromosome* is composed of two weight vectors. The two vectors have m *genes* whose sum is equal to 1.

The structure of a Genetic Algorithm is quite simple and can be represented by the Figure **??**:

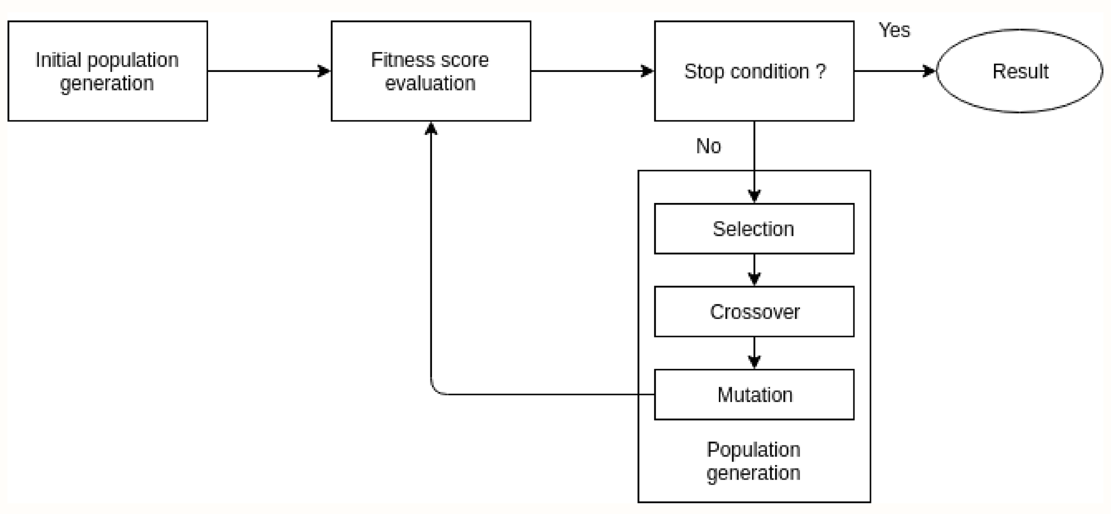

Figure 1: Structure of a Genetic Algorithm

The algorithm has five main steps:

1. **Initial population generation**: the population is initialized with elements that are potential solutions of the problem.
2. **Fitness score evaluation**: performance of all elements is evaluated.
3. **Selection**: according to their fitness score, some elements from the *generation t* are selected to be used in the *generation* $t + 1$.
4. **Crossover**: the selected elements are combined two-by-two to obtain a new complete population.
5. **Mutation**: some random elements are mutated.

The process is repeated until to reach a termination condition.

Initial population generation

There are different methods to generate an initial population. The most intuitive and used is a completely random generation. It is also possible to generate randomly a part of the population, and the last elements are specific ones.

Fitness score evaluation

Each population element is evaluated according to a specific criterion. In this work, two different criteria were implemented: *distance* and *AUC*.

- **Distance criterion**: For each chromosome in the population, the WOWA function is computed on all examples of the dataset. The obtained results are substracted to the results given in the training dataset. All these differences are added to obtain a total distance that is the fitness criterion of the chromosome. The smaller the distance is, better is the chromosome.
- **AUC criterion**: For each population element, the WOWA function is also computed on all examples of the dataset. Then, these results are used to obtain the Receiver Operating Characteristic (ROC). The ROC curve is a plot that illustrates the ability to classify correctly elements as its discrimination threshold varied. It is created by plotting the *true positive rate* (true detection) against the *false negative rate* (false alarm). The Area Under the Curve (AUC) is a measurement of the classification efficiency. Bigger is the AUC, better is the classification. This criterion is only usable for binary classification systems.

Selection

According to their fitness score, some elements from *generation t* are selected to be used in *generation* $t + 1$. We implemented two selection methods:

- **Roulette Wheel Selection**: This method gives to each element i of the population a probability $p(i)$ to be selected proportional to its fitness score.
- **Tournament Selection**: This method picks randomly two elements of the population and keeps the chromosome with the best fitness score for the next *generation*.

Crossover

The selected elements are combined two-by-two to generate new *children*. The *children* keeps characteristics from their *parents*. The Figure **??** is a slightly simplified representation of the implementation of this work.

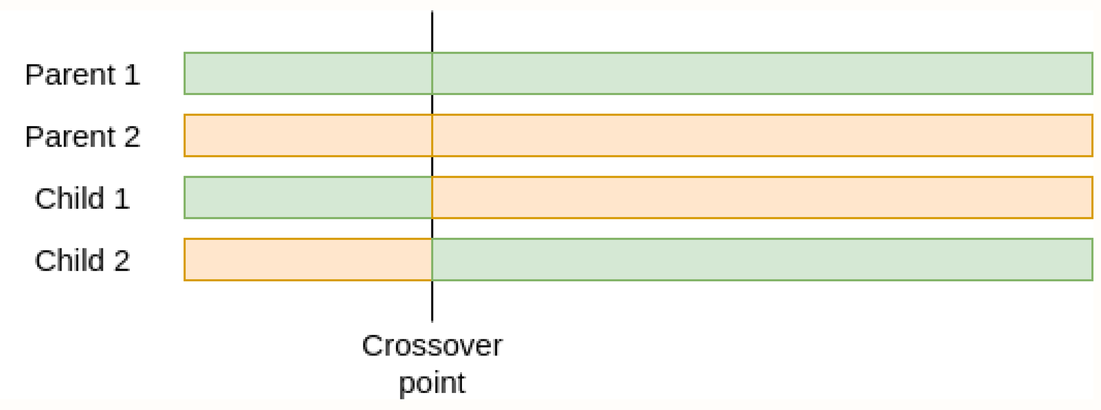

Figure 2: Simplified representation of the crossover implementation

Mutation

The mutation is very important to avoid converging too quickly on a local minimum. Without this mechanism, generation after generation, the population will converge to a minimum. It is very likely that this minimum is local and no global. Mutation allow to "jump" on another location in the space of solutions and can discover new and potentially better results. Concretely, a random *gene* is selected and replaced by a random value (between 0 and 1).

Parameters study and experimental evaluation

Test setup

To evaluate the efficiency of our training method, we perform a parameters study on data provided by a webshell detector. This detector analyses PHP files on five criteria and gives a score between 0 and 1 for each of them. The training dataset used for this work is composed of 12,468 PHP files that contains 206 real PHP webshells. A part of this dataset was used to learn the weights and the rest of the data were used to evaluate the efficiency of the model.

The algorithm has different parameters that can be changed to optimize results:

- **Population size**: number of elements in each generation of the population.
- **Crossover rate**: percentage of the population kept to generate the next generation.
- **Mutation rate**: percentage of genes which are mutated in each generation.
- **Generation number**: number of generation before stopping the algorithm.
- **Fitness score evaluation**: method used to determine the fitness of chromosomes (distance or AUC).
- **Initial population generation method**: method used to generate the initial population.

A complete parameter study was performed and the efficiency of a parameters combination was evaluated according to two measurements: (i) the value of the AUC and (ii) the range of threshold values that allows a correct classification for more than 95% of the files.

Results

The Figure **??** is a typical ROC curve obtained by a classification using the weights learned thanks to the Genetic Algorithm. This ROC curve has an AUC greater than 0.99.

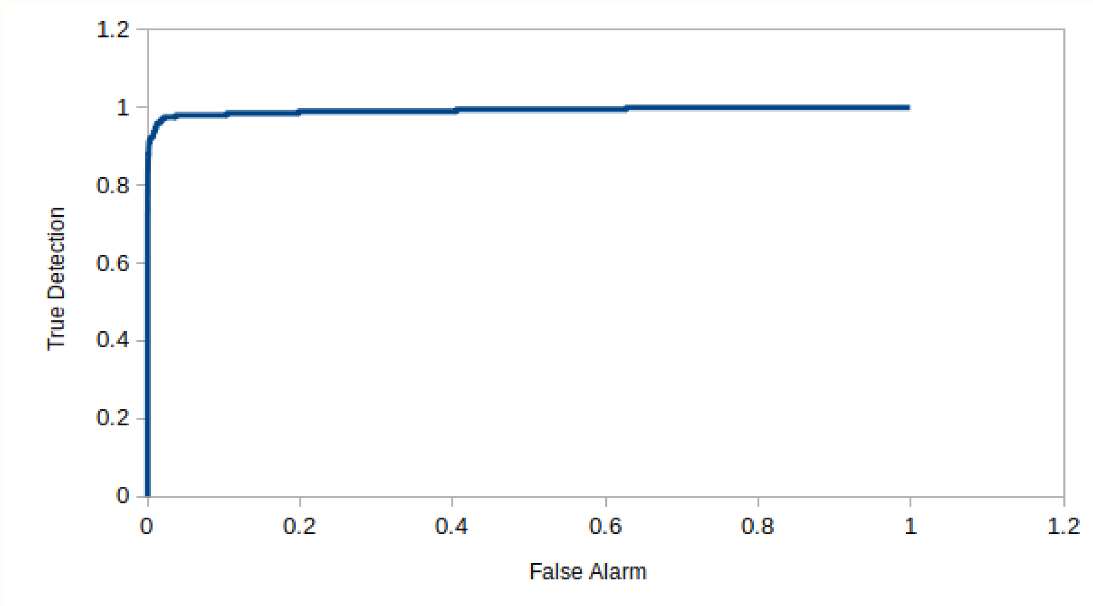

Figure 3: Typical ROC curve obtained thanks to the learning of the algorithm

This algorithm has an advantage on other learning methods as deep-learning. The weights obtained after the learning have a physical meaning. It is possible to deduce information from these coefficients.

DEFENCE

Cyber Defence Lab
Royal Military Academy
www.cylab.be