Journal of Theoretical and Applied Information Technology
15th December 2018. Vol.96. No 23

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

DATA MINING TECHNIQUES FOR DATABASE PREDICTION: STARTING POINT.

WAEL SAID[1], M.M. Hassan [2], AMIRA M. FAWZY [3]

[1]Computer Science Department, [2]Professor on Information System Department, [3]Information Systems Department.

Faculty of Computers & Information, Zagazig University, Egypt.

ABSTRACT

With more evolution in technology, the database increased and came more complex and huge. Also, the search process in these data became complicated and takes a lot of time to reach valuable information. So, we need to an intelligent tool to deal with this problem. Data mining techniques can a solution to this problem. It is point out excavating knowledge from massive amounts of data. Techniques of data mining used in many fields such as pattern recognition and recommender systems here used within a database to improve the search process. This research displays different data mining techniques (decision tree, association rule, a neural network, fuzzy set ...) and shows cases of data mining techniques combined with a database. This paper directed to help new researchers take an overview of data mining techniques that can use in database prediction to enhance the search process.

Keywords: *Data Mining, Prediction Methods, Classification, Cluster, Database.*

1. INTRODUCTION

Recently with improvement in information technology, data mining techniques became the suitable tool and achieved a great success in different fields. Before discussing these techniques, we need to understand what meant by data mining. As stated in [1-3], Data mining is an extraction of interesting patterns or knowledge from huge amount of data. Data mining affected on Multiple Disciplines such as database technology, machine learning, statistics, pattern recognition, recommender systems and other fields.

Max Bramer in 2007[1] described the basic process of data mining consists of three steps: data preparation, data analysis and evaluation process. We concentrate on data analysis process that contains data mining techniques, these techniques classified into clustering and classification or prediction. Classification build models (functions) that depict and distinguish classes or notions for future prediction with unknown or missing numerical values. We introduce basic techniques for data classification, such as how to build decision tree classifiers, Bayesian classifiers, Bayesian belief networks, and rule based classifiers. Back propagation (a neural network technique) is also discussed, in addition to support vector machines method. Classification based on association rule mining is explored. Other approaches to classification, such as *k*-nearest-neighbor classifiers, case-based reasoning, genetic algorithms, rough sets, and fuzzy logic techniques, are introduced. This paper directed to help new researchers take a brief overview about techniques of data mining that can use in various fields especially in database prediction.

The structure of this paper as following: section 2 briefly outlines the process of data mining, section 3 discuss the different techniques of data mining. Also, section 4 reviews some of cases that used these techniques in database. Finally, section 5 contains on the conclusion.

2. PROCESS OF DATA MINING

Before one attempts to extract useful knowledge from data, it is important to understand the overall approach. Simply knowing many algorithms used for data analysis is not sufficient for a successful data mining (DM) project. Finding a good model of the data, which at the same time is easy to understand, is at the heart of DM. to apply this, process of data mining consist a set of steps followed[3][4] , these steps described in Figure 1 and discussed with some details in following subparts.

2.1 Data

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The outcome of data mining heavily depends on the quality and quantity of available data. Data described by three issues: data types, data storage techniques, and amount and quality of the data. Data defined as a collection of objects and their attributes, where a property or characteristic of an object.

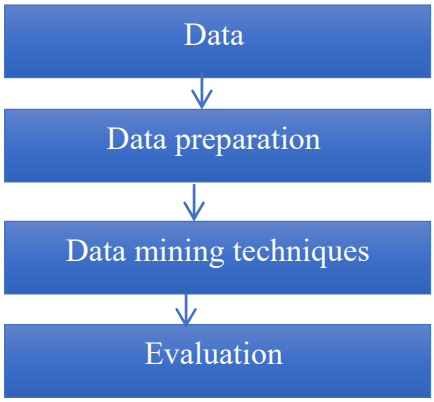

Figure1: Process of Data Mining

2.2 Data preparation

Data preprocessing techniques, when applied before mining, can substantially improve the overall quality of the patterns mined and/or the time required for the actual mining [3]. There are a number of data preprocessing techniques [4], [5]:

- ✓ *Data cleaning* can be applied to remove noise and correct inconsistencies in the data.
- ✓ *Data integration* merges data from multiple sources into a coherent data store, such as a data warehouse.
- ✓ *Data transformations*, such as normalization, may be applied. For example, normalization may improve the accuracy and efficiency of mining algorithms involving distance measurements.
- ✓ *Data reduction* can reduce the data size by aggregating, eliminating redundant features, or clustering, for instance.

2.3 Data analysis (modeling)

This process is considered the most important step in data mining processes because different methods of data mining described here. Modeling usually involves the use of several methods for the same DM problem type and calibration of their parameters to optimal values. This paper concentrate on a variety of data mining techniques, so, we deal with this point with more details in section 3.

2.4 Evaluation

Han and Kamber introduced in 2011[2], the criteria used to evaluate different models of data mining:

- ✓ *Accuracy:* The accuracy of a classifier refers to the ability of a given classifier to correctly predict the class label of new or previously unseen data. Similarly, the accuracy of a predictor refers to how well a given predictor can guess the value of the predicted attribute for new or previously unseen data.
- ✓ *Speed:* This refers to the computational costs involved in generating and using the given classifier or predictor.
- ✓ *Robustness*: This is the ability of the classifier or predictor to make correct predictions given noisy data or data with missing values.
- ✓ *Scalability*: This refers to the ability to construct the classifier or predictor efficiently given large amounts of data.
- ✓ *Interpretability:* This refers to the level of understanding and insight that is provided by the classifier or predictor. Interpretability is subjective and therefore more difficult to assess.

3 - DATA MINING TECHNIQUES

The following figure (*Figure 2*) overview different techniques of data mining with their correspondence to paper sections. As described, data analysis or modeling steps divided into supervised and unsupervised learning based on the method of learning (machine learning), and then classified into classification, prediction and clustering as the of type problems.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

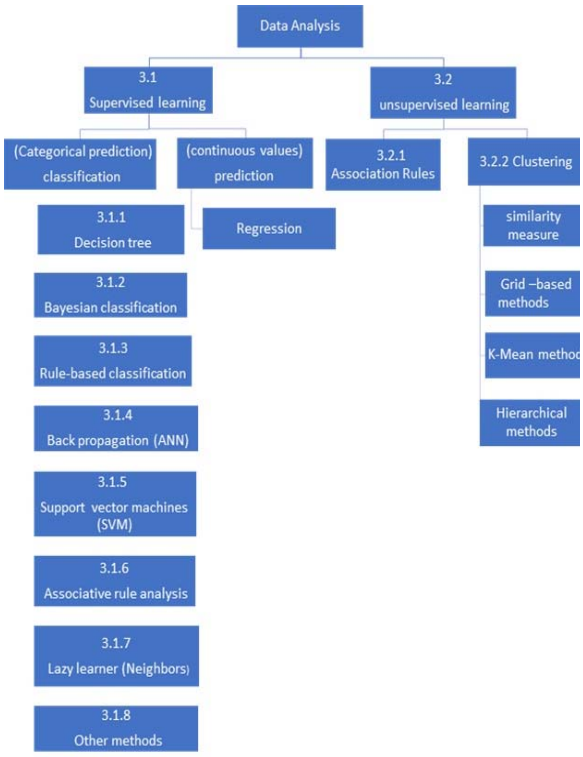

Figure 2: Different Techniques of Data Mining.

3.2 Supervised Learning

In supervised learning, finding a model that could predict the true underlying probability for each test case would be optimal [6]. Techniques for supervised learning can be further subdivided into *classification* and *regression* depending on the type of response variable (categorical or numerical) [7, 8, 9]. Next subsections cover different classification or categorical prediction methods.

3.2.1 Decision tree

Decision tree induction is the learning of decision trees from class-labeled training tuples. A decision tree is a flow chart-like tree structure, where each internal node(nonleaf node) denotes a test on an attribute, each branch represents an outcomeof the test, and each leaf node (or *terminal node*) holds a class label.

A researcher in machine learning developed a decision tree algorithms known as ID3, C4.5, and CART [2,10,11,12], these algorithms adopted a greedy approach in which decision trees are constructed in a top-down recursive divide-and-conquer manner. Most algorithms for decision tree induction also follow such a top-down approach, which starts with a training set of tuples and their associated class labels for more details in [2]. This algorthim used a heuristic procedure for selecting the attribute that "best" discriminates the given tuples according to class. This procedure employs an attribute selection measure, such as information gain,Gain-ratio and Gini index [13].

Information Gain

Information gain allowing multiway splits tree. ID3 uses information gain as its attribute selection measure. This measure is based on pioneering work by Claude Shannon on information theory, which studied the value or "information content" of messages. The expected information needed to classify a tuple in D is given by

$$Info(D) = -\sum_{i=1}^{m} pi \log_2(pi) \quad (1)$$

Where p_i is the probability that an arbitrary tuple in D belongs to class C_i and is estimated by $|C_{i,D}|/|D|$. A log function to the base 2 is used, because the information is encoded in bits. *Info(D)* is just the average amount of information needed to identify the class label of a tuple in *D*. *Info(D)* is also known as the entropy of *D*. to calculate the expected information after partitioning use,

$$info_A(D) = \sum_{j=1}^{v} \frac{|Dj|}{|D|} * Info(Dj) \quad (2)$$

The term $\frac{|Dj|}{|D|}$ acts as the weight of the *j*th partition. $Info_A(D)$ is the expected information required to classify a tuple from *D* based on the partitioning by *A* with distance *v*. The smaller the expected information (still) required, the greater the purity of the partitions. Information gain is defined as the difference between the original information requirements and the new requirement (i.e., obtained after partitioning on *A*). That is,

$$Gain(A) = Info(D) - info_A(D) \quad (3)$$

Gain ratio

The information gain measure is biased toward tests with many outcomes. C4.5, a successor of ID3, uses an extension to information gain known as *gain ratio*, which attempts to overcome this bias. It applies a kind of normalization to information gain using a "split information" value defined analogously with *Info*(*D*) as

$$SplitInfo_A(D) = -\sum_{j=1}^{v} \frac{|Dj|}{|D|} * \log_2\left(\frac{|Dj|}{|D|}\right) \quad (4)$$

This value represents the potential information generated by splitting the training data set, *D*, into *v* partitions, corresponding to the *v* outcomes of a test on attribute *A*. the gain ratio is defined as

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$GainRatio(A)=\frac{Gain(A)}{SplitInfo(A)} \quad (5)$$

The attribute with the maximum gain ratio is selected as the splitting attribute.

Gini index

The Gini index is used in CART [14]. Using the notation described above, the Gini index measures the impurity of *D*, a data partition or set of training tuples, as

$$\text{Gini(D)}=1-\sum_{i=1}^{m} p_i^2 \quad (6)$$

The Gini index considers a binary split for each attribute. We compute a weighted sum of the impurity of each resulting partition. For example, if a binary split on *A* partitions *D* into D_1 and D_2, the gini index of *D* given that partitioning is

$$Gini_A(\text{D})=\frac{|D1|}{|D|}\,Gini(D_1)+\frac{|D2|}{|D|}\,Gini(D_2) \quad (7)$$

The reduction in impurity that would be incurred by a binary split attribute A is

$$\Delta Gini(A)=Gini(D)-Gini_A(D) \quad (8)$$

The attribute that maximizes the reduction in impurity (or, equivalently, has the minimum Gini index) is selected as the splitting attribute.

3.2.2 Bayesian classification

Bayesian classifiers are statistical classifiers. Studies comparing classification algorithms have found a simple Bayesian classifier known as the *naïve Bayesian classifier* to be comparable in performance with decision tree and selected neural network classifiers. Bayesian classifiers have also exhibited high accuracy and speed when applied to large databases [2].

The Naive Bayes algorithm is a simple probabilistic classifier that calculates a set of probabilities by counting the frequency and combinations of values in a given data set. The algorithm uses Bayes theorem and assumes all attributes to be independent given the value of the class variable [15][16]. Bayes theorem provides a way of calculating posterior probability P(c|x) from P(c), P(x) and P(x|c). Look at the equation below:

$$p(c|x)=\frac{p(x|c)p(c)}{p(x)} \quad (9)$$

Above,

- $P(c|x)$ is the posterior probability of *class* (c, *target*) given *predictor* (x, *attributes*).
- $P(c)$ is the prior probability of *class*.
- $P(x|c)$ is the likelihood which is the probability of *predictor* given *class*.
- $P(x)$ is the prior probability of *predictor*.

Naïve Bayesian Classification algorithm work as follow [17]:

1: Let *D* be a training set of tuples and their associated class labels represented by an *n*-dimensional attribute vector, $\boldsymbol{X} = (x_1, x_2, .. , x_n)$.

2: Suppose that there are *m* classes, $C_1, C_2, .. , C_m$. Given a tuple, $\boldsymbol{X}$, the classifier will predict that $\boldsymbol{X}$ belongs to the class having the highest posterior probability, conditioned on $\boldsymbol{X}$. Create Likelihood table by finding the probabilities of $P(x|c)$.

3: Now, use Naive Bayesian equation (equation 9) to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of prediction.

The main benefits of Naive Bayes classifiers are that they are robust to isolated noise points and irrelevant attributes, and they handle missing values by ignoring the instance during probability estimate calculations [18]. But, limitation of Naive Bayes is the assumption of independent predictors. In real life, it is almost impossible that we get a set of predictors which are completely independent. To deal with this problems, there is another bayesian classification called *bayesian belief network*.

Bayesian belief networks are graphical models, which unlike naïve Bayesian classifiers, allow the representation of dependencies among subsets of attributes. Bayesian belief networks can also be used for classification.

Another solution for the problem of independent of naïve bayesian classifier is the attribute reduction. Attribute reduction is an effective way to improve the performance of this classification. liu and song in [19] take advantage of mixed Simulated Annealing and Genetic Algorithms to optimize attribute set.

3.2.3 Rule-based classification

Rules are a good way of representing information or bits of knowledge. A rule-based classifier uses a

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

set of IF-THEN rules for classification. An IF-THEN rule is an expression of the form "IF *condition* THEN *conclusion*". These rules can be generated by direct methods (from data set) or indirect methods. Direct methods such as *sequential covering algorithm* (AQ, CN2, RIPPER) . Indirect methods as PART and decision tree. The popular sequential covering algorithm is RIPPER [20], we next portray basic sequential covering algorithm.

Algorithm 1: Sequential covering. Learn a set of IF-THEN rules for classification.

Input:

- D, a data set class-labeled tuples;
- Att _vals, the set of all attributes and their possible values.

Output: A set of IF-THEN rules.

Method:

```
Rule set = fg; // initial set of rules learned is empty
for each class c do
  repeat
  Rule = Learn One Rule(D, Att vals, c);
  remove tuples covered by Rule from D;
  until terminating condition;
  Rule set = Rule set +Rule; // add new rule to rule set
endfor
return Rule Set;
```

Figure 3: Basic Sequential Covering Algorithm.

A basic sequential covering algorithm is shown in *Figure 3*. Here, rules are learned for one class at a time. Ideally, when learning a rule for a class, *Ci*, we would like the rule to cover all (or many) of the training tuples of class *C* and none (or few) of the tuples from other classes. In this way, the rules learned should be of high accuracy. The rules need not necessarily be of high coverage. A rule *R* can be assessed by its coverage and accuracy, these calculated by

$$coverage(R) = \frac{n_{covers}}{|D|} \quad (10)$$

$$Accuracy(R) = \frac{n_{correct}}{n_{covers}} \quad (11)$$

Where

n_{covers} : number of tuples covered by *R*;

$n_{correct}$: number of tuples correctly classified by *R*;

$|D|$: number of tuples in *in data set*.

Han and Mihaela in 2018 introduced a new rule based algorithm called Gini-index based rule generation (GIBRG), which uses Gini-Index in rule learning for the selection of attribute-value pairs by measuring the amount of increase in terms of the quality of a single rule being learned. GIBRG leads to an improvement in classification accuracy compared with popular rule based algorithms [21].

3.2.4 Back propagation (ANN)

neural network is a set of connected input/output units in which each connection has a weight associated with it. During the learning phase, the network learns by adjusting the weights so as to be able to predict the correct class label of the input tuples [22]. Neural network consist of three layers input, hidden and output layer. NN include their high tolerance of noisy data as well as their ability to classify patterns on which they have not been trained. They can be used when you may have little knowledge of the relationships between attributes and classes. They are well-suited for continuous-valued inputs *and outputs*, unlike most decision tree algorithms.

There are many kinds of neural network algorithms. The most popular is backpropagation, which learns by iteratively processing a data set of training tuples, comparing the network's prediction for each tuple with the actual known *target* value. the weights are modified so as to minimize the mean squared error between the network's prediction and the actual target value. These modifications are made in the "backwards" direction [23]. Figure. 4 display steps of backpropagation learning algorithm, where Ij, input unit, Oj is the output, w_{ij} is the weight from unit i to unit j, Θj is the bias of the unit. *Err* denote to the error of network's prediction, Tj is the target value.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Algorithm 2: backpropagation algorithm.

Input:

- *D*, a data set consisting of the training tuples and their associated target values;
- *l*, the learning rate;
- *network*, a multilayer feed-forward network.

Output: A trained neural network.

Method:

Initialize all weights and biases in *network*;

while terminating condition is not satisfied {

for each training tuple ***X*** in *D* {

for each input layer unit *j* {

$Oj = Ij$; // output of an input unit is its actual input value

for each hidden or output layer unit *j* {

$Ij = \sum w_{ij} O_i + \Theta j$;

$Oj = \frac{1}{1+e^{-Ij}}$;}

for each unit *j* in the output layer

$Err_j = O_j(1- O_j)(T_j - O_j)$;

for each unit *j* in the hidden layers, from the last to the first hidden layer

$Err_j = O_j(1- O_j)\sum Err_k w_{jk}$;

for each weight w_{ij} in *network* {

$\Delta w_{ij} = (l) Err_j Oi$; // weight increment

$w_{ij} = w_{ij} + \Delta w_{ij}$; } // weight update

for each bias Θj in *network* {

$\Delta\Theta j = (l) Err_j$; // bias increment

$\Theta j = \Theta j + \Delta\Theta j$;} // bias update

}}

Figure 4: Backpropagation Learning Algorithm.

Weipeng in 2018 discussed the evolvement of feed-forward neural networks with random weights (NNRW), especially its applications in deep learning. In NNRW, due to the weights and the threshold of hidden layer are randomly selected and the weights of output layer are obtained analytically, NNRW achieved much faster learning speed than BP-based methods [24].

3.2.5 Support vector machines (SVM)

Support Vector Machines, a promising method for the classification of both linear and nonlinear data. It used a nonlinear mapping to transform the original training data into a higher dimension. SVM can be used for prediction as well as classification. the aim of SVM is to find the best classification function to distinguish between members of the two classes in the training data. the best such function founded by maximizing the margin between the two classes [25], To ensure that the maximum margin hyperplanes are actually found, an SVM classifier attempts to maximize the following function with respect to **w** and *b*:

$$l_p = \frac{1}{2}\|\vec{w}\| - \sum_{i=1}^{k} \alpha_i y_i(\vec{w}.\vec{x}_i + b) + \sum_{i=1}^{t} \alpha_i$$

where *t* is the number of training examples, and α_i , $i = 1, \ldots, t$, are non-negative numbers such that the derivatives of L_p with respect to α_i are zero. α_i are the Lagrange multipliers and L_p is called the Lagrangian. In this equation, the vectors $\vec{w}$ and constant *b* define the hyperplane.

SVM has been successfully applied to various classification tasks that use numerical data. However, the topic of training SVM with heterogeneous data has not been fully examined. Shili and Qinghua 2015 introduced design a novel heterogeneous support vector machine (HSVM) algorithm to classify heterogeneous data. HSVM maps nominal attributes into a real space by minimizing generalization error. The main advantages of HSVM are listed as follows. 1) HSVM effectively improved the performance of SVM in dealing with nominal data or heterogeneous data. 2) HSVM improved the interpretability of decisions, and 3)HSVM was effective in learning with imbalanced data[26].

Ali and Amir presented a model called polar SVM that introduced for one output case based on polar coordinate system and then is expanded to multiple output problems. polar became an efficient and flexible technique for mapping data and obtaining a more accurate model.This reconnoitering comes from this verity that the linearly non-separable data may be usually distributed similar to a sphere, hyperbolic or ellipsoidal form in two dimensional

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

spaces which stimulate hiring the polar coordinate system [27][28].

Rupan and Nikhil proposed an algorithm called Minimally-Spanned Support Vector Machine (MSSVM) with a view to reducing the number of Support Vectors compared to SVM. This reduced the classification time of a test data point significantly [29]. Support vector machine hybrid with other techniques such as genetic algorithm to improve the performance of each other, this introduced by Bernardo et al in 2018 [30]. Tongguang et al 2018 , TSVM-GP integrates a transfer term and group probabilities into the support vector machine (SVM) to improve the classification accuracy [31].

3.2.6 Associative rule analysis

Associative classification is rule-based involving candidate rules as criteria of classification that provide both highly accurate and easily interpretable results to decision makers. The important phase of associative classification is rule evaluation consisting of rule ranking and pruning in which bad rules are removed to improve performance. Existing association rule mining algorithms relied on frequency-based rule evaluation methods such as support and confidence [32], Support determines how often a rule is applicable to agiven data set, while confidence determines how frequently items in Y (right-hand-side or RHS of rule) appear in transactions that contain X (left-hand-side or LHS). support and confidence calculated as follow [33]:

$$\text{Support}(X \Rightarrow Y) = \frac{\sigma(X,Y)}{N} \quad (13)$$

$$\text{Confidence}(X \Rightarrow Y) = \frac{\sigma(X,Y)}{\sigma(X)} \quad (14)$$

where, σ is summation notation, and N represents the total numberof all transactions.

Kiburm and Kichun in 2017, proposed PCAR (Predictability-based collective class association rule) algorithms that applied a new ranking rule called prediction power, the PCAR calculated the predictive power of candidate rules that removed not only redundant rules, but also rules with low predictive power[32]. To identify interesting rule, you need to define measures and constraints such as a matrix-based visualization method [34] to present the measure computation, the distribution of interesting itemsets and the intermediate results of rule mining.

The association rule updated by using fuzzy concept lattice [35], When a new attribute added into the fuzzy concept lattice, it was not necessary to calculate all the frequent nodes and association rules. Therefore, the amount of calculation reduced. Another updating [36] to assocation rule by using Niching Genetic Algorithms which provide a good coverage of the dataset.

Many algorithms introduced to generate assocation rule [37][38] such as Eclat, DHP (Direct Hashing Pruning) and others, but the most popular algorithm called apriori algorithm[39] with their improvements [40-42] aprioriTID, AprioriHybrid and STEM which extended to implemented directly in SQL.

data mining coupled with relational database systems to introduce powerful tool extended in SQL, different methods of coupling stated in[43]. Compared the performance of extended SQL with following alternative: Read directly from DBMS, Cache-mine and User-defined function (UDF).

Finally, Associative classification uses association mining techniques that search for frequently occurring patterns in large databases. The patterns may generate rules, which can be analyzed for use in classification.

3.2.7 Lazy learner (Neighbors)

Decision tree classifiers, Bayesian classifiers, classification by backpropagation, support vector machines, and classification based on association are all examples of eager learners in that they use training tuples to construct a generalization model and in this way are ready for classifying new tuples. This contrasts with lazy learners or instance based methods of classification, such as nearest-neighbor classifiers and case-based reasoning classifiers, which store all of the training tuples in pattern space and wait until presented with a test tuple before performing generalization. Hence, lazy learners require efficient indexing techniques [2].

K-nearest Neighbors

Nearest-neighbor classifiers are based on learning by analogy, that is, by comparing a given test tuple with training tuples that are similar to it. The purpose of the k Nearest Neighbours (kNN) algorithm is to use a database in which the data points are separated into several separate classes to predict the classification of a new sample point [44].

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The kNN algorithm work as follow:

The training tuples are described by n attributes. Each tuple represents a point in an n-dimensional space. *k* is the nearest neighbors of the unknown tuples. This is shown in the following Figure. (*Figure* 5) in the first starting with $k = 1$, we use a test set to estimate the error rate of the classifier. This process can be repeated each time by incrementing *k* to allow for one more neighbors. The *k* value that gives the minimum error rate may be selected. Second compute distance between two points or tuples x_1, x_2 by Distance Metrics, such as Euclidean Distance [45]:

$$dist(X_1, X_2) = \sqrt{\sum_{i=1}^{n}(x_{1i} - x_{2i})^2} \qquad (15)$$

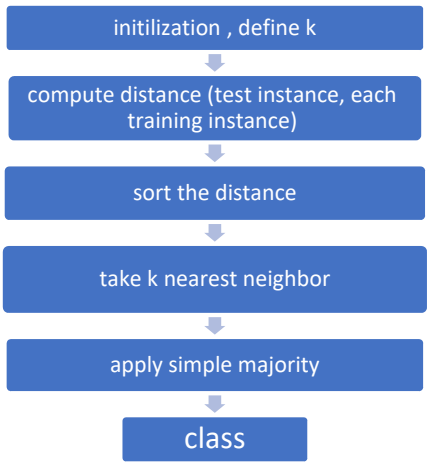

Figure 5: The Steps Of Knn Algorithm.

To improve the performance of conventional kNN method, different methods have been proposed for gaining a higher accuracy [46,47]. Yong et al [48], proposed a classifier called a coarse to fine K nearest neighbor classifier (CFKNNC). This classified much more accurately than various improvements such as the nearest feature line (NFL)[49] classifier, the nearest feature space (NFS) classifier, nearest neighbor line classifier (NNLC) and center-based nearest neighbor classifier (CBNNC). And also, kNN integrated with other methods such as fuzzy set theory in [50], and with neural [51] that achieved high performace.

Case-based reasoning

Case-based reasoning (CBR) involves case storage, indexing, and adaption (the modifying of existing information in order to be able to deal with new problems), CBR depends on classification by similarty, CBR classifiers use a database of problem solutions to solve new problems [52]. Challenges in case-based reasoning include finding a good similarity metric and suitable methods for combining solutions. Other challenges include the selection of salient features for indexing training cases and the development of efficient indexing techniques. A trade-off between accuracy and efficiency evolves as the number of stored cases becomes very large. As this number increases, the case-based reasoner becomes more intelligent [53]. Figure 6 illustrates the case-based reasoning process. Given a description of a problem, CBR relies on indices to find potentially useful cases.

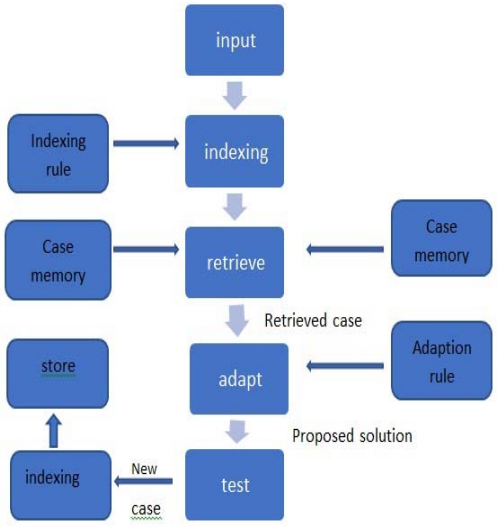

Figure 6: Overview Of The Case-Based Reasoning Process [54]

CBR improved performace by combined with other methods such as learning pseudo metric instead of Euclidean distance to measure the closeness [55], and used fuzzy set, neural network [65-60] to achieve high accuracy.

3.2.8 Other methods

- **Genetic algorithm**

Genetic algorithm is a search algorithm that comes under the range of techniques, collectively known as "evolutionary computing". This is based on the principles of natural genetics and natural selection. The major benefit of this algorithm is that it provides a robust search in complex spaces and is usually less expensive[61]. Thus GA is a powerful adaptive method to solve search and optimization

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

problems. The mechanics of a simple genetic algorithm are surprisingly simple, involving nothing more complex than copying strings and swapping partial strings. In GA, a candidate solution for a specific problem is called an individual or a chromosome and consists of a linear list of genes. Each individual represents a point in the search space, and hence a possible solution to the problem. A population consists of a finite number of individuals. Each individual is decided by an evaluating mechanism to obtain its fitness value. Based on this fitness value and undergoing genetic operators, a new population is generated iteratively [62]. Simple Genetic Algorithm(SGA) that yields good results in many practical problems is composed three operators:

1. Reproduction is a process by which the most highly rated individuals in the current generation are reproduced in the new generation

2. Crossover operator produces two offsprings (new candidate solutions) by recombining the information from two parents.

3. Mutation is a random alteration of some gene values in an individual.

Shounak 2017 proposed a hybrid genetic algorithm-decision tree approach for prediction. DT algorithm generates an ensemble of linear models, which through the GA is transformed into a model with best fit [63]. Also, GA combined with particle swarm optimization (PSO) , individuals in a new generation are created, not only by crossover and mutation operation as in GA, but also by PSO [64]. And combined support vector machine with genetic algorithm to improve the prediction process [65].

- **Fuzzy set approaches**

Fuzzy sets (FSs), which were introduced by Zadeh [66], are well known for their ability to model linguistics and system uncertainties. Due to this ability [67], Fuzzy logic systems are powerful tools especially in modelling, control of complex nonlinear systems and classification. The concepts of fuzzy sets theory are based on fuzzy logic. While in classical logic the membership of sets is binary (0 or 1), in fuzzy logic there are multiple degrees of membership (μ) valued in the real interval from 0 to 1.

There are two parts of this theory used in the development of linguistic models. The first is fuzzification. The fuzzification phase aims to organize a knowledge base that captures the experts' knowledge about a topic. It represents the translation of the expert and observed data into fuzzy sets and a corresponding knowledge base composed of inference rules describing different implications among these fuzzy sets. The second phase, referred to as the defuzzification phase, involves the results of the model, manipulating uncertainty and providing a well-defined output [68].this parts described in the following Figure(Figure 7).

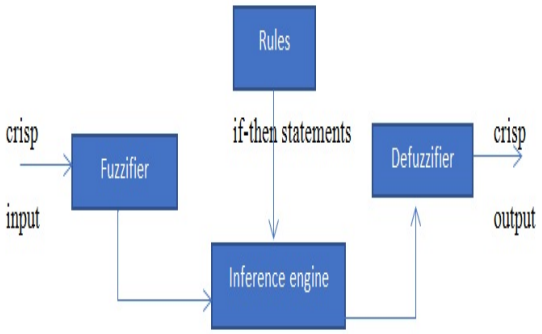

Figure 7: Fuzzy Logic Systems

There are two types of fuzzy logic, type-1 [69] and type-2 [69,70, 71, 72]. The concept of type-2 fuzzy set is an extension and generalization of ordinary (type-1) fuzzy sets. Type-2 fuzzy sets are more suitable in circumstances where it is difficult to determine the accurate membership function for a fuzzy set. Type-2 fuzzy systems integrated with many different optimization methods to improve its performance and modeling, such as with bio-inspired methods [73]and genetic algorithm [74]. Also, fuzzy set used neural network to improve the training process [75]. Fuzzy systems used in many application such as controller, pattern recognition, classification [76,77, 78] and prediction [79,80].

3.2 Unsupervised learning

The term unsupervised learning is derived from the fact that no class labels are used to guide the learning process; hence, the algorithm learns without a supervisor or teacher In this paper, we

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

consider two types of unsupervised learning: clustering and association rule

3.2.1 Association rules

Association rules mining is another key unsupervised data mining method, after clustering, that finds interesting associations (relationships, dependencies) in large sets of data items. The items are stored in the form of transactions that can be generated by an external process, or extracted from relational databases or data warehouses. Due to good scalability characteristics of the association rules algorithms and the ever-growing size of the accumulated data, association rules are an essential data mining tool for extracting knowledge from data [3, 32]. Association rules described with more details in (section 3.1.6), as we noticed association rules used in supervised learning and also used in unsupervised learning.

3.2.2 Clustering

A loose definition of clustering could be "the process of organizing objects into groups whose members are similar in some way." Each cluster is then characterized by the common attributes of the entities it contains. Usually, the objects in a cluster are "more similar" to each other and "less similar" to the objects of the other clusters [81]. The classification of clustering techniques discribed as

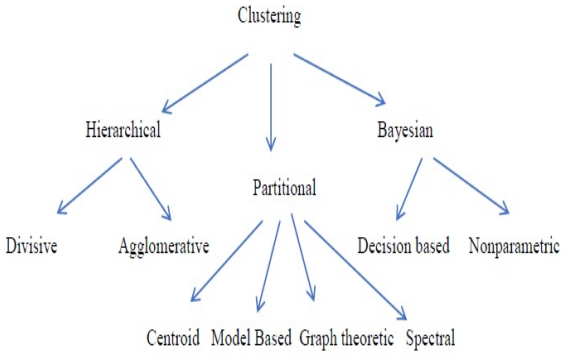

Figure 8: Taxonomy of Clustering Techniques

Figure 8 shows taxonomy of clustering techniques. The three main classes of clustering techniques are hierarchical, partition, and Bayesian [82]. The idea of a hierarchical technique is that it generates a nested sequence of clustering. So, usually one must choose a threshold value indicating how far along in the procedure to go to find the best clustering in the sequence. By contrast, nonhierarchical methods such as partition methods usually require that the number of clusters *K* and an initial clustering be specified as an input to the procedure, which then tries to improve the initial assignment of data points. Bayesian clustering techniques are different from the first two classes because they try to generate a posterior distribution over the collection of all partitions of the data, with or without specifying *K*; the mode of this posterior is the optimal clustering [3].

The *k*-means algorithm is one of the most popular partitioning methods [81], mainly because of its ease of use. It groups data into a user-defined number of clusters, *k*. Each observation is assigned to the nearest cluster, the center of which is defined by the arithmetic mean (average) value of its points, the centroid [82]. The algorithm of k-mean described with more details in [3, 83, and 84]. Clustering improved its performance by combining with fuzzy set [85, 86].

4 Applications

Some of data mining techniques combined with database to achieve more of enhancements in searching process. This section presents these cases.

Case 1: Database with fuzzy sets

In this case fuzzy sets combined with database and introduced fuzzy database. A fuzzy database is a database which is able to deal with uncertain or incomplete information using fuzzy logic. There are many forms of adding flexibility in fuzzy databases. The simplest technique is to add a fuzzy membership degree to each record, an attribute in the range [0,1]. However, there are others kind of databases allowing fuzzy values to be stored in a fuzzy attribute using fuzzy sets or possibility distributions or fuzzy degrees associated to some attributes and with different meanings (membership degree, importance degree, fulfillment degree...).[87].

Cadenas et al 2014 [88] introduced Context-Aware model to deal with Fuzzy Databases. this model improved the user interaction with data stored in form of fuzzy database. Also, fuzzy combined with XML database and proposed heterogeneous fuzzy XML data sources. Jian et al [89] proposed a general solution to search tree patterns over heterogeneous fuzzy XML data sources. The solution provided users with a friendly query

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

interface against the sources independent from their schemas and the heterogeneity of their data.

Carmen [90] distributed two technologies, namely fuzzy and semantic queries, converge into a single system that solves flexible queries on relational databases. On one hand, fuzzy query solving process consists in defining fuzzy sets associated with the attributes involved in the query. Answered tuples accomplish a membership degree to these fuzzy sets. On the other hand, semantic queries use ontologies to return tuples with information semantically similar to the concepts appeared in the query. The following Figure. (Figure. 9) describe The system architecture and the development process that allow to execute flexible queries in the context of relational databases.

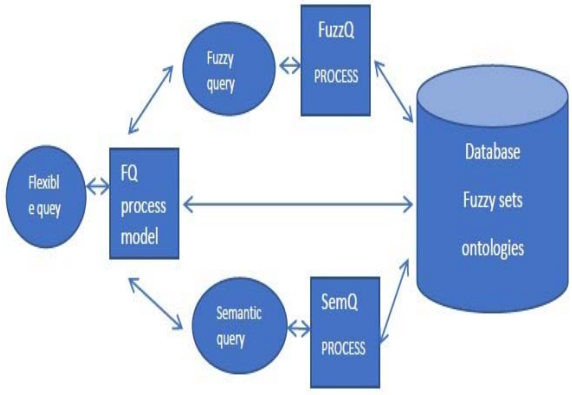

Figure 9: System Architecture.

The system architecture consist of three parts; Semantic Queries (SemQ) process module that executes an algorithm to evaluate the similarity degree, Fuzzy Query (FuzzQ) process module . In this module a fuzzy query is executed performing several subqueries to the fuzzy catalog, and Flexible Queries (FQ) process module which divide the input data and combine the output to respond to user query.

Case 2: Database with association rule

Association rules are used with database to show the relationships between data items and detect hidden information, but most of algorithms suffer from its time-consuming process. Parallel and Distributed algorithms resolve this problem by applying the parallelization techniques [91]. This techniques proposed that database divided into partitions of variable sizes, each partition will be called a cluster than each cluster is converted into matrix by matrix algorithm in the slave system and generate frequent item set from each cluster. It reduced the redundant database scan and improves the efficiency.

Case 3: Database with *k*-NN and case-based reasoning

Sheng and Chun [92] proposed a three-tier Proxy Agent working as a mediator between the users and the backend process of a FAQ system. The first-tier makes use of an improved sequential patterns mining technique to propose effective query prediction and cache services. The second-tier employs an ontology-supported case-based reasoning technique to propose adapted query solutions and tune itself by retaining and updating highly-satisfied cases identified by the user, and the third-tier utilizes an ontology-supported rule-based reasoning to generate possible solutions for the user.

k-NN used with database with name similarity measure expressed in terms of distance based queries in metric spaces [93]. Marcos et al 2018 presented system Merkurion, which is designed to support similarity searching as an extension for a DBMS Query Optimizer. The system adopts a flexible architecture for the handling of a wide set of synopses of distance distributions. For instance, it enables the creation of synopses from several paradigms, such as Nearest Neighbor Histogram, Distance Plot, Compact Distance Histogram and Event Search.

Case 4: Database with decision tree

Zengchang and Jonathan 2011 proposed Linguistic decision tree as a classification model for its advantages of handling uncertainties and being transparent. LDT can be interpreted as a set of linguistic rules, which provided with the information of how the predictions made. This method achieved high performance with more accuracy when compared with other prediction algorithms such as a Support Vector Regression (SVR) system and Fuzzy Semi-Naive Bayes [94].

Keeley et al 2017, Proposed method that uses fuzzy decision trees to build a series of fuzzy learning style predictive (FLSP) models using behavior variables captured from natural language within a CITS tutorial for the four dimensions of the Felder and Silverman Learning Styles model.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The results have indicated that the use of fuzzy predictive models has substantially increased the predictive accuracy of the OSCAR-FLSP compared with the one variable predictor currently used [95]. Also, decision tree used to improve the prediction accuracy of database as described in [96].

4. CONCLUSION

This paper provides a comprehensive review of different data mining techniques. Firstly, demonstrates the steps of data mining which include data, data preparation, mining techniques and evaluation. Mining techniques are considered the most important step in process of data mining. So, this research focused on it. Secondly, it dealt with many data mining techniques with more details especially that integrated with the database. Finally, described some of the applications of using these techniques combined with a database to improve the search process (prediction). As shown database combined with fuzzy logic, decision tree, k-NN, and association rule. From this review, noticed that the popular method used in database prediction is a similarity measure or called k-nearest neighbor. *k-NN* integrated with the case-based reasoning in some cases and with fuzzy sets in other cases.

This review assists the researchers in a database to implement a method from these different methods to predict with next query to achieve some improvements in searching process inside the database.

REFERENCES

[1]. Bramer, Max. ,"Principles of data mining", Vol. 180, London: Springer, 2007.

[2]. Han, Jiawei, Jian Pei, and Micheline Kamber, "Data mining: concepts and techniques", Elsevier, 2011.

[3]. Cios, K. J., Pedrycz, W., Swiniarski, R. W., and Kurgan, L. A., "Data mining: a knowledge discovery approach", Springer Science & Business Media, 2007.

[4]. Wirth, R., and Hipp, J., "CRISP-DM: Towards a standard process model for data mining". *In Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*, April 2000, pp. 29-39.

[5]. Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J.," Data Mining: Practical machine learning tools and techniques". *Morgan Kaufmann*, 2016.

[6]. Van der Aalst, W. M., "Data Mining. In *Process Mining"*, Springer, Berlin, Heidelberg, 2011, pp. 59-91.

[7]. Amatriain, X., & Pujol, J. M., "Data mining methods for recommender systems", In *Recommender Systems Handbook*, Springer US, 2015, pp. 227-262.

[8]. Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. ,"Supervised machine learning: A review of classification techniques", *Emerging artificial intelligence applications in computer engineering*, vol. *160*, 2007, pp. 3-24.

[9]. Caruana, Rich, and Alexandru Niculescu-Mizil. "Data mining in metric space: an empirical analysis of supervised learning performance criteria." *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, 2004, pp. 69-78.

[10]. Chawla, Nitesh V., "C4. 5 and imbalanced data sets: investigating the effect of sampling method, probabilistic estimate, and decision tree structure." *Proceedings of the ICML*, Vol. 3, 2003, pp.66.

[11]. Vieira, J., and Antunes, C., "Decision Tree Learner in the Presence of Domain Knowledge." In *Chinese Semantic Web and Web Science Conference*, Springer, Berlin, Heidelberg, August 2014, pp. 42-55.

[12]. Zaremotlagh, S., and A. Hezarkhani. "The use of decision tree induction and artificial neural networks for recognizing the geochemical distribution patterns of LREE in the Choghart deposit, Central Iran." *Journal of African Earth Sciences*, vol. 128, 2017, pp. 37-46.

[13]. Kim, K., and Hong, J. S., "A hybrid decision tree algorithm for mixed numeric and categorical data in regression analysis". *Pattern Recognition Letters*, *vol. 98, 2017*, pp. 39-45.

[14]. Patil, Tina R., and S. S. Sherekar. "Performance analysis of Naive Bayes and J48 classification algorithm for data classification." *International journal of computer science and applications*, vol. 6, No. 2, 2013, pp. 256-261.

[15]. Wu, Xindong, et al. "Top 10 algorithms in data mining." *Knowledge and information systems*, vol. 14, No. 1, 2008, pp. 1-37.

[16]. El Hindi, K., "A noise tolerant fine tuning algorithm for the Naïve Bayesian learning algorithm.", *Journal of King Saud University-*

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Computer and Information Sciences, *vol. 26*, No. 2, 2014, pp. 237-246.

[17]. Lotte, F., Congedo, M., Lécuyer, A., Lamarche, F., and Arnaldi, B., "A review of classification algorithms for EEG-based brain–computer interfaces", *Journal of neural engineering*, *vol. 4, No. 2*, 2007, R1.

[18]. Bouhamed, H., Masmoudi, A. and Rebai, A.,"Bayesian classifier structure-learning using several general algorithms.", *Procedia Computer Science*, *vol. 46*, 2015, pp. 476-482.

[19]. Jie, L. and Bo, S., "Naive Bayesian classifier based on genetic simulated annealing algorithm." *Procedia Engineering*, *vol. 23*, 2011, pp. 504-509.

[20]. Mahajan, A. and Ganpati, A., "Performance evaluation of rule based classification algorithms." International *Journal of Advanced Research in Computer Engineering & Technology (IJARCET) , Vol.3*, 2014, pp. 3546-3550.

[21]. Liu, Han, and Mihaela Cocea. "Induction of classification rules by gini-index based rule generation." *Information Sciences*, vol. 436, 2018, pp. 227-246.

[22]. Zhang, Guoqiang Peter. "Neural networks for classification: a survey." *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 30, No. 4, 2000, pp. 451-462.

[23]. Cilimkovic, Mirza., "Neural networks and back propagation algorithm.", *Institute of Technology Blanchardstown, Blanchardstown Road North Dublin* , vol.15, 2015.

[24]. Cao, W., Wang, X., Ming, Z., and Gao, J. (2018). "A review on neural networks with random weights." Neurocomputing, vol. *275*, 2018, pp. 278-287.

[25]. Burges, Christopher JC. "A tutorial on support vector machines for pattern recognition.", Data *mining and knowledge discovery*, *vol. 2, No. 2*, 1998, pp. 121-167.

[26]. Peng, Shili, et al. "Improved support vector machine algorithm for heterogeneous data." *Pattern Recognition*, vol. 48, No. 6, 2015, pp. 2072-2083.

[27]. Nedaie, A., and Najafi, A. A. "Polar support vector machine: Single and multiple outputs." *Neurocomputing*, *vol. 171*, 2016, pp. 118-126.

[28]. Nedaie, A., and Najafi, A. A. "Support vector machine with Dirichlet feature mapping." *Neural Networks*, *vol. 98*, 2018, pp. 87-101.

[29]. Panja, Rupan, and Nikhil R. Pal. "MS-SVM: minimally spanned support vector machine." *Applied Soft Computing*, vol. 64, 2018, pp. 356-365.

[30]. de Almeida, Bernardo Jubert, Rui Ferreira Neves, and Nuno Horta. "Combining Support Vector Machine with Genetic Algorithms to optimize investments in Forex markets with high leverage." *Applied Soft Computing*, vol. 64, 2018, pp. 596-613.

[31]. Ni, Tongguang, et al. "Scalable transfer support vector machine with group probabilities." *Neurocomputing*, vol. 273, 2018, pp. 570-582.

[32]. Song, Kiburm, and Kichun Lee. "Predictability-based collective class association rule mining." *Expert Systems with Applications*, vol. 79, 2017, pp. 1-7.

[33]. Doostan, Milad, and Badrul H. Chowdhury. "Power distribution system fault cause analysis by using association rule mining." *Electric Power Systems Research*, vol.152, 2017, pp. 140-147.

[34]. Chen, W., Xie, C., Shang, P., and Peng, Q., "Visual analysis of user-driven association rule mining." Journal of Visual Languages & Computing, vol. 42, 2017, pp. 76-85.

[35]. Zou, Caifeng, et al. "Mining and updating association rules based on fuzzy concept lattice." *Future Generation Computer Systems, vol.* 82, 2018, pp. 698-706.

[36]. Martín, D., Alcalá-Fdez, J., Rosete, A., and Herrera, F. "NICGAR: A Niching Genetic Algorithm to mine a diverse set of interesting quantitative association rules." *Information Sciences*, *vol. 355*, 2016, pp. 208-228.

[37]. Agrawal, R., Imieliński, T., and Swami, A. (1993, June). "Mining association rules between sets of items in large databases." In *Acm sigmod record* , ACM, Vol. 22, No. 2, June 1993, pp. 207-216.

[38]. Park, Jong Soo, Ming-Syan Chen, and Philip S. Yu. "An effective hash-based algorithm for mining association rules." ACM, Vol. 24. No. 2, 1995.

[39]. Hipp, Jochen, Ulrich Güntzer, and Gholamreza Nakhaeizadeh. "Algorithms for association rule mining—a general survey and comparison." *ACM sigkdd explorations newsletter*, vol. 2, No. 1, 2000, pp. 58-64.

[40]. Yu, H., Wen, J., Wang, H., and Jun, L. "An improved Apriori algorithm based on the Boolean matrix and Hadoop." *Procedia Engineering*, vol. *15*, 2011, pp. 1827-1831.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[41]. Bhandari, Akshita, Ashutosh Gupta, and Debasis Das. "Improvised apriori algorithm using frequent pattern tree for real time applications in data mining." *Procedia Computer Science*, vol. 46, 2015, pp. 644-651.

[42]. Djenouri, Youcef, and Marco Comuzzi. "Combining Apriori heuristic and bio-inspired algorithms for solving the frequent itemsets mining problem." *Information Sciences*, vol. 420, 2017, pp. 1-15.

[43]. Sarawagi, Sunita, Shiby Thomas, and Rakesh Agrawal. "Integrating association rule mining with relational database systems: Alternatives and implications." *Data mining and knowledge discovery*, vol. 4, 2000, pp. 89-125.

[44]. Zeng, Yong, Yupu Yang, and Liang Zhao. "Pseudo nearest neighbor rule for pattern classification." *Expert Systems with Applications*, vol. 36, No. 2, 2009, pp. 3587-3595.

[45]. Zhang, Shichao, et al. "Self-representation nearest neighbor search for classification." *Neurocomputing*, vol. 195, 2016, pp.137-142.

[46]. Souza, Roberto, Letícia Rittner, and Roberto Lotufo. "A comparison between k-optimum path forest and k-nearest neighbors supervised classifiers." *Pattern recognition letters*, vol. 39, 2014, pp. 2-10.

[47]. Ertuğrul, Ömer Faruk, and Mehmet Emin Tağluk. "A novel version of k nearest neighbor: Dependent nearest neighbor." *Applied Soft Computing*, vol. 55, 2017, pp. 480-490.

[48]. Xu, Yong, et al. "Coarse to fine K nearest neighbor classifier." *Pattern Recognition Letters*, vol. 34, No. 9, 2013, pp. 980-986.

[49]. Zheng, Wenming, Li Zhao, and Cairong Zou. "Locally nearest neighbor classifiers for pattern classification." *Pattern recognition*, vol. 37, No. 6, 2004, pp. 1307-1309.

[50]. Ezghari, Soufiane, Azeddine Zahi, and Khalid Zenkouar. "A new nearest neighbor classification method based on fuzzy set theory and aggregation operators." *Expert Systems with Applications*, vol. 80, 2017, pp. 58-74.

[51]. Gallego, Antonio-Javier, et al. "Clustering-based k-nearest neighbor classification for large-scale data with neural codes representation." *Pattern Recognition*, vol. 74, 2018, pp. 531-543..

[52]. McKenzie, Dean P., and Richard S. Forsyth. "Classification by similarity: An overview of statistical methods of case-based reasoning." *Computers in Human Behavior*, vol. 11, No. 2, 1995, pp. 273-288.

[53]. Jurisica, I., Mylopoulos, J., Glasgow, J., Shapiro, H., and Casper, R. F. "Case-based reasoning in IVF: Prediction and knowledge mining." *Artificial intelligence in medicine*, *vol. 12*, No. 1, 1998, pp. 1-24.

[54]. Liu, Cheng-Hsiang, and Hung-Chi Chen. "A novel CBR system for numeric prediction." *Information Sciences*, vol. 185, No. 1, 2012, pp. 178-190.

[55]. Yan, Aijun, Hang Yu, and Dianhui Wang. "Case-based reasoning classifier based on learning pseudo metric retrieval." *Expert Systems with Applications*, vol. 89, 2017, pp. 91-98.

[56]. Wang, Mingyang, et al. "Development a case-based classifier for predicting highly cited papers." *Journal of Informetrics*, vol. 6, No. 4, 2012, pp. 586-599.

[57]. Lao, S. I., et al. "Achieving quality assurance functionality in the food industry using a hybrid case-based reasoning and fuzzy logic approach." *Expert Systems with Applications,* vol. 39, No. 5, 2012, pp. 5251-5261.

[58]. Thirugnanam, Mythili, et al. "Improving the prediction rate of diabetes diagnosis using fuzzy, neural network, case based (FNC) approach." *Procedia engineering*, vol. 38, 2012, pp. 1709-1718.

[59]. Han, Min, and ZhanJi Cao. "An improved case-based reasoning method and its application in endpoint prediction of basic oxygen furnace." *Neurocomputing*, vol. 149, 2015, pp. 1245-1252.

[60]. Hu, Xin, et al. "The application of case-based reasoning in construction management research: An overview." *Automation in Construction*, vol. 72, 2016, pp. 65-74.

[61]. Man, Kim-Fung, Kit-Sang Tang, and Sam Kwong. "Genetic algorithms: concepts and applications [in engineering design]." *IEEE transactions on Industrial Electronics*, vol. 43, No. 5, 1996, 519-534.

[62]. Reeves, Colin R. "A genetic algorithm for flowshop sequencing." *Computers & operations research*, vol. 22, No. 1, 1995, pp. 5-13.

[63]. Datta, Shounak, Vikrant A. Dev, and Mario R. Eden. "Hybrid genetic algorithm-decision tree approach for rate constant prediction using structures of reactants and solvent for Diels-

Alder reaction." *Computers & Chemical Engineering*, vol. 106, 2017, pp. 690-698.

[64]. Juang, Chia-Feng. "A hybrid of genetic algorithm and particle swarm optimization for recurrent network design." *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 34, No. 2, 2004, pp. 997-1006.

[65]. de Almeida, Bernardo Jubert, Rui Ferreira Neves, and Nuno Horta. "Combining Support Vector Machine with Genetic Algorithms to optimize investments in Forex markets with high leverage." *Applied Soft Computing*, vol. 64, 2018, pp. 596-613.

[66]. Zadeh, Lotfi Asker. "The concept of a linguistic variable and its application to approximate reasoning—I." *Information sciences*, vol. 8, No. 3, 1975, pp. 199-249.

[67]. Kumbasar, Tufan, et al. "Exact inversion of decomposable interval type-2 fuzzy logic systems." *International Journal of Approximate Reasoning*, vol. 54, No. 2, 2013, pp. 253-272.

[68]. Canavese, Daniel, Neli Regina Siqueira Ortega, and Margarida Queirós. "The assessment of local sustainability using fuzzy logic: An expert opinion system to evaluate environmental sanitation in the Algarve region, Portugal." *Ecological indicators*, vol. 36, 2014, pp. 711-718.

[69]. Biglarbegian, Mohammad, William Melek, and Jerry Mendel. "On the robustness of type-1 and interval type-2 fuzzy logic systems in modeling." *Information Sciences*, vol. 181, No. 7, 2011, pp. 1325-1347.

[70]. Mendel, Jerry M., and Hongwei Wu. "New results about the centroid of an interval type-2 fuzzy set, including the centroid of a fuzzy granule." *Information Sciences, vol.* 177, No. 2, 2007, pp. 360-377.

[71]. Karnik, Nilesh N., and Jerry M. Mendel. "Applications of type-2 fuzzy logic systems to forecasting of time-series." *Information sciences*, vol. 120, 1999, pp. 89-111.

[72]. Hassan, Saima, et al. "Optimal design of adaptive type-2 neuro-fuzzy systems: A review." *Applied Soft Computing*, vol. 44, 2016, pp. 134-143.

[73]. Castillo, Oscar, and Patricia Melin. "Optimization of type-2 fuzzy systems based on bio-inspired methods: A concise review." *Information Sciences*, vol. 205, 2012, pp. 1-19.

[74]. Sanz, José Antonio, et al. "Improving the performance of fuzzy rule-based classification systems with interval-valued fuzzy sets and genetic amplitude tuning." *Information Sciences*, vol. 180, No. 19, 2010, pp. 3674-3685.

[75]. Rajan, Susmitha, and Saurabh Sahadev. "Performance Improvement of Fuzzy Logic Controller Using Neural Network." *Procedia Technology*, vol. 24, 2016, pp. 704-714.

[76]. Ozols, Juris, and Arkady Borisov. "Fuzzy classification based on pattern projections analysis." *Pattern Recognition*, vol. 34, No. 4, 2001, pp. 763-781.

[77]. Ebert, Christof. "Rule-based fuzzy classification for software quality control." *Fuzzy Sets and Systems*, vol. 63, No. 3, 1994, pp. 349-358.

[78]. Fernández, Alberto, et al. "A study of the behaviour of linguistic fuzzy rule based classification systems in the framework of imbalanced data-sets." *Fuzzy Sets and Systems*, vol. 159, No. 18, 2008, pp. 2378-2398.

[79]. Kuvayskova, Yu E. "The prediction algorithm of the technical state of an object by means of fuzzy logic inference models." *Procedia Engineering*, vol. 201, 2017, pp. **767-772.**

[80]. Bacani, Felipo, and Laécio C. de Barros. "Application of prediction models using fuzzy sets: a Bayesian inspired approach." *Fuzzy Sets and Systems, vol.* 319, 2017, pp. 104-116.

[81]. Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. "Unsupervised learning." *The elements of statistical learning*. Springer, New York, NY, 2009. pp. 485-585.

[82]. Odziomek, Katarzyna, Anna Rybinska, and Tomasz Puzyn. "Unsupervised Learning Methods and Similarity Analysis in Chemoinformatics." *Handbook of Computational Chemistry*, 2016, pp. 1-38.

[83]. López-Rubio, Ezequiel, Esteban J. Palomo, and Francisco Ortega-Zamorano. "Unsupervised learning by cluster quality optimization." *Information Sciences*, vol. 436, 2018, pp. 31-55.

[84]. Berkhin, Pavel. "A survey of clustering data mining techniques." *Grouping multidimensional data*. Springer, Berlin, Heidelberg, 2006, pp. 25-71.

[85]. Pham, Van Nha, Long Thanh Ngo, and Witold Pedrycz. "Interval-valued fuzzy set approach to fuzzy co-clustering for data classification." *Knowledge-Based Systems*, vol. 107, 2016, pp. 1-13.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[86]. Xu, Zeshui, Jian Chen, and Junjie Wu. "Clustering algorithm for intuitionistic fuzzy sets." *Information Sciences*, vol. 178, No. 19, 2008, pp. 3775-3790.

[87]. Galindo, José, ed. *Handbook of research on fuzzy information processing in databases*. IGI Global, 2008.

[88]. Cadenas, José Tomás, Nicolás Marín, and M. A. Vila. "Context-aware fuzzy databases." *Applied Soft Computing, vol.* 25, 2014, pp. 215-233.

[89]. Liu, Jian, X. X. Zhang, and Lei Zhang. "Tree pattern matching in heterogeneous fuzzy XML databases." *Knowledge-Based Systems, vol.* 122, 2017, pp. 119-130.

[90]. Martínez-Cruz, Carmen, José M. Noguera, and M. Amparo Vila. "Flexible queries on relational databases using fuzzy logic and ontologies." *Information Sciences*, vol. 366, 2016, pp. 150-164.

[91]. Vasoya, Anil, and Nitin Koli. "Mining of association rules on large database using distributed and parallel computing." *Procedia Computer Science*, vol. 79, 2016, pp. 221-230.

92. Yang, Sheng-Yuan, and Chun-Liang Hsu. "An ontological Proxy Agent with prediction, CBR, and RBR techniques for fast query processing." *Expert Systems with Applications*, vol. 36, No. 5, 2009, pp. 9358-9370.

[93]. Bedo, Marcos VN, et al. "The Merkurion approach for similarity searching optimization in Database Management Systems." *Data & Knowledge Engineering*, vol. 113, 2018, pp. 18-42.

[94]. Qin, Zengchang, and Jonathan Lawry. "Prediction and query evaluation using linguistic decision trees." *Applied Soft Computing*, vol. 11, No. 5, 2011, pp. 3916-3928.

[95]. Crockett, Keeley, Annabel Latham, and Nicola Whitton. "On predicting learning styles in conversational intelligent tutoring systems using fuzzy decision trees." *International Journal of Human-Computer Studies*, vol. 97, 2017, pp. 98-115.

[96]. Zhao, Bi, and Bin Xue. "Improving prediction accuracy using decision-tree-based meta-strategy and multi-threshold sequential-voting exemplified by miRNA target prediction." *Genomics, vol.* 109, No. 3, 2017, pp. 227-232.