CHAPTER - 4

Literature Review

In the previous chapter, various applications of Machine Learning (ML) techniques in different agriculture domains were discussed and also, it can be inferred from Chapter 1 and 2 that the focus of this research is on the application of ML techniques to discover knowledge from agricultural soil health card database. This chapter comprises a detailed literature review of the ML classifier technique called k-Nearest Neighbour (k-NN) which is chosen for the classification problem related to this research work. Moreover, the prototype generation and selection methods which are targeted at performance improvement of k-NN are reviewed.

In Chapter 2: Section 2.3.1, the brief introduction is given about nearest neighbour classifier which is now elaborated in the following section as a k-NN classifier.

4.1 k-Nearest Neighbor classification

4.1.1 Concept

It is essential to know the type of data for the classification task and these data can be of two types, either parametric or non-parametric. For parametric data, there is some type of statistical distribution exists between its instances which is identified previously and vice versa. Such information is very important to know because according to data the various algorithms and techniques can be appropriately applied. One of the unique algorithms for Machine learning (ML) is k-Nearest Neighbor (k-NN), which assumes that the class of an instance is the same as the class of the nearest instance. The k nearest neighbour rule is one of the most simple non-parametric decision rules, which applies a similarity metric to measure the propinquity of an instance to the other instances. In k-NN, the assumption is that instances in the data are not depending on each other, they are identically distributed. Hence, the instances, which are close to the given instance, have the same class (Cover et al. 1967).
The $k$-NN classification many times called as lazy learning method and it is one of the instance-based learning approaches. “$k$-NN is purely lazy, it simply stores the entire training set and postpones all efforts towards inductive generalization until classification time” (Wettschereck et al. 1997) In Chapter 1: Section 1.3, we have discussed the motivation for using a $k$-NN classifier for this research.

Three properties of instance-based learning algorithms:

- During the learning process, they store all of the training data.
- It searches for a case that is similar to the new case from the training data.
- Until a new case value is predicted, the generalization outside the training data is deferred because any new search is responded by comparing the new case to the training data.

4.1.2 The $k$-NN rule

In $k$-NN, each instance is defined by a number of attributes, and all the instances inside the data are represented by the same number of attributes. Although there may be some missing attribute values. One of these attributes is called the class attribute, which contains the class value (label) of the data whose values are predicted for new, unseen instances.

The nearest neighbour rule of $1$-NN assumes that the class value of the immediate neighbour to be the class of the new instance. Instead of $1$-NN, the $k$-NN rule is used to assigns an instance the class label which is represented mainly in its $k$ neighbours. Here, $k$ is a number of its neighbours, $k = 1, 2, 3 \ldots n$.

The training instances nearness is the nearness of the well-established neighbours and the new example. The attribute values of the nearest training instances who is alike to that of the new instance are calculated. Though it may happen that the precise similar instance is not found, hence the nearest instance can be the one with minimum dissimilarity.

In $k$-NN from available space, only a small volume of the attributes it considered and the new case is taken as the centre of this space volume. The radius of this volume is the distance from the new case to the $k^{th}$ nearest neighbour. The probability of the new case is estimated to fit into a certain class and it is derived from the relative frequencies of the classes of the training cases in this volume. The highest estimated probability of the class is assigned to the new class (Hand et al. 2001).
FIGURE 4.1 1-NN (a), 2-NN (b) and 3-NN (c). “+” and “-” are cases of positive and negative classes and “x” represents the new case (Tan et al. 2007).

In its basic form where \( k = 1 \), see Fig. 4.1 2(a), the result is unbalanced because of its high variance and sensitiveness and hence it is generally not used (Hand et al. 2001), therefore larger value of \( k \) is used. Sometimes different values of \( k \) which depends on different distribution analysis on the data are used. In Fig. 4.1 (b), it can be seen that a tie condition in the case of two neighbours is nominated from two different classes. In Fig. 4.1(c), the value of \( k = 3 \), which is based on random selection.

4.1.3 Proximity measures

It was discussed in the previous section, that the nearest neighbour algorithm requires the distance between training instances and a new instance. The notion of “distance” is subjected to the availability of data, means unrelated types of data have different techniques for finding out the distance. Already, we have discussed types of data in Chapter 2: Section 2.1. Further, the notion of following two types of data is deliberated in connection with proximity measures for the \( k\)-NN classifier.

- Homogeneous data: It is a special case in which all the attributes are of the same type i.e. nominal or interval type of attributes.

- Heterogeneous data: Data in which there are different types of attributes. For example, one attribute is nominal while the other is an interval.
4.1.3.1 Proximity measures for homogeneous data

For homogeneous data, the proximity measures define the distance measures, a metric is a dissimilarity function that fulfils four properties (Hand et al. 2001).

1. \(d(a, b) \geq 0\) (for each a and b distances are nonnegative numbers)
2. \(d(a, a) = 0\) (distance of an object to itself is zero) (also called reflexivity)
3. \(d(a, b) = d(b, a)\) (symmetric)
4. \(d(a, b) \leq d(a, c) + d(c, b)\) (triangle inequality: Going directly from a to b is shorter than making a detour over object c.)

The proper distance function is very important for a good learning system. There are variety of distance functions being traditionally used (Wilson et al. 1997). Let \(x\) and \(y\) are two input instances and \(m\) is the number of attributes of training and test records for which we require to calculate as follows:

- **Euclidian Distance Similarity Measure**: It is straight line distance between two points in Euclidian space. It is derived from Pythagoras metric (Deza et al. 2006), Eq. 4.2.

\[
D(X, Y) = \sqrt{\sum_{i=1}^{m} (x_i - y_i)^2} \tag{4.2}
\]

- **Manhattan Distance Similarity Measure**: It is the distance between two points of city road grid and it is also known as city block distance (Zezula et al. 2006). It is used to examine the absolute difference between two points of the object. As mentioned in the Eq. 4.3, the Manhattan distance is the simple sum of the difference between the horizontal and vertical components:

\[
D(X, Y) = \sum_{i=1}^{m} |x_i - y_i| \tag{4.3}
\]

- **Minkowski Similarity Measure**: This distance measure is a generalization of the Euclidian and Manhattan similarity measure as shown in Eq. 4.4 (Charulatha et al. 2013).

\[
D(X, Y) = \left( \sum_{i=1}^{m} |x_i - y_i|^r \right)^{1/r} \tag{4.4}
\]
Proximity measures

- Canberra Similarity Measure: Is it weighed version of Manhattan similarity measure, it is used for data scattered around the origin (Charulatha et al. 2013) as shown in Eq. 4.5.

\[ D(X,Y) = \sum_{i=1}^{m} \frac{|x_i - y_i|}{|x_i + y_i|} \]  \hspace{1cm} (4.5)

- Chebyshev Similarity Measure: It is defined on a vector space where the distance between vectors is the greatest of their differences along any coordinate dimension as presented in Eq. 4.6 (Wilson et al. 1997).

\[ D(X,Y) = \max_{i=1}^{m} |x_i - y_i| \]  \hspace{1cm} (4.6)

- Cosine Similarity Measure: It measures the cosine angle between two non-zero vectors of an inner product space. If two vectors are having same orientation, means cosine similarity of 1, and if the orientation is at 90 means similarity 0. Diametrically opposite vectors have a similarity 0 (Zezula et al. 2006) as shown in Eq. 4.7.

\[ S_{\cos} = \frac{\sum_{i=1}^{m} x_i y_i}{\sqrt{\sum_{i=1}^{m} x_i^2} \sqrt{\sum_{i=1}^{m} y_i^2}} \]  \hspace{1cm} (4.7)

- Correlation Similarity Measure: To quantify the correlation between similarity measures, a correlation similarity is used. Strength and direction between two distance measures are indicated by it. If the value gets close to 1, it represents a good fit, i.e., two distance measures are semantically similar (Gavin et al. 2003). As indicated in Eq. 4.8. Correlation coefficient approaches zero when the fit gets worse. When either two distance or two similarity measures are compared, the correlation coefficient is a positive value.

\[ D(X,Y) = \frac{\sum_{i=1}^{m} (x_i - \mu_x)(y_i - \mu_y)}{\sqrt{\sum_{i=1}^{m} (x_i - \mu_x)^2 \sum_{i=1}^{m} (y_i - \mu_y)^2}} \]  \hspace{1cm} (4.8)

- Chi-square Similarity Measure: The Chi-square distance is calculated on relative counts, and it is standardized by the mean and not by the variance (Ibrahimov et al. 2002), Eq. 4.9.
\[ D(X,Y) = \sum_{i=1}^{m} \frac{1}{\text{sum}_x} \left( \frac{x_i}{\text{size}_x} - \frac{y_i}{\text{size}_y} \right)^2 \] (4.9)

- I-divergence Similarity Measure: It is usually used in a positive, linear inverse problem, and it is a kind of distance metric showing the difference between measured value and true value (Meng et al. 2014), Eq. 4.10.

\[ D(X,Y) = \sum_{i=1}^{m} y_i^i \log \frac{y_i^i}{x_i^i} - y_i^i + x_i^i \] (4.10)

4.1.3.2 Proximity measures for heterogeneous data: Distance functions for heterogeneous datasets have the ability to support both types of attributes. The *Heterogeneous Euclidean-Overlap Metric (HEOM*) (Wilson et al. 1997), (Eq. 4.11), is one such distance function which uses different functions applicable to different types of attributes, means for nominal attributes it uses overlap function and for ordinal and quantitative attributes is uses normalized Euclidean distance. The distance between two input vectors \( x \) and \( y \) is given by the Eq. 4.11.

\[ HEOM (x, y) = \sqrt{\sum_{a=1}^{m} d(x_a, y_a)^2} \] (4.11)

where, \( a \) stand for an attribute and \( m \) stands for the total number of attributes. The distance between two values of input vectors \( x \) and \( y \) of a given attribute \( a \) is given by, Eq. 4.12

\[ 1, (\text{if } x_a \text{ or } y_a \text{ is unknown}) \]

\[ d(x_a, y_a) = \text{overlap}(x_a, y_a), \text{ (if } a \text{ is nominal}) \] (4.12)

\[ \text{rn_diff}(x_a, y_a) \]

For the unknown input vectors attributes, the distance value returned would be maximum. The overlap function is defined as, Eq. 4.13

\[ d(x_a, y_a) = 0, (\text{if } x_a = y_a) \]

\[ 1, (\text{otherwise}) \] (4.13)

and the range-normalized difference function is Eq. 4.14

\[ \text{rn_diff}(x_a, y_a) = \frac{|x_a - y_a|}{\text{range}_a} \] (4.14)
The range is used to normalize the attribute values and is defined as Eq. 4.15

\[ \text{range}_a = \max_a - \min_a \]  

Here, from the training set, the values of \textit{maximum} (\(\max_a\)) and \textit{minimum} (\(\min_a\)) of the attribute are taken. The square root of the sum of the squared distances of all the attributes is the distance between two input vectors.

4.2 Advantages and disadvantages of k-NN

The main advantage of \(k\)-NN for classification is that, it is non-parametric, i.e. the distribution of the data does not need to be known. Also, it is theoretically and computational quite simple because it is based on distances; it is multiclass; it does not assume a linear separability of the data and it is very stable. If there are small changes in the training data do not lead to significantly different classification results (Breiman et al. 1996). It can learn from a small set of objects and can incrementally add new information and give competitive performance (Bay 1996). The limitations of \(k\)-NN are that it does not achieve well if the classes are unbalanced, i.e. if a number of the objects in the training classes is very dissimilar from one class to another, because it increases the probability of finding nearest neighbours fitting to the class with the largest number of objects. Also, it is sensitive to the \(k\) value (Coomans et al. 1982) which must be optimized. Although several probabilistic methods are known for \(k\)-NN, they are not used to provide the consistency of the classification for a particular object (Yuan et al. 2004). One reason is that probabilistic methodologies only work well when the number of training objects is very large (Lavine et al. 2012). Another important restriction of the \(k\)-NN method is the course of dimensionality which suggests the peaking phenomenon, i.e. for a constant number of objects. The peak of classification accuracy decreases when the number of variables increases (Fukunaga et al. 1989, Reunanen et al. 2004, Sima et al. 2008). This can be escaped by using a large number of objects or by reducing the dimensionality of the data (Yang et al. 2010, Yang et al. 2011, Villagas et al. 2011).
4.3 Accelerating $k$-NN

In ML literature, the $k$-NN classifier is considered as a powerful and widely used nonparametric technique for classification. Though it is exhaustive to perform a $k$-NN search which requires a lot of computational resources in case there is a large training data set, in this case, $k$-NN is not preferable (Chang et al. 1974, Ritter et al. 1975). Since many decades accelerating the $k$-NN search, is one of the active areas of research.

To speed up the $k$-NN searching is an interesting area of research and it is mainly divided into two categories: template condensation and template reorganization (Zhang et al. 2004). Template condensation identifies the redundant patterns in template set and removes it (Chang et al. 1974, Ritter et al. 1975). While the restructuring of templates is done in the template reorganization algorithms (Broder et al. 1990, Faragó et al. 1993, Kim et al. 1986, Ruiz 1986). Lot of work has been done to find a new approach and in one such method, the classification performance is not affected while reducing the storage and computation cost (Derrac et al. 2012).

In some method out of total training set, representative samples are selected and remaining ones are deleted to reduce the amount of training sample set. In text categorization research (Wang et al. 2010), the training set is reduced based on the density. Here text density is calculated and if it is found bigger than the average density then removes some samples to reduce training samples in the training set. Some research has extended the features affecting the $k$-NN performance, the best $k$ value, the training sample size, etc. Majumdar and Ward (Majumdar et al. 2010) combined the $k$-NN classifier with the random projection technique. Ghosh et al. 2006, estimated the optimal value of the $k$ in $k$-NN.

Hu et al. 2011, applied sample weight learning on the nearest neighbour classifier. Domeniconi et al. 2005, studied theoretically the large margin nearest neighbour classifiers. Parthasarathy et al. 1990, explored the way to use $k$-NN in case sample size is small. Some researchers have analyzed the data point relationships to the nearest neighbour relationships, like the centres of the classes and hyperplane data points. Gao et al. (2007) have designed a nearest neighbour classifier based on the centre called the centre base nearest neighbour classifier. Li et al. (2008) used the local probabilistic centres of each class in the classification process. Vincent et al. 2002, applied the $k$-local hyperplane NN technique.
In some research work, researchers have explored the efficiency of the *k*-NN classifier. Hernández-Rodríguez et al. (2010) has proposed $n$ approximate fast $k$ most similar neighbour classifier based on a tree structure and checked the efficiency of the *k*-NN classifier. Zhang et al. 2004, explored cluster based tree algorithms for the fast *k*-NN classifier. Ghosh et al. 2005, explored the visualization and aggregation of nearest neighbour classifiers. Some research work explored the distance metrics. Derrac et al. 2012, proposed a method to improve the performance of the *k*-NN classifier based on cooperative coevolution. Triguero et al. 2011, adopted the differential evolution to optimize the positioning of the prototypes to address the limitations of the nearest neighbour classifier. Yu et al. 2006, presented adaptive *k*- nearest neighbor classifier.

### 4.4 Variations of *k*-NN

With the aim of improving classification performance, several variations of *k*-NN have been proposed.

#### 4.4.1 Changes in the metric used to find the neighbors

The metric affects the outcomes of *k*-NN. When different metrics were tested (Euclidean, Manhattan, Cosine coefficient, Canberra, Lance-Williams and Lagrange), i.e. the Lance-Williams, Manhattan and Canberra gave comparable classification error rates and, in some cases, *k*-NN gave better results than *LDA* (Todeschini et al. 1989). In agriculture domain, classification of soil samples into specific fertilizer deficiency, where values of input vectors are positive, the *I*-divergence distance measure is the best similarity measure in terms of time and accuracy (Prajapati et al. 2016). The next best performance reported here is achieved by Cosine, Correlation and Euclidean in terms of accuracy.

#### 4.4.2 Variable reduction

Several methods have been applied before classifying with *k*-NN with aim of reducing the dimensionality of a data matrix specifically to remove the uninformative variables that can affect negatively the classification results (Villegas et al. 2011, Wu et al. 1997). For example, local *PCA* (for each individual class) or global *PCA* (for entire training dataset) have been used before the classification with *k*-NN (Parveen et al. 2006, He et al. 2008). The *Multi-label dimensionality reduction method (MDDM)* has also been used. *MDDM* attempts to project the
actual data into a lower-dimensional feature space exploiting the dependence between the original feature description and the associated class labels (Zhang et al. 2010).

4.4.3 Combination with other classifiers

To obtain more accurate classification, sometimes the combination of two or more classifiers is done at the cost of increasing their complexity (Kuncheva et al. 2004). $k$-NN has been joined in three different ways. First, $k$-NN has been combined with other methods such as LDA (Peng et al. 2001); support vector machines (Pan et al. 2004); multi-label learning (Zhang et al. 2007); fuzzy methods (Petridis et al. 2007, Alsberg et al. 1997); classification trees (Buttery et al. 2002); Lineal Discriminant Analysis (Zhang et al. 2010) and Differential Evolution to optimization problems (Buttery 2002). Second, $k$-NN has been combined with variations of itself. For example, Wilson 1972, used $k$-NN to reduce the number of objects in the dataset and then used 1NN to classify unknown objects. Finally, bagging has been used for producing multiple versions of $k$-NN (Breiman 1996). In this case, the classifiers are built on bootstrap duplicates of the training set. A bootstrap duplicate (also called bootstrap sample) is a new dataset generated by sampling with replacement from the original training set (Efron 1992). Then, for each bootstrap sample, a given unknown object is classified using $k$-NN. This process is repeated B times and finally, the unidentified objects are classified in the class in which it was more frequently classified (Breiman 1996). The new bootstrap training set is then used, to classify the unknown objects using $k$-NN. Although this method has low classification error rates, it does not provide the value of reliability of classification.

4.4.4 Reducing the number of objects

If the number of objects in the training set is fewer than it results in reduced storage and computational requirements needed by $k$-NN and the improvement in the results of classification. Cover et al. 1967, proposed the condensed nearest neighbor’s rule (CNN). In CNN, a consistent subset is obtained from the collected dataset. A consistent subset is a training set, which classifies correctly the objects in the test set. Variations of this method have been proposed (Gates 1972, Kuncheva 1995). Other strategies to reduce the number of objects and some applications of them have been described by Desarathy et al 2000, Sanchez et al. 2003 and Raicharoen et al. 2005. Kuncheva 1995, used genetic algorithms for selecting the objects in the dataset.
4.5 Selection or generation of a prototype

$k$-NN has a high computational cost requirement and it is a major and severe drawback in spite of various advantages. To achieve two major advantages of the low computational cost and improved storage need to store the subset (a small set from training set) the selecting prototypes is applied for similar or sometimes even an improves classification performance. Different ways of taking an optimized and proper set of representatives have been studied so far. There are two methods, which lead to the reduction of the training set size are editing and condensing, they are giving optimized set and mentioned as Prototype Selection (PS) and Prototype Generation (PG) respectively (Wilson et al. 2000).

Based on the technique followed by $PS$ or $PG$ method, it chooses a subset of original training data set to remove noisy and redundant instances. The $PS$ methods have one main advantage over the $PG$ method, which enables it to select instances without generating new edited data. Improving $k$-NN with $PS$ is a promising method to obtain expected results to be achievable.

The learning process consists of two steps, editing and condensing, in the case when the classifier uses the $NN$ rule. The main focus of editing is to remove noisy instances and the condensing maintains only the representative instances means it generates prototypes.

4.5.1 Wilson’s editing

Wilson’s editing algorithm (Wilson 1972) is a basic editing algorithm to remove noisy instances, eliminating near border cases, excluding overlap between the regions of different classes. Editing is the step in the learning process to increase the accuracy of classification when the amount of noise is extremely high in training data. Briefly, the approximation of the class of each case in the training set is done by $k$-NN then the elimination of those examples whose true class labels do not agree with the ones adjudged by the $k$-NN rule. Algorithm 4.1 is pseudocode for Wilson’s editing algorithm.
ALGORITHM 4.1: Wilson’s editing algorithm (Wilson 1972)

\[
\begin{align*}
X &= TS \\
S &= X \\
&\text{for } x_i = \text{each}_\text{instance}_\text{of} (X) \text{ do} \\
&\quad \text{kNearestNeighbours}( x_i, X - \{x_i\}) \\
&\quad \text{if } \text{kNN}(x_i) \neq \Theta_\text{i} \text{ then} \\
&\quad \quad S = S - \{x_i\} \\
&\quad \text{end if} \\
&\text{end for} \\
\end{align*}
\]

Let \( \{X, \Theta\} = \{(x_1, \theta_1), (x_2, \theta_2), \ldots, (x_n, \theta_n)\} \) be training set with \( n \) instances and \( J \) possible classes, and \( k \) be the number of nearest neighbours to determine the class for each instance \( x \).

In Wilson's editing algorithm the estimation method is based on a leave-one-out procedure. Here every instance from the training set is used to determine the \( k \) nearest neighbours.

4.5.2 Condensing techniques

In favour of reducing both storage and time required to process the selected data set, the condensing step is applied to select a subset of examples without a noteworthy degradation in classification accuracy. Two main groups of techniques exist for condensing, the selective and the adaptive scheme. In the selective scheme, merely a subset of an original set is constructed (Aha et al. 1991, Hart 1968, Toussaint 1985, Tomek 1976). The adaptive techniques change or generate a subset (Kohonen et al. 2001, Chang et al. 1974).

Hart’s condensing algorithm’s pseudocode is shown in algorithm 4.2.

Definition: A set of instances \( X \) is said to be consistent with respect to another set \( S \), if \( X \) correctly classifies every instance in \( S \), by using the \( 1-\text{NN} \) rule.

As it is said by definition, Hart’s condensing algorithm should reduce the original set to condense set, which is consistent. The Hart’s algorithm is simple and fast and by eliminating instances which are not needed for the correct classification. In most cases, Hart’s algorithm generates the condensed set, which is significantly small in comparison to the original training set. The possible drawback of Hart’s algorithm is to judge whether a resulting condensed set is the smallest consistent set.
4.6 State of the art fast \(k\)-Nearest Neighbour classification based on prototype generation

As discussed in Section 4.5, the training set reduction can be done in two distinct ways i.e. prototype generation and prototype selection. In this section, some distinct state of the art the prototype generation research papers are discussed. As discussed earlier, the prototype generation methods generate edited set from the training set, by considering this fact, the following outlined research papers have a central theme of generating a new training set from the original set.

Chang et al. 1974, introduced the concept of the prototype set derived from the training set for the nearest neighbour classifier. In Chang’s algorithm (in literature is referred as \(PNN\)), the nearest instances from the same class are merged together as a single prototype. The idea behind this algorithm is as follows: At the beginning, every instance in the training set (TS) is considered as a prototype. Then two prototypes \(p^1\) and \(p^2\) will be merged, as averaged single prototype vector \(p\) if they both have the same class label. The merging of two prototypes will continue until the incorrect classifications of patterns in TS start to increase. The experiment was carried out on a dataset of liver disease with 514 training samples, after applying Chang’s algorithm 34 prototypes were generated. It was also observed that with 514 training set the accuracy of classification was 92.5% and with 34 prototypes the accuracy was 91.7%, means the accuracy was decreased by mere 0.8% while the number of prototypes is only 6% of original training samples.
Xie et al. 1993, introduced a novel approach of nonparametric data reduction using the vector quantization techniques, namely, $VQ$-kernel and $VQ$-kNN. In these algorithms, the first step is the construction of an optimal quantizer vector for each class is built from the training dataset. Then, the quantizer vectors used as a reduced set to represent the original set. With reduced quantizers, the $VQ$-kernel and $VQ$-kNN classifiers are built. For vector quantization, a technique called Method I was implemented. The experiment was performed with real speech data of unknown probability distribution. For comparison, different algorithms like Condensed Nearest Neighbour (CNN), Reduced Nearest Neighbour (RNN) and Edited Nearest Neighbour (ENN) were implemented. It was observed that both $VQ$-kernel and $VQ$-kNN give much better results in terms of training set reduction rate, better accuracy and significantly less computational complexity.

Hamamoto et al. 1997, proposed a fast k-NN method called bootstrap technique for nearest neighbour classification. In this research, three methods of bootstrap were implemented namely, Bootstrap1, Bootstrap2 and Bootstrap3. These techniques of bootstrap generate bootstrap samples by locally combining the original training samples, then generated samples used by classifiers $1$-NN and $k$-NN. The experiment involved two artificial datasets and one real dataset to compare the error rate of proposed bootstrap techniques, and it was observed that suggested implementation is effective to remove outliers from the training set.

Mollineda et al. 2002, taken prototype replacement algorithms called Modified Chang Algorithm (MCA) 1998 and proposed an improved version Generalized Modified Chang Algorithm (GMCA). The resulting GMCA approach is a prototype replacement algorithm which uses the hierarchical agglomerative framework to obtain a reduced training set. This agglomerative framework is replacing the original training set by prototype set based on cluster generation and consistency. In this implementation, four datasets from UCI machine learning repository (Iris dataset, a synthetic two dimensional dataset, DNA dataset and Landsat satellite image data) were used to evaluate the effectiveness of the proposed scheme, and classifiers $1$-NN, edited $1$-NN and $4$-NN were implemented. The empirical results suggest that GCMA is able to obtain results better than the existing approach.

Lam et al. 2002, proposed a new framework called Integrated Concept Prototype Learner (ICPL), which generates prototypes by instance-filtering and instance-abstraction to deal with
bottlenecks (high storage requirement, computational cost and sensitivity to noise) of nearest neighbour classifiers. At first, ICPL performs abstraction on the training set, to retain non-border instances. ICPL separately applies to filter on the training set. Next, both abstraction and filtering prototypes are integrated to form final concept prototype set. Four different versions of the proposed algorithm, namely, ICPL1, ICPL2, ICPL3 and ICPL4 implemented and tested on 35 benchmark datasets from the UCI machine learning repository. It was observed that ICPL offers a promising model of integration, which is good at data retention rate and moderate classification accuracy.

Lozano et al. 2006, discussed four prototype optimization methods, namely, MaxNcN, Reconsistent, LVQ and MixtGauss. The MaxNcN technic is based on the concept of NcN (Chaudhari 1996), which try to improve the nearest neighbour approach by using specific information (relation to the training objects which are nearby the decision boundaries). The training instances belong to the same class are located in a neighboring area, hence they can be replaced by a single representative. The reconsistent technique is a modification of MaxNcN algorithm, while MaxNcN can remove some prototypes close to decision boundary because the order in which the instances are taken during prototype generation step. The reconsistent technique tries to address this issue. Learning Vector Quantization (LVQ) algorithm approximate the distribution of classes by using a reduced set of prototypes while minimizing the classification error. MixtGauss assumes the statistical independence of the features and prototypes are selected as the mean vectors, whose mixtures are fit to model each of the classes. The experiment was performed on eleven real datasets taken from the UCI machine learning repository. The 1-NN classifier was used for classification and it was observed that there was no noteworthy difference between LQV and MixtGauss in terms of reducing the rate of prototypes. However, both LQV and MixtGauss give bigger reduction rate of the training set and higher accuracy than MaxNcN and Reconsistent.

Fayed et al. 2007, proposed prototype generating technique called self-generating prototypes. This technique forms a number of groups from the training set. Each group contains some patterns of the same class, then each group’s mean is taken as a prototype for the group. The self-generating prototypes technique with two variations, namely, SPG1 (no merging and pruning steps) and SPG2 (with merging and pruning steps) were implemented.
along with *Gaussian Mixture Model (GMM)* and *self-generating tree (SGNT)*. Two synthetic datasets and four real datasets were used for classification using 1-NN and k-NN. Apart from accuracy as performance measurement, various other criteria (CPU time elapsed in training, CPU time elapsed in testing, number of obtained prototypes) were compared. It was observed that *SPG1* and *SPG2* require less computational effort with better classification accuracy than other techniques.

Nanni et al. 2009, adapted *Particle Swarm Optimization (PSO)* to generate a novel set of prototypes. The social behavior of movement of birds flocking motivated the *PSO* algorithm for generating prototypes from the training set. The concept is, each bird (particle) adjust its flight according to its own flying experience and nearby birds flying experience. In this solution, initially small random set $K$ is taken as a training set, then by applying *PSO* the optimized solution (prototype of the original training set) is generated so that classification error rate can be reduced. The experiment was conducted on six datasets from *UCI machine learning repository* and five classifiers, namely, *PSO, Learning Prototype and Distance Method (LDP), NN, Center based NN (CNN)* and *Genetic Algorithm (GA)* were compared. It was observed that the proposed method produces the lowest error rate.

Triguero et al. 2010, proposed a novel prototype generation approach, called *Iterative Prototype Adjustment based on Differential Evolution (IPADE)*, which follows an iterative scheme to decide the number of prototypes per class. *IPADE* is executing in three different stages: initialization, optimization and prototypes addition. In the initialization phase, it iteratively generates initial *Generated Set (GS)* such that, GS covers each class prototype. The optimization phase, mutation and crossover are applied on GS and trial solution $GS'$ is generated, then 1-NN rule is applied to get fitness value (accuracy) for GS and $GS'$. Depends on fitness value either GS or $GS'$ will be retained. In the last phase of IPADE decides which classes require more prototypes to properly representing their respective class distribution. In this experiment, fifty datasets were used from the *KEEL* dataset repository and eight other algorithms were tested with *IPADE*. It was observed that *IPADE* is a suitable method of prototype generation for NN classification.
4.7 State of the art fast k-Nearest Neighbour classification based on prototype selection

As discussed in Section 4.5, prototype selection removes superfluous instances from the training set. Following research papers discusses the state of the art training set reduction techniques.

Hart 1968, initiated the idea of training set reduction with the new rule called Condensed Nearest Neighbour rule (CNN). The algorithm finds training set $T$ and then finds subset $S$ from $T$, for that it takes each output class instance from $T$ and put them in $S$. Now, each instance in $T$ is classified using only the instances in $S$, in case it is misclassified, then it is added to $S$ to make sure it will be classified correctly. The process is repeated until there is no instance left in $T$. The drawback of this algorithm is that it retains noisy instances which lead to storage of such unwanted noise samples and the degradation in accuracy.

Gates 1972, introduced the Reduced Nearest Neighbour rule (RNN). The algorithm starts with subset $S$ is same as training set $T$. Then from $S$ it removes instances such that the removal does not lead to misclassification of any instances from $T$ by remaining instances in $S$. The drawback of this algorithm is that it computationally more expensive than Hart’s CNN rule, but it produces a subset of CNN and hence it is computationally less expensive and needs less storage during classification stage. The experiment was performed on Iris dataset and CNN and RNN were implemented. It was observed that CNN presented an 83% improvement in memory and time efficiency while RNN offered 87% improvement with no degradation in performance.

Tomek 1976, extended the Wilson’s edited nearest neighbour rule of Edited Nearest Neighbour (ENN) and proposed a new method called All $k$-NN. The algorithm takes first takes $S=T$ (training set and subset both are same), then it flags those bad instances which are not correctly classified by its $i$ nearest neighbour (for $i=1$ to $k$) from $S$. This algorithm leaves internal point intact but removes points at decision boundaries means it performs noise reduction effectively. For the experiment, the one-dimensional pseudonormal distributions dataset was designed using a random number generator and Wilson’s editing and All $k$-NN were implemented to check the performance. It was observed that All $k$-NN produces the
reduced training set of the very desirable structure with a non-overlapping probability distribution.

Brighton et al. 2002, implemented state of the art prototype selection algorithms, namely, *Iterative Case Filtering (ICF)* (Brighton 1999) and *Reduction Technique (RT)* (Wilson 1997) with the aim to find the smallest set of instances which still able to classify with the same accuracy than the original set. In *RT*, two sets are selected for any instance $p$ called nearest neighbors and associates. Then, *RT* will look to see that there is no detrimental effect on the cases which have $p$ as a nearest neighbour if the case $p$ is removed, hence it achieves implicit noise removal. The *ICF* algorithm uses the concept of the reachable and coverage sets for instance $p$, which is synonymous to the neighborhood and associate sets used by *RT*, but the key alteration is that the reachable set size is not fixed but restricted by the nearest case of different class. In *ICF*, the instance $p$ is removed if its reachable set size is greater than the coverage set size. The comparison between Wilson’s editing algorithm (Wilson 1972), *ICF* and *RT* was done. It was observed that both *ICF* and *RT* achieves the highest degree of the instance set reduction (80%) while retaining classification accuracy.

Wu et al. 2002, proposed two methods: template condensing and preprocessing and implemented *Improved k-Nearest Neighbour (IKNN)* classifier. In vector space, for any instance, there are large number of prototypes surrounding its vicinity which forms a homogeneous cluster in feature space. The idea behind the template condensing is that “sparcify” the dense homogeneous cluster by iteratively removing patterns which shows high “attractive capacities”, which in turn reduce the template size for training and maintains high accuracy. The preprocessing operation matches an unknown pattern against the prototype in two sequential stages. In the first stage quick match with the potential pattern is done followed by the second stage where the complete match is done based on thresholding value. The prototypes which failed in the first stage will not be considered further, hence it reduces a large number of prototypes such that it do not sacrifices accuracy. The experiment was performed on a dataset of handwritten numeral recognition with 126,000 patterns. It was observed that prosed algorithms reduced the number of training patterns drastically, reduced the classification time by half and accuracy remains as original.
State of the art fast $k$-Nearest Neighbour classification based on prototype selection

Zhang et al. 2002, taken a meta-heuristic search method *Tabu Search (TS)* (Glover et al. 1998) to obtain the reference subset selection. The *TS* is a dynamic neighborhood method, where the neighborhood of instances in training set can change according to the history of the search because *TS* uses the short--term memories in the form of tabu list that keeps track of recently evaluated solutions. The *TS* will remove redundant samples from the training dataset. For the experiment, five datasets were taken and to compare the reduction rate three reduction techniques were implemented, namely, *CNN, Minimal Consistent Set (MCS)* and *TS*, then to check accuracy *1-NN* algorithm was implemented. It was observed that *TS* finds near optimal reference subset for the nearest neighbour classification and the performance of the proposed scheme is better in term of good training set reduction rate and high classification accuracy.

Barandela et al. 2005, proposed a novel reduction technique *Modified Selective Subset (MSS)* to reduce the training set. The *MSS* uses criteria to decide whether the instances are close to decision boundaries and to measure the closeness it finds its distance to the nearest enemy. Using this measure, it is possible to define the best selective subset as the one that contains the best related neighbor for each prototype in the training set, hence there is no need to compute related neighborhoods for each instance in training set. The experiment was performed with ten *UCI machine learning* dataset and apart from *MSS*, seven other reduction techniques were implemented followed by executing *1-NN* classifier to check accuracy. It was observed that two algorithms (*MSS* and *TSA*) produced better reduction rate with retention of high accuracy.

Angiulli et al. 2007, proposed novel order-independent algorithm call *Fast Condensed Nearest Neighbour (FCNN)* for finding consistent subset from the training set in an incremental manner for *Nearest Neighbour (NN)* rule. The *FCNN* differs with *CNN* in term of order-independent as it always produces the same reduced dataset no matter in which the data is processed. The experiment executed on three large datasets (Checkerboard, Forest Cover Type, DARPA), and four variations of *FCNN* (*FCNN1-4*) along with *CNN* and *MCNN* implemented. It was observed that *FCNN1* and *FCNN2* are noticeable faster and all *FCNN* methods way ahead than conventional methods in the training set reduction and guaranteeing the same accuracy.
Fayed et al. 2009, introduced a new condensing algorithm, *Template Reduction for KNN (TRKNN)*. The idea behind this algorithm is the concept is to build a chain of nearest neighbour, then by selecting the cutoff value of distance amongst chains will effectively separate chains and patterns. The goal is to separate prototype from training set such that instances in it are far from the boundaries and have a minute impact on $k$-NN classification, hence such instance are kept out of prototype by means of breaking chains. The proposed algorithm was compared with traditional $k$-NN, $DROP2$, $IKNN$ for five real world dataset. It was observed that TRKNN gives a smaller number of prototypes than $IKNN$ for two datasets and TRKNN reduces template set size without losing the accuracy.

### 4.8 Research gaps

From the above state of the art literature, following research gaps were identified.

- Absence of application of ML classification technic on soil health card dataset.
- Existing prototype generation techniques are poor in optimizing generation of subset from original dataset.
- Present prototype selection methods could not deal with noisy data effectively.
- Due to high dimensional search space and many attributes, the efficiency may be compromised.
- The performance of prototype generating and prototype selection based classifier for multi-class problems are not explored much in the literature.
- The issues related to combining prototype selection and prototype generation techniques are quite untouched.
- The opportunity to improve the performance of *k-Nearest Neighbour* classifier by applying hybrid method to generate prototype from given training set.

In all the algorithms proposed in the literature including latest algorithm have limit the performance and efficiency of prototype generation and prototype selection based k-NN classifier and post an urgent challenge to the data mining community. In the subsequent chapters, we have proposed new classification models to effectively deal with classification problem by k-Nearest Neighbour classifier.