Intrusion Detection Based on Genetic Fuzzy Classification System

Mariem Belhor
University of Sousse
Sousse, Tunisia
Mariem.belhor@gmail.com

Farah Jemili
University of Sousse
Sousse, Tunisia
Jmili_farah@yahoo.fr

Abstract — **Information system is vital for any company. However, the opening to the outside world makes the computer system more vulnerable to attack. It is essential to protect it. Intrusion Detection System (IDS) is an auditing mechanism that analyzes the traffic system and applications to identify normal use of the system and an intrusion attempt and also it prevent security managers. Despite the advantages of IDS, they suffer from a few problems. The major problem in the field of intrusion detection is the classification problem. Genetic Fuzzy System (GFS) are models capable of integrating accuracy and high comprehensibility in their results. They have been widely employed to solve classification problems. In this paper, we use a new GFS model called Genetic Programming Fuzzy Inference System for Classification (GPFIS-Class). It based on Multi-Gene Genetic Programming (MGGP). This model is not used in the intrusion detection area. We use an efficient feature selection method to eliminate data redundancy and irrelevant features in order to analyze the huge data namely the NSL-KDD data set.**

Keywords — **Intrusion detection, Genetic-fuzzy rule based classification, NSL-KDD, Feature selection. Multi-Gene Genetic programming**

I. INTRODUCTION :

Intrusion detection system (IDS) is a type of security mechanism for computer networks; it checks all the incoming and outgoing network activity and identifies suspicious patterns that may indicate a network attack from every intruder [1]. There are three types of IDS, which are; host-based IDS (HIDS), network-based IDS (NIDS) and hybrid-based IDS (HIDS). There are two main categories of intrusion detection techniques: signatures based detection and anomaly based detection [2]. Intrusion detection system has become the prime focus in the area of network security research. The challenging and critical problem in intrusion detection is the classification of attacks and normal network traffic. [3,4,5] there are various research has been applied with different data mining based classification techniques to classify the normal or attack data. Neural Networks [19], Support Vector Machines [20] and Genetic Programming [6] make it possible to solve classification problems with great accuracy. Fuzzy rule based classification systems (FRBCSs) are well known tools in the machine learning framework, since they can provide interpretable model [4, 25]. Genetic algorithms have been used for rule generation and optimization methods in the design of fuzzy rule based classifier [25]. The genetic algorithm based design of FRBCSs is usually referred as GFRBCSs. The main feature that highlights Genetic Fuzzy System (GFS) is its capability of extracting knowledge from datasets and states it in linguistic terms with reasonable accuracy. This is provided by the bond between a Fuzzy Inference System (FIS) and a Genetic Based Meta-Heuristic (GBMH), which is based on Darwinian concepts of natural selection and genetic recombination. Most works focus on developing or modifying methods in the GBMH component. However, few researchers have focused on developments of the Fuzzy Inference component to improve the performance obtained by the GFS and there is a lack of works using Genetic Programming (GP) [17, 18] as a GBMH for a GFS [6]. Our objective in the current work is first to take advantage of data mining techniques such as the feature selection method to eliminate data redundancy and irrelevant features in order to analyze the huge data namely the NSL-KDD. The second objective is the use of a new model of GFS which was not used in the intrusion detection area in order to solve the problem of classification and therefore to obtain a reliable IDS. The proposed model is the Genetic Programming Fuzzy Inference System for Classification (GFIS-CLASS).GPFIS-CLASS is a genetic fuzzy system based on Multi-Gene Genetic Programming (MGGP) [16,17,18] GPFIS-CLASS is an improvement to the GPF CLASS model [7].

This work is organized as follows: Section 2 describes GPFIS-CLASS and some basic concepts of the metaheuristic Multi-Gene Genetic Programming [11]. Section 3 presents the NSL-KDD dataset. The proposed system and

978-1-5090-4320-0/16/$31.00 ©2016 IEEE

experimental results will be discussed in section 4. Section 5 concludes the work.

II. GENETIC FUZZY INFERENCE SYSTEM FOR CLASSIFICATION

Genetic Fuzzy Classification Systems (GFCS) combine a FIS for classification and a metaheuristic inspired on Darwin's principle of natural selection and genetic recombination as a way of learning fuzzy rules "If-Then" [7,8]. The fuzzy rules consequents are crisp values that indicate to which class a given pattern belongs to [7, 8]. The rule is of the Disjunctive Normal form (DNF):

“If $X_1$is A_{m1}… and $X_J$is A_{MJ} then C is class k with CD”

Where X_j (j = 1... J) represents input pattern, A_{mj} (m= 1, ..., M) represents the linguistic term, while the output class C is characterized by one of the k possible class (k = 1, ..., K). CD ∈ [0, 1] represents the confidence of the respective class k based on its support in the database. [7, 8]

Among several learning and codification schemes for a Genetic Fuzzy Rule Based Classification System (GFRBCS), the Pittsburgh [21] the Michigan [22] Genetic Competitive-Cooperative Learning (GCCL) [23] and Iterated Rule Learning (IRL) [24]. In the literature of GFRBCS works using Genetic Programming (GP), generally follow the Pittsburgh style [21] and the GCCL configuration [23]. When GP is combined with a FIS in a classical manner, GP becomes a supporting component to the inference process; its abilities for feature selection and finding a functional structure are little explored. [8] The proposed model (GPFIS-Class) searches for a greater integration with GP, or, as in the current case, Multi-Gene Genetic Programming (MGGP) [6]. It is therefore highly hybridized. GPFIS CLASS seeks to explore the potentialities of MGGP, and at the same time to provide a linguistic comprehension similar to that of traditional GFCS. [7,8]

A. Multi-Gene Genetic Programming

Genetic Programming (GP) [6] is an extension of genetic algorithms [9]. It is a search method that encodes multi-potential solutions for specific problems. The programs can be represented as parse trees. Employs a population of individuals, each of them denoted by a tree structure which describes the relationship between a set of input features X_j (j =1,.., J) and the output Y. Multi-Gene Genetic Programming (MGGP) generalizes GP, as it denotes an individual as set of tree structures, commonly called genes, that also receives the set of features X_j and tries to predict Y (See fig. 1).

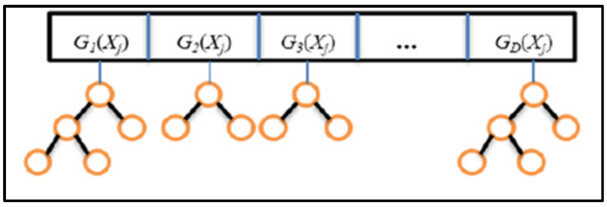


Fig. 1: Example of multi-gene individual [7,8]

Each individual is composed of D genes (d = 1. . . D), where each gene is a partial solution to the problem. when D = 1, MGGP generates solutions similar to GP. [6,7] In relation to the genetic operators, the mutation operation in MGGP is similar to that in GP. As for crossover: we can apply crossover at high and low levels.

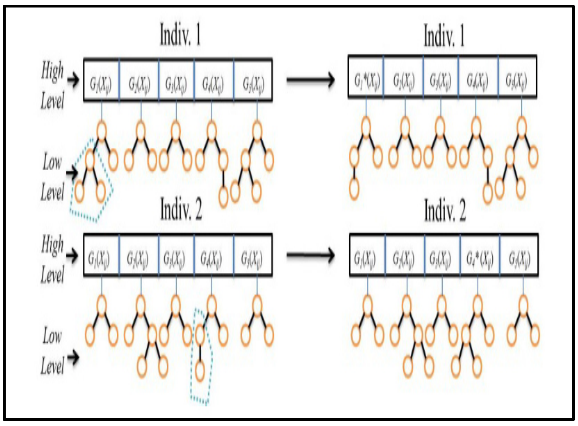


Fig. 2: Crossover in low level [7,8]

Figure 2 presents a multi-gene individual with five equations (D=5) accomplishing a low-level crossover. Figure 3 exhibit the mutation operation. [7, 8]

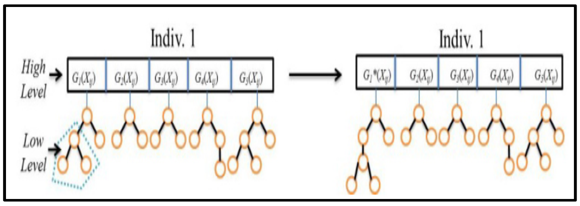


Fig.3: mutation operation [8]

The low-level is the space where it is possible to manipulate the structures (terminals and functions) of equations presented in an individual. The high-level is the space where the expressions can be manipulated. An example of high-level crossover is displayed in Figure 4. [7,8]

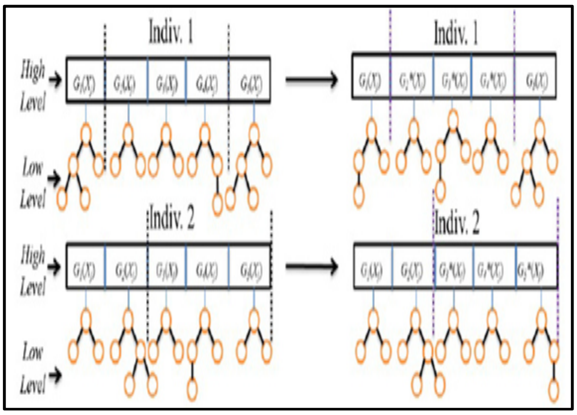


Fig. 4: High level crossover in MGGP [8]

B. *Components of GPFIS-CLASS:*

GPFIS-CLASS is a typical Pittsburgh-type GFS [21], in which each individual represents a fuzzy rule base. Fig. 5 exhibits the main modules of the GPFIS-CLASS model. [8]

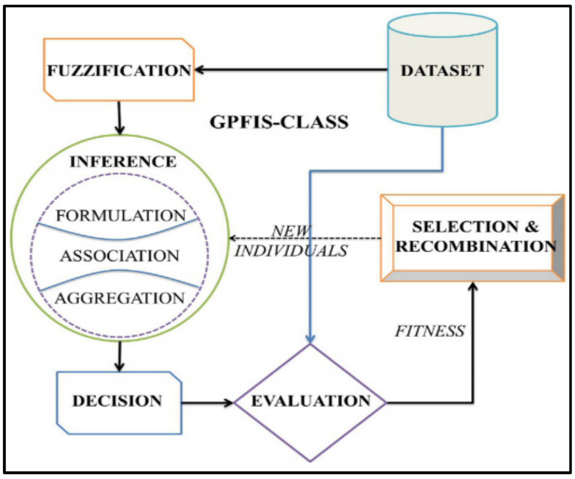


Fig. 5: Diagram describing the main stages of GPFIS-CLASS [8]

1) *Fuzzification :*

The fuzzification step involves the association of fuzzy sets to each input feature. Therefore, each j-th feature has L associated fuzzy sets: [7, 8]

A_{lj} = {(x_{ij}, $\mu_{lj}(x_{ij})$)| x_{ij} ∈X_j}, where $\mu_{A_{lj}}$: X_j→ [0, 1] is a membership function that assigns to each value x_{ij} a membership degree $\mu_{lj}(x_{ij})$ to the fuzzy set A_{lj}. Each i-th pattern belongs to a class C of K possible classes. [8]

2) *Fuzzy inference*

The inference process in GPFIS-CLASS is subdivided into 3 steps: [8] (show fig.8)

a) Formulation: responsible for combining the linguistic terms of each feature to build a fuzzy rule premise by using MGGP.A fuzzy rule premise is generally defined as:

“If $X_1$is A_{l1}, and…, and $X_J$is A_{lJ}"

Each gene encoded a fuzzy rule premise .see fig 6

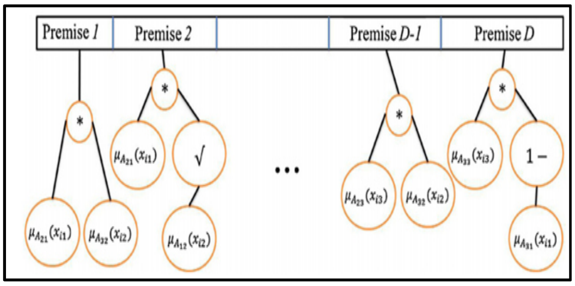


Fig. 6: set of premises encoded by a MGGP individual [8]

b) Association : given a set of premises, this step verifies the consequent class that is most suited to each premise (fuzzy rule creation and screening) [8]

“If X_1 is A_{l1}, and.., and X_J is A_{lJ} Then x_i is Class k”

A simple association technique is called "uniform Division" which is calculated for each individual MGGP.

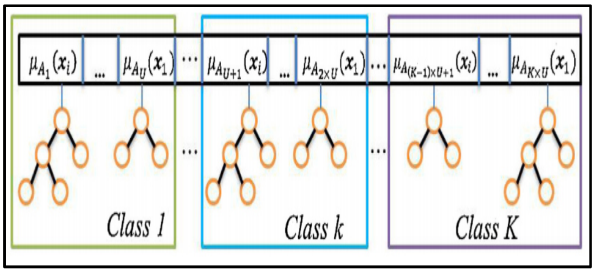

$\mu_{A_1}(x_i)$
…
$\mu_{A_U}(x_1)$
$\mu_{A_{U+1}}(x_i)$
$\mu_{A_{2\times U}}(x_1)$
$\mu_{A_{(K-1)\times U+1}}(x_i)$
$\mu_{A_{K\times U}}(x_1)$
Class 1
Class k
Class K

Fig. 7: Example of uniform division method [8]

U = D / K, where D is the number of premises and K is the number of possible classes. The premises are divided as follows: [8]

Class 1: $\mu_{A_{1(1)}}(x_i)$, …,$\mu_{A_{U(1)}}(x_i)$

Class 2: $\mu_{A_{U+1(2)}}(x_i)$, …., $\mu_{A_{2\times U(2)}}(x_i)$

…

Class K: $\mu_{A_{(K-1)\times U+1(K)}}(x_i)$, … , $\mu_{A_{K\times U(K)}}(x_i)$

This method may result in the association of a consequent premise incorrect class. To avoid this, each premise must be assigned to the k-th class that maximizes a measure of compatibility between a specific premise and consequent class. A compatibility and widely used measure is the degree of certainty: [8]

$$CD_k = \frac{\sum_{i\in k}\mu_{A_d(x_i)}}{\sum_i^n \mu_{A_d(x_i)}} \quad (1)$$

Where $\sum_{i\in k}\mu_{A_d(x_i)}$ the compatibility degree of d-th is premise with respect to the k-th class, and $\sum_i^n \mu_{A_d(x_i)}$ is its compatibility degree to all classes. If CD_k = 0 for a certain premise, then no consequent is associated to it. [8]

c) Aggregation: receives as input the activated fuzzy rules and computes a consensual value for each consequent class.

Consider $D^{(k)}$ the number of fuzzy rules associated to the k-th class. Given an aggregation operator g: $[0, 1]^{D^{(k)}} \rightarrow [0, 1]$, the merged membership degree of x_i^* to each of the K classes $(\hat{\mu}_{C_{i\in k}(x_i^*)})$ is:

$(\hat{\mu}_{C_{i\in 1}(x_i^*)})$ = g[$\mu_{A_{1(1)}}(x_i^*)$,..., $\mu_{A_{d(1)}}(x_i^*)$,, ..., $\mu_{A_{D(1)}}(x_i^*)$] (2)

$(\hat{\mu}_{C_{i\in 2}(x_i^*)})$ = g[$\mu_{A_{1(2)}}(x_i^*)$,..., $\mu_{A_d}(x_i^*)$,, ..., $\mu_{A_{D(2)}}(x_i^*)$] (3)

…

$(\hat{\mu}_{C_{i\in k}(x_i^*)})$ = g[$\mu_{A_{1(k)}}(x_i^*)$,..., $\mu_{A_{d(k)}}(x_i^*)$,, ..., $\mu_{A_{D(k)}}(x_i^*)$] (4)

The aggregation operator used in this step is called Weighted Average Restricted Least Squares (WARLS), and its mathematical expression is presented below:

$$\hat{\mu}_{C_{i\in 1}(x_i^*)} = \sum_{d^{(1)}=1}^{D^{(1)}} w_{d^{(1)}}\, \hat{\mu}_{A_{d^{(1)}}(x_i)} \quad (5)$$

$$\hat{\mu}_{C_{i\in 2}(x_i^*)} = \sum_{d^{(2)}=1}^{D^{(2)}} w_{d^{(2)}}\, \hat{\mu}_{A_{d^{(2)}}(x_i)} \quad (6)$$

…

$$\hat{\mu}_{C_{i\in k}(x_i^*)} = \sum_{d^{(k)}=1}^{D^{(k)}} w_{d^{(k)}}\, \hat{\mu}_{A_{d^{(k)}}(x_i)} \quad (7)$$

3) Decision:

Given a new pattern x_i^*, the decision on the k-th class it belongs to is performed by: [8]

$\hat{C}_i = arg_k \max \{\hat{\mu}_{C_{i\in 1}(x_i^*)}, ..., \hat{\mu}_{C_{i\in k}(x_i^*)}, ..., \hat{\mu}_{C_{i\in K}(x_i^*)}\}$ (8)

Where $\hat{C}_i$is the predicted class: a result of the k-th argument with the maximum value in (8).

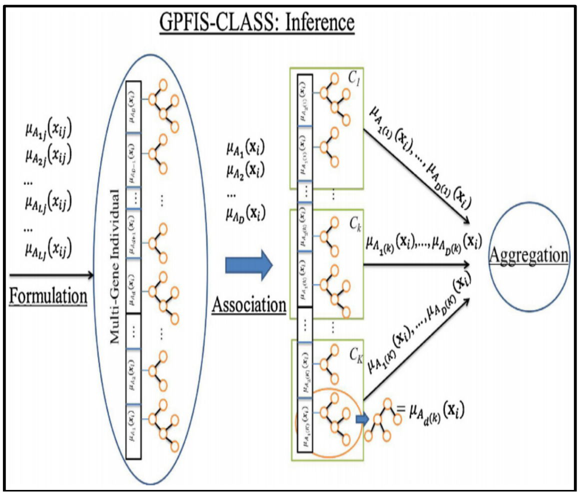


Figure 8: GPFIS-CLASS Inference process [8]

4) Evaluation:

Evaluation in GPFIS-CLASS consist on minimize classification error. It is responsible for ranking the individuals in the population. A simple fitness function for classification problems is the Mean Classification Error (MCE):

$$\text{MCE} = \frac{\sum_{i=1}^{n} |C_i - \hat{C}_i|}{n} \quad (9)$$

Where $|C_i - \hat{C}_i|$ = 0 if $C_i - \hat{C}_i$and 1 otherwise. The individual that minimizes MCE is considered the best one in the population.

5) Selection and recombination

After the evaluation process, a set of individuals is selected and recombined. Mutation, low-level crossover or high-level crossover, is applied to some subset. Finally, the new population is generated.

In this work, we use the new classification model GPFIS-CLASS in the intrusion detection area whose goal is to solve the problem of classification and improve the performance of IDS. In the following description, we will introduce the dataset NSL-KDD on which we will evaluate our contribution.

III. NSL-KDD DATASET:

A. Dataset description

The NSL-KDD data set is the refined version of the KDD cup99 data set [26]. Many types of analysis have been carried out by many researchers on the NSL-KDD dataset employing different techniques and tools with a universal objective to develop an effective intrusion detection system. [25]

NSL-KDD dataset contains one type of normal data and 22 different types of attacks which falls into one of four categories. These are Dos (deny of service), probe (information gathering), R2L (remote to local) and U2R (user to root). [28] (See table I)

TABLE I: attacks categories

Class	**Elements**
Normal	normal
DOS	Neptune, back, land, smurf, teardrop, pod
Probe	Ipsweep,nmap,portsweep,satan
U2R	Buffer_overflow, loadmodule,perl,rootkit
R2L	ftp_write, guess_passwd ,imap, multihop, phf, warezmaster, warezclient

It contains essential records of the complete KDD data set. In each record there are 41 attributes. The 42nd attribute contains data about the various 5 classes of network connection vectors and they are categorized as one normal class and four attack class.

B. Feature selection

Feature selection is important to improving the efficiency of data mining algorithms. [28,29] It is one of the important preprocessing steps for reducing features for NSL KDD dataset due to its high dimensionality data.[3]Not all attributes that exist in the dataset has an influence on class labels. Therefore eliminating attributes that are not relevant to class labels of a dataset is an important thing to do to improve the performance of the classifier. [3,28]

In this paper, we propose to use a new feature The proposed method is called "Modified FVBRM (Feature Vitality Based Reduction Method)" [3]. Modified FVBRM is the modification of FVBRM [28].It is determinate by considering three parameters of performance measurement of classification:

- Classification Accuracy (CA)
- Detection rate (DTR)
- False positive rate (FPR)

This method uses a sequential search to identify the important set of features: starting with the set of all features, one feature was removed at a time until the accuracy of the classifier was below a certain threshold. Then, it compares the new results with the original results. The experiment is repeated 41 times to ensure that each feature is either important, unimportant or less important. [3, 28]

This new method was compared with four other feature selection methods: [3]

- Correlation-based Feature Selection (CFS)
- Information Gain (IG)
  Gain Ratio (GR)
- Original Feature Vitality Based Reduction Method (FVBRM)

The feature reduction obtained 10, 25, 4, 24 and 21 by CFS, GR, IG, FVBRM and Modified FVBRM respectively. [3] (See TableII)

TABLE II: Selected attribute by each method [3]

Feature selection method	**Number of selected attribute**	**Selected attribute**
CFS	10	3, 4, 5, 6, 12, 26, 29, 30, 37,38
IG	4	3, 4, 5,30
GR	25	3,4,5,6,10,11,12,14,17,18,22,23,25,26,27,29,30, 31,33,34,35,37,38,39
FVBRM	24	2,3,4,8,10,11,12,14,17,18,19,22,23,24,25,30,31, 32,33,35,36,37,38,39
Modified FVBRM	21	2,3,4,8,10,11,12,14,17,18,19,22,23,24,30,31,32, 33,35,36,37

Accuracy classification (CA), false positive rate (FPR) and detection rate (DTR) are based on the confusion matrix shown in table III: [3, 28]

Confusion Matrix: [3]

This may be used to summarize the predictive performance of a classifier on test data.

TABLE III: Confusion Matrix

	Normal	**Attack**
Normal	**TP**	**FP**
Attack	**FN**	**TN**

- True Positive (TP), the actual class of the test instance is positive and the classifier correctly predicts the class as positive.
- False Positive (FP), the actual class of the test instance is negative but the classifier incorrectly predicts the class as positive.
- True Negative (TN), the actual class of the test instance is negative and the classifier correctly predicts the class as negative.
- False Negative (FN), the actual class of the test instance is positive but the classifier incorrectly predicts the class as negative.

According to the results observed in [3], Modified FVBRM has the highest with 21 features at 88, 22%, the highest TPR 86, 9% and the lowest FPR 0,02%. For this reason, we chose Modified FVBRM.

In this work, we will apply this method with a new classifier (GPFIS-CLASS) which hasn't been used in the intrusion detection area to improve the performance of IDS.

IV. EXPERIMENTATIONS:

We propose a new classification model called (GPFIS-Class), it has been proposed by [8] in 2015. It has-been evaluated in two sets of benchmarks comprising a total of 45 datasets, and has been compared with eight different classifiers, six of them based on GFSs. The results obtained in both sets demonstrate that GPFIS-CLASS provides better results for most benchmark datasets. [8]We decided to use this model because it provides a very high classification accuracy rate. Additionally, we propose an efficient feature selection method in order to eliminate irrelevant features from NSL KDD, the proposed method is named (Modified FVBRM). To evaluate our contribution, we use 20% de NSL-KDD dataset and we compare our results with other GFCS which are used in intrusion detection area.

A. Architecture of the proposed method

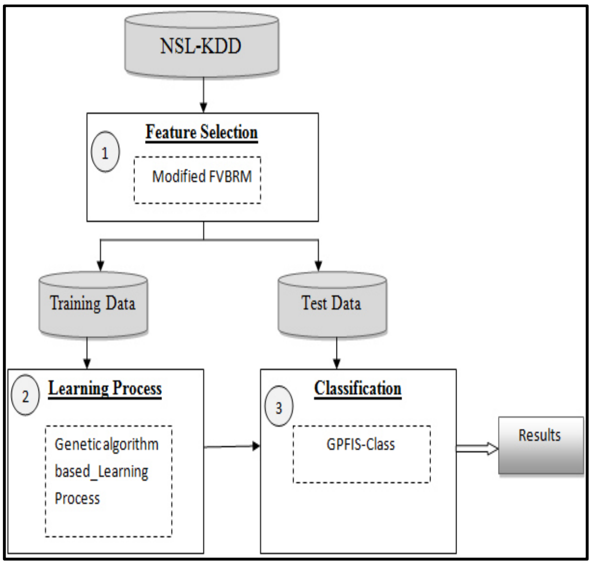


Figure 6: Architecture of the proposed method

1) Feature selection:

We applied the "Modified FVBRM" attribute selection method on NSL-KDD dataset and got under the following set of relevant attributes: (21 attributes):

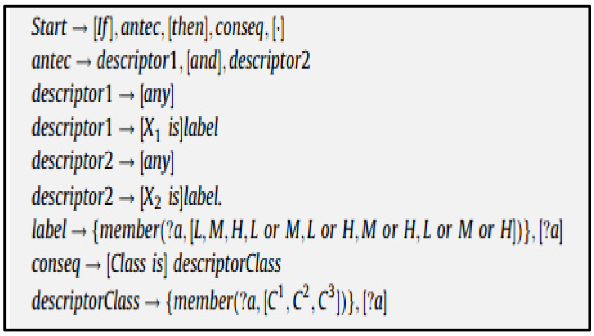
Protocol_type(2),Service(3),Flag(4),Wrong_fragment(8),Hot(10),Num_failed_logins(11),Logged_in(12),Root_shell(14),Num_file_creations(17),Num_shells(18),Num_access_files(19),Is_guess_login(22),Count(23),Service_count(24),Serror_rate(25)Diff_srv_rate(30),diff_host_rate(31),Dst_host_count(32),Dst_host_srv_count(33),Dst_host_diff_srv_rate(35),Dst_host_same_src_port_rate(36),Dst_host_srv_diff_host_rate (37)

2) Learning process:

The table below summarizes the methods and approaches used in our contribution:

FRBS	GFRBS
Learning method	Pittsburgh
Meta-heuristic	MGGP
FIS type	Mamdani

We build a population from the available options of the rules. Each rule represents one chromosome (Pittsburgh approach). Fig.10 shows a grammar example:



$$Start \rightarrow [If], antec, [then], conseq, [\cdot]$$
$$antec \rightarrow descriptor1, [and], descriptor2$$
$$descriptor1 \rightarrow [any]$$
$$descriptor1 \rightarrow [X_1\ is]label$$
$$descriptor2 \rightarrow [any]$$
$$descriptor2 \rightarrow [X_2\ is]label.$$
$$label \rightarrow \{member(?a, [L, M, H, L\ or\ M, L\ or\ H, M\ or\ H, L\ or\ M\ or\ H])\}, [?a]$$
$$conseq \rightarrow [Class\ is]\ descriptorClass$$
$$descriptorClass \rightarrow \{member(?a, [C^1, C^2, C^3])\}, [?a]$$

Fig.10: Grammar example

The rules are of the following form:

If **Protocol_type** is Medium AND **Service** is high AND **Flag** is Medium AND **Wrong_fragment** is high AND **Hot** is high AND **Num_failed_logins** is high AND **Logged_in** is Medium AND **Root_shell** is low AND **Num_file_creations** is high AND **Num_shells** is low AND **Num_access_files** is low AND **Is_ guess_ login** is low AND **Count** is high AND **Service_count** is low AND **Diff_srv_rate** is medium AND **Srv_diff_host_rate** is low AND **Dst_host_count** is Medium AND **Dst_host_srv_count** is high AND **Dst_host_diff_srv_rate** is Medium AND **Dst_host_same_src_port_rate** is Medium AND **Dst_host_srv_ diff_host_rate** is high Then Class is **R2L**

A new population is obtained through standard crossover and mutation operators applied to the chromosomes. The fitness of an individual is determined as the root mean square error (RMSE) between the actual and predicted values. The predicted values are obtained from fuzzy reasoning using the Mamdani model. The final solution is obtained as the best individual after generating the maximal number of generations. [30]

(1) *Generate an initial population of linguistic rules;*
(2) *Evaluate each linguistic rule in the current population;*
(3) *Generate new linguistic rules by means of genetic operations;*
(4) *Replace a part of the current population with the newly generated rules;*
(5) *Terminate the algorithm if a stopping condition is satisfied, otherwise return to Step 2.*

3) Classification:

In this step, we use the KDDTest+ dataset for evaluate our classifier. The individual in GPFIS-CLASS represents a fuzzy rule base. Our model is composed of the following steps: (show section II)

i. *Fuzzification: involves the association of fuzzy sets to each input feature*

ii. *Formulation: generation of fuzzy rule premises*

iii. *Association: assignment of the best suited consequent term for each premise*

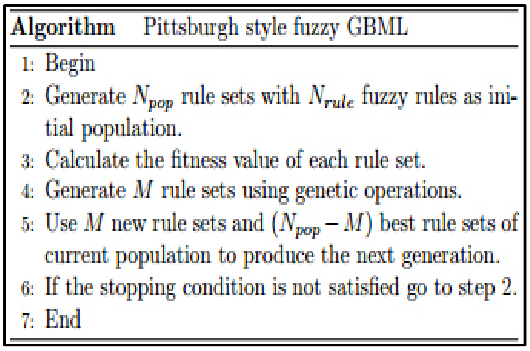
iv. *Aggregation: aggregation of activated fuzzy rules*

v. *Decision: associates to the activate fuzzy rule the class that provides the highest merged membership degree*

vi. *Evaluation: minimize classification error (fitness:MCE)*

vii. *Application of genetic operators like crossover (high/low level) and mutation*

Figure below shows an algorithm of Pittsburgh style:



```
Algorithm Pittsburgh style fuzzy GBML
1: Begin
2: Generate $N_{pop}$ rule sets with $N_{rule}$ fuzzy rules as initial population.
3: Calculate the fitness value of each rule set.
4: Generate $M$ rule sets using genetic operations.
5: Use $M$ new rule sets and $(N_{pop} - M)$ best rule sets of current population to produce the next generation.
6: If the stopping condition is not satisfied go to step 2.
7: End
```

Fig.11: Pittsburgh Algorithm

The main configurations of GPFIS-CLASS are shown in table VI.

TABLE VI: Main configuration of GPFIS-ClASS

Number of features	21
Population size	100
Number of generation	100
Tree Maximum Depth	7
T-norm/T-conorm	Min/Max
Fitness	RMSE
High level crossover rate	30%
lowlevel crossover rate	15%
Mutation rate	10%
Association	CD
Aggregation	WARLS
Decision	Class with higher membership
Evaluation	MCE

1. Evaluation of classifier:

To evaluate the classifier we relied on the following measures: True Positive rate (TPR), False Positive Rate (FPR) and Classification accuracy (CA):

*True Positive Rate (TPR) is defined as:

$$TPR=TP/\ (TP+FN) \qquad (10)$$

*False Positive Rate (FPR) is:

$$FPR=FP/\ (TN+FP) \qquad (11)$$

*We can obtain the accuracy of a classifier by:

$$Accuracy=(TP+TN)/(TP+FN+FP+TN) \qquad (12)$$

*Mean F-Measure: We compute the average for the F-measure achieved for each class (taken as positive) and the remaining ones (taken as a whole as negative):

$$\mathrm{MFM} = \frac{\sum_{i=1}^{C} \mathrm{FM_i}}{\mathrm{C}} \qquad (13)$$

$$\mathrm{FM_i} = \frac{2.\mathrm{Recall_i}.\mathrm{Precision_i}}{\mathrm{Recall_i}.\mathrm{Precision_i}} \qquad (14)$$

$$\mathrm{Precision} = \frac{\mathrm{TP_i}}{\mathrm{FP_i}+\mathrm{TP_i}} \qquad (15)$$

$$\mathrm{Recall} = \frac{\mathrm{TP_i}}{\mathrm{FN_i}+\mathrm{TP_i}} \qquad (16)$$

*Attack Detection Rate (ADR): It stands for the accuracy rate for the attack classes. Therefore, it is computed as:

$$\mathrm{ADR}= \frac{\sum_{i=2}^{C} TP_i}{\sum_{i=2}^{C} TP_i+\mathrm{FN_i}} \qquad (17)$$

Reader must take into account that also in this case, the first class (i=1) is considered to be the ‘‘Normal’’ class.

We compare GPFIS-CLASS with 2 others classifiers based on GFS which are used in intrusion detection area:

-FARCHD-OVO: is the combination of genetic fuzzy systems and pairwise learning for improving detection rates on Intrusion Detection Systems. [30]

-MOGFIDS: multi-objective genetic fuzzy intrusion detection system [31]

TABLE VII: comparison of classification accuracy between classifiers

Classifier	Acc	MFM	ADR
FARCHD-OVO	99,00%	84,12%	97.77%
MOGFIDS	92,77%	61,68%	91,47%
Proposed Method	98,70%	81,60%	98,50%

According to the results in Table VII and Table VIII, we observe that our approach has higher classification accuracy with 21 features. One of the main advantages of our new method is the homogenous accuracy for all classes of the problem. The good behavior shown by our methodology is supported by the advantages derived from the use of fuzzy logic, Genetic programming multi-gene, linguistic labels, GFS and the feature selection method.

TABLE VIII: classification accuracy results for every class

Classifier	Normal	Dos	Probe	R2L	U2R
FARCHD-OVO	99.81 %	98.05 %	95.83 %	87.54 %	65.38 %
MOGFIDS	98.36 %	97.20 %	88.60 %	11.01 %	15.79 %
Proposed Method	99,0%	99,0%	96,60 %	82,4%	67,3%

V. CONCLUSION:

The main objective of this work is to solve the problem of classification in intrusion detection field and obtain reliable IDS. We used a new GFS which was not used in the intrusion detection domain named GPFIS-CLASS. This model is based on genetic programming multi-gene. We also used an effective method of feature selection. It has been tested on NSL-KDD using Naive Bayes classifier. We integrate this solution, which is called Modified FVBRM, it consist to select the relevant features from NSL-KDD to improve classification accuracy. We compare our method with other classifiers which are based GFS and are already used in the intrusion detection field. Our results show that the proposed method Achieve the Higher classification accuracy compared with other GFS. The hybridization between Fuzzy logic and evolutionary algorithms is the major factor in the success of this method. In future work, we will apply a new GFS hybridized with neural networks in order to resolve the classification problem in intrusion detection area.

REFERENCES:

[1] L. Riza, F. Herrera, C. Bergmeir, J. Benitez. “Fuzzy Rule-Based Systems for Classification and Regression in R”, Journal of Statistical Software, May 2015, Volume 65, Issue 6.

[2] H. Debar, M. Dacier, and A. Wespi. “A revised Taxonomy for intrusion detection systems“. Annales des télécommunications, 2000.

[3] Jupriyadi,A. Kistijantoro, Vitality Based Feature Selection for Intrusion Detection, 2014 International Conference of Advanced Informatics: Concept, Theory nd Application (ICAICTA)

[4] P.Prakash, R. Bharti, Intrusion Detection System using Genetic-Fuzzy Classification

[5] I.Gaied, F.Jemili, O. Korbaa. Intrusion Detection Based on Neuro-Fuzzy Classification.2015

[6] Francisco-Herrera-Frank-Hoffmann-Luis Magdalen Oscar-Cordon-Oscar-Cordon-Genetic-fuzzy-systems_-evolutionary-tuning-and-learning-of-fuzzy-knowledge-bases-World-Scientific-Publishing-Company-2001

[7] A. Koshiyama, T. Escovedo, D. Dias, M. Vellasco, R. Tanscheit, GPF-class: a genetic fuzzy model for classification, in: 2013 IEEE Congress on Evolutionary Computation (CEC), 2013, pp. 3275–3282.

[8] A. Koshiyama, T. Escovedo, D. Dias, M. Vellasco, R. Tanscheit, GPF-class: a genetic fuzzy model for classification, Elsevier2015

[9] S. M. Bridges and R. B. Vaughn, “Fuzzy data mining and genetic algorithms applied to intrusion detection,” in proc. National Information Systems Security Conference (NISSC), Baltimore, MD, 2000, pp.16-19.

[10] J. Gomez and D. Dasgupta, “Evolving fuzzy classifiers for intrusion detection,” in proc. IEEE Workshop on Information Assurance, United States Military Academy, West Point, NY, 2002, pp. 68-75.

[11] Z. Lei, M. Lingrui, and H. Chunjie, “Intrusion detection based on immune principles and fuzzy association rules,” Intelligence Computing and Evolutionary Computation Advances in Intelligent Systems and Computing, vol. 180, pp. 31-35, Springer, 2013.

[12] Science, Springer Berlin Heidelberg, 2006, pp. 182–191. D. Boughaci, M.D.E. Kadi, M. Kada, Fuzzy particle swarm optimization for intrusion detection, in: T. Huang, Z. Zeng, C. Li, C.S. Leung (Eds.), Neural Information Processing, vol. 7667 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2012, pp. 541–548.

[13] M.F. Ganji, M.S. Abadeh, A fuzzy classification system based on ant colony optimization for diabetes disease diagnosis? Expert Syst. Appl. 38 (12) (2011) 14650–14659.

[14] C.-F. Juang, P.-H. Chang, Designing fuzzy-rule-based systems using continuous ant-colony optimization? IEEE Trans. Fuzzy Syst. 18 (1) (2010) 138–149.

[15] S. Luke, L. Panait, Lexicographic parsimony pressure, in: W.B. Langdon, E. Cantú- Paz, K. Mathias, R. Roy, D. Davis, R. Poli, K. Balakrishnan, V. Honavar, G. Rudolph, J. Wegener, L. Bull, M.A. Potter, A.C. Schultz, J.F. Miller, E. Burke, N. Jonoska (Eds.), GECCO 2002: Proceedings of the Genetic and Evolutionary Computation Conference, Morgan Kaufmann Publishers, New York, 2002, pp. 829–836.

[16] K.M. Faraoun,A. Boukelif. Genetic Programming Approach for Multi-Category Pattern Classification Applied to Intrusions Detection, International Scholarly and Scientific Research & Innovation , 2007

[17] John R. Koza, Genetic Programming: On the Programming of Computers by Means of Natural Selection, Complex adaptive systems "A Bradford book",1992.

[18] Linqiang Pan Gheorghe J. Pérez-Jiménez Tao Song (Eds.),Bio-Inspired Computing – Theories and Applications, 9th International Conference, BIC-TA 2014 Wuhan, China, October 16-19, 2014

[19] S. Haykin, Neural Networks: A comprehensive foundation, ed.2. New Jersey: Prentice Hall, 2004.

[20] N. Cristianini and J. Shawe-Taylor, Support Vector Machines. Cambridge: Cambridge University Press, 2000

[21] A.A. Tsakonas, A comparison of classification accuracy of four genetic programming-evolved intelligent structures. Information Sciences, vol. 176, pp. 691-724, 2006.

[22] J. Casillas, B. Carse, L. Bull, Fuzzy-XCS: A Michigan genetic fuzzy system. IEEE Transactions on Fuzzy Systems, vol.15, pp. 536-550, 2007.

[23] H. Ishibuchi, T. Nakashima and T. Murata, Design of accurate classifiers with a compact fuzzy-rule base using an evolutionary scatter partition of feature space. IEEE Transactions on Systems, Man and Cybernetics, Part B, vol. 29, pp. 601–618,1999.

[24] González and R. Pérez, Selection of relevant features in a fuzzy genetic learning algorithm. IEEE Transactions on Systems,Man and Cybernetics, Part B, vol. 31, pp. 417-425, 2001.

[25] L.Dhanabal, S.P. Shantharajah, A Study on NSL-KDD Dataset for Intrusion Detection System Based on Classification Algorithms, *International Journal of Advanced Research in Computer and Communication Engineering Vol. 4, Issue 6, June 2015*

[26] http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.tml

[27] https://web.archive.org/web/20150205065733/http:// sl.cs.unb.ca/NSL-KDD/

[28] Saurabh Mukherjee, NeelamSharma, Intrusion Detection using Naive Bayes Classifier with Feature Reduction, Elsevier, 2011.

[29] Hee-su Chae, Byung-oh Jo, Sang-Hyun Choi, Twae kyung Park, Feature Selection for Intrusion Detection using NSL-KDD, 2014.

[30] S.Elhag , A.Fernández , A.Bawaki, S. Alshomrani , .Herrera, On the combination of genetic fuzzy systems and pairwise learning for improving detection rates on Intrusion Detection Systems, ELSERVER,2015.

[31] Özyer, T., Alhajj, R., & Barker, K. (2007). Intrusion detection by integrating boosting genetic fuzzy classifier and datamining criteria for rule pre-screening. Journal of Network and Compute Applications, 30(1), 99–113.