Knowledge Discovery and Data Mining in Agricultural Database Using Machine Learning Techniques

A Thesis submitted to Gujarat Technological University

For the award of

Doctor of Philosophy

In

Computer IT Engineering

By

Bhagirath Parshuram Prajapati

Enrollment No. 119997107001

Under supervision of

Dr. Dhaval R. Kathiriya

GUJARAT TECHNOLOGICAL UNIVERSITY

AHMEDABAD

April 2019

Knowledge Discovery and Data Mining in Agricultural Database Using Machine Learning Techniques

A Thesis submitted to Gujarat Technological University

For the award of

Doctor of Philosophy

In

Computer IT Engineering

By

Bhagirath Parshuram Prajapati

Enrollment No. 119997107001

Under supervision of

Dr. Dhaval R. Kathiriya

GUJARAT TECHNOLOGICAL UNIVERSITY

AHMEDABAD

April 2019

DECLARATION

I declare that the thesis entitled **Knowledge Discovery and Data Mining in Agricultural Database Using Machine Learning Techniques** submitted by me for the degree of Doctor of Philosophy is the record of research work carried out by me during the period from 2011 to 2018 under the supervision of **Dr. Dhaval R. Kathiriya** and this has not formed the basis for the award of any degree, diploma, associate-ship, fellowship, titles in this or any other University or other institution of higher learning.

I further declare that the material obtained from other sources has been duly acknowledged in the thesis. I shall be solely responsible for any plagiarism or other irregularities, if noticed in the thesis.

Signature of Research Scholar: …… Date: ……..

Name of Research Scholar: Bhagirath Parshuram Prajapati

Place: Anand

CERTIFICATE

I certify that the work incorporated in the thesis **Knowledge Discovery and Data Mining in Agricultural Database Using Machine Learning Techniques** submitted by Shri Bhagirath Parshuram Prajapati, was carried out by the candidate under my supervision, guidance and is to my satisfaction. To the best of my knowledge:

1. The candidate has not submitted the same research work to any other institution for any degree/diploma, associate-ship, fellowship or other similar titles
2. The thesis submitted is a record of original research work done by the research scholar during the period of study under my supervision
3. The thesis represents independent research work on the part of the research scholar

Signature of Supervisor: ……. Date: ……

Name of Supervisor: Dr. Dhaval R. Kathiriya

Place: Anand

Course-work Completion Certificate

This is to certify that Mr. Bhagirath Parshuram Prajapati enrolment no. 119997107001 is a PhD scholar enrolled for PhD program in the branch Computer IT Engineering of Gujarat Technological University, Ahmedabad.

(Please tick the relevant option(s))

☐ He/She has been exempted from the course-work (successfully completed during M.Phil Course)

☐ He/She has been exempted from Research Methodology Course only (successfully completed during M.Phil Course)

☐ He/She has successfully completed the PhD course work for the partial requirement for the award of PhD Degree. His/ Her performance in the course work is as follows-

Grade Obtained in Research Methodology (PH001)	Grade Obtained in Self Study Course (Core Subject) (PH002)
BB	AA

Supervisor's Sign

Dr. Dhaval R. Kathiriya

ORIGINALITY REPORT CERTIFICATE

It is certified that PhD Thesis titled **Knowledge Discovery and Data Mining in Agricultural Database Using Machine Learning Techniques** by Bhagirath Parshuram Prajapati has been examined by us. I undertake the following:

a. Thesis has significant new work / knowledge as compared already published or are under consideration to be published elsewhere. No sentence, equation, diagram, table, paragraph or section has been copied verbatim from previous work unless it is placed under quotation marks and duly referenced

b. The work presented is original and own work of the author (i.e. there is no plagiarism). No ideas, processes, results or words of others have been presented as author's own work

c. There is no fabrication of data or results which have been compiled / analyzed

d. There is no falsification by manipulating research materials, equipment or processes, or changing or omitting data or results such that the research is not accurately represented in the research record

e. The thesis has been checked using Turnitin and found within limits as per GTU Plagiarism Policy and instructions issued from time to time (i.e. permitted similarity index <=25%)

Signature of Research Scholar: …… Date: …….

Name of Research Scholar: Bhagirath Parshuram Prajapati
Place: Anand

Signature of Supervisor: …… Date: ……...

Name of Supervisor: Dr. Dhaval R. Kathiriya
Place: Anand

ORIGINALITY REPORT

7%	6%	5%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	"Progress in Advanced Computing and Intelligent Engineering", Springer Nature, 2019 Publication	3%
2	tampub.uta.fi Internet Source	1%
3	www.ijcaonline.org Internet Source	1%
4	tesisenxarxa.net Internet Source	1%
5	c2learn.com Internet Source	1%

Exclude quotes	On	Exclude matches	< 1%
Exclude bibliography	On		

PhD THESIS NON-EXCLUSIVE LICENSE TO GUJARAT TECHNOLOGICAL UNIVERSITY

In consideration of being a PhD research scholar at GTU and in the interests of the facilitation of research at GTU and elsewhere, I, Bhagirath Parshuram Prajapati, Enrollment No. 119997107001 hereby grant a non-exclusive, royalty free and perpetual license to GTU on the following terms:

a. GTU is permitted to archive, reproduce and distribute my thesis, in whole or in part, and/or my abstract, in whole or in part (referred to collectively as the "Work") anywhere in the world, for non-commercial purposes, in all forms of media

b. GTU is permitted to authorize, sub-lease, sub-contract or procure any of the acts mentioned in paragraph (a)

c. GTU is authorized to submit the work at any national, international library, under the authority of their "Thesis Non-Exclusive License"

d. The Universal copyright notice (©) shall appear on all copies made under the authority of this license

e. I undertake to submit my thesis, through my university, to any library and archives. Any abstract submitted with the thesis will be considered to form part of the thesis

f. I represent that my thesis is my original work, does not infringe any rights of others, including privacy rights, and that I have the right to make the grant conferred by this non-exclusive license

g. If third party copyrighted material was included in my thesis for which, under the terms of the copyright act, written permission from the copyright owners is required, I have obtained such permission from the copyright owners to do the acts mentioned in paragraph 1 above for the full term of copyright protection

h. I retain copyright ownership and moral rights in my thesis, and may deal with the copyright in my thesis, in any way consistent with rights granted by me to my university in this non-exclusive license

i. I further promise to inform any person to whom I may hereafter assign or license my copyright in my thesis of the rights granted by me to my university in this nonexclusive license

j. I am aware of and agree to accept the conditions and regulations of PhD including all policy matters related to authorship and plagiarism

Signature of Research Scholar: ……..

Name of Research Scholar: Bhagirath Parshuram Prajapati

Date: …….. Place: Anand

Signature of Supervisor: …….

Name of Supervisor: Dr. Dhaval R. Kathiriya

Date: …… Place: Anand

Seal:

THESIS APPROVAL FORM

The viva-voce of the PhD thesis submitted by Shri Bhagirath Parshuram Prajapati, enrollment no. 119997107001 entitled **Knowledge Discovery and Data Mining in Agricultural Database Using Machine Learning Techniques** was conducted on …… (day and date) at Gujarat Technological University.

Please tick any one of the following option

☐ The performance of the candidate was satisfactory. We recommend that he be awarded the PhD degree

☐ Any further modifications in research work recommended by the panel after 3 months, from the date of first viva-voce, upon request of the supervisor or request of independent research scholar after which, viva-voce can be re-conducted by the same panel again

Briefly specify the modifications suggested by the panel

☐ The performance of the candidate was unsatisfactory. We recommend that he/she should not be awarded the PhD degree

The panel must give justifications for rejecting the research work

---

Name and Signature of Supervisor with Seal

---

1) (External Examiner 1) Name and Signature

---

2) (External Examiner 2) Name and Signature

---

3) (External Examiner 3) Name and Signature

ABSTRACT

Since the invention of computer, the information available in every field has been digitized to be accessible by people using computer resources. Hence, data are growing rapidly in gargantuan amount in every domain. One such domain of interest for researchers is agriculture field. To digitalize agriculture data, Government of Gujarat introduced *Soil Health Card* which contains macro and micronutrients records of soil samples taken from farms and examined in soil laboratory to record the details into database. It is a very large database containing information of all the farms in Gujarat. To find interesting patterns from the database, we applied and analyzed concepts of data mining on soil health card database in this thesis

Data mining is the process of getting useful information by analyzing different kind of data. The object classification is an important area within the field of data mining and its application extends to various areas, whether or not in the branch of science. In this research, we have concentrated on *k-Nearest Neighbor* classification algorithm to classify soil sample instances into appropriate fertilizers deficiency category. Although *k-Nearest Neighbor* classification is simple and effective technique, having an extensive training set is an element of importance in order to obtain a high accuracy, on the other hand, it makes the classification of each object slower due to its *lazy-learning* algorithm nature. The *k-Nearest Neighbour* classifier has the large computational and storage requirements. In addition, the effectiveness of classification decreases because of uneven distribution of training data.

To overcome above mentioned limitations, we proposed *fast k-Nearest Neighbor* which generates training set prototype based on either *Elbow* method or *Silhouette* method, *training set reduction k-Nearest Neighbor* which reduces training set based on prototype selection and *hybrid k-Nearest Neighbor* classification methods which combines both prototype generation and prototype selection mechanism to generates prototype from original training set. These all methods aims at, how to decrease the requirement of time and space for classification task done by *k-Nearest Neighbour* classifier. We have applied our new approaches of classification on soil health card agriculture data set and our evaluation illustrates that these approaches can solve the mentioned problems effectively.

ACKNOWLEDGEMENT

In the first place, I would like to thank Dr. Dhaval R. Kathiriya, Dean and Principal, AIT, AAU, for being generous to accept me as a PhD student. The dynamism and zeal with which Dr. Kathiriya works have always inspired me to achieve from better to best. It is because of Dr. Kathiriya that I was given access to use soil health card database which is part of soil health card scheme launched by Government of Gujarat in the year 2003. Throughout my PhD research tenure, I have been continuously motivated by Dr. Kathiriya. He showed me different ways to approach a research problem and the need to be persistent to accomplish research goals.

I also thank Dr. Apurva M. Shah (Asso. Prof., MSU) and Dr. Ramji M. Makwana (CEO, AI eSmart Solutions) for being part of my doctoral progress committee. Their valuable inputs during DPCs and open seminar helped me a lot to get into various unexplored research directions. The technical discussions during DPCs and suggestions by them have made me achieve state of the art research incorporating the latest ideas and features. I would like to thank also many eminent professors for their recommendations during the annual research week at Gujarat Technological University.

I take this opportunity to thank, staff members of College of Agricultural Information Technology, Anand, for their support whenever it needed. I would like to thank Dr. R. S. Parmar (Prof., AIT, AAU) for helping to know technical know-how of agricultural information systems and soil health card scheme.

I would like to extend my thanks to Charutar Vidya Mandal and Computer Engineering Department, A. D. Patel Institute of Technology, where I carried out my research while continuing my service. I would like to thank Er. Bhikhubhai B. Patel(Chairman, CVM) for his blessings and Dr. R. K. Jain (Principal, ADIT) for supporting on various occasions.

I would like to thank Mr. Anil Patelia for his technical assistance, Prof. Priyanka D. Puvar, (Asst. Prof., ADIT) for her scholarly interactions about the technical writing skills and Prof. Deep Trivedi for helping in proofreading.

Finally, I am very much indebted to my family, who supported me in every possible way to see the completion of this work.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

CHAPTER - 1

Introduction

With the advent of new technologies, automation of information processing has been adapted into various fields such as finance, medicine, weather forecasting, e-commerce, etc., which affects day-to-day activities of human life. The usage of automation in such domains has given plenty of advantages such as online banking, drug discovery, weather prediction, etc. One such area of prime importance for human beings is agriculture, which is prime economic activity of any country's development. It is an emerging field for application of automation (Singh et al. 2015)

We, the human beings are dependent on agriculture for our daily needs of nourishment, since the inception of civilization. Moreover, due to the industrial revolution, requirements of human beings are increasing and most of the needs are fulfilled by agriculture products like milk, food, clothes, medicine, etc. A good agriculture yielding leads to self-sustainability, and hence, plays a major role in the country's development.

The soil, which is a most important factor to cultivate any crop, faces significant changes in its content over the time due to the occurrence of various external factors such as mining, deforestation, flooding, etc. Since last few decades, during the green revolution, various chemical fertilizers are being supplemented in the soil, to increase the productivity of the crops, which updates the natural content of the soil. For example, to improve nitrogen content in soil, urea fertilizer is being used extensively by farmers. But other nutrient requirements of the soil were not observed properly.

To monitor the health of the soil in order to get a better crop yield, a periodic evaluation of various soil constituents (such as PH value, salt content, phosphorous content, carbon content etc.) is recorded in *Soil Health Condition (SCH)* report. The continuous observations of the soil health reveal the required changes in the soil, for example, the soil has a deficiency of micro nutrients. Based on those changes, expert can provide valuable suggestions (such as different

types of fertilizers to fulfill the requirement of nutrients) for better crop yield. These *SHC* observations are very useful in the areas where farms have varying soil characteristics. Though such an approach of monitoring soil helps in proper yielding of the crops. Further, useful knowledge discovery for soil from huge record presents many-faceted challenges to the experts.

Data Mining (DM) techniques, which are applied to find out suitable information from the huge dataset, can be utilized to discover useful knowledge from the database such as *SHC*. In concern with soil health database, various approaches for data mining were implemented to discover soil health condition. In Western Australia, based on information of different soil health conditions, soil profiles were created and classification of the soil profile was done at multiple levels by applying different classification and statistical methods (Armstrong et al. 2007). In Egypt, for the apprehension of food security, a *DM* approach was applied on agriculture dataset to outline the food concern for the near future. In this approach, neural network modelling for *DM* framework was suggested. The use of such advanced learning approach presents a good retrieval efficiency in mining agriculture dataset (Khedr et al. 2015). The similar objective was observed at Aurangabad region in India. A framework for classification based on different classification model is suggested in this approach. The *J48* classifier model is found to be effective in the classification task (Bhuyar et al. 2015).

Many aspects related to agriculture can produce a large collection of records (such as a soil health, weather parameter affecting crop yield, crop pest control, etc.). In the present era, research is being conducted on to discover useful knowledge from large agriculture dataset by applying data mining techniques. The focus of this thesis is to find approaches to representation, clustering, classification and pruning in order to improvise performance of knowledge discovery from large agriculture dataset.

1.1 Knowledge discovery in agriculture database and machine learning

From many resources related to human activity, there is a growing amount of data available, that can be used for the betterment of the world. One such huge unexplored dataset is the Human Genome Project, which codifies the human genome. Another example is web pages on the Internet, where useful relationships can be found between the web pages to improve search results. Similarly, in agriculture, various sensors are used to record continuous data of many parameters such as humidity, temperature, images, sounds, etc. Many challenges are confronted

in terms of analyzing and finding useful information (knowledge discovery) from the ocean of data and *DM* is designed to address such problems as mentioned above. *Machine Learning (ML)* is a sub-domain of *DM* that is focused on developing algorithms that allow computers to learn to resolve problems based on past records (Tan et al. 2007). *ML* differs in terms of computational aspect than traditional approaches. In conventional computing approach, algorithms are made of instructions, which are programmed to solve the specific computational problem, while the *ML* techniques allow the computers to train on input instances and use statistical analysis techniques for output, which lie in a specific range. In *ML* knowledge discovery can be done in two ways, supervised learning and unsupervised learning. Each record in the database is called instance and it is made of the same set of fields (features, attributes, inputs or variables). In supervised techniques like classification, the *ML* algorithm is trained on known records then it classifies the unknown instances in the specific category (Witten et al. 2016). The other *ML* approach is clustering which works without knowing the class label of instances and that is called unsupervised learning (Rokach et al. 2005) . The focus of this research is on classification and clustering for agricultural soil health card database.

1.2 Soil health card program

The consumption of the fertilizers is highly imbalanced in India, with intercrop, inter-district and inter-state variations. The ratio of *Nitrogen*, *Phosphorous* and *Potash (NPK)* is a measure to indicate the use of fertilizer in a balanced way and in the current scenario in the traits of *NPK* is in extensively inter-state despair. In recent years the productivity of farming is declining because the consumption of fertilizers had been reckless during the green revolution. Various studies have shown that one of the reasons for declining productivity is due to the lack of soil testing facilities. In absence of the soil testing, the farmers are relying on fertilizer dealers or any other unreliable resources for further counselling, to use particular fertilizer without knowing the consequences, wherein long time such ill advice may lead to the unproductive soil (Arthpedia 2017).

In India, the beginning of soil testing program is marked back to the year 1955-56, when the setup of 16 *Soil Testing Laboratories (STLs)* were working for sampling and testing of soils under the scheme of "Determination of Soil Fertility and Fertilizer Use". Facility for soil testing is provided by the State Governments at zero or nominal fees to the farmers. The aim of this soil

testing is to use chemical fertilizers in a balanced way to achieve *Integrated Nutrient Management (INM)*. Hence, sensible and watchful use of chemical fertilizers, bio-fertilizers and locally accessible organic manures can be promoted to obtain better crop productivity and soil health. Two national agencies, *National Mission for Sustainable Agriculture (NMSA)* and Rashtriya Krishi Vikas Yojana, have implemented Soil Testing Programs in India (Agricoo 2017).

The soil testing labs to be endorsed are essential to smooth and quick access of soil health cards. Due to soil health card scheme which endorses cropping, and fertilizer usage based on soil parameters present in a given soil sample of a particular farm, where the farmers have sown previously unknown crops. The Government of Gujarat had launched a soil health card scheme called, *Soil Health Cad Program (SHCP)* in the year 2003 to improve the agricultural output and quality of the crop, is a popular scheme amongst farmers. Till now the Soil health cards are issued to 12 lakh 70 thousand farmers of the state of Gujarat (Agri 2017).

The *SHCP* program is designed to fill in the gap between scientific knowledge and traditional farming knowledge. The backbone of the *SHCP* is *Gujarat State Wide Area Network (GSWAN)*, which runs on internet and intranet. *SHCP* is a warehouse of agricultural information for the benefits of agricultural scientist, policy makers and farmers. This technologically advanced program generates the fertilizer recommendations on the basis of soil analysis and the micro and macro nutrients for the crop for each farm in the state of Gujarat. The generated recommendations suggest how to use different fertilizers in proper proportion, so that the yielding more crop and the effective utilization of the fertilizer can be achieved.

As discussed in Section 1.1, knowledge discovery from dataset such as agriculture dataset (soil health card) by applying *ML* algorithm can help to automate the process of nutrients deficiencies classification of a particular soil sample.

1.3 Motivation of the problem

The *k-Nearest Neighbour (k-NN)* algorithm is a nonparametric instance-based learning algorithm. With *k-NN* a sample is classified according to the same class that outweighs in its *k* closest instances. Hence *k-NN* has a very simple implementation. In addition to that, the key advantage to handle few parameters namely the distance and the value of *k.*

However, *k-NN* is also having some disadvantages. Like, the selection of the attributes must be careful, because in case of too noisy or irrelevant attributes the estimate of the classifier may deviate. Moreover, as *k-NN* is a lazy-learning algorithm, for classifying each instance it needs to calculate the distance to all other known instances *N* (training samples). This unique problem of *k-NN* is a major focus of this work.

Now, this scenario is problematic for large data sets and/or with a large number of attributes. In many domains (e.g. agriculture, multispectral images, text categorization, biometrics or retrieval of multimedia databases) the size of the data set is large that decision systems cannot meet the expectations of the time and storage requirements to process them. In influence of such conditions, classifying becomes a very problematic task for an algorithm such as *k-NN*. In addition, as the nearest neighbour rule stores every instance in the *Training Set (TS)*, noisy and redundant instances are stored as well, which can considerably degrade the classification accuracy.

In *ML* literature, two proposals have been discussed on removing or editing the training instances (Kittler et al. 1982). We can differentiate between these two methods as prototype selection and prototype generation approaches. Firstly, in prototype selection, the condensing algorithm approach aims at a selection of a small subset of instances without a significant degradation of the resultant classification accuracy (Dasarathy et al. 1994, Aha et al. 1991, Hart et al. 1968, Toussaint et al. 1985, Tomek et al. 1976). Secondly, in prototype generation, it eliminates erroneously labelled instances by editing approaches (Chen et al. 1996, Sánchez et al. 2004, Kohonen et al. 2001, Chang et al. 1974).

This research will introduce various algorithms aiming at the problem of appropriately reducing the size of the training set by both approaches. First, a method of selection of some of the already existing instances and second, generating a subset of new representatives with the aim to obtain a substantially decreased in size of the training set. This research aims at finding a balance between the two objectives: the first objective is to reduce/edit the training dataset and the second objective is to monitor the accuracy of the classification task.

1.4 Definition of the problem

To explore the applicability of machine learning techniques on an agricultural dataset of soil health card and to propose improved efficient machine learning algorithm to classify soil sample into the categories of the deficiencies of macro and micro nutrients.

1.5 Objective and scope of work

Objectives:

- To study and analyze agriculture soil health card database, data mining on *Soil Health Card Database (SHCD)* and applicability of machine learning on it.
- To design the concept to carry out machine learning, specifically classification algorithm on *SHCD.*
- To identify the concept of improving time and space complexity of classification algorithm to classify soil samples in respective nutrient deficiencies category.
- To measure the performance of proposed algorithms based on accuracy, classification time and training set reduction rate.
- To design and develop a software prototype to prove the above concept.

Scope:

- Based on the nature of this research, sub-dataset is abstracted from agricultural soil health card database which consists of micro and macro nutrients for the individual farm of the selected districts of Gujarat. This data set is preprocessed for applying machine learning techniques.
- Classification algorithm *k-Nearest Neighbor* is applied to soil health card data set to classify each soil sample into the categories of deficiencies of the nutrients.
- The primary limitation of *k-Nearest Neighbor* is that it retains all the training data because of that it is prone to high computational cost, henceforth, this research proposes efficient *Fast k-Nearest Neighbor*, *Training Set Reduction k-Nearest Neighbor* and *Hybrid k-Nearest Neighbor* classifiers with their comparative evaluation.

1.6 Original contribution

This research is distinctive in terms of its application in the agricultural domain and encompassing *ML* technique *k-Nearest Neighbor* algorithm with its improvements. Agricultural database soil health card is used and from this database macro and micronutrients are abstracted for the particular farm for classification purpose. This research work is original, as there is no such work carried out on soil health program database in the state of Gujarat, so far. Though an effective *k-Nearest Neighbor* algorithm is proposed for classification, it suffers from large storage and computational requirements. In this research, we present a state of the art *Fast k-Nearest Neighbor, Training Set Reduction k-Nearest Neighbour* and novel Hybrid *k-Nearest Neighbour* classification methods for decreasing the requirement of time and space. The original contribution is also observed in the research papers listed at the end.

1.7 Outline of the thesis

This thesis is organized into seven chapters:

Chapter 1: Introduction;
It covers, Introduction, briefly understanding of knowledge discovery in agricultural data set i.e. soil health card, motivation and definition of the problem, objectives and scope and original contribution.

Chapter 2: Fundamentals;
It focuses on different fundamentals concept like data mining, knowledge discovery in database and its models including proposed model for our research and brief about machine learning and its techniques.

Chapter 3: Applications;
It covers, a summary of different applications of machine learning techniques in agriculture domains: namely, remote sensing *ML* applications, crop production estimation, *ML* applications in animal husbandry, *ML* applications in soil science and some recent trends of *ML* applications in agriculture.

Chapter 4: Literature review;

It elaborates in-depth coverage of the state of the art of the research background of this PhD work. It also covers, the concept of k-NN rule, its advantages and disadvantages. The variations of k-NN is discussed. In later part of the chapter, state of the art k-Nearest Neighbour classifier based on prototype selection and generation is discussed.

Chapter 5: Issues and challenges;

This chapter covers, the understanding of macro-micro soil nutrients deficiency in soil health card database and its labeling. It elaborates challenges related to soil health card database. It also covers, details of selection of proposed classification methods followed by its comparisons to available techniques and its limitation.

Chapter 6: Proposed methodologies;

This chapter covers, the implementation of the concept of *PS* and *PG* applied for efficient classification of soil samples into macro and micro nutrients deficiency using *k-NN* classifier on *Soil Health Card Database (SHCD)*. It discusses *PG* based *Fast k-Nearest Neighbour (F-kNN)* classifiers, *PS* based *Training Set Reduction k-Nearest Neighbour (TSR-kNN)* classifier and hybrid *Training Set Reduction F-kNN (TSR-FkNN)* classifiers which employs both *PG* and *PS* models.

Chapter 7: Empirical results and analysis;

In this chapter, comparisons between the above existing and proposed classifier are carried out in terms of accuracy (%), training set reduction rate (%) and classification time (milliseconds).

Chapter 8: Conclusion and future scope;

This is concluding chapter of the thesis. The quantifiable conclusion is discussed for proposed methods followed by future directions.

Chapter 9: Bibliography

This chapter contains bibliography references referred in this thesis.

CHAPTER - 2

Fundamentals

Data Mining (DM) is the process of automatically discovering useful information in the large dataset (Han et al. 2011). It is an interdisciplinary area of research in which experts from multiple fields such as computer science, mathematics, statistics, etc. often collaborate in order to solve problems of different areas by analyzing the huge dataset. *DM* is a result of the natural evolution of database and data management industry into critical functional information technology. The goal of data mining is to proficiently extract information and knowledge from the data that makes sense, i.e. the knowledge should be understandable, valid and useful (Cios et al. 2007).

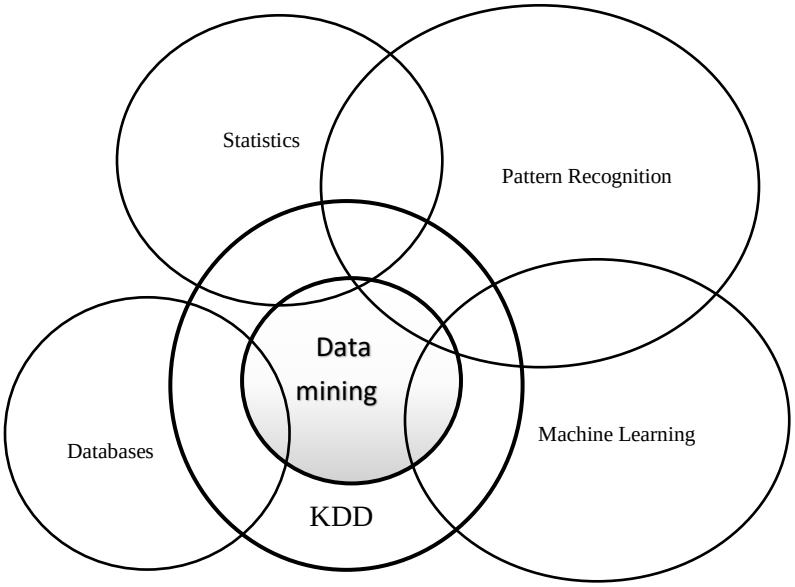


FIGURE 2.1 Venn diagram of *DM*, *ML*, Statistics etc. (Guthire 2017)

As discussed in Chapter 1: Section 1.3, *Knowledge Discovery in Database (KDD)* is identifying valid, potentially useful, and previously unknown patterns in a huge dataset. *KDD* addresses the problem of mapping low-level data into the knowledge forms that might be more compact, more abstract and more useful (Fayyad et al. 1996).

KDD consists of various overlapping areas as shown in Fig. 2.1 (Guthire 2017). *Machine Learning (ML)* is a field of computer science in which computer learns from data using statistical techniques without explicitly being programmed. Pattern recognition is the application of statistical techniques to recognize a pattern in data. The database is a collection of related records stored on electronic media. Databases are the backbone of *DM* system as it stores data and patterns.

2.1 Types of data

Four types of data can be classified in *KDD*; nominal, ordinal, interval and ratio. Nominal and ordinal types fit into the group of categorical or qualitative attributes which lack properties of numbers. Nominal means relating to names. In computer science, nominal attributes are represented by enumerations. Nominal values are symbols of names of things and not containing specific meaningful order (e.g., occupation, zip codes, gender, etc.). For a meaningful order or ranking among the values, an ordinal attribute is referred. In ordinal attributes, the magnitude between successive values is not known (e.g., army ranks, with values Major, Lt. Colonel, Colonel: and so on). Interval and ratio attribute types belong to the group of numeric and quantitative attributes. They are presented by real numbers. Measurement of interval attributes is done on equal-size units. Order and positive, 0 or negative values define interval attributes. It allows to compare and quantify the difference between values (e.g. outdoor temperature). Due to its numeric nature, we can compute the mean, median and mode value of interval attributes (Han et al. 2011). The ratio attributes are with inherent zero points. It is ordered and we can compute mean, median and mode (e.g. weight, height, longitude and latitude). Attributes can also be distinguished between discrete and continuous attributes. Discrete attributes are finite or countably finite in range. A special case of the discrete attribute is a binary attribute. Here values assume two values (e.g. true/false, yes/no…).

2.2 Steps in knowledge discovery in databases

Knowledge extraction is the central theme of entire knowledge discovery process and it revolves around many activities like the techniques for storing data, techniques for analyzing huge datasets, interpretation and visualization of results. Initially academic institutes have developed *KDD* since the 1990s. Subsequently, industry adapted and enhanced *KDD*. A basic schematic *KDD* process is shown in Fig. 2.2. The first basic structure of the model consists of sequential steps where each step is executed in a specific sequence and the next step starts as it requires the result generated by the previous step (Fayyad et al. 1996). Many of the proposed *KDD* models consist of feedback loops for the revision process as shown in Fig. 2.2.

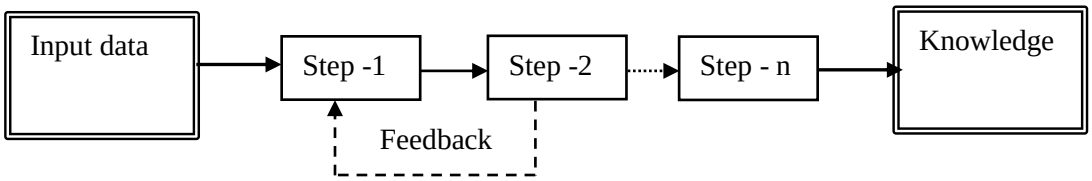


FIGURE 2.2 Sequential structure of *KDD* process (Fayyad et al. 1996)

2.2.1 Academic research model of *KDD*

Fig 2.3 shows academic research model *KDD* consists of nine steps as outlined below:

1. Understanding: It is desirable to learn the relevant prior knowledge and the end user goals from discovered knowledge, it is achieved through this comprehensive step.

2. Selection: To perform discovery task of knowledge, a subset of attributes and data points are selected by various attribute selection techniques such as information gain, chi-square test etc.

3. Preprocessing: This step performs the various preprocessing tasks such as removing outliers, handling noise and missing data values.

4. Transformation: To find important attributes, dimensionality reduction and alteration methods (such as *Principal Component Analysis (PCA)*, *Latent Semantic Indexing (LSI)*) are applied on the dataset.

5. Selecting data mining method: *DM* techniques such as classification, regression, clustering, etc. are selected based on goals defined in step1.

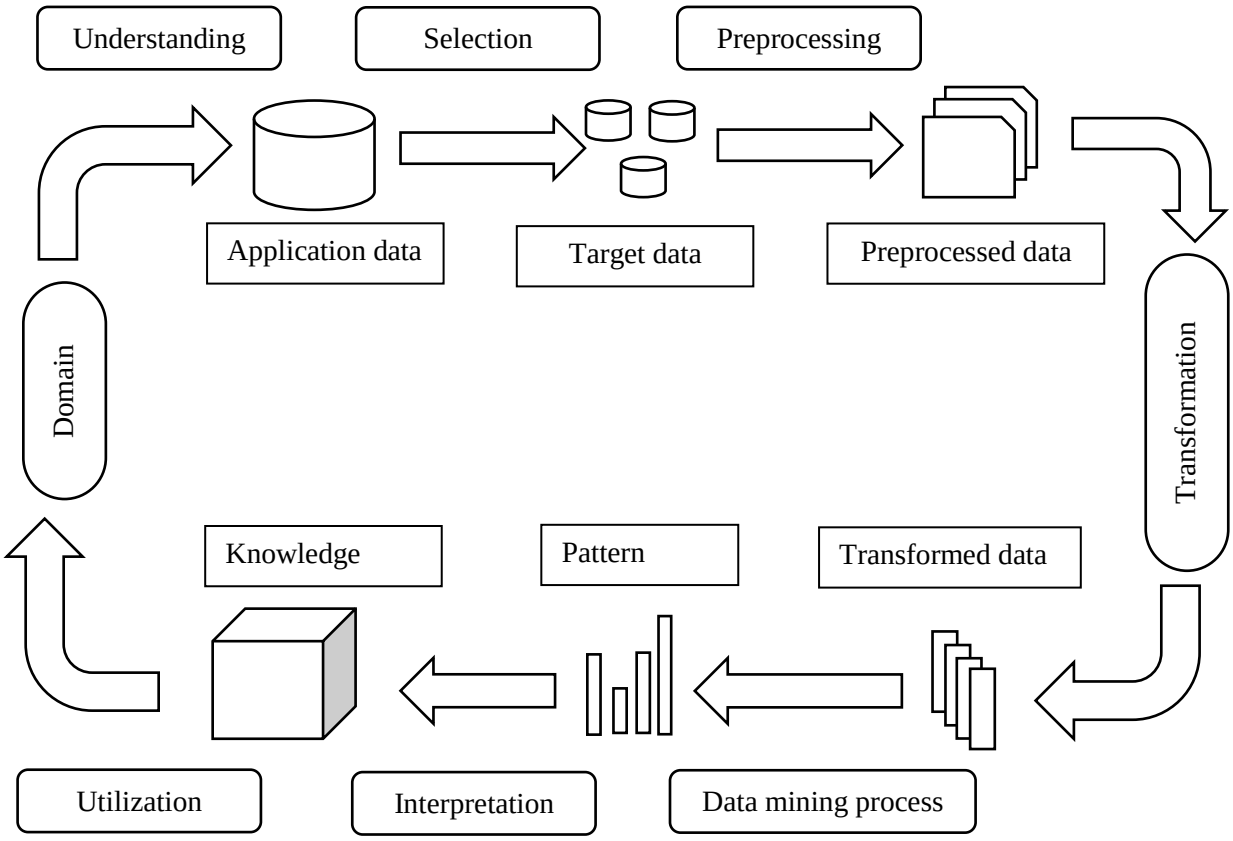


FIGURE 2.3 Steps in *KDD* (Khedr et al. 2015)

6. Choosing data mining algorithm: To search for patterns in the data, the data miner selects the method which is appropriate. Not only method but also the models and the parameters of the methods are selected by data miner.

7. Pattern search: In the form of association rules, closed/max pattern, approximate pattern, etc. are generated in this step (Han et al. 2011).

8. Interpretation: In this step, the visualization of the extracted pattern and models is performed by the analyst.

9. Utilization: This is the final step in which the discovered knowledge is incorporated in the current system of performance and reporting is done to interested entities by checking conflicts with previously existed knowledge are resolved.

2.2.2 Industrial model of *KDD*

With the academic efforts, the industrial model of *KDD* was proposed with several different approaches. This model was named as *Cross-Industry Standard Process for Data Mining (CRISP-DM)*. The *CRISP-DM* model was a collaborative efforts of four companies: DaimlerChrysler, Integral Solutions Ltd., NCR and OHRA. They all are experts in their respective fields: DaimlerChrysler is an automobile manufacture giant; Integral Solutions Ltd. is a provider of commercial data mining solutions; NCR is a database provider; and OHRA is an insurance company. As shown in Fig. 2.4 this model consists of six steps (Chapman et al. 2000).

1. Business understanding: This step converts objectives and business requirements into a DM problem definition and designs a preliminary project plan to achieve the objectives. It further divides into sub-steps namely, business objectives determination, assessment of the situation, generation of a project plan and *DM* goals determination.

2. Data understanding: This step aims at specifically the identification of data quality as a part of initial data collection. Data understanding is further broken down into preliminary data collection, description of data, examination of data and verification of the quality of data.

3. Data preparation: This step covers actions to build the final dataset, which includes a table, record and attribute selection, data cleaning and construction of new attributes and transformation of data. It is divided into a selection of data, cleansing of data, construction of data, integration of data and formatting of data sub steps.

4. Modeling: This step is bifurcated into the selection of modelling technique(s), generation of test design, the creation of models and assessment of generated models. For the same *DM* problem, various methods are used to check the optimal values. Here, methods may require a specific format for input data hence reiteration into the previous step is done.

5. Evaluation: This step performs an assessment of *DM* results in terms of business accomplishment benchmarks against generated model from the pervious step. If model fulfills the benchmarks then it becomes the approved model.

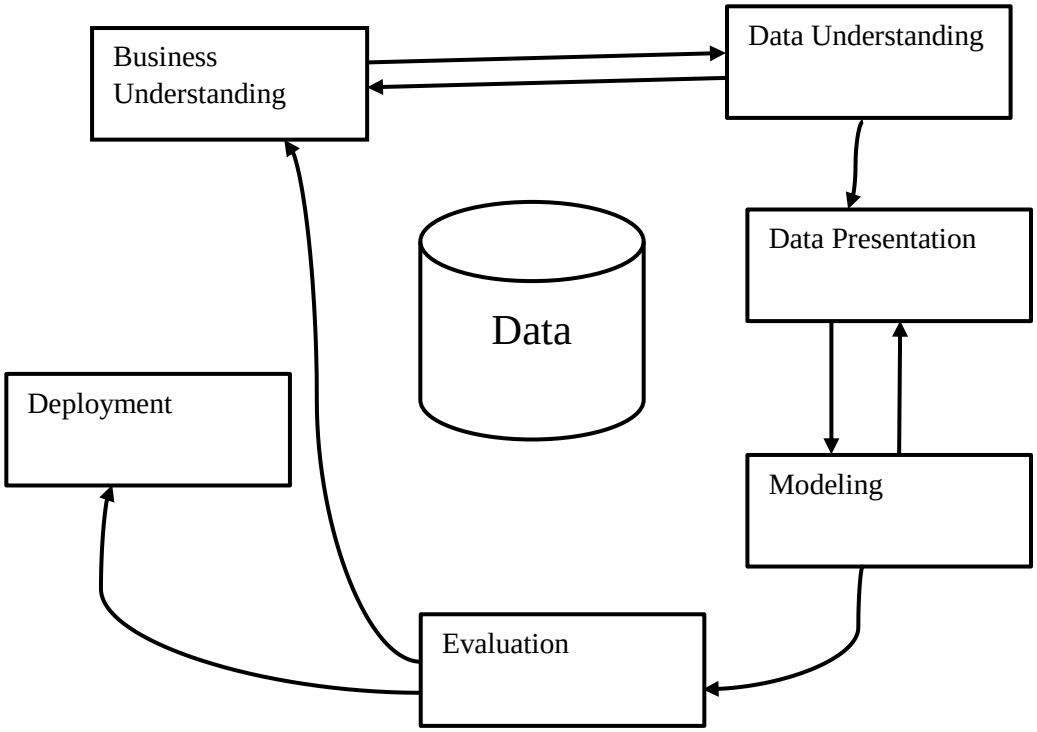


FIGURE 2.4 The *CRISP-DM KDD* model (Chapman et al. 2000)

6. Deployment: This step is sub-divided into plan deployment, plan monitoring and maintenance, generation of final report and review of the process sub-steps. The organization of the discovered knowledge must be presented in the customer friendly way.

2.2.3 Hybrid model

This model combines the merger of academic and industrial models. The following hybrid model sees Fig. 2.5 is a six-step *KDD*. The idea of the academic model is adopted in conjunction with the *CRISP-DM* model to develop this model. This extension includes more general research-oriented description of the steps instead of modelling and it includes data mining step, which is explicit feedback mechanism and in the last step the knowledge discovery for a particular domain, which may be applied in other domain. A description of the six steps is given below:

1. Understanding of the problem domain: It involves the expertise of the domain experts to define the problem, identifying the key people and determine the project goals. Domain-specific terminology is also being learned and finally, project goals are translated into *DM* goals.

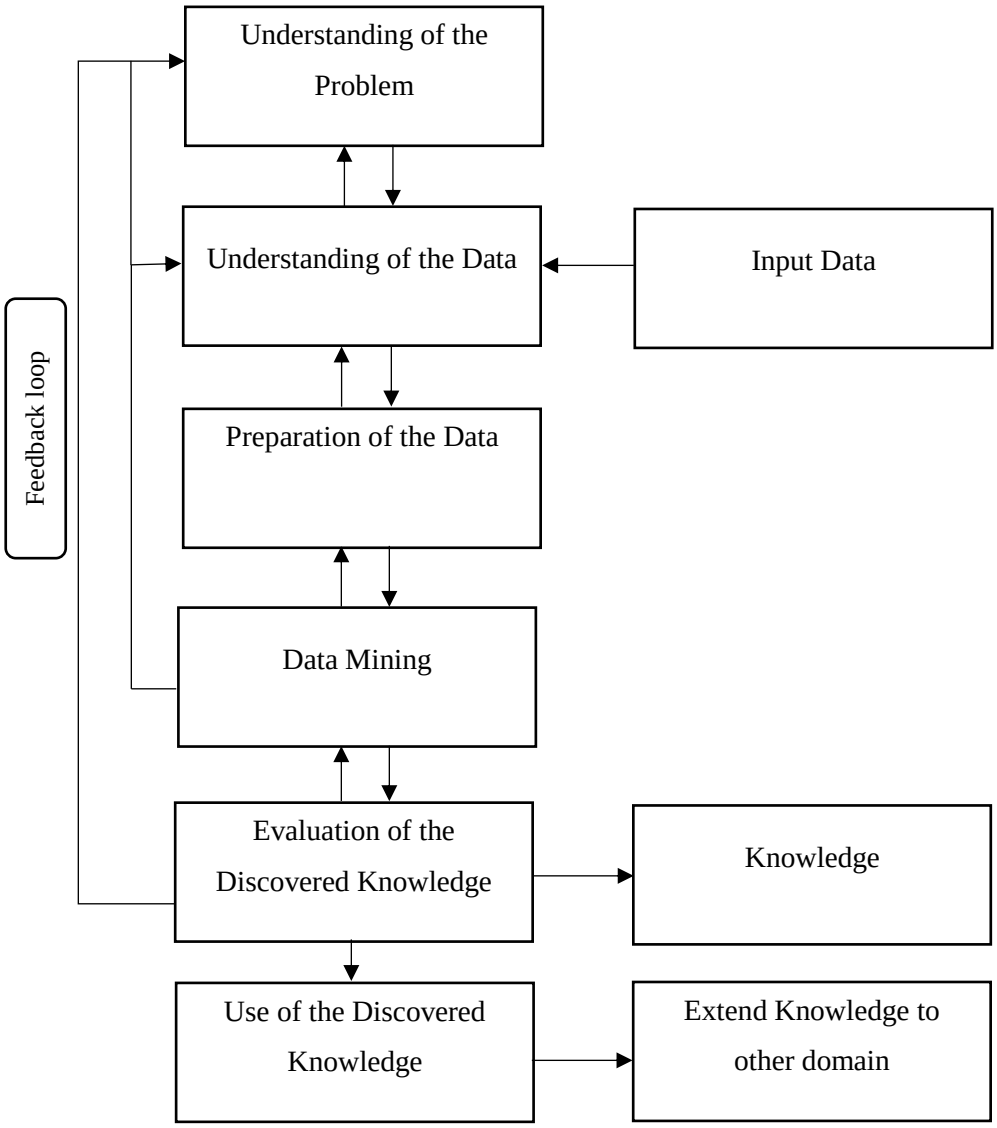


FIGURE 2.5 The six-step *KDD* model (Pan et al. 2000)

2. Understanding of the data: In this step, the collection of sample data is done followed by checking the usefulness of the data in consideration with the DM goals.
3. Preparation of the data: In this step, various preprocessing techniques are applied, such as sampling, running correlation tests, data cleaning, checking the completeness of data records, noise removal and missing values, etc. to meet the specific input requirements.
4. Data mining: To derive the knowledge the various *DM* methods are applied to processed data. For example, association rule mining, classification, clustering etc.

5. Evaluation of discovered knowledge: This step checks whether the discovered knowledge is novel and interesting, also the interpretation of the results by domain experts and how it impacts on available knowledge. At the end of this step, the approved models are retained and the entire process is revisited to identify the alternative actions which could have been performed to improve results.

6. Use of the discovered knowledge: It is the final step of planning to use discovered knowledge. Finally, deployment of discovered knowledge is performed.

Feedback loop: The most common reasons for the feedback loop is a poor understanding of the data, which requires modification of the project's goal. If problem restrictions, requirements and understanding are poor then it may require a repetition of complete *KDD*. Loop in *KDD* helps for a better selection of the data if the results are unsatisfactory and if results are satisfactory then the feedback loops may be used to provide important information for subsequent runs of the *KDD*.

2.2.4 Proposed training set reduction *KDD*

For the current research work, we have considered five-step *KDD*, which includes data extraction, data pre-processing, prototype generation, applying *ML* and evaluation.

1. Data extraction: For this research, we have extracted data from the *Soil Health Card Database (SHCD)* (Chapter 1: Section 1.2), which is stored in MS-SQL database management software. The target dataset of Kutch district is abstracted and stored in MS-EXCEL format for experiment purpose.

2. Data pre-processing: Correct and consistent data is imported for good models and then the cleansing of the data is carried out. Missing values can create big problems and hence need to

be dealt with carefully. For this experiment, missing value records are dropped followed by applying normalization on given attributes to give equal weight to each attribute. In this research, we have pre-processed eights (macro and micro) attributes from *SHCD*.

3. Prototype generation: The prime objective of this research is to classify *SHCD* records into predefined categories of nutrient deficiencies. To fulfill the mentioned objective we have used *ML* classification technique *k-Nearest Neighbour (k-NN)*, which is having its distinguished advantages and disadvantages. To overcome the bottlenecks of *k-NN*, we have adapted its modified version, which first edit and/or reduce training set of *k-NN* by applying prototype selection and/or prototype generation methods.

4. Applying ML: This step applies modified *k-NN* classifiers to classify *SHCD* samples into appropriate macro and micro nutrient deficiencies.

5. Evaluation: Once a classification model has been constructed, it can be used to predict the class of previously unseen objects and accuracy is measured to check the effectiveness of the classifier.

The classifier's effectiveness is measured in accuracy, as we are interested to identify a classifier, which is most accurate in recognizing the category of soil sample. Further, for computer-based expert system, the most accurate classifier can be implemented for the classification of macro and micro nutrient deficiencies.

2.3 Machine learning techniques

Machine learning (ML) is an emerging area of data mining that permits the computer program to become more precise in predicting results without being explicitly programmed. As discussed in Chapter 1: section 1.1, These *ML* algorithms are often categorised as supervised or unsupervised. Supervised learning algorithms use labelled training data for inference (classification, regression), while unsupervised learning algorithms use unlabelled data to find hidden existing patterns (clustering).

2.3.1 Classification

Classification is the method of mapping the input set of instances *P* into a special set of attributes *Q* which is also known as target attributes or labels. This can be explained by the

example of classifying Iris species as presented in Table 2.1. The input set of attributes will be presented as sepal length, sepal width and petal length, petal width. Based on these attribute values the Iris data set can be classified into one of the following categories of the class labels: Versicolor, Setosa, and Virginica (Tan et al. 2007).

TABLE 2.1 Iris data set (Asuncion et al. 2007)

Sepal length(cm)	sepal width(cm)	petal length(cm)	petal width(cm)	Class Label
5.1	3.5	1.4	0.2	setosa
6.4	3.2	4.5	1.5	versicolor
4.4	3.0	1.3	0.2	setosa
6.4	2.7	5.3	1.9	verginica
5.8	2.6	4.0	1.2	versicolor
7.6	3.0	6.6	2.1	verginica
5.0	3.3	1.4	0.2	setosa

There are numerous classification techniques like *decision tree classifiers, bayesian classifiers, artificial neural networks, nearest neighbour classifier, random forest* and *support vector machines* being used by various applications (Anzai 2012). We are going to cover briefly about each one of them. Each technique works based on the learning algorithm it employs.

Performance evaluation of classification model is being measured by the number of correct labels assigned to respective input attributes with respect to a total number of input attributes offered to the classification model, Eq. 2.1.

$$Accuracy = \frac{number\_of\_attributes\_correctly\_classified\_by\_model}{Total\_number\_of\_attributes} \tag{2.1}$$

The performance of a classifier can also be expressed in terms of error rate, which is represented as Eq. 2.2,

$$ErrorRate = \frac{number\_of\_incorrect\_mapping\_done\_by\_classifier}{Total\_number\_of\_attributes} \tag{2.2}$$

All most all classification methods look for developing models, which can attain highest accuracy level and minimum error rate. In upcoming sections, numerous classification methods are explained thoroughly.

2.3.1.1 Decision tree classifier

A decision tree is one of the most widely used, simplest classifiers for solving classification problem. A decision tree is a graph in which sorting of instances based on their feature values is performed to classify them. The decision tree is built of nodes and branches, where each node represents an instance to be classified and each branch represents a value the node can assume. Classification of instances in decision starts at the root node and sorting of instances depends on their feature values.

Example of constructing a decision tree: for predicting, whether an employee would be promoted to the higher position or not, based on the input attributes given in terms of degree of an employee, years of experience of an employee and the research projects an employee is working with. The training set for the same is depicted in table 2.2. The rules being designed for getting a promotion are specified. The person must have the degree of Ph.D. with minimum 5 years of experience and at least one research project, the person should be working on. If we try to construct a decision tree for the above-given training set, we get the Fig. 2.6.

TABLE 2.2 Training data set for decision tree (Tan et al. 2007)

Name	PhD Degree holder	Years of experience	Research projects	Class label Promoted?
Sushmita	No	10	1	No
Ravi	Yes	5	0	No
Ashmita	Yes	4	3	No
Kunal	Yes	6	1	Yes

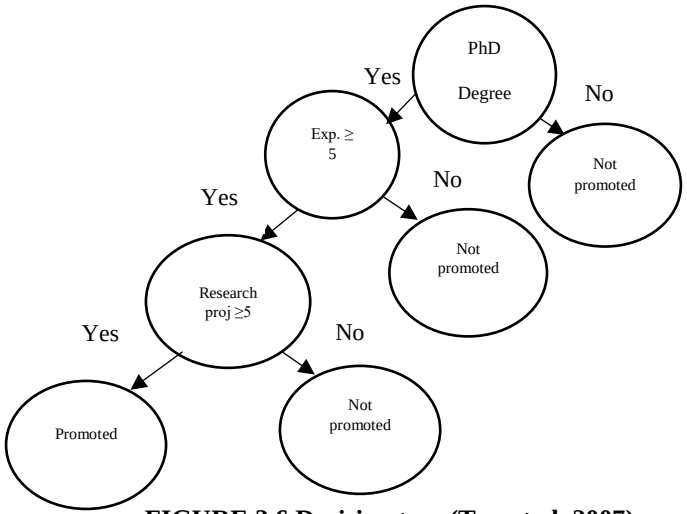


FIGURE 2.6 Decision tree (Tan et al. 2007)

2.3.1.2 Bayesian classifier

In some applications it is difficult to predict class label for the given set of input attributes. Moreover, class variables are non-deterministic even the given input attribute set values to get a match to some of the attributes of training data set. This is possible due to the presence of some noisy data and certain perplexing factors which are not considered for analysis. For example, predicting the chance of heart disease for a particular person based on the routine followed by that person. It is possible in this case that most people who eat healthy food and doing exercise regularly also have a chance of developing heart disease due to other factors such as smoking, alcohol consumption and may be heredity. In such cases, the classification model defined based on commonly known attributes for heart disease, which cannot specify accurate information. In such type of applications, there is a need to model probabilistic relationships between the attribute set and the class label and Bayesian classifier is all about to justify such tasks (Murphy 2012).

The idea behind a *Bayesian* classifier is to build the probabilistic model (*Bayesian* theorem) from the given features and then to predict the classification of a new sample. Here, if the class label is known then the probabilistic model can predict other features but if the class is unknown then a *Bayesian* theorem can be used to predict the class given the feature values. *Bayesian* theorem consists of conditional probability and joint probability, wherein conditional probability refers to the probability that a variable A takes the value a while the value of variable B is observed as b and the same is presented as $P(A=a \mid B=b)$ and the joint probability refers to the probability that a variable A will take the value a and variable B will take value b again the same is presented as $P(A=a, B=b)$. Both these probabilities are related in the following manner, Eq. 2.3.

$$P(A, B) = P(B|A) \times P(A) = P(A|B) \times P(B) \tag{2.3}$$

With this given relationship of probabilities, the Bayesian theorem is obtainable as, Eq. 2.4:

$$P(B|A) = P(A|B)\, P(B) / P(A) \tag{2.4}$$

Naïve Bayes classifier also known as simple *Bayes* or *independence Bayes* is a *Bayesian* classifier. Naïve Bayes works based on the regulation that class label is drawn from the finite set of features by considering that each feature is independent of another one. For illustration,

any fruit can be considered as orange if its colour is orange, the shape is round and it is about 10-11 cm in diameter. *Naïve Bayes* classifier considers the contribution of each one of these three features of fruit independently regardless of any possible correlation amongst them. Conceptually, Naïve Bayes is a conditional probability model in which given problem instance represented by the vector $X = (x_1, x_2, \ldots\ldots, x_n)$ representing some *n* features needs to be classified, it assigns to instance probabilities, Eq. 2.5.

$$P(C_k \mid X) = P(C_k)P(X/C_k)/P(X) \tag{2.5}$$

2.3.1.3 Artificial neural network

Artificial neural network (ANN) is inspired by the concept of biological neural networks which are used to constitute animal brains. *ANN*, is also known as connectionist systems because it is composed of interconnected nodes and directed links. Each connected link is assigned some weight and it is responsible for transmitting a signal from one node to another as shown in Fig. 2.7. A node, which receives signal will process it and then by transmitting to another node. In common *ANN* implementations signal at the connection between artificial neurons is basically a real number and output of every neuron is calculated by a non-linear function of the sum of all its inputs. As learning proceeds, the strength of the signal increases or decreases by weights of artificial neurons and the connections between them (Hassoun 1995).

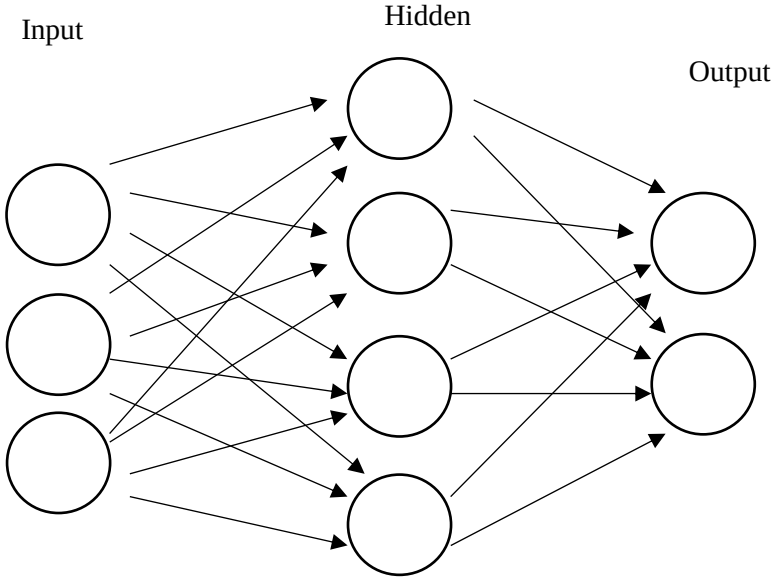


FIGURE 2.7 An architecture of *ANN* (Hassoun 1995)

Such systems are used to learn tasks by making an allowance for examples without considering task-specific knowledge. For instance, the task of identifying the image of a cat by observing the images of a cat which are labeled as “cat” and “no cat” and using the results of the analysis of cat images for further learning without having prior knowledge about features of cat-like tails, stubbles, cat-like faces and fur. In literature, many *ANN* models like perceptron model and multilayer artificial neural network are explored. Multilayer artificial neural network is more complex structure as compared to perceptron in the sense the network may contain a number of intermediate layers between input and output layer called as hidden layers as shown in Fig. 2.7, as well as the network, may apply other activation functions like linear and sigmoid other than sign functions.

2.3.1.4 Nearest neighbour classifier

In *ML* classification, there are two strategies available for making model learned. One of them is as soon as the training set available the model starts learning and such models are called eager learners and in another one model observes all training examples but performs classification only if the attributes of test instance match with any one of training instances exactly. Such learners are known as lazy learners (Statnikov et al. 2008). As shown in Fig. 2.8 *Nearest Neighbour (NN)* classifier considers each example as a data point in a *d*-dimensional space, where *d* is the number of attributes. For the given test example the proximity of it with all data points of the training set is calculated. The *k-Nearest Neighbours* of data point *X* refer to the *k* points that are closest to the *X*.

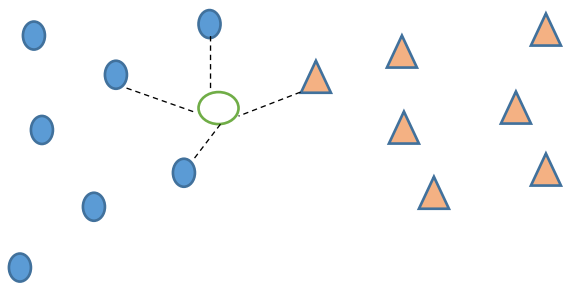


FIGURE 2.8 *k*-NN classifier (Statnikov et al. 2008)

The data point is then classified, based on the class labels of its neighbours. If there exist more than one class labelled neighbours then the data point is assigned a class label of majority

amongst them. The value of *k's* nearest neighbours should be precisely decided. If the value of *k* is too small it can misclassify because of noise availability in the training data. On the other side if the value of *k* is too large then also there is the possibility of misclassification as the set of nearest neighbour may contain data points which are located far away from test attribute's neighbourhood.

2.3.1.5 Random forest

First, Random forest is the supervised machine learning algorithm which constitutes the forest of decisions made by multiple decision trees which are generated with the use of random vectors. This method can be used to solve the problems of classifications as well as regression techniques. The result generated by the random forest is related to a number of trees it combines in the forest in the manner that as the number of trees in forest increases, there is a possibility to get the more accuracy. One thing to clarify is that, creating the forest is not as same as creating decision trees (Statnikov et al. 2008). The main difference between decision trees and the random forest is that finding the root node and splitting the feature nodes will run randomly in case of random forest classification. Random forest classification is popular due to some advantages of it. One of them is; it can be used for both classification and regression. Another is, if enough number of trees are available, then the problem of overfitting does not occur with this method. In addition to this, a random forest classifier can handle missing values also, as well as, it can be modelled in case of categorical values. Random forest classifier can be used in the field of Medicine, Banking, E-commerce, and Stock market. In the case of banking Random classifier is used to find loyal customers and fraud customers. While in case of medicine Random forest is used to identify the correct combination of medicines and to recognize the disease from the past medical records of a patient. As an application in the stock market, Random Forest classier is used to observe a stock's behaviour and then to identify the loss and profit. In case of E-commerce Random forest can be used to predict the recommendation of customers about the products.

2.3.1.6 Support vector machine

Support Vector Machine (SVM) is the supervised learning model used for classification. It has received substantial attention in the area of classification. In *SVM* model instances of the distinct categories are divided by a clear gap in vector space. As soon as new example arrives, it is mapped into the particular vector space and its label is predicted to a category depends on which side of the gap they fall (Schölkopf et al. 2002). An *SVM* can efficiently perform the non-linear classification using the concept of kernel trick.

We are given a training dataset of n points of the form $(x_1,y_1),\ldots\ldots(x_n,y_n)$, where the y_i are either 1 or −1, each indicating the class to which the point x_i belongs. Each x_i is a p-dimensional real vector. We want to find the "maximum-margin hyperplane" that divides the group of points x_i for which $y_i = 1$ from the group of points for which $y_i = -1$, which is defined so that the distance between the hyperplane and the nearest point x_i from either group is maximized, Fig. 2.9. Any hyperplane can be written as the set of points x_i satisfying, Eq. 2.5.

$$\mathbf{w} \cdot \mathbf{x} + \mathbf{b}=\mathbf{0} \tag{2.5}$$

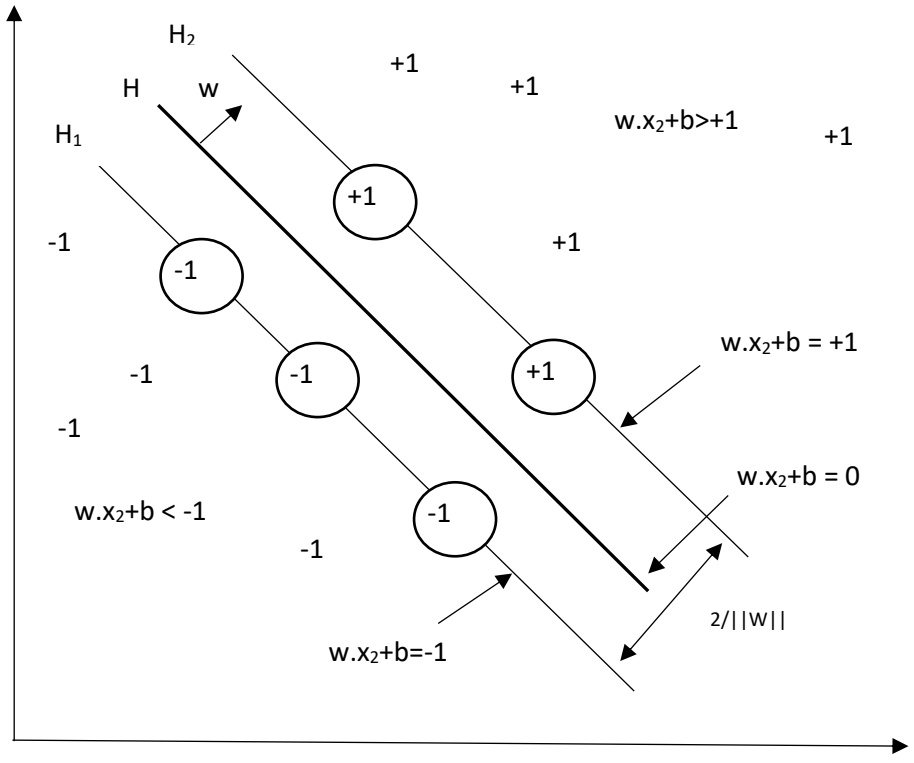


Figure 2.9 Maximum hyperplane and margins for *SVM* (Schölkopf et al. 2002)

We can select two parallel hyperplanes if the training data are linearly separable to achieve maximum distance. These two hyperplane forms a region called the “margin”, and the hyperplane that lies in the middle of the region is called maximum-margin hyperplane.

These hyperplanes can be described by the Eq. 2.6, 2.7.

$$\mathbf{w} \cdot \mathbf{x} + \mathbf{b} = \mathbf{1} \tag{2.6}$$

$$\mathbf{w} \cdot \mathbf{x} + \mathbf{b} = \mathbf{-1} \tag{2.7}$$

The non-linear classifier can be created by applying the kernel trick to maximum-margin hyperplanes. The resulting algorithm will be similar except that the dot operation is replaced by the non-linear kernel function.

2.3.2 Clustering

Clustering or cluster analysis is the chore of grouping a set of objects in such a way that objects in one group are more similar to each other than objects into another group as shown in Fig. 2.10. As the similarities among the objects in one group and dissimilarities among objects in different groups increase, the clustering would be better. Clustering is the core task of data mining and even can be used in other many fields like image processing, data compression, computer graphics, machine learning and many more. Clustering can be correlated with other techniques which are used to divide objects into different groups such as classification, segmentation and partitioning. If we try to compare Cluster analysis with classification then it can be said that clustering is the unsupervised learning. Cluster analysis differs from classification in the manner that in case of classification there remains the knowledge of classes in erstwhile, while in clustering it is not so. In addition to this, in case of classification new samples are classified into known classes while in case of cluster analysis groups are suggested based on patterns in data. At last, in classification labelled input samples are required which not the case in cluster analysis (Mirkin 1998).

Figure 2.10 Illustration of clustering (Mirkin 1998)

Another, similar term ‘segmentation’ refers to the division of data in any form into groups using simple techniques. For example, an image can segment based on pixel intensity and colour or employees can divide based their salary. On the other side, partition, which is also similar to the clustering, refers to the division of graphs into subgraphs. There are various types of clusters as well as types of clustering techniques available, which will be explained in the following section.

There are different types of clustering methods available including hierarchical clustering, partitioning method, fuzzy clustering and complete and partial clustering.

2.3.2.1 Hierarchical clustering

Hierarchical clustering is a set of nested clusters that are organized as a tree. In this type of clustering, clusters are allowed to have sub-cluster. In a tree formed hierarchical clustering every node except leaf node which represents a cluster is the union of its children and the root node of the tree is the cluster containing all the objects as shown in Fig. 2.11. Hierarchical clustering is popular because of some reasons like 1) It does not require any particular value as in case of *k-means* clustering 2) Generated tree provides meaningful taxonomy 3) Only distance matrix is needed to compute the hierarchical clustering. There are two types of algorithms available for hierarchical clustering: one of them is an agglomerative algorithm, which is using a bottom-up approach, and another one is a divisive algorithm, which uses the top-down approach.

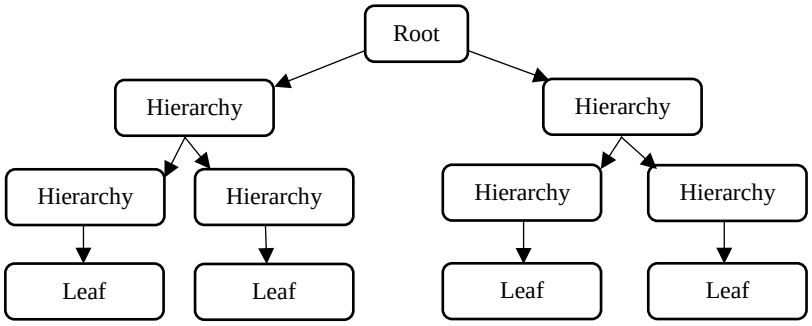


Figure 2.11 *Hierarchical clustering* representation (Mucherino et al. 2009)

2.3.2.2 Partitioning method

Partitioning method of clustering divides the dataset into non-overlapping subsets in such a way that each object in the set will fall into only one cluster as shown in Fig. 2.12. The partitioning is essential to discover the groupings in the data. That means if we want to partition the dataset *D* of objects n into *K* number of partitions then it is called *K*-partitioning method and as a result, we do get a group of *K* clusters. Then it is improved iteratively by optimizing an objective function which is known as *Sum of Squared Errors (SSE)* (Mirkin 1998)].

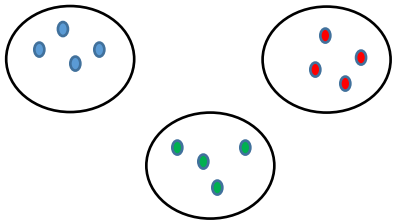


Figure 2.12 Partitioning method of clustering (Mucherino et al. 2009)

2.3.2.3 Fuzzy clustering

As we have seen in the case of partitioning clustering method, each object is assigned to specifically one class and this type of clustering is known as exclusive clustering. In contrast to this, there are many applications where there is a need for considering an object to be part of more than one cluster or an object belonging somewhere among two or more clusters, and it can be assigned to any one of clusters around it. Such type of clustering is known as Fuzzy clustering. In fuzzy clustering, it is possible that every object can be part of every cluster with their different membership weights which can be between 0 and 1. This means clusters in

fuzzy clustering can be treated as fuzzy sets. Fuzzy clustering always considers that the sum of weights of each object should be 1. This is also known as probabilistic clustering by means of which the probability with which each object belongs to each cluster is calculated, and the sum of these probabilities should be 1. Fuzzy clustering or probability clustering can be converted into exclusive clustering by allocating each object to the cluster in which membership weight of it is highest (Mirkin 1998).

2.3.2.4 Complete and partial clustering

Complete clustering and partial clustering are opposite to each other. *Complete clustering* allocates every object to a cluster, while *partial clustering* ignores assigning some objects to the cluster (Mirkin 1998). Complete clustering is used in the applications where there is a need for every object to describe the cluster completely. For example, the task of organizing all documents to be browsed requires completer clustering while in some of the tasks it is important to ignore some objects which may present noise, outliers or uninteresting background and there we can use the concept of partial clustering.

CHAPTER - 3

Applications

All aspects of socio-economic development have been influenced enormously by the convergence of computing, information technology and communication technologies into human activities since last few decades (Mucherino et al. 2009). In modern days, the focus of various agencies (such as Government, agricultural research institutes, policy makers, etc.) is to integrate information technology into agriculture sector to effectively utilize the knowledge of farming in order to enhance the growth of its products. Information and knowledge-based precision farming systems are being developed to bring new contents and concepts into the agriculture and food sector.

Precision Agriculture (PA) is a term coined to describe the use of the latest information technology in agriculture to improve crop production and quality while optimizing usage of fertilizer, irrigation, human efforts, etc. (Whelan et al. 2013). As we discussed in the previous chapter, *Machine Learning (ML)* facilitates better decision-making process in real-word scenarios with nominal human assistance. The *ML* techniques (such as prediction, classification and clustering) are being extensively applied in many domains and would also be useful to *PA*. In this chapter, we will look at how *ML* techniques were applied and useful in agriculture activities.

3.1 Machine learning applications in remote sensing agriculture data

With the availability of enhanced spectral, spatial and temporal resolution from satellite data and advances in computing resources, the methods for accurately classifying remote sensing images have been improved in recent years.

3.1.1 Crop classification from remote sensing data

For proper utilization of limited crop resources, it is very essential to have correct and on time estimate of upcoming crop. In last few years, the use of remote sensing data has been adapted from land cover mapping to assessments of farm related activities as a part of *PA* (Schellberg et al. 2008). *ML* techniques are used in the classification of more frequent crops from remote sensing images taken from space or airborne remote sensing apparatus. Various *ML* methods are applied to classify land cover (Muchoney et al. 2000) taken Moderate Resolution Image Spectro-radiometer (MODIS) and Advance Very High Resolution Radiometer (AVRHH) remote sensing data to study the area comprising of southern Mexico and nearby regions and applied *Decision Tree (DT)*, *Maximum Likelihood Classification (MLC)* and Gaussian *Adaptive Resonance Theory(ART)* to classify land cover mapping. The performance of classification is measured in terms of accuracy and it was observed that the accuracy of *DT*= 88%, *MLC*=53% and *ART*=87%.

Zhang et al. 2009, studied area of Zhejiang Province, China to map paddy rice from the remote sensing image taken from Advanced Land Observing Satellite (ALOS) and Phased Array type L-band Synthetic Aperture Radar (PALSAR). The classification task was performed by *Support Vector Machine (SVM)* to divide the data into five different classes namely paddy rice, dry land, water, orchard and urban. It was observed that the overall accuracy of 80.10% is achieved by *SVM*. Similarly, Yang et al. 2011, have classified corn, cotton, sorghum, sugarcane and non-crop from the study area of Texas (cropping area near Alamo and Elsa). The remote sensing images were taken by Satellite Pour l'Observation de la Terre (SPOT-5) satellite which gives *Color Infrared Image (CIR)* image of the study area. From *CIR* images (Fig. 3.1) it can be seen that the vegetation has a reddish color; grayish and cyanish color is for soil, and the dark bluish color is for water. Five classifiers were applied for the classification task and it was observed that *MLC* and *SVM* classifiers were having best classification accuracy.

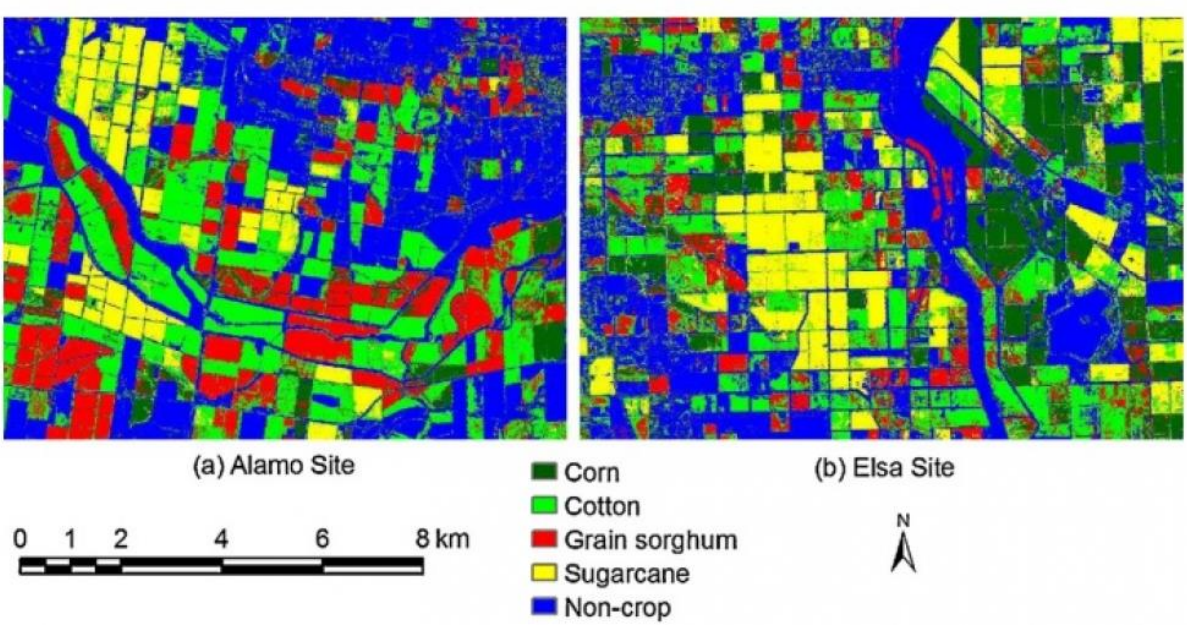


FIGURE 3.1 Five-class classification maps generated based on the *MLC* (Petropoulos et al. 2012)

Petropoulos et al. 2012, obtained hyperion imaginary dataset from United States Geological Survey (USGS) archive for study area north of the Athens, Greece, which was collected from Advance Land Imager (ALI) sensor. In this study, land use classification was conducted by applying the *SVM* and *Artificial Neural Network (ANN)* to classify 10 different classes namely conifer forests, broadleaved forests, sparsely vegetated areas, heterogeneous agricultural areas, sclerophyllous vegetation, transitional woodland/ scrubland, bare rocks, urban areas, burnt areas and sea. It was observed that the accuracy of *SVM* was 89.26% while the accuracy of *ANN* was 85.95%. McNairn et al. 2014, used TerraSAR-X and RADARSAT-2 data for corn and soybean farm images in early vegetative growth corn (at approximately fourth collar development) and in soybeans (first trifoliate emerged). They have applied the *DT* classifier to classify the crops into corn, soybean and pasture-forage classes. The *DT* classifier was run on data stacks: TerraSAR-X multi-temporally filtered images, TerraSAR-X spatially filtered images, RADARSAT-2 multi-temporally filtered images, RADARSAT-2 spatially filtered images, and TerraSAR-X and RADARSAT-2 multi-temporally filtered images. It was observed that the accuracy of *DT* was above 90%.

Studies of ML applications on remote sensing data indicates that it is not only limited to crop yield estimation but also other activities related to crops like nitrogen concentration in sugarcane leaf (Abdel-Rahman et al. 2013) and losses of crops due to flood (Tapia-Silva et al. 2011).

3.1.2 Soil moisture content estimation from remote sensing data

For the water cycle, soil moisture (SM) is a significant variable as it regulates the penetration during precipitation events (Rodriguez-Iturbe et al. 1999) and affects the availability of water. Information about the accurate spatial soil moisture distribution is of great importance in hydrological applications, such as irrigation scheduling, *PA* and prediction of the flood in extreme rainfall (Heathman et al. 2003). Estimation of soil moisture has been carried out by several methods and *ML* is one of them.

Baghdadi et al. 2012, developed an approach to estimate soil surface parameter from remote sensing data available from Synthetic Aaperture Radar (SAR) called RADARSAT-2 dataset. The study site is located in the Thau basin near Montpellier in the South of France. The objective was to develop a technique to use a neural network to estimate moisture content and it was observed that the *ANN*'s performance in *Root Mean Square Error (RMSE)* was 2.0 %.

Zamaz et al. 2012, applied *Relevance Vector Machines (RVMs)* and *SVM* to build estimation functions. The results obtained for both machines are then compared. The data used for this study were a part of soil moisture experiments 2002 conducted at Ames, Iowa. Three different models were built based on a different set of input to the classifier and it was observed that the performance in terms of *RMSE,* which was 1.7% and 1.4% respectively for *SVM* and *RVM*.

Lakhankar et al. 2009, implemented three methods: multivariate regressions, *ANN* and *fuzzy logic* to estimate soil moisture on RADARSAT-1 datasets. From the experiment, it was observed that *ANN*: *RSME*=3.39%, *Fuzzy logic*: *RSME*=3.45% and *multivariate regressions*: *RSME*=4.48% and it was indicated that fuzzy logic and neural network models performed better compared to multiple regression. Xie et al. 2014, implemented *Back Propagation Neural Network (BPNN)* on data of Sichuan Middle Hilly Area, South-west China and remote sensing images are taken from the Advanced Microwave Scanning Radiometer of the Earth Observing System (AMSR-E). It was observed that the performance of *BPNN* was *RMSE*= 3%.

3.2 Machine learning applications in the prediction of agriculture crop production

Prediction of crop yield is a significant aspect of the agrarian economy like India, as various factors affect the quality and quantity of the crop production. These factors are due to natural phenomenon and/or human activities associated with farming.

It is desirable to have a system which can be implemented with the proper understanding and consideration of factors affecting the crop yield and can predict the yield of particular crop well ahead of its harvest. Such a system will definitely help the farmers, the policy makers and the manufacturing industries associated with farming for applying appropriate measures. Various *ML* techniques are applied in agricultural related activities and it is found that the *ML* techniques are helpful to provide alternative solutions (McQueen et al. 1995).

Smith et al. 2009, developed a model to using *ANNs* to predict year-round temperature, as extreme temperature adversely affects the growth of plants. The dataset used is from Automated Environmental Monitoring Network (AEMN) system to collect weather data from sites across the state of Georgia. It was observed the developed model having accuracy more than previous models. Chen et al. 2016, studied the application of *SVM* in defining the comparative importance of various climate factors i.e. rainfall, relative humidity, temperature, sunshine hours and rainy days to rice yield variation in southwestern China. Moreover, *SVM* was compared with traditional *ANN* and *multiple linear regression*. It was observed that the *SVM* with radial basis function performed best.

Veenadhari et al. 2014, developed a web based system based on *DT*. In this system, they have studied the influence of climatic parameters includes rainfall, maximum & minimum temperature, potential evapotranspiration, cloud cover and wet day frequency on the production of soybean, maize, paddy and wheat from five districts of Madhya Pradesh. It was observed that the accuracy of the model was above 80% for most of the crops except accuracy of maize yield prediction was 76%. Gandhi et al. 2016, applied an *SVM* to predict the yield prediction of rice in Maharashtra district. The dataset was of Kharif season of rice production from the years 1998 to 2002. In this study, total six parameters were taken for prediction namely area, evaporation rate, maximum temperature, average temperature, minimum temperature and precipitation.

Petridis et al. 2003, implemented a modified version of *k-nearest neighbor (k-NN)* algorithm called *Fuzzy Interval Number k-Nearest Neighbor (FINkNN)* classifier which operates on the set of conventional interval-supported convex fuzzy sets to predict the sugar production from the crop of sugar-beet in Greece. In this study dataset was consists of ten production and eight meteorological variables. It was observed that the *FINkNN* classifier obtaining minimum prediction error with an expert input variable selection. Papageorgiou et al. 2013, implemented *PA* based on a fuzzy cognitive map which was applied to predict the yield of apples. The study was carried out at the apple orchard located at Agia, central Greece. The experiment lasted 3 years (2005-2007) that included soil, yield and quality mapping. The interesting part of this study was that the properties of soil i.e. soil texture, phosphorus (P), calcium (Ca), potassium (K) available zinc (Zn) and organic matter (OM) concentration were taken in consideration. It was observed that the accuracy of the proposed system was high at 75% in comparison with an accuracy of DT at 53.37% and *Bayesian Network (BN)* at 44.64%.

Chen et al. 2013, presented the application of *SVM* to estimate daily solar radiation using sunshine duration for the Liaoning province situated southern part of Northeast China. The radiation from sunshine is of the important factors for the development of the crop. In this study 7 *SVM* models were implemented using different input attributes. These 7 *SVM* models were compared with five empirical sunshine-based models. It was observed that *SVM* shows good generalization, and all the *SVM* models give good performances, and the developed *SVM* models outperform the empirical models. Khoshnevisan et al. 2004, implemented *Adaptive Neuro-Fuzzy Inference System (ANFIS)* was employed to forecast greenhouse strawberry yield on the basis of a different combination of energy inputs. The study was conducted at Guilan province of Iran. The greenhouse strawberry production energy inputs namely human labor, chemical fertilizers, farmyard manure (FYM), diesel fuel, electricity, natural gas, biocides, machinery and water for irrigation were considered. It was observed that in the comparison between *ANN* and *ANFIS* model the *ANFIS* model can predict yield relatively better than *ANN* model. Navarro-Hellín et al. 2016, designed and developed an automatic decision support system to manage irrigation in agriculture using *ANFIS* model. The main characteristic of

the system was the use of continuous soil measurements to complement climatic parameters to precisely predict the irrigation needs the crops.

3.3 Machine learning applications in animal husbandry

Agricultural crop production and livestock are inherently linked to each other as both are vital for the country's food security. Livestock is also an important income generating activity for most of the farmers. *ML* methods have the potential to be alternative methods to predict the livestock production such as meat, milk, egg, etc. (Ciham et al. 2017) . Also, *ML* techniques are used in the diagnosis of animal diseases.

Slósarz et al. 2011, attempted an experiment by applying *ANN* on a dataset of the muscular parts of the lamb to estimate fat content. It was observed that the relationship between intramuscular fat content and the body weight was weak, while the important relation exists between the body weight and the lamb's age before getting slaughtered.

Mcevoy et al. 2013, evaluated two *ML* techniques, *ANN* and *Partial Least Squares Discriminant Analysis (PLS-DA)* to classify hip images in canine. In this study, 120 images of the hip area of canine were used to training *PLS-DA* and *ANN*. It was observed that the classification error, sensitivity and specificity of *PLS-DA* and *ANN* was 6.7%, 100%, 89% and 8.9%, 86%, 100% respectively. Caraviello et al. 2006, implemented *DT, BN and instance-based algorithms* to determine the factors affecting the first-service conception rate and pregnancy in cows. For Holstein cows, the pregnancy status rate was 71.4% and for the rate of the first-service conception was of 75.6%.

Meyer et al. 2007, applied *Principal Component Analysis (PCA)* to identify traits to model the genetic covariance structure among six traits observe on the live animal (Angus cow) and eight traits recorded from the carcass. It was observed that seven traits out of fourteen were enough for breeding genetic improvement programs of Angus cow. Casanova et al. 2012 applied *PCA* on morphological properties of cows to classify them into two categories of French and Spanish breeds. It was observed that the primary principal component was the length of face and skull with a variance of 49.9% while secondary principal component was with of head and skull with a variance of 19.2%.

Yunusa et al. 2013, examined the morphological structure of breeds Nigerian Uda and Balami sheep. It was observed that the significant features for identifying both breeds were features linking to bone development and cranial measurements. Warns-Petit et al. 2010, used wildlife necropsy data to group the wild animals by applying *k-Means* clustering to identify most frequent diseases. It was observed that *k-Means* was a helpful technique by obtaining 9 clusters representing the most occurring diseases.

Dupuy et al. 2013, applied *Multiple Factor Analysis (MFA)* in combination with clustering methods on cows' health-related and demographic data, which led to 12 clusters. Out of these 12 clusters, two clusters were linked to the slaughtering process, one cluster was of a specific disease, etc. Bank et al. 2015, studied the characteristics of bacterial composition in the gut of neonatal piglets and diarrhoeic piglets. They applied k-means clustering method to characterize the bacterial composition and noticed that the diarrohoeic piglets affect the neonatal piglets in the first week of birth. Nantima et al. 2015, applied the ward's hierarchical clustering method to determine the number of clusters to identify the risk aspects related with the existence and spread of African swine fever among the pig farmers. The prominent differences were observed among the three clusters. Pelaez et al. 2008, implemented *logistic regression*, *DT* and *factor analysis* to classify the existence of the infectious disease in cattle. They have observed the high risk of regions by identifying that the dense population and frequent movement of cattle. Hempstalk et al. 2015, studied eight various *ML* methods to estimate the success of conception and insemination in dairy cows. It was observed that the performance logistic regression was best.

3.4 Machine learning applications in soil science

As discussed earlier, *ML* is a common term for a broad set of models which discovers patterns from data to make predictions. In soil science, *ML* techniques have been applied to a broader range of problems.

Boer et al. 2016, used a maximum likelihood classifier to model soil depth classes. This study was carried out in the semi-arid zone of southeast of Spain. *PCA* was carried out on the maps of the following terrain attributes: slope angle, wetness index, specific catchment area and length-slope factor. Cross validation yielded 61-81% accuracies for the maps of the shale are, 40-55% for the maps of the phyllite area, and 72-78% for the limestone area.

Behrens et al. 2006, studied comparison of 10 different *ML* algorithms to classify soil terrain data into soil classes in the study area of southern Rhineland Palatinate between Kaiserslautern in the north and the German–French border in Germany. This study focuses on *ANNs, DTs, linear regression model, linear vector quantization* and *SVMs*. All methods are compared in terms of two class problem, where each soil class is extrapolated separately.

Li et al. 2014, adopted *ML* like *SVM, Multiple Linear Regression (MLR)* and *ANNs* methods to evaluate soil nutrients. In this study soil parameters like the content of organic matter, total nitrogen, alkali-hydrolysable nitrogen, rapidly available phosphorus, and rapidly available potassium as independent variables, while the rank of soil nutrient content was taken as dependent variable. It was observed form the results that the average prediction accuracies of *SVM* models were 77.87% and 83.00%. Bhattacharya et al. 2006, investigated the problem with a specific interest in automating classification of soil layers from measured data. In this study, the classifier was built called *Constraint Clustering and Classification (CONCC),* which can be used in the automatic classification with the constraint of contiguity from soil samples data.

3.5 Recent research of applications of *ML* in agriculture

Richardson et al. 2016, applied *ML* technique for interpretation of pathology tests. The practical use of ML for enhanced prediction in clinical biochemistry is illustrated using data obtained from routine pathology testing performed in Australia.

Cramer et al. 2017, focus on the prediction of rainfall over specific period. Seven *ML* algorithms, namely: *support vector regression, markov chain, genetic programming, radial bias neural network, M5 rules, M5 model trees* and *k-Nearest Neighbour* applied to evaluate predictive performance. From results it is deduced that *radial basis function, support vector regression* and *genetic programming* were the best algorithms.

Rahee et al. 2017, developed a high resolution meteorological drought forecast model to provide forecast information. The aim of the study was to develop a drought forecast model based on the combination of remote sensing and long-range forecast data using machine learning for ungauged areas, also to provide improved ranges of drought forecast in case of the improvement of forecasting skill of the long-range forecast data. In this study *ML* models *DT,*

RF, ERT where compared to standard statistical models *SPI* and *SPEI*. It was found that *ERT* model was best in predicting drought accuracy.

Aybar-Ruiz et al. 2016, had proposed a novel scheme for global solar radiation prediction, based on *hybrid neural-genetic algorithm*. The merging of *grouping genetic algorithm (GGA)* and an *external learning machine algorithm (ELM)* have been merged in a single technique, such that the *GGA* solves the optimal selection features, and the *ELM* performs the prediction task.

Zhou et al. 2018, proposed automatic feeding decision making based on the appetite of fish, which is based on *near infrared computer vision* and *neuro-fuzzy* model was proposed. The algorithm called the *adaptive network-based fuzzy inference system (ANFIS)* was developed, which shows feeding accuracy of 98%.

Fragni et al. 2018, proposed a novel mineral elements database for the authentication of Italian processed tomato. Optical emission spectrometry were used for quantifying 26 minerals. *LDA* was applied to discriminate between Italian and non-Italian tomato samples. In the experiment 98.8% cross-validation prediction ability was achieved.

Maione et al. 2018, reviewed multivariate data analysis and data mining techniques when combined with specific parameters for ascertaining authenticity and many other useful characteristics of rice, such as quality, yield and others. It was found that, *LDA* and *principal component analysis* are recommended methods, but numerous additional data analysis classification techniques: like: *SVM, ANN*, etc. presents high performance for discrimination of rice.

Ebrahimi et al. 2017, utilized a new image processing technique to detect parasites of strawberry plants. For classification by *SVM*, the ratio of major diameter to minor diameter as region index as well as Hue, Saturation and Intensify as color indexes were utilized to design the *SVM* structure. *Mean square error (MSE), root of mean square error (RMSE), mean absolute error (MAE)* and *mean percent error (MPE)* were used for evaluation of the classification. Results show that using *SVM* method with region index and intensify as color index make the best classification with mean percent error of less than 2.25%.

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 neatest 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.

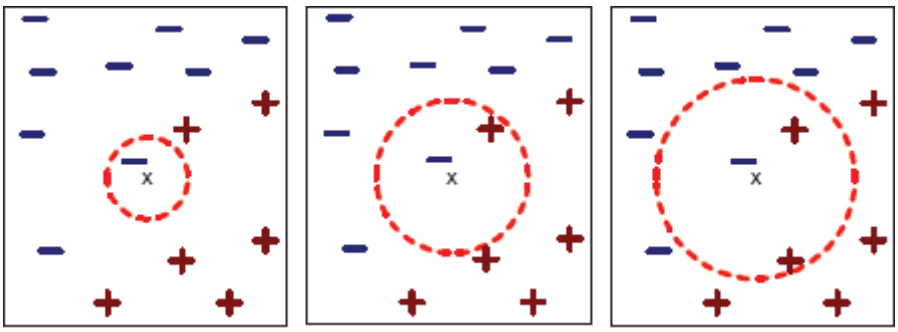
4.1.2 The *k-NN* rule

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).



(a) (b) (c)

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)* ≥ *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)* ≤ *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.

$$D(X,Y) = \sqrt{\sum_{i=1}^{m} (xi - yi)^2} \tag{4.2}$$

- 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:

$$D(X,Y) = \sum_{i=1}^{m} |xi - yi| \tag{4.3}$$

- 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).

$$D(X,Y) = \left(\sum_{i=1}^{m} |xi - yi|^r \right)^{1/r} \tag{4.4}$$

- 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.

$$D(X,Y) = \sum_{i=1}^{m} \frac{|xi - yi|}{|xi + yi|} \tag{4.5}$$

- 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).

$$D(X,Y) = \max_{i=1}^{m} |xi - yi| \tag{4.6}$$

- 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.

$$S_{\cos} = \frac{\sum_{i=1}^{m} xiyi}{\sqrt{\sum_{i=1}^{m} xi^2} \sqrt{\sum_{i=1}^{m} yi^2}} \tag{4.7}$$

- 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.

$$D(X,Y) = \frac{\sum_{i=1}^{m} (xi - \mu i)(yi - \mu i)}{\sqrt{\sum_{i=1}^{m} (xi - \mu i)^2 \sum_{i=1}^{m} (yi - \mu i)^2}} \tag{4.8}$$

- 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.

$$D(X,Y)=\sum_{i=1}^{m}\frac{1}{sum_i}\left(\frac{xi}{size_x}-\frac{yi}{size_y}\right)^2 \tag{4.9}$$

- 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.

$$D(X,Y)=\sum_{i=1}^{m} yi^i \log\frac{yi^i}{xi^i}-yi^i+xi^i \tag{4.10}$$

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.

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

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

$$d(x_a,y_a)=\begin{cases}1,(if\_x_a\_or\_y_a\_is\_unknown)\\ overlap(x_a,y_a),\ (if\_a\_is\_nominal)\\ rn\_diff(x_a,y_a)\end{cases} \tag{4.12}$$

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

$$d(x_a,y_a)=\begin{cases}0,(if\_x_a=y_a)\\ 1,(otherwise)\end{cases} \tag{4.13}$$

and the range-normalized difference function is Eq. 4.14

$$rn\_diff(x_a,y_a)=\frac{|x_a-y_a|}{range_a} \tag{4.14}$$

The range is used to normalize the attribute values and is defined as Eq. 4.15

$$range_a = \max_a - \min_a \tag{4.15}$$

Here, from the training set, the values of *maximum* (max_a) and *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 Euclidian 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 of 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)

```
X=TS
S=X
for xi = each_instance_of (X) do
   kNearestNeighbours( xi, X – {xi})
   if kNN(xi) != Θi then
   S = S – {xi}
   end if
end for
```

Let $\{ X , \Theta \} = \{ (x_1, \theta_1), (x_2, \theta_2), ...,(x_n, \theta_n) \}$ 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-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.

ALGORITHM 4.2: Hart's condensing algorithm (Hart 1968)

```
X=TS
S= ϕ
repeat
for xi = each_instance_of (X) do
        nearest neighbor(xi,S)
        if 1-NN(xi) != Θi then
        X = X – xi
        S = S + xi
   end if
 end for
 until (eliminated_instances( X  = = 0) or ( X = = ϕ)
```

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 nonborder 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 neatest 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.

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 todecision 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.

CHAPTER - 5

Issues and Challenges

In the previous chapter, we have gone through state of the art literature related to *k-Nearest Neighbour (k-NN)* classifier and its extension as *fast k-NN*. In this chapter, we'll look at challenges pertaining to *Soil Health Card Database (SHCD)* and selection of proposed classification methods to deal with these challenges.

5.1 Soil health card data set and macro-micro nutrients deficiency

This research work is concentrated on exploring the applicability of *Machine Learning (ML)* techniques on an Agricultural dataset of soil health card. The agricultural dataset is collected from *SHCD*, which is available at *Anand Agriculture University*. Our research aims at proposing an improved efficient *ML* algorithm to classify soil sample into the categories of the deficiencies of *micro* and *macro* nutrients.

As discussed in Chapter1: Section 1.2, *SHCD* for *Gujarat* state stores and maintains soil health card data of individual farm of all districts of Gujarat. Every district of Gujarat has its respective database table in *SHCD* having **49** attributes varies from identification of soil sample to soil characteristics of the particular land. This *SHCD* has maintained in *Microsoft Structured Query Language (MSSQL) DBMS* system. For our research purpose, we were provided *SHCD* of six districts namely *Kutch*, *Rajkot*, *Banaskantha*, *Vadodara*, *Anand*, and *Surat*.

Out of 49 attributes in *SHCD*, we are concerned with *macro* and *micro* nutrients, henceforth we have generated *Soil health card dataset (SHCDS)* which consists of four *macro* and four *micro* nutrients and a label assigned to each sample indicating deficiencies in it. As it is shown in Fig. 5.1, in *SHCDS* each row is an entry of individual farm having *macro* nutrients *SHC_POTASS*, *SHC_SULPHUR*, *SHC_MG*, *SHC_PHOSPHORUS*, *micro* nutrients *SHC_IRON*,

SHC_MANGANESE, *SHC_ZINC*, *SHC_CU* and a corresponding class label indicating one or more attributes deficiency.

	A	B	C	D	E	F	G	H	I
1	SHC_POTA	SHC_SULP	SHC_MG	SHC_PHOS	SHC_IRON	SHC_MAN	SHC_ZINC	SHC_CU	lable
2	240	42	2	27	2	6.3	7.5	0.25	MaMi210
3	113	4.5	4.5	67	0.25	0.25	1	14.2	MaMi49
4	146	2.5	260	26	0.45	7	54	7.9	MaMi51
5	248	5.6	5.3	30	0.25	35	2.5	13.6	MaMi183
6	214	6.3	2	59	0.52	7.5	2.5	13.2	MaMi147
7	274	6.6	4.6	23	0.25	1	1.5	13	MaMi179
8	242	3.5	5.4	48	0.36	5.5	1	13.5	MaMi177
9	232	2.5	5.5	34	0.25	7.5	1	7.9	MaMi177
10	123	2	2	20	0.36	7.5	4.5	14.2	MaMi3
11	325	5.2	5.5	15	0.63	4.5	1	14.5	MaMi161
12	392	5.2	6.5	11	0.42	0.45	2.5	8.7	MaMi163
13	147	47	4.5	25	0.25	8	2	12.5	MaMi115
14	110	3.6	7	57	0.36	6.3	3.5	13.5	MaMi51
15	162	2.5	45	38	0.32	0.25	1	142	MaMi177

FIGURE 5.1 Sample of *Soil health card data set (SHCDS)* (Soilhealth 2017)

Based on different macro and micro deficiency, recommended soil fertilizer treatments are mentioned below (Soils 2017):

For Macro soil nutrients parameters:

- Potassium (*K*): if the content of *K* <=150 ppm, than soil needs potassium fertilizer treatment.
- Sulfur (*S*): if the content of *K* <= 20 ppm, than soil needs sulfur fertilizer treatment.
- Magnesium (*Mg*): If the content of *Mg* <= 2 ppm, than soil needs treatment.
- Phosphorus (*P*): It the content of *P* <= 20 ppm, than soil needs phosphorus treatment.

For Micro soil nutrients parameters:

- Iron (*Fe*): if the content of *Fe* <=10 ppm, than soil requires Ferrous Sulfate.
- Manganese (*Mn*): if the content of *Mn* <=10 ppm, than soil requires Manganese Sulfate.
- Zinc (*Zn*): if the content of *Zn* <=1 ppm, than soil requires than Zinc Sulfate.
- Copper (*Cu*): if the content of *Cu* <= 0.4 ppm, than soil requires Copper Sulfate.

5.2 Challenges of soil health card dataset

Massive data: According to information provided on the website of the Department of Agriculture, Cooperation & Farmers Welfare, India, farmers are provided soil health cards for their farms in soil health card scheme, which has two cycles. In the first cycle, soil health cards will be distributed to farmers. And data of 5,41,80,895 health cards will be made available on the website. In the second phase 4,15,66,644 soil health cards will be prepared and distributed will be on the website **(Soilhealth 2017)**. Hence, total 9,57,47,539 of data (huge data set) of health card are available on the web portal.

	A	B	C	D	E	F	G	H	I
1	SHC_POTA	SHC_SULP	SHC_MG	SHC_PHOS	SHC_IRON	SHC_MAN	SHC_ZINC	SHC_CU	lable
2	240	42	2	27	2	6.3	7.5	0.25	MaMi210
3	113	4.5	4.5	67	0.25	0.25	1	14.2	MaMi49
4	146	2.5	260	26	0.45	7	54	7.9	MaMi51
5	248	5.6	5.3	30	0.25	35	2.5	13.6	MaMi183
6	214	6.3	2	59	0.52	7.5	2.5	13.2	MaMi147
7	274	6.6	4.6	23	0.25	1	1.5	13	MaMi179
8	242	3.5	5.4	48	0.36	5.5	1	13.5	MaMi177
9	232	2.5	5.5	34	0.25	7.5	1	7.9	MaMi177
10	123	2	2	20	0.36	7.5	4.5	14.2	MaMi3
11	325	5.2	5.5	15	0.63	4.5	1	14.5	MaMi161
12	392	5.2	6.5	11	0.42	0.45	2.5	8.7	MaMi163
13	147	47	4.5	25	0.25	8	2	12.5	MaMi115
14	110	3.6	7	57	0.36	6.3	3.5	13.5	MaMi51
15	162	2.5	45	38	0.32	0.25	1	142	MaMi177

FIGURE 5.2 Presence of redundant instances in *SHCDS* (Soilhealth 2017)

Superfluous data: Human experts collect soils samples for each farm and test the sample in soil testing laboratories and record the macro and micro parameters pertaining to the sample into *Soil Health Card*. Each record in the soil health card dataset is generated from a soil sample taken from the farm. Each record contains the value of micro, macro nutrients of the soil sample and farm id, village id, taluka id, district id, etc. As the soil samples are taken from adjacent farms, it is highly possible that a large number of records have similar values in a range of micro and macro nutrients attributes in the datasets. Fig. 5.1 shows one such example of redundant data records, which are shown as in yellow and green background color. The records having same values create the problem of data redundancy for machine learning algorithms. It is desirable to classify records without superfluous training records (Ohno-Machado et al. 1998).

	A	B	C	D	E	F	G	H	I
1	SHC_POTA	SHC_SULP	SHC_MG	SHC_PHOS	SHC_IRON	SHC_MAN	SHC_ZINC	SHC_CU	lable
2	240	42	2	27	2	6.3	7.5	0.25	MaMi210
3	113	4.5	4.5	67	0.25	0.25	1	14.2	MaMi49
4	146	2.5	260	26	0.45	7	54	7.9	MaMi51
5	248	5.6	5.3	30	0.25	35	2.5	13.6	MaMi183
6	214	6.3	2	59	0.52	7.5	2.5	13.2	MaMi147
7	274	6.6	4.6	23	0.25	1	1.5	13	MaMi179
8	242	3.5	5.4	48	0.36	5.5	1	13.5	MaMi177
9	232	2.5	5.5	34	0.25	7.5	1	7.9	MaMi177
10	123	2	2	20	0.36	7.5	4.5	14.2	MaMi3
11	325	5.2	5.5	15	0.63	4.5	1	14.5	MaMi161
12	392	5.2	6.5	11	0.42	0.45	2.5	8.7	MaMi163
13	147	47	4.5	25	0.25	8	2	12.5	MaMi115
14	110	3.6	7	57	0.36	6.3	3.5	13.5	MaMi51
15	162	2.5	45	38	0.32	0.25	1	142	MaMi177

FIGURE 5.3 Presence of attribute noise in *SHCDS* (Soilhealth 2017)

Noisy data: There are many factors having huge impact on the data collection process in any real world applications. The factors can be a data source, techniques to collect the data, the sampling period etc. In this process, human experts could make mistakes and those mistakes may be reflected in the database. Sometimes, the equipment used to test a soil sample in laboratories gave values which contain errors. Such records containing the errors are referred to as noise. Fig. 5.3 shows one such sample dataset which contains attribute noise is highlighted by pink background color on a particular field (Zhu et al. 2004). Even though great efforts are made, still data collection process remains error prone, and according to some specific study it is estimated that there are at least 5% of errors in the dataset (Wu et al. 2004, Maletic et al. 2000). The classification accuracy heavily depends on the quality of training data, but the presence of noise in training data may affect the performance of classification problem (Zhu et al. 2004), hence it is desirable to run classification algorithm without a noisy dataset.

5.3 Choice of classifier and instance reduction techniques

5.3.1 Choice of *k-Nearest Neighbour* as a classifier

There are many supervise classifiers *k-NN, Support Vector Machine (SVM), Decision Tree (DT), Naïve Bayes (NB)* etc. According to the *No Free Lunch Theorem (NLT)*, in supervise classification there is no such method which outperformed all others. For every method, there is a dataset for which the method fails. In other words, all *Machine Learning (ML)* methods are equally good if we do not place strong assumptions on the input dataset (Wolpert et al. 2002, Wolpert et al. 1997). Bhavsar & Ganatra 2012, compared five classifiers and concluded that each algorithm has its own set of advantages and disadvantages.

To calculate the probability of features in *NB*, we need to make an assumption regarding data distributions. *NB* assumes that attributes are conditionally independent to each other given the class. Hence decision boundary for *NB* results into linear, elliptic form. That would create problems for large datasets because the assumption might produce a negative effect on the performance of classification. However, *k-NN*'s decision boundary can take on any form this is because *k-NN* is non-parametric, i.e. it makes no assumption about the data distribution. Ashari et al. 2013, implemented k-*NN*, *NB* and *DT* classifiers to simulate energy tools used before it is built. For this research total 13 parameters of building taken. It was observed that the *k-NN* has lowest classification time and best average precision and recall than *NB* and *DT* classifiers.

The *SVM* classifier is not suitable when the dataset is very large because it requires a large amount of training time to find support vectors and decision hyperplane. However, *k-NN* is a non-linear model and effective in case of very large data and having low dimension. Moreover, the *k-NN* classifier is not a model based, it does not loose any detail and compares every training sample to give the prediction. The main advantage of such a memory-based approach like the *k-NN* is that the classifier immediately adapts to new training data. Amancio et al. 2014 implemented nine different classifiers to classify dataset having ten features, it was observed that *k-NN* classifier provides by a large margin, the highest average accuracy than other classifiers like *SVM*, *NB*, *Random forest* etc. Kuramochi et al. 2005 observed that *k-NN* outperformed the support vector machine classifier for prediction of protein function (in details) using expression profiles. The *decision tree classifier* requires a lot of pruning in finding

optimal decision tree. It is very sensitive to noise and could generate a different decision tree in presence of noise.

As mentioned earlier, this research focuses on the specific trait of soil nutrient deficiencies classification on *SHCD*, and for the current research purpose, we took sample dataset of district Kutch in Gujarat with 14000 entries. In future, if we consider the full soil health card scheme dataset of India, which is a too large dataset, naturally we are inspired to take *k-NN* as classification technique.

5.3.2 Choice of *TRS-kNN* (*prototype selection*)

As mentioned in Section 5.2, soil health card dataset contains many redundant records. There are two major disadvantages of dataset containing redundant records. The first disadvantage is the redundant record will require a large amount of storage space which would make inefficient use of memory. The second one is if we apply traditional *k-NN* on such dataset then it will perform unnecessary calculation between redundant training records and test data. Hence time requires to classify test data will increase compared to if the dataset without redundant records.

TRS-kNN classifier bases on state of the art *prototype selection* (Wilson et al. 2000, Garcia et al. 2013, Jankowski et al. 2004), which aims at reducing the original training set size. To select reduced training set it is important to observe the performance of classifiers because it should be same either original or reduced training set it took.

In Chapter 6: Section 6.3, state of the art training set reduction *k-NN* is implemented to remove redundancy from the training dataset. For that, we have applied the condensation method, which removes nonessential instances that will not affect the classification accuracy.

5.3.3 Choice of *F-kNN* (*prototype generation*)

The *k-Nearest Neighbor* classifier is considered as lazy learning method as it does not attain the learning model and this algorithm is storing the complete training data (Garcia et al. 2012). As mentions in Section 5.2., the actual soil health card dataset is very large, which can be the bottleneck to implement a *k-NN* classifier because. For training purpose, the complete dataset has to be stored because they define the decision rule for *k-NN* classification. Furthermore, the storage of large training examples leads to high computational cost for the decision rule computation. Moreover,

the traditional *k-NN* algorithm is sensitive to noise (Wilson et al. 2000), as mention in Section 5.2, soil health card database is prone to noise due to multiple factors.

Hence, to overcome the drawbacks of high storage, high computation cost and sensitivity to noisy data of soil health card data, it is desirable to modify the traditional *k-NN* algorithm so that it can perform accurate and speedy classification. Looking at the scenario, we have implemented novel *F-kNN* classifiers in Chapter 6: Section 6.2 and 6.5, by editing training set based on prototype generation method, *F-kNN* classifiers are based on generating a prototype of training space by applying clustering (Zhang et al. 2004, Ougiaroglou et al. 2012, Franti et al. 2006, Yong et al. 2009) on it. The prototype generation method (*F-kNN*) is different from the prototype selection method (*TSR-kNN*) in terms of training set generation. In case of prototype generation, the new training set is generated from existing training set (applying *k-means* in *F-kNN*), while in prototype selection method it choses relevant examples without generating new data points.

In this research, for clustering of training space, we have implemented a *k-Means* clustering technique (Chapter 6: Section 6.2.1). Moreover, we have proposed an optimization on the number of clusters for *k-Means* clustering, for that two methods are implemented namely *elbow* method and *silhouette value* (Chapter 6: Section 6.2.2). In this experiment at first, either of *elbow* or *silhouette value* methods is applied on training data, which in result suggest the optimized value of *k* for *k-Means* and further *k-Means* clustering will generate clusters centroids, which is edited set and it is very less in size in comparison to the original training set. This edited set it is the prototype of the original training set, and it participates in the training phase. Hence, by means of applying the *F-kNN* method, we have tried to overcome the issue of large storage requirement, computational cost and handling noise of large dataset in our case *SHCD*.

5.3.4 Choice of hybrid method *TSR-FkNN*

As discussed in previous Sections 5.3.2 and 5.3.3, the approaches of *F-kNN* and *TRS- kNN* having distinguished property of dealing with noisy and redundancy respectively in case of huge data set such as soil health card database. Taking inspiration from the advantages of *F-kNN* and *TRS-kNN*, we have proposed a novel hybrid method called *Training Set Reduction Fast k-NN (TSR-FkNN)*. This novel hybrid method is implemented such that, first it removes redundant

samples by applying prototype selection to obtain a reduced training set and then it removes noise by applying prototype generation by *k-Means* clustering.

5.3.5 Comparison of proposed methods with state of the art methods

As discussed earlier, for our research *PG* and *PS* techniques are adopted. In literature (Chapter 4: Section 4.6 and 4.7) there are various methods of *PG* and *PS* had been proposed and compared by looking at criteria, namely, "type of reduction/selection", "resulting generation set", "generation mechanism" and "evaluation of search" (Garcia 2012, Triguero 2012). The brief discussion of these criteria is discussed below, following by comparison of *PG* and *PS* methods (discussed Chapter 4: Section 4.6 and 4.7) in table 5.1 with proposed techniques of our research work.

Type of reduction/selection: It denotes the direction in which the search to find reduced/selected subset from the original training set is done. It can be incremental, decremental, batch, mixed and fixed. The incremental search begins with empty subset *S* and adds each instance from *Training Set (TS)* to *S* bases on predefined criteria. In decremental search, initially *S=TS,* then from *S* instances are removed. The batch search takes a batch of instances from *TS* at a time for removal based on some criteria. In a mixed search, initially, some instances are pre-selected in subset *S* and then based on some criteria instance are added or removed iteratively. The fixed search takes random instances from *TS* into *S*, and then a fixed number of the instance are added or removed based on some criteria into *S*.

Resulting generation set: This criterion denotes the type of search carried out is tending to retain border points, central points or some other set of points. The resulting generated set can be retaining points by the implementation of various generation techniques, namely, condensation, edition and hybrid. The condensation techniques are applied to retain border points and removal of internal points of feature space. The edition techniques are applied to remove the border points. In hybrid schemes, modification/removal of both border and internal points is carried out.

Generation mechanisms: This feature defines the different mechanism applied in the process to build the final reduced generated training set. A number of mechanisms are adapted in literature to achieve generated prototype set from the original set, namely, class re-labeling,

centroid based, space splitting, position adjustment. This criterion is applicable to only prototype generation schemes. The class re-labeling scheme involves changing class labels of instances from *TS* which could be error prone and appropriate to some other class. The centroids based scheme involves merging a set of similar instances and producing the centroids. In space splitting, different heuristics are applied to make a partition in the feature space to define a new prototype subset *S*. The position adjustment scheme the subset *S* is obtained by the optimization process of adding and subtracting quantities to the values of instances in subset *S*.

TABLE 5.1 Comparison of state of the art *PG/PS* methods with proposed methods

Sr. No	Method	Type of reduction/ selection	Resulting generation set	Generation mechanisms	Evaluation of search
1	*PNN* (Chang et al. 1974)	Decremental	Condensation	Centroids	Semi-wrapper
2	*VQ* (Xie et al. 1993)	Fixed	Condensation	Positioning adjustment	Semi-wrapper
3	*BTS3* (Hamamoto et al.1997)	Fixed	Condensation	Centroids	Semi-wrapper
4	*MCA* (Mollineda et al. 2002)	Decremental	Condensation	Centroids	Semi-wrapper
5	*IPCL2* (Lam et al. 2002)	Mixed	Hybrid	Centroids	Semi-wrapper
6	*MixtGauss* (Lazano et al.2006)	Fixed	Condensation	Centroids	Semi-wrapper
7	*SPG* (Fayed et al. 2007)	Incremental	Hybrid	Centroids	Semi-wrapper
8	*PSO* (Nanni et al. 2009)	Fixed	Hybrid	Positioning adjustment	Wrapper
9	*IPADE* (Triguero et al. 2010)	Decremental	Hybrid	Positioning adjustment	Wrapper
10	*CNN* (Hart 1968)	Incremental	Condensation	NA	Wrapper
11	*RNN* (Gates 1972)	Decremental	Condensation	NA	Filter
12	*All k-NN* (Tomek 1976)	Batch	Edition	NA	Filter
13	*ICF* (Bringhton et al. 2002)	Batch	Hybrid	NA	Filter
14	*IKNN* (Wu et al. 2002)	Batch	Condensation	NA	Filter
15	*ZanghTS* (Zhang et al. 2002)	Mixed	Hybrid	NA	Wrapper
16	*MSS* (Barandela et al. 2005)	Decremental	Condensation	NA	Filter
17	*FCNN* (Angiulli et al. 2007)	Incremental	Condensation	NA	Filter
18	*RKNN* (Fayed et al. 2009)	Batch	Condensation	NA	Filter
19	*F-kNN* (Prajapati et al. 2019)	Decremental	Edition	Centroids	Wrapper
20	*TRS-kNN* (Prajapati et al. 2019)	Decremental	Condensation	NA	Wrapper
21	*TRS-FkNN* (Prajapati et al. 2019)	Decremental	Hybrid	Centroids	Wrapper

State of the art PG/PS methods Proposed methods

Evaluation of search: This standard relates to the way in which the search of the prototype is evaluated. It is divided into three categories, namely, filter, wrapper and semi-wrapper. The filter techniques do not use *k-NN* rule during the evaluation phase, instead, it uses various heuristics. In the semi-wrapper scheme, use of *k-NN* is done on partial data to make evaluation decision. The wrapper techniques use *k-NN* on full reduced/selected prototype subset *S* for evaluation.

In table 5.1, serial no. 1 to 18 list the methods discussed in Chapter 4: Section 4.6 and 4.7, while serial no. 19 to 21 list the methods proposed for our research work. As it is indicated all proposed methods (*F-kNN*, *TRS-kNN* and *TRS-FkNN*) are based on decremental "type of selection/reduction" technique. For "resulting generation set" measure, the proposed algorithm F-kNN implements edition technique, while TRS-kNN and TRS-FkNN adopt condensation and hybrid techniques respectively. For "generation mechanism" standard, F-kNN and TRS-FkNN use the centroids technique, while TRS-kNN does not use any such mechanism, hence it is written as Not Applicable (NA). For "evaluation of search" measure, wrapper technique is adopted for all proposed methods (*F-kNN*, *TRS-kNN* and *TRS-FkNN*).

5.3.6 Choice of "type of reduction/election", "resulting generation", "generation mechanism" and, "evaluation of search" for proposed classifiers

Choice of decremental search ("type of reduction/selection") for proposed classifiers: The decremental selection/reduction begin with prototype subset *S* is the same as the *training set (TS)*. We have chosen the decremental method, so that the advantage of the availability of full training set instances can be exploited to take the decision by observing the whole of the training set. Though the initial computational cost in the decremental method is high as it checks every *TS* instance, if we can get greater storage reduction, then the reduced computational cost in latter stages can be less and compensated for the extra cost of the initial phase.

Choice of edition, condensation and hybrid techniques ("resulting generation set") for proposed classifiers: The *F-kNN* classifiers implements edition technique to generate resulting generated prototype subset *S*. Large number of *SHCD* contains attribute noise (Section 5.2), which may lead to incorrect class prediction in *k-NN* classification, hence edition method is implemented to deal with noisy instances. The *TRS-kNN* classifier adopts condensation technique to reduce the number of similar training instances as large number of *SHCD* records

contains superfluous instances (Section 5.2), which causes extra computational burden. The *TRS-FkNN* classifiers implement hybrid (condensation followed by edition) method to take the advantage of reduction of noise and removal of superfluous instances consecutively.

Choice of centroids ("generation mechanism") for proposed classifiers: The proposed *F-kNN* classifiers generates centroids from *TS* by merging similar prototypes by applying *k-Means* clustering. As discussed earlier *SHCD* contains attribute noise and to deal with it, unique optimized *k-Means* clustering by adapting *elbow method* and *silhouette value* techniques are implemented. The *TRS-FkNN* classifiers adopt condensation (reducing superfluous instances) followed by generation of centroids by optimized *k-Means* clustering.

Choice of the wrapper ("evaluation of search") for proposed classifiers: The classification of *SHCD* records is done by a *k-NN* algorithm. The training set for the *k-NN* classifier is now reduced to prototype set generated/selected by condensation and/or edition, which is free from anomalies like noisy and/or superfluous instances.

5.4 Limitation of proposed algorithms

The proposed algorithms works on soil health card dataset. As mentioned earlier this dataset contain macro and micro attributes which are numeric in nature. Hence, proposed algorithms are designed to work with numeric attributes. If any non-numeric attribute is to taken into consideration like any characteristic related to soil which is nominal or ordinal in nature could not be taken into the resultant vector space model.

CHAPTER - 6

Proposed Methodologies

In Chapter 4, we discussed the *k-Nearest Neighbour* (*k-NN*) classifier and its advantages and disadvantages. Moreover, in reference to the drawback of *k-NN* in terms of large storage need and computational cost, we have reviewed state of the art *Prototype Selection (PS)* and *Prototype Generation (PG)* techniques which generates a compact prototype of *Training Set (TS)*. The Chapter 5, covered issues with *Soil Health Card Database (SHCD)* and choice of proposed classification methods in context with *SHCD*.

In this chapter, we will see the implementation of the concept of *PS* and *PG* applied for efficient classification of soil samples into macro and micro nutrients deficiency using *k-NN* classifier on *Soil Health Card Database (SHCD)*. For that, we have discussed *PG* based *Fast k-Nearest Neighbour (F-kNN)* classifiers, *PS* based *Training Set Reduction k-Nearest Neighbour (TSR-kNN)* classifier and hybrid *Training Set Reduction F-kNN (TSR-FkNN)* classifiers which employs both *PG* and *PS* models.

6.1 *k-Nearest Neighbor* classifier applied on *SHCD*

We first applied *k-Nearest Neighbor classifier* (algorithm 6.1) on data set of district Kutch, which has 14000 samples of soil parameters from *SHCD*. In our experiment, we calculated accuracy and execution time for classification in milliseconds for each value of *k* in range (31, 33, . . ., 45).

ALGORITHM 6.1: *k*-Nearest Neighbour (*k*-NN) classifier (Prajapati et al. 2019)

Input: A set of Agriculture records $R = \{R_1, R_2, \ldots R_n\}$, where n is the total number of *SHCD* records.

Procedure:

Step 1: Divide the record data into one training set and one test set as 50-50 split.

Step 2: For each test record, calculate similarity with each training record.

Step 3: Sort the training records in the descending order of the maximum cosine similarity for each test record and select the top *k* training records.

Step 4: Assign a class to test record which occurs maximum times in the top *k* training records.

Step 5: Construct a confusion matrix.

Step 6: Calculate accuracy from confusion matrix.

6.2 *Fast k-Nearest Neighbor (F-kNN)* applied on *SHCD*

The primary limitation of the simple *k*-NN algorithm is that it needs to retain all the training data and prone to high computational cost. Moreover, the accuracy of classifier highly depends on the worth of the training set, and if there are mislabeled instances, noise and outliers present in the training set then it negatively affects its accuracy. The *SHCD* contain attribute level noise (Chapter 5: Section 5.2). Noisy instances are located at border points of decision boundaries in the vector space model, which can be removed without any ill effect on classification accuracy. Further, a number of similar instances can exist in proximity in vector space model, which can be represented as single centroid as representative of vectors in its vicinity. Hence, the edition of *TS* by means of generating centroids can lead to the edited training set, which can reduce the computation cost (in terms of classification time) of simple *k*-NN.

We have proposed *F-kNN* (editing algorithm), which is known in the literature as *PG* (Chapter 4: Section 4.6) (Garcia et al. 2012) techniques. The *PG e*diting algorithms we have implemented is based on finding centroids by applying *k-Means* clustering.

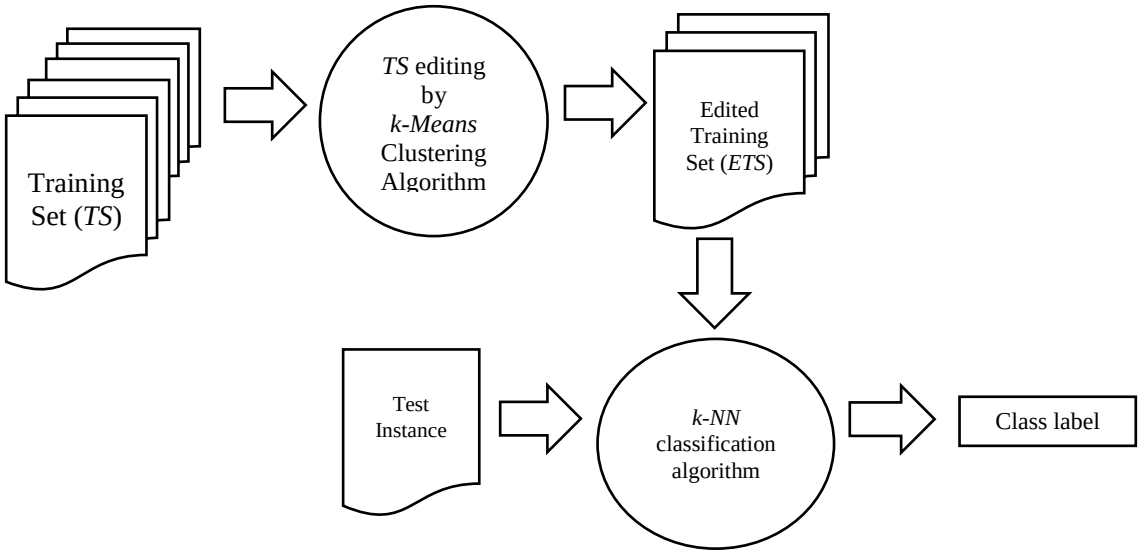


FIGURE 6.1 Overview of *Fast k-NN (F-kNN)* (Prajapati et al. 2019)

Fig 6.1 shows an overview of *F-kNN* classifier. It takes *TS* as an input set and applies *k-Means* clustering to produce *Edited Training Set (ETS)*, which contains *k* centroids representing training samples for *k-NN* classification. Now, the *k-NN* classification algorithm uses *ETS* to classify any test instance.

The aforementioned process of generating *ETS* by applying *k-Means* clustering for proposed *F-kNN* classifier is shown in Fig 6.2, for example, that the initial training set contains two classes, circle and triangle as shown in Fig. 5.2a. The second step computes two class-means. As in Fig. 6.2b, *k*-Means is performed on the training set to build two clusters. As in Fig 6.2c, cluster *P* is containing more than one item from the same class, thus its cluster centroid is placed in the edited set. *Q* is containing more than one item from different classes. Hence, again computes two class-means Fig. 6.2d, and then *k*-Means is applied on *Q* and clusters *R* and *S* Fig. 6.2e are built. Now, *R* cluster is having instances form same class and its cluster centroid is placed in the edited set. But *R* is having more than one instances from different classes, so its centroids are computed Fig. 6.2f and *k*-Means are applied on *R* to build clusters *S* and *T* Fig. 6.2g. Then, cluster centroid of the cluster *T* is placed in the edited set Fig. 6.2h, while centroids of cluster *S*

are computed, which generates cluster *U* and *V*, both are having homogeneous instances Fig. 6.2i. The final edited set is consisting of only four items in Fig. 6.2j.

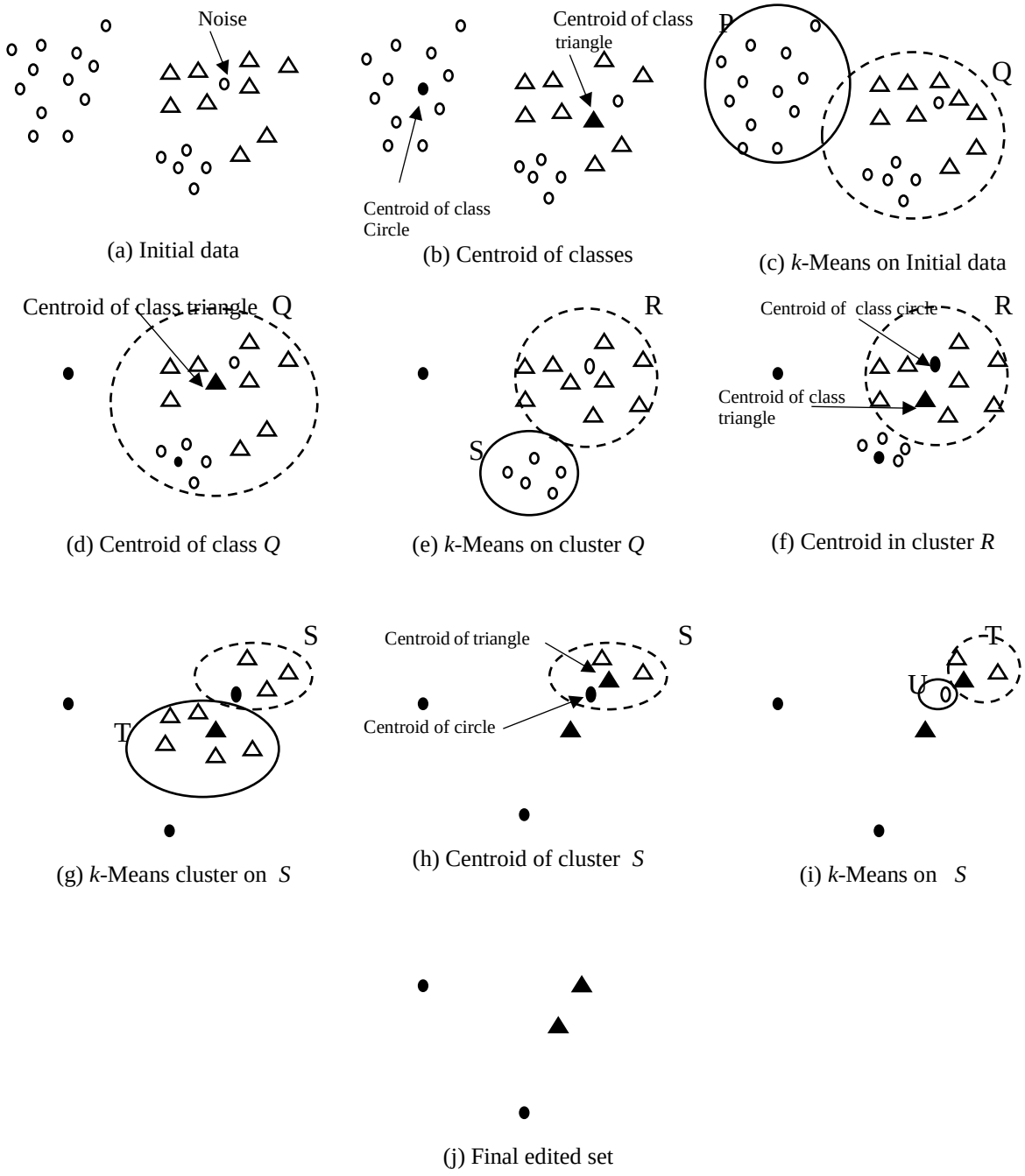


FIGURE 6.2 Editing process by *k-Means* clustering (Prajapati et al. 2019)

In this research work, we considered two methods of finding the optimal value of clusters produced by *k-Means* clustering, the first method is the elbow method which computes the *Sum of Squared Error (SSE)* Eq. 6.1, which is the sum of the squared distance between each member of the cluster and its centroid (Milligan et al. 1985, Lee et al. 2013). The second method of cluster validation is to compute silhouette value (Rendón et al. 2011).

6.2.1

The concept of elbow method is to choose a number of clusters by adding clusters till much better modelling of the data does not achieve. It can be seen in Fig. 6.3., the elbow method starts with $k = 2$, then in each step, it keeps increasing by 1 with each increment. At some particular value of k, the percentage of variance (against the number of clusters) i.e. *SSE* drops significantly which indicates it has reached to a plateau and further increase in k does not give much advantage. This is the point where you get desired k values. If it increases the clusters beyond this point then the new cluster will be near to existing clusters.

$$SSE = \sum_{i=1}^{k} \sum_{x \in c_i} dist(x, c_i)^2 \tag{6.1}$$

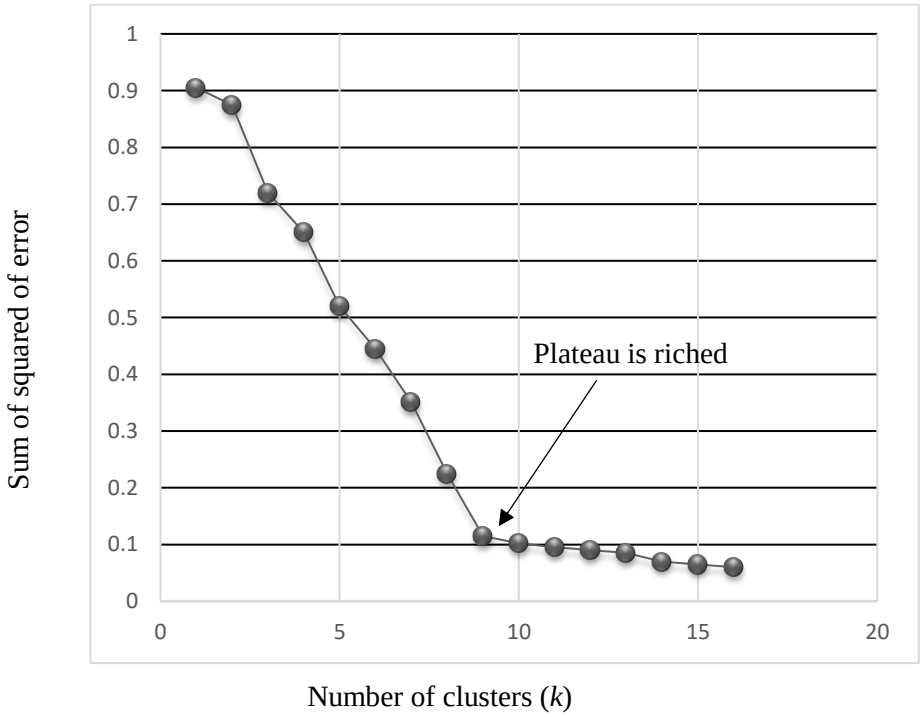


FIGURE 6.3 *SSE* vs. number of clusters (*k*) (Prajapati et al. 2019)

As shown in Fig 6.3, till point k = 9, it has reached the elbow point, means after k = 9 the percentage of variance is not reducing significantly. Hence, the recommended clusters for the data set by applying the elbow method is 9. For k-Means clustering, it is desirable to perform clustering with optimum k value to avoid finding patterns in noise (Milligan et al 1985). The *SSE* is defined as the sum of the squared distance between each member of the cluster and its centroid. It checks measures cohesion (Arbelaitz et al. 2013), which is how closely objects are related in a cluster and the process of finding optimal k by the elbow method is mentioned in algorithm 6.2.

ALGORITHM 6.2: Elbow method to find an optimum number of clusters (Prajapati et al. 2019)

Input: A set of Agriculture records $R = \{R_1, R_2 \ldots Rm\}$, where m is the total number of training records from *SHCD*.

Procedure:

Step 1: Initialize k = 1.

Step 2: Measure the value of *SSE*

Step 3: Increment $k = k+2$

Step 4: At some point, the percentage of variance of the solution reaches significantly, then store that value of k and stop. If not then repeat steps 2 to 4.

Output: Take the value of k.

6.2.2 Silhouette value

The *SSE* neither calculate the cluster consistency nor the well-separated distance between clusters. Hence, we have adopted silhouette value to calculate an optimum number of clusters for *k-Means* clustering algorithm. The silhouette value considers both the intra and inter-cluster distances (Starczewski et al. 2015).

Let μ be a space of the vectors with $x \in \mu$, $\{A_k\}_{k=1\ldots N}$ is a set of clusters, so that $\bigcup_k A_k = \mu$, C_k and C_o are centroids of A_k and μ respectively.

The distance of vector x_i to the cluster A_k is defined as Eq. 6.2,

$$a_k(x_i) = \frac{1}{|A_k|-1} \sum_{x_j \in A_k} \left(d(x_i, x_j)\right) \tag{6.2}$$

The distance from x_i to the nearest cluster is defined $x_i \in A_k$, Eq. 6.3,

$$b_k(x_i) = min_{j=1\ldots N}^{j \neq k} \{a_j(x_i)\} \tag{6.3}$$

The average of the distances from the centroid to the global center is defined as *InterMean,* Eq. 6.4,

$$InterMean = \frac{1}{N} \sum_{k=1}^{N} d(c_k, C_o) \tag{6.4}$$

The average of the distances between each point to its centroid in particular cluster is defined as *IntraMean*, Eq.6.5,

$$IntraMean = \frac{1}{|\mu|} \sum_{x \in \mu} d(x_i, C_k) \text{ with } x_i \text{ and } A_k \tag{6.5}$$

Silhouette value is defined as Eq. 6.6,

$$Silhouette = \frac{InterMean - IntraMean}{max\{InterMean - IntraMean\}\ with\ x_i \in A_x} \tag{6.6}$$

It is the interpretation and validation of consistency within cluster data, means it combines ideas of both cohesion and separation (Brun et al. 2007). Cluster separation measure how distinct or well separated a cluster is from other clusters. Silhouette value is typically between 0 and 1, and its value closest to 1 is considered better.

ALGORITHM 6.3: Silhouette value method to find an optimum number of clusters (Prajapati et al. 2019)

Input: A set of Agriculture records $R = \{R_1, R_2 \ldots Rm\}$, where m is the total number of training records from *SHCD*.

Procedure:

Step 1: Initialize $k = 1$.

Step 2: Measure *IntraMean*

Step 3: Measure *InterMean*

Step 4: if *InterMean* ≥ *IntraMean* then
max = *InterMean*
else
max = *IntraMean*

Step 5: Measure *Silhouette*

Step 6: if *Silhouette* < 1 then
Increment $k = k+2$
Repeat Step 2 to 6,
else
Stop

Output: Take the value of *k*.

Algorithm 6.3, calculates silhouette value to find an optimum number of clusters. The same concept is depicted in Fig. 6.4, the silhouette value reaches to one as a number of clusters increases. For *k*=19, silhouette value reaches to 1, means the optimum number of clusters for given data points is 19.

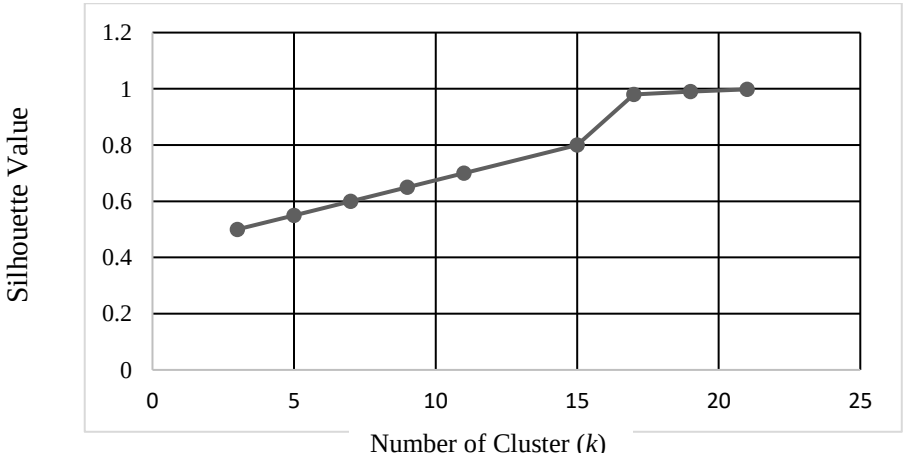


FIGURE 6.4 *Silhouette value* vs. number of clusters (*k*) (Prajapati et al. 2019)

In Section 6.2.1 and 6.2.2, we have discussed optimization methods to find the optimum number of clusters. These methods are used to generate *Edited Training Set (ETS) by* the *k-Means* clustering algorithm. This *ETS* is used for training in *k-NN* algorithm. In algorithm 6.4, step-2 is depicting application of elbow or silhouette value method to find *ETS*, which is less in a number of records than the original training set and helps to speed up the classification process.

ALGORITHM 6.4: *Fast k-Nearest Neighbour (F-kNN)* classifier (Elbow method / Silhouette value) (Prajapati et al. 2019)

Input: A set of Agriculture records $R = \{R_1, R_2, \ldots R_n\}$, where n is the total number of *SHCD* records.

Procedure:

Step 1: Divide the record data into one training set and one test set as 80-20 split.

Step 2: Run **elbow method** to find optimum clusters, take training set as input (algorithm 6.2) and generate *ETS* by applying *k-Means* Clustering.

OR

Run **silhouette value method** to find optimum clusters, take training set as input (algorithm 6.3) and generate *ETS* by applying *k-Means* Clustering.

Step 3: For each test record, calculate similarity with each training record from *ETS*.

Step 4: Sort the training records in the descending order of the maximum cosine similarity for each test record and select the top *k* training records.

Step 5: Assign a class to test record which occurs maximum times in the top *k* training records.

Step 6: Construct a confusion matrix.

Step 7: Calculate accuracy from confusion matrix, reduction rate of training records and classification time.

6.3 *Training Set Reduction k-Nearest Neighbor (TSR-kNN)* applied on *SHCD*

In the previous section, we have discussed the *F-kNN* algorithm which implements the concept of *PG* to reduces the number of training instances accelerate classification task by *k-NN* and. Now we'll explore *PS* approach to reduce the number of training instances for *k-NN* classification. The *PS* techniques do not generate prototypes of the training set by editing process, but it tries to extract the instances, which are superfluous by applying the instance selection process. The removal of a set of instances from the database will reduce the response time for classification query as few instances are examined.

As discussed in the Chapter 5, Section 5.2, the *SHCD* contains superfluous instances, which leads to large memory requirement and large query response time. To overcome this drawback we have proposed *Training Set Reduction k-Nearest Neighbour (TRS-kNN)* which uses state of the art shrink (subtractive) algorithm (Aha et al. 1991) to reduce the training set by removing superfluous records from *SHCD.*

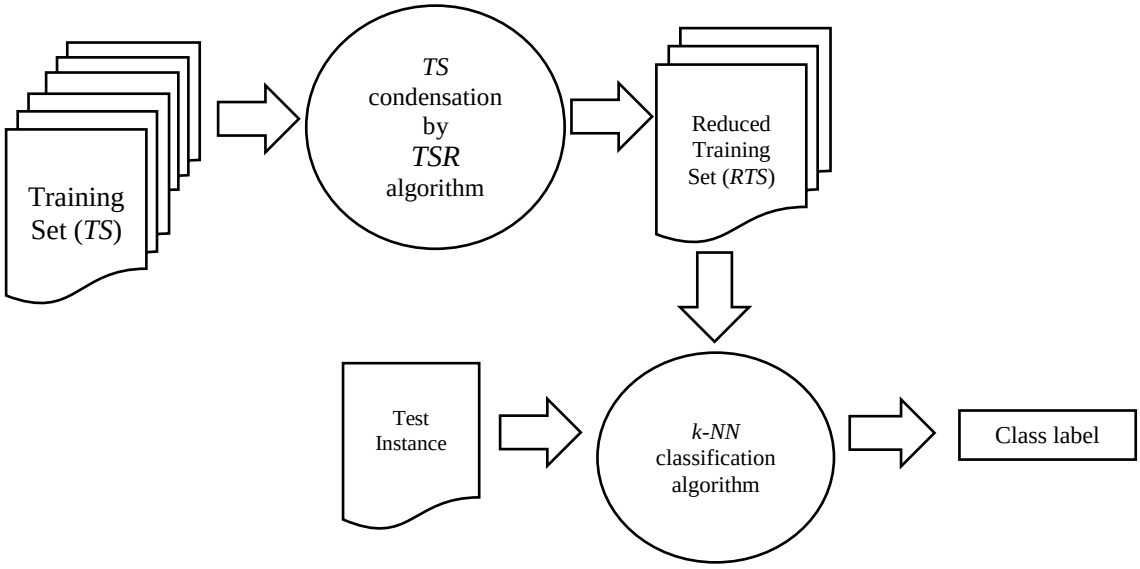


FIGURE 6.5 Overview of *Training Set Reduction k-NN (TSR-kNN)* (Prajapati et al. 2019)

Fig. 6.5 shows an overview of *TSR-kNN*. The training set is given to the *Training Set Reduction (TSR)* algorithm which removes superfluous instances and produces *Reduced Training Set (RTS)*. Further, the *k-NN* classifier uses *RTS* for the training phase and accordingly it classifies test instance. Algorithm 6.5, shows steps of *TRS* technique. The proposed *TRS-kNN* classifier has adopted *TRS* in step-2 of algorithm 6.6.

ALGORITHM 6.5: ***Training Set Reduction (TRS)*** **(Prajapati et al. 2019)**

Input: A set of Agriculture records $R = \{R_1, R_2 \ldots Rm\}$, where m is the total number of training records from *SHCD*.

Procedure:

Step 1: Assign all records of R into *Reduced Training Set (RTS)*.

Step 2: Select randomly an instance p from *RTS*.

Step 3: Classify the instance p using remaining instances from *RTS*.

Step 4: Remove the instance p if it is correctly classified.

Step 5: Repeat step 2 to 4 till no such instance left in *RTS*.

Output: Take the new reduced set *RTS* as a training set.

ALGORITHM 6.6: ***Training Set Reduction k-Nearest Neighbour (TRS-kNN)*** **classifier (Prajapati et al. 2019)**

Input: A set of Agriculture records $R = \{R_1, R_2, \ldots R_n\}$, where n is the total number of *SHCD* records.

Procedure:

Step 1: Divide the record data into one training set and one test set as 80-20 split.

Step 2: Run *TRS* algorithm, take training set and produce *Reduced Training Set (RTS)* (algorithm 6.4).

Step 3: Sort the training records from *RTS* in the descending order of the maximum cosine similarity for each test record and select the top k training records.

Step 4: Assign a class to test record which occurs maximum times in the top k training records.

Step 5: Construct a confusion matrix.

Step 6: Calculate accuracy from confusion matrix, reduction rate of training records and classification time.

6.4 *Training Set Reduction Fast k-Nearest Neighbor (TSR-FkNN)* applied on *SHCD*

In section 6.2 and 6.3, we have discussed training set editing and condensing techniques respectively. In this section, we have proposed a novel hybrid technique, which uses both editing and condensing. As shown in Fig. 6.6. It takes Initial *TS*, then it will condense it followed by editing. Finally, the edited set is taken as a training set for the *k*-NN classifier.

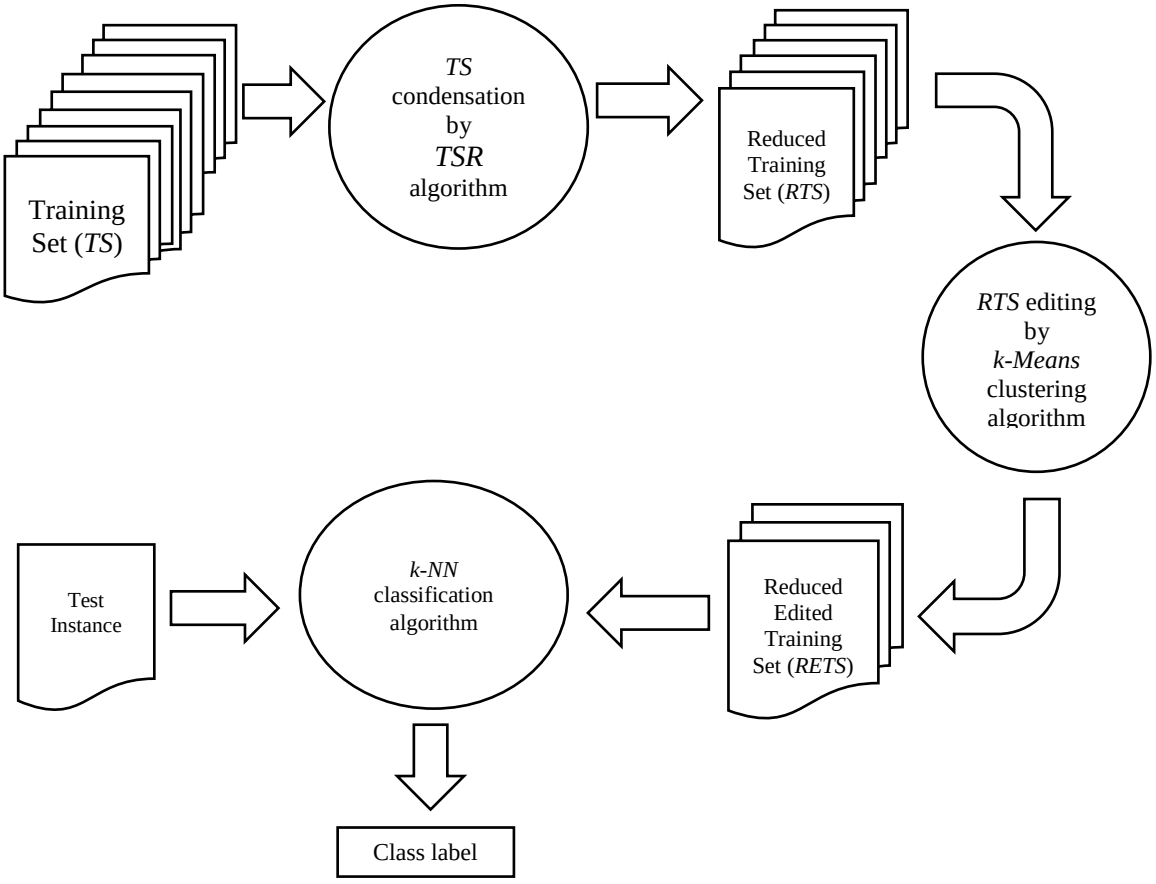


FIGURE 6.6 Overview of hybrid machine learning technique *TSR-FkNN* (Prajapati et al. 2019)

Fig. 6.6 provided an overview of a hybrid approach to generate a prototype of the training set for *k-NN* classifier. At first, *TS* from *SHCD* is taken and condensation is applied on it by *TSR* algorithm, which produces *RTS*. Then *RTS* becomes input to editing process, which adopts optimum *k-Means* clustering technique (Elbow method/ Silhouette value) and generates Reduced Edited Training Set (*RETS)*. Subsequently, *ETS* is used for the training of *k-NN* classification.

Algorithm 6.7 depicts the hybrid *TSR-FkNN* applied on *SHCD*. In step 2, the *TSR* algorithm runs and reduces the training set as *RTS*. Step-3 takes *RTS* as input and generates a set of cluster centroids called *RETS* which is further by *k-NN* classifier for training phase.

ALGORITHM 6.7: ***Training Set Reduction Fast k-Nearest Neighbour (TRS-FkNN)*** **classifier (Elbow method / Silhouette value) (Prajapati et al. 2019)**

Input: A set of Agriculture records $R = \{R_1, R_2, \ldots R_n\}$, where n is the total number of *SHCD* records.

Procedure:

Step 1: Divide the record data into one training set and one test set as 80-20 split.

Step 2: Run *TSR* algorithm, take training set and produce *Reduced Training Set (RTS)* (algorithm 6.4).

Step 3: Run **elbow method** to find optimum clusters, take *RTS* as input (algorithm 6.2) and generate *RETS* by applying *k-Means* Clustering.

OR

Run **silhouette value method** to find optimum clusters, take *RTS* as input (algorithm 6.3) and generate *RETS* by applying *k-Means* Clustering.

Step 4: For each test record, calculate similarity with each training record from *RETS*.

Step 5: Sort the training records in the descending order of the maximum cosine similarity for each test record and select the top *k* training records.

Step 6: Assign a class to test record which occurs maximum times in the top *k* training records.

Step 7: Construct a confusion matrix.

Step 8: Calculate accuracy from confusion matrix, reduction rate of training records and classification time.

6.5 Existing prototype generation algorithm applied on *SHCD*

Chang 1974, proposed an algorithm which checks the nearest two instances that have identical class and merge them as a single prototype. This process of merging is repeated untill the classification accuracy starts suffering. Algorithm 6.8 shows Chang's algorith is applied on *SHCD* to generate reduced set which can be used in the training phase of *k-NN* classification.

ALGORITHM 6.8: Chang's algorithm (Chang 1974)

Input: A set of Agriculture records $R = \{R_1, R_2, \ldots R_n\}$, where n is the total number of *SHCD* records.

Procedure:

Step 1: Divide the record data into one training set and one test set as 80-20 split.

Step 2: $A = \phi$, B = training set, find *old_accuracy* by *1-NN* rule

Step 3: A = random object from B

Step 4: $B = B - A$

Step 5: Find $p \in A$ and $q \in B$, and find minimum distance $d(p,q)$

Step 6: calculate $p^* = (p + q) / 2$,

Step 7: $B = B - \{ p , q \}$

$A = A \cup \{ p^* \} - \{ p , q \}$

Step 8: *new_accuracy* = find accuracy by *1-NN* rule (take B as training set)

Step 9: if *new_accuracy* $\leq$ *old_accuracy* then

Edited Training Set = A

stop

else

goto step 3

Output: *Edited Training Set (ETS)*

The Chang's algorithm is *PG* method to generate *ETS* from *TS* which can be used by training phase of *k-NN* classifier. When *k-NN* classifier uses *ETS* which is having less number of instances than *TS*, then we can have the advantage of speedy classification process, hence the Chang's algorithm based *k-NN* classifier's name is given as *Fast k-Nearest Neighbour* classifier

(*F-kNN (Chang's algorithm)*) which is shown in algorithm 6.9. The step-2 of algorithm 5.6 takes training set from *SHCD* and produces *ETS*, which is used by *k-NN* for training.

ALGORITHM 6.9: *Fast k-Nearest Neighbour (F-kNN)* classifier (Chang's algorithm)

Input: A set of Agriculture records $R = \{R_1, R_2, \ldots R_n\}$, where n is the total number of *SHCD* records.

Procedure:

Step 1: Divide the record data into one training set and one test set as 80-20 split.

Step 2: Run Chang's algorithm, take training set and produce *Edited Training Set (ETS)* (algorithm 6.7).

Step 4: For each test record, calculate similarity with each training record from *ETS*.

Step 5: Sort the training records in the descending order of the maximum cosine similarity for each test record and select the top *k* training records.

Step 6: Assign a class to test record, which occurs maximum times in the top *k* training records.

Step 7: Construct a confusion matrix.

Step 8: Calculate accuracy from confusion matrix, reduction rate of training records and classification time.

CHAPTER - 7

Empirical Results and Analysis

In the previous chapter, we have discussed implementation of various *Prototype Generation (PG)* and *Prototype Selection (PS)* based classification algorithms; *k-NN, F-kNN, TSR-kNN, TSR-FkNN* and *F-kNN (Chang's algorithm)* to classify soil samples of *Soil Health Card Database (SHCD)* into macro and micro nutrients deficiencies.

The *k-NN* algorithm is an existing classification technique discussed in standard *Machine Learning (ML)* literature, where all training set instances are utilized in the training phase of *k-NN* classifier. The *TSR-kNN* algorithm is based on existing *PS* scheme called shrink subtractive technique for training set reduction (Aha et al. 1991), and *F-kNN (Chang's algorithm)* is based on existing *PG* method to produce prototype set by editing training set (Chang et al. 1974). The proposed *PG* based *F-kNN* algorithms are generating prototype sets from training set by editing it, and the proposed hybrid method based *TSR-FkNN* algorithm is constructing a prototype set by reduction and edition of the training set.

In this chapter, comparisons between the above existing and proposed classifier are carried out in terms of accuracy (%), training set reduction rate (%) and classification time (milliseconds). For the experiment, we have taken Kutch district data set from *SHCD,* having total 14000 entries. Further, these results are performed on a computer with Intel i5 processor and 4GB Ram, the software IDE is NetBeans 8.2. Depends on hardware some of the results may vary. The observed results are on average of five times run.

7.1 Dataset normalization and evaluation measures

7.1.1 Dataset normalization

The attributes in the *SHCD* data set have different ranges. For example, the minimum and maximum values for attribute *SHC_POTASS* are 100 and 854 accordingly while the range of values for *SHC_IRON* is from 0.2 to 2. The differences in ranges of attributes would lead to a scenario where attributes having greater value in range will have an undesirable influence on classification efficiency. Distance measures would be highly affected by attributes having larger values compared to attributes having smaller values. Hence, it is necessary to normalize the numerical variables to standardize the scale of the effect of each variable on the results (Larose et al. 2014).

As discussed in (Chapter 5: Section 5.1), the *SHCD* contains eight different attributes and these attributes have a range of nonnegative values. We applied *min-max* normalization to normalize the value of an attribute in the dataset. The *min-max* technique maps the attribute value in [0, 1] range by applying Eq. 7.1 (Pandey et al. 2017).

As per Eq. 7.1 *min-max* normalization technique,

$$A' = \left(\frac{A - \min\_value\_of\_A}{\max\_value\_of\_A - \min\_value\_of\_A} \right) * (D - C) + C \qquad (7.1)$$

Where, *A'* contains normalized data, *A* is original data and *[C, D]* is a predefined boundary. For example, for one record from *SHCDS*, *SHC_POTASS*=274 means for above equation *A*=274, *min_value_of_A*=100, *max_value_of_A*=854, *C*=0 and *D*=1. If we put these values in Eq. 7.1, we get normalizes value *A'*=0.23.

	A	B	C	D	E	F	G	H
1	SHC_POTA	SHC_SULP	SHC_MG	SHC_PHOS	SHC_IRON	SHC_MAN	SHC_ZINC	SHC_CU
2	146	2.5	2.6	26	0.45	7	3.5	7.9
3	274	6.6	4.6	23	0.25	1	1.5	13
4	194	3.5	4.5	66	0.25	0.65	3	7.9
5	392	5.2	6.5	11	0.42	0.45	2.5	8.7
6	124	5.2	5.5	21	0.52	4.5	3.5	14.5
7	110	3.6	7	57	0.36	6.3	3.5	13.5

FIGURE 7.1 Sample from *SHCDS* without normalization (Soilhealth 2017)

Fig. 7.1 shows a sample records taken from *SHCD*, here values of all attributes of soil samples can be seen having values in different range. After min-max normalization the resultant records having values between range (0-1) for all attributes is show in Fig 7.2.

	A	B	C	D	E	F	G	H
1	SHC_POTA	SHC_SULP	SHC_MG	SHC_PHOS	SHC_IRON	SHC_MAN	SHC_ZINC	SHC_CU
2	0.06	0.01	0.09	0.18	0.14	0.19	0.36	0.50
3	0.23	0.10	0.40	0.15	0.03	0.02	0.07	0.84
4	0.12	0.03	0.38	0.64	0.03	0.01	0.29	0.50
5	0.39	0.07	0.69	0.01	0.12	0.01	0.21	0.56
6	0.03	0.07	0.54	0.13	0.18	0.12	0.36	0.94
7	0.01	0.04	0.77	0.53	0.09	0.17	0.36	0.87

FIGURE 7.2 Sample from *SHCD* after applying *min-max* normalization (Soilhealth 2017)

7.1.2 Evaluation measures

There is a broad range of measures available in *ML* literature; for proposed and existing classifiers we are interested in three quantitative measures: *Accuracy, Training Set Reduction Rate (TSRR) and Classification Time (CT).*

***Accuracy*:** It simply calculates how often the classifier marks the precise prediction. It represents the ratio between a number of predictions those are correctly classified to the total number of predictions (Hossin et al. 2015). For multiclass classification, the confusion matrix is built to compare predicted classes to actual classes. If we have *N* classes, then the confusion matrix would be *N x N* as shown in Fig. 7.3.

$$C = \begin{array}{c} \\ Actual \end{array} \begin{array}{c} \quad\quad\quad\quad Classified \\ \begin{array}{ccccc} C_{11} & \ldots & \ldots & \ldots & C1n \\ . & \ldots & \ldots & \ldots & . \\ . & \ldots & \ldots & \ldots & . \\ Cn1 & \ldots & \ldots & \ldots & Cnn \end{array} \end{array}$$

FIGURE 7.3 Confusion matrix for multiclass classifier (Hossin et al. 2015)

Each raw of the matrix represents the actual class, while each column presents the results of prediction for the corresponding class of that column. The correct classification predictions are available in the diagonal cells and misclassified cells are presented at the off diagonal cells.

The confusion elements for each class are given by:

tp_i = the number of records correctly assigned to the class

tn_i = the number of records correctly rejected assigned to the class

fp_i = the number of records incorrectly rejected assigned to the class

fn_i = the number of records incorrectly assigned to the class

The accuracy for multiclass classifier is derived as Eq. 7.2

$$Accuracy = \sum_{i=1}^{n} \frac{tp_i + tn_i}{tp_i + fp_i + tn_i + fn_i} \tag{7.2}$$

***Training Set Reduction Rate (TSRR)*:** The proposed research work's prime aim is to reduce training set instances for *k-NN* classification algorithm. The projected and existing classification methods take *TS* and then applies *PS* and/or *PG* scheme to reduce *TS,* so that the effectiveness of the classifier maintains or improved. *TSRR* (Eq. 7.3) is proposed to check and compare the effectiveness of *PS* and *PG* method in terms of reduction of *Training Set (TS).*

N_I = Total number of instances in training set

RN_I = Total number of instances in the reduced set

$$TSRR = \frac{N_I - RN_I}{N_I} \tag{7.3}$$

***Classification Time (CT)*:** As training set size is reduced, the computation to perform the classification time reduced significantly. To measure this performance gain in terms of classification speed, we proposed to measure classification time in milliseconds.

7.2 Empirical results

7.2.1 Reduction in training set size

As discussed in Chapter 6, the *k-NN* (Chapter 6: Section 6.1) algorithm does not uses any *PG/PS* technique, while *F-kNN* (Chapter 6: Section 6.2), *TSR-kNN* (Chapter 6: Section 6.3), *TSR-FkNN* (Chapter 6: Section 6.4) and *F-kNN (Chang's algorithm)* (Chapter 6: Section 6.5) uses *PG and/or PS* technique to reduce instances in *TS*. The *F-kNN* uses *PG* methods *elbow method/silhouette value/Chang's algorithm* to generate *Edited Training Set (ETS)* from *TS*. Existing shrink subtractive method (Aha et al. 1991) is applied on *TS* to produce *Reduced Training Set (RTS)* in *TSR-kNN*. The *TSR-FkNN* uses a hybrid scheme which combines both *PS* and *PG* techniques to generate *Reduced Edited Training Set (RETS)*.

As discussed in Chapter 6: Section 6.1, the step-1 (Algorithm 6.1) of the *k-NN* classifier takes half of total 14000 instances in the training phase (*TS* size is 7000). The algorithms *F-kNN*, *TSR-kNN* and *TSR-FkNN* takes training and test set in ration 80:20 respectively (Chapter 6: Section 6.2, 6.3. 6.4 and 6.5). Hence for all these techniques, the number of instances in *TS* is 11,200.

TABLE 7.1 *TSRR* of all classifiers

1	Type of Prototype set	*TS*	*ETS (PG)*			*RTS (PS)*	*RETS (Hybrid)*	
2	Algorithm	*k-NN*	*F-kNN* (Elbow method)	*F-kNN* (Silhouette value)	*F-kNN* (Chang's algorithm)	*TSR-kNN*	TSR-FkNN (Elbow Method)	TSR-FkNN (Silhouette value)
3	Number of Instances in *TS*	7000	11200	11200	11200	11200	11200	11200
4	Number of Instances in Prototype set	7000	521	663	4537	4849	361	441
5	*TSRR* (%)	0	95.34	94.08	59.59	56.70	**96.77**	<u>96.06</u>

(Note: The above observed results are average of five times run; bold numbers indicate highest reduction rate while underlined number indicate the second highest)

Table 7.1 shows a comparison among different classifiers' effectiveness in reducing *TS*. It is observed that *k-NN* classifier does not apply any *PG/PS* technique on the training set, hence for *k-NN* the *TSRR* = 0. *Chang's algorithm's* and *TSR-kNN's* is having lowest (59.59%) and second lowest (56.70%) *TSRR* respectively amongst implemented *PG/PS* techniques. *TSR-FkNN (Elbow method)* and *TSR-FkNN (Silhouette value)* classifiers' have highest (96.77%) and second

highest (96.06%) *TSRR* respectively, while *TSRR* of *F-kNN (Elbow method)* and *F-kNN (Silhouette value)* algorithms' is 95.34 % and 94.08% respectively. The observation of table 7.1 is depicted in Fig. 7.4 (*k-NN's TSRR* is zero).

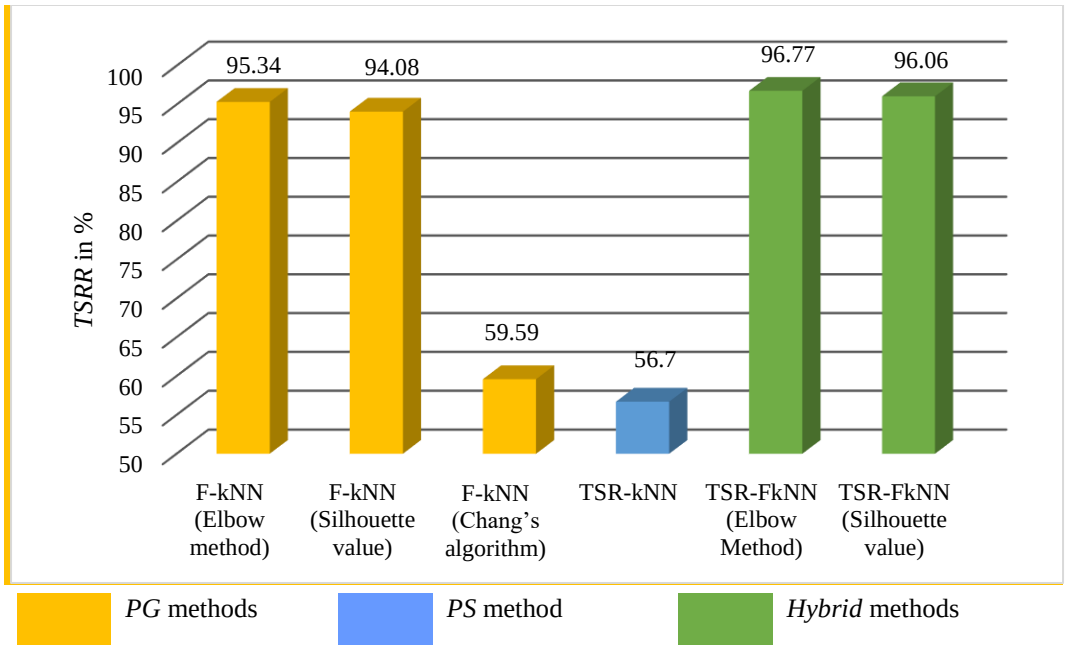


FIGURE 7.4 *TSRR* of *PG/PS* classifiers

7.2.2 Accuracy of various classifiers

The prime aim of this research is to efficiently classify soil samples from *SHCD* into particular macro and micro deficiencies categories. For that, we have adopted the *k-NN* classifier and implemented *PG* and/or *PS* techniques to reduce the training size of *k-NN*. For the results, all proposed and existing algorithms discussed in Chapter 6 are implemented to classify soil samples of *SHCD*. Except for *k-NN*, other algorithms have step to reduce the size of *TS*. The generated prototype set is being used in the training phase of *k-NN* classification, hence accuracy is measured to check whether the proposed and existing *PG/PS* techniques affects the performance of *k-NN* classifier.

The accuracy of existing and proposed classifiers is measured and shown in table 7.2. For *k-NN* classification, *k* values are takes from 31 to 45. Remember, except *k-NN* classifier, all other techniques are using a prototype training set. Table 7.2 is depicted in Fig. 7.5 (average accuracy of table 7.2 is not mapped).

TABLE 7.2 Accuracy of all classifiers

	Value of *k* for *k*-NN	Classifiers accuracy in %						
Sr. No	*k*	*k-NN*	*F-kNN* (Elbow method)	*F-kNN* (Silhouette value)	*F-kNN* (Chang's algorithm)	*TSR-kNN*	*TSR-FkNN (Elbow method)*	*TSR-FkNN (Silhouette value)*
1	31	88.85	87.76	88.71	87.23	88.21	89.28	92.14
2	33	90	87.92	89.28	87.68	88.75	90.35	91.75
3	35	90.21	88.78	89.78	87.78	89.07	90.42	91.85
4	37	90.27	89.23	90.35	87.84	90.15	90.75	92.35
5	39	90.35	88.85	90.54	88	89.71	<u>90.85</u>	92.64
6	41	90.55	88.64	90.45	88.56	89.57	90.62	**93.85**
7	43	89.51	88.46	89.91	88.34	89.35	90.47	93.34
8	45	89.21	88.45	89.88	87.93	89.25	90.23	92.46
Average accuracy		89.86	88.51	89.86	87.92	89.25	90.37	92.54

(Note: The above observed results are average of five times run; bold numbers indicate the highest accuracy while underlined number indicate the second highest)

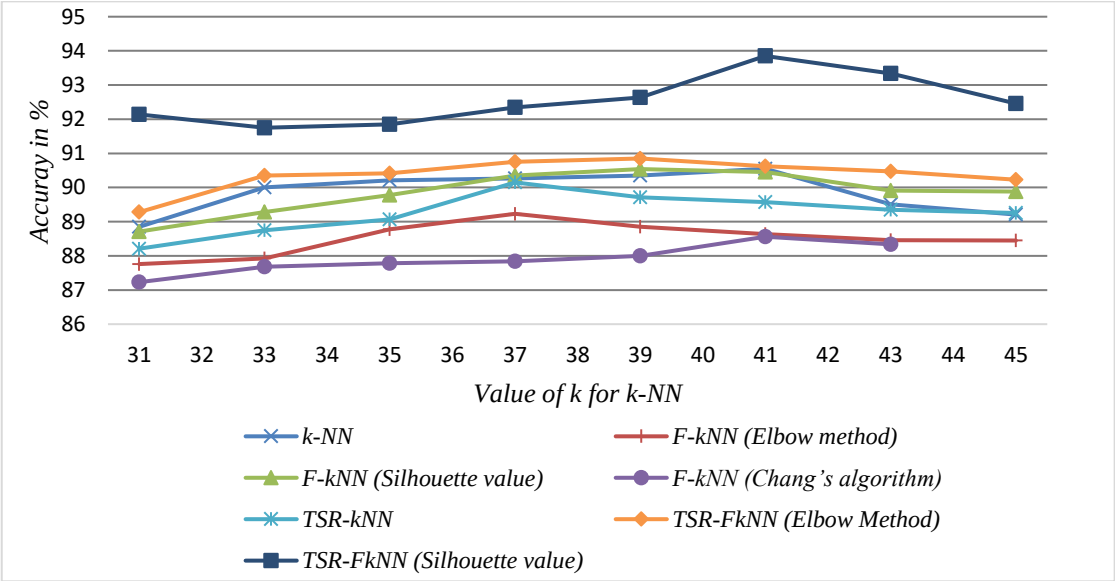


FIGURE 7.5 Accuracy of different classifiers

The highest measured accuracy is 93.85% for proposed *TSR- FkNN (Silhouette value)* classifier when k = 41. Second highest accuracy 90.85% is measured for proposed *TRS-FkNN (Elbow method)* when k = 39. For all classifiers' accuracy increases with increase in value of k, then after certain level it decreases even if value of k increases due to large value of k lead to over simplified decision boundaries. It implies, with large value of k the k-NN classifier tend to

misclassifies instances at decision boundaries, which lead to decrease in accuracy. The average accuracy is also calculated, which is highest 92.54 % for *TSR-FkNN (Silhouette value)* classifier. The second highest average accuracy is 90.37% for *TSR-FkNN (Elbow method)* classifier.

7.2.3 Classification time of various classifiers

In *k-NN* classifier's testing phase, calculation of similarity between each test record is carried out with all training record. These similarities for each test records are shorted and then top *k* similarities and their corresponding class labels are going through the voting process, hence the majority vote of a particular class is assigned as a label to test instance. The proposed *PG/PS* schemes are reducing the size of *TS*, which is resulted in less classification time as a number of training instances will be fewer compared original *TS*.

TABLE 7.3 Classification time for all classifiers

	Value of *k* for *k*-NN	Classification time (milliseconds)						
Sr. No	*k*	*k-NN (TS= 7000)*	*F-kNN* (Elbow method) *(ETS=521)*	*F-kNN* (Silhouette value) *(ETS=663)*	*F-kNN* (Chang's algorithm) *(ETS=4537)*	*TSR-kNN (RTS= 4849)*	*TSR-FkNN (Elbow method) (RETS=361)*	*TSR-FkNN (Silhouette value) (RETS=441)*
1	31	5745	219	235	3452	3670	**143**	192
2	33	5768	226	248	3487	3713	157	213
3	35	5798	241	255	3524	3745	175	217
4	37	5824	257	271	3556	3782	180	219
5	39	5855	267	285	3595	3815	185	224
6	41	5893	281	297	3634	3842	217	235
7	43	5912	292	306	3678	3876	221	245
8	45	5950	302	317	3724	3904	248	261
Average Classification time in milliseconds		5843.13	260.63	276.75	3581.25	3793.38	190.75	225.75

(Note: The above observed results are average of five times run; bold numbers indicate lowest classification time. Values of *TS*, *ETS*, *RTS* and *RETS* is referred from table 6.1)

The observed *classification time* for each classifier is recorded in table 7.3. The simple *k-NN* classifier is utilizing 7000 training and 7000 test records. For other classifiers, total 14000 records of Kutch district is divided into 80:20 ratio for training set and test set respectively.

Hence, classifiers (except *k-NN*) uses 2800 (20% of 1400) records for testing, and for training they are used respectively generated prototype set i.e. *ETS, RTS, RETS* (shown in the bracket with classifiers' name).

The lowest classification time 143 milliseconds is observed for *TRS-FkNN* classifier when value of $k = 31$. For all classifiers, as we increase the value of k, the classification time increases. The results of table 7.3 is presented graphically in Fig 7.6 (average classification time is not mapped)

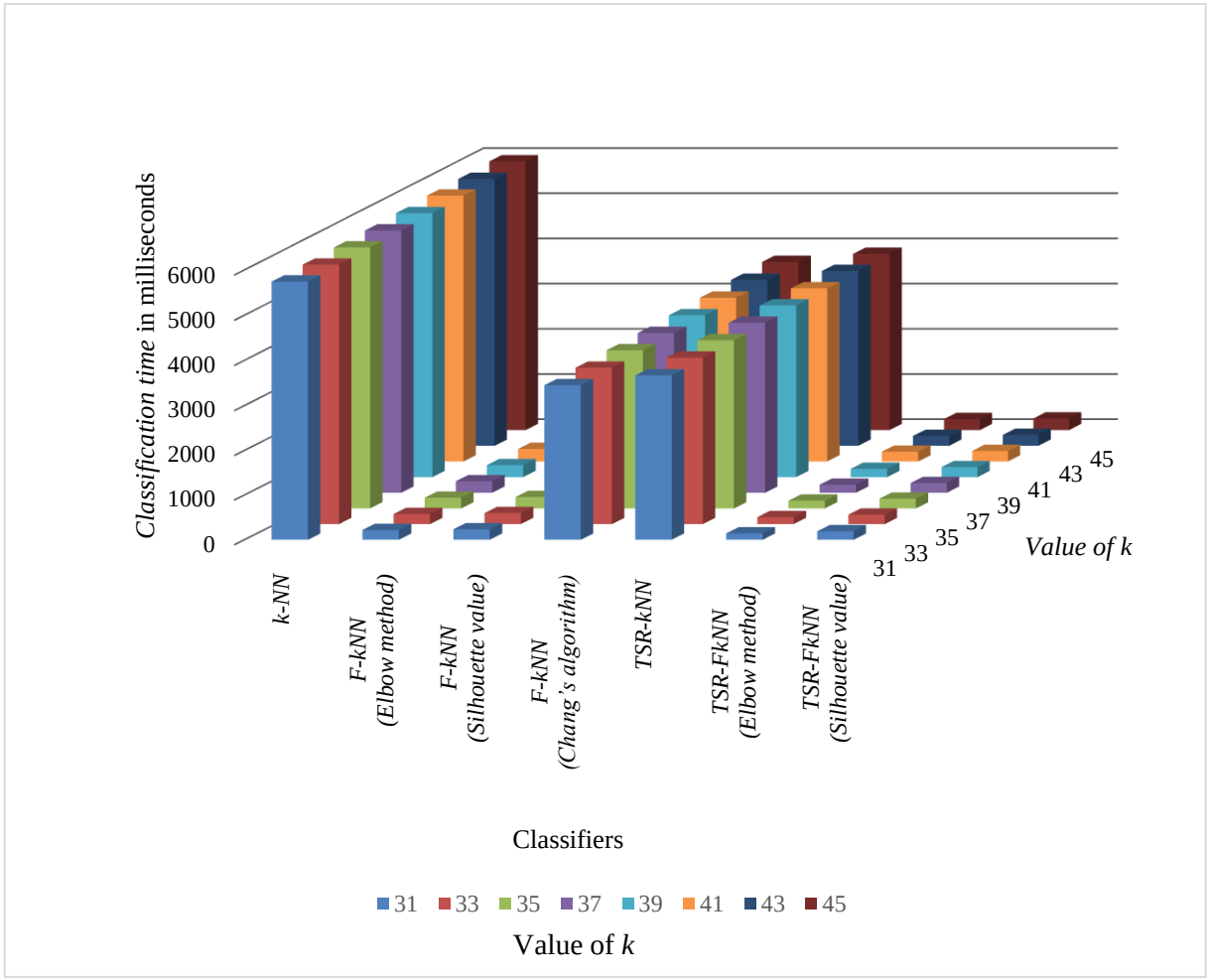


FIGURE 7.6 Classification time of different classifiers

7.3 Results analysis

7.3.1 *TSRR* vs. Accuracy

Each proposed and existing classifiers (except *k-NN*) of this research work have adapted *PG/PS* technique, which reduces the number of training instances significantly. Hence, it is anticipated to check whether there is an impact of the reduction in training instances on classification accuracy or not. To evaluate the impact of the reduction in training set size on accuracy, the measured values of *TSRR* and average accuracy from table 7.1 and 7.2 respectively are taken into table 7.4. The graphical representation of table 7.4 is shown in Fig 7.7.

TABLE 7.4 Comparison of *TSRR* and average accuracy

Sr. No	Classifier's evaluation measure	Classifiers						
		k-NN (TS= 7000)	*F-kNN (Elbow method) (ETS=521)*	*F-kNN (Silhouette value) (ETS=663)*	*F-kNN (Chang's algorithm) (ETS=4537)*	*TSR-kNN (RTS= 4849)*	*TSR-FkNN (Elbow method) (RETS=361)*	*TSR-FkNN (Silhouette value) (RETS=441)*
1	*TSRR* (%)	0	95.34	94.08	59.59	56.70	**96.77**	<u>96.06</u>
2	*Average accuracy*	89.86	88.51	89.86	87.92	89.25	<u>90.37</u>	**92.54**

(Note: Bold numbers indicate the highest value, while underline number indicate second highest value)

The observed average accuracy of *k-NN* is 89.86%, while it does not have any *TS* reduction phase, means it uses 7000 (50% of total 14000 instances of *SHCD*, indicated in the bracket with classifier name) training instances. As discussed in Chapter 5: Section 5.2, the 7000 training instances of *SHCD* contains noisy and superfluous records, it implies that the accuracy 89.86 % of *k-NN* is computed in existence of such noisy and superfluous instances.

The *F-kNN (Elbow method)*, *F-kNN (Silhouette value)* and *F-kNN (Chang's algorithm)* begin with 11,200 (80% of 14000 *SHCD* records) training set instances, which encloses many noisy instances (Chapter 5: Section 5.2). To deal with these noisy instances we have adapted *PG* method called *elbow method* (Chapter 6: Section 6.2), *silhouette value* (Chapter 6: Section 6.2) and *Chang's algorithm* (Chapter 6: Section 6.5) to generate prototype from *TS*. The proposed *elbow method* achieves *TSRR* of 95.34 %, means the *ETS* retains 521 training instances from initial 11,200 training instances. The average accuracy of *F-kNN (Elbow method)* is 88.51%,

which is less than *k-NN*. It implies that the elbow method eliminates not only noisy instances but also some useful instances, which resulted in a minor decrease in accuracy than *k-NN*.

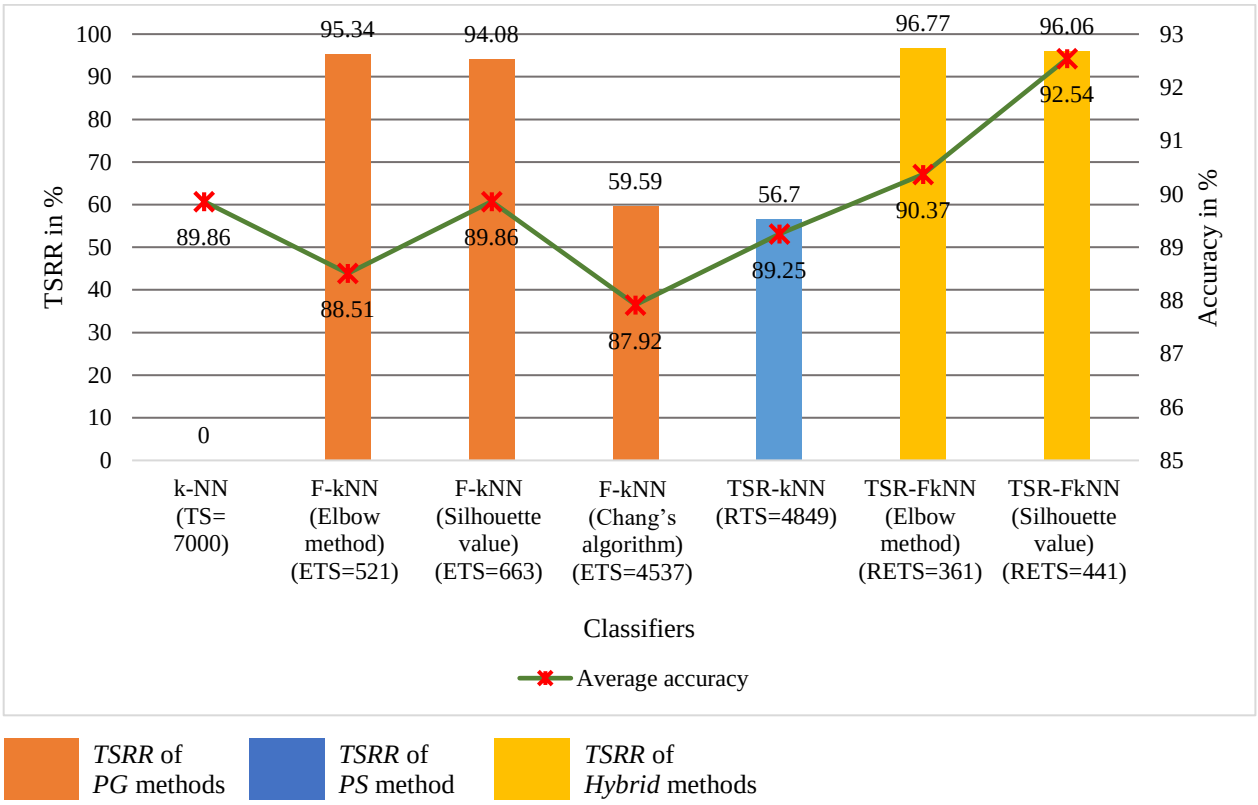


FIGURE 7.7 *TSRR* vs. *Accuracy*

The observed *TSRR* for *F-kNN (Silhouette value)* is 94.08 %, means it retains only 663 training instance in *ETS* out of 11,200 instances in *TS*. The average accuracy of *F-kNN (Silhouette value)* is 89.86%, which is more than *F-kNN (Elbow method)* and equal to *k-NN*. It implies *F-kNN (Silhouette value)* eliminates noisy instances and generated prototype set, which is an effective alternative to *TS* with less training instances and high accuracy than *F-kNN (Elbow method)*.

The *F-kNN (Chang's algorithm)* have worst *TSRR* of 59.59 % among the implemented *PG* based classifiers. The *ETS* generated by applying Chang's algorithm consists of 4537 instances, which quite large than *ETS* of *F-kNN (Elbow method)* and *F-kNN (Silhouette value)*. The average accuracy of *F-kNN (Chang's algorithm)* is 87.92, which less than *k-NN*, *F-kNN (Elbow method)* and *F-kNN (Silhouette value)*. It conveys that to generate prototype by applying Chang's algorithm is not that worthy as it retains a high number of unwanted instances, which lead to low accuracy of classification.

As discussed in Chapter 5, Section 5.2, *SHCD* record contains superfluous instances. To eliminate such similar records in *SHCD*, we have implemented *PS* method called *TSR-kNN*. It is observed that *TSRR* of *TRS-kNN* is 56.70%, means it retains 4849 training instances in *RTS* out of 11,200 training instances. The observed average accuracy for *TSR-kNN* is 89.25%, which is higher that *F-kNN (Chang's algorithm)*, *F-kNN (Elbow method)* and slightly lower *k-NN* and *F-kNN (Silhouette value)*. It signifies that, *TSR-kNN* effective *PS* based technique to eliminate superfluous instances from training set, but it is not effective in removing noisy instances.

The proposed *TSR-FkNN (Elbow method)* and *TSR-FkNN (Silhouette value)* are hybrid methods. These two methods are having benefit of both noisy and superfluous instances removal as both of them have employed *PG* and *PS* technique to generate prototype from training set.

The *TSRR* for *TSR-FkNN (Elbow method)* is highest 96.77% among all classifiers, while its average accuracy is second highest 90.37%. The *TSR-FkNN (Elbow method)* retains 361 instances in *RETS* out of 11,200 training instances by applying training set reduction technique (*PS* technique) to remove superfluous instances followed by elbow method (*PG* technique) to remove noisy instances. The highest reduction rate and second highest average accuracy of *TSR-FkNN (Elbow method)* points to its effectiveness of representing prototype training set that contains noise free and dissimilar training instances.

The proposed *TSR-FkNN (Silhouette value)* classifier is hybrid method in terms of generating prototype form training set. It has highest 92.54% average accuracy, while its *TSRR* is second highest 96.06%. *TSR-FkNN (Silhouette value)* is able to generate 441 prototype training instances in *RETS* by applying training set reduction followed by silhouette value method to edit reduced set. The highest average accuracy of *TSR-FkNN (Silhouette value)* denotes that it is best classifier in spite of its *TSRR* is slightly less than *TSR-FkNN (Elbow method)*. The *TSR-FkNN (Silhouette value)* classifier effectively removes superfluous instances and overcome the problem of noisy instances by training set editing.

7.3.2 *TSRR* vs. Classification time

As discussed in Chapter 6: Section 6.1, every time the *k-NN* classifier utilizes all *TS* instances in computation while assigning a class label to every test instance. It indicates that if there is number of training instances then it will affect the classification speed. The proposed and existing classifiers in this research work are tending to reduce the number of training instances (except *k-NN* algorithm) by use of proposed and existing *PS/PG* techniques. Hence, it is desirable to compare *TSRR* and classification time of various classifiers. The *TSRR* and average classification time of different classifiers are derived into table 7.5 from 7.1 and 7.3. The statistics of table 7.5 is presented in a graphical format is depicted in Fig. 7.8.

TABLE 7.5 Comparison of *TSRR* and average classification time

Sr. No	Classifier's evaluation measure	Classifiers						
		k-NN (TS= 7000)	*F-kNN (Elbow method) (ETS=521)*	*F-kNN (Silhouette value) (ETS=663)*	*F-kNN (Chang's algorithm) (ETS= 4537)*	*TSR-kNN (RTS= 4849)*	*TSR-FkNN (Elbow method) (RETS=361)*	*TSR-FkNN (Silhouette value) (RETS=441)*
1	*TSRR* (%)	0	95.34	94.08	59.59	56.70	**96.77**	<u>96.06</u>
2	*Average classification time* in milliseconds	5843.13	260.63	276.75	3581.25	3793.38	<u>190.75</u>	**225.75**

(Note: Bold numbers indicate highest value, while underline number indicate second highest value)

The *k-NN* classifiers does not apply any reduction technique on *TS*, means it utilizes all 7000 training instances to test 7000 test instances (training and test sets are divided into 50-50 percent ratio). Naturally, *k-NN* is performing computationally exhaustive task to assign class label to each test instance. Hence, the average classification time for *k-NN* is 5843.13 milliseconds, which is highest average classification time observed among all classifiers.

The proposed *F-kNN (Elbow method)* and *F-kNN (Silhouette value)* are generating ETS of 521 and 663 instances respectively from *TS* with 11,200 instances. For both classifiers, the test set is consist of 2800 instances (20% of 14000 total instances). The average classification time of *F-kNN (Elbow method)* is 260.63 milliseconds which is third lowest amongst all classifiers due to less number of instances in *ETS*. Similarly, *F-kNN (Silhouette value)* has fourth lowest

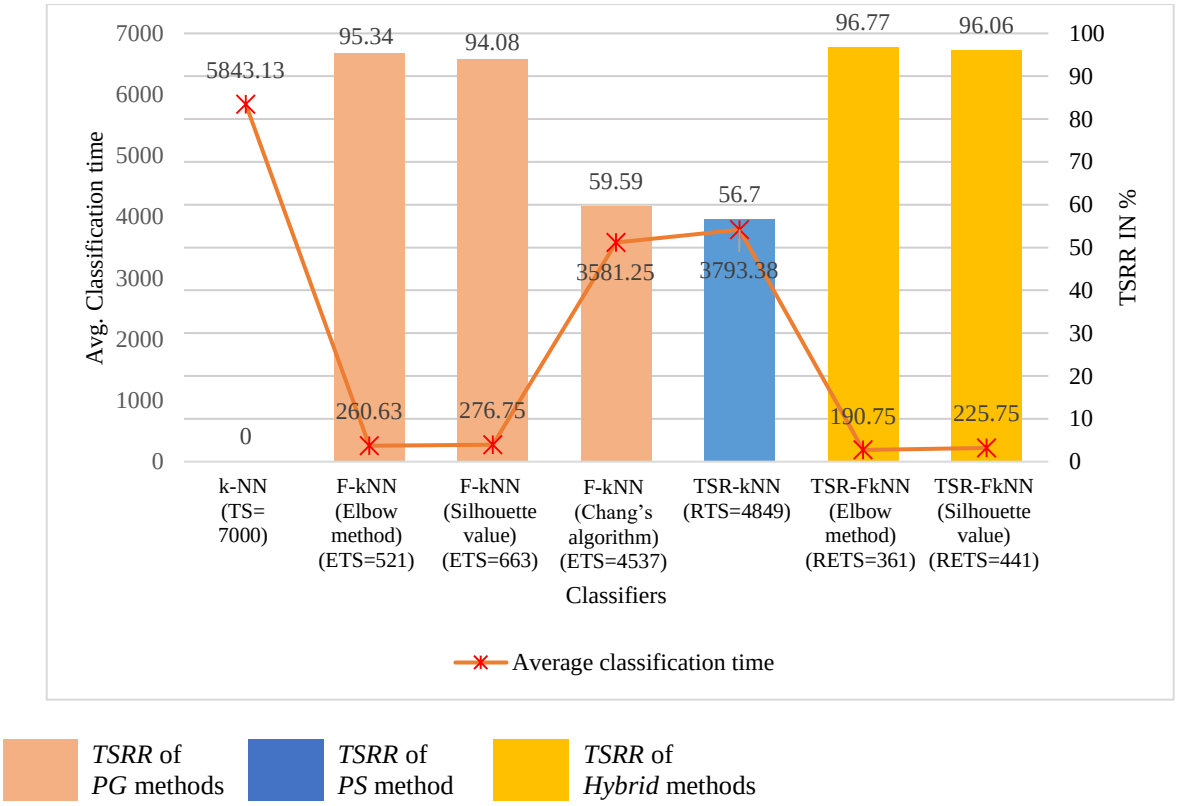


FIGURE 7.8 *TSRR* vs. Classification time

average classification time of 276.75 milliseconds, which is higher than the average classification time of *F-kNN (Elbow method)* due to *ETS* of *F-kNN (Silhouette value)* contains more instances than *F-kNN (Silhouette value)*. For both *F-kNN (Elbow method)* and *F-kNN (Silhouette value)* classifiers, the computational cost of the classification process is not that intensive due to small *ETS* size. Consequently, average classification time is quite low compared to *k-NN*, *F-kNN (Chang's algorithm)* and *TSR-kNN*.

The *F-kNN (Chang's algorithm)* has implemented an existing prototype generation scheme, which generates ETS of 4537 instances from *TS* with 11,200 instances. The classification step in *F-kNN (Chang's algorithm)* utilizes the *ETS* and test set with 2800 instances (20% of 14000 instances). The observed average classification time of *F-kNN (Chang's algorithm)* is 3581.25 milliseconds which is third highest among all classifiers' average classification time. In *F-kNN (Chang's algorithm)*, a number of instances in *ETS* is quite high than other algorithms (except *k-NN* and *TSR-FkNN*), which resulted in the computationally exhaustive classification task.

In *TSR-kNN*, existing training set reduction technique is implemented (Aha et al. 1991), which produces the *RTS* with 4849 instances out of 11,200 instances in *TS*. With this *RTS*, the *TSR-kNN* accomplishes the classification of 2800 test instances (20% of 14000 instances). As a number of instances in *RTS* is quite high, which resulted in a high computational cost of the classification process. Hence, the average classification time of *TSR-kNN* is second highest amongst all classifiers.

The *TSR-FkNN (Elbow method)* is a proposed hybrid method, which generates 361 instances in *RETS* from 11,200 TS instances. For the test phase, *TSR-FkNN (Elbow method)* uses 2800 (20% of 14000) instance. As a number of generated prototype instances in *RETS* is lowest compared to other algorithms, the average classification time of *TSR-FkNN* is 190.75 milliseconds which is lowest amongst all classifiers.

In proposed hybrid *TSR-FkNN (Silhouette value)* method, *RETS* with 441 instances is the generated prototype set from 11,200 *TS*. The classification task of *TSR-FkNN* utilizes *RETS* and test set with 2800 (20% of 14000 instances) instances. The observed average classification time of *TSR-FkNN (Silhouette value)* is 225.75 milliseconds which is second lowest due to low computational requirement due to less number of instances in *RETS*.

CHAPTER – 8

Conclusion and Future Scope

8.1 Conclusion

The proposed research work titled "Knowledge discovery and data mining in the agricultural database using machine learning techniques" is targeted at applying machine learning technique called k-NN classification on agricultural *Soil Health Card Database (SHCD)* to classify soil records of *SHCD* into categories of macro-micro nutrients deficiencies. The *k-NN* classifier is having demerits of high storage requirement and computational cost. Moreover, for an experiment, we have taken a dataset of Kutch district from *SHCD*, which contains 14000 records consists of noisy and superfluous instances.

To overcome disadvantages of *k-NN* and to deal with noisy and redundant instances in *SHCD*, we have proposed prototype generation techniques *F-kNN (Elbow method)*, *F-kNN (Silhouette value)* and hybrid techniques *TSR-FkNN (Elbow method)*, *TSR-FkNN (Silhouette value)*. We have also implemented *TSR-kNN* based on an existing prototype selection method and *F-kNN (Chang's algorithm)* based on prototype generation technique to compare the performance of proposed methods. To evaluate proposed and existing methods, we have suggested performance measures accuracy, *TSRR* and classification time. Further, the comparison between *TSRR* vs. accuracy and *TSRR* vs. classification time is done to accomplish assessment of proposed and existing methods. Fig. 8.1 depicts consolidated evaluation of all classifiers.

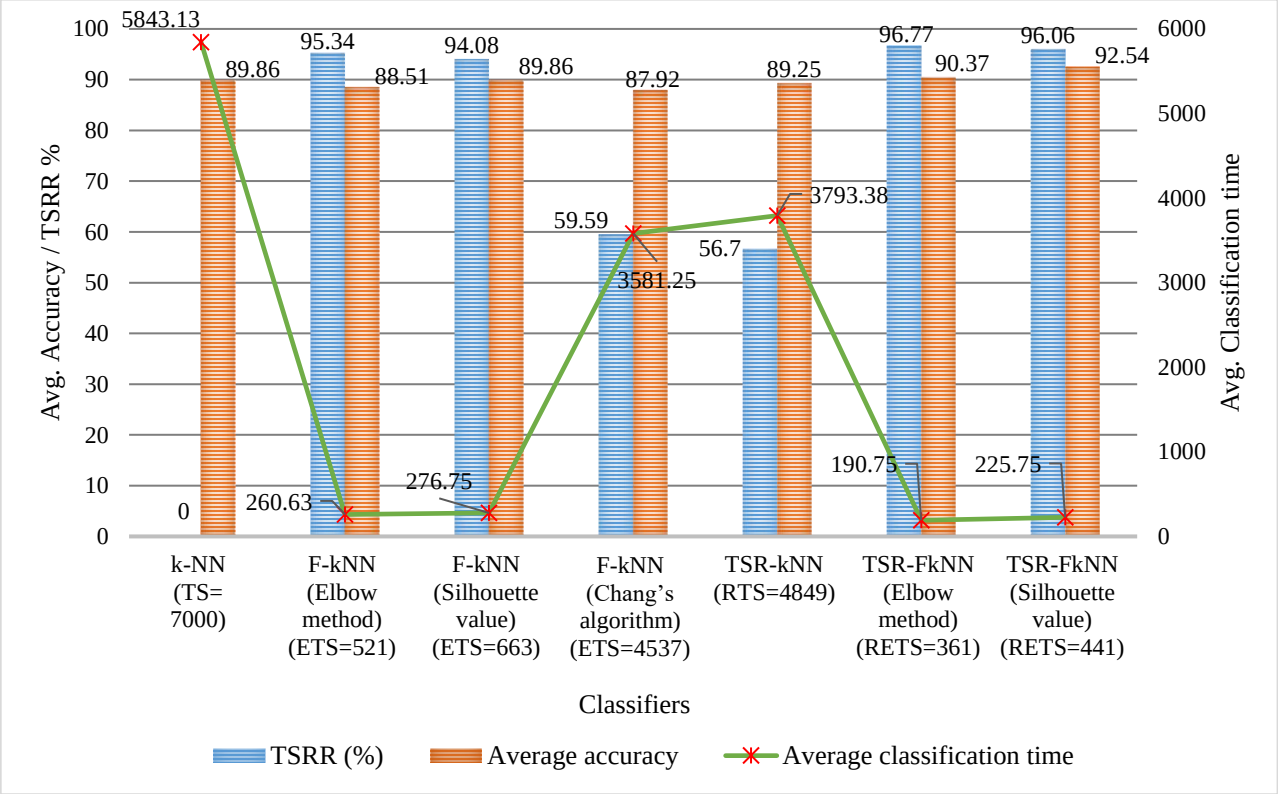


FIGURE 8.1 *TSRR* vs. Average accuracy vs. Average classification time

Storage requirement:

- The simple *k-NN* classifier is weakest in term of storage requirement as it does not apply any *PG/PS* technique on *TS* and retains all 7000 instances. The existing *PG* based classifier *F-kNN (Chang's algorithm)* generates *ETS*=4537, while existing *PS* based *TSR-kNN* reduces training instances with *RTS*=4849. Both *F-kNN (Chang's algorithm)* and *TSR-kNN* are not effective solutions as they retain a large number of training set instances.
- The proposed hybrid algorithm based classifier *TSR-FkNN (Elbow method)* is a most efficient algorithm in terms of storage reduction as it retains the lowest number of instances with *RETS*=361 followed by *TSR-FkNN (Silhouette value)* with *RETS*= 441.

Similarly, the proposed *PG* based classifiers *F-kNN (Elbow method)* and *F-kNN (Silhouette value)* stand 3rd and 4th, with *ETS*=521 and *ETS*=663 respectively in terms of storage requirement.

Classification time:

- The classification task performed by the simple *k-NN* classifier is longest with average classification time = 5843.13 milliseconds due to the extensive calculation performed by classification phase in presence of a high number of training set instances. Likewise, the existing *PG* based *TSR-kNN* and *PS* based *F-kNN (Chang's algorithm)* classifiers have high average classification time 3793.38 and 3581.25 milliseconds respectively because of a high number of instances in its prototype set.
- The proposed *PG* based *F-kNN (Elbow method)* and *F-kNN (Silhouette value)* classifiers stand 3rd and 4th in terms of average classification time with 260.63 and 276.75 milliseconds respectively.
- The highest performance is achieved by *TSR-FkNN (Elbow method)* on the basis of the speed of classification task as its average classification time is 190.75 milliseconds followed by *TSR-FkNN (Silhouette value)* with 225.75 milliseconds. Less number of training instances in its prototype set leads to the superior speedy performance of both hybrid method based classifiers.

Accuracy:

- The simple *k-NN* classifier has an average accuracy of 89.86%, which it has achieved with 7000 *TS* instances consisting of noisy and redundant instances. The proposed *PG* based *F-kNN (Elbow method)* classifier's average accuracy is 88.51%, which is less than *k-NN* classifier. It means, the generated prototype set (*ETS*=521) by *PG* method applied in *F-kNN (Elbow method)* is not an effective replacement of the original training set as the utilization of *ETS* resulted in a decrease in accuracy.
- The average accuracy of *F-kNN (Silhouette value)* is 89.86%, which is more than *F-kNN (Elbow method)* and same as *k-NN* classifiers' average accuracy. This implies, the proposed *PG* based generates effective prototype set (*ETS*=663) which resulted in a higher accuracy of *F-kNN (Silhouette value)*.

- Existing *PG* based *F-kNN (Chang's algorithm)* and *PS* based *TSR-kNN* classifiers' average accuracy is 87.92% and 89.25%, which is obtained by utilizing *ETS*=4537 and *RTS*=4849 respectively. It means, both classifiers are efficient but they are utilizing more number of training instances than proposed *F-kNN (Elbow method)* and *F-kNN (Silhouette value)*.
- The proposed *TSR-FkNN (Elbow method)* and *TSR-FkNN (Silhouette value)* have second highest and highest accuracy i.e. 90.37% and 92.54% respectively amongst all classifiers, while their prototype sets (*RETS*) are having lowest and second lowest 361 and 441 instances. It implies the proposed hybrid method based *TSR-FkNN (Elbow method)* and *TSR-FkNN (Silhouette value)* are able to classify unknown *SHCD* instance most effectively with generated prototype set. The *TSR-FkNN (Silhouette value)* have 80 more instances than *TSR-FkNN (Elbow method)* in its prototype set (*RETS*), but the accuracy of *TSR-FkNN (Silhouette value)* is superior than *TSR-FkNN (Elbow method)*.

From above discussion, it is deduced that, in terms of classification time, storage requirement and accuracy, the proposed hybrid method based *TSR-FkNN (Silhouette value)* classifier is recommended for classifying soil samples of *SHCD* into appropriate categories of macro and micro nutrients deficiencies.

8.2 Future scope

8.2.1 Open issues for future research

- As show in table 5.1, our proposed methods are performing decremental search to reduce the training set. Instead, other type of search namely, fixed, incremental or batch can be applied in proposed methods to explore the possibility of finding more appropriate condensed training set.
- The proposed *F-kNN* method is generating training sub set by applying optimization techniques in clustering namely, Elbow method and Silhouette value. Here, for optimization, relevant optimization methods can be explored and applied to current research.

- The proposed techniques adopted wrapper evaluation mechanism. Instead, other type of evaluation mechanism namely, Semi-wrapper and filter can be explored.
- The proposed methods takes soil macro and micro nutrients as input parameters which are numeric. For categorical or nominal values there is no provision provided which can be included in input and explored for classification accuracy.

8.2.2 Future scope based on latest research

- The proposed various prototype selection schemes applied to agriculture soil health card data set have given promising results. Still, in future, the scope to preserve the instances at class boundaries by applying the selection and the abstraction can be applied. For this, geometric characteristics of the distribution can be used, which may give good condensation rate. However, the impact of high computational cost can be one of the concerned areas for this method.
- In big data scenario, the huge amount of data becomes bottleneck to overcome by standard classification techniques. Further, in big data the skewed distribution of class may make classification task more difficult. In such scenario, evolutionary under sampling techniques shown promising results. Hence, for such situation, proposed prototype selection and generation based *k-NN* classifier can be applied for better efficiency with utilizing subset of original training set.
- The proposed instance selection techniques also can be applied to other classifiers such as artificial neural network (ANN) before it goes into training phase. For training of ANN, instead of original training instances if less number of instances are taken into consideration, then it will improve training phase in terms of speed.

List of References

1. Prajapati, B.P. and Kathiriya, D.R., (2019). A Hybrid Machine Learning Technique for Fusing Fast k-NN and Training Set Reduction: Combining Both Improves the Effectiveness of Classification. In *Progress in Advanced Computing and Intelligent Engineering*. Springer, Singapore. (Vol. 174),pp. 229-240.
2. Zhou, C., Lin, K., Xu, D., Chen, L., Guo, Q., Sun, C. and Yang, X., (2018). Near infrared computer vision and neuro-fuzzy model-based feeding decision system for fish in aquaculture. Computers and Electronics in Agriculture, 146, pp.114-124.
3. Fragni, R., Trifirò, A., Nucci, A., Seno, A., Allodi, A. and Di Rocco, M., (2018). Italian tomato-based products authentication by multi-element approach: a mineral elements database to distinguish the domestic provenance. Food Control.
4. Maione, C. and Barbosa, R.M., (2018). Recent applications of multivariate data analysis methods in the authentication of rice and the most analyzed parameters: A review. Critical reviews in food science and nutrition, pp.1-12.
5. Arthpedia,(2017),*Soilhealthcard*[online],http://www.arthapedia.in/index.php?itle=Soil_Health _Card_ (SHC), [Accessed on 25th Decemeber 2017].
6. Agricoo, (2017), *Operational guidelines for including the component of Soil Testing Labs* [online],http://agricoo-nic.in/sites/default/files/Corrected/Operational/guidelines,[Accessed on 25th Decemeber 2017].
7. Agri, (2017), *Soil health card project* [online], https://agri.gujarat.gov.in/soil-health-card-project.htm, [Accessed on 1st January 2018].
8. Guthrie P. M., (2017), *Looking backwards, looking forwards: SAS, data mining, and machine learning* [online], https://blogs.sas.com/content/subconsciousmusings, looking backwards looking forwards sas data mining and machine learning, [Accessed on 10th Jan. 2017]

9. Cihan, P., Gökçe, E. and Kalıpsız, O., (2017). A review of machine learning applications in veterinary field. *Kafkas Univ Vet Fak Derg*, *23*(4), pp.673-680.
10. Soils, (2017), Macro and Micro nutrient deficiency, [online], http://soils.wisc.edu/facstaff/barak/soilscience326/macronut.htm, [Accessed on 20th Jan 2018].
11. Soilhealth, (2017), *Soil health card, Govt. of India* [online], https://soilhealth.dac.gov.in/, [Accessed on 15th Jan 2018].
12. Pandey, A. and Jain, A., (2017). Comparative Analysis of KNN Algorithm using Various Normalization Techniques. *International Journal of Computer Network and Information Security*, *9*(11), p.36.
13. Cramer, S., Kampouridis, M., Freitas, A.A. and Alexandridis, A.K., (2017). An extensive evaluation of seven machine learning methods for rainfall prediction in weather derivatives. Expert Systems with Applications, 85, pp.169-181.
14. Rhee, J. and Im, J., (2017). Meteorological drought forecasting for ungauged areas based on machine learning: Using long-range climate forecast and remote sensing data. Agricultural and Forest Meteorology, 237, pp.105-122.
15. Ebrahimi, M.A., Khoshtaghaza, M.H., Minaei, S. and Jamshidi, B., (2017). Vision-based pest detection based on SVM classification method. Computers and Electronics in Agriculture, 137, pp.52-58.
16. Pantazi, X.E.; Moshou, D.; Oberti, R.; West, J.; Mouazen, A.M.; Bochtis, D., (2017) Detection of biotic and abiotic stresses in crops by using hierarchical self-organizing classifiers. Precis. Agric. 2017, 18, 383–393.
17. Witten, I.H., Frank, E., Hall, M.A. and Pal, C.J., (2016).'Introduction to Data Mining', *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann. 4th ed., pp. 3-38
18. Chen, H., Wu, W. and Liu, H.B., (2016). Assessing the relative importance of climate variables to rice yield variation using support vector machines. *Theoretical and applied climatology*, *126*(1-2), pp.105-111.
19. Gandhi, N., Armstrong, L.J., Petkar, O. and Tripathy, A.K., (2016), July. Rice crop yield prediction in India using support vector machines. In *Computer Science and Software Engineering (JCSSE), 2016 13th International Joint Conference.* IEEE. pp.1-5.

20. Navarro-Hellín, H., Martínez-del-Rincon, J., Domingo-Miguel, R., Soto-Valles, F. and Torres-Sánchez, R., (2016). A decision support system for managing irrigation in agriculture. *Computers and Electronics in Agriculture*, *124*, pp.121-131.
21. Prajapati, B.P. and Kathiriya, D.R., (2016). Towards the new Similarity Measures in Application of Machine Learning Techniques on Agriculture Dataset. *International Journal of Computer Applications*, *156*(11).
22. Richardson, A., Signor, B.M., Lidbury, B.A. and Badrick, T., (2016). Clinical chemistry in higher dimensions: machine-learning and enhanced prediction from routine clinical chemistry data. *Clinical biochemistry*, *49*(16-17), pp.1213-1220.
23. Aybar-Ruiz, A., Jiménez-Fernández, S., Cornejo-Bueno, L., Casanova-Mateo, C., Sanz-Justo, J., Salvador-Gonzalez, P. and Salcedo-Sanz, S., (2016). A novel grouping genetic algorithm–extreme learning machine approach for global solar radiation prediction from numerical weather models inputs. Solar Energy, 132, pp.129-142.
24. Singh, B. and Ryan, J., (2015). Managing fertilizers to enhance soil health. *International Fertilizer Industry Association, Paris, France*, pp.1-24.
25. Khedr, A.E., Kadry, M. and Walid, G., (2015). Proposed framework for implementing data mining techniques to enhance decisions in agriculture sector applied case on food security information center ministry of agriculture, Egypt. *Procedia Computer Science*, *65*, pp.633-642.
26. Hermann-Bank, M.L., Skovgaard, K., Stockmarr, A., Strube, M.L., Larsen, N., Kongsted, H., Ingerslev, H.C., Mølbak, L. and Boye, M., (2015). Characterization of the bacterial gut microbiota of piglets suffering from new neonatal porcine diarrhoea. *BMC veterinary research*, *11*(1), p.139.
27. Nantima, N., Ocaido, M., Ouma, E., Davies, J., Dione, M., Okoth, E., Mugisha, A. and Bishop, R., (2015). Risk factors associated with occurrence of African swine fever outbreaks in smallholder pig farms in four districts along the Uganda-Kenya border. *Tropical animal health and production*, *47*(3), pp.589-595.
28. Hempstalk, K., McParland, S. and Berry, D.P., (2015). Machine learning algorithms for the prediction of conception success to a given insemination in lactating dairy cows. *Journal of dairy science*, *98*(8), pp.5262-5273.

29. Starczewski, A. and Krzyżak, A., (2015), June. Performance evaluation of the silhouette index. In *International Conference on Artificial Intelligence and Soft Computing* (pp. 49-58). Springer, Cham.
30. Hossin, M. and Sulaiman, M.N., (2015). A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining & Knowledge Management Process*, *5*(2), p.1.
31. Bhuyar, V., (2014). Comparative Analysis of classification techniques on soil data to predict fertility rate for Aurangabad district. *Int. J. Emerg. Trends Technol. Comput. Sci*, *3*(2), pp.200-203.
32. McNairn, H., Kross, A., Lapen, D., Caves, R. and Shang, J., (2014). Early season monitoring of corn and soybeans with TerraSAR-X and RADARSAT-2. *International Journal of Applied Earth Observation and Geoinformation*, *28*, pp.252-259.
33. Xie, X.M., Xu, J.W., Zhao, J.F., Liu, S. and Wang, P., (2014). Soil Moisture Inversion Using AMSR-E Remote Sensing Data: An Artificial Neural Network Approach. In *Applied Mechanics and Materials*, Trans Tech Publications. (Vol. 501, pp. 2073-2076).
34. Veenadhari, S., Misra, B. and Singh, C.D., (2014), January. Machine learning approach for forecasting crop yield based on climatic parameters. In *Computer Communication and Informatics (ICCCI), 2014 International Conference on*. IEEE. pp.1-5.
35. Khoshnevisan, B., Rafiee, S. and Mousazadeh, H., (2014). Application of multi-layer adaptive neuro-fuzzy inference system for estimation of greenhouse strawberry yield. *Measurement*, *47*, pp.903-910.
36. Li, H., Leng, W., Zhou, Y., Chen, F., Xiu, Z. and Yang, D., (2014). Evaluation models for soil nutrient based on support vector machine and artificial neural networks. *The Scientific World Journal*, *2014*.
37. Meng, X., Zhang, Z. and Xu, X., (2014). A Novel K-Nearest Neighbor Algorithm Based on I-Divergence with Application to Soil Moisture Estimation in Maize Field. *JSW*, *9*(4), pp.841-846.
38. Amancio, D.R., Comin, C.H., Casanova, D., Travieso, G., Bruno, O.M., Rodrigues, F.A. and da Fontoura Costa, L., (2014). A systematic comparison of supervised classifiers. *PloS one*, *9*(4), p.e94137.

39. Larose, D.T. and Larose, C.D., (2014). *Discovering knowledge in data: an introduction to data mining*. John Wiley & Sons.

40. Whelan, B. and Taylor, J., (2013). 'Introduction to precision agriculture',*Precision agriculture for grain production systems*. Csiro publishing, pp.1-9.

41. Abdel-Rahman, E.M., Ahmed, F.B. and Ismail, R., (2013). Random forest regression and spectral band selection for estimating sugarcane leaf nitrogen concentration using EO-1 Hyperion hyperspectral data. *International Journal of Remote Sensing*, *34*(2), pp.712-728.

42. Papageorgiou, E.I., Aggelopoulou, K.D., Gemtos, T.A. and Nanos, G.D., (2013). Yield prediction in apples using Fuzzy Cognitive Map learning approach. *Computers and electronics in agriculture*, *91*, pp.19-29.

43. Chen, J.L., Li, G.S. and Wu, S.J., (2013). Assessing the potential of support vector machine for estimating daily solar radiation using sunshine duration. *Energy conversion and management*, *75*, pp.311-318.

44. McEvoy, F.J. and Amigo, J.M., (2013). Using machine learning to classify image features from canine pelvic radiographs: evaluation of partial least squares discriminant analysis and artificial neural network models. *Veterinary Radiology & Ultrasound*, *54*(2), pp.122-126.

45. Yunusa, A.J., Salako, A.E. and Oladejo, O.A., (2013). Principal component analysis of the morphostructure of Uda and Balami sheep of Nigeria. *Int. Res. J. Agric. Sci*, *1*(3), pp.45-51.

46. Dupuy, C., Morignat, E., Maugey, X., Vinard, J.L., Hendrikx, P., Ducrot, C., Calavas, D. and Gay, E., (2013). Defining syndromes using cattle meat inspection data for syndromic surveillance purposes: a statistical approach with the 2005–2010 data from ten French slaughterhouses. *BMC veterinary research*, *9*(1), p.88.

47. Charulatha, B.S., Rodrigues, P., Chitralekha, T. and Rajaraman, A., (2013). A Comparative study of different distance metrics that can be used in Fuzzy Clustering Algorithms. *International Journal of Emerging Trends and Technology in Computer Science (IJETTICS)*.

48. Ashari, A., Paryudi, I. and Tjoa, A.M., (2013). Performance comparison between Naïve Bayes, decision tree and k-nearest neighbor in searching alternative design in an energy simulation tool. *International Journal of Advanced Computer Science and Applications (IJACSA)*, *4*(11).

49. Garcia, S., Luengo, J., Sáez, J.A., Lopez, V. and Herrera, F., (2013). A survey of discretization techniques: Taxonomy and empirical analysis in supervised learning. *IEEE Transactions on Knowledge and Data Engineering*, *25*(4), pp.734-750.
50. Lee, P.H., Yu, Y.Y., McDowell, I., Leung, G.M. and Lam, T.H., (2013). A cluster analysis of patterns of objectively measured physical activity in Hong Kong. *Public health nutrition*, *16*(8), pp.1436-1444.
51. Arbelaitz, O., Gurrutxaga, I., Muguerza, J., Pérez, J.M. and Perona, I., (2013). An extensive comparative study of cluster validity indices. *Pattern Recognition*, *46*(1), pp.243-256.
52. Anzai, Y., (2012). 'Learning classificaion by dicovery', *Pattern recognition and machine learning*. Elsevier. pp.265-295.
53. Murphy, K.P., (2012). 'Bayesian statistics', *Machine Learning: A Probabilistic Perspective*. MIT Press., pp.155-180.
54. Petropoulos, G.P., Arvanitis, K. and Sigrimis, N., (2012). Hyperion hyperspectral imagery analysis combined with machine learning classifiers for land use/cover mapping. *Expert systems with Applications*, *39*(3), pp.3800-3809.
55. Baghdadi, N., Cresson, R., El Hajj, M., Ludwig, R. and La Jeunesse, I., (2012). Estimation of soil parameters over bare agriculture areas from C-band polarimetric SAR data using neural networks. *Hydrology and Earth System Sciences*, *16*, pp.p-1607.
56. Zaman, B., McKee, M. and Neale, C.M., (2012). Fusion of remotely sensed data for soil moisture estimation using relevance vector and support vector machines. *International journal of remote sensing*, *33*(20), pp.6516-6552.
57. i Casanova, P.M.P., Sinfreu, I. and Villalba, D., (2012). Principal component analysis of cephalic morphology to classify some Pyrenean cattle. *Animal Genetic Resources/Resources génétiques animales/Recursos genéticos animales*, *50*, pp.59-64.
58. Derrac, J., Triguero, I., García, S. and Herrera, F., (2012). Integrating instance selection, instance weighting, and feature weighting for nearest neighbor classifiers by coevolutionary algorithms. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 42(5), pp.1383-1397.
59. Bhavsar, H. and Ganatra, A., (2012). A comparative study of training algorithms for supervised machine learning. *International Journal of Soft Computing and Engineering (IJSCE)*, *2*(4), pp.2231-2307.

60. Garcia, S., Derrac, J., Cano, J. and Herrera, F., (2012). Prototype selection for nearest neighbor classification: Taxonomy and empirical study. *IEEE transactions on pattern analysis and machine intelligence*, *34*(3), pp.417-435.

61. Ougiaroglou, S. and Evangelidis, G., (2012), October. Fast and accurate k-nearest neighbor classification using prototype selection by clustering. In *Informatics (PCI), 2012 16th Panhellenic Conference on* (pp. 168-173). IEEE.

62. Triguero, I., Derrac, J., Garcia, S. and Herrera, F., (2012). A taxonomy and experimental study on prototype generation for nearest neighbor classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *42*(1), pp.86-100.

63. Han, J., Pei, J. and Kamber, M., (2011). 'Introducion', *Data mining: concepts and techniques*. Elsevier. 3rd ed., pp. 1-35.

64. Yang, C., Everitt, J.H. and Murden, D., (2011). Evaluating high resolution SPOT 5 satellite imagery for crop identification. *Computers and Electronics in Agriculture*, *75*(2), pp.347-354.

65. Tapia-Silva, F.O., Itzerott, S., Foerster, S., Kuhlmann, B. and Kreibich, H., (2011). Estimation of flood losses to agricultural crops using remote sensing. *Physics and Chemistry of the Earth, Parts A/B/C*, *36*(7-8), pp.253-265.

66. Ślósarz, P., Stanisz, M., Boniecki, P., Lisiak, D. and Ludwiczak, A., (2011). Artificial neural network analysis of ultrasound image for the estimation of intramuscular fat content in lamb muscle. *African Journal of Biotechnology*, *10*(55), pp.11792-11796.

67. Yang, J., Zhang, L., Yang, J.Y. and Zhang, D., (2011). From classifiers to discriminators: A nearest neighbor rule induced discriminant analysis. *Pattern Recognition*, *44*(7), pp.1387-1402.

68. Villegas, M. and Paredes, R., (2011). Dimensionality reduction by minimizing nearest-neighbor classification error. *Pattern Recognition Letters*, *32*(4), pp.633-639.

69. Hu, Q., Zhu, P., Yang, Y. and Yu, D., (2011). Large-margin nearest neighbor classifiers via sample weight learning. *Neurocomputing*, *74*(4), pp.656-660.

70. Triguero, I., García, S. and Herrera, F., (2011). Differential evolution for optimizing the positioning of prototypes in nearest neighbor classification. *Pattern Recognition*, *44*(4), pp.901-916.

71. Rendón, E., Abundez, I., Arizmendi, A. and Quiroz, E.M., (2011). Internal versus external cluster validation indexes. *International Journal of computers and communications, 5*(1), pp.27-34.

72. Warns-Petit, E., Morignat, E., Artois, M. and Calavas, D., (2010). Unsupervised clustering of wildlife necropsy data for syndromic surveillance. *BMC veterinary research, 6*(1), p.56.

73. Yang, J.M., Yu, P.T. and Kuo, B.C., (2010). A nonparametric feature extraction and its application to nearest neighbor classification for hyperspectral image data. *IEEE Transactions on Geoscience and Remote Sensing*, *48*(3), pp.1279-1293.

74. Wang, C.Y., Zhang, K., Yan, Y.G. and Li, J.G., (2010), August. A K-Nearest Neighbor Algorithm based on cluster in text classification. In *Computer, Mechatronics, Control and Electronic Engineering (CMCE), 2010 International Conference on, IEEE.* (Vol. 1), pp.225-228.

75. Majumdar, A. and Ward, R.K., (2010). Robust classifiers for data reduced via random projections. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *40*(5), pp.1359-1371.

76. Hernández-Rodríguez, S., Martínez-Trinidad, J.F. and Carrasco-Ochoa, J.A., (2010). Fast k most similar neighbor classifier for mixed data (tree k-MSN). *Pattern recognition*, *43*(3), pp.873-886.

77. Zhang, Y. and Zhou, Z.H., (2010). Multilabel dimensionality reduction via dependence maximization. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *4*(3), p.14.

78. Triguero, I., García, S. and Herrera, F., (2010). IPADE: Iterative prototype adjustment for nearest neighbor classification. *IEEE Transactions on Neural Networks*, *21*(12), pp.1984-1990.

79. Mucherino, A., Papajorgji, P. and Pardalos, P.M., (2009).'Clustering by k-Means', *Data mining in agriculture* (Vol. 34). Springer Science & Business Media,pp.47-80.

80. Zhang, Y., Wang, C., Wu, J., Qi, J. and Salas, W.A., (2009). Mapping paddy rice with multitemporal ALOS/PALSAR imagery in southeast China. *International Journal of Remote Sensing*, *30*(23), pp.6301-6315.

81. Lakhankar, T., Ghedira, H., Temimi, M., Sengupta, M., Khanbilvardi, R. and Blake, R.,(2009). Non-parametric methods for soil moisture retrieval from satellite remote sensing data. *Remote sensing*, *1*(1), pp.3-21.

82. Smith, B.A., Hoogenboom, G. and McClendon, R.W., (2009). Artificial neural networks for automated year-round temperature prediction. *Computers and Electronics in Agriculture*, *68*(1), pp.52-61.

83. Nanni, L. and Lumini, A., (2009). Particle swarm optimization for prototype reduction. *Neurocomputing*, *72*(4-6), pp.1092-1097.

84. Fayed, H.A. and Atiya, A.F., (2009). A novel template reduction approach for the k-nearest neighbor method. *IEEE Transactions on Neural Networks*, *20*(5), pp.890-896.

85. Yong, Z., Youwen, L. and Shixiong, X., (2009). An improved KNN text classification algorithm based on clustering. *Journal of computers*, *4*(3), pp.230-237.

86. Statnikov, A., Wang, L. and Aliferis, C.F., (2008). A comprehensive comparison of random forests and support vector machines for microarray-based cancer classification. *BMC bioinformatics*, *9*(1), p.319.

87. Schellberg, J., Hill, M.J., Gerhards, R., Rothmund, M. and Braun, M., (2008). Precision agriculture on grassland: Applications, perspectives and constraints. *European Journal of Agronomy*, *29*(2-3), pp.59-71.

88. Ortiz-Pelaez, Á. and Pfeiffer, D.U., (2008). Use of data mining techniques to investigate disease risk classification as a proxy for compromised biosecurity of cattle herds in Wales. *BMC Veterinary Research*, *4*(1), p.24.

89. Sima, C. and Dougherty, E.R., (2008). The peaking phenomenon in the presence of feature-selection. *Pattern Recognition Letters*, *29*(11), pp.1667-1674.

90. Li, B., Chen, Y.W. and Chen, Y.Q., (2008). The nearest neighbor algorithm of local probability centers. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *38*(1), pp.141-154.

91. He, Q.P. and Wang, J., (2008), June. Principal component based k-nearest-neighbor rule for semiconductor process fault detection. In *American Control Conference*, IEEE. pp. 1606-1611.

92. Armstrong, L.J., Diepeveen, D. and Maddern, R., (2007), December. The application of data mining techniques to characterize agricultural soil profiles. In *Proceedings of the sixth Australasian conference on Data mining and analytics.* Australian Computer Society, Inc., Vol (70),.pp. 85-100.

93. Tan, P., Steinbach, M., Karpatne, A. and Kumar, V. (2007). 'Introduction', *Introduction to data mining*, Pearson Education India.. 3rd ed., pp. 30-65.

94. Cios, K.J., Pedrycz, W., Swiniarski, R.W. and Kurgan, L.A., (2007). 'Introduction', *Data mining: a knowledge discovery approach*. Springer Science & Business Media. pp.3-7

95. Pal, N. ed., (2007). 'Trends in Data Mining and Knowledge Discovery', *Advanced techniques in knowledge discovery and data mining*. Springer Science & Business Media., pp.1-12.

96. Asuncion, A. and Newman, D.J., (2007). UCI Machine Learning Repository [http://www. ics. uci. edu/~ mlearn/MLRepository. html]. Irvine, CA: University of California. *School of Information and Computer Science*, *12*.

97. Meyer, K., (2007). Multivariate analyses of carcass traits for Angus cattle fitting reduced rank and factor analytic models. *Journal of Animal Breeding and Genetics*, *124*(2), pp.50-64.

98. Gao, Q.B. and Wang, Z.Z., (2007). Center-based nearest neighbor classifier. *Pattern Recognition*, *40*(1), pp.346-349.

99. Zhang, M.L. and Zhou, Z.H., (2007). ML-KNN: A lazy learning approach to multi-label learning. *Pattern recognition*, *40*(7), pp.2038-2048.

100. Fayed, H.A., Hashem, S.R. and Atiya, A.F., (2007). Self-generating prototypes for pattern classification. *Pattern Recognition*, *40*(5), pp.1498-1509.

101. Angiulli, F., (2007). Fast nearest neighbor condensation for large data sets classification. *IEEE Transactions on Knowledge and Data Engineering*, *19*(11).

102. Brun, M., Sima, C., Hua, J., Lowey, J., Carroll, B., Suh, E. and Dougherty, E.R., (2007). Model-based evaluation of clustering validation measures. *Pattern recognition*, *40*(3), pp.807-824.

103. Caraviello, D.Z., Weigel, K.A., Craven, M., Gianola, D., Cook, N.B., Nordlund, K.V., Fricke, P.M. and Wiltbank, M.C., (2006). Analysis of reproductive performance of lactating cows on large dairy farms using machine learning algorithms. *Journal of dairy science*, *89*(12), pp.4703-4722.

104. Behrens, T. and Scholten, T., (2006). A comparison of data-mining techniques in predictive soil mapping. *Developments in Soil Science*, *31*, pp.353-617.

105. Bhattacharya, B. and Solomatine, D.P., (2006). Machine learning in soil classification. *Neural networks*, *19*(2), pp.186-195.
106. Deza, E. and Deza, M.M., (2006). Ch. 22. Dictionary of Distances, *Elsevier*.,p.279.
107. Zezula, P., Amato, G., Dohnal, V. and Batko, M., (2006). Foundations of Metric Space Searching. *Similarity Search The Metric Space Approach*, pp.5-66.
108. Lavine, B.K. and Mirjankar, N., (2012), Clustering and classification of analytical data. *Encyclopedia of Analytical Chemistry: Applications, Theory and Instrumentation*., pp.92-97.
109. Ghosh, A.K., Chaudhuri, P. and Murthy, C.A., (2006). Multiscale classification using nearest neighbor density estimates. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *36*(5), pp.1139-1148.
110. Yu, X.G. and Yu, X.P., (2006), August. The Research on an adaptive k-nearest neighbors classifier. In *Machine Learning and Cybernetics, 2006 International Conference on*, IEEE., pp.1241-1246.
111. Parveen, P. and Thuraisingham, B., (2006), November. Face recognition using multiple classifiers. In *Tools with Artificial Intelligence, 2006. ICTAI'06. 18th IEEE International Conference on,* IEEE.,pp. 179-186.
112. Lozano, M., Sotoca, J.M., Sánchez, J.S., Pla, F., Pękalska, E. and Duin, R.P., (2006). Experimental study on prototype optimisation algorithms for prototype-based classification in vector spaces. *Pattern Recognition*, *39*(10), pp.1827-1838.
113. Franti, P., Virmajoki, O. and Hautamaki, V., 2006. Fast agglomerative clustering using a k-nearest neighbor graph. *IEEE transactions on pattern analysis and machine intelligence*, *28*(11), pp.1875-1881.
114. Rokach, L. and Maimon, O., (2005). 'Clustering methods'. In *Data mining and knowledge discovery handbook,* Springer, Boston, MA, pp. 321-352.
115. Domeniconi, C., Gunopulos, D. and Peng, J., (2005). Large margin nearest neighbor classifiers. *IEEE transactions on neural networks*, *16*(4), pp.899-909.
116. Ghosh, A.K., Chaudhuri, P. and Murthy, C.A., (2005). On visualization and aggregation of nearest neighbor classifiers. *IEEE transactions on pattern analysis and machine intelligence*, *27*(10), pp.1592-1602.

117. Angiulli, F., (2005), August. Fast condensed nearest neighbor rule. In *Proceedings of the 22nd international conference on Machine learning* (pp. 25-32). ACM.

118. Raicharoen, T. and Lursinsap, C., (2005). A divide-and-conquer approach to the pairwise opposite class-nearest neighbor (POC-NN) algorithm. *Pattern recognition letters*, *26*(10), pp.1554-1567.

119. Barandela, R., Ferri, F.J. and Sánchez, J.S., (2005). Decision boundary preserving prototype selection for nearest neighbor classification. *International Journal of Pattern Recognition and Artificial Intelligence*, *19*(06), pp.787-806.

120. Kuramochi, M. and Karypis, G., (2005). Gene classification using expression profiles: A feasibility study. *International Journal on Artificial Intelligence Tools*, *14*(04), pp.641-660.

121. Sánchez, J.S., (2004). High training set size reduction by space partitioning and prototype abstraction. *Pattern Recognition*, *37*(7), pp.1561-1564.

122. Yuan, W., Liu, J. and Zhou, H.B., (2004), August. An improved KNN method and its application to tumor diagnosis. In *Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference on*. IEEE.,(Vol. 5), pp. 2836-2841

123. Reunanen, J., (2004), April. A Pitfall in Determining the Optimal Feature Subset Size. In *PRIS* (pp. 176-185).

124. Zhang, B. and Srihari, S.N., (2004). Fast k-nearest neighbor classification using cluster-based trees. *IEEE Transactions on Pattern analysis and machine intelligence*, *26*(4), pp.525-528.

125. Kuncheva, L.I., (2004). 'Multiple classifier systems', *Combining pattern classifiers: methods and algorithms*. John Wiley & Sons.,pp.101-109.

126. Pan, F., Wang, B., Hu, X. and Perrizo, W., (2004). Comprehensive vertical sample-based KNN/LSVM classification for gene expression analysis. *Journal of Biomedical Informatics*, *37*(4), pp.240-248.

127. Zhu, X. and Wu, X., (2004). Class noise vs. attribute noise: A quantitative study. *Artificial intelligence review*, *22*(3), pp.177-210.

128. Jankowski, N. and Grochowski, M., (2004), June. Comparison of instances seletion algorithms i. algorithms survey. In *International conference on artificial intelligence and soft computing* (pp. 598-603). Springer, Berlin, Heidelberg.

129. Heathman, G.C., Starks, P.J., Ahuja, L.R. and Jackson, T.J., (2003). Assimilation of surface soil moisture to estimate profile soil water content. *Journal of Hydrology*, *279*(1-4), pp.1-17.

130. Petridis, V. and Kaburlasos, V.G., (2003). FINkNN: A fuzzy interval number k-nearest neighbor classifier for prediction of sugar production from populations of samples. *Journal of Machine Learning Research*, *4*(Apr), pp.17-37.

131. Gavin, D.G., Oswald, W.W., Wahl, E.R. and Williams, J.W., (2003). A statistical approach to evaluating distance metrics and analog assignments for pollen records. *Quaternary Research*, *60*(3), pp.356-367.

132. Petridis, V. and Kaburlasos, V.G., (2003). FINkNN: A fuzzy interval number k-nearest neighbor classifier for prediction of sugar production from populations of samples. *Journal of Machine Learning Research*, *4*(Apr), pp.17-37.

133. Sánchez, J.S., Barandela, R., Marqués, A.I., Alejo, R. and Badenas, J., (2003). Analysis of new techniques to obtain quality training sets. *Pattern Recognition Letters*, *24*(7), pp.1015-1022.

134. Schölkopf, B., Smola, A.J. and Bach, F., (2002). 'A tutorial Introduction',*Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press.,pp.1-21.

135. Ibrahimov, O., Sethi, I. and Dimitrova, N., (2002). The performance analysis of a Chi-square similarity measure for topic related clustering of noisy transcripts. In *Object recognition supported by user interaction for service robots*, IEEE.,(Vol. 4), pp. 285-288.

136. Vincent, P. and Bengio, Y., (2002). K-local hyperplane and convex distance nearest neighbor algorithms. In *Advances in Neural Information Processing Systems*,pp. 985-992.

137. Buttrey, S.E. and Karo, C., (2002). Using k-nearest-neighbor classification in the leaves of a tree. *Computational Statistics & Data Analysis*, *40*(1), pp.27-37.

138. Mollineda, R.A., Ferri, F.J. and Vidal, E., (2002). A merge-based condensing strategy for multiple prototype classifiers. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *32*(5), pp.662-668.

139. Lam, W., Keung, C.K. and Liu, D., (2002). Discovering useful concept prototypes for classification based on filtering and abstraction. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (8), pp.1075-1090.

140. Brighton, H. and Mellish, C., (2002). Advances in instance selection for instance-based learning algorithms. *Data mining and knowledge discovery*, *6*(2), pp.153-172.

141. Wu, Y., Ianakiev, K. and Govindaraju, V., (2002). Improved k-nearest neighbor classification. *Pattern recognition*, *35*(10), pp.2311-2318.

142. Zhang, H. and Sun, G., (2002). Optimal reference subset selection for nearest neighbor classification by tabu search. *Pattern Recognition*, *35*(7), pp.1481-1490.

143. Wolpert, D.H., (2002). The supervised learning no-free-lunch theorems. In *Soft computing and industry* . Springer, London. pp. 25-42.

144. Kohonen, T., (2001). 'Learning vector quantization'. *In Self-organizing maps,* Springer, Berlin, Heidelberg.pp. 245-261.

145. Hand, D.J., Mannila, H. and Smyth, P., (2001). *Principles of data mining (adaptive computation and machine learning)*. Cambridge, MA: MIT press.,pp. 361-452

146. Peng, J., Heisterkamp, D.R. and Dai, H.K., (2001). LDA/SVM driven nearest neighbor classification. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on* . IEEE.(Vol. 1), pp.I-I.

147. Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C. and Wirth, R., (2000). *CRISP-DM 1.0 Step-by-step data mining guide [online]*, http://www.crisp-dm.org/CRISPWP-0800.pdf, [Accessed on 10th Jan 2018].

148. Muchoney, D., Borak, J., Chi, H., Friedl, M., Gopal, S., Hodges, J., Morrow, N. and Strahler, A., (2000). Application of the MODIS global supervised classification model to vegetation and land cover mapping of Central America. *International Journal of Remote Sensing*, *21*(6-7), pp.1115-1138.

149. Dasarathy, B.V., Sánchez, J.S. and Townsend, S., (2000). Nearest neighbour editing and condensing tools–synergy exploitation. *Pattern Analysis & Applications*, *3*(1), pp.19-30.

150. Wilson, D.R. and Martinez, T.R., (2000). Reduction techniques for instance-based learning algorithms. *Machine learning*, *38*(3), pp.257-286.

151. Maletic, J.I. and Marcus, A., (2000), October. Data Cleansing: Beyond Integrity Analysis. In *Iq* , pp. 200-209.

152. Rodriguez-Iturbe, I., D'odorico, P., Porporato, A. and Ridolfi, L., (1999). On the spatial and temporal links between vegetation, climate, and soil moisture. *Water Resources Research*, *35*(12), pp.3709-3722.

153. Brighton, H. and Mellish, C., (1999), September. On the consistency of information filters for lazy learning algorithms. In *European conference on principles of data mining and knowledge discovery* . Springer, Berlin, Heidelberg., pp. 283-288.

154. Mirkin, B., (1998). 'In Classification, data analysis, and data highways',*Mathematical classification and clustering: From how to what and why*. Springer, Berlin, Heidelberg,pp.172-181.

155. Bay, S.D., (1998), July. Combining Nearest Neighbor Classifiers Through Multiple Feature Subsets. In *ICML* (Vol. 98), pp. 37-45.

156. Bezdek, J.C., Reichherzer, T.R., Lim, G.S. and Attikiouzel, Y., (1998). Multiple-prototype classifier design. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 28*(1), pp.67-79.

157. Glover, F. and Laguna, M., (1998). Tabu search. In *Handbook of combinatorial optimization* Springer, Boston, MA. pp. 2093-2229.

158. Ohno-Machado, L., Fraser, H.S. and Ohrn, A., (1998). Improving machine learning performance by removing redundant cases in medical data sets. In *Proceedings of the AMIA Symposium*. American Medical Informatics Association. p. 523

159. Wettschereck, D., Aha, D.W. and Mohri, T., (1997). A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms. *Artificial Intelligence Review*, *11*(1-5), pp.273-314.

160. Wilson, D.R. and Martinez, T.R., (1997). Improved heterogeneous distance functions. *Journal of artificial intelligence research*, *6*, pp.1-34.

161. Wu, W. and Massart, D.L., (1997). Regularised nearest neighbour classification method for pattern recognition of near infrared spectra. *Analytica chimica acta*, *349*(1-3), pp.253-261.

162. Alsberg, B.K., Goodacre, R., Rowland, J.J. and Kell, D.B., (1997). Classification of pyrolysis mass spectra by fuzzy multivariate rule induction-comparison with regression, K-nearest neighbour, neural and decision-tree methods. *Analytica Chimica Acta*, *348*(1-3), pp.389-407.

163. Hamamoto, Y., Uchimura, S. and Tomita, S., (1997). A bootstrap technique for nearest neighbor classifier design. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *19*(1), pp.73-79.

164. Wilson, D.R. and Martinez, T.R., (1997), July. Instance pruning techniques. In *ICML* (Vol. 97), pp. 403-411.

165. Wolpert, D.H. and Macready, W.G., (1997). No free lunch theorems for optimization. IEEE transactions on evolutionary computation, 1(1), pp.67-82.

166. Chen, C.H. and Jóźwik, A., (1996). A sample set condensation algorithm for the class sensitive artificial neural network. *Pattern Recognition Letters*, *17*(8), pp.819-823.

167. Fayyad, U., Piatetsky-Shapiro, G. and Smyth, P., (1996). From data mining to knowledge discovery in databases. *AI magazine*, 17(3), p.37.

168. Boer, M., Del Barrio, G. and Puigdefábres, J., (1996). Mapping soil depth classes in dry Mediterranean areas using terrain attributes derived from a digital elevation model. *Geoderma*, *72*(1-2), pp.99-118.

169. Breiman, L., (1996). Bagging predictors. *Machine learning*, *24*(2), pp.123-140.

170. Chaudhuri, B.B., (1996). A new definition of neighborhood of a point in multi-dimensional space. *Pattern Recognition Letters*, *17*(1), pp.11-17.

171. Hassoun, M.H., (1995). 'Computational capabilities of ANN', *Fundamentals of artificial neural networks*. MIT press., pp.37-50.

172. McQueen, R.J., Garner, S.R., Nevill-Manning, C.G. and Witten, I.H., (1995). Applying machine learning to agricultural data. *Computers and electronics in agriculture*, *12*(4), pp.275-293.

173. Kuncheva, L.I., (1995). Editing for the k-nearest neighbors rule by a genetic algorithm. *Pattern Recognition Letters*, *16*(8), pp.809-814.

174. Wu, X., (1995). 'Constructing decision tree with ID3 and C4.5', Knowledge Acquisition from Databases. *Ablex Pulishing Corp*.,pp.33-48.

175. Dasarathy, B.V., (1994). Minimal consistent set (MCS) identification for optimal nearest neighbor decision systems design. *IEEE Transactions on Systems, Man, and Cybernetics*, *24*(3), pp.511-517.

176. Faragó, A., Linder, T. and Lugosi, G., (1993). Fast nearest-neighbor search in dissimilarity spaces. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (9), pp.957-962.

177. Xie, Q., Laszlo, C.A. and Ward, R.K., (1993). Vector quantization technique for nonparametric classifier design. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *15*(12), pp.1326-1330.

178. Efron, B., (1992). Bootstrap methods: another look at the jackknife. In *Breakthroughs in statistics* . Springer, New York, NY.pp. 569-593.

179. Aha, D.W., Kibler, D. and Albert, M.K., (1991). 'Instance-based learning algorithms'. *Machine learning*, Springer Boston, MA, *6*(1), pp.37-66.

180. Broder, A.J., (1990). Strategies for efficient incremental nearest neighbor search. *Pattern Recognition*, *23*(1-2), pp.171-178.

181. Parthasarathy, G. and Chatterji, B.N., (1990). A class of new KNN methods for low sample problems. *IEEE transactions on systems, man, and cybernetics*, *20*(3), pp.715-718.

182. Fukunaga, K. and Hayes, R.R., (1989). Effects of sample size in classifier design. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *11*(8), pp.873-885.

183. Todeschini, R., (1989). k-nearest neighbour method: The influence of data transformations and metrics. *Chemometrics and intelligent laboratory systems*, *6*(3), pp.213-220.

184. Kim, B.S. and Park, S.B., (1986). A fast k nearest neighbor finding algorithm based on the ordered partition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6), pp.761-766.

185. Ruiz, E.V., (1986). An algorithm for finding nearest neighbours in (approximately) constant average time. *Pattern Recognition Letters*, *4*(3), pp.145-157.

186. Toussaint, G.T., (1985). 'The application of Voronoi diagrams to nonparametric decision rules'. *Computer Science and Statistics: The Interface Billard, L.*, pp.97-108.

187. Milligan, G.W. and Cooper, M.C., (1985). An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, *50*(2), pp.159-179.

188. Kittler, J. and Devijver, P.A., (1982). Statistical properties of error estimators in performance assessment of recognition systems. *IEEE transactions on pattern analysis and machine intelligence*, (2), pp.215-220.

189. Coomans, D. and Massart, D.L., (1982). Alternative k-nearest neighbour rules in supervised pattern recognition: Part 1. k-Nearest neighbour classification by using alternative voting rules. *Analytica Chimica Acta*, *136*, pp.15-27.

190. Tomek, I., (1976). A generalization of the k-NN rule. *IEEE Transactions on Systems, Man, and Cybernetics*, (2), pp.121-126.

191. Tomek, I., (1976). An experiment with the edited nearest-neighbor rule. *IEEE Transactions on systems, Man, and Cybernetics*, (6), pp.448-452.

192. Ritter, G., Woodruff, H., Lowry, S. and Isenhour, T., (1975). An algorithm for a selective nearest neighbor decision rule (Corresp.). *IEEE Transactions on Information Theory*, *21*(6), pp.665-669.

193. Chang, C.L., (1974). Finding prototypes for nearest neighbor classifiers. *IEEE Transactions on Computers*, *100*(11), pp.1179-1184.

194. Wilson, D.L., (1972). Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man, and Cybernetics*, (3), pp.408-421.

195. Gates, G., (1972). The reduced nearest neighbor rule (Corresp.). *IEEE transactions on information theory*, *18*(3), pp.431-433.

196. Hart, P., (1968). The condensed nearest neighbor rule (Corresp.). *IEEE transactions on information theory*, *14*(3), pp.515-516.

197. Cover, T. and Hart, P., (1967). Nearest neighbor pattern classification. *IEEE transactions on information theory*, *13*(1), pp.21-27.

Publications

1. Prajapati, B.P. and Kathiriya, D.R., (2016). A Novel Framework for Association Rule Mining to observe Crop Cultivation Practices based on Soil type. *International Journal of Computer Science and Information Security*, *14*(9), p.523. ISSN 1947-5500.

2. Prajapati, B.P. and Kathiriya, D.R., (2016). Reducing execution time of Machine Learning Techniques by Applying Greedy Algorithms for Training Set Reduction. *International Journal of Computer Science and Information Security*, *14*(12), p.705., ISSN 1947-5500.

3. Prajapati, B.P. and Kathiriya, D.R., (2016). Evaluation of Effectiveness of k-Means Cluster based Fast k-Nearest Neighbor classification applied on Agriculture Dataset. *International Journal of Computer Science and Information Security*, *14*(10), p.800., ISSN 1947-5500.

4. Prajapati, B.P. and Kathiriya, D.R., (2016). Towards the new Similarity Measures in Application of Machine Learning Techniques on Agriculture Dataset. *International Journal of Computer Applications*, *156*(11)., ISSN 0975–8887.

5. Prajapati, B.P. and Kathiriya, D.R., (2019). A Hybrid Machine Learning Technique for Fusing Fast k-NN and Training Set Reduction: Combining Both Improves the Effectiveness of Classification. In *Progress in Advanced Computing and Intelligent Engineering*. Springer, Singapore. (Vol. 174),pp. 229-240. (Paper presented in ICACIE 2017, 23-25 Nov. 2017, available as book chapter, DOI: https://doi.org/10.1007/978-981-13-0224-4_21).