

# Classification Using Flexible Neural Tree

Smrity Gupta\*  
Deepti razdan\*

## *Abstract*

*The purpose of my work is to develop data mining techniques based on flexible neural tree FNT. Based on the pre-defined instruction/operator sets, a flexible neural tree model can be created and evolved.. The FNT structure is developed using genetic programming (GP) and the parameters are optimized by a emetic algorithm (MA). The proposed approach was applied for two real-world problems involving designing intrusion detection system (IDS) and for breast cancer classification. The IDS data has 41 inputs/features and the breast cancer classification problem has 30 inputs/features. The results show that proposed method is efficient for both input feature selection and improved classification rate.*

**Keywords:** *Flexible neural tree model; Genetic programming; Intrusion detection system; Breast cancer classification.*

**1. Introduction:** Variable selection refers to the problem of selecting input variables that are most predictive for a given outcome.. Various data mining techniques have been applied for designing efficient intrusion detection systems because it has the advantage of discovering useful knowledge that describes a user's or program's behavior from large audit data sets. This paper proposes a flexible neural tree (FNT). [5] For selecting the input variables and detection of network intrusions. Based on the pre-defined instruction/operator sets, a FNT model can be created and evolved. FNT allows input variables selection, over-layer connections and different activation functions for different nodes. In our previous work, the hierarchical structure was evolved using probabilistic incremental program evolution algorithm [20] with specific instructions. In this research work, the hierarchical structure is evolved using genetic programming (GP). The fine tuning of the parameters encoded in the structure is accomplished using mimetic algorithm (MA).The proposed method interleaves both optimizations. Starting with random structures and corresponding parameters, it first tries to improve the structure and then as soon as an improved structure is found, it fine tunes its parameters. It then goes back to improving the structure again and, fine tunes the structure and rules' parameters. This loop continues until a satisfactory solution is found or a time limit is reached.

**2. A mimetic algorithm:** MAs are population-based approaches for heuristic search in optimization problems [18]. Basically, they are genetic algorithms that apply a separate local search process to refine individuals. One big difference between memes and genes is that memes are processed and possibly improved by the people that hold them—something that cannot happen to genes. Experimental results show that the MAs have better results over simple genetic or evolutionary Algorithms.

---

\*Department of computer Science and Engineering University of Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal

**2.1. Genetic algorithm:** Genetic algorithm based on the Darwinian survival of the fittest theory, is an efficient and broadly applicable global optimization algorithm [10]. In contrast to conventional search techniques, genetic algorithm starts from a group of points coded as finite length alphabet strings instead of one real parameter set.

**2.2. Local search method:** Local search method is a method of searching a small area around a solution and adopting a better solution if found. The proposed MAs can be described as follows:

S1 Generate an initial GA population;

S2 Evaluate all individuals in the population;

S3 For each individual in the population perform local search on it and Replace it with locally improved solution;

S4 Apply standard genetic algorithm operators to create a new population.

S5 if satisfactory solution is found then stop, otherwise go to step S2.

In this research, the MA is employed to optimize the parameter vector of FNT and the weights and bias of a NN.

**3. Flexible neural tree classifier:** In this research, a tree-structural-based encoding method with specific instruction set is selected for representing a FNT mode 3.2. The optimization of FNT models the optimization of FNT including the tree-structure and parameter optimization. Finding an optimal or near optimal neural tree is formulated as a product of evolution. A number of neural tree variation operators are developed as follows: Mutation: Five different mutation operators were employed to generate offspring from the parents. These

Mutation operators are as follows:

(1) Changing one terminal node: randomly select one

Terminal node in the neural tree and replace it with another terminal node.

(2) Changing all the terminal nodes: select each and every terminal node in the neural tree and replace it with another terminal node.

(3) Growing: select a random leaf in hidden layer of the Neural tree and replace it with a newly generated subtree.

(4) Pruning: randomly select a function node in the neural tree and replace it with a terminal node

(5) Pruning the redundant terminals: if a node has more than 2 terminals, the redundant terminals should be deleted.

Crossover: Select two neural trees randomly and select one non-terminal node in the hidden layer for each neural tree randomly, and then swap the selected sub tree. The Crossover operator is implemented with a pre-defined a probability 0.3 in this study.

Selection: Evolutionary programming (EP) style tournament selection was applied to select the parents for the next generation [4]. Pair wise comparison is conducted for the Union of  $m$  parents and  $m$  offspring's. For each individual, opponents are chosen uniformly at random from all the parents and offspring. For each comparison, if the Individual's fitness is no smaller than the opponent's, it receives a selection. Select  $m$  individuals out of parents and offspring's, that have most wins to form the next enervation. This is repeated for each generation until a predefined number of generations

or when the best structure is found. 3.2.1. Parameter optimization by MA Parameter optimization is achieved by the MA algorithm as described in Section 2. In this stage, the architecture of FNT model is fixed, and it is the best tree developed during the end of run of the structure search. The parameters (weights and flexible activation function parameters) encoded in the best tree formulate an individual. The GA algorithm works as follows:

- (a) Initial population is generated randomly. The learning parameters crossover and mutation probabilities in MA should be assigned in advance.
- (b) The objective function value is calculated for each individual.
- (c) Implementation of the local search, selection, crossover and mutation operators.
- (d) If maximum number of generations is reached or no better parameter vector is found for a significantly long time (100 steps), then stop, otherwise go to step (b)

**4. NN classifier:** A neural network classifier trained by MA with flexible bipolar sigmoid activation functions at hidden layer was constructed. Before describing details of the algorithm for Training NN classifier, the issue of coding is presented. Coding concerns the way the weights and the flexible activation function parameters of NN are represented by individuals or particles. A float point coding scheme is adopted here. For NN coding, suppose there are  $M$  nodes in hidden layer and one node in output layer and  $n$  input variables, then the number of total weights is  $n \times M + 1$ , the number of thresholds is  $M + 1$  and the number of flexible activation function parameters is  $M + 1$ , therefore the total number of free parameters in a NN to be coded is  $n \times M + M + 2M + 1$ . These parameters are coded into an individual or particle orderly. The simple loop of the proposed training algorithm for neural network is as follows.

**S1** Initialization. Initial population is generated randomly. The learning parameters, i.e., crossover and mutation probabilities, should be assigned in advance

**S2** Evaluation. The objective function value is calculated for each individual;

**S3** Implementation of the local search, selection, crossover and mutation operators;

**S4** If maximum number of generations is reached or no better parameter vector is found for a significantly long time (100 steps), then stop, otherwise goto step S2.

**5. Decision tree classification:** For comparison purpose, a decision tree (DT) classification method is also implemented. Feature selection is done based on the contribution the input variables make to the construction of the decision tree. Feature importance is determined by the role of each input variable either as a main splitter or as a surrogate. Surrogate splitters are defined as back-up rules that closely mimic the action of primary splitting rules. Suppose that, in a given model, the algorithm splits data according to variable protocol type and if a value for protocol type is not available, the algorithm might substitute service as a good surrogate. Variable importance, for a particular variable is the sum across all nodes in the tree of the improvement scores that the predictor has when it acts as a primary or surrogate (but not competitor) splitter. Example, for node  $i$ , if the predictor appears as the primary splitter then its contribution towards importance could be given as  $I$  importance. But if the variable appears as the  $n$ th surrogate instead of the primary variable, then the importance becomes  $I$  importance  $\times p^n$  \*  $I$  improvement, in which  $p$  is the surrogate improvement weight which is a user-controlled parameter set between (0-1) [3].

Decision tree induction is one of the classification algorithms in data mining. Classification algorithm is inductively learned to construct a model from the reclassified data set. The inductively learned model of classification algorithm is used to develop IDS 6. Experiment results and analysis .In addition, all experiments were performed using a 2.8GHz processor with 512MB of RAM.

**5.1. Intrusion detection system:** Intrusion detection is classified into two types: misuse intrusion detection and anomaly intrusion detection. Misuse intrusion detection uses well-defined patterns of the attack that exploit weaknesses in system and application software to identify the intrusions. Anomaly intrusion detection identifies deviations from the normal usage behavior patterns to identify the intrusion.

**5.2. Breast cancer classification:** Breast cancer is the most common cancer in women in many countries. Most breast cancers are detected as a lump/mass on the breast, or through self-examination or Mammography screening mammography is the best tool available for detecting cancerous lesions before clinical symptoms appear [13]. Surgery through a biopsy or Lumpectomy has been also the most common methods of removal. Fine needle aspiration (FNA) of breast masses is a cost-effective, non-traumatic, and mostly invasive Diagnostic test that obtains information needed to evaluate malignancy. Recently, a new less invasive technique, which uses super-cooled nitrogen to freeze and shrink a no cancerous tumor and destroy the blood vessels feeding the growth of the tumor, has been developed in the USA. Various artificial intelligence techniques have been used to improve the diagnoses procedures and to aid the physician's efforts

**5.2.1. Data sets:** As a preliminary study, we made use of the Wisconsin breast cancer data set from the UCI machine-learning database repository [16]. This data set has 32 attributes (30 real-valued input features) and 569 instances of which 357 are of benign and 212 are of malignant type. We randomly divided the training and test data sets. The first 285 data is used for training and the remaining 284 data is used for testing the performance of the different models.

**Table 1**  
The important features selected by the FNT algorithm

Attack	class Important variables
Normal	X <sub>3</sub> ; X <sub>11</sub> ; X <sub>21</sub> ; X <sub>40</sub>
Probe	X <sub>1</sub> ; X <sub>3</sub> ; X <sub>12</sub> ; X <sub>18</sub> ; X <sub>20</sub> ; X <sub>21</sub> ; X <sub>23</sub> ; X <sub>26</sub> ; X <sub>27</sub> ; X <sub>31</sub> ; X <sub>37</sub> ; X <sub>41</sub>
DOS	X <sub>1</sub> ; X <sub>8</sub> ; X <sub>10</sub> ; X <sub>11</sub> ; X <sub>16</sub> ; X <sub>17</sub> ; X <sub>20</sub> ; X <sub>12</sub> ; X <sub>23</sub> ; X <sub>28</sub> ; X <sub>29</sub> ; X <sub>31</sub>

**Table 2**

Detection performance using FNT, NN and DT classification models

Attack class	FNT (%)	NN (%)	DT (%)
Normal	99.19	95.69	82.32
Probe	98.39	95.53	94.83
DOS	98.75	90.41	77.10

**Table 3**

Comparison of false positive rate (fp) and true positive rate (tp) for FNT, NN and DT classifiers

Cancer type	FNT		NN		DT	
	fp%	tp%	fp%	tp%	fp%	tp%
Normal	0.98	65.00	4.388	70	29.66	99.60
Probe	0.18	55.86	0.40	37.15	0.24	31.00
DOS	1.59	98.98	3.68	89.38	72.10	97.9%

**6. Conclusion:** In this paper, we presented a flexible neural tree (FNT) model for intrusion detection systems (IDS) and breast cancer classification with a focus on improving the detection/classification performance by reducing the input features.

### 7. References :

1. Abraham, Evolutionary computation in intelligent web management. Evolutionary computing in data mining, in: A. Ghosh, L. Jain (Eds.), Studies in Fuzziness and Soft Computing, Springer, Germany, 2004, pp. 189–210 (Chapter 8).
2. L. Breiman, J. Friedman, R. Olshen, C. Stone, (1984) Classification of Regression Trees, Wadsworth Inc., Belmont, CA, 1984.
3. S. Chebrolu, A. Abraham, J.P. Thomas, (2005) Feature Deduction and Ensemble Design of Intrusion Detection Systems, Computers and Security, vol. 24/4, Elsevier, Amsterdam, 2005, pp. 295–307.
4. K. Chellapilla, (1997) Evolving computer programs without subtree crossover, IEEE Trans. Evol. Comput. 1 (1997) 209–216.
5. KDD cup 99 data set, <http://kdd.ics.uci.edu/database/kddcup99/>. [16] J. Merz, P.M. Murphy, UCI repository of machine learning databases, <http://www.ics.uci.edu/~earn/MLRepository.html>, 1996.
6. S. Morio, S. Kawahara, N. Okamoto, T. Suzuki, T. Okamoto, M. Haradas, A. Shimizu, (1989) An expert system for early detection of cancer of the breast, Comput. Biol. Med. 19 (5) (1989) 295–305.

7. P. Moscato, (1989) On evolution, search, optimization genetic algorithms and martial arts: towards memetic algorithms, Caltech Concurrent Computation Program, C3P Report 826.
8. S. Mukkamala, A.H. Sung, A. Abraham,(2005)Intrusion detection using an ensemble of intelligent paradigms, J. Network Comput. Appl. 28 (2) 167–182.
9. R.P. Salustowicz, J. Schmidhuber, Probabilistic incremental program evolution, *Evol. Comput.* 2 (5) (1997) 123–141.
10. K. Shah, N. Dave, S. Chavan, S. Mukherjee, A. Abraham, S. Sanyal, (2004) Adaptive Neuro-Fuzzy Intrusion Detection System, in: IEEE International Conference on ITCC'04, vol. 1, 2004, pp. 70–74.
11. The Weekend Australia, Health Section, July 13–14, 2002, 7pp.