Comparison of a Supervised Trained Neural Network Classifier and a Supervised Trained Aggregation Function Classifier

Alexandre Croix, Thibault Debatty, Wim Mees

Cyber Defense Lab, Royal Military Academy, Belgium

Email: a.croix@cylab.be

Presenter Information - Alexandre Croix

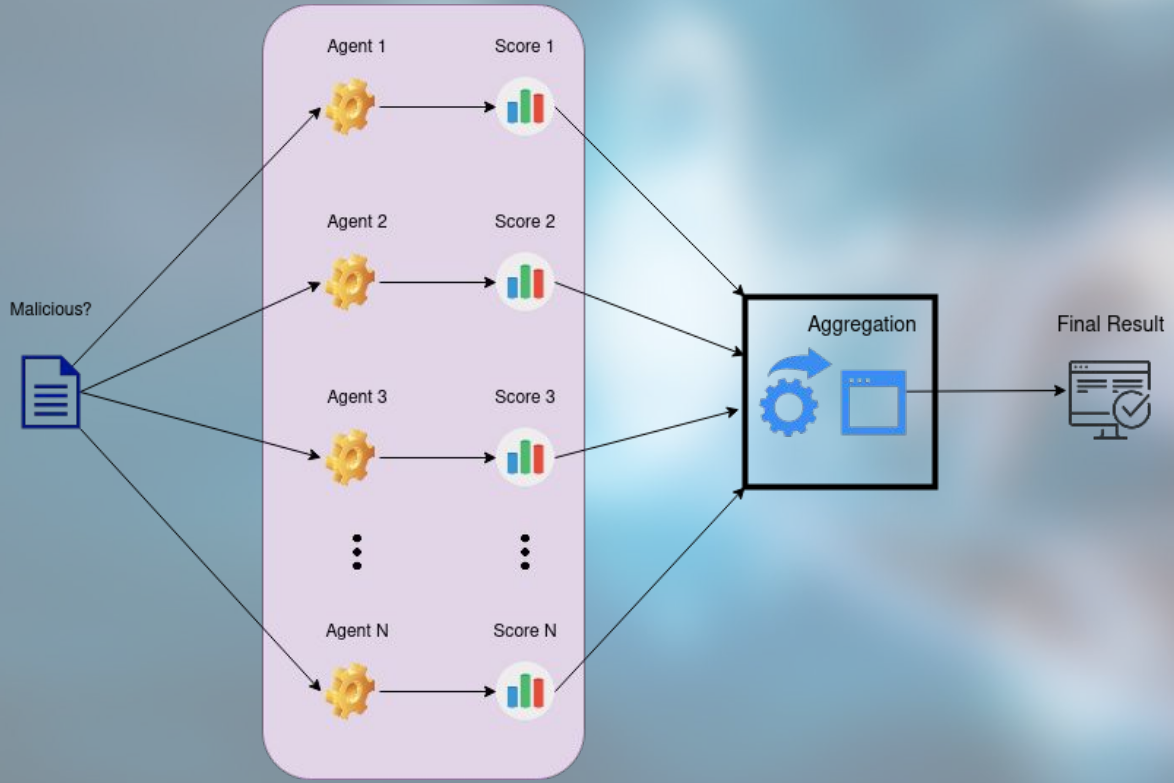
- Graduated as Industrial Engineer in 2018.
- Member of the Cyber Defense Research Unit at Royal Military Academy, Brussels
- GIAC Mobile Device Security Analyst (2019)

Context

- Cyber-defense systems are complex
- Multi-agents decision systems
- Several evaluation criterion
  - Agents
  - Scores

⇒ Need to aggregate scores

Context


Agent 1
Score 1
Agent 2
Score 2
Malicious?
Aggregation
Final Result
Agent 3
Score 3
Agent N
Score N

Context

- Compare efficiency two aggregation methods
  - Aggregation function trained by Genetic Algorithm
  - Artificial Neural Network trained by backpropagation
- Supervised training

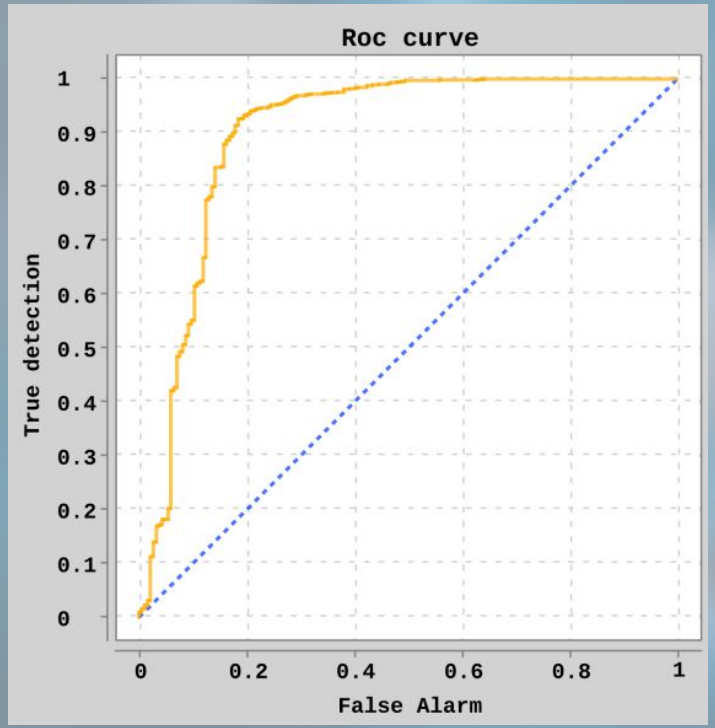
Context: task

- Training classifiers to distinguish if a PHP file is a webshell or an harmless file
- 23,415 PHP files (from PHP project)
  - 1,833 webshells
  - Analyzed by a 5-agents webshell detector

Evaluation criterion: ROC AUC

- **R**eceiver **O**perating **C**haracteristics curve(ROC)
  - Graphical tool
  - TPR against FPR
  - **A**rea **U**nder the **C**urve (AUC)

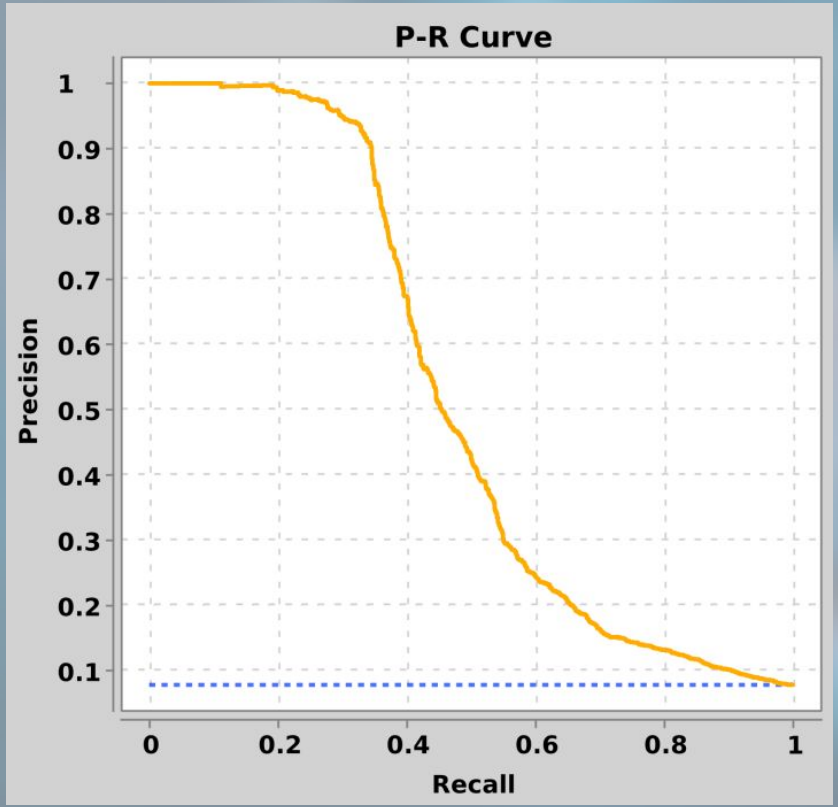
Evaluation criterion: ROC AUC



Evaluation criterion: P-R AUC

- **P**recision-**R**ecall curve (P-R)
  - Graphical tool
  - Precision against Recall
  - More informative for imbalanced dataset
  - **A**rea **U**nder the **C**urve (AUC)

Evaluation criterion: P-R AUC



Aggregation function: WOWA

- **W**eighted **O**rdered **W**eighted **A**veraging
- Introduced in 1997 by Vicenç Torra
- Combines WM and OWA advantages
- Requires two parameters for each data source
  - More complex

Aggregation function: WOWA

$$WOWA = f(a_1, a_2, \ldots, a_n, w_1, w_2, \ldots, w_n, p_1, p_2, \ldots p_n)$$

where

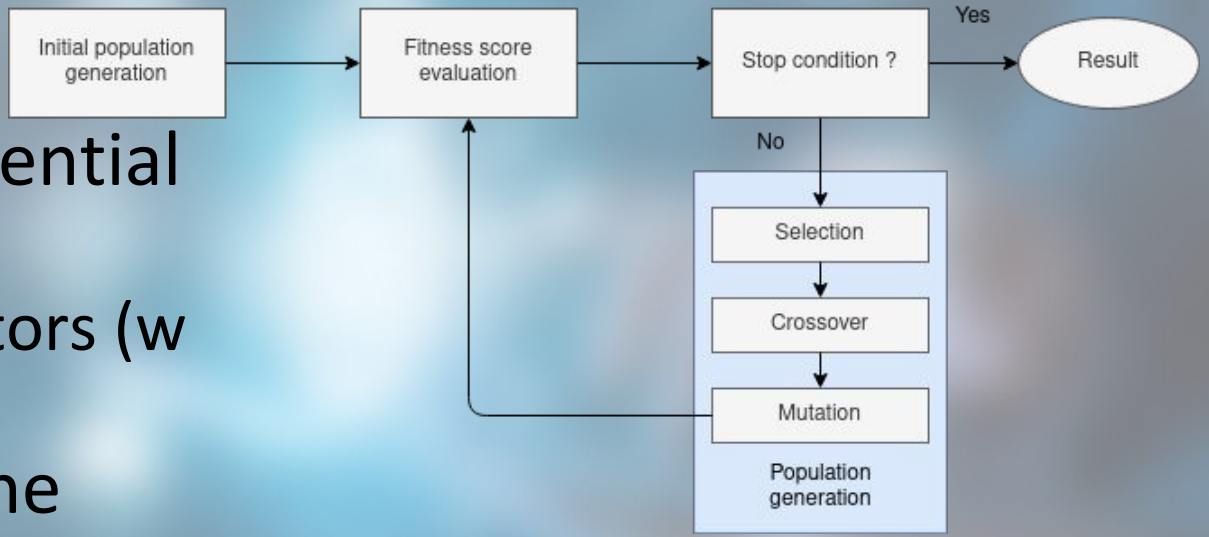
a_i are data sources

w_i are WM weights

p_i are OWA weights

Genetic Algorithm

- Iterative process
- Population of potential solutions
  - Two weight vectors (w and p)
- Parameters to tune



Genetic Algorithm: parameters

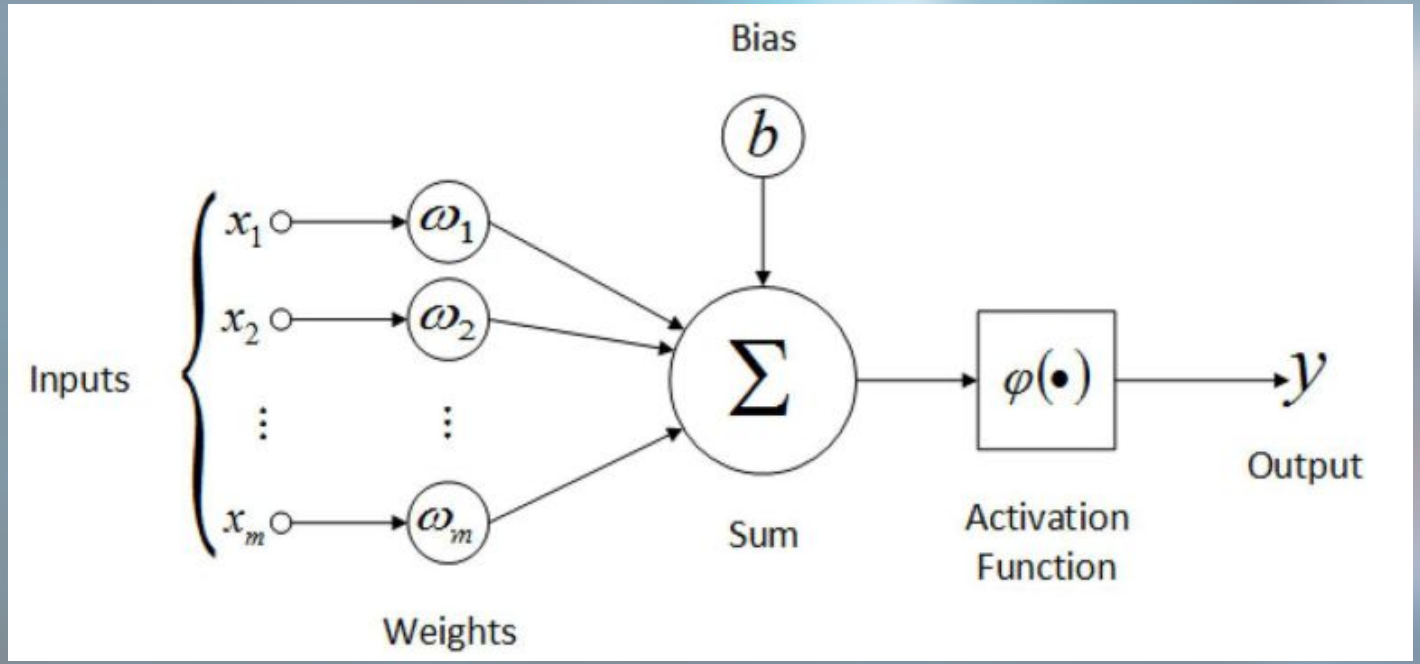
- Parametric study
  - **Population number**: 40 to 200
  - **Crossover rate**: 5 to 95
  - **Mutation rate**: 5 to 95
  - **Fitness score evaluation**: “Distance” or “AUC”

Genetic Algorithm: fitness score

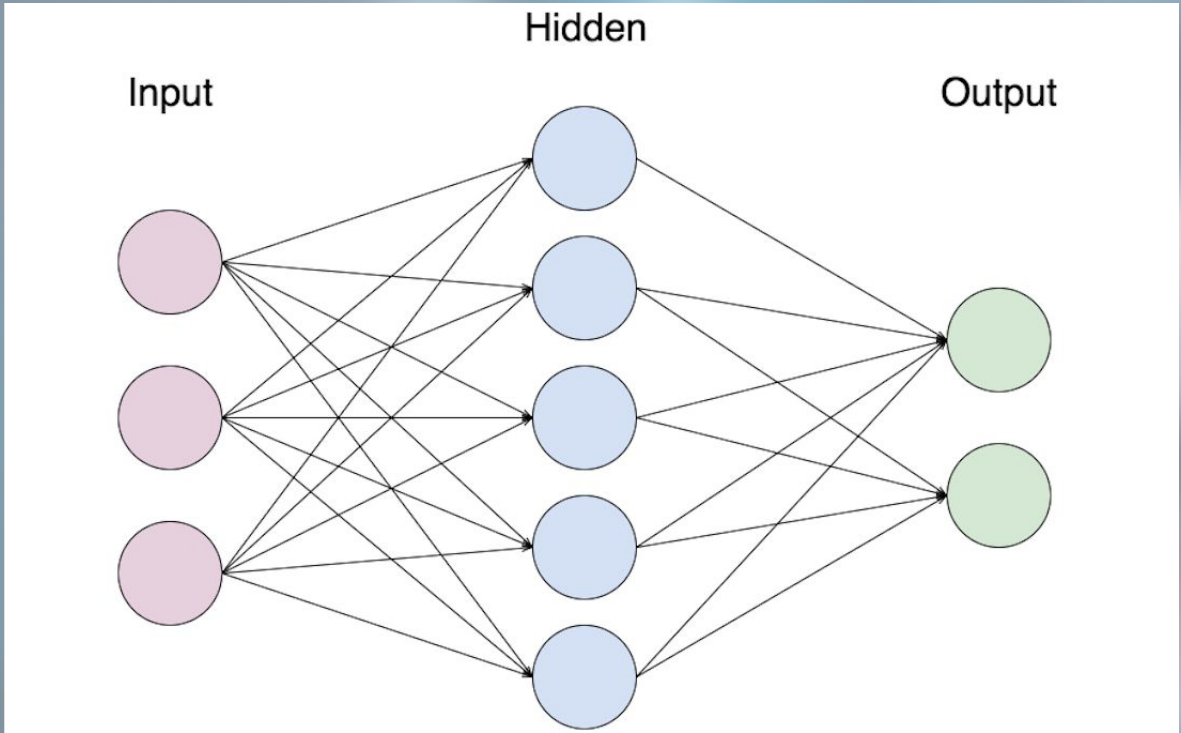
- Distance:
  - WOWA for each population element
  - Difference with the dataset result
  - Add all differences
- AUC:
  - WOWA for each population element
  - AUC of ROC curve

Neural Network

- Interconnection of neurons



Neural Network: structure



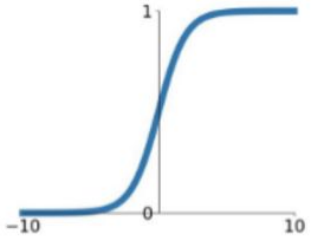
Neural Network: parameters

- Hyperparametric study
  - **Neurons number**: 5 to 50
  - **Learning rate**: 0.1 to 0.9, 0.01 to 0.09, 0.001 to 0.009
  - **Batch Size**: 1000 to 2000
  - **Epochs Number**: 100 to 350
  - **Activation Function**: tanh, ReLu, Sigmoid

Neural Network: parameters

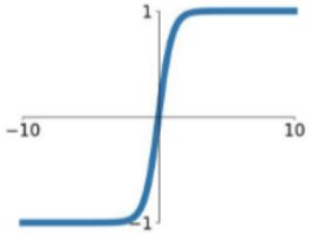
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



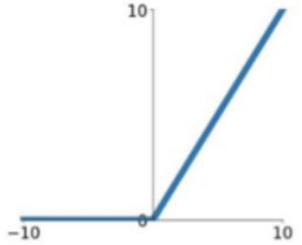
tanh

$$\tanh(x)$$



ReLU

$$\max(0, x)$$



Evaluation

- k-fold cross validation
  - Dataset separates in k folds
  - k - 1 folds used for training
  - Last fold for evaluation (P-R AUC and ROC AUC)
  - Repeat k times, rotating test set
  - Mean of intermediate results

Results: Genetic Algorithm

ROC criteria

- Population size: 75
- Crossover rate: 40
- Mutation rate: 20
- Fitness function: AUC

Result ≈0.88

P-R criteria

- Population size: 130
- Crossover rate: 30
- Mutation rate: 5
- Fitness function: AUC

Result ≈ 0.73

Results: Neural Network

ROC criterion

- Neurons number: 38
- Learning rate: 0.04
- Batch size: 2000
- Epochs number: 350
- Activation function: ReLu

Results [0.92;0.95[

P-R criterion

- Neurons number: 38
- Learning rate: 0.05
- Batch size: 2000
- Epochs number: 350
- Activation function: ReLu

Results [0.78; 0.84[

Results: comparison

- 10 times a k-fold cross validation
- Variance minimization

Classifier	ROC	P-R
Genetic Algorithm	0.900598	0.745871
Neural Network	0.946812	0.812567

Conclusions and future works

- Neural Network
  - More efficient
  - ReLu activation function always the best
  - Slow
  - Requires GPU (expensive)
- Genetic Algorithm
  - AUC fitness score always the best
  - Results can be interpreted

Conclusions and future works

- Bigger parametric study
- Other type of data
  - Dataset dependent?
- Correlation between parameters