I. JULIE
Department of Computer Science
Arignar Anna Government Arts College
Musiri– 621 201

Dr. E. KIRUBAKARAN
Additional General Manager
Outsourcing, BHEL
Trichy – 620 014

Abstract: The most important application of Microarray for gene expression analysis is used to discover or classify the unknown tissue samples with the help of known tissue samples. Several Data Mining Classifiers have been proposed recently to predict/identify the cancer patterns. In this research work, we have focused and studied a few Classification Techniques such as Support Vector Machine (SVM), Nearest Neighbor Classifier (k-NN), ICS4, Non-Parallel Plane Proximal Classifier (NPPC), NPPC-SVM and Margin-based Feature Elimination-SVM (MFE-SVM). The performances of these classifiers have been analysed in terms of Threshold Level, Execution Time, Memory Usage and Memory Utilization. From our experimental results, we revealed that the Threshold level and Execution Time to predict the Cancer Patterns are different for different Classifiers. Our experimental results established that among the above identified classifiers, the k-NN classifier achieves less Threshold to predict the cancer pattern, but however it consumes more execution time, whereas the MFE-SVM achieves less execution time to predict the cancer pattern, but it still needs more threshold to predict the Pattern. That is to find the best single classifier in terms of Threshold and Execution Time is still complicated. To address this major issue, we had proposed an efficient Classifier called Maximizing Feature Elimination Technique based Hybrid Classifier (MFE-HC), which is the combination of both k-NN and SVM classifiers. From the results, it was established that our proposed work performs better than both the k-NN and MFE-SVM Classifiers in terms of Threshold and Execution Time. However, from our experimental results, we revealed our MFE-HC consumes more execution time when the Database is larger one. To overcome this identified issue, we have enhanced MFE-HC and proposed an efficient Cache based Hybrid Classifier called Cache-HC, which is performing well as compared with our previous MFE-HC in terms of Execution Time and it retains all the qualities of our previous Hybrid Classifier MFE-HC.

Key Words: Pattern Recognition; SVM; k-NN; ICS4; NPPC; NPPC-SVM; MFE-SVM, Cache Oblivious.

I. INTRODUCTION
Bioinformatics is one of emerging and rapidly growing fields of science. The existence public databases of Bioinformatics Projects have billions of data entries have required for a robust analytical scheme for classification and representing this with respect to the biological significance. Thus, the computational tools have been needed to analyze this type of collected data in an efficient manner [1, 2, 3, 4, 11].

With the recent development of genomics and proteomics, the molecular diagnostics has appeared as one of the novel tools to diagnose and to predict the cancer patterns.

In this study, we have mainly focused the various recently proposed Gene Classifiers to measure/predict the accuracy of Cancer Molecular Pattern Identification. The considered best Gene Classifiers [1, 2, 3, 4, 5, 6, 10, 11] are Support Vector Machine (SVM), Nearest Neighbor Classifier (k-NN), ICS4, Non-Parallel Plane Proximal Classifier (NPPC), NPPC-SVM, Margin-based Feature Elimination-SVM (MFE-SVM). Our work has implemented the above identified classifiers and thorough comparative study is made. From the Study, we have observed that the k-NN classifier achieves less Threshold to predict the cancer pattern and it consumes more execution time, whereas the MFE-SVM achieves less execution time to predict the cancer pattern, but it still needs more threshold to predict the Pattern.

To address this issue, in our earlier work, we had proposed Maximizing Feature Elimination Technique based Hybrid Classifier (MFE-HC), it was suitable to predict the pattern earlier with less threshold. However, in our recent study, we noted that its performance in term of Execution Time is slightly degrading for analyzing the large amount of Data. In the following sections, all the above mentioned Classifiers have been discussed.
II. DATA MINING CLASSIFIERS

In this section, we are discussing the recently proposed Data Mining Classifiers [11] and our earlier Hybrid Classifier MFE-HC[1], which are used for classifying the Cancer Patterns.

2.1. Support Vector Machine (SVM)

The Support Vector Machine (SVM) is one of the promising techniques for data classification and regression. It employs two different techniques to deal this issue. First it is introduced a soft margin called hyperplane which adds a penalty function, which is a violation of constraints to the optimization criterion. Secondly the non-linearly transform which is the original input space into a greater dimension feature space. Further in this new feature space it is higher possibility to find a linear optimal separating hyperplane. In the following given training vectors, the xi, i = 1; …. l with the length n, and a vector y, which is defined as

\[ y_i = \begin{cases} 1 & \text{if } x_i \text{ in class 1} \\ -1 & \text{if } x_i \text{ in class 2} \end{cases} \]

The Support Vector technique is trying to find the separating hyperplane with this largest margin between two classes and it measured along a line perpendicular to the this Hyperplane[1, 2, 11].

![Fig. 1 Separating Hyperplane](image)

2.2. Nearest Neighbor Classifier k-NN

Among the various methods and techniques of supervised statistical pattern recognition, this Nearest Neighbor[1,2,3,8,11] rule does achieve consistently good performance, without any priori assumptions, the training examples were drawn. There are various approaches/procedures were proposed to improve both the performance and speed of k-NN. The pre-sort is one of the approaches to train the sets. The nearest neighbor rule is very simple; however its computational cost is high. For numerical example, each and every classification does require 60,000 distance calculations between 784 (28x28 pixels) vectors (dimensional one) [8].

This is one of the simplest classification procedures. To classify a query, find the best similar example in D and predict that x has the same label like that example. To work out this procedure, it has to define a similarity measure on these types of expression patterns. This work used the Pearson correlation which used as a measure of similarity. Let

\[ k_p(x, y) = \frac{E[(x_i - E[x])(y_i - E[y])]}{\sqrt{\text{Var}[x] \text{Var}[y]}} \]

be the Pearson correlation between any two vectors of expression levels. For given a new vector x, the NN classification procedure searches for the vector xi in the training data that maximizes Kp(x,xi), and li , the label of xi.

This simple non-parametric classification scheme does not take any level of global properties of the training set into attention. However, it is effective in various types of classification problems.

2.3. ICS4

The Traditional ensemble method is to build and refine the tree committee and also to derive significant classification rules is impossible still now. So, an efficient new incremental decision learning algorithm introduced which uses this skeleton of ITI and it accepts CS4 methods to break the constraint of various singleton classification rules by producing various significant rules.

2.4. Non-Parallel Plane Proximal Classifier (NPPC) and NPPC-SVM

The Nonparallel Plane Proximal Classifier (NPPC)[3] which is the combine ideas from both Twin Support Vector Machine (TWSVM)[1,2,3] and Proximal SVM[1,3]. The NPPC is used to find two nonparallel hyperplanes and each plane is clustered around with a particular class data, which is shown in the Fig. 2. The design of NPPC for different binary data classification is based on two similar Mean Square Error (MSE). It leads to solve two small systems, which are of linear equations. These linear equations are derived as NPPC1 and NPPC2 [3].
To improve the diagnostic accuracy observed, this classification accuracy of a single NPPC do not satisfy on microarray data by which using a small set of informative, Santanu Ghorai and et.al. introduced this NPPC ensemble with SVM and it is established that this ensemble NPPC-SVM[1,3] is performing better than single NPPC.

2.5. Margin-based Feature Elimination-SVM (MFE-SVM)

This Margin-based backward Feature Elimination (MFE)[1,4] is developed for Linear and Nonlinear as well and then it was designed to consider the nonlinear kernels. MFE for the nonlinear kernel is giving both the better margin and generalization accuracy. The authors Yaman Aksu and et.al. then present an MFE extension model which stepwise achieves further gains in margin at small computational cost.

This extension solves the SVM optimization problem which is used to maximize the classifier’s margin at every feature elimination step, albeit in a lightweight fashion by adjusting only over a small set of parameters, which are very similar to a method recommended in [1,4].

2.6. Maximizing Feature Elimination Technique based Hybrid Classifier (MFE-HC)

In our earlier work, Maximizing Feature Elimination-Hybrid Classifier (MFE-HC)[1], which is the hybrid algorithm which hybrids two methods k-NN Classifier and MFE-SVM Classifier. The basic idea of this proposed mechanism was that it finds the close neighbors and trains a local SVM that focuses the distance function on the group of neighbors. This method outperforms both the Nearest Neighbor and SVM. The selection strategy which was used in this technique was developed based on the idea of Yaman Aksu and David J. Miller [4], which is the Sequential Backward Feature Selection. But, we had modified this sequential procedure in to parallel framework, which eliminates the irrelevant data earlier. The selection process was starting from the full set of feature and then it removes sequentially the most irrelevant features. To find the best and most irrelevant feature of the existing and current surviving subset, one of the features was removed and then the size of margin is calculated.

It was denoted as MM(i), that was the maximized portion or boundary without the i\textsuperscript{th} feature, which returns to the subset, where this procedures work till the features are over. Finally, this best irrelevant feature, which its most removal produced highest value of maximized margin, which can be found. This procedure is repeated till all of the features have been removed. By monitoring this type of identification of the best subset, this has the maximum generalization performance of the produced ranking of the each feature. This is the Parallel Backward Feature Selection. The procedure of Hybrid Classifier is given below.

Step 1: Initialization:
Subset of surviving features ie s=[1,2,…,n]

Step2: i . Compute distances of the query to all the training examples and choice/pick the nearest K neighbors;
ii if the K neighbors have the same labels, this query is labeled and then exit;
else, compute the pairwise distances between all the K neighbors;
i. Convert the distance matrix into kernel matrix and then apply multiclass SVM as follows;

Step3: Repeat

For ∀S\textsubscript{i} \in\{1 ≤ i ≤ S \}

Do train the Hybrid Classifier (k-NN+SVM) without the i\textsuperscript{th} feature and
Do compute J\textsubscript{m}=MM(i) (Where MM(i) is the maximized margin level without i\textsuperscript{th} feature)

Discovery the worst feature the k

k = m \textsuperscript{-} \arg \max(J_q)

Remove this feature of maximizes MM

s = [1,…….,k-1,k+1,……,n]

Step 4: Until s is empty

It will identify the best subset and it has the best generalization performance of this feature ranking. This procedure was implemented under Parallel Execution Environment to predict the Cancer Pattern with Minimum Execution Time. The performance analysis of the above discussed all Classifiers had been implemented with the help of Bio-Weka and VC++ Tools and analyzed.
TABLE. 1. Cancer Patterns Prediction Results (Classifiers Vs Threshold & Classifiers Vs Execution Time)

<table>
<thead>
<tr>
<th>Classifiers</th>
<th>Cancer Patterns</th>
<th>Bladder</th>
<th>Threshold</th>
<th>Execution Time (ms)</th>
<th>Breast</th>
<th>Threshold</th>
<th>Execution Time (ms)</th>
<th>Colon</th>
<th>Threshold</th>
<th>Execution Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td></td>
<td></td>
<td>55</td>
<td>200</td>
<td>42</td>
<td>205</td>
<td>38</td>
<td>204</td>
<td></td>
<td></td>
</tr>
<tr>
<td>kNN</td>
<td></td>
<td></td>
<td>51</td>
<td>205</td>
<td>38</td>
<td>206</td>
<td>38</td>
<td>206</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IC54</td>
<td></td>
<td></td>
<td>56</td>
<td>150</td>
<td>43</td>
<td>150</td>
<td>39</td>
<td>152</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NPPC</td>
<td></td>
<td></td>
<td>57</td>
<td>140</td>
<td>44</td>
<td>141</td>
<td>41</td>
<td>142</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NPC-SVM</td>
<td></td>
<td></td>
<td>54</td>
<td>122</td>
<td>41</td>
<td>126</td>
<td>38</td>
<td>125</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MFE-SVM</td>
<td></td>
<td></td>
<td>55</td>
<td>120</td>
<td>42</td>
<td>122</td>
<td>38</td>
<td>124</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MFE-HC</td>
<td></td>
<td></td>
<td>51</td>
<td>120</td>
<td>38</td>
<td>122</td>
<td>35</td>
<td>124</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

For this experimental study of our earlier work MFE-HC, we had used 10 different Cancer Patterns/ Gene Sequences datasets, which were downloaded from National Center for Biotechnology Information (NCBI)[7] website. A few used cancer patterns are Lymphoma Cancer, Breast Cancer, Colon Cancer, Lung Cancer, Melanoma Cancer, Thyroid Cancer, Kidney Cancer, Leukemia Cancer, Pancreatic Cancer, and Endometrial Cancer.

From the output of our earlier MFE-HC, we were observed that the prediction level (Threshold value) for a Cancer Pattern is different for different Classifiers. It was also noted that the execution time of classifiers are different to predict the cancer pattern, which are shown in the Table 1. From the Table 1, we had established that our earlier Hybrid Classifier, Maximizing Feature Elimination Technique based Hybrid Classifier (MFE-HC) outperforms both the k-NN and MFE-SVM Classifiers in terms of Threshold, Execution Time, Memory Usage and Memory Utilization.

2.7. Identified Problems

Our earlier work, Maximizing Feature Elimination Technique based Hybrid Classifier (MFE-HC) outperforms both the k-NN and MFE-SVM Classifiers in terms of Threshold, Execution Time, Memory Usage and Memory Utilization. However, in our recent study, we noted that the performance of MFE-HC in term of Execution Time is slightly degrading while predicting or analyzing the Cancer Patterns from the large amount of Data Set. Thus we need an efficient Classifier to predict the Cancer Patterns from the large volume of Data Set. To address and to overcome this identified issue, this research work has proposed an efficient classifier called Cache-HC : Cache Oblivious based Hybrid Classifier For Cancer Molecular Pattern Discovery in Large Dataset. The procedure and methodology of this proposed model is discussed in the following section.

III. PROPOSED CACHE-HC : CACHE OBLIVIOUS BASED HYBRID CLASSIFIER

As stated in the previous section, to improve the performance of MFE-HC in term of Execution Time, we have proposed an efficient Cache- Oblivious based Hybrid Algorithm called Cache-HC. It performs better in term of Execution Time and it retains the capability of predicting Cancer Patterns. The model of Cache-HC is shown in the Fig. 3.

Fig.3. Cache- HC based Parallel Processing Model

As shown in the Fig. 3, the Pattern String ie data is transferred from Main Memory to allotted Cache Level based upon the procedure of our proposed Cache-HC Scheme, which is developed based on Parallel Computing Methodology. In this Parallel Processing Model, each Processor has its own private cache of size C₁ to each Core/CPU, which is called as Distributed Caches L₁ and single shared L₂ cache of size C₂ where C₂ > p. C₁, where p is the number of Processors involved in this Model.

To implement the Cache-Oblivious based Hybrid Classifier, the I/O complexity is analyzed and for this, we have denoted that let K is the number of Strings to be Searched and Matched, let N is the number of words in the K Strings, let M is the number of words fitting in the main memory and let...
$B$ is the number of words per disk block, where $M < N$ and $1 < B \leq M/2$. The input sequence is assumed as $x_1, x_2, \ldots, x_k$ is given in a form such that the I/O complexity is defined as $O(N/B)$ I/Os.

For $K$ binary strings comprising $N$ words in total, this technique does find the searching order and the longest common prefix sequence of the strings using $O(K/B \log M/B(K/M) \log(N/K) + N/B)$ I/Os.

The pseudo code for Cache Simulation is given below.

```
CacheHCProc()
{
    Proc *p; Block *b; Inst *i;
    AddCallProto("LoadReference(VALUE)");
    AddCallProto("StoreReference(VALUE)");
    AddCallProto("PrintResults()");
    for (p = GetFirstProc(); p != NULL; p = GetNextProc(p) )
    {
        for (b = GetFirstBlock(p); b != NULL; b = GetNextBlock(b) )
        {
            for (i = GetFirstInst(b); i != NULL; i = GetNextInst(i) )
            {
                if (GetInstClass(i) == ClassLoad || GetInstClass(i) == ClassFload)
                {
                    AddCallInst(i,InstBefore,"LoadReference",EffAddrValue);
                }
                if (GetInstClass(i)== ClassStore || GetInstClass(i)== ClassFstore)
                {
                    AddCallInst(i,InstBefore,"StoreReference",EffAddrValue);
                }
            }
        }
    }
    AddCallProgram(ProgramAfter,"PrintResults");
}
```

As discussed above, the following Hybrid Classifier is implemented and tested thoroughly.

**Step 1**: Initialization:
Subset of surviving features ie $s = [1, 2, \ldots, n]$

**Step 2**:

- i. Compute distances of the query to all the training examples and choice/pick the nearest $K$ neighbors;
- ii. If the $K$ neighbors have the same labels, this query is labeled and then exit; else, compute the pairwise distances between all the $K$ neighbors;
- iii. Convert the distance matrix into kernel matrix and then apply multiclass SVM as follows;

**Step 3**: Repeat

For $S_{i}(1 \leq i \leq |S|)$

Do train the Hybrid Classifier (k-NN+SVM) without the $i^{th}$ feature and
Do compute $J = MM(i)$ (Where $MM(i)$ is the maximized margin level without $i^{th}$ feature)

Discovery the worst feature the $k$

$$k = m \arg \max_q \left( J_q \right)$$

Remove this feature of maximizes MM

$$s = [1, \ldots, k-1, k+1, \ldots, n]$$

**Step 4**: Until $s$ is empty
IV. PERFORMANCE ANALYSIS AND DISCUSSIONS

The proposed Cache Oblivious based Hybrid Classifier is implemented with Bio-WEKA and it is studied thoroughly in terms of Threshold and Execution Time. The Bio-Weka is configured with our designed VC++ Tool which is shown in the Fig. 4.

This is implemented successfully and compared with the recently proposed our MFE -HC and other popular Data Mining Classifiers namely Support Vector Machine (SVM), Nearest Neighbor Classifier (k-NN), ICS4, Non-Parallel Plane Proximal Classifier (NPPC), NPPC-SVM, Margin-based Feature Elimination-SVM (MFE-SVM). The experimental results are shown in the Figures from Fig. 5 to Fig. 8.

To implement our Cache-HC, we have used System which has the following Windows configurations. We have used Windows 7 Home Premium 64 bit with 2.5 GHz Intel Core i5-2450M, 4 GB RAM, 64KB L1 Primary Cache and 256 KB L2 Secondary Cache.

For this experimental study, as we have discussed in the previous section, we have used 10 different Cancer Patterns, which are used for studying our earlier Hybrid Classifier MFE-HC and name of the few patterns which are used are Bladder Cancer, Breast Cancer and Colon Cancer.

![Fig. 4 Execution Part of the proposed Cache-HC](image)

From the Fig. 5, it is observed that for Bladder Cancer Pattern Identification/Prediction, the proposed Cache-HC predicts the Bladder Cancer pattern with minimum Execution time, 109ms as compared with that of our earlier proposed Hybrid Classifier, MFE-HC, which needs 120ms to complete the prediction procedure. From the Fig. 5, it is also noted that both of our Techniques (MFE-HC and Cache-HC) predicts the pattern with the same minimum threshold of 51. This is achieved because of using Cache
Oblivious Approach. As we are using Cache Oblivious Approach, this proposed Cache-HC achieving better Memory Utilization and Memory Usage. Similarly our proposed Cache-HC outperforms our previous Classifier MFE-HC to predict the Breast Cancer Pattern and Colon Cancer Patterns as well in terms of Execution Time, Memory Usage and Memory Utilization, which are shown in the Fig. 6 and Fig. 7.

**Fig. 6. Threshold and Execution Time of Classifiers for Predicting Breast Cancer Pattern**

The overall Performance Comparison of our proposed model in terms of Execution Time, Threshold, Memory Usage and Memory Utilization with the existing Classifiers is shown in the Fig. 8

**Fig. 7. Threshold and Execution Time of Classifiers for Predicting Colon and Rectal Cancer Patterns**

**Fig. 8. Performance Comparison of Proposed Cache-HC Classifier**
V. CONCLUSION

We have proposed Cache Oblivious based Hybrid Classifier called Cache-HC to predict/identify the cancer patterns. This is implemented with Bio-Weka through our developed VC++ Interfacing Tool and studied thoroughly. The performance of our proposed Cache-HC is compared and analyzed with our previous Hybrid Classifier, MFE-HC in terms of Threshold Level, Execution Time, Memory Usage and Memory Utilization. From our experimental results, it is established that the proposed Cache-HC achieves better performance as compared with MFE-HC in terms of Execution Time, Memory Usage and Memory Utilization and retains the same level of Threshold of MHE-HC.

REFERENCES


