International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2016): 79.57 | Impact Factor (2017): 7.296

Simulated Relief Algorithm for Feature Selection Approach

S. Francisca Rosario[1], Dr. K. Thangadurai[2]

[1]Research Scholar, St.Peter's University, Chennai, India

[2]Head, PG & Research department of Computer Science, Government Arts College(Autonomous), Karur-639 005, TamilNadu, India

Abstract: ***Feature selection is an important preprocessing technique in data mining and it is the process of selecting the relevant features from the data sets. The objective of the feature selection techniques are to reduce the number of features and to improve the classification accuracy. Three contributions such as sequential backward selection algorithm, Relief algorithm and simulated annealing algorithm are combined and a new novel algorithm known as Simulated Relief is proposed in this paper. The efficiency and the effectiveness of the proposed algorithm is evaluated with cotton data set provided by central cotton Research station at Coimbatore. Weka tool and Microsoft excel sheet contribute the data manipulation task for computation process. The experimental study concludes that the Simulated Relief algorithm using Multilayer perceptron classifier provides higher classification accuracy than using the Naïve bayes classifier.***

Keywords: Feature selection, Relief, attribute, classification, Naïve Bayes, Multilayer perceptron, Weka, instances, Simulated Relief

1. Introduction

Data mining is the process of analyzing data from different perspectives and summarizing it into useful information. Classification is a data mining function that assigns items in a collection of target categories or classes. The goal of classification is to accurately predict the target class for each case in the data. Feature subset selection is a technique for reducing the attribute space of a feature subset by removing irrelevant or redundant attributes as possible. This operation reduces the dimensionality of the data sets, which in turn to allow the learning algorithms to work faster and more effectively. The prime objective of the feature selection approach is machine learning as well as data mining with a minimum feature to get maximum accuracy. A good feature set contains a highly relevant feature which helps to improve the efficiency of the classification algorithms and to classify accurately. In the past two decades, it had been observed that a tremendous growth in the field of data mining regarding both numbers of instances and number of features. This growth causes serious problems to many existing data mining algorithms. Data mining applications consist of the high dimensionality of data which contain many inappropriate features. Feature selection is an important and frequently used technique in data mining for dimensionality reduction by removing irrelevant, redundant and noisy features. It brings the immediate effects of speeding up of data mining algorithms by selecting the relevant features and improving classification accuracy. The past literature showed that various research works were carried out to select the most relevant features and to improve the classification accuracy, but still the problems persist. Hence, a new methodology known as Simulated Relief is proposed in this paper.

2. Literature Review

Kira and Rendell proposed the Relief Algorithm. The statistical method is used in Relief instead of Heuristic search. Relief requires linear time in the number of given features and number of training instances regardless of the target concept to be learned. It selects the statistically relevant features (*Kira et al.*, 1992). The Euclidean Based Feature Selection algorithm (EUBAFES) weights and selects features similarly to the Relief algorithm. It is also a distance-based approach that reinforces the similarities between instances that belong to the same class while deteriorating similarities between instances in different classes. A gradient descent approach is employed to optimize feature weights on this goal (*Scherf et al*., 1997).Relief is considered as one of the most successful algorithms for assessing the quality of features due to its simplicity and effectiveness (*Dietterich*, 1997). The Relief algorithms are a family of attribute weighting algorithms that can efficiently identify associations between attributes and the class even if the attributes have nonlinear interactions without significant main effects (*Kira et al*., 1992) (*Dietterich*, 1997). Relief was extended to handle noisy and missing data (Kononeko, 1994).Kirkpatrick realized the similarity between the optimization of combinational optimization problems and the physical process of annealing. Simulated Annealing became one of the more popular optimization algorithms (Kirkpatrick, 1983). Sullivan and Jacobson studied generalized hill climbing algorithms and their performance. They extended necessary and sufficient convergence conditions for Simulated Annealing (*Sullivan et al*., 2001). Nader Azizi and Zolfaghari addressed changes in temperature based on the number of consecutive moves showing improvement by comparing two variations of the SA method in adaptive temperature control (*Nader Azizi et al.*, 2004).The Naive Bayes classifier is a straightforward probabilistic classifier stand on applying Bayes theorem with strong naive independence assumptions. A more expressive term for the underlying probability model would be “independent feature model." An inclusive comparison with other classification algorithms in 2006 showed that Bayes classification is outperformed by other approaches, such as boosted trees or random forests (*Caruna et al*., 2006), (*Manikandan et al.*, 2014).The J48 algorithm builds the decision tree from labeled training data set using information gain, and it examines the same that results from choosing an attribute for splitting the data. The measure to compare the difference of impurity degrees is called information gain. The attribute with highest normalized information gain is used to make the decision. Then the algorithm recurs on smaller subsets. The splitting procedure stops if all instances in a subset belong to the same class. Then the leaf node is created in a decision tree telling to choose that class (*Trilok et al.,* 2013) (*Nurul Amin et al.,* 2015).Multilayer Perceptron classifiers are universal function approximators, and they can be used to create mathematical models by regression analysis *(Cybeako,* 1989) (*Nurul Amin et al.,* 2015). Powers and David describe the systematic analysis of performance measures for classification tasks regarding Precision, Recall and F-measure (*Powers et al.,* 2011).

3. Feature Selection Algorithms

3.1 Relief algorithm

Relief algorithm was proposed by Kira and Rendell in the year 1992. According to Kira and Rendell (*Kira et al.,* 1992b, *Kira et al.,* 1992a) this algorithm weights each feature according to its relevance to the class. Initially, all weights are set to zero and then updated iteratively. In each iteration, this non-deterministic algorithm chooses a random instance i in the dataset and estimates how well each feature value of this instance distinguishes between instances close to i. In this process two groups of instances are selected: some closest instances belonging to the same class and some belonging to a different class. With these instances, Relief will iteratively update the weight of each feature, and it differentiates data points from different classes while, simultaneously, recognizing data points from the same class. In the end, a certain number of features with the highest weights are selected. In an alternative version, a threshold may be used in such a way that only the features with weights above this value are selected. The output of the Relief algorithm is a weight between −1 and 1 for each attribute, with more positive weights indicating more predictive attributes. The weight of an attribute is updated iteratively as follows. A sample is selected from the data, and the nearest neighboring sample that belongs to the same class (nearest hit) and the nearest neighboring sample that belongs to the opposite class (nearest miss) are identified. A change in attribute value accompanied by a change in class leads up to the weighting of the attribute based on the intuition that the attribute change could be responsible for the class change. On the other hand, a change in attribute value accompanied by no change in class leads to down-weighting of the attribute based on the observation that the attribute change had no effect on the class. This procedure of updating the weight of the attribute is performed for a random set of samples in the data or every sample in the data. The weight updates are then averaged so that the final weight is in the range [−1, 1]. The attribute weight estimated by Relief has a probabilistic interpretation. It is proportional to the difference between two conditional probabilities, namely, the probability of the attribute's value being differently conditioned on the given nearest miss and nearest hit respectively (Robnik Sikonjam et al., 2003).

Relief Algorithm

```
Set W[a] = 0 for each attribute a
for i = 1 to n do
select sample si from data at random
find nearest hit sh and nearest miss sm
for each attribute a do
W[a] = W[a] + Wi[a]
end for
end for
for each attribute a do
W[a] = W[a] / n
end for
where diff(a, si, sj) = 0, if si[a] = sj[a]
                      = 1, if si[a] ≠ sj[a]
```

3.2 Simulated Annealing algorithm

The Simulated Annealing algorithm was originally inspired by the process of annealing in metal work. Annealing involves in heating and cooling a material to alter its physical properties due to the changes in its internal structure. This gradual 'cooling' process is what makes the Simulated Annealing algorithm remarkably effective at finding a close to the optimum solution when dealing with large problems which contain numerous local optimums. To apply Simulated Annealing, one must specify three parameters. First is an annealing schedule, which consists of an initial and final temperature, T_0 and T_{final}, along with an annealing (cooling) constant ΔT. Together these govern how the search and proceed until the search stops. The second parameter is a function used to evaluate potential solutions (feature subsets). The goal of Simulated Annealing is to optimize this function. For this discussion, the mean squared error is used to estimate the function. The final parameter for Simulated Annealing is a neighbor function, which takes the current solution and temperature as an input, and returns a new nearby solution. The role of the temperature is to govern the size of the neighborhood. At high temperature the neighborhood should be large, allowing the algorithm to explore broadly. At low temperature, the neighborhood should be small, forcing the algorithm to explore locally. For example, one represents the set of available features as a bit vector, such that each bit indicates the presence or absence of a particular feature. This algorithm attempts to iteratively improve a randomly generated initial solution. On each iteration, the algorithm generates a neighboring solution and computes the difference in quality (energy, by analogy to metallurgy process) between the current and candidate solutions. If the new solution is better, then it is retained. Otherwise, the new solution is retained with a probability that is dependent on the quality difference, ΔE, and the temperature. The temperature is then reduced for the next iteration. Success in Simulated Annealing epends heavily on the choice of the annealing schedule. One obvious criterion is to accept a solution when it has a less error than the previous solution. The probability of occurrence of a perturbed solution is computed by Metropolis algorithm shown by the following expression

$$\mathrm{Exp}\left(\frac{-\Delta E}{TK}\right) \qquad \text{... (3.2.1)}$$

Where ΔE is the difference between the solution error after it has perturbed, and the solution error before it was perturbed. T is the current temperature, and k is a suitable constant. From the metropolis algorithm it can be observed that when ΔE is negative, the solution is always accepted. However, the algorithm may accept a new solution, if the solution has not a smaller error than the previous one (a positive ΔE) and the probability of doing this decreases when the temperature decreases or when ΔE increases. If the metropolis algorithm takes the value in between 0.7 and 0.9, the new solution will be accepted, and otherwise, the new solution will not be accepted. An estimate for mean squared error which is represented by ΔE can be computed from $\Delta E = \sigma^2/n$. The initial value of T_k is taken as 0.95, and the value of k is a random number between 0 and 1. In the successive iterations the value of T will be taken as $T_k + 1 = \alpha \times T_k$, $0 < \alpha < 1$ where $\alpha = 0.5$. In the context of feature selection, relevant evaluation functions include the accuracy of a given learning algorithm using the current feature subset (creating a wrapper algorithm), or a variety of statistical scores (producing a filter algorithm). If ΔT is too large (near one), the temperature decreases slowly, resulting in slow convergence. If ΔT is too small (near zero), then the temperature decreases quickly and convergence will likely to reach a local extreme. Moreover, the range of temperatures used for an application of Simulated Annealing must be scaled to control the probability of accepting a low-quality candidate solution.

Simulated Annealing algorithm

Examples X =< x1; y1 >,... < xm; ym >
Annealing schedule, T0; T_{final} and ΔT with 0 < ΔT < 1
Feature subset evaluation function Eval (. , .)
Feature subset neighbour function Neighbour (. ; .)
Algorithm:
S_{best} random feature subset
while $T_i > T_{final}$ do
$S_i \leftarrow$ Neighbor (S_{best} ; T_i)
$\Delta E \leftarrow$ Eval(S_{best} ; X) - Eval (S_i ; X)
if ΔE < 0 then //if new subset better
$S_{best} \leftarrow$ Si
else //if new subset worse
$S_{best} \leftarrow$Si with probability exp ($\Delta E/T_i$)
T_i+1 $\leftarrow \Delta T \times T_i$
return(S_{best})

3.3 Sequential Backward Selection Algorithm

Sequential Backward Selection (SBS) starts with all features and iteratively remove a single feature to increase the classification accuracy. Although the combination of features is taken into account with this technique, a high number of computations are necessary since it starts with the set of all features. This may not be feasible for the very high dimensional data set. Starting from the full set, sequential backward selection algorithm removes the feature X^- that results in the smallest decrease in the value of the objective function. Mean value is considered as the potential function for the sequential backward selection algorithm and the attribute with least mean value should be removed in each iteration of the Simulated Annealing algorithm in the neighbourhood generation process.

Algorithm

1) Start with the full set Y0=X
2) Remove the worst feature X-=argmax[J(Y_k-X)]; x Y_k
3) Update $Y_{k+1}=Y_k$- X-; k=k+1
4) Go to 2

4. Classifiers

4.1 Multi-Layer Perceptron (MLP)

The simplest form of neural network needs to classify linearly separable patterns. While for non-linear patterns multi-layer.Perceptron neural network model performs well. It maps set of input data onto a set of appropriate outputs. Multi-Layer Perceptron consists of multiple layers of nodes in a directed graph with each layer fully connected to the next one. Except for the input nodes, each node is a neuron (or processing element) with a non-linear Activation function. Multi-Layer Perceptron uses back propagation learning algorithm for training and widely used in pattern classification and recognition. The simplest form of MLP is shown in fig. 4.1

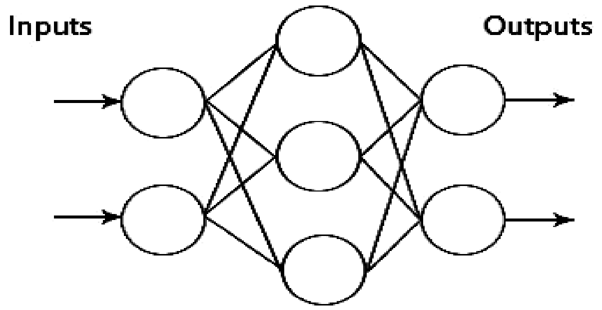

Figure 4.1: Multilayer Perceptron

Multi-layer Perceptron is a supervised learning algorithm that learns a function $f(\cdot): R^m \rightarrow R^0$ by training on a dataset, where m is the number of dimensions for input and *o* is the number of dimensions for output. Given a set of features $X = x_1, x_2 ... x_m$ and a target *y*, it can learn a non-linear function approximator for either classification or regression. It is different from logistic regression, in that between the input and the output layer, there can be one or more non-linear layers, called hidden layers. The leftmost layer, known as the input layer, consists of a set of neurons $\{x_i \mid x_1, x_2 ... x_m\}$ representing the input features. Each neuron in the hidden layer transforms the values from the previous layer with a weighted linear summation $w_1x_1 + w_2x_2 + ... + w_mx_m$, followed by a non-linear activation function $g(\cdot): R \rightarrow R$- like the hyperbolic tan function. The output layer receives the values from the last hidden layer and transforms them into output values.

4.2 Naive Bayes

The Naive Bayes classifier is based on Baye's Theorem with independent assumptions between predictors. Naive Bayesian model is easy to build without complicated iterative parameter estimation. It analyzes all the attributes in the data individually, means the value of a predictor (X) on a given (C) is independent of the values of other predictors. This assumption is called class conditional independence. The working steps for Naive Baye's classifier are as follows.

1. First calculate the posterior probability and construct the frequency table against the target
2. Transforming the frequency table into likelihood table and using the Naive Baye's equation to calculate the posterior probability for each class
3. Class with highest probability is the outcome of prediction

   P (C/X) = P (X/C) * P (C) / P (X) (4.2.1)

   P (C/X) is posterior probability of class (target) given predictor (attribute)

   P (X/C) is likelihood which is the probability of predictor given class

   P (C) is prior probability of the class

   P (X) is prior probability of predictor

5. Simulated Relief Algorithm

As the Relief algorithm estimates the weight of feature by selecting the instances randomly, the weight estimation of the features is uncertain. Also, the chance to select the irrelevant attributes may happen. Since this algorithm selects the instances randomly for weight calculation, there is a possibility of relevant features become irrelevant. Because of the randomicity and the uncertainty of the instances used for calculating the feature weight vector in the Relief algorithm, the results will fluctuate with the instances, which lead to poor evaluation accuracy. To overcome this issue, and to reduce the computation time of the classification task, a new algorithm known as Simulated Relief has been proposed. This new algorithm introduces the Simulated Annealing algorithm to select the feature subset in incporpratimg the Relief algorithm. While using Simulated Annealing algorithm for feature subset generation, sequential backward selection strategy is applied to reduce the features at each iteration. Simulated Annealing algorithm takes Metropolis algorithm for subset selection which makes the result more stable and accurate. The new Simulated Relief algorithm estimates the weight of the features by implementing the Relief algorithm and it identifies the individual weight of each feature and ranks it according to the weight with its own merits and demerits. Among many variants in Relief algorithm, Euclidean distance formula is used to estimate the weight and it ranks the features. The weight of the attributes above a threshold value may be taken as the selected attributes. A threshold value may be taken up by arranging the weight of attributes in ascending order and considering the weight of attribute which is in the middle position of the sorted order. Weights of the attribute which are below the threshold value are rejected, and those values which are above the threshold may be considered as the selected attributes in the feature subset. The new feature subset of selected attributes are proceeded to perform the classification task by the Naive Bayes and Multilayer Perceptron classifiers. Then the classification accuracy can be measured with the help of accuracy evaluation measures such as precision, Recall and Fmeasure. Figure 5.1 represents the work flow of Simulated Relief algorithm.

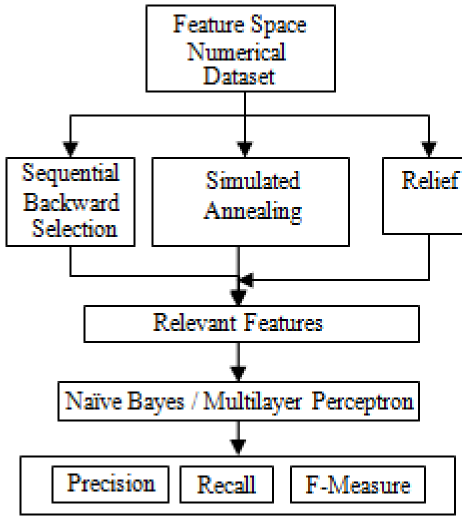

Figure 5.1: Work flow of Simulated Relief Algorithm

Simulated Relief Algorithm

- $S_{best} \leftarrow$ full set = Y_k = X
- Annealing Schedule

T_0 = Initial Temperature = 0.95

T_{final} = Final Temperature = 0, and ΔT with $0 < \Delta T < 1$

- Feature subset evaluation Eval(. , .)
- Feature Subset neighbor function Neighbor (. , .)

1. $S_{best} \leftarrow$ full set = Y_k = X, Initial k value = 0
2. $S_i \leftarrow$ Neighbor (S_{best}, T_i)
3. while $T_i > T_{final}$ do
4. Remove the worst feature $X^- = (X \in Y_k \text{ argmax } [j(Y_k - X)]$
5. Update $Y_k = Y_k - X^-$
6. $S_i \leftarrow Y_k$
7. $\Delta E \; \Delta$ Eval (S_{best}, X) – Eval(S_i, X)
8. If $\Delta E < 0$ then
9. $S_{best} \leftarrow S_i$ // if new subset better
10. else
11. $S_{best} \leftarrow S_i$ with probability $\exp\left(\frac{-\Delta E}{TK}\right)$
12. $T_{i+1} \leftarrow \Delta T \times T_i$
13. k = k + 1
14. Go to step-3
15. End while
16. return (S_{best})
17. Relief (S_{best}, m, τ)
18. Separate S_{best} in to
    S^+ = Positive instances and
    S^- = Negative instances
19. W = (0, 0, ..., 0)
20. for i = 1 to m
21. Pick at random an instances $X \in S_{best}$
22. Pick at random one of the positive instances closed to X, $Z^+ \in S^+$,
23. Pick at random one of the negative instances closes to X, $Z^- \in S^-$
24. If X is a positive instance then nearhit = Z^+, nearmiss = Z^-,
25. else
26. nearhit = Z^-, nearmiss = Z^+
27. Update – weight(W, X, nearhit, nearmiss)

28. Relevance = $\left(\frac{1}{m}\right)W$
29. for i = 1 to P
30. if relevance i $\geq \tau$
31. then f_i is a relevant feature
32. else f_i is an irrelevant feature
33. update weight (W, X, nearhit, nearmiss)
34. for i = 1 to P
35. $W_i = W_i - diff(X_i, nearhit_i)^2 + diff(X_i, nearmiss_i)^2$

End

6. Data Source

To evaluate and analyze classification accuracy of the algorithms agriculture data has been taken up. The efficiency and effectiveness of the proposed algorithm are evaluated cotton data set provided by cotton research station at Coimbatore. This data deals with the classification of two pests namely Mirid bug and Mealy bug which affects the cotton plant. This two-class classification technique is implemented in the cotton data set, and the accuracy of the classification has been measured regarding the evaluation measures such as Precision, Recall, and F-measure. The computations of the sequential backward selection algorithm and the Simulated Annealing algorithm are carried out in the Microsoft Excel sheet. Ranking of attributes, subset selection, classification of the subsets and measuring the classification accuracy of the selected subsets can be carried out by the Weka tool

7. Experimental Results

The data set contains one hundred and fifty six instances and thirteen attributes such as crop, location, pest, observation, standard week, maximum temperature, minimum temperature, relative humidity1, relative humidity2, rainfall, wind speed, sunshine hours and evaporation. Sequential Backward Selection strategy computes the mean value of the features to generate the neighborhood solution in the Simulated Annealing process. The graphical representation of mean values of the attribute is depicted in Figure 7.1

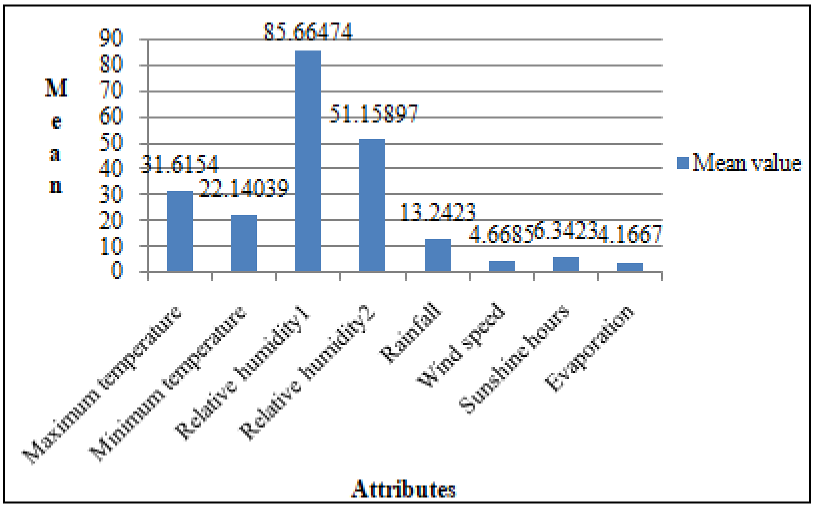

Figure 7.1: Mean Values of Attributes

The mean squared error value of each attribute is computed to implement the Metropolis algorithm in the subset evaluation process of the Simulated Annealing algorithm. The computations of the mean value, mean squared error value and the Metropolis algorithm are carried out through the Microsoft Excel sheet. Fig.7.2 shows the mean squared error value of the attributes.

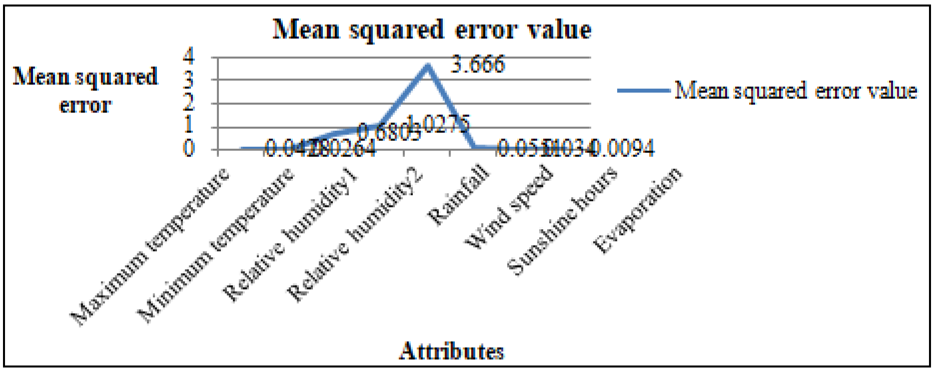

Figure 7.2: Graphical Representation of Mean Squared Error Values of Attributes

The result of the Simulated Annealing algorithm generates the feature subset that contains attributes such as, crop, location, pest, observation, standard week, maximum temperature, minimum temperature, relative humidity1, relative humidity2, rainfall, wind speed and sunshine hours. This attribute subset has been generated after the fifth iteration of the Simulated Annealing process. This new feature subset is passed as input to the Relief algorithm to estimate the weight of the attributes. Estimating the weight of features is carried out by the Relief algorithm through the Weka tool. Figure 7.3 depicts the graphical representation of the weight of the attributes.

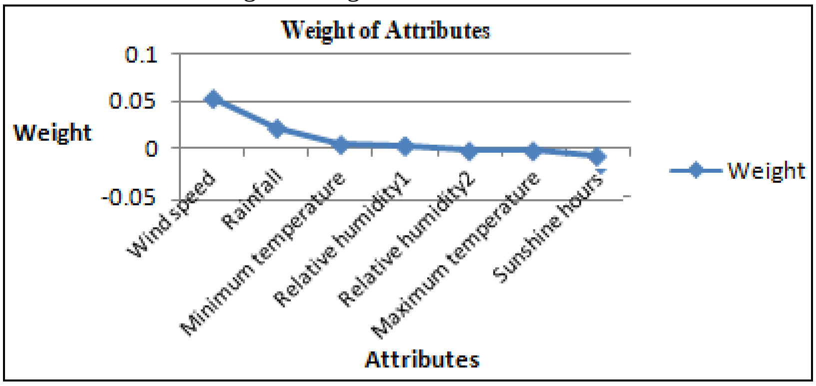

Figure 7.3: Graphical Representation of the Weight of the Attributes

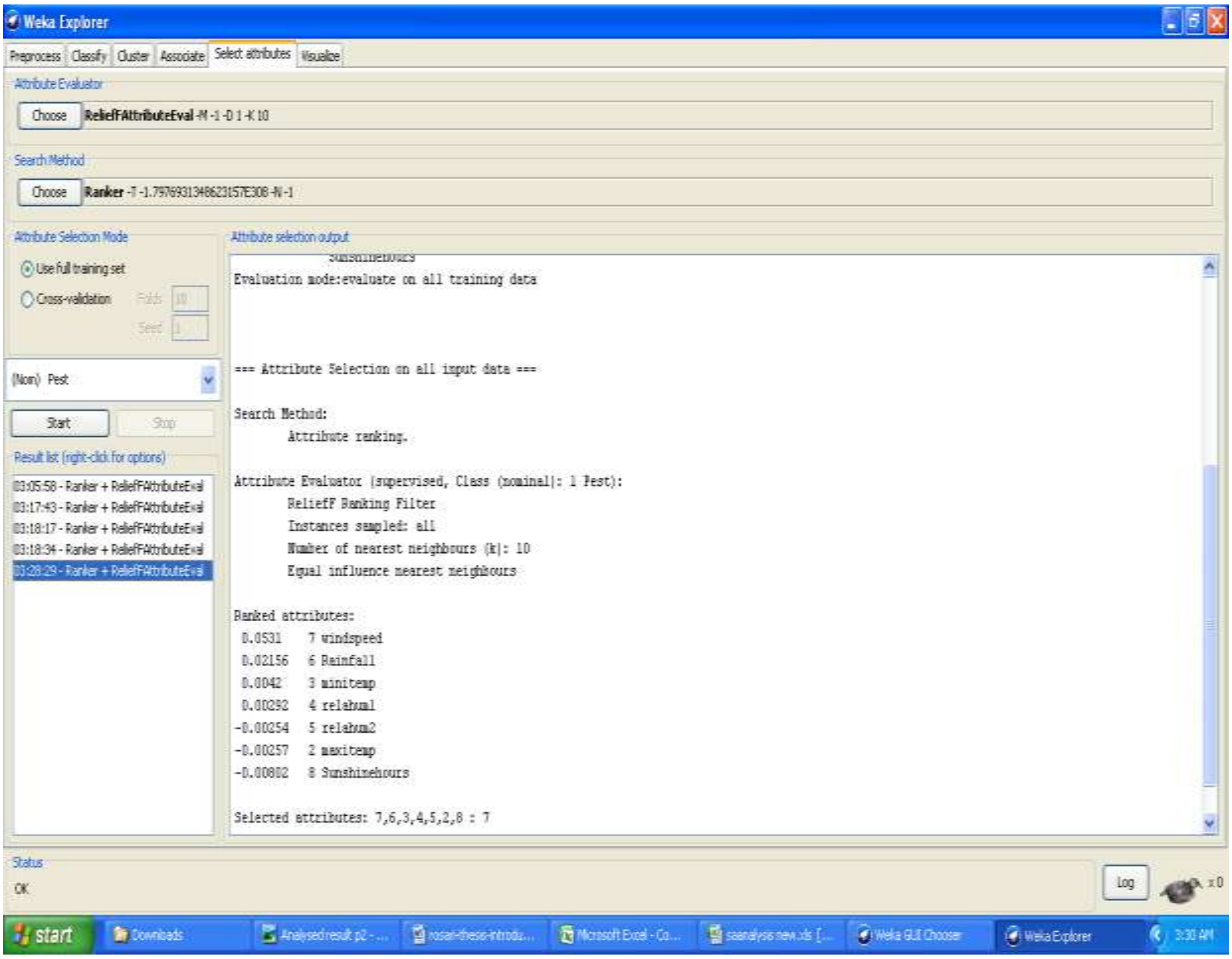

Figure 7.4: Screen Shot of the Ranked Attributes in the Weka Explorer

Figure 7.4 shows the output of the ranked attributes in the Weka Explorer screen. In this experimental study, the attribute relahum1 ranks in the middle position and its weight is 0.00292. So it is the threshold value. The attributes that have the weight above the threshold value are considered as the selected feature subset. Hence the attributes wind speed, rainfall, and minimum temperatures are selected as feature subset in the output of simulated Relief algorithm. Then the classification task is carried out by the Naïve Bayes classifier through the Weka tool. The result of the experiment shows that seventy six instances are classified as Mirid bug, and eighty instances are classified as Mealy bug. Table 7.1 shows the true positive, false positive, precision, recall, fmeasure, and Receivers Operating Characteristics curve area values, which are the representations of classification accuracy of pests namely Mirid bug and Mealy bug. It also presents the weighted average value of the accuracy evaluation measures of the classification task.

Table 7.1: Evaluation Measures of Naïve Bayes Classifier

True positive	*False Positive*	*Precision*	*Recall*	*Fmeasure*	*Roc area*	*Class*
0.769	0.346	0.526	0.769	0.625	0.77	Mirid bug
0.654	0.231	0.85	0.654	0.739	0.77	Mealy bug
0.692	0.269	0.742	0.692	0.701	0.77	Weighted average

The result of classification accuracy for two pests namely Mirid bug and Mealy bug are shown as a screen shot in Figure 7.5 in the Weka Explorer

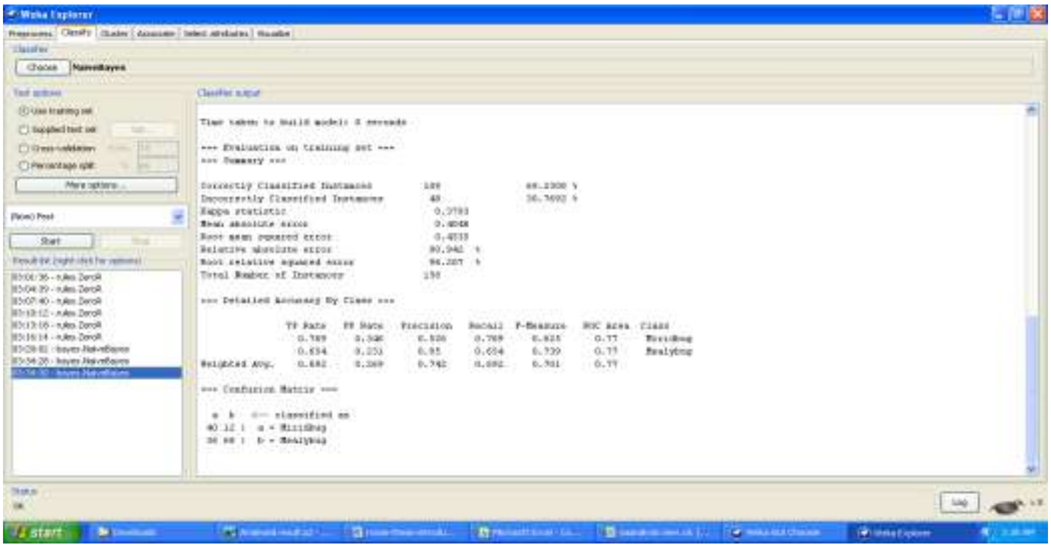

Figure 7.5: Screen shot of Weka Explorer Representing Classification Accuracy of Naïve Bayes Classifier

The classification task in the proposed Simulated Annealing algorithm is also carried out by multilayer Perceptron algorithm which is a function based classifier. Multilayer Perceptron classifier classifies fifty four instances as Mirid bug and one hundred and two instances as Mealy bug. Table 7.2 shows the classification accuracy of the proposed simulated Relief algorithm using the multilayer Perceptron classifier.

Table 7.2: Evaluation Measures of the Multilayer Perceptron classifier

True positive	False Positive	Precision	Recall	Fmeasure	Roc area	Class
0.923	0.385	0.545	0.923	0.686	0.833	Mirid bug
0.615	0.77	0.941	0.615	0.744	0.833	Mealy bug
0.718	0.179	0.809	0.718	0.725	0.833	Weighted average

Figure 7.6 depicts the Weka screen which represents the classification evaluation measures of the proposed simulated Relief using the multilayer Perceptron classifier.

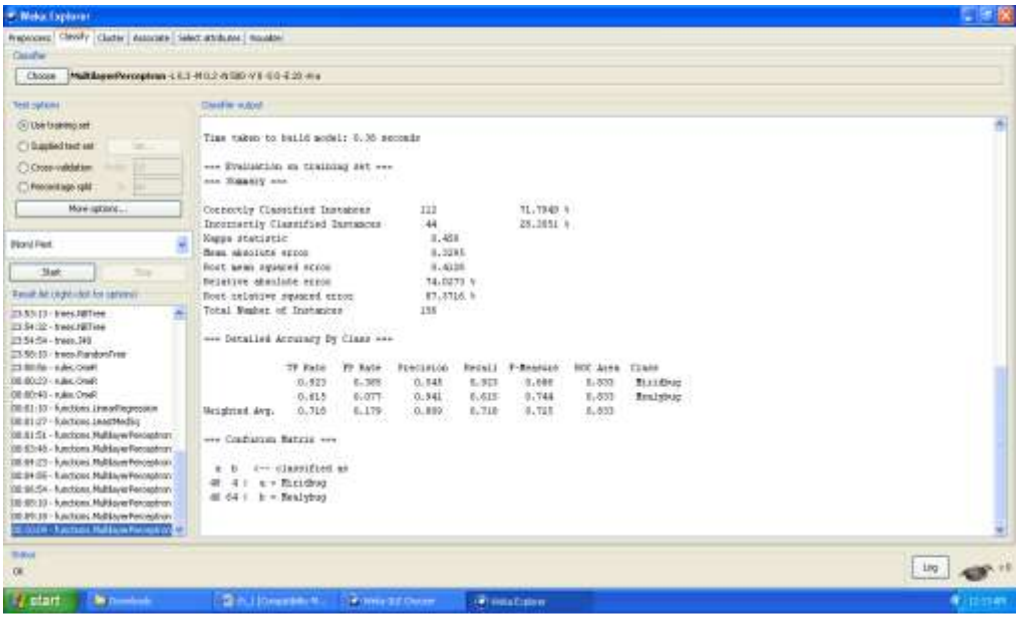

Figure 7.6: Classification Accuracy of Simulated Relief using Multilayer Perceptron Classifier

The weighted average values of the classification evaluation measures for the proposed simulated Relief algorithm is compared with the two classifiers namely Naïve Bayes and Multilayer Perceptron, and the result of the comparison shows that the multilayer Perceptron classifier produces better results than the Naïve Bayes classifier. Table 7.3 describes the comparison of Naïve Bayes and Multilayer perceptron classifiers in the simulated Relief algorithm.

Table 7.3 Comparison of the Naïve Bayes and Multilayer Perceptron Classifiers in the Simulated Relief Algorithm

Algorithm	Classifier	TP	FP	Precision	Recall	Fmeasure	Roc area
Simulated Relief	Naïve Bayes	0.692	0.269	0.742	0.692	0.701	0.77
	Multilayer Perceptron	0.718	0.179	0.809	0.718	0.725	0.833

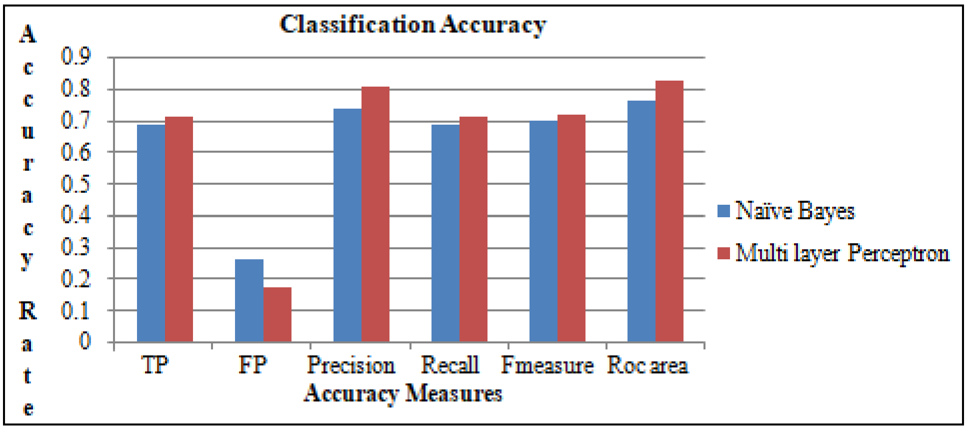

Figure 7.7: Comparisons of Naïve Bayes and Multilayer Perceptron

Classifiers in Simulated Relief Algorithm

Figure 7.7 presents the chart which represents the comparison of the Naïve Bayes and the Multilayer Algorithms for classification.

8. Conclusion

This research contributes the comparative study of the classification results of Naïve Bayes and Multilayer Perceptron classifiers in Simulated Relief algorithm and it concludes that the Multilayer Perceptron classifier produces higher classification accuracy than the Naïve Bayes classifier.

References

[1] Azizi, Nader. and Zolfaghari, Saeed. "Adaptive Temp erature Control For Simulated Annealing: A Comparative Study", Published in Journal "Computers and Operations Research, Vol. 31, No. 14 , pp. 2439-2451, 2004.
[2] Cybeako, G. "Approximation by Superpositions of a S igmoidal Function Mathematics of Control, Signals, And Systems", Vol. 2, No.4, pp.303-314, 1989.
[3] Dietterich, T.G. "Machine Learning Research: Four C urrent Directions", AI Magazine, Vol.18, No.4, pp. 97-136, 1997
[4] Gelfand, S.B. and Mitter, S.K. "Simulated Annealing Type Algorithms for
[5] Kira, K. and Rendell, L.A. "A Practical Approach to Feature Selection" In Proceedings of the 9th International Conference on Machine Learning. Morgan Kaufmann, pp. 249-256, 1992
[6] Kirkpatrick, S., Gelatt, C.D. and Vecchi, M.P. "Opt imization by Simulated Annealing Science", pp. 671-680, 1983.
[7] Kononenko, I. "Estimating attributes: Analysis and Extensions of RELIEF" Proceedings of the European Conference on Machine Learning, pp. 171-182, 1994
[8] Kul, S. Nakarlya and Casasent, DF. "An Improvement on Floating Search Algorithms for Feature Subset Selection", Pa ttern Recog Vol. 42: pp. 1932-40, 2009.
[9] Kulkarni, Pallavi. and Ade, Roshani. "Prediction of Student's Performa nce based on Incremental Learning", International Journ al of Computer Applications, Vol.99, No.14, 2014.
[10] Kumar, V. and Minz, S. "Mood Classification of Lyri cs using SentiWordNet", in Proc. of ICCCI-2013, 2013.
[11] Kumar, V. and Minz, S. "Multi-view Ensemble Learnin g for Poem Data Classification using SentiWordNet", in Proc. of 2nd International Conference on Advanced Computing, Networking, and Informatics (ICACNI-2014), 2014.
[12] Kumar, V. and Minz, S. "Poem Classification using Machine Learning Approach," International Conference on Soft Computi ng for problem Solving (SocPro 2012*)*, 2012.
[13] Ladha. and Deepa. "Feature Selection Methods and Al gorithms", International Journal on Computer Science and Engineering, Vol. 3, No.5, pp.1787-1797, 2011.
[14] Manikandan, P. and Ramyachitra, D. "Naive Bayes Cla ssification Technique for Analysis of Ecoli Imbalance Dataset", Internati onal Journal of Computational Intelligence and Informatics, Vol. 4, No. 2, 2014.
[15] Multivariate Optimization Algorithmica", Vol. 6 , pp. 419-436, 1991a.
[16] Powers. and David, M.W. "Evaluation: From Precision , Recall and F-Measure TOROC, Informedness,

Markedness and Correlation", Journal of Machine Learning Technologies, Vol. 2, No. 1, pp. 37-63, 2011.

[17] Scherf, M. and Brauer, W. "Feature Selection By Mea ns of a Feature Weighting Approach", 1997.

[18] Sharma, Trilok Chand. and Jain, Manoj. "Weka Approa ch for Comparative Study of Classification Algorithm", International J ournal of Advanced Research in Computer and Communication Engineering" , Vol. 2, pp.1925-1931, 2013

[19] Sikonjam, Robnik. and Kononenko, I. "Theoretical an d Empirical Analysis of Relief and ReliefF", Machine Learning, Vol. 53, No.1, pp. 23-69, 2003

[20] Sikonjam, Robnik. and Kononenko, I. "Theoretical an d Empirical Analysis of Relief and ReliefF", Machine Learning, Vol. 53, No.1, pp. 23-69, 2003

[21] Sullivan, K.A. and Jacobson, S.H. " A Convergence Analysis of Generalized Hill Climbing Algorithms", IEEE Transactions on Aut omatic Control, Vol. 46, pp. 1288-1293, 2001.