CHAPTER 5

ALL-C USING MACHINE LEARNING CLASSIFIER

Phase II of the research work performs ALL classification using machine learning classifier. The proposed system, as mentioned in Chapter 3, Section 3.3, consists of two major steps, namely, feature engineering and classification. This chapter discusses the various methods used in each of these steps and explains how they are enhanced to improve ALL classification. The proposed classification system uses the segmented WBC regions obtained in Phase I.

5.1. FEATURE ENGINEERING

Machine Learning Classifiers (MLC) have become a powerful tool in healthcare industries. To design an accurate classifier, it is required to use appropriate features. Selection of correct set of features by itself is a challenging task. The collection, cleaning and selecting optimal features is the most challenging task of MLC. MLCs use significant characteristics or features of an image as input to calibrate and produce classification result (Sheriff *et al.,* 2020). As the features used is directly responsible for the success of the MLC, it is vital to construct a quality feature vector.

The features are defined as individual quantifiable properties of microscopic images and are generally stored as a 2-dimensional array, where the rows represent the microscopic image and columns represents the corresponding features extracted. The 2-dimensional form of the features extracted is general referred to as feature vector. The feature vector originally is in its basic form and as such can be used directly with a MLC. However, when the same feature vector is enhanced, further improvement to the process of MLC can be achieved. This enhancement operation is called feature engineering (Ayuya *et al.*, 2020).

The goal of using feature engineering with MLC is two-folds and are listed below.

1. To construct an input feature vector that is compatible and best fits the MLC used to perform ALL-C, and
2. To improve the accuracy of the MLC.

The first task of Phase II is focused on feature engineering that results with a feature vector having optimal features. Optimal features, at this point, is defined as a feature vector that has only significant and relevant features without redundancy. To attain this goal, the feature engineering task uses two steps, where the first step performs feature extraction and the second step performs feature selection to identify optimal features.

Several researches (Khaire and Dhanalakshmi, 2022; Tran *et al.*, 2019; Liu *et al.*, 2008) have proved that the usage multiple features that combine different types of characteristics from an image are more successful in improving classification accuracy, in contrast to the usage of a single type of feature. The reason behind this success is because each set of features extracted portray certain aspect of the WBC region content, which when fused together can provide a more clear description of the region. This work, thus, extracts multiple heterogeneous features from the blood cell regions identified, with the aim of improving ALL detection accuracy.

One side effect of using multiple features is the curse of dimensionality. This problem is often solved through the use of feature selection algorithm (Saarela *et al.*, 2021), which, as mentioned earlier, selects only relevant and non-redundant features, without compromising classifier performance. The usage of feature selection algorithm provides multiple advantages to ALL-C system. They include,

(i) Improve ALL-C system accuracy,
(ii) Reduction of feature vector size,
(iii) Reduction of training time, and
(iv) Decrease over-fitting (removal of redundant data decreases the probability of making wrong decision based on noise).

The feature selection algorithm can be applied separated to each type of feature extracted and be used directly to classify the blood cells. But, this is time consuming and inefficient. Hence, a feature fusion algorithm is included, which is defined as the art of merging various types of features into a single super feature vector (Wang *et al.*, 2017). This improves the representativeness of the target features and aids to learn microscopic images completely for description of their internal data (Wonjun *et al.*, 2017).

Thus, the feature engineering step of the proposed ALL-C system using machine learning algorithm has three tasks, namely, feature extraction, fusion and selection. The process of engineering used is shown in Figure 5.1.

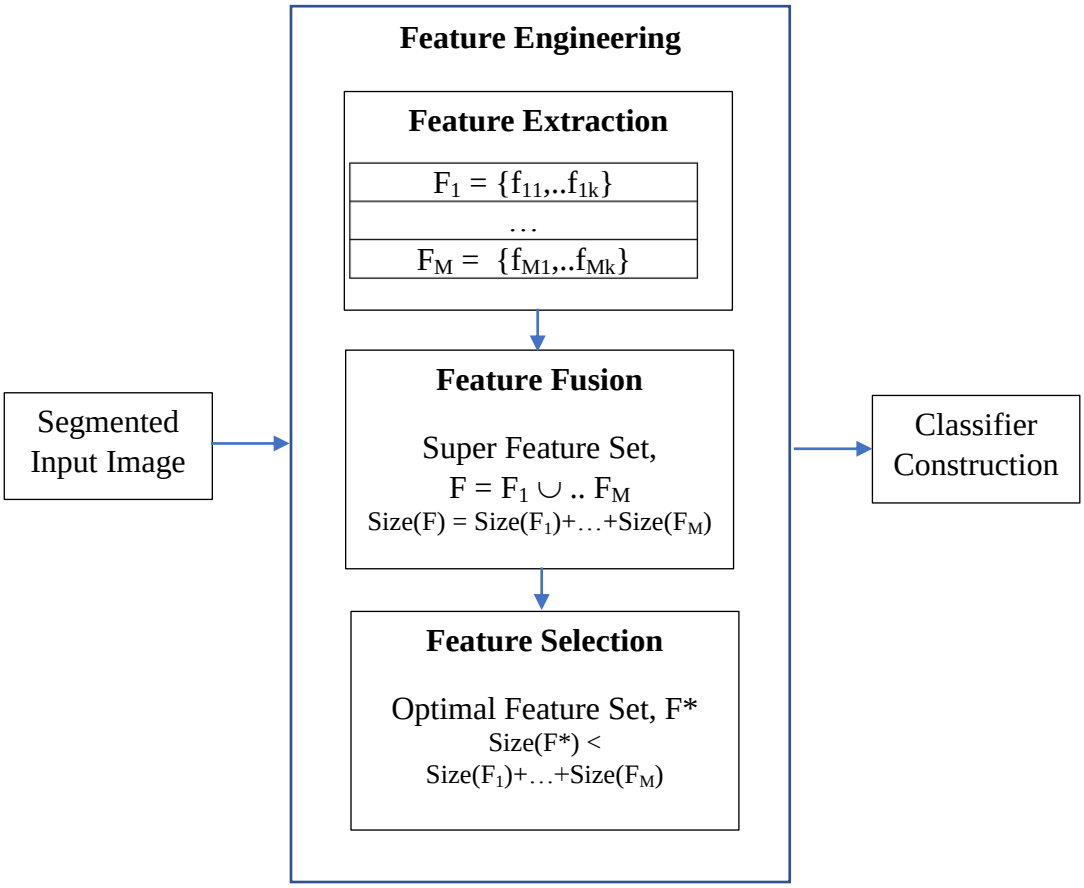


Figure 5.1 : Process of Feature Engineering

5.1.1. Feature Extraction

Feature extraction is defined as the task that extracts stable and unique features that best represent the microscopic image. This task helps to represent the WBC region in a low-dimensional form (Mutlag *et al.*, 2020). It is also considered as an algorithm that redefines a large set of data into a set of size reduced vectors. The main aim of the feature extraction task in this research is to extract a set of descriptors that best represent the patterns in the microscopic image to which the pathologists refer. In order to accurately project the various characteristics of the microscopic image, a total of 19 features, belonging to the four types (Table 5.1) are extracted.

TABLE 5.1

FEATURES EXTRACTED FROM MICROSCOPIC IMAGE

Feature Type	Features Extracted
Texture Features (5)	Energy, Entropy, Contrast, Correlation, Homogeneity
Shape Features (10)	Area, Perimeter, Eccentricity, Elongation, Compactness, Minor Axis, Major Axis, Solidity, Form Factor, Nucleus-Cytoplasm Ratio
Color Features (2)	Mean, Standard Deviation
Irregularity (2)	Horizontal Direction, Vertical Direction

(i) Texture Feature

Texture of the blood cells in microscopic images were measured using GLCM (Gray-Level Co-occurrence Matrices). Here, the texture is defined as a function of the spatial variations in pixel intensities. The gray-level distribution of an image is defined using second order statistics based on a co-occurrence matrix. A co-occurrence matrix is a distribution of co-occurring values, defined over an image, at a given offset. In other words, the texture using GLCM is defined as the probability of two pixels having a same gray levels with specific spatial relationships.

The Haralick features (Haralick, 1979) are also used as texture descriptors and the most widely used statistical method used to analyse texture. A total of 14 texture features are available to describe spatial relationship amongst gray level values of pixels with in a region. A comprehensive review on statistical algorithms is provided by Haralick (1979) and Gool *et al.* (1985). The GLCM is a tabulation of how often different combinations of pixel grey level values occur in a microscopic image. The features extracted are presented in Table 5.2.

TABLE 5.2

TEXTURE FEATURES EXTRACTED

S.No.	Texture Feature	Formula	Description
1	Energy	$\sqrt{\sum_{i,j=0}^{N-1} P_{i,j}^{2}}$	Used to measure the uniformity of an image or the angular second moment of an image.
2	Entropy	$\sum_{i,j=0}^{N-1} P_{i,j}(-\ln P_{i,j})$	Used to measure the disorder of an image. This measure is high when the image is not texturally uniform.
3	Contrast	$\sum_{i,j=0}^{N-1} P_{i,j}(i-j)^{2}$	Used to measure the amount of local variations present in the image.
4	Correlation	$\sum_{i,j} \frac{(i-\mu_i)(j-\mu_i)p(i,)}{\sigma_i \sigma_j}$	Used to estimate the regional-pattern linear dependence in the image.
5	Homogeneity	$\sum_{i,j=0}^{N-1} \frac{P_{i,j}}{1+(i-j)^{2}}$	Used as a measure of the degree of variation.

(ii) Shape Features

The second set of feature extraction focus on the most important and powerful feature that can identify unique shapes in microscopic images that may match the blood cells. The shape features extracted should be invariant to position, scaling and rotation variations. The main motive of using shape features is to extract features that can determine how close a shape in the image that correlate the ROI. In ALL-C, all shape features extracted from the blood cells detected which are represented by non-zero values. The haematologists have reported that shape is an important feature that can competently discriminate the blood cells. The analysis using shape features can be performed efficiently using the region and boundary-based features. The extracted features are described in Table 5.3.

TABLE 5.3

SHAPE FEATURES EXTRACTED

S. No.	Shape Feature	Description
1	Area	Determined as the summation of non-zero pixels within the region
2	Perimeter	Determined by calculating the distance between successive boundary pixels
3	Eccentricity	Used to measure how much a shape of a regions detected deviates from being circular. It is an important feature since mature blood cells are more circular than normal cells and can be calculated using Equation (5.1). In this equation, a is the minor axis and b is the major axis. $$\text{Eccentricity} = \frac{\sqrt{a^2 - b^2}}{a} \quad (5.1)$$
4	Elongation	Abnormal bulging of the nucleus is also a feature which signifies toward leukemia, which can be efficiently measured using a ration called elongation. It is defined as the ratio between maximum distance (*R*max) and minimum distance (*R*min) from the centroid to the detected objects' boundary (Equation 5.2). $$\text{Elongation} = \frac{R_{max}}{R_{min}} \quad (5.2)$$
5	Compactness	Compactness or circularity represents the degree to which a shape is compact. The circle is the most compact shape since among the shapes with the same perimeter, it has the least area and can be estimated using Equation (5.3) $$\text{Compactness} = \frac{\text{Perimeter}^2}{\text{Area}} \quad (5.3)$$
6	Minor Axis	Described as the length of the shortest line which passes through the centroid of the nucleus and is measured in pixels.
7	Major Axis	Described as the length of the longest line which passes through the centroid of the nucleus and is measured in pixel
8	Solidity	Defined as the ratio of actual area and convex hull area is known as solidity (Equation 5.4). $$\text{Solidity} = \frac{\text{Area}}{\text{Convex Area}} \quad (5.4)$$
9	Form Factor	It is a dimensionless parameter which measures the circularity of the nucleus and is estimated using Equation (5.5). $$\text{Form Factor} = \frac{4x\pi x\text{Area}}{\text{Perimeter}^2} \quad (5.5)$$
10	N-L Ratio	Nucleus-Lymphocytic is a ratio of the area of the nucleus to the area of the lymphocyte. The N:L ratio indicates the maturity of a cell because as a cell matures, the size of its nucleus generally decreases. Hence, "blast" forms start with an N:L ratio of 4:1, which decreases as they mature to 2:1 or even 1:1. Thus, N:L ratio can be used as an efficient discriminative feature to classify ALL. It is estimated using Equation (5.6). $$\text{N} - \text{L Ratio} = \frac{\text{Area of Nucleeus}}{\text{Area of Blasts}} \quad (5.6)$$

(iii) Color Features

Out of the many feature extraction techniques, color is considered as another most dominating and distinguishing content feature of the microscopic image (Vaghela *et al.*, 2015). Color is a property that depends on the reflection of light to the eye and the processing of that information in the brain. Humans use colors every day to differentiate between objects, places and the time of day. Out of the various color feature vector construction method, this research work extracts two types of features, namely, mean (Equation 5.7) and standard deviation (Equation 5.8).

$$\text{Mean, } \mu = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} P_{ij} \tag{5.7}$$

$$\text{Standard Deviation, } \sigma = \sqrt{\frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} \left(P_{ij} - \mu\right)^2} \tag{5.8}$$

In the above two equations, M and N denotes the dimension of the image, p_{ij} is the color value of the jth pixel in the ith color channel.

(iv) Irregularity Features

The irregularities in the boundary is also another feature that can be used to classify blood cells. The boundary of the image is generally represented using the contour of two dimension. The irregularity of the boundary described using the contour is calculated as the distances from each contour point (otherwise known as edge pixels) to a reference point. The distance is measured using Euclidean distance, which measures the distance from the centroid point to the contour points using the steps given below.

Step 1 : Obtain boundary pixel position using the edge image obtained using Canny edge detector.

Step 2 : Calculate centroid coordinates of the detected region using the relations given in Equations (5.9) and (5.10). In these equations, x, y are the coordinates of the pixels along the contour and N is the total number of pixels in the contour.

$$\bar{x} = \frac{1}{N}\sum_{n=0}^{N-1} x(n) \quad (5.9)$$

$$\bar{y} = \frac{1}{N}\sum_{n=0}^{N-1} y(n) \quad (5.10)$$

Step 3 : Estimate the Euclidean distance between each boundary and centroid

Step 4 : Calculate the variance of all the distances (obtained from previous step) from the distance in order to measure the irregularity of the boundary.

(v) Normalization

The above extracted features, before further processing, are normalized. Normalization is a process that balances the range of values of difference features extracted within a fixed range [0-1] and can help to improve the classification performance (Ramezani *et al.*, 2014). Without normalization, some large feature values may cause some small feature values to be ignored, which might be important during ALL identification. In this research, all feature extracted are normalized to zero mean and standard deviation = 1 (Equation 5.11).

$$\hat{x}_i = \frac{x_i - x_{i,min}}{x_{i,max} - x_{i,min}}, \; i = 1, \ldots, N \quad (5.11)$$

In the above equation, $\hat{x}_i$ is the normalized feature value of feature x_i and $x_{i,min}$, $x_{i,max}$ respectively denotes the minimum and maximum range of x_i.

5.1.2. Feature Fusion

The application of the above step, feature extraction, results with five sets of normalized feature vectors. The main objective of using feature fusion step is to combine these five sets of features into a single feature vector. Feature fusion is also considered as an important step while improving classification performance (Zhou *et al.*, 2022). The feature fusion algorithm is used because of the following causes.

(i) Issues raised by incompatible features raised due to the multiple modalities of the features,

(ii) Issues raised by compatibility that occur because of unknown relationships that exist between different feature sets, and

(iii) Issue of 'Curse of Dimensionality' that is caused by the huge feature vector obtained as the result of combining various feature vectors.

The procedure used to perform feature fusion is described below.

Let D_{f1}, D_{f2}, D_{f3} and D_{f4} be the dimensions of the five vectors generated by the feature extraction algorithms. The dimensions of the five feature vectors may be dissimilar. To solve this issue, the proposed feature fusion algorithm uses a weighted coefficient, θ, to handle the problem of size imbalance. The steps performed by the feature fusion algorithm to form a super feature vector is given in Figure 5.2. The process of integration is repeatedly applied between two pairs of feature set, until the final super feature set is formed. The dimension of the super set will be $\sum_{i=1}^{5} d_i$.

Input : Normalized Feature Sets ($f_1 \ldots f_4$) ; Output : Super feature vector, F.

Step 1 : Let F = {}

Step 2 : i=2; TF = f_i; D_{TF} = Dimension of f_1

Repeat

if D_{TF} = = D_{fi} then θ = 1 and F = $\begin{pmatrix} TF \\ f_i \end{pmatrix}$

else if $D_{TF} > D_{fi}$ then θ = $\frac{D_{TF}^{\ 2}}{D_{f_i}^{\ 2}}$ and F = $\begin{pmatrix} \theta TF \\ f_i \end{pmatrix}$

elseif $D_{TF} < D_{fi}$ then θ = $\frac{D_{f_i}^{\ 2}}{D_{TF}^{\ 2}}$ and F = $\begin{pmatrix} f_i \\ \theta TF \end{pmatrix}$

i = i+1;

TF = F; D_{TF} == Dimension of TF

Until i > 6;

Step 4 : Output TF and D_{TF}

Figure 5.2 : Steps in Feature Set Fusion Algorithm

5.1.3. Feature Selection

In general, the discriminative capability is high with a large-sized feature set. However, usage of large-sized feature set has several disadvantages including slowing down classification process and overfitting. According to Liu *et al.*, 2017), a smaller training set with only relevant features can construct a more sophisticated classifier. This can be achieved through the use of a feature selection algorithm. A feature selection algorithm has several advantages like reducing classification complexity, making the class scalable and improving classifier's generalization ability and accuracy, thus helping to construct a fast and cost-effective classifier.

A feature selection relies on two types of concepts of information theory, namely, relevancy and redundancy. An optimal feature set can be constructed by using a feature selection algorithm that can implicitly handle these two concepts (Kumar and Minz, 2013). In this research work, a Maximum Relevance Minimum Redundant (MRMR) feature selection algorithm is used.

The features in an image dataset can belong to into three main categories, namely, relevant features (Re), redundant features (R) and non-redundant (NR) features. The relevant features can further be grouped as strongly relevant (SRe), weakly relevant (WRe) and irrelevant (IR) features (John *et al.*, 1994). This is portrayed in Figure 5.3.

Figure 5.3 : Types of Features

All application desire for a feature selection algorithm that can construct the optimal feature vector that includes all SRe features and all WRe with NR features. That is the desired feature selection algorithm removes all IR features and redundant WRe features. This desired concept is presented in Equation (5.12).

$$F^* = \{SRe + WRe \text{ but } NR\} \quad (5.12)$$

Search for such an ideal feature selection revealed a filter-based algorithm, called MRMR or Maximum Relevant Minimum Redundant feature selection algorithm (Ding and Peng *et al.*, 2003). The algorithm is a minimal-optimal algorithm that identifies a subset of features which has maximum possible classification power.

The MRMR algorithm select features that correlate strongest to the classification variable. This step performs maximum-relevance selection. Alternatively, features that are mutually far from each other, which having high correlation to the classification variable is also selected. This step performs minimum redundant selection. Thus, MRMR select features having maximum relevance and minimum redundancy.

The MRMR was so named because, at each iteration, it is desired to select features that have maximum relevancy with respect to the target feature and minimum redundancy with respect to the features selected during the previous iterations. For this, a score is calculated for each feature to be evaluated, during each iteration. Let f be the feature to be evaluated and i denote the iteration, then the score is calculated using Equation (5.13).

$$\text{Score}_i(f) - \frac{F(F, \text{target})}{\sum_{a \in \text{features selected until i-1}} \frac{|\text{corr}(f,s)|}{(i-1)}} \tag{5.13}$$

In the above equation, corr refers to correlation and refers to the Information Gain (IF). The correlation is taken in absolute value, which means, that if two features have -4 and +4 correlation, it is considered as highly redundant feature. The algorithm selects the feature as best if its score is high. The function F denotes the F-statistical measure used. Equation 5.14 shows the method used to estimate correlation. This method is defined as a measure of uncertainity of a feature X. Similarly, Equation (5.15) presents the method used to estimating the historical values of another feature Y.

$$H(X) = -\sum_i P(X_i) \log_2(P(x_i)) \tag{5.14}$$

$$H(X \mid Y) = \sum_j P(y_i) \sum_i P(x_i \mid y_i) \log_2(P(x_i \mid y_i)) \tag{5.15}$$

In the above equation, the historical probabilities for all values of X is denoted by P(xi) and P(xi|yi) is the posterior probabilities of the feature X given the values of Y.

Additional information of X provided by Y can be measured by analysis the amount by which the entropy of X decreases and this information is termed as "Information Gain (IG)" (Quinlan, 1993) (Equation 5.16).

$$IG(X \mid Y) = H(X) - H(X \mid Y) \tag{5.16}$$

With IG, a feature Y is more correlated to a feature X than a feature Z, if it satisfies Equation (5.17).

$$IG(X \mid Y) > IG(Z, Y) \tag{5.17}$$

Thus, IG is a symmetric measure that estimates the correlation between features irrespective of its order. The values thus obtained are normalized using a Symmetric Uncertainty (SU) method (Press *et al.*, 1988) using Equation (5.18).

$$SU(X, Y) = 2\left(\frac{IG(X \mid Y)}{H(X) + H(Y)}\right) \tag{5.18}$$

The SU(X, Y) from the above equation helps to compensate IG's bias towards features that has more values and restricts them to values in the range [0 1]. During analysis, a value of 1 denotes highly correlated features, while 0 denotes that the features are independent. The IG of the feature set is estimated using the method proposed by Usama and Keki (1993) and is modified to consider class to feature and feature to feature relationships. Consideration of these relationships helps to find a feature subset that is small and similar to the optimal set desired (Equation 5.12). Figure 5.4 presents the general steps involved in the MRMR algorithm that considers both relevancy and redundant analysis that takes into account the class-feature and feature-feature relationships.

The MRMR feature selection algorithm consists of two main stages, where the first stage analyzes the features to identify relevant features by exploiting the class to feature relationships that exist between features, while the subsequent second stage identifies the non-redundant features among relevant features by analyzing the feature-feature relationship through Markov Filter (MF) method. Both the stages using the SU (Equation 4.9) as a measure to estimate the relevancy and redundancy of a feature. The class to

feature relationship exhibits the correlation between a feature and the class it belong to, while the feature to feature relationship exhibits the correlation between two features.

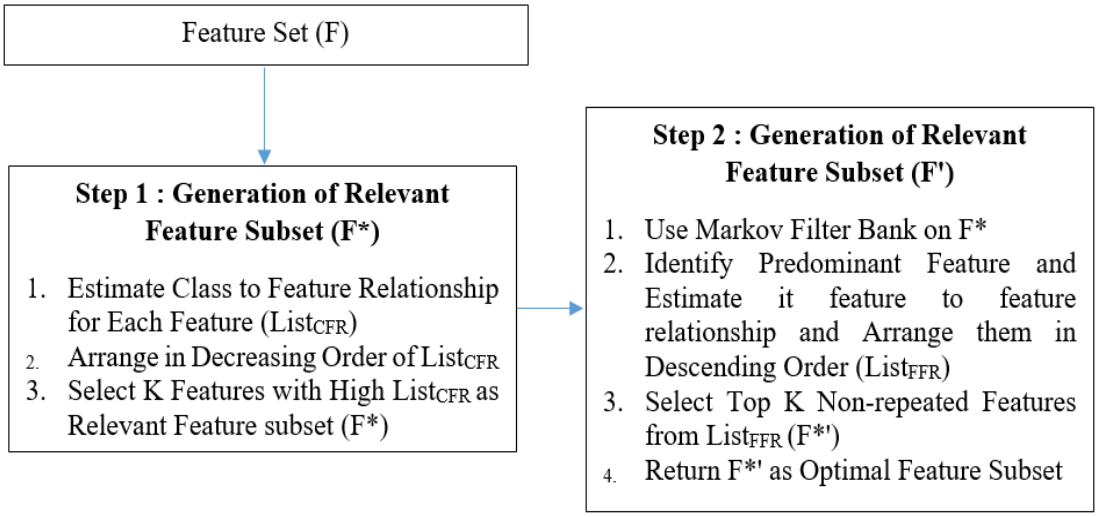


Figure 5.4 : MRMR Algorithm

The MRMR feature selection algorithm consists of two main stages, where the first stage analyzes the features to identify relevant features by exploiting the class to feature relationships that exist between features, while the subsequent second stage identifies the non-redundant features among relevant features by analyzing the feature-feature relationship through MF method. Both stages use the SU (Equation 4.9) as a measure to estimate the relevancy and redundancy of a feature. The class to feature relationship exhibits the correlation between a feature and the class it belongs to, while the feature to feature relationship exhibits the correlation between two features.

Let F_i and F_j be any two features, with $F_i \neq F_j$ and let C_i be a class label. While estimating class to feature relationship, both F_i and F_j are treated as single feature, which is referred to as F_{ij} and the Cartesian produce of F_i and F_j is taken as the domain of $SF_{ij.}$ The first step, estimation of relevant features, starts by calculating the class to feature relationship of all features in the feature set and the result is arranged in descending order. All the features that has high class to feature relationship value contains maximum class information, thus exhibiting high correlation between the feature and its class and therefore, are treated as relevant features. This decision is made based on the condition given in Equation (5.19).

$$SF_i = \begin{cases} \geq T & \text{Relevant} \\ < T & \text{Irrelevant} \end{cases} \quad (5.19)$$

where T is a user-defined relevance threshold. Careful selection of this threshold will improve the prediction performance and repeated runs showed that the relevant and irrelevant feature identification is more accurate when T value is near to unity. Hence, it was set to one.

In the second step, the identified relevant features are further analyzed to detect redundant features. For this purpose, details in relation to feature to feature relationship that exist between individual features are calculated. This considers the correlation relationship between features leaving out the correlation relationship that exists between feature subsets. During redundancy feature identification, care is taken to consider two concerns. The first is related to uncertainty redundancy and the second is involved with the time complexity. Uncertainty redundancy reflects the situation when two features are not completely correlated and the decision of which one to remove has to be made. Time complexity reflects to the calculations that has to be performed on all pairs of features, which can become inefficient and time consuming with large and high dimensional datasets. Both these concerns are solved using predominant features and Markov blanket method.

The definition of redundancy states that a feature is said to be redundant if it is weakly relevant and has a Markov blanket inside the current feature set. Markov Blanket (MB) method is a computationally exhaustive subset selection algorithm that uses the relevant feature subset identified in Step 1 as input, in order to find and remove redundant features.

Let G be a subset of the feature set SF. Let f_G denote the projection of f onto the features in G. The MB works to reduce the discrepancy between the conditional distributions P(C|SF = f) and P(C|G = f_G), as measured by a conditional entropy (Equation 5.20).

$$\Delta_G = \sum_f P(f) D(P(C \mid SF = f) \,\|\, P(C \mid G = f_G)) \quad (5.20)$$

Where, $D(P||Q) = \Sigma_x P(x) \log(P(x)/Q(x))$ and is the Kullback-Leibler divergence. The goal is to find a small feature set G for which Δ_G is small. Intuitively, if a feature SF_i is conditionally independent of the class label given some small subset of the other features, then it should be possible to omit F_i without compromising the accuracy of class prediction. A MB is thus defined as follows. Let SF_i be a feature and $MB_i \subset SF(SF_i \notin MB_i)$. MB_i is said to be a Markov blanket for SF_i if and only if the condition in Equation (5.21) is satisfied.

$$P(SF - MB_i - \{SF_i\}, C|SF_i, MB_i) = P(SF - MB_i - \{SF_i\}, C|MB_i) \quad (5.21)$$

The above condition makes sure that MB_i includes both class information from SF_i and information about all other features. Koller and Sahami (1996) noted that optimal feature subset can be constructed using backward elimination method and termed this method as Markov Blanket filter and is described as follows.

Let G* be the current set of features and let G* = SF at the start. The feature SF_i is removed from G* if there exists a Markov blanket for it within G*. This guarantees that the removal of a feature do not render the previously removed features needed to be included in the optimal feature subset. Thus, from this property of MB, a redundant feature removed earlier remains redundant even when other features are removed.

It is assumed that a feature that has high class to feature relationship has more information regarding the class than the feature with low class to feature relationship (Shanbehzadesh and Yazdani, 2020). In order to determine whether a pair of features (SF_i, SF_j) is correlated with respect to their feature to feature relationship correlation value (SU_{ij}), the following steps are used.

When $SU_{j,c} \geq SU_{i,c}$, then an evaluation method that determines whether feature SF_j can be an approximate Markov blanket for feature SF_i (this will help to retain information regarding the class) is used. The threshold T is used to determine whether the SU_{ij} reflecting the feature to feature relationship is strong or not. Given two relevant features SF_i and SF_j ($i \neq j$), the feature SF_j is considered as an approximate Markov blanket of SF_i, if and only if both Equations (5.22) and (5.23) are satisfied.

$$SU_{j,c} \geq SU_{i,c} \quad (5.22)$$

$$SU_{i,j} \geq SU_{i,c} \quad (5.23)$$

The above two equations makes sure that in case of uncertainty redundancy between features SF_i and SF_j, SF_i is chosen over SF_j, if SF_i has more class information than SF_j. While considering uncertainty redundancy, the key point is to search approximate Markov blanket for the relevant features. This method utilizes the backward elimination method to make sure only the redundant features are removed.

However, this method has a small issue which can be best explained using the following example. Let SF_j be the only feature that forms an approximate Markov blanket for SF_i and let SF_k form an approximate MB for SF_j. After removing SF_i based on SF_j and later removal of SF_j based on SF_k will result in no approximate MB for SF_i in the current set. However, this situation can be avoided by removing a feature only when it can find an approximate Markov blanket formed by a predominant feature. If in the current set, a relevant feature does not have any approximate Markov blanket, then it is said be a predominant feature.

Since the goal is to find a small non-redundant feature subset and those features that form an approximate Markov blanket of feature SF_i are most likely to be more strongly correlated to SF_j, a candidate MB for SF_i is constructed by collecting all features that have the highest correlations with SF_i. The algorithm is given in Figure 5.5. This heuristic sequential method is far more efficient than methods that conduct an extensive combinatorial search over subsets of the feature set.

Initialize

- $G^* = SF$

Iterate

- For each feature $SF_i \in G^*$, let MB_i be the set of k features $SF_j \in G^* - \{SF_i\}$ for which the correlations between SF_i and SF_j are the highest.
- Compute $\Delta(SF_i|MB_i)$ for each i

Figure 5.5 : Markov Blanket Method

The algorithmic steps in MRMR is given in Figure 5.6, whose input is the training dataset (Tr) and a relevancy threshold (T =1) and successful completion outputs the optimal feature subset, F*'. In this algorithm, N denotes the total number of features, C denotes the set of class labels and $S_{candidates}$ denotes the set of candidate subsets constructed. This algorithm works in two stages. The first stage (Line Nos. 1-7) focuses on creating the $S_{candidates}$. The algorithm begins by calculating the SU values for each feature, from which the relevant features are selected and stored in $S_{candidates}$. This set is then arranged in descending order of SU.

```
// Stage 1 : Construct Candidate List
1.  for i = 1 to N do
2.      Estimate SUic for SFi
3.      if SUic > T
4.              SCandidates = SCandidates + SFi
5.      end if
6.  end for
7.  Arrange SCandidates in increasing order of SUic value

// Stage 2 : Find Optimal Feature Subset
8.  Fj = SCandidates(1)
9.  do  {
10.     Fi = getNextElement(SCandidates, Fj)
11.     if Fi ≠ NULL
12.     do {
13.             if SUi,j >= SUic
14.                     SCandidates = SCandidates - Fi
15.             end if
16.             Fi = getNextElement(SCandidates, Fi)
17.     }until(Fi == NULL)
18.     Fj = getNextElement(SCandidates, Fj)
19. }until (Fj == NULL)
20. F*' = SCandidates
```

Figure 5.6 : Steps of MRMR

The second stage (Line Nos. 8-19) uses the arranged $S_{candidates}$ to identify and select predominant features. A feature SF_j that has already been declared as predominant are used to remove other features for which SF_j forms an approximate MB. All the features with highest class to feature relationship do not have approximate MB and hence are

considered as predominant features. Thus, the iteration starts with the first element in $S_{Candidates.}$ For the remaining features, if SF_j forms an approximate MB for SF_i, then SF_i is removed from $S_{Candidates}$. This is repeated for all features. Then, the iterative steps are repeated for the next element of $S_{Candidates}$. The iteration is stopped when the set of predominant features is empty. Finally, the elements in $S_{Candidates}$ are returned as the selected optimal feature subset, F*' in Line No. 20.

5.2. MACHINE LEARNING ENSEMBLE CLASSIFICATION

After obtaining the optimal feature vector from feature engineering, the next step of ALL-C system is classification. As outlined in Chapter 3, Methodology, an ensemble classification system is proposed in this research work. The Ensemble Classifier (EC) is referred using several other names like fusion classifier and multiple classifier. It is mainly used to decrease the risk of associated with wrong selection of classifier for identifying Leukemia. The risk is reduced by aggregating the output of two or more classifiers called base classifier (Abuassba *et al.*, 2017). The usage of multiple classifiers offer additional degree of freedom in the classical bias/variance trade-off, so as to allow results that could not be obtained through the use of single classification system (Priya *et al.*, 2020; Smitha *et al.*, 2020). Figure 5.7 shows the general architecture of an EC system.

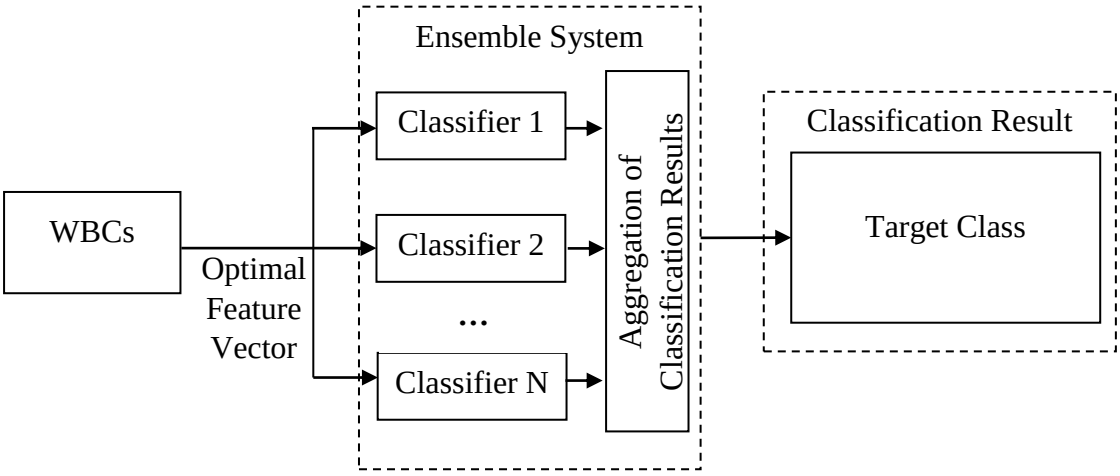


Figure 5.7 : General Architecture of an EC System

The construction of an ensemble system starts with the use of independently trained. The results from these classifiers are then combined or aggregated using statistical or algebraic algorithms. The performance of an EC system depends on two factors. The

first is the individual capability of the base classifiers and the second is base classifier diversity. A high diversity between base classifiers produce a high performing EC system. The EC system is constructed using three main steps, namely, base classifier, ensemble construction and classification result aggregation. The following section describes the working of these three steps.

5.2.1. Base Classifier

The vital step of an EC system construction is to decide on the base classifier used and number of base classifiers used. In this research work, the EC system is build using Support Vector Machine (SVM) classifier. The EC system is built to have 100 variants of SVM classifiers.

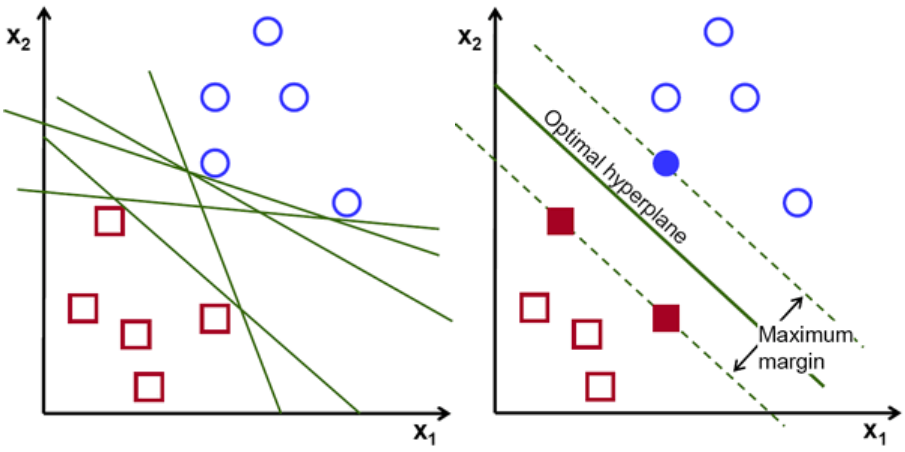
Support Vector Machines (SVMs) were introduced by Vapnik *et al.* (1995) and is a concept in computer science for a set of related supervised learning methods that analyze data and recognize patterns that are mainly used for classification and regression analysis. The standard SVM takes a set of input data and predicts, for each given input, which of two possible classes the input is a member of, which makes the SVM a non-probabilistic binary linear classifier.

Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

Although SVMs were originally designed as binary classifiers, approaches that address a multi-class problem as a single “all-together” optimization problem exist (Weston and Watkins *et al.*, 1999). A multi-class classification task usually involves separating data into training and testing sets. Each instance in the training set contains one ‘target value" (i.e. class labels) and several “attributes" (i.e. features). The goal of SVM is to produce a model (based on the training data) which predicts the target values of the test data given only the test data attributes. Mathematically SVM can be described as follows (Boser *et al.*, 1992; Cortes and Vapnik *et al.*, 1995).

Considering the binary classification case, let $((x_1, y_1) \ldots (x_n, y_n))$ be the training dataset where xi are the feature vectors that represent the observations and $y_i \in (-1, +1)$ be the two labels that each observation can be assigned to. From these observations, SVM builds an optimum hyperplane (a linear discriminant in the kernel transformed higher dimensional feature space) that maximally separates the two classes by the widest margin by minimizing the objective function. For a linearly separable set of 2D-points which belong to one of two classes, find a separating straight line is shown in Figure 5.8(a). In this example, there exist multiple straight lines that separate the data points into two groups. Deciding the optimal divider is an intuitive criterion.

In general, a line is considered bad if it passes too close to the points because it will be noise sensitive and it will not generalize correctly. Therefore, the goal here is to find the line passing as far as possible from all points. Thus, the operation of the SVM algorithm is based on finding the hyperplane that gives the largest minimum distance to the training examples. Twice, this distance receives the important name of margin within SVM's theory. Therefore, the optimal separating hyperplane *maximizes* the margin of the training data. Example of an optimal hyperplane is shown in Figure 5.8(b).



(a) **(b)**

Figure 5.8 : Support Vector Machine Hyperplane

The hyperplane computation method of SVM is described below. Let the hyperplane be defined as below (Equation 5.24).

$$f(x) = \beta_0 + \beta^T x \qquad (5.24)$$

where β is known as the weight factor and β_0 is the bias. The optimal hyperplane is represented in an infinite number of different ways by scaling of β and β_0.

As a matter of convention, among all the possible representations of the hyperplane, the one chosen is given by Equation (5.25).

$$|\beta_0 + \beta^T x| = 1 \tag{5.25}$$

where x symbolizes the training examples closest to the hyperplane. In general, the training examples that are closest to the hyperplane are called **support vectors** and this representation is known as the **canonical hyperplane.** Now, the result of geometry that gives the distance between a point x and a hyperplane $\{\beta, \beta_0\}$ is estimated using Equation (5.26).

$$\text{distance} = \frac{|\beta_0 + \beta^T x|}{\|\beta\|} \tag{5.26}$$

In particular, for the canonical hyperplane, the numerator is equal to one and the distance to the support vectors is Equation (5.27).

$$\text{distance}_{\text{support\_vectors}} = \frac{|\beta_0 + \beta^T x|}{\|\beta\|} = \frac{1}{\|\beta\|} \tag{5.27}$$

Let M denote the margin which is twice the distance to the closest examples (Equation 5.28).

$$M = \frac{2}{\|\beta\|} \tag{5.28}$$

Now, the problem of maximizing M is equivalent to the problem of minimizing a function $L(\beta)$ subject to some constraints. The constraints model the requirement for the hyperplane to classify correctly all the training examples x_i formally, it can be defined as Equation (5.29).

$$\min_{\beta,\beta_0} L(\beta) = \frac{1}{2}\|\beta\|^2 \qquad \text{subject to } y_i(\beta^T x_i + \beta_0) \geq 1 \ \forall i \tag{5.29}$$

Where, y_i represents each of the labels of the training examples.

5.2.2. Construction of EC System

The second step in the design of EC system is the construction of EC system using SVM classifiers. The multiple variants of SVM classifiers can be build using several methods like knowledge-based methods, randomization methods, training data manipulation methods, manipulation of input data/features and manipulation of target labels (Yildırım *et al.*, 2019). In this research work, training data manipulation method is used. For this purpose, the bagging (Bootstrap Aggregating) random subspace selection algorithm (Breiman *et al.*, 1996) is used.

In bagging, the training set is randomly sampled k times with replacement, producing k training sets with sizes equal to the original training set. Theoretical results show that the expected error of bagging has the same bias component as a single bootstrap replicate, while the variance component is reduced (Bhlman and Yu *et al.*, 2002).

Bagging is based on the idea of bootstrap sampling (Efron and Tibshirani *et al.*, 1993). A different training set is presented to each predictor, by sampling m samples with replacement from the original training set of m patterns. Each sample is called a bootstrap sample and hence its name (Hastie *et al.*, 2009). By sampling with replacement, some patterns will be repeated more than once, others will be absent, however, on the average, 63.2% of the original training set will be represented (Baszczyski *et al.*, 2009). After training all the individual predictors, the final ensemble output is the simple average of the component predictors outputs. Figure 5.9 presents the bagging procedure.

The performance of Bagging has been proved by several proposals (Hall *et al.*, 2003; Dietterich, 1999). It was also shown that it can perform better than other ensemble learning techniques and the main advantages of bagging are its simplicity in application and parallelizability. The individual classifiers are independent of each other and so they can be created simultaneously. During classification, the hybrid weighted majority voting scheme is used.

Inputs: Feature set S of m training features : $S = \{(x_i, y_i), i = 1, \ldots, m\}$, where x_i is the ith vector of features and y_i is the ith target label, LEARN: a learning algorithm, K: the number of classifiers to return.

Outputs: An ensemble E of K Classifier : $E = \{C_k, k = 1, \ldots, K\}$

Procedure:

1: **for** k = 1 : K **do**
2: create T_k {a bootstrap training set of size m}
3: C_k :=LEARN(T_k) {train the classier}
4: **end for**

Figure 5.9 : Bagging Algorithm

5.2.3. Aggregation Method

The final step of EC system construction is to combine the outputs of the various base classifiers. For this, a weighted majority voting algorithm, an enhanced version of the conventional majority voting algorithm, is used in this research work. The weighted majority voting algorithm is simple, yet effective and fast in aggregating the results of base classifiers.

Let the decision of the i^{th} classifier be defined as $d_{t,j} \in \{0, 1\}$, t = 1, …, T and j = 1, …, C, where T is the number of classifiers and C is the number of classes. If the i^{th} classifier chooses class ω_j, then $d_{t,j}$ = 1 and 0, otherwise. In majority voting scheme, a class ω_j is chosen according to Equation (5.30).

$$\sum_{t=1}^{T} d_{t,J} = \max_{j=1}^{c} \sum_{t=1}^{T} d_{t,j} \qquad (5.30)$$

The majority voting is an optimal combination rule under the minor assumptions of:

- An odd number of classifiers for a two class problem,
- The probability of each classifier choosing the correct class is p for any instance x, and
- The classifier outputs are independent.

Then, with majority voting, the fusion classifier makes the correct decision if at least $\lfloor T/2 \rfloor + 1$ classifiers choose the correct label, where the floor function $\lfloor . \rfloor$ returns the largest integer less than or equal to its argument. The accuracy of the fusion classifier can be represented by the binomial distribution as the total probability of choosing $k \geq \lfloor T/2 \rfloor + 1$ successful ones out T of classifiers, where each classifier has the success rate of p. Hence, P_{ens}, the probability of fusion classification success is given as in Equation (5.31).

$$P_{ens} = \sum_{k=(T/1)+1}^{T} \binom{T}{k} p^k (1-p)^{T-k} \tag{5.31}$$

In majority voting, each classifier has the same weight and voting by ensemble members determines the final result. Usually, it takes over half of ensemble members to agree in order for a result to be accepted as the final output of the ensemble, regardless of diversity and accuracy of each classifier's generalizability. The following explains the method used to combine the weighting scheme with majority voting.

Let the decision of the ith classifier be defined as d_t, $j \in \{0, 1\}$, t = 1, …, T and j = 1, …, C, where T is the number of classifiers and C is the number of classes. If the ith classifier chooses class ω, then $d_{t,j} = 1$ and 0, otherwise. In majority voting scheme, a class ω_j is chosen, using Equation (5.32)

$$\sum_{t=1}^{T} d_{t,J} = \max_{j=1}^{c} \sum_{t=1}^{T} d_{t,j} * w^t \tag{5.32}$$

Here w_t is the weight assigned to the classifier t and is calculated using Kuncheva (2004) method (Equation 5.33).

$$w^t = \log \frac{p^t}{1-p^t} \tag{5.33}$$

5.2.4. Design of EC System

The EC system that uses SVM as base classifier is designed as stated below. Let EC be the ensemble system, which is denoted as a group of classifiers, that is, EC = {BC_1,

…, BC_N}. Here, N is the total number of base classifiers used and is set to 100 in this research. Let F denote the optimal feature set obtained after using the MRMR feature selection algorithm. The bagging feature subspace selection algorithm is then used to create 100 versions of F. Each of this version of feature vector is divided into training and testing feature sets using hole-out partitioning method. Let the training set be denoted as Tr, whose elements are a combination of features and its corresponding target label, that is, Tr = {(Tr_1, TL), …, (Tr_n, TL). Here TR_i denotes the ith feature in Tr, with n denoting the number of instances in the training set. Further, TL denotes the pre-defined target label set which is equal to {0, 1, 2, 3} indicating {normal, L1, L2, L3} in this research work. Each classifier in EC is trained using Tr.

Let Te denote a new testing optimal feature set, whose target label has to be found. The Te is supplied to each trained classifier which outputs a target label, R_{TL}, which is a subset of TL. The 100 results, thus obtained, are aggregated using the weighted majority voting algorithm, to obtain the final TL of Te. The details of the proposed EC system is given in Table 5.4.

TABLE 5.4

DETAILS OF EC SYSTEM

Factors	Details
Base Classifier Used	SVM
No. of Base Learning Algorithms	100
Ensemble Creation Methods	Bagging Feature Subspace Selection Technique
Partitioning Method Used	Hold-Out Method
Aggregation Method	Weighted Majority Voting Algorithm

5.3. ENHANCED EC SYSTEMS

The underlying notion of EC systems is that a robust and highly accurate classification can be achieved by while considering multiple views of the same problem.

This is proven by several theoretic and empirical studies, which has proved that, an amalgamation of multiple individual classifiers is advantageous in terms of accuracy and the stability of classifiers (Nti *et al.*, 2020). Several studies have also proved that the performance of the EC systems are high when compared to single classification systems (Chung and Teo *et al.*, 2023; Ekman and Harnet *et al.*, 2022). The main issue in EC systems is its time complexity. Researchers have been attracted by the area of reducing the time complexity of the EC system, while maintaining or improving classification accuracy (Abdar *et al.*, 2019). In this research work, to decrease the time complexity, an additional step called, 'Classifier Selection' is included in the architecture of the EC system.

The accuracy of the EC system depends on the quality of the training set. In this research work, the quality of the feature set is improved by the use of the MRMR feature selection algorithm. To further improve the quality, an optimization algorithm that improves the quality of the training set is also proposed. The modified architecture of the EC system, after including the classifier selection algorithm and training set optimization algorithm, is shown in Figure 5.10.

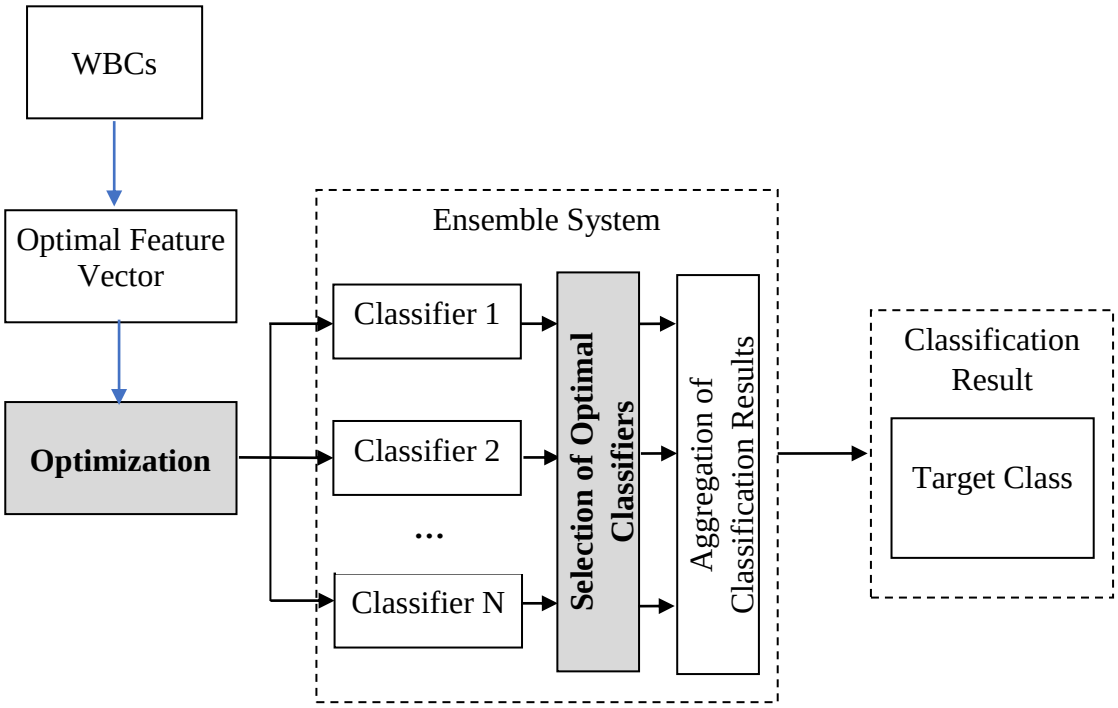


Figure 5.10 : General Architecture of the Modified EC System

5.3.1. Training Set Optimization Algorithm

The optimal feature set is divided into training and testing sets and the optimization algorithm is used to improve the quality of the training set, which can improve the accuracy of the classifier. A training set may have data that have very little impact on the classification process. Some features present in the training set might even increase the error rate. To avoid these scenarios, the optimization algorithm is used. First the training set is used to train the SVM classifier. Then, the test set is used to evaluate the trained classifier.

The classifier results are reported in the form of confusion matrix (Sokolova and Lapalme *et al.*, 2009; Demsar *et al.*, 2006). The confusion matrix is defined as a cross table that records the number of occurrences between two situations. The first is the true/actual classification and the second is the predicted classification. An example of a 2-class confusion table is shown in Figure 5.11. Here, a is the number of correct classification that an instance is negative (True Positives), b is number of incorrect classifications that an instance is positive (False Negatives), c is the number of incorrect classification that an instance is negative (False Positives) and d is the number of correct classifications that an instance is positive (True Negatives). Here, positive refers to 'correctly identified' and 'correctly rejected' features, while negative refers to 'wrongly accepted' and 'wrongly rejected' feature data.

		Predicted	
		Negative	Positive
Actual	Negative	a	b
	Positive	c	d

Figure 5.11 : Confusion Matrix

The optimization algorithm constructs a new training set by collecting only true positives and true negatives and removing the wrongly classified features as noisy features. The newly created training set has features with maximum discriminating capacity and therefore, the classifier training using this new set can improve the accuracy of the classifier. The process is presented in Figure 5.12.

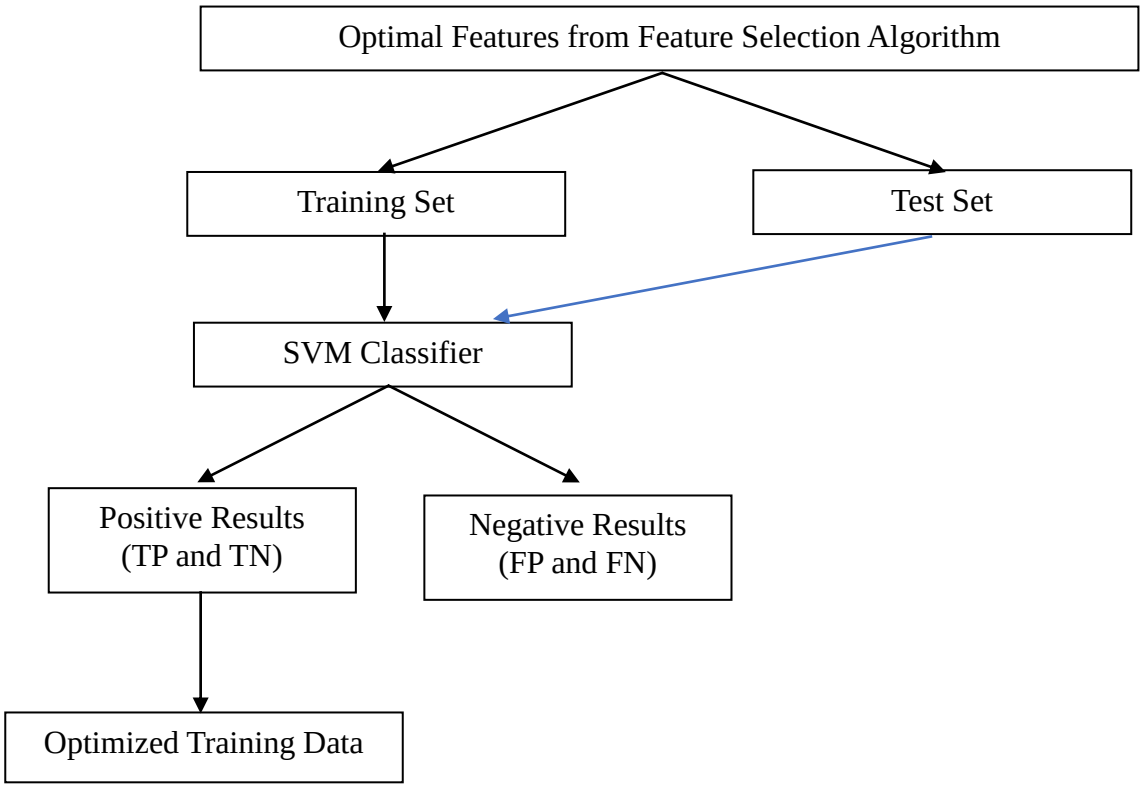


Figure 5.12 : Optimizing Training Feature Set

5.3.2. Selection of Optimal Base Classifiers

The second enhancement method, called as selection, is used to improve the performance of EC system by selecting optimal base classifiers that can improve the classification process, without compromising on accuracy. The selection step also helps to reduce the time complexity of the EC system.

The selection step adopts either a static or dynamic method to identify optimal base classifiers. The static method is used in the training phase, while dynamic method is used during testing phase. The static method uses a pruning algorithm to identify a set of best performing base classifiers. On the other hand, the dynamic method considers each base classifier as an expert in distinct regions of the feature space and selects the most competent base classifiers in the local region where the new input test feature is located. The dynamic method can be implemented in two ways. The first method, called Dynamic Classifier Selection (DCS), works to identify a single classifier from the set of base classifiers to obtain the final classification out (Figure 5.13). The second manner of dynamic selection is called Dynamic Ensemble Selection (DES). The DES works to

identify a subset of best performing base classifiers from the original set of base classifiers, which are ensembled to obtain the final classification result (Figure 5.14).

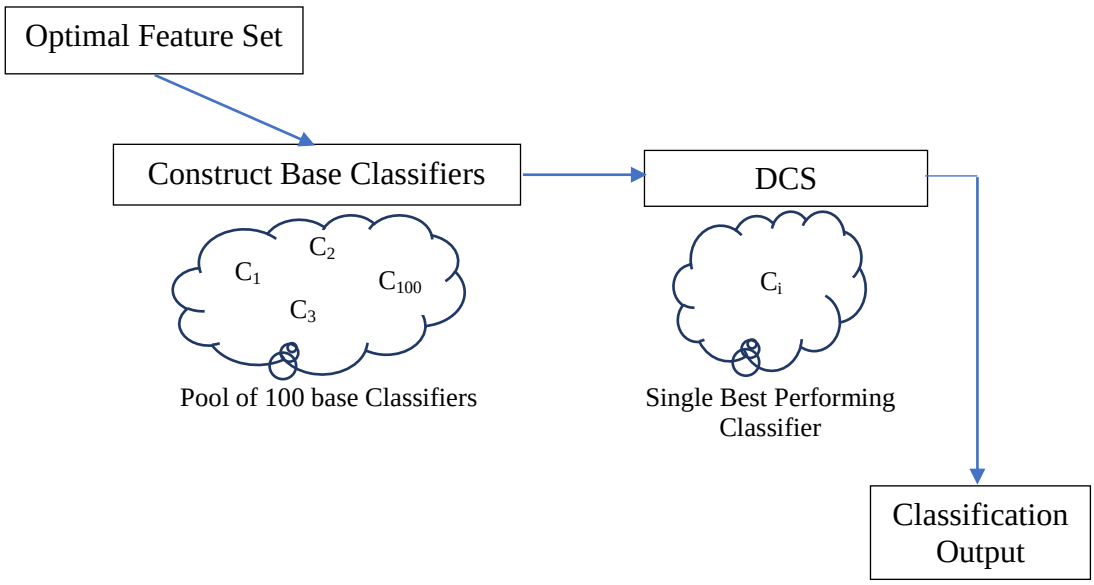


Figure 5.13 : Usage of DCS in EC System

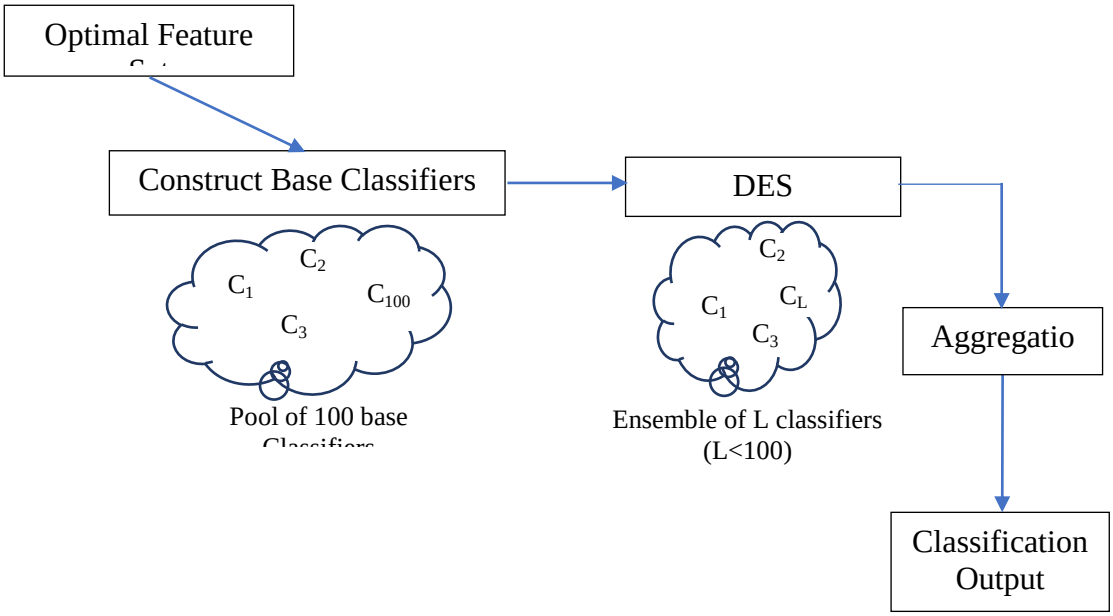


Figure 5.14 : Usage of DES in EC System

This research work analyzes both the methods in their efficiency to identify ALL. Both static and dynamic selection methods have been proved to be successful during classification and this research work proposes two EC systems that combine static and

dynamic selection methods. The two EC systems differ in the manner of combining static selection with the two types of dynamic selection methods. The two proposed systems are

(i) EC system using static and DCS method, and

(ii) EC system using static and DES method.

Both the proposed system performs two steps to perform classification. The first step uses static pruning algorithm to select optimal classifiers from the set of base classifiers. The resultant set is then used by the dynamic classifier selection or dynamic ensemble selection algorithm, to obtain the final classification result. These steps are projected in Figure 5.15.

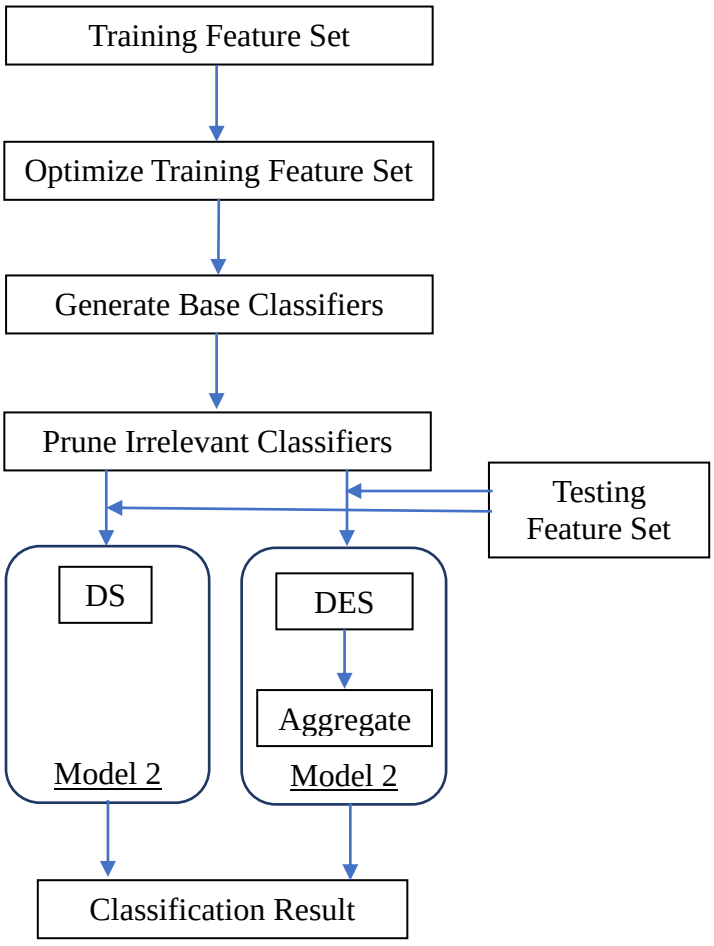


Figure 5.15 : Proposed Enhanced Ensemble Classification Systems

The refined training step obtained after the application of the optimization algorithm is used by the bagging algorithm to produce 100 feature vectors. These feature

vectors are used to generate 100 SVM classifiers. Let C = {C_1, …, C_{100}} denote this scenario. Next, the static selection is performed using a pruning algorithm. This retains only those classifiers that have positive impact on ALL classification. Consequently, this step produces a set of Optimal Classifiers, C_O = {C_1…C_L}, where $C_O \in C$ and size (C_O) < size (C), that is L < 100. When C_O is used as input by DCS algorithm, it selects one best classifier ($C_j \in C_O$) from L classifiers in C_O. The final reported ALL classification result is produced by C_j. When C_O is used as input by DES algorithm, R best classifiers are selected from C_O. That is, it selects an ensemble of Classifiers, C_k (k=1..15), $C_k \in C_O$, size(C_k) < size(C_O), that K < L. The results of the C_k classifiers is aggregated using weighted majority voting algorithm to produce ALL classification result. The following subsections describes static selection, DCS and DES in detail.

(i) Static Selection

The main objective of static selection algorithm is to select a subset of base classifiers using a pruning algorithm during the training phase. The usage of pruning algorithm offers three main interrelated advantages during classification. They are,

(i) Reduces the high RAM requirement of the EC system,

(ii) Reduces the time complexity of the EC system, and

(iii) Reduces the size of the EC system.

The design of the proposed enhanced EC system uses static pruning algorithm as a preprocessing step that selects candidate optimal base classifiers that can improve EC system performance (Margineantu and Dietterich *et al.*, 1997; Zhang *et al.*, 2006). The subset of base classifier, thus obtained, should be able to produce the same advantages of the full EC system apart from the advantages listed above.

The static pruning algorithm used in this research work is designed to have two major steps. The first step uses classification error rate (Equation 5.34) as a threshold to select optimal classifiers. All the base classifiers are arranged in ascending order of error rate and the top 30 classifiers are selected as best performing classifier. Let this set of classifiers selected be denoted as {C_s}

$$\text{Error Rate} = \frac{\text{False Positive} + \text{False Negative}}{\text{True Positive} + \text{True Negative} + \text{False Positive} + \text{False Negative}} \quad (5.34)$$

In the second step, the kappa statistic pruning technique is used to select the final set of optimal classifiers, $\{C_O\}$, which is a subset of $\{C_s\}$. The kappa statistic (Martinez, 2011) is a technique that compares the observed accuracy to an expected accuracy based on the number of instances in each class. This measure is used to get performance information of a classifier when compared to a model that classifies instances at random according to the frequency of each class.

The kappa pruning technique selects a set of classifiers to form an ensemble based on their pairwise diversity using k-statistic. The k coefficients of a pair of classifiers h_α and h_β is calculated using the estimated probability that the classifiers coincide in the classification of an instance, Θ_1 and the estimated probability that the classifiers coincide by chance in the classification of an instance, Θ_2. Both the statistics are defined using Equation (5.35).

$$\Theta_1 = \frac{1}{m}\sum_{i}^{l} C_{ii} \qquad \Theta_2 = \sum_{i=1}^{l}\left(\frac{1}{m}\sum_{i}^{1} C_{ij}\right)\left(\frac{1}{m}\sum_{i}^{1} C_{ji}\right) \quad (5.35)$$

In the above equation, C_{ij} is the number of instances in the training set for which $h_\alpha = y_i$ and $h_\beta = y_j$. The kappa statistic (Equation 5.36) is then used to measure the agreement between the classifiers.

$$k = \frac{\Theta_1 - \Theta_2}{1 - \Theta_2} \quad (5.36)$$

The kappa returns a value that can be interpreted using Table 5.5 (Landis and Koch *et al.*, 1977). This technique incorporates pairs of classifiers to the sub-ensemble with minimum value of kappa, until t hypothesis (Equation 5.37) have been selected.

$$s_u = \arg\min_{k} k(h_k, H_{S_{u-1}}) \quad (5.37)$$

The steps involved during static selection using pruning algorithm is presented in Figure 5.16.

TABLE 5.5

KAPPA INTERPRETATION

Kappa	Interpretation
<0	Poor Agreement
0.0 – 0.60	Moderate Agreement
0.61-0.99	Among Perfect Agreement
1	Perfect Agreement

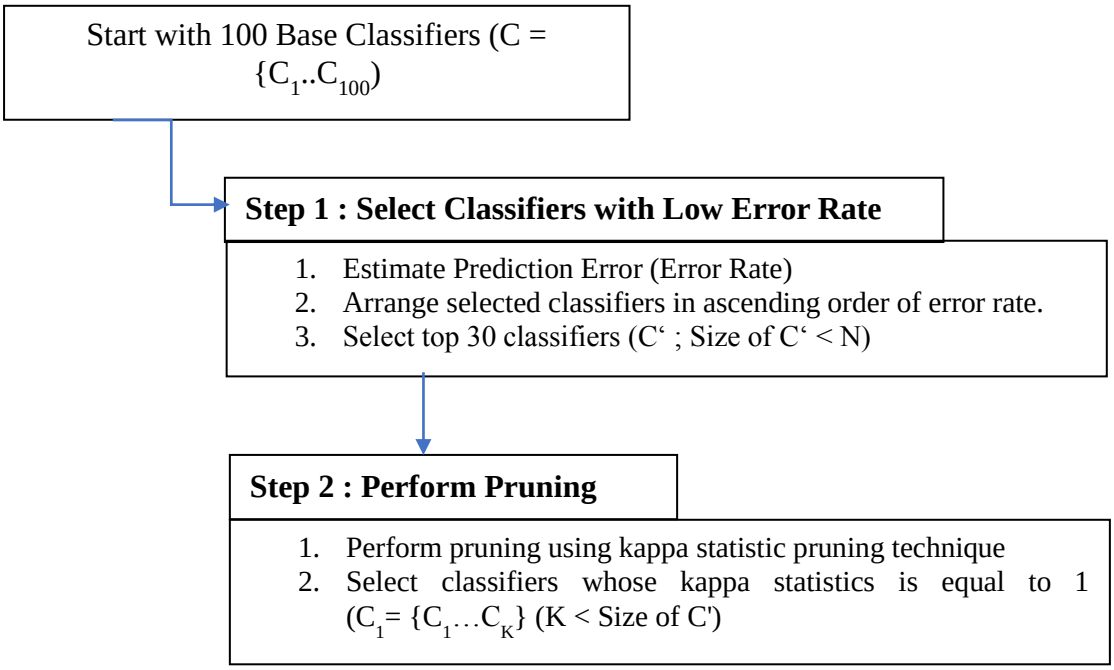


Figure 5.16 : Static Selection Using Pruning Algorithm

(ii) Dynamic Classifier and Dynamic Ensemble Selection

The output of the previous step is a set of classifiers $C_O = \{C_1...C_{30}\}$, which forms the initial set of base classifier. The next step of the proposed hybrid EC system is the dynamic selection of classifiers. In dynamic selection, the classification of a test data is done in three steps.

- Step 1 : Region of Competence (RoC) Identification. Here, a local region surrounding the test data (Tei), is used to estimate the competence level of the base classifier,

- Step 2 : Determine Selection Criteria. This criteria is used to estimate the competence level of the base classifiers. In this research work, the criteria used is the classification accuracy, and
- Step 3 : Determine selection mechanism, that is, DES or DCS.

The first step is the most important part of dynamic selection algorithm. The local regions are defined using the K-Means clustering algorithm (Rishabh *et al.*, 2022). The RoC identification starts with the application of K-Means on the training feature set to obtain K clusters (C_L = { C_{L1} ... C_{Lk}}) and its corresponding centroids (U = {u_1...u_k}). Here, K is set to 4 as we have three groups (Normal, L1, L2 and L3) regions to classify. The next step performs clustering. It estimates CL_i^* using Equation (5.38) with respect to a test feature set (F_i) and selects all features in CL_{i*} as the validation set. Using accuracy as performance, the algorithm selects N classifiers from C and the resultant set of classifiers is denoted as C_O'.

$$i^* = \underset{1 \le i \le k}{\operatorname{argmin}} \| F_i - u_i \| \tag{5.38}$$

Once the training set has been cluster, the algorithm searches for clusters that is nearest to the current test feature set. These nearest clusters are the RoCs for this test data. This region is used to determine the competence level of all the base classifiers.

When supplied with a new feature set, the algorithm estimates the distance between the new test feature set and centroid of each cluster. Next for all clusters the competence of the base classifiers are determined. The algorithm works on the notion that classifiers with high accuracy are best classifier and can improve the performance of the EC system. Accordingly, the algorithm selects all classifiers with high classification accuracy. As mentioned earlier, the DCS algorithm select one classifier with the highest accuracy, while DES selects a top N classifiers with high accuracy. In this research work, N is set to 15. The algorithm used during best classifier selection is summarized in Figure 5.17, where the final best performing classifier(s) is stored in C_R.

Set of base classifiers defined by RoC $\rightarrow C_O'$ and Classification accuracy $\rightarrow$ A

Step 1 : $C_R = \{\ \}$

Step 2 : Arrange base classifiers in C_0' in descending order in terms of accuracy

Step 3 : $C_R = C_R + \{C_{O1}'\}$ // if selection method is DCS

(or)

$C_R = C_R + \{C_{O1}'.. C_{O15}'\}$ // if selection method is DES

Return C_R

Figure 5.17 : Best Base Classifier Selection Procedure

The details of the proposed enhanced EC systems are presented in Table 5.6.

TABLE 5.6

DETAILS OF ENHANCED EC SYSTEMS

Factors	Details
Base Classifier Used	SVM
Ensemble Creation Methods	Bagging Algorithm
No. of Base Learning Algorithms	100
Partitioning Method Used	Hold-Out Method
No. of Base Learning Algorithms after Static Pruning	30
No. of Base Learning Algorithms after DCS Pruning	1
No. of Base Learning Algorithms after DES Pruning and Aggregation Method Used	15 and Weighted Majority Voting Algorithm

5.4. CHAPTER SUMMARY

This chapter described the proposed enhanced EC system used to improve the ALL-C system. The improvements were incorporated through the use of multiple features, feature selection algorithm and enhanced EC systems. The classifier used is the SVM machine learning classifier. Another classification group that has gained wide acceptance is deep learning classifier. The next phase of the research work is focused on analysing and enhancing deep learning classifier to improve the ALL-C system. The following chapter, Chapter 6, Classification Using Deep Learning Classifier, describes deep learning classifier along with the methods used to enhance its performance.