An Overview on Classification Algorithms in Data Mining

V. Rengaraj[1] D. Elavarasi[2]
[1]Assistant Professor [2]M.Phil Research Scholar
[1,2]Department of Computer Science
[1,2]Thanthai Hans Roever College, Perambalur-621212, India

Abstract— Data classification is the method of organizing data into categories for its most valuable and efficient use. A well planned data classification system makes vital data easy to find and retrieve. It is used for classifying data into different classes according to some constrains. A Major classification technique includes decision trees and neural networks. This paper aimed to do the analysis of several data mining classification techniques and inclusive survey of different classification algorithms and their features and limitations.

Key words: Data Mining Classification Techniques, Decision Trees, Artificial Neural Networks, Bayesian classification, K-Nearest Neighbor

I. INTRODUCTION

The term data mining refers to the finding of relevant and useful information from data in the databases. It is the process of discovering interesting knowledge from large amount of data that are stored either in databases, data warehouses. Data mining applications use different kind of parameters to examine the data. Data Mining basic techniques are Clustering, Association rule discovery, Classification , Sequential pattern discovery and the Distributed data mining Techniques are Distributed Classifier learning, collective data mining , distributed clustering and others.

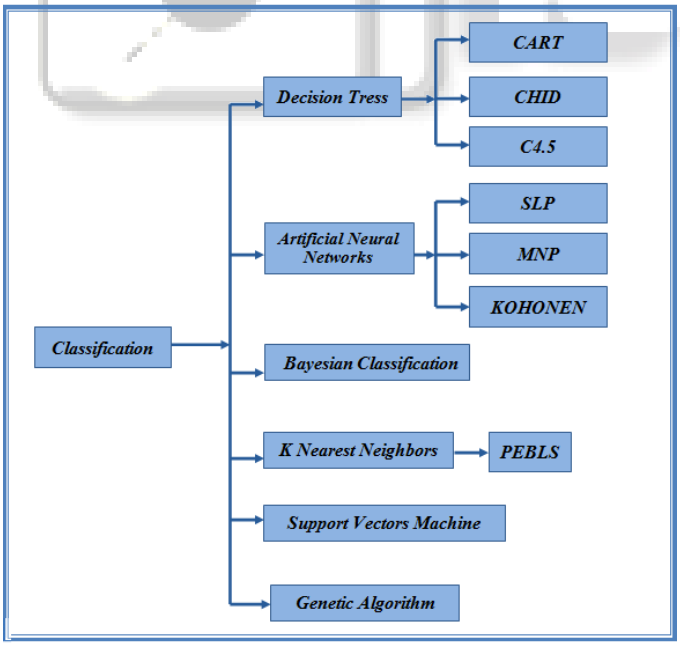

Fig. 1: Data mining techniques

Classification algorithms in data mining are capable of processing a large amount of data. It can predict categorical class labels and classifies data based on training set and class labels and hence can be used for classifying newly available data. Thus it can be outlined as an inevitable part of data mining and is gaining more popularity. The algorithm is a computational procedure which takes some value or set of value as input and generates some value or set as output. The result of a given problem is the output that we got after solving the problem. The algorithm is considered to be correct , if for every input instance, it generate the correct output and it gets terminated and give the desired output otherwise it does not considered as a correct algorithm.

In Fig.1, we present the basic classification techniques. Several major kinds of classification method including Decision Trees, Artificial Neural Networks, Bayesian classification, k-nearest neighbor classifier, Support Vector Machine and Genetic Algorithm. The goal of this survey is to provide a comprehensive review of different classification techniques in data mining.

II. DECISION TREES

Magnetic Decision tree is a predictive modeling technique most often used for classification in data mining [1]. The Classification algorithm is inductively learned to construct a model from the preclassified data set. Each data item is defined by values of the attributes and classification may be viewed as mapping from a set of attributes to a particular class. Each non-terminal node in the decision tree represents a test or decision on the considered data item. Choice of a certain branch depends upon the outcome of the test. To classify a particular data item, we start at the root node and follow the assertions down until we reach a terminal node (or leaf). A decision is made when a terminal node is approached [2]. An advantage of using decision tree algorithms for IDS is that its construction does not require any domain knowledge. Hence a data mining expert with little knowledge of networking can help build accurate decision tree models. Another significant advantage is that decision trees can handle high dimensional data [3]. This increases the suitability of decision tree algorithms for IDS especially while considering the heterogeneity of network connection data and its ever increasing size. Decision trees are able to process both numerical and categorical data (this suits the alphanumeric nature of network connection data) [4]. Finally, decision tree representations are easy to understand hereby making it easier for the network analyst to identify network trends and deviations from normal traffic [3]. Disadvantages of decision tree algorithms in IDS are that the output attribute must be categorical (normal or anomaly) and limited to one output attribute. Decision tree algorithms are also known to be unstable and trees created from numeric datasets can be complex [2]. A typical decision tree is shown in the following Figure,

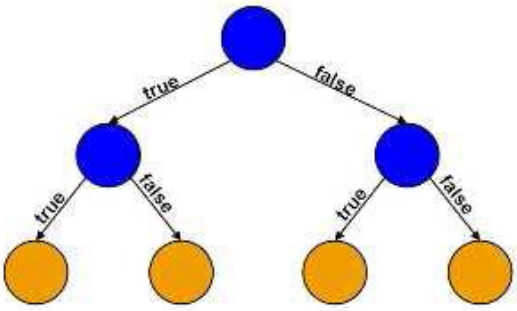

Fig. 2: Decision Tree Induction

III. ARTIFICIAL NEURAL NETWORKS

Neural networks (NN) are systems modeled based on the working of the human brain [1]. As the human brain consists of millions of neurons that are interconnected by synapses, a neural network is a set of connected input/output units in which each connection has a weight associated with it. The network learns in the learning phase by adjusting the weights so as to be able to predict the correct class label of the input [5]. Neural networks have been used both in anomaly intrusion detection as well as in misuse intrusion detection [7]. For anomaly intrusion detection, neural networks were modeled to learn the typical characteristics of system users and identify statistically significant variations from the user's established behavior. In misuse intrusion detection the neural network would receive data from the network stream and analyze the information for instances of misuse. An Advantage of neural network algorithms as a classifier in IDS is that it requires less formal statistical training [2]. Neural networks are able to implicitly detect complex nonlinear relationships between dependent and independent variables. Neural networks are also known to exhibit a high tolerance to noisy data (this would come in handy while dealing with noisy connection data). Neural networks also boast of an availability of multiple training algorithms [6]. It can be argued that the "Black box" nature of neural networks as limited its potential as a classifier for IDS [1]. Another disadvantage of the neural network algorithm is its relatively greater computational burden. Artificial neural networks are known for their proneness to over fitting and to require long training time [2]. The three layer neural network is illustrated in the following figure,

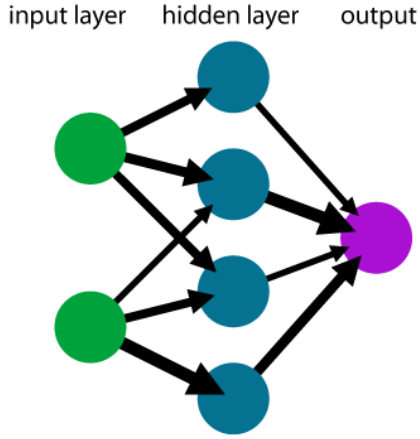

Fig. 3: A simple neural network

IV. BAYESIAN CLASSIFICATION

The Bayesian classification is a simple probabilistic classifier based on applying Bayes' theorem with strong (naive) independence assumptions. A more descriptive term for the underlying probability model would be "independent feature model". Bayesian classification provides practical learning algorithms and prior knowledge and observed data can be combined. Bayesian Classification provides a useful perspective for understanding and evaluating many learning algorithms. It calculates explicit probabilities for hypothesis and it is robust to noise in input data. A naive Bayesian classifier assumes that the presence or absence of a particular feature is unrelated to the presence or absence of any other feature, given the class variable. For some types of probability models, naive Bayesian classifiers can be trained very efficiently in a supervised learning setting. It also called idiot‟s Bayes, simple Bayes, and independence Bayes. This method is important for several reasons. It is very easy to construct, not needing any complicated iterative parameter estimation schemes. This means it may be readily applied to huge data sets. It is easy to interpret, so users unskilled in classifier technology can understand why it is making the classification it makes. And finally, it often does surprisingly well: it may not Probabilistic approaches to classification typically involve modeling the conditional probability distribution P(C|D), where C ranges over classes and D over descriptions, in some language, of objects to be classified. Given a description d of a particular object, we assign the class argmaxc P(C = c|D = d). A Bayesian approach splits this posterior distribution into a prior distribution P(C) and a likelihood P(D|C):P(D = d|C = c)P(C = c)

$$argmaxc\ P(C = c|D = d) = argmaxc$$

$$p\left(D = \frac{d}{c} = c\right) p(C = c) \longrightarrow (1)$$

The denominator P(D = d) is a normalizing factor that can be ignored when determining the maximum a posteriori class, as it does not depend on the class. The key term in Equation (1) is P(D = d|C = c), the likelihood of the given description given the class (often abbreviated to P(d|c)). A Bayesian classifier estimates these likelihoods from training data, but this typically requires some additional simplifying assumptions. For instance, in an attribute-value representation (also called propositional or single-table representation), the individual is described by a vector of Values a1. . . an for a fixed set of attributes A1, . . . ,An. Determining P(D = d|C = c) here requires an estimate of the joint probability P(A1 = a1, . . . ,An = an|C = c), abbreviated to P(a1, . . . ,an|c). This joint probability Distribution is problematic for two reasons:

1) Its size is exponential in the number of attributes n, and
2) It requires a complete training set, with several examples for each possible description. These problems vanish if we can assume that all attributes are independent

Given the class:

$$P(A1 = a1, \ldots, An = an|C = c) = \Pi^{n}_{i=1} P(Ai = ai|C = c) \longrightarrow (2)$$

This assumption is usually called the naive Bayes assumption, and a Bayesian classifier using this assumption

is called the naive Bayesian classifier, often abbreviated to "naive Bayes". Effectively, it means that we are ignoring interactions between attributes within individuals of the same class. [8], [9].

V. K-NEAREST NEIGHBOR

K-Nearest Neighbor (k-NN) is an instance based learning method for classifying objects based on the closest training examples in the feature space [3]. It is a type of lazy learning where the function is only approximated locally and all computations are deferred until classification. The k-nearest neighbor algorithm is amongst the simplest of all machine learning algorithms: an object is classified by a majority vote of its neighbors, with the object being assigned to the class most common amongst its k nearest neighbors. If k=1, then the object is simply assigned to the class of its nearest neighbor [9]. The k-NN algorithm uses all labeled training instances as a model of the target function. During the classification phase, k-NN uses a similarity-based search strategy to determine a locally optimal hypothesis function. Test instances are compared to the stored instances and are assigned the same class label as the k most similar stored instances. Generally it is used for intrusion detection in combination with statistical schemes (anomaly detection) [10]. An advantage of the K-Nearest Neighbor Algorithm as a classifier for an IDS is that it is analytically tractable. KNN is simple in implementation and it uses local information, which can yield highly adaptive behavior. Finally, a major strength of the KNN algorithm is that it lends itself very easily to parallel implementations. One of the weaknesses of the K-Nearest Neighbor Algorithm as a classifier for an IDS is its large storage requirements. KNNs are also known to be highly susceptible to the curse of dimensionality and slow in classifying test tuples.

A K-Nearest Neighbor process is shown in the following Figure, Example of k-NN classification. The test sample (green circle) should be classified either to the first class of blue squares or to the second class of red triangles. If k = 3 (solid line circle) it is assigned to the second class because there are 2 triangles and only 1 square inside the inner circle. If k = 5 (dashed line circle) it is assigned to the first class (3 squares vs. 2 triangles inside the outer circle).

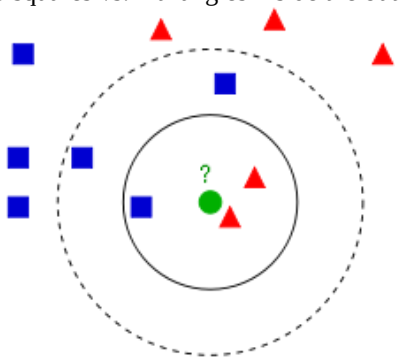

Fig. 4: A K-Nearest Neighbor process

VI. SUPPORT VECTOR MACHINES (SVM)

SVMs introduced in COLT-92 by Boser, Guyon & Vapnik. Became rather popular since. Theoretically well motivated algorithm developed from Statistical Learning Theory (Vapnik & Chervonenkis) since the 60s.The support vector machine usually deals with pattern classification that means this algorithm is used mostly for classifying the different types of patterns. Now, there is different type of patterns i.e. Linear and non-linear. Linear patterns are patterns that are easily distinguishable or can be easily separated in low dimension whereas non-linear patterns are patterns that are not easily distinguishable or cannot be easily separated and hence these type of patterns need to be further manipulated so that they can be easily separated. Basically, the main idea behind SVM is the construction of an optimal hyper plane, which can be used for classification, for linearly separable patterns. The optimal hyper plane is a hyper plane selected from the set of hyper planes for classifying patterns that maximizes the margin of the hyper plane i.e. the distance from the hyper plane to the nearest point of each patterns. The main objective of SVM is to maximize the margin so that it can correctly classify the given patterns i.e. larger the margin size more correctly it classifies the patterns.The equation shown below is the hyper plane:

Hyper plane, aX + bY = C

The given pattern can be mapped into higher dimension space using kernel function, Φ(x).

i.e. x Φ(x) \\\electing different kernel function is an important aspect in the SVM-based classification, commonly used kernel functions include LINEAR, POLY, RBF, and SIGMOID. For e.g.: the equation for Poly Kernel function is given as:

$$K(x, y) = <x, y>^p$$

The main principle of support vector machine is that given a set of independent and identically distributed training sample {(xi ,yi)}N i=1, where x ϵ Rd and yi ϵ {−1,1} , denote the input and output of the classification. The goal is to find a hyper plane wT.x + b = 0, which separate the two different samples accurately. Therefore, the problem of solving optimal classification now translates into solving quadratic programming problems. It is to seek a partition hyper plane to make the bilateral blank area (2/||w||) maximum, which means we have to maximize the weight of the margin. It is expressed as:

$$\text{Min } \Phi(w) = \tfrac{1}{2} \| w \|^2 = \tfrac{1}{2} (w, w),$$

$$\text{Such that: } y_i (w.x_i + b) >= 1$$

SVM can be easily extended to perform numerical calculations. Here we discuss two such extensions. To extend SVM to perform regression analysis, where the goal is to produce a linear function that can approximate that target function [11].

VII. GENETIC ALGORITHM

Genetic Algorithm attempt to incorporate ideas of natural evaluation The general idea behind GAs is that we can build a better solution if we somehow combine the "good" parts of other solutions (schemata theory), just like nature does by combining the DNA of living beings [12]. Genetic Algorithm is basically used as a problem solving strategy in order to provide with a optimal solution. They are the best way to solve the problem for which little is known. They will work well in any search space because they form a very general algorithm. The only thing to be known is what the particular situation is where the solution performs very well, and a genetic algorithm will generate a high quality solution.

Genetic algorithms use the principles of selection and evolution to produce several solutions to a given problem. It shown in following figure,

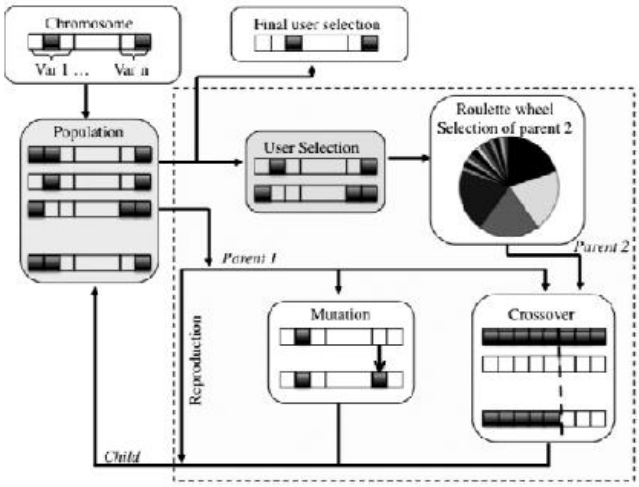

Fig. 5: Structural view of Genetic Algorithm

Genetic algorithms (GAs) [13] are based on a biological applications; it depends on theory of evolution. When GAs is used for problem solving, the solution has three distinct stages:

- The solutions of the problem are encoded into representations that support the necessary variation and selection operations; these representations, are called chromosomes, are as simple as bit strings.
- A fitness function judges which solutions are the "best" life forms, that is, most appropriate for the solution of the particular problem. These individuals are favored in survival and reproduction, thus giving rise to generation.

Crossover and mutation produce a new generation of individuals by recombining features of their parents. Eventually a generation of individuals will be interpreted back to the original problem domain and the fit individual represents the solution.

VIII. Conclusion

This paper includes different classification techniques used in data mining and a study on each of them. Data mining is a wide area that incorporates techniques from various fields including machine learning, artificial intelligence, statistics and pattern recognition, for the analysis of large volumes of data. Classification systems are usually tough in modeling interactions. Each of these methods can be used in various situations as needed where one tends to be useful while the other may not and vice-versa. These classification algorithms can be implemented on different types of data sets like data of patients, financial data according to performances. Hence these classification techniques show how a data can be determined and grouped when a new set of data is available.

IX. References

[1] Carbone, P. L. (1997). "Data mining or knowledge discovery in databases: An overview", In Data Management Handbook, New York: Auerbach Publications.

[2] Kesavulu, E., Reddy, V. N. and Rajulu, P. G. (2011). "A Study of Intrusion Detection in Data Mining". Proceedings of the World Congress on Engineering 2011 Vol IIIWCE 2011, July 6 - 8, 2011, London, U.K.

[3] Lee, W., S. J. Stolfo, & Mok, K. W. (1999). "A data mining framework for building intrusion detection models," In Proc. of the 1999 IEEE Symp. On Security and Privacy (pp. 120-132), Oakland, CA: IEEE Computer Society Press.

[4] Lee, W., Stolfo, S.J. & Mok, K.W. (1999). "Mining in a data-flow environment: Experience in network intrusion detection," (Chaudhuri, S. & Madigan, D. Eds.). Proc. of the Fifth International Conference on Knowledge Discovery and Data Mining (KDD-99) (pp. 114-124), San Diego, CA: ACM,

[5] Lappas, T. and Pelechrinis, K. (2006). Data Mining Techniques for (Network) Intrusion Detection Systems, Department of Computer Science and Engineering Riverside, Riverside CA. [9]. Lee, W. & Stolfo, S.J. (1998). Data mining approaches for intrusion detection, In Proc. of the Seventh USENIX Security Symp., San Antonio, TX.

[6] Frank, J. (1994). "Artificial intelligence and intrusion detection: Current and future directions", In Proc. of the 17th National Computer Security Conference, Baltimore, MD. National Institute of Standards and Technology (NIST).

[7] Lee, W. & Stolfo, S.J et al. (2000). "A data mining and CIDF based approach for detecting novel and Distributed intrusions", In Proc. of Third International Workshop on Recent Advances in Intrusion Detection (RAID 2000), Toulouse, France.

[8] H. Bhavsar, A. Ganatra,"Variations of Support Vector Machine Classification: A survey", International Journal of Advanced Computer Research, Volume 2, Number 4, Issue 6 (2012) 230–236.

[9] Ms. Aparna Raj, Mrs. Bincy, Mrs. T.Mathu "Survey on Common Data Mining Classification Techniques", International Journal of Wisdom Based Computing, Vol. 2(1), April 2012

[10] Lane, T. D. (2000). "Machine Learning Techniques for the computer security domain of anomaly detection", Ph.D. Thesis, Purdue Univ., West Lafayette, IN.

[11] Ashis Pradhan., "Support Vector Machines a survey, ISSN 2250-2459, Volume 2, Issue 8, August 2012.

[12] AnkitaAgarwal,"Secret Key Encryption algorithm using genetic algorithm", vol.-2, no.-4, ISSN: 2277 128X, IJARCSSE, pp. 57-61, April 2012.

[13] Li Lin, Longbing Cao, Jiaqi Wang, Chengqi Zhang, "The Applications of Genetic Algorithms in Stock Market Data Mining Optimisation", Proceedings of Fifth International Conference on Data Mining, Text Mining and their Business Applications, pp-593-604,sept 2005.

