Investigation and comparative analysis of data mining techniques for the prediction of crop yield

Kanwal Preet Singh Attwal* and
Amardeep Singh Dhiman

Department of Computer Science and Engineering,
Punjabi University,
Patiala, India
Email: kanwalp78@yahoo.com
Email: amardeep_dhiman@yahoo.com
*Corresponding author

Abstract: Crop yield is affected by climatic, management, geographical, biological and other such factors. Data mining techniques can be used to analyse the effect of these factors on crop yield and to predict crop yield based on these factors. The current paper focuses on the sequence of steps to be followed in data mining process for prediction of crop yield – starting from the determination of research goals to the application of the data mining techniques to build a model. The study applies the defined data mining process to build a model for the prediction of paddy yield based on different climatic factors. The current research also provides an insight to the different metrics that can be used to evaluate various supervised data mining techniques. The metrics have been divided into three categories – threshold evaluation metrics, numerical evaluation metric, and built time and size metrics. Comparative analysis of five supervised data mining techniques has been carried out on the basis of their performance in these three categories of metrics.

Keywords: agricultural data mining; yield prediction; data mining process; data mining tasks; data mining techniques; classification techniques; classification evaluation metrics.

Reference to this paper should be made as follows: Attwal, K.P.S. and Dhiman, A.S. (2020) 'Investigation and comparative analysis of data mining techniques for the prediction of crop yield', *Int. J. Sustainable Agricultural Management and Informatics*, Vol. 6, No. 1, pp.43–74.

Biographical notes: Kanwal Preet Singh Attwal is an Assistant Professor in Computer Science and Engineering at the Punjabi University, Patiala. He has to his credit more than 15 research papers published in reputed international and national journals, besides various presentations in international and national seminars and conferences. He has guided more than ten research scholars in the field of Data Mining. His teaching experience of 16 years is reflected in his endeavours to take the research in data mining to advance level in agriculture.

Amardeep Singh Dhiman is a Professor and the Head of the Department of Computer Science and Engineering at the Punjabi University, Patiala. In his teaching experience of more than 20 years, he has guided 60 research scholars at Master's level and five research scholars at Doctorate level. He has to his credit more than 40 research papers published in reputed international and national journals, besides two books on *Bioinformatics Computing* and *Soft Computing*. He has successfully handled a number of the State funded projects.

1 Introduction

With the technological advancement, the research and development in agriculture have also been revolutionised. The development of techniques and agricultural innovations require a planned, organised and reliable database of agriculture. The agricultural data collected by statistics department extends well beyond the data requirements of the immediate sector. Besides, the continuous generation of data has resulted in an extensive online pool of data. It is a very tedious task to find the desired data from this vast pool, hence requires an efficient technological process to organise such data. Similar to gold mining, data mining techniques help to extract the desired data from a vast amount of data. Thus, data mining is the task of obtaining potentially interesting patterns and relationships from data. Data mining is actually a step of a larger process known as knowledge discovery in databases (KDD) (Fayyad et al., 1996). It consists of selection, pre-processing and transformation of data, mining useful patterns from transformed data and interpretation or evaluation of those patterns to gain knowledge.

However, the robust nature of agricultural data due to its different formats, complexity, multidimensionality and noise, makes mining of desired data a challenging task. Agricultural yield is affected by climatic, management, geographical, biological and other such factors, thus these factors are essential elements to build a model for crop yield prediction. The patterns are mined from these factors and actual experimental data, using statistical and mathematical modelling. This generates characterisation and prediction models which are employed by both farmers and researchers for taking vital decisions (Cruz et al., 2014). This foregrounds the essentiality of advance research in agricultural data mining process and a comprehensive analysis of data mining process for the prediction of crop yield.

It is in this context that the current paper examines various tasks and techniques adopted in data mining. An attempt has been made to acquaint the researchers with the various techniques used in data mining. However, its much wider scope puts certain limits on the inclusiveness of the paper. Besides, the paper focuses on the investigation of the data mining process to be followed for prediction of crop yield.

2 Literature survey

A substantial amount of research has been done in the application of data mining technique in agriculture. The extensive scope of using data mining in agriculture foregrounds the essentiality of the investigation of the research done in this field till now; this also helps in achieving its comprehensive understanding.

Landau et al. (2000) used multiple-regression to build a model which predicts wheat yield in response to various environmental factors. Ekasingh et al. (2005) used decision tree approach to simulate land use pattern in Northern Thailand based on various socioeconomic data such as land unit, estimated cost of production, etc. Ruß et al. (2008) made a model using neural networks to predict wheat yield by analysing factors such as electrical conductivity of soil, vegetation index and nitrogen fertilisation. In another study, Rub (2009) evaluated four regression techniques for prediction of wheat yield and concluded that support vector regression gave the best results. Vagh and Xiao (2012b) used different classification techniques to mine effect of temperature on wheat yield in Western Australia. They extended their study in Vagh and Xiao (2012a) to study the

effect of both rainfall and temperature on wheat yield at shire level in Western Australia. Farook et al. (2012) mined the effect of climatic factors on mango yield using regression analysis. Ghosh et al. (2012) applied decision tree technique to soil database to classify soil texture based on soil properties. Leona and Jalao (2013) used rule set induction for prediction of corn yield based on different climate related and agronomic variables. Haghverdi et al. (2014) employed decision trees and neural network to develop Production functions to estimate wheat yield under simultaneous salinity and water stress. The study compared the developed production functions with some four well known Production functions and found that the performance of neural network performance function was better than the others. Everingham et al. (2016) used random forest (decision tree technique) to predict sugarcane yield. The yield was predicted at three stages of crop growth so that farmer could accordingly adjust the management practices to get the optimum yield and for getting better economic and environmental outcomes. Majumdar et al. (2017) did a comparative analysis of different clustering techniques to find out the optimal climate requirement of wheat and concluded that DBSCAN gave the best clustering quality. Chlingaryan et al. (2018) presented a review of machine learning approaches used for crop yield prediction. The study found out that M5 – prime regression trees are a suitable tool for crop yield prediction. Jambekar et al. (2018) showed that regression analysis can be successfully used to predict production of rice, wheat and maize with accuracy. Costa et al. (2019) made use of SVM and neural network to classify Merlot wines according to their geographical origin based on the chemical properties of the wine. Trajanov et al. (2019) applied data mining techniques to study the effect of management and soil fertility parameters on crop yield in Austria.

From the above study, it is clear that data mining techniques have been applied to analyse various agricultural datasets. Researchers all over the world are applying various data mining techniques to predict yield of different crops. There is a need to investigate the data mining process in a comprehensive manner for the prediction of crop yield.

3 Data mining approach

Data mining refers to extraction of knowledge, i.e., mining of knowledge from large amounts of data (Han et al., 2011). Data mining and KDD are often considered as one and the same thing, whereas it is actually only a part of KDD (Fayyad et al., 1996) as shown in Figure 1. “KDD is the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data” (Fayyad et al., 1996). Data is a set of facts, and pattern is an expression or model that is applicable to the data or its subset. The nontrivial nature of process means that it is not a simple computational process rather it involves some degree of search and inference.

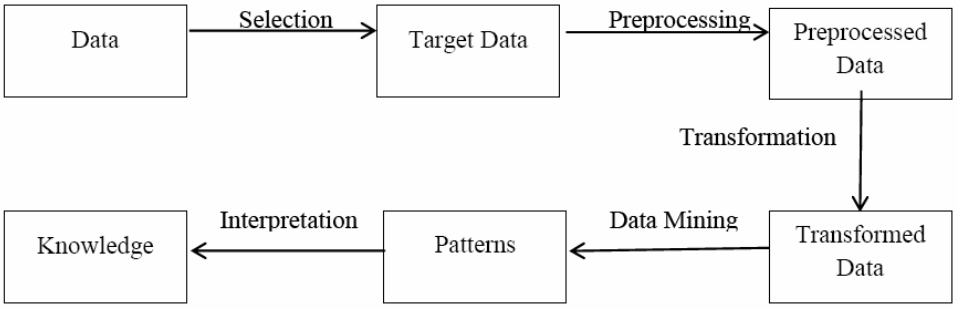
KDD consists of an iterative sequence of the following steps:

- *Data cleaning:* It is the process of removal of noise and inconsistent data. The original data may be inconsistent or it may contain missing values. In this step, the erroneous data is either corrected or removed whereas missing data is generated using various data mining tools.
- *Data integration:* In this course of action, varied multiple data sources are integrated in a common source. The data collected for the process may be from different and

heterogeneous data sources; it may be obtained from different types of databases and other non-electronic sources. The entire collected data from varied sources is, thus, integrated.

- *Data selection:* In this step, the data which is relevant to the analysis task is retrieved from the database.
- *Data transformation:* Before data mining is performed, the transformation or consolidation of data into appropriate forms may be necessary. This is done by performing different aggregate or summary functions. An example of data transformation is changing of continuous values of an attribute into discrete values. Another example is replacement of current date and date of birth by age.
- *Data mining:* Once the data has been cleaned, integrated, selected and transformed, it is ready for data mining process. In this step, different algorithms and techniques are applied for the extraction of useful patterns and to induce new knowledge by studying the relationships among the values of different attributes of the dataset under study.
- *Pattern evaluation:* This step identifies the truly interesting patterns representing knowledge based on some interestingness measures. A pattern is interesting if it has the following characteristics:
  a It is easily understandable.
  b The pattern is valid on new or test data with some degree of certainty.
  c It is potentially useful.
  d The new pattern is novel (Han et al., 2011).
- *Knowledge representation:* The final phase of KDD process is knowledge representation. In this phase, the knowledge that has been mined from data, or the interesting patterns that have been found are presented to the user in a form in which they are understandable. Different visualisation techniques can be used to summarise data and to present complex results in a better way than mathematical or textual representation.

Figure 1 Knowledge discovery in databases



4 Data mining tasks

As discussed earlier, data mining finds its application in various fields, e.g., in market-basket analysis; it can be used to find which products are often bought together and hence help in arrangement of the products on the shelves. In marketing, data mining can be used to target the customers who are likely to buy a particular product. In agriculture, data mining can be used to predict crop yield. These are the business or research goals of data mining. These goals can be achieved by performing one or more data mining tasks. Data mining tasks are the technical activities that can be carried independently of any particular business or research goal (Linoff and Berry, 2017). Data mining tasks can be broadly characterised into descriptive data mining tasks and predictive data mining tasks.

4.1 Descriptive data mining tasks

Descriptive data mining tasks are used to describe the properties of data in a target dataset. They describe trends and patterns in data. Such descriptions usually give possible explanations for these trends and patterns. Descriptive data mining tasks consist of data characterisation, link analysis and clustering.

4.1.1 Data characterisation

Data characterisation is done to summarise the general characteristics of a target class of data (Han et al., 2011). For numerical data, characterisation can be done by finding averages, correlation of variables, standard deviation, etc. The visualisation tools such as pie charts and bar charts are used to represent the data characterisation output.

4.1.2 Link analysis

Link analysis helps to identify relationships among values of different attributes in a database through descriptive approach. Link analysis explores data through association discovery approach and sequence discovery approach. Association discovery identifies the special type of data associations. It does the job of finding which attributes 'go together' (Larose, 2014). Association analysis discovers the association rules. It analyses the frequency of concurrence of the items in transactional databases (Zaïane, 1999), for example, in case of market-basket analysis, it finds the items that are frequently bought together. Sequential analysis studies the value link relation to their association over a sequence of time, that is, it is used to determine the sequential patterns in data, for example, most people who buy computers are found to purchase antivirus software within one week. So, there is a sequence followed in the purchase – the purchase of a computer is followed by purchase of antivirus software.

4.1.3 Clustering

Clustering divides database objects or records into different groups. The objects are clustered based on the principle of maximising the intra group similarity and minimising the inter group similarity based on a criteria defined on the attributes of the objects (Han

et al., 2011). It means the formation of clusters of objects having high similarity with one another. These objects are quite dissimilar to the objects in other clusters.

4.2 Predictive data mining tasks

The predictive data mining tasks perform Induction on the given dataset so as to make predictions. These tasks allow the prediction of the value of target attribute of an object based on the observed values of some other attributes of that object. The two most common predictive data mining tasks are classification and regression.

4.2.1 Classification

Classification, like clustering, maps objects into groups, but in this case, the groups are predefined. Classification algorithms define the class on the basis of the attribute values of the data; the description of these classes is usually according to the characteristics of data or the already classified objects (Dunham, 2012). Classification task, then, consists of examining the features of a newly presented object and assigning a predefined class to it.

4.2.2 Regression

Regression, like classification, is a predictive data mining task, which predicts the value of target attribute of an object, based on value of one or more predictors, that is, based on the attributes of the object whose value is known. Regression differs from classification; the target variable for regression tasks, is continuous unlike classification where the target variable, generally has few discrete values such as 'high' and 'low' (Weiss and Davison, 2010).

5 Building models

Data mining task aims at the production of new knowledge on which user can act, for this a model is built, based on the collected data. A model is a high level global description of the dataset (Hand et al., 2001). It explains how something works and brings forth its clear and real picture so that it can be used for making inferences in real world (Linoff and Berry, 2017).

Model building in data mining is data driven (Hand et al., 2001). The model is built from the data when it is analysed in terms of its properties (attribute values). The goal is to find patterns that hold for some part of the database or to find certain rules that are applicable to the whole or part of the database.

A model can be descriptive or predictive. A descriptive model highlights the main features of the data in a convenient form. It summarises the data and allows the study of the vital features of the data without any concern about its size. A predictive model, on the other hand, predicts the value of a target characteristic (attribute) of an object, based on the observed values of the other characteristics of the object (Hand et al., 2001).

6 Data mining techniques

For performing a particular data mining task, a model has to be built. A model can be built by using a particular data mining technique. Being a multi-disciplinary field, data mining adopted its techniques from many research areas such as statistics, machine learning, pattern recognition, databases, visualisation, etc. A data mining technique can be categorised as unsupervised or supervised. Unsupervised learning techniques are used to build descriptive models. In these techniques, there is no target variable. Instead, the algorithm searches for patterns and structures among all the variables. This means that the data provided in the training set to build the model is not labelled. K-means and association rules are the examples of unsupervised learning techniques. Supervised learning techniques are used to build predictive models which aim at finding a relationship of input attributes to a target attribute (Rokach and Maimon, 2010b). In the case of supervised learning techniques, the data provided in the training set is labelled, which means that the training dataset contains the pre-classified values of target variable in addition to the input attributes. A model is then built by analysing the relationship between the input attributes and the target attribute. This model is then used to predict the values of target attributes for the objects where the value of target attribute is not known. Regression, Naïve Bayes, decision tree induction, rule set induction, neural networks are some of the important supervised learning techniques.

6.1 Statistical techniques

The statisticians started extracting knowledge from data much before the advent of artificial intelligence. Statistical tools are applied in correlation analysis to find correlation between two variables. Similarly, some other statistical concepts such as determining a data distribution, calculating a mean or variance can be viewed as data mining techniques for extracting data under consideration (Dunham, 2012). Each of these is a statistical model in its own right. Some other statistical techniques used in data mining are correlation analysis, regression analysis and Naïve Bayes classification.

6.1.1 Measuring central tendency: mean, median and mode

The measures of central tendency measure the location of the middle or centre of data distribution. Suppose, there is a numerical attribute X, e.g., age, which has been recorded for a set of objects. Let $x_1, x_2, ..., x_n$ be the set of n observed values for X. The *mean* ($\overline{x}$) is defined as the average value of all the observations

$$\overline{x} = \frac{\sum_{i=1}^{n} x_i}{n} = \frac{x_1 + x_2 + \ldots + x_n}{n} \tag{1}$$

Median (M) is the data item which lies in the middle of the dataset. To find the median, first the dataset is set in an order. When the dataset is arranged in an ascending or a descending order, median is the observation or value that divides the dataset into two equal parts – the value of half of the data items is less than the median and the value of the remaining half is greater than the median.

$$M = \left(\frac{n+1}{2}\right)^{\text{th}} \text{ observation, when } n \text{ is odd;} \tag{2}$$

$$M = \frac{1}{2}\left[\left(\frac{n}{2}\right)^{\text{th}} \text{ observation} + \left(\frac{n}{2}+1\right)^{\text{th}} \text{ observation}\right], \text{ when } n \text{ is even} \tag{3}$$

Mode (Z) is the value that occurs most frequently in the dataset. It is possible that a dataset may have more than one mode (when 2 or more values occur for the same number of times) or it may have no mode at all (when all the values occur only once).

6.1.2 Measuring dispersion

Dispersion is the measure of scatteredness or spread of the observations in the dataset (Kothari and Garg, 2014). Range, mean deviation and standard deviation are some of the important measures of dispersion. *Range* (R) is the difference between the extreme values of the items in a dataset, that is, it is the difference between the largest value and the smallest value in a dataset.

$$\begin{aligned} R = &(\textit{Highest value of an item in dataset}) \\ &-(\textit{Lowest value of an item in dataset}) \end{aligned} \tag{4}$$

Mean deviation is the average difference of the observed values of an attribute from some central tendency such as mean, median or mode. While calculating the mean deviation, only the absolute values of the deviations are considered and the negative sign of deviations is ignored.

$$\textit{Mean deviation from mean}\left(\delta_{\bar{x}}\right) = \frac{\sum |x_i - \bar{x}|}{n} \tag{5}$$

$$\textit{Mean deviation from median}\left(\delta_M\right) = \frac{\sum |x_i - M|}{n} \tag{6}$$

$$\textit{Mean deviation from mode}\left(\delta_Z\right) = \frac{\sum |x_i - Z|}{n} \tag{7}$$

Another method of calculating dispersion is variance. *Variance* is defined as the average of squares of deviations, when such deviations for the individual observations of the attributes are obtained from the arithmetic average.

$$\textit{Variance} = \sigma^2 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n} \tag{8}$$

Standard deviation is defined as square root of variance.

$$\textit{Stddev} = \sigma = \sqrt{\frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n}} \tag{9}$$

6.1.3 Correlation analysis

So far, a univariate population, that is datasets which have only one attribute, is considered. However, the datasets normally contain multiple attributes. Such population is called multivariate population. When the datasets contain two or more attributes, there may be a need to study a relationship between the attributes (variables). The measurement of relationships between two variables can give an idea of the effect of one variable on the other (Kothari and Garg, 2014). Suppose a particular dataset has numerical attributes – X and Y, which have been recorded for a set of objects. Let x_1, x_2, …, x_n be the set of n observed values for X and let y_1, y_2, …, y_n be the set of n corresponding observed values for Y, then the covariance between X and Y is

$$Covariance = \sigma_{XY} = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{n} \tag{10}$$

Here $\bar{x}$ is the mean of observations on X and $\bar{y}$ is the mean of observations on Y. Covariance can take any value from $-\infty$ to $+\infty$. A positive covariance indicates that the increase in value of one variable leads to increase in value of the other. In case of a negative covariance, with increase in value of one variable the value of other variable decreases. A covariance value of 0 indicates that there is no relationship between the variables. If the unit of measurement for observations of X and/or Y is changed, the value of covariance is changed, though there is no change in relationship between X and Y. So, a better measure of relationship between two variables is Karl Pearson's coefficient of correlation. Karl Pearson's coefficient of correlation (r) is defined as

$$r = \frac{Covariance(X, Y)}{Stddev(X) \cdot Stddev(Y)} = \frac{\sigma_{XY}}{\sigma_X \cdot \sigma_Y} \tag{11}$$

Karl Pearson's coefficient of correlation is not affected by units of measurement and takes values in [–1, +1]. There is no relationship between two variables if the value of r is 0. The relationship between two variables increases as the value of r approaches (–1, +1). A positive value of r indicates that the increase in value of one variable leads to increase in value of the other. In case of a negative value, with increase in value of one variable, the value of other variable decreases.

6.1.4 Regression

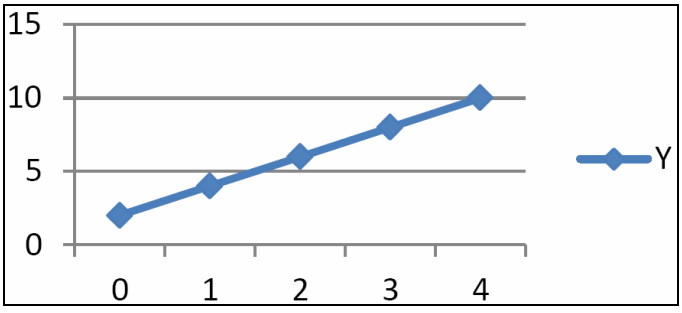
In regression, the existing values are used to predict the other values (Edelstein, 2005). A model is created by mapping the past values of the output variable to the values of input variables. Once a model is created, it is used to predict the value of output variable based on the value of input variables in such a way that lowest error occurs in making a prediction. There are a number of variations of regression – simple linear, multiple linear, nonlinear and logistic regression.

6.1.4.1 Simple linear regression

Simple linear regression contains one dependent variable and one independent variable. The simplest relationship between two variables – X and Y can be represented as

$$Y = a + bX \tag{12}$$

Figure 2 Simple linear regression (see online version for colours)



The values of a and b are chosen so that the error is minimum. To calculate the error, squared difference of the predicted and the actual value is found.

6.1.4.2 *Multiple linear regression*

In multiple linear regression, there is one dependent variable and a number of independent variables. If there is a dependent variable Y and three independent variables – X_1, X_2, X_3, then the equation will be of the form

$$Y = a + b_1(X_1) + b_2(X_2) + b_3(X_3) \tag{13}$$

Simple linear regression describes a line in two dimensional space whereas, multiple regression describes a line in (n + 1) dimensional space, where n is the number of independent variables.

6.1.4.3 *Nonlinear regression*

In nonlinear regression, the regression equation is formed by squaring, cubing or finding square root of the independent variables. A nonlinear regression model may look like

$$Y = a + b_1(X_1)^{1/2} + b_2(X_2)^2 + b_3(X_3)^3 \tag{14}$$

6.1.4.4 *Logistic regression*

Linear and nonlinear regression modules are well suited to estimating continuous quantities that can take on a wide range of values, but they are not suitable for modelling binary outcomes. The regression model which is used to handle binary outcomes is called logistic regression. In this model, there are only two categories such as yes/no, or good/bad, and the task is to assign each record to any one of the categories.

6.1.5 *Naïve Bayes classifiers*

Bayesian classifiers are statistical classifiers which can predict class membership probabilities, that is, given a tuple – T, which has to be classified, the Naïve Bayes method calculates the probability of T belonging to a particular class and then assigns T

to the class with the higher probability value. The probability is calculated from the given set of data as evidence or input. Naïve Bayes classifiers assume independence between different attribute values; it implies that the effect of attribute value on a given class is independent of the values of other attributes (Han et al., 2011). Suppose there is a training set of class-labelled tuples. Let there be m classes – C_1, C_2, …, C_m and A_1, A_2, …, A_n be the set of attributes of the dataset. Given a tuple $T = (t_1, t_2, \ldots, t_n)$, the probability of T belonging to C_1 is calculated and the probability of T belonging to C_2, C_3, …, C_m. T will be assigned to a class for which it has got higher probability value. According to Bayes theorem, probability is calculated as:

$$P(C_i|T) = \frac{P(T|C_i)P(C_i)}{P(T)} \tag{15}$$

where

C_i i^{th} class

$P(C_i|T)$ probability of C_i to hold for tuple T

$P(C_i)$ the prior probability of C_i.

This is calculated from the training dataset as

$$\frac{No.\ of\ tuple\ which\ belong\ to\ C_i}{Total\ number\ of\ tuples} \tag{16}$$

$P(T|C_i)$ Prior probability of each attribute value of T to hold for C_i.

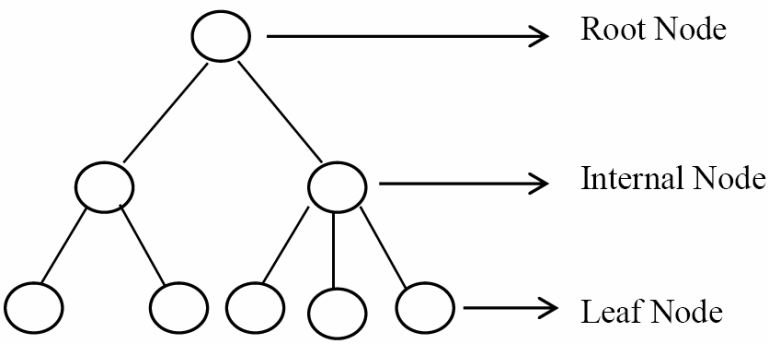
$$P(T|C_i) = \frac{Number\ of\ instances\ where\ A1 = t1\ and\ class = C_i}{Total\ number\ of\ instances\ where\ class = C_i} * \frac{Number\ of\ instances\ where\ A1 = t1\ and\ class = C_i}{Total\ number\ of\ instances\ where\ class = C_i} * \ldots \tag{17}$$

$P(T)$ is prior probability of T and is constant for all the classes. So, when a tuple T, is presented for classification, by Bayes method, then $P(C_i|T) = P(T|C_i) * P(C_i)$ for all classes – C_1, C_2, …, C_m. T is assigned to the class which has the highest value of $P(C_i|T)$.

6.2 Decision tree induction

Decision tree induction is the learning of decision trees from class-labelled training tuples (Han et al., 2011). A decision tree classifier recursively makes partitions between the instance spaces. As shown in Figure 3, a decision tree is a directed and rooted tree with a top most node called root node. This node has no incoming edges. All the other nodes have exactly one incoming edge. A node with one incoming edge and multiple outgoing edges is called an internal node or a test node. Each test node considers a single attribute of the dataset such that instance space is partitioned into two or more sub-spaces according to the value of the attribute (Rokach and Maimon, 2010a).

Figure 3 Decision tree



The nodes with no outgoing edges are the leaves. Each leaf represents a class. To classify an instance, navigate it starting from the root of the tree down towards a leaf. The leaf node reached at the end of the path indicates the class of the instance. The path to be followed is decided according to the outcome of tests along the path. Classification trees are the decision trees used for the prediction of categorical variables, whereas regression trees are the decision trees used for the prediction of continuous variables (Edelstein, 2005). The classification using decision trees is done in two steps:

1 Decision tree induction – Class labelled training dataset is used for the construction of a decision tree.

2 The class of an instance is determined by navigating the decision tree starting from its root.

The issues faced by most decision tree induction algorithms are – choosing of splitting attributes, ordering of splitting attributes and the number of splits to make. These issues are handled differently by different algorithms. The ID3 algorithm for decision tree induction uses information gain as the criteria for choosing the order of splitting attributes. Entropy is a concept used to quantify information. It is defined as the amount of uncertainty or surprise or randomness in a set of data (Dunham, 2012). Entropy is calculated as:

$$Entropy = \sum_{i=1}^{m} p_i \log\left(1/p_i\right) \tag{18}$$

where p_i is the probability that a tuple in the dataset belongs to class C_i.

The entropy before the split and the entropy after the split are measured, and their difference gives the information gain.

$$Information\ gain = Entropy\ before\ the\ split - Entropy\ after\ the\ split \tag{19}$$

The attribute with the maximum information gain is chosen for split.

6.3 Rule set induction

Rule Induction is a method for deriving a set of rules to classify cases (Edelstein, 2005). A rule is represented by an expression of the type – "if this and this and this, then this"

(Berson and Smith, 2018). Rule-based classifiers generate rules from a class labelled training dataset which is supplied to them. The rules are extracted using a sequential covering algorithm. The learning process progresses by learning one rule at a time and removing the tuples covered by the rule. This process is repeated on all the tuples (Han et al., 2011). A classification rule is of the form – if condition, then conclusion; the 'if' part of a rule is known as rule antecedent, and the 'then' part as a rule consequent. The rule antecedent consists of one or more tests applied on attribute values as:

$$\text{If } attribute_1 = value_1 \text{ and } attribute_2 = value_2 \text{ and } \ldots\ldots \tag{20}$$

The consequent part of the rule contains the class prediction.

If all the attribute tests in a rule antecedent hold for a given tuple, it implies that the rule antecedent is satisfied and that the rule covers the tuple. Coverage of a rule (R) is defined as:

$$Coverage\,(R) = \frac{Number\ of\ tuples\ cevered\ by\ R}{Total\ number\ of\ tuples\ in\ the\ training\ dataset} \tag{21}$$

Accuracy of a rule (R) is defined as:

$$Accuracy\,(R) = \frac{Number\ of\ tuples\ correctly\ classified\ by\ R}{Number\ of\ tuples\ covered\ by\ R} \tag{22}$$

Given a tuple T to be classified, if R_1 is the only rule satisfied then the rule fires by returning the class prediction for T. But there may be cases where more than one rule is satisfied. In such a case, there is a conflict as to which rule should fire, thus a conflict resolution strategy is used. Normally, an ordering is given to the rules, and in case of a conflict, the higher order rule will fire. The ordering can be class-based or rule-based. When class-based ordering is used, the classes are arranged in decreasing order of prevalence, that is, the rules for most commonly occurring class come first, then the second and so on. In case of rule-based ordering, the rules are arranged as per the measure of rule quality, that is size or accuracy. Size ordering ascribes the highest priority to the rule that has the highest antecedent size, that is, the rule which covers the maximum number of attributes. Accuracy ordering gives highest priority to the rule which has the highest percentage of accuracy.

Rules can also be extracted from a decision tree. To extract rules from a decision tree, start from the root node, each path from the root node to a leaf node will give a rule. The conditions at various nodes along the path except the leaf node are ANDed and form the rule antecedent. The leaf node holds the class prediction forming the rule consequent. The rules mined from decision trees are different from those extracted directly from the dataset, using sequential covering algorithm. The rules extracted from decision trees are non-overlapping. Also, the rule spans the entire instance space, which implies that each possible combination of attribute values will be covered by exactly one rule (Furnkranz et al., 2012).

6.4 Association rules

Association rules are used to find relationships between the data items in a set of transactions. While the rules generated by decision tree or rule set induction are used to

build a model which is used for classifying instances, the goal of association rules is data analysis. Association rule learning is an unsupervised learning method which is used to discover patterns inside data. A database in which association rules have to be found can be viewed as a set of transactions (or tuples), where each transaction contains a set of items. An example transaction database is shown in Table 1.

Table 1 Transaction database

Transactions	*Items*
T1	Bread, butter, jam
T2	Bread, butter, milk
T3	Bread, eggs, juice
T4	Bread, butter
T5	Juice, milk

The above database contains a set of transactions $T = \{T_1, T_2, T_3, T_4, T_5\}$, a set of items, $I = \{$bread, butter, eggs, jam, juice, milk$\}$ and each transaction T_i contains a set of items. So given a set of items $I = \{I_1, I_2, \ldots, I_m\}$ and a set of transactions $T = \{T_1, T_2, \ldots, T_n\}$, an association rule is an implication of the form $X => Y$, where $X, Y \subset I$ and $X \cap Y = \emptyset$.

Association rules do not differentiate between the class attribute and any other attribute in the dataset. A rule may have a class attribute on either of its side or it may not contain a class attribute at all (Furnkranz et al., 2012). When the data in the database is analysed, there may be a vast number of possible rules but only the best are chosen. Support and confidence are the two measures of goodness of an association rule. Support is defined as the ratio of the number of transactions that contain all the items in an association rule to the total number of transactions in the database (Linoff and Berry, 2017). For an association rule, $X => Y$.

$$Support\ (X => Y) = \frac{Number\ of\ transactions\ containing\ both\ X\ and\ Y}{Total\ number\ of\ transactions\ in\ the\ database} \tag{23}$$

Confidence measures the goodness of the rule in predicting the consequent by comparing how often the consequent appears when antecedent is true. For an association rule, $X => Y$.

$$Confidence\ (X => Y) = \frac{Support\ (X\ U\ Y)}{Support\ (X)} \tag{24}$$

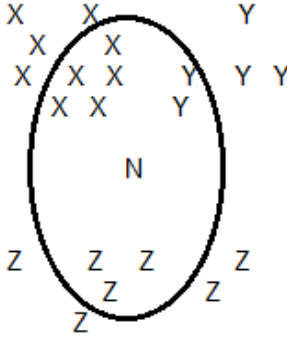
The association rule problem is to identify all association rules with a minimum support and confidence.

6.5 k-nearest neighbour

Nearest neighbour is an *instance-based* classification technique which is quite different from the other classification methods, as in this technique no explicit model is ever built (Weiss and Davison, 2010). There is no training phase and the entire work of making the prediction is done when the instance is presented. In k-nearest neighbour technique, the class-labelled training set is provided and the classified training set, itself becomes the model for further classification. When a new object (instance) is presented for

classification, its distance from each object in the training set is determined. Only the k-closest entries in the training set are considered. The new object is placed in the class that contains most of the k-closest objects. As shown in Figure 4, the training dataset contains objects belonging to three classes – X, Y and Z. When a new instance (N) is presented for classification, its distance from each object in the training set is determined. In this case, let the value of k be 10. Only the ten closest entries in the training set are considered. The new object is placed in the class that contains most of the ten-closest objects. As seen in Figures 4, 5 of the closest objects belong to class-X, 2 to class-Y and 3 to class-Z. As class-X contains the maximum closest entries, the new object N is assigned to X.

Figure 4 k-nearest neighbours



6.6 k-means

k-means is an iterative partitioning technique of dividing given instances into groups or clusters. Partitioning techniques work by moving instances from one cluster to another starting from an initial partitioning. k-means assumes that each instance is represented by one numeric instead of a tuple with many attribute values. The value of k which is the number of clusters to be formed has to be specified by the user. The technique starts by randomly choosing k cluster centres (Rokach, 2010). All the instances are assigned to their closest cluster centre according to the Euclidean distance between the two, and hence the whole dataset is divided into k clusters – C_1, C_2, …, C_k (Witten et al., 2016). Then the centre of each such cluster is re-calculated by finding the mean of all the instances belonging to that cluster.

$$M_k = \frac{1}{N_k} \sum_{i=1}^{N_k} x_i \tag{25}$$

where M_k is the mean of k^{th} cluster, N_k is the number of instances belonging to cluster k and x_i is the value of i^{th} instance of cluster k.

These means are taken to be new centre values for their respective clusters. The whole process is repeated with the new cluster centres. The iterations are continued until the same points are obtained as cluster centres in consecutive rounds. At this stage the cluster centres have stabilised and will remain the same forever.

6.7 Artificial neural networks

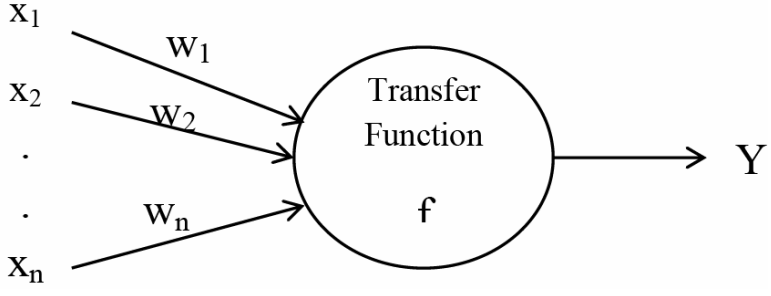
Neural networks are information processing computing models that are used to perform a number of data mining tasks such as prediction, classification and clustering. They are modelled on the working of human brain and consist of a graph representing the processing system as well as various algorithms that access the graph. The nodes in the graph are like individual neurons while the arcs are their interconnections. A neural network model consists of three parts:

1 A graph that defines the structure of neural network.

2 A learning algorithm that indicates how learning takes place.

3 Recall techniques that indicate how information is obtained from the network (Dunham, 2012).

Figure 5 Perceptron



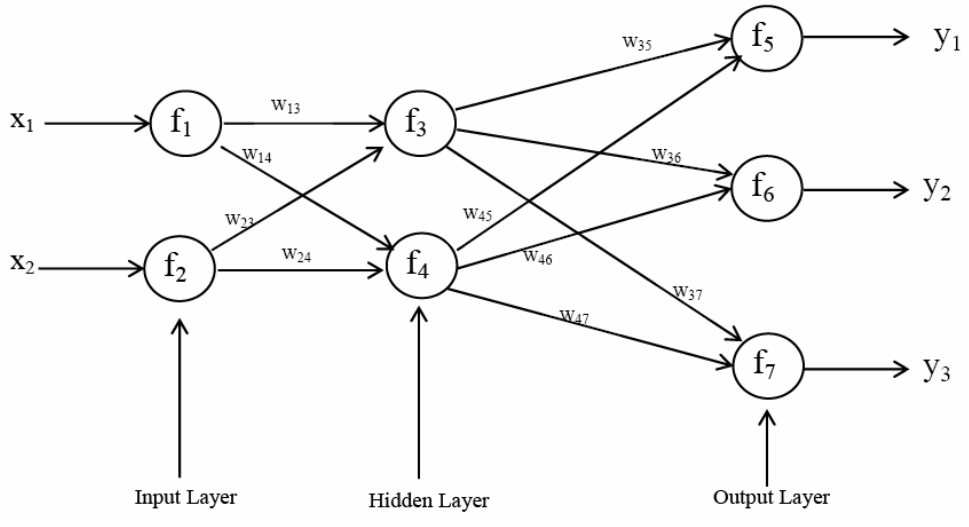
The simplest type of neural network is a perceptron. It consists of a single computing unit (also called a neuron or node), a single output and multiple weighted inputs. The output of the perceptron is given as:

$$Y = f\left(w_1 x_1 + w_2 x_2 + \ldots + w_n x_n\right) \tag{26}$$

$$Y = f\left(\sum_{i=1}^{n} w_i x_i\right) \tag{27}$$

A multilayer perceptron (MLP) is a network of perceptrons. It consists of number of nodes (or neurons) organised in layers. Each MLP consists of an input layer, zero or more hidden layers and an output layer. The arcs linking the neurons have weights associated with them (Weiss and Davison, 2010). The data mining task is performed by inputting a tuple through the input nodes, and the output nodes determine the prediction. The neural network has one input node for each attribute value of the tuple to be examined. The input layer nodes are passive and are simply used to receive the data and pass them to the next layer nodes. Although, MLP may have multiple hidden layers, yet most applications use a single hidden layer. The hidden layer nodes are connected to both input and output layer nodes. The nodes in one layer are fully connected to all the nodes in the adjacent layer. Each node processes information by first combining the input and weights to form a weighted sum and then performing a transformation function on that weighted sum.

Figure 6 Multilayer perceptron



The arc weights may initially be determined by some domain expert. The process of training the neural network (or the learning phase) refers to the adjustments of arc weights to achieve the best results. Initially, the neural network is fed with training data which consists of a set of input values and their desired outputs. For each tuple in the training dataset, the weighted sum of input values is calculated at each hidden layer node and after applying the transfer function, it is outputted by the hidden node and becomes input to the output layer nodes. The network output values are calculated and compared with the desired output. The network weights are modified so that the network produces a better approximation of the output. The process is repeated for each row in the training set. Such a pass through all rows in the training dataset is called an epoch. The process is repeated until difference between the output value calculated by the neural network and the known target values is as small as possible. To calculate the difference between the expected and actual output some overall error measure such as mean squared error (MSE) is calculated:

$$MSE = \frac{1}{M} \cdot \frac{1}{N} \sum_{m=1}^{M} \sum_{n=1}^{N} \left(d_{mj} - y_{mj} \right)^2 \tag{28}$$

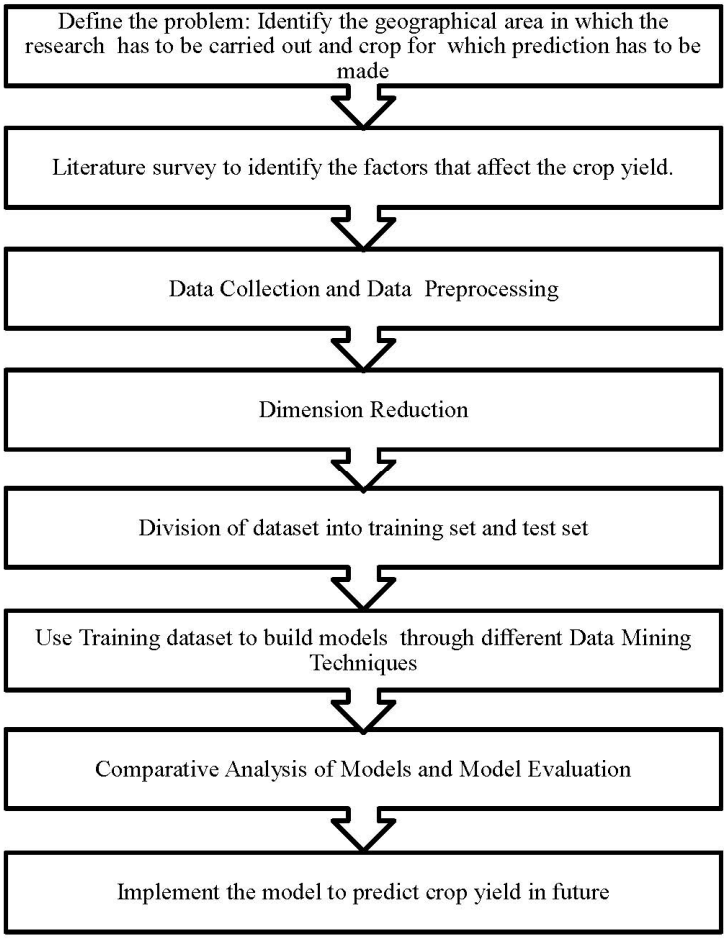
where d_{mj} and y_{mj} represent the desired value and the network output value at the m^{th} node for the j^{th} training pattern. M is the number of output nodes and N is the number of training tuples (Zhang, 2010).

7 Data mining process for prediction of crop yield

As discussed in Subsection 4.2, the predictive data mining tasks are used to predict the value of some target attribute of an object on the basis of observed values of some other attributes of that object. The problem of crop yield prediction falls in this category, as the crop yield is predicted by observing certain factors such as temperature, rainfall,

fertilisation, etc. So either regression or classification is used to predict yield depending on whether prediction has to be made for a continuous value or yield class such as high and low. This can be done by building a model using a supervised learning technique. Figure 7 lists the sequence of steps to be followed while using the data mining process for crop yield prediction.

Figure 7 Data mining process for prediction of crop yield



1 *Define the problem:* The first and foremost step is to define the research problem. The geographical area in which the research has to be carried out and the crop for which the prediction is to be made is identified in this step.

2 *Literature survey to identify the factors:* A literature survey has to be carried out to study the research done in this field till now. A survey should also be carried out to study the morphological and phenological stages of the crop under investigation. This helps in better understanding of the factors that affect the crop at various stages of its growth. At the end of this phase, the list of factors which affect the crop yield is identified. According to the need of the research, the type of factors, that is meteorological or management or soil factors for which the yield variation has to be

studied are declared, e.g., Vagh and Xiao (2012b) studied effect of temperature on crop yield whereas (Trajanov et al., 2019) applied data mining techniques to study the effect of management and soil fertility parameters on crop yield.

3 *Data collection and data pre-processing:* Once a decision is made about the list of factors for which yield variations are to be studied, the next task is collection of data. To mine relationships between various factors and yield variations, historical data of the selected factors and crop yield is required. This data is collected from different government agencies, meteorological centres, on site experiments or sampling methods. After data collection, the data needs to be pre-processed. Pre-processing is the process of cleaning, integration and transformation of data. Data may be inconsistent or it may contain missing values; so the erroneous data is either corrected or removed whereas missing data is generated using various tools. The data collected for the process may be from different and heterogeneous data sources; it may be obtained from different types of databases and other non-electronic sources. The entire collected data from varied sources is, thus, integrated. Further data transformation may be required to transform or consolidate data into forms appropriate for mining, by performing different aggregate or summary functions.

4 *Dimension reduction:* Dimension reduction is the process of reduction of factors (or independent variables) in the dataset. All the variables of a high-dimensional dataset may not be of importance to understand the underlying phenomena of interest. So, it is desirable to reduce the number of independent variables and to remove those variables which do not substantially affect the dependent variable (Fodor, 2002). The original representation of the data may be redundant because of the following reasons:

- Many of the variables have a variation smaller than the measurement noise and thus are irrelevant.
- Many of the variables are correlated with each other (e.g., through linear combinations or other functional dependence), therefore a new set of uncorrelated variables should be found (Carreira-Perpin, 1997).

Different techniques such as feature selection (Liu et al., 2010), principal component analysis, forward selection method (Everingham et al., 2016), association rule mining (Kaur and Attwal, 2017), etc. can be used for reducing the dimensions of the dataset.

5 *Division of dataset into training set and test set:* For supervised learning problems, the performance of a technique is measured in terms of the error rate. The model predicts the class of each instance – if it is correct, it is counted as a success; if not, it is an error. The error rate is just the proportion of errors made over a whole set of instances, and it measures the overall performance of the model. The model should be evaluated on the future performance and not the past performance on old data. Therefore, it is always better to evaluate a model with a test set that is different from the training set. If the same training set is used as test set also, the results can be misleading. So given that there are enough instances, the dataset can be divided into training set and the test set. Both the training set and test set should have right proportion of each class value. If the dataset is small, then instead of using the holdout method explained above, the k-fold cross validation is used to build and

evaluate a particular classifier. In k-fold cross validation, the dataset is divided into k equal parts. Each part in turn is used for testing and the remainder $k - 1$ parts are used for training. The procedure is repeated k times, so that in the end every instance is used for testing exactly once. The learning procedure is repeated k times – each time holding one set for testing, and calculating error rate on the hold out set. The k error estimates are averaged to yield an overall error estimate (Witten et al., 2016).

6 *Building models using different data mining techniques:* The goal of a data mining process is to produce new knowledge on which user can act upon. This is done by building a model based on the data that has been collected. A decision has to be made as to which supervised techniques can be used to build the model. A model built by decision tree or rule set induction does a good job of explaining the relationships between the input and output variables whereas a model built by using techniques such as neural networks or regression do a poor job of explaining their behaviour. The techniques that fulfil the given criteria are used to build the model using training dataset.

7 *Comparative analysis of models and model evaluation:* A number of metrics have been defined to evaluate the performance of a model built for classification of data in Subsection 8.4. These metrics are used to evaluate the performance of different models that have been built in the previous step. The model that gives the best performance for the required metrics can be chosen to make future predictions about the crop yield based on the factors that have been retained after dimension reduction step.

8 *Implementing the model for prediction of crop yield:* The model that gives the best performance is stored and used for making future predictions of yield. The various independent variables form the model input and based on the values of these variables, the model makes a prediction of crop yield.

8 Comparative analysis of different data mining techniques

In this section, the data mining process defined in previous section to build a model is used for paddy yield prediction. As discussed in Section 7, crop prediction is a predictive data mining task and supervised learning techniques can be used to build a crop prediction model. In Section 6, several supervised learning techniques have been discussed namely – Naïve Bayes, decision tree induction, rule set induction, nearest neighbour and neural networks. Five different models are built using the above mentioned techniques and a comparative analysis is carried out using various metrics to find the best model. The data mining tool – WEKA is used for carrying out different data mining tasks.

8.1 Previous work

Previously, a work was carried out to study the effect of temperature and rainfall on Paddy yield using data mining in Ludhiana and Patiala districts of Punjab (Kaur and

Attwal, 2017). It was observed that the effect of temperature was maximum during the vegetative and grain filling and ripening phase. The rainfall had maximum effect during grain filling and ripening phase. So it was concluded that paddy yield can be classified as high or low by using three meteorological variables – temperature during vegetative phase (TV), temperature during grain filling and ripening phase (TG), and rainfall during grain filling and ripening phase (RG). In this study, these three variables are used to classify whether the paddy yield is low or high for a particular year.

8.2 Data and sources

The study was carried out in Ludhiana and Patiala districts of Punjab, India. Paddy is a Kharif crop sown in June and harvested in October. The study required time series data about paddy yield, and temperature and rainfall during the paddy growing season. The annual yield data was obtained from the Agriculture Department of Punjab while the temperature and rainfall data of the two districts was acquired from Indian Meteorological Department. Data was considered from year 1995 to 2015. For Ludhiana district, the data was available for all the years, while for Patiala district complete data of only ten years was available.

8.3 Methodology

The paddy growing season was divided into three phases – the vegetative phase (from germination to panicle initiation), the reproductive phase (from panicle initiation to heading) and the grain filling and ripening phase (from heading to maturity). Average maximum temperature and total rainfall for each of the stages were calculated for each year, separately for the districts of Ludhiana and Patiala. The temperature and rainfall data for all the years for a particular district were considered. A metric was developed to categorise the temperature and rainfall data as low, moderate or high for a particular stage for a particular year for a particular district. Similarly, a metric was developed to categorise yield data for a particular year for a particular district as high or low. As discussed in Subsection 8.1, three meteorological variables – temperature during vegetative phase (TV), temperature during grain filling and ripening phase (TG), and rainfall during grain filling and ripening phase (RG) are used to predict whether the yield is high or low. The datasets formed for Ludhiana and Patiala districts are in Table 2:

The data mining tool – WEKA is used to build the models and evaluate their performance. Usually, the dataset is divided into two parts – the training set and the test set. The training set is used to build the model and the test set is used to evaluate the model based on different metrics. But in this case, the dataset is small, so k-fold cross validation is used to build and evaluate a particular model. In k-fold cross validation the dataset is divided into k equal parts. Each part in turn is used for testing and the remainder $k - 1$ parts are used for training. The procedure is repeated k times, so that in the end every instance has been used for testing exactly once. The learning procedure is repeated k times – each time holding one set for testing, and calculating error rate on the hold out set. The k error estimates are averaged to yield an overall error estimate (Witten et al., 2016). In this study, the value of k is taken as 4.

Table 2 Paddy yield dataset

Year	*Patiala*	*TV*	*TG*	*RG*	*Yield*
1995	Ludhiana	Mod	Low	High	Low
1996	Ludhiana	Low	Low	High	Low
1997	Ludhiana	Mod	Mod	High	Low
1998	Ludhiana	Low	Mod	Low	Low
1999	Ludhiana	Mod	High	Low	Low
2000	Ludhiana	Low	High	Mod	Low
2001	Ludhiana	Low	High	Low	Low
2002	Ludhiana	High	Low	High	Low
2003	Ludhiana	Mod	Mod	High	Low
2004	Ludhiana	Mod	Mod	Low	High
2005	Ludhiana	High	Mod	Mod	High
2006	Ludhiana	Mod	Low	Low	High
2007	Ludhiana	Low	Mod	Low	High
2008	Ludhiana	Low	Low	Low	High
2009	Ludhiana	High	Low	Mod	High
2010	Ludhiana	Mod	Low	Mod	High
2011	Ludhiana	Low	Low	Mod	High
2012	Ludhiana	High	Low	Low	High
2013	Ludhiana	Low	Low	Low	High
2014	Ludhiana	High	Low	Mod	High
2015	Ludhiana	Mod	High	Low	High
2001	Patiala	Low	Mod	High	Low
2003	Patiala	Low	Low	High	Low
2004	Patiala	Mod	Low	Low	High
2005	Patiala	Mod	Mod	High	Low
2006	Patiala	Low	High	Low	Low
2007	Patiala	Mod	High	Low	High
2008	Patiala	Low	High	Low	High
2009	Patiala	High	Low	Mod	High
2011	Patiala	Low	Low	High	Low
2012	Patiala	High	Low	Mod	High

8.4 Metrics used

To understand the different metrics used to evaluate the performance of a classifier, first understanding of the confusion matrix is required. The size of the matrix depends on the number of classes in the dataset. For n classes, an $n * n$ matrix will be created. The rows depict the actual class to which the instances belong and columns depict the class predicted by the classifier. The number of correctly classified instances is given by the diagonal elements.

Table 3 Confusion matrix

		Predicted class		*Actual total*
		A	*B*	
Actual class	A	TP_A*	FP_B	$TP_A + FP_B$
	B	FP_A	TP_B*	$FP_A + TP_B$
	Predicted total	$TP_A + FP_A$	$FP_B + TP_B$	

Note: *correctly classified instances.

In Table 3, TP_A is the number of instances in the dataset that actually belong to class A and the classifier has also predicted their class as A. These are known as true positives for class A. FP_A is the number of instances in the dataset that have been predicted as belonging to class A but which actually belong to class B. These are known as false positives for class A. TP_B is the number of instances in the dataset that actually belong to class B and the classifier has also predicted their class as B. These are known as true positives for class B. FP_B is the number of instances in the dataset that have been predicted as belonging to class B but which actually belong to class A. These are known as false positives for class B. The metrics used for evaluation of classifiers are categorised into three types – threshold evaluation metrics (TEMs), numerical evaluation metrics (NEMs) and build time and size metrics (BTSMs).

Table 4 Classification metrics categorisation

Threshold evaluation metrics (TEMs)	*Numerical evaluation metrics (NEMs)*	*Build time and size metrics (BTSMs)*
Percent correct	Mean absolute error	Elapsed time training
True positive rate	Root mean square error	Serialised model size
False positive rate	Relative absolute error	
Precision	Root relative squared error	
Recall		
F measure		
Kappa statistic		

8.4.1 Threshold evaluation metrics

These metrics are based on a threshold and a qualitative understanding of error (Ferri et al., 2009). These metrics measure the performance of the classifier based on the predicted class of the instance.

- *Percent correct:* The basic and most commonly used metric for evaluating the classifier is percent correct, which gives the percentage of instances that have been correctly classified. Looking at the confusion matrix, it is the sum of diagonal elements, divided by the total number of elements.

$$Percent\ correct = \frac{TP_A + TP_B}{TP_A + TP_B + FP_A + FP_B} * 100 \tag{29}$$

$$Percent\ incorrect = 100 - Percent\ correct \tag{30}$$

- *True positive rate* – For a particular class, true positive rate is defined as the total number of instances that have been correctly classified, divided by the total number of instances actually belonging to that class.

$$True\ positive\ rate\ (A) = \frac{TP_A}{TP_A + FP_B} \tag{31}$$

- *False positive rate* – For a particular class, false positive rate is defined as the total number of instances of that class that have been incorrectly classified, divided by the total number of instances actually belonging to that class.

$$False\ positive\ rate\ (A) = \frac{FP_B}{TP_A + FP_B} \tag{32}$$

- *Precision* – Precision is the measure of exactness. It tells what percentage of instances that have been classified as belonging to a particular class, actually belong to that class. Precision for class A is defined as

$$Precision\ (A) = \frac{TP_A}{TP_A + FP_A} \tag{33}$$

- *Recall* – Recall is the measure of completeness (Han et al., 2011). It tells what percentage of instances of class A have been actually classified as belonging to class A. Recall for class A is defined as

$$Recall\ (A) = \frac{TP_A}{TP_A + FP_B} \tag{34}$$

- *F-measure* – F-measure is the harmonic mean of precision and recall. F-measure of class A is defined as

$$F - measure = \frac{2 * Precision_A * Recall_A}{Precision_A + Recall_A} \tag{35}$$

- *Kappa statistic (KS)* – KS is a measure of how well a classifier is performing as compared to random guessing (Ferri et al., 2009). Suppose $P(A)$ is the observed agreement of a classifier (actual accuracy of a classifier) and $P(E)$ is the probability that agreement is due to chance (expected accuracy), then KS is defined as

$$KS = \frac{P(A) - P(E)}{1 - P(E)} \tag{36}$$

8.4.2 Numeric evaluation methods

The above metrics measure the performance of the classifier based on the predicted class of the instance. But the errors are not simply present or absent, they come in different sizes (Witten et al., 2016). Using numeric evaluation methods, the error rate is evaluated by comparing the predicted and the actual values of instances in the test set. If a_1, a_2, …, a_n are the actual values and p_1, p_2, …, p_n are the predicted values of the n test instances, then the different numerical estimation methods are defined as follows:

- *Mean absolute error* – Mean absolute error is the average of the magnitude of individual errors without taking their sign into account (Witten et al., 2016).

$$\textit{Mean absolute error} = \frac{\sum_{i=1}^{n} |p_i - a_i|}{n} \tag{37}$$

- *Root mean squared error* – Root mean squared error is the most commonly used measure for numeric evaluation of errors. It is the quadratic version of mean absolute error and is defined as square root of the average of the square of individual errors.

$$\textit{Root mean square error} = \sqrt{\frac{\sum_{i=1}^{n} (p_i - a_i)^2}{n}} \tag{38}$$

- *Relative absolute error* – In some cases, the calculation of relative error values rather than absolute values is desired. The relative error is calculated by comparing the error with the error made by simple predictor. For a simple prediction, the predicted value is simply the mean of training dataset values. Thus, relative absolute error is found by dividing the total absolute error of the classifier by the total absolute error of the simple predictor (Witten et al., 2016).

$$\textit{Relative absolute error} = \frac{\sum_{i=1}^{n} |p_i - a_i|}{\sum_{i=1}^{n} |a_i - \bar{a}|} \tag{39}$$

  Here, $\bar{a}$ is the mean value of the training data.

- *Root relative squared error* – Relative squared error also uses the same kind of normalisation. Here instead of taking the absolute error, the squared error is found and is normalised by dividing with the squared error of the simple predictor. Root relative squared error is calculated by finding the square root of relative squared error.

$$\textit{Root relative square error} = \sqrt{\frac{\sum_{i=1}^{n} (p_i - a_i)^2}{\sum_{i=1}^{n} (a_i - \bar{a})^2}} \tag{40}$$

  Here, $\bar{a}$ is the mean value of the training data.

8.4.3 Build time and size metrics

These metrics do not measure the accuracy of a classifier. They are just used to give an estimate of the time required to build the model and the memory or space required to store the model.

- *Elapsed time training* – Elapsed time training is the time taken in seconds to train the classifier, that is, the time taken to build the model.
- *Serialised model size* – This gives the idea of the size of the model developed using the training dataset. The model can then be stored and used to classify the instances.

8.5 Results and discussion

WEKA experimenter interface is used to carry out the evaluation of the previously mentioned classification techniques. The experimenter interface in WEKA is used to evaluate different techniques using a particular dataset for a particular metric. The results of TEMs for different techniques are given in Table 5.

Table 5 TEMs for different techniques

Metric	*Naive Bayes*	*J48 (decision tree)*	*Decision table (rule set)*	*Ibk (nearest neighbour)*	*Multilayer perceptron (neural network)*
Percent correct	77.68#	83.93	83.93	81.25	84.38*
Weighted average true positive rate	0.78	0.84	0.84	0.81	0.84*
Weighted average false positive rate	0.24	0.17	0.17	0.2	0.14*
Weighted average precision	0.79#	0.89*	0.89*	0.82	0.86
Weighted average recall	0.78#	0.84*	0.84*	0.81	0.84*
Weighted average F-measure	0.77#	0.83	0.83	0.81	0.84*
Kappa statistic	0.54#	0.67	0.67	0.62	0.69*

Notes: *best performing technique for a particular metric.
#worst performing technique for a particular metric.

For the metrics such as true positive rate which are evaluated for each class, weighted average of all the classes in the dataset is displayed. From Table 5, it is clear that for this particular dataset, Naïve Bayes is the worst performer. MLP has the best performance for most of the metrics but the performance of J48 (decision tree) and decision table (rule set) is also comparable. So the confusion matrix of these three techniques is analysed.

From the confusion matrix in Table 6, it is observed that the number of correctly classified instances is same for all three techniques. The number of misclassified instances is 5 for all three techniques. In case of MLP, two instances belonging to low have been classified as high and 3 instances belonging to high have been classified as low. The class wise false positive rate is 0.176 (17.6%) for low and 0.143 (14.3%) for high which is quite balanced. For J48 and decision table, all five instances that were misclassified belonged to class low. The class wise false positive rate is 0.00 (0%) for low and 0.357 (35.7%) for high which is quite unbalanced. This implies that the classifiers are more biased towards class – high. So it may be concluded though the correctly classified instances in all the three techniques is same, the performance of MLP is better.

Table 7 shows slightly surprising results when compared with Table 6. The decision table classifier (rule-based technique) is one of the best performers when only threshold values are considered, but in this case it is clearly the worst performer. The MLP is apparently the best performer as the predicted values show minimum deviation from the actual values. The performance of J48 is also comparable to that of MLP.

Table 6 Confusion matrix for MLP, J48 and decision table

Multilayer perceptron		*Predicted class*	
		Low	*High*
Actual class	Low	12	2
	High	3	14
J48		*Predicted class*	
		Low	*High*
Actual class	Low	9	5
	High	0	17
Decision table		*Predicted class*	
		Low	*High*
Actual class	Low	9	5
	High	0	17

NEMs are based on probabilistic understanding of error. They measure the deviation from the true probability (Ferri et al., 2009). The error rate is evaluated by comparing the predicted and the actual values of instances in the test set. The results of classification evaluation metrics for different techniques are given in Table 7.

Table 7 NEMs for different techniques

Metric	*Naive Bayes*	*J48 (decision tree)*	*Decision table (rule set)*	*Ibk (nearest neighbour)*	*Multilayer perceptron (neural network)*
Mean absolute error	0.29	0.25	0.33#	0.25	0.21*
Root mean squared error	0.37	0.36	0.38#	0.38	0.35*
Relative absolute error	59.15	50.24	65.53#	50.24	41.18*
Root relative squared error	73.68	72.96	75.88	76.09#	69.33*

Notes: *best performing technique for a particular metric.
#worst performing technique for a particular metric.

Table 8 BTSMs for different techniques

Metric	*Naïve Bayes*	*J48 (decision tree)*	*Decision table (rule set)*	*Ibk (nearest neighbour)*	*Multilayer perceptron (neural network)*
Elapsed time training (in seconds)	0.01	0*	0.01	0*	0.09#
Serialised model size (in bytes)	2,665*	3,771	6,563	4,560.25	13,919#

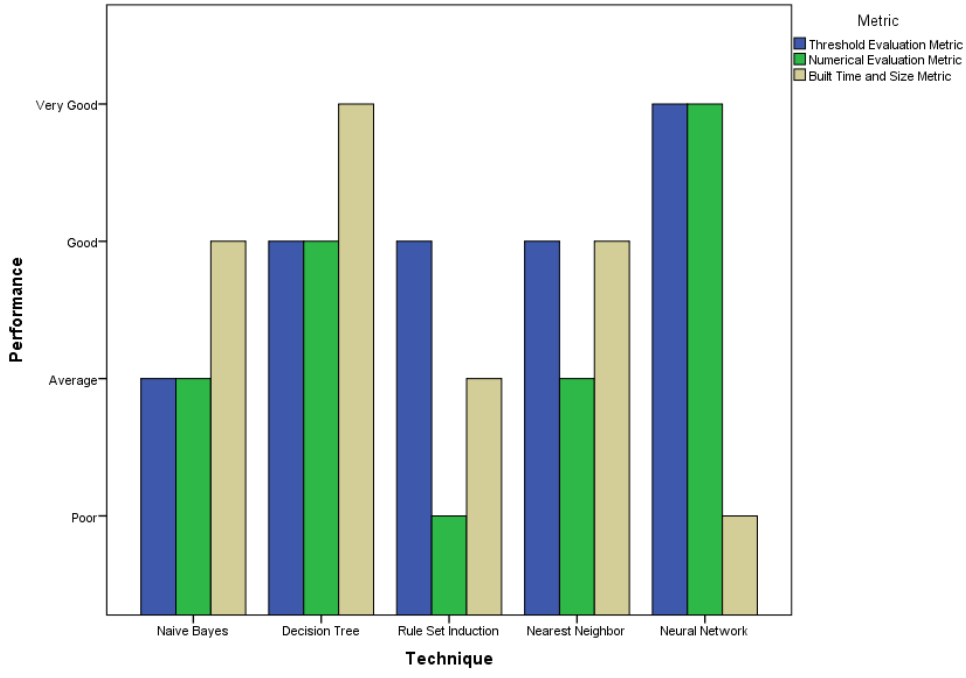
Notes: *best performing technique for a particular metric.
#worst performing technique for a particular metric.

BTSMs give measure of the time required to build a model by a particular classifier and the size of the model that is developed. Table 8 shows the time taken to build the model and size of the model developed for different techniques.

From Table 8 it is clear that MLP takes significantly more time to build the model and significantly more space is required to store the model. Though Naïve Bayes and decision table take much less time than MLP, it is significantly greater than J48 and Ibk classifiers. The smallest model size is that of Naïve Bayes followed by J48.

The performance of different techniques for different categories of metrics for paddy yield dataset is summarised in the graph in Figure 8.

Figure 8 Performance of various data mining techniques for different metrics (see online version for colours)



This is apparent from the analysis of the above graph that the neural networks technique gives the best performance for both TEM and NEM but it takes significantly more time than other techniques to build the model and requires a lot space to save the model. The decision tree induction technique also gives good performance for TEM but classification is biased towards high class; for NEM, it closely follows neural networks and the time and space requirements are considerably less than neural network technique. The performance of rule set induction is at par with decision tree induction for TEM but it gives an average performance for NEM. Naïve Bayes gives the worst performance for both TEM and NEM, though its time and space requirements are minimal.

9 Application of data mining process for paddy yield prediction

- *Problem definition* – The research work is to predict Paddy yield. The geographical areas of study are Ludhiana and Patiala districts of Punjab, India.

- *Literature survey* – An extensive literature survey is conducted to find out the factors that affect paddy yield. The findings of the survey are published in Kaur and Attwal (2016); the focus of the study was to analyse yield variations as a function of temperature and rainfall, so in the current paper only these factors are considered. The final list of selected factors are – temperature during vegetative phase, temperature during reproductive phase, temperature during grain filling and ripening phase, rainfall during vegetative phase, rainfall during reproductive phase, and rainfall during grain filling and ripening phase.
- *Data collection and pre-processing* – The annual yield data is obtained from the Agriculture Department of Punjab while the temperature and rainfall data of the two districts is obtained from Indian Meteorological Department. The collected data is from year 1995 to 2015. For Ludhiana district, the data is available for all the years, while for Patiala district complete data of only ten years is available. Data pre-processing is done to categorise the values of different independent and dependent variables and has already been explained in Subsection 8.3.
- *Dimension reduction* – During dimension reduction, the extraneous attributes (or variables) that do not significantly affect the paddy yield have been left out. Predictive apriori algorithm is used to analyse the effect of different factors on Paddy yield. The findings of the study are published in (Kaur and Attwal, 2017). Only three of the initially chosen six attributes are found to have impact on Paddy yield. These factors are – temperature during vegetative phase (TV), temperature during grain filling and ripening phase (TG), and rainfall during grain filling and ripening phase (RG).
- *Division of data into training and test set* – Normally, the dataset is divided into two parts – the training set and the test set. The model based on different metrics is built using the training set and is evaluated using the test set. But in this case, the dataset is small, so four-fold cross validation is used (already explained in Subsection 8.3) to build and evaluate model.
- *Use training dataset to build models through different data mining techniques* – In Section 6, five supervised learning techniques are discussed namely – Naïve Bayes, decision tree induction, rule set induction, nearest neighbour and neural networks. Five different models are built using the above mentioned techniques.
- *Evaluate the performance of the models based on different metrics using test dataset* – The performance evaluation of different models built using various supervised learning data mining techniques is carried out in Subsection 8.5, using the metrics defined in Subsection 8.4. It is found that the model developed using neural network technique gives the best performance.
- *Using the model for paddy yield prediction* – The model developed using neural network technique is stored and used for predicting Paddy yield by providing temperature during vegetative phase (TV), temperature during grain filling and ripening phase (TG), and rainfall during grain filling and ripening phase (RG) as input.

10 Conclusions

From the above study, it can be inferred that various data mining tasks and techniques can be used to interpret a vast amount of agricultural data available, to find interesting patterns in data and to extract knowledge from that data. The study investigated the different steps that have to be followed to predict a crop yield using data mining. The data mining process defined in the study was used to build a model for the Paddy yield prediction. Five models were built using five different supervised learning data mining techniques. The model built using neural network technique performed better for both TEM and NEM, but the time taken to build the model and the memory required to store the model is significantly larger than other models. So, if time and space is an issue, the model built using decision tree induction technique can be used whose performance in TEM is at par with neural networks and does not lag much behind in NEM.

References

Berson, A. and Smith, S.J. (2018) *Data Warehousing, Data Mining and OLAP*, McGraw Hill Education, Chennai.

Carreira-Perpin, M.A. (1997) *A Review of Dimension Reduction Techniques*, Department of Computer Science, University of Sheffield.

Chlingaryan, A., Sukkarieh, S. and Whelan, B. (2018) 'Machine learning approaches for crop yield prediction and nitrogen status estimation in precision agriculture: a review', *Computers and Electronics in Agriculture*, Vol. 151, pp.61–69.

Costa, N.L., Llobodanin, L.A., Castro, I.A. and Barbosa, R. (2019) 'Using support vector machines and neural networks to classify merlot wines from South America', *Information Processing in Agriculture*, Vol. 6, No. 2, pp.265–278.

Cruz, G.B., Gerardo, B.D. and Tanguilig, B.T. (2014) 'Agricultural crops classification models based on PCA-GA implementation in data mining', *International Journal of Modeling and Optimization*, Vol. 4, No. 5, pp.375–382.

Dunham, M.H. (2012) *Data Mining Introductory and Advanced Topics*, Pearson, New Delhi.

Edelstein, H.A. (2005) *Introduction to Data Mining and Knowledge Discovery*, Two Crows Corporation, Potomac, MD, USA.

Ekasingh, B., Ngamsomsuke, K., Letcher, R.A. and Spate, J. (2005) 'A data mining approach to simulating farmers' crop choices for integrated water resources management', *Journal of Environmental Management*, Vol. 77, No. 4, pp.315–325.

Everingham, Y., Sexton, J., Skocaj, D. and Inman-Bamber, G. (2016) 'Accurate prediction of sugacane yield using a random forest algorithm', *Agronomy for Sustainable Development*, Vol. 36, No. 2, p.27.

Farook, R.S., Aziz, A.H., Harun, A., Husin, Z., Shakaff, A.Y., Jaafar, M.N., et al. (2012) 'Data mining on climatic factors for Harumanis mango yield prediction', *Third International Conference on Intelligent Systems Modelling and Simulation*, IEEE, pp.115–119.

Fayyad, U., Piatetsky-Shapiro, G. and Smyth, P. (1996) 'From data mining to knowledge discovery in databases', *AI Magazine*, Vol. 17, No. 3, pp.37–54.

Ferri, C., Hernandez-Orallo, J. and Modroiu, R. (2009) 'An experimental comparison of performance measures for classification', *Pattern Recognition Letters*, Vol. 30, No.1, pp.27–38.

Fodor, I.K. (2002) *A Survey of Dimension Reduction Techniques*, Lawrence Livermore National Lab, CA, USA.

Furnkranz, J., Gamberger, D. and Lavrac, N. (2012) *Foundations of Rule Learning*, Springer, Berlin, Heidelberg.

Ghosh, S., Biswas, S., Sarkar, D. and Sarkar, P.P. (2012) 'Soil data mining using decision tree classifier', *Computer Science and Engineering Research Journal*, Vol. 8, pp.27–31.

Haghverdi, A., Ghahraman, B., Leib, B.G., Pulido-Calvo, I., Kafi, M., Davary, K. et al. (2014) 'Deriving data mining and regression based water-salinity production', *Computers and Electronics in Agriculture*, Vol. 101, pp.68–75.

Han, J., Kamber, M. and Pei, J. (2011) *Data Mining Concepts and Techniques*, Morgan Kaufmann Publishers, Waltham, MA, USA.

Hand, D., Mannila, H. and Smyth, P. (2001) *Principles of Data Mining*, A Bradford Books The MIT Press, Cambridge, MA.

Jambekar, S., Nema, S. and Saquib, Z. (2018) 'Prediction of crop production in India using data mining techniques', *Fourth International Conference on Computing Communication Control and Automation*, IEEE, pp.1–5.

Kaur, K. and Attwal, K.P. (2017) 'Effect of temperature and rainfall on Paddy yield using data mining', *7th International Conference on Cloud Computing, Data Science & Engineering – Confluence*, IEEE, pp.506–511.

Kaur, K. and Attwal, K.S. (2016) 'Factors affecting Paddy yield at different growth stages', *International Journal of Advanced Technology in Engineering and Science*, Vol. 4, No. 5, pp.193–199.

Kothari, C.R. and Garg, G. (2014) *Research Methodology Methods and Techniques*, New Age International, New Delhi.

Landau, S., Mitchell, R., Barnett, V., Colls, J., Craigon, J. and Payne, R. (2000) 'A parsimonious multiple regression model of wheat yield response to the environment', *Agricultural and Forest Meteorology*, Vol. 101, No. 2–3, pp.151–166.

Larose, D.T. (2014) *Discovering Knowledge in Data: An Introduction to Data Mining*, John Wiley & Sons, Hoboken, New Jersey.

Leona, M.R. and Jalao, E.R. (2013) *A Prediction Model Framework for Crop Yield Prediction*, Asia Pacific Industrial Engineering and Management System, Cebu, Philippines.

Linoff, G.S. and Berry, M.J. (2017) *Data Mining Techniques*, Willey, New Delhi.

Liu, H., Motoda, H., Setiono, R. and Zhao, Z. (2010) 'Feature selection: an ever evolving frontier in data mining', in *Proceedings of The Fourth International Workshop on Feature Selection in Data Mining*, Hyderabad, India, 21 June, pp.4–13.

Majumdar, J., Naraseeyappa, S. and Ankalaki, S. (2017) 'Analysis of agriculture data using data mining techniques: application of big data', *Journal of Big Data*, Vol. 4, No. 1, p.20.

Rokach, L. (2010) 'A survey of clustering algorithms', in Rokach, L. and Maimon, O. (Eds.): *Data Mining and Knowledge Discovery Handbook*, pp.269–298, Springer, Heidelberg.

Rokach, L. and Maimon, O. (2010a) 'Classification trees', in Rokach, L. and Maimon, O. (Eds.): *Data Mining and Knowledge Discovery Handbook*, pp.148–174, Springer, Heidelberg.

Rokach, L. and Maimon, O. (2010b) 'Supervised learning', in Rokach, L. and Maimon, O. (Eds.): *Data Mining and Knowledge Discovery Handbook*, pp.133–147, Springer, Heidelberg.

Rub, G. (2009) 'Data mining of agricultural yield data: a comparison of regression models', *Advances in Data Mining. Applications and Theoretical Aspects*, pp.24–37, Springer, Berlin Heidelberg.

Ruβ, G., Kruse, R., Schneider, M. and Wagner, P. (2008) 'Data mining with neural networks for wheat yield prediction', in *Proceeding of Advances in Data Mining – Medical Applications, E-Commerce, Marketing and Theoretical Aspects*, 8th Industrial Conference, Leipzig, Germany, Springer-Verlag Berlin Heidelberg, 16–18 July, pp.47–56.

Trajanov, A., Spiegel, H., Debeljak, M. and Sandén, T. (2019) 'Using data mining techniques to model primary productivity from international long-term ecological research (ILTER) agricultural experiments in Austria', *Regional Environmental Change*, Vol. 19, No. 2, pp.325–337.

Vagh, Y. and Xiao, J. (2012a) 'A data mining perspective of the dual effect of rainfall and temperature on wheat yield', *International Journal of Computer and Communication Engineering*, Vol. 1, No. 4, pp.358–364.

Vagh, Y. and Xiao, J. (2012b) 'Mining temperature data for shire-level crop yield prediction', *International Conference on Machine Learning and Cybernetics*, IEEE, Xian, pp.77–83.

Weiss, G.M. and Davison, B.D. (2010) 'Data mining', in Bidgoli, H. (Ed.): *Handbook of Technology Management*, pp.542–555, John Wiley and Sons, Hoboken, NJ.

Witten, I.H., Frank, E. and Hall, M.A. (2016) *Data Mining: Practical Machine Learning Tools and Techniques*, Morgan Kaufmann, Burlington, MA, USA.

Zaïane, O.R. (1999) *Principles of Knowledge Discovery in Databases* [online] http://webdocs.cs.ualberta.ca/~zaiane/courses/cmput690/slides/ch0s.pdf (accessed 20 June 2019).

Zhang, G.P. (2010) 'Neural networks for data mining', in Rokach, L. and Maimon, O. (Eds.): *Data Mining and Knowledge Discovery Handbook*, pp.419–444, Springer, Heidelberg.