

Comparison of SIS and Issues Related with Constructing the Compact and Accurate Classification Models

¹Preethi Jeevan and ²Dr.V.S.S.S.Balaram

¹Research Scholar, Sri SatyaSai University of Technology

²HOD information technology SNIST. Hyderabad

ARTICLE DETAILS

Article History

Published Online: 25 May 2019

Keywords

SIS, database, Models

ABSTRACT

Data mining can be said to have the aim to dissect the observational datasets to discover connections and to introduce the data in manners that are both justifiable and valuable. Models constructed utilizing instances chosen by SIS give better precision as practically identical to the models constructed utilizing instances chosen by algorithms, for example, DROP 3, DROP 5, EXPLORE, and instances chosen utilizing the methods, for example, sampling and instance commonality. A variety to the algorithm SIS, called, adjusted SIS (MODSIS) is additionally recommended for consequently choosing number of instances to be chosen from the train set by reviewing number of unmistakable sureness estimations of the base neural system in arranging training instances. This paper is fundamentally founded on Comparison of SIS with administered, unsupervised and sampling of artificial intelligence and basically decide the certainty, class of instance and designing issues in constructing the compact and accurate classification models in SIS. The proposed technique is helpful in constructing the Compact and Accurate Classification neural system models for better attribution of missing qualities and for the construction of space effective neural system gatherings.

1. Introduction

Advances in data collection and capacity innovation have empowered scientists to gather huge amounts of data. With the enormous amount of data put away in records, databases, and different stores, scientists from numerous territories have been animated to receive and grow new strategies for data analysis in various fields of science. Powerful methods for investigating, deciphering and removing intriguing learning that could help in basic leadership forms have additionally been planned Data mining is such an innovation, to the point that consolidates data analysis strategies (wise frameworks procedures) with modern algorithms for preparing substantial volumes of data.

Data mining is the way toward applying neural systems, bunching, fuzzy logic and choice tree and so on to data with the aim of revealing concealed examples and Extricating information It includes a joining of these diverse strategies from various controls which can encourage the extraction of information from a lot of data.

2. Data Mining

Data mining is a quickly advancing exploration zone that is at the crossing point of a few orders, including databases, machine learning, insights, design acknowledgment, artificial intelligence, optimization, representation and high-performance parallel processing. Numerous conventional systems, particularly the decision trees, are well known. Neural systems have been effectively utilized in a few differentiated applications, for example, transcribed character acknowledgment, content ID, portrayal of electric energy consumers, detection of sensitive and influential buses in power systems, prediction of stock market, powerfully gathering clients dependent on their web get to designs, criminal examination. In spite of their higher speculation

capacity, strength to clamor and precision, neural systems have not been successfully utilized in data mining

2.1 Data mining challenges

Albeit conventional data mining procedures have made extraordinary progress, they likewise experienced viable troubles in meeting difficulties presented by new sort of datasets. Ache Ning Tan et al. had distinguished the accompanying testing issues: basically high dimensionality, heterogeneous and complex data and non-traditional analysis:

- **Heterogeneous and complex data**Conventional datasets frequently contain qualities of a similar sort, either consistent numerical or all out. As datasets and data mining applications in industry, business, science, engineering, bio-informatics, medication and different fields have developed, the requirement for strategies that can manage heterogeneous properties has likewise risen. Ongoing years have additionally observed the rise of increasingly complex data objects
- **Non-traditional analysis**The conventional breaking down methodology depends on a hypothesize-and-test worldview. At the end of the day, the examination structure and data analysis are both identified with the hypothesis. In any case, this procedure can be incredibly work escalated and could even quit working if the quantity of hypothesis and test worldview is in the thousands. Accordingly, there is no hypothesis-and-test system in the data mining and the procedure of hypothesis generation and assessment is consequently fused in the algorithms.
- **High Dimensionality**A data set may have hundreds or even a great many attributes (highlights) contrasted with a few a couple of decades prior. One outrageous exists in bio-informatics: microarray, consisting of an

array series of thousands of microscopic spots, each containing a particular DNA grouping, has empowered quality articulation in a huge number of highlights. For the majority of data analysis algorithms, the computational multifaceted nature increments exponentially as the dimensionality (the quantity of highlights) increments.

3. Artificial Neural Networks

Artificial neural systems (ANNs) give a general, handy strategy for adapting genuine esteemed, discrete-esteemed, and vector-esteemed capacities from precedents. Algorithms, for example, BACKPROPAGATION use slope plummet to tune arrange parameters to best fit a training set of input-output sets. ANN learning is strong to mistakes in the training data and has been effectively connected to issues, for example, translating visual scenes, speech recognition, and learning robot control techniques.

Neural systems (NNs) first began to be produced during the 1940s, spurred by, for instance, an endeavor to recreate the human brain on PCs. It has been effectively connected to prediction issues, design acknowledgment, and characterization and optimization issues.

Neural systems typically develop complex condition surfaces through rehashed emphases, each time modifying the parameters that characterize the surface. After much cycle, a surface might be "inside" characterized that approximates the focuses inside the dataset.

As a rule, ANNs consist of straightforward parallel interconnected and typically versatile preparing components or neurons whose availability mirrors the neurobiological framework. These neurons are normally appropriated more than a few layers. Forward Neural Network (FNN) is a standout amongst the most well known system topologies.

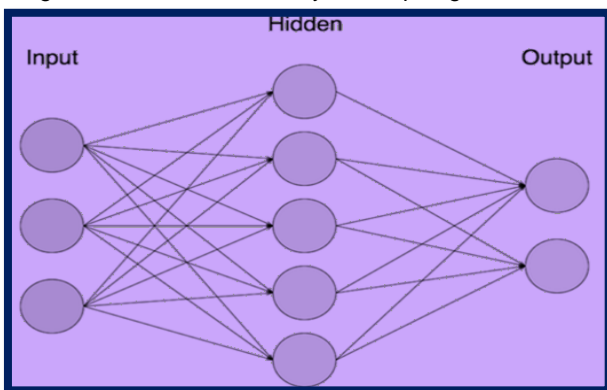


Figure 1 Artificial neural network

- (1) The input neurons get information as input values. They exchange the data to the following layer. Neurons in the input layer play out no neural capacity.
- (2) Hidden neurons get the output from the input neurons or other shrouded neurons through the associations.
- (3) Output layer neurons carry on correspondingly to the ones in the shrouded layer with the exception of that the example of the result from output neurons might be deciphered as the result of utilizing the NN to process the input data.

The investigation of artificial neural systems (ANNs) has been roused to some extent by the perception that natural learning frameworks are worked of complex snare of interconnected neurons. In unpleasant analogy, artificial neural systems are worked out of a thickly interconnected set of straightforward units, where every unit takes various genuine valued inputs (conceivably the outputs of different units) and produces a solitary genuine valued output (which may turn into the input to numerous different units)

The fundamental capacity of every neuron is to: (an) assess input values, (b) compute an aggregate for the consolidated input values, (c) contrast the aggregate and a threshold value and (d) figure out what its very own output would be. While the task of every neuron is genuinely straightforward, associating number of neurons together frequently makes complex structure and conduct.

At first the model is prepared utilizing the training set and after that used to make predictions. An approval set may likewise be utilized amid the training stage to check the advancement of the training. Every neuron as a rule has a set of loads that decide how it assesses the consolidated quality of the input signs. Inputs for a neuron might be either positive (excitatory) or negative (inhibitory). Learning happens by changing the loads utilized by the neuron as per order mistakes that were made by the system overall. The inputs are generally scaled and standardized to deliver a smooth conduct.

While ANNs are inexactly propelled by biological neural frameworks, there are numerous complexities to biological neural frameworks that are not demonstrated by ANNs, and numerous highlights of the ANNs we talk about here are known to be inconsistent with biological frameworks. For instance, we consider here ANNs whose singular units output a solitary consistent value, while biological neurons output a complex time arrangement of spikes.

3.1 Recent Subjects in Artificial Neural Networks

• Alternative Error Functions

As noted before, inclination drop can be performed for any capacity E that is differentiable concerning the parameterized hypothesis space. While the essential algorithm characterizes E regarding the whole of squared mistakes of the system, different definitions have been recommended so as to join different limitations into the weight-tuning rule. For each new meaning of E another weight-tuning rule for gradient descent must be determined.

• Alternative Error Minimization Procedures

While gradient descent is a standout amongst the most broad scan strategies for finding a hypothesis to limit the error function, it isn't generally the most proficient. It isn't unprecedented to require a huge number of emphases through the weight refresh circle when training complex systems. Thus, various elective weight optimization algorithms have been proposed and investigated. To see a portion of alternate conceivable outcomes, it is useful to think about a weight refresh strategy as including two decisions: picking a course in which to adjust the present weight vector and picking a separation to move.

One optimization strategy, known as line look, includes an alternate way to deal with picking the separation for the weight update. Specifically, when a line is picked that indicates the

course of the update; the update separate is picked by finding the base of the error function along this line. Notice this can result in an expansive or exceptionally little weight update, contingent upon the situation of the point along the line that limits error. A second technique that expands on line seek is known as the conjugate gradient strategy. Here, an arrangement of line looks is performed to scan for a base in the error surface. On the initial phase in this arrangement, the heading picked is the negative of the gradient. On each resulting venture, another heading is picked so the part of the error gradient that has quite recently been made zero stays zero. While elective error-minimization techniques in some cases lead to enhanced proficiency in training the system, strategies, for example, conjugate gradient will in general have no huge effect on the speculation error of the last system. The main likely effect on the last error is that diverse error-minimization strategies may fall into various local minima

• Recurrent Networks

So far we have considered just system topologies that compare to non-cyclic coordinated charts. Repetitive systems are artificial neural systems that apply to time arrangement data and that utilization outputs of system units at time t as the input to different units at time $t + 1$. Along these lines, they bolster a type of coordinated cycles in the system. To delineate, consider the time arrangement prediction errand of foreseeing the following day's stock market normal $y(t + 1)$ in light of the present day's financial pointers $x(t)$. Given a period arrangement of such data, one clear methodology is to prepare a feed forward system to anticipate $y(t + 1)$ as its output, in light of the input values $x(t)$.

• Dynamically Modifying Network Structure

So far we have considered neural system learning as an issue of adjusting weights inside a settled chart structure. An assortment of strategies have been proposed to powerfully develop or shrink the quantity of system units and interconnections trying to enhance speculation precision and training proficiency

A thought for progressively changing system structure is to adopt the contrary strategy. Rather than starting with the easiest conceivable system and including complexity, we start with a complex system and prune it as we locate that specific associations are inessential. One approach to choose whether a specific weight is inessential is to see whether its value is near zero. A second way, which gives off an impression of being increasingly fruitful practically, speaking, is to consider the impact that a little variety in the weight has on the error.

4. Issues In Data Mining Using Artificial Neural Networks

Neural systems have been effectively connected for wide assortment of issue areas to learn models that can perform intriguing assignments, for example, controlling a vehicle, perceiving qualities in DNA successions, planning payloads for space carry and foreseeing market patterns. Nonetheless, neural systems are not all around used in data mining applications. The significant reactions and somewhere else are:

a) Prepared neural systems are secret elements where the information learned is hidden in an extensive number of

associations. Accordingly people can barely use this information to help themselves in decision-production.

b) The business databases are generally exceptionally huge and updated much of the time and accordingly requests for gradual learning. It is felt that a neural system can't perform steady learning and in this way the prepared neural system can scarcely reflect current business viably.

c) Learning time of a neural system is normally long.

In any case, the specialists from the field of machine learning have created answers for every one of these challenges and in this way these disadvantages can be survived and neural systems should assume a huge job for productive or potentially successful data mining. This is on the grounds that, neural systems have demonstrated preferences now and again, for example, to give bring down prescient error rate than customary procedures. For a few issues, neural systems have an increasingly reasonable inductive inclination. For such issues, they complete a superior employment of learning the objective idea than other usually utilized data-mining strategies.

5. Supervised Instance Selection Algorithm (SIS)

The wrapper approach utilizes the strategy for arrangement as objective function to measure the significance of highlights set. Thus the element chose relies upon the classifier demonstrate utilized. In the wrapper approach, the component subset determination is finished by acceptance algorithm. The element subset choice algorithm leads a scan for a decent subset utilizing the induction algorithm as a piece of the assessment function. The reason for the induction algorithm is to instigate from training data a classifier that will be helpful in classifying future cases. The precision of prompted classifiers is evaluated by exactness estimation strategy. The characterization algorithm itself is utilized to decide the trait subset. Since the wrapper approach upgrades the assessment measure of the grouping algorithm while evacuating highlights, it for the most part prompts more prominent precision than the channel methodologies, for example, X2 Statistic, Information increase, Symmetrical vulnerability, Relief and Correlation based element determination. Wrapper strategies for the most part result in preferable performance over channel techniques in light of the fact that the element choice process is advanced by the grouping algorithm to be utilized

The wrapper-based algorithm called supervised instance selection (SIS) has been created and displayed in this thesis. The algorithm abuses promptly accessible values at the outputs of the neurons in the output layer of a prepared neural system (additionally called assurance of the system) to recognize and expel anomalies just as excess instances. It is autonomous of size of the training set, type and number of properties, nearness or nonattendance of missing values, number of classes, engineering of the system, sort of training algorithm utilized, sort of hardware utilized and sort of preprocessing assignments performed. It results into about 70% decrease away necessities for the models manufactured utilizing the instances chosen by SIS, when contrasted with the models worked without utilizing SIS, with a slight corruption in order exactness. The time complexity of SIS is $O(n)$, where n is the extent of the training set and it is similar to the next surely understood instance selection algorithms.

5.1 Time and Space Complexity of SIS

The primary concern performance measures for instance selection are time complexity and space complexity. Time spent on instance selection should, when all is said in done, be close to what a data mining algorithm requires. The space prerequisite for instance selection 61 ought not be as requesting as data mining. Instance selection comes from the requirement for decreasing the data, as it is simply an excessive amount to deal with in a few applications. All things considered, one can now and again loosen up the prerequisites on reality a little since instance selection is regularly performed once in a while. As it were, regardless of whether it requires comparative or higher time or space complexities as a data mining algorithm does, it might be as yet satisfactory. Time complexity of the proposed instance selection algorithm SIS is $O(n^2)$, (n indicates size of the training set) as it needs to sort training instances according to the classes and assurance values and in this manner limited by the span of the training set utilized by the base classifier. The time complexity is similar to that of the time required to prepare the model and furthermore instance selection algorithms DROP3, DROP5 and EXPLORE. The space complexity of the algorithm is $O(k)$.

5.2 Performance measurement

Being an instance selection algorithm, it is normal that the models constructed utilizing instances chosen by SIS for any dataset ought to give huge decrease away space without definitely influencing classification accuracy. Following measurements are regularly utilized for estimating the performance of instance selection algorithms like SIS:

- ✓ Average storage decrease,
- ✓ Average debasement in classification accuracy and
- ✓ Number of catastrophic failures.

The algorithm is relied upon to furnish huge capacity decrease with least debasement in accuracy when utilized with any sort of dataset and ought to be autonomous of nature of dataset, sort of preprocessing performed and sort of hardware utilized for mining. Albeit, a few procedures for instance selection as of now exists, models constructed utilizing instances chosen by SIS are relied upon to give better performance when contrasted with the models constructed utilizing instances chosen utilizing existing methods.

6. Constructing the compact and accurate classification models

Classification accuracy, time required for training the model and capacity necessities of the model are a portion of the issues in the design of a classifier. Classification is a supervised learning technique. In a supervised learning technique, a model is prepared utilizing accessible training precedents. Every one of the training models consists of values for input qualities and a class. A model is prepared utilizing instances from the training set and tried against the instances from the test set. When a model is prepared, it is utilized for classifying inconspicuous instances into one of the predefined classes. Accuracy of the models is measured as far as classification rate, for example number of unseen instances accurately grouped. Training time and amount of capacity required by a classification show relies on number of highlights (qualities) and training instances. Training time and capacity needs will be higher if there is progressively number of

properties and additionally training instances. Data decrease systems are regularly used to diminish training time and capacity needs of the model. The strategies perform highlight or potentially instance selection to diminish dimensionality of the input data that assistance in decreasing training time and capacity demands.

A neural system display is constructed in the wake of performing preprocessing and optional feature selection tasks. The model is then prepared and tried. When the model is prepared, the instance selection algorithm is executed to choose input instances utilizing sureness of the system in classifying every one of the training. When the instances are chosen, a model is retrained with the diminished set of instances and gives a smaller and accurate trained model.

6.1 Certainty of neural networks

The proposed algorithm utilizes the idea of certainty of the neural system for choosing a couple of training instances. Certainty of a neural system $y = f_i(x)$, as characterized is the likelihood that an instance x is in a class, at that point as y approaches 1, it is increasingly sure that the instance is in the class. As y approaches 0, it is increasingly sure that the instance isn't in the class. In this way, the certainty c (y) of a neural system output might be characterized as:

$$y \quad \text{if } y \geq 0.5 \quad (1)$$

$$c(y) = \begin{matrix} 1 - y & \text{if } y < 0.5 \end{matrix} \quad (2)$$

The certainty ascends for outputs $y < 0.5$ as y falls, and outputs $y > 0.5$ as y raises One can say that a system output y_1 relating to input instance i_1 is less sure than another system output y_2 for some instance i_2 (i_1 and i_2 belongs to the same class), is $c(y_1) < c(y_2)$. For instance, consider a neural system to take care of a classification issue with two classes, state, class 0 and class 1. Accepting the decision boundary at 0.5, and the system gives an output y between 0 - 0.5, for the instances that are arranged into, state, class 0 and produces output y between 0.5 - 1.0 for instances that are ordered into, state, class 1. Think about two instances i_1 and i_2 both have a place with a similar class, say, class 0 and y_1 and y_2 are the corresponding outputs of the system individually. In the event that y_1 is state 0.45 and y_2 is 0.03, one can say that the system output y_1 is less sure than the output y_2 , as $c(y_1) = 0.55$ and $c(y_2) = 0.97$. The proposed strategy utilizes certainty values for the selection of input instances so as to build compact and accurate models.

6.2 Class instance space

Instance space for a class might be generally partitioned into two regions: stable (focal) and unsteady (boundary) regions (37). Instances that lie in stable region are model or most regular instances of the class and instances having a place with precarious region lie on the boundary and might be treated as atypical or less run of the mill instances. On the off chance that the decision boundary is at 0.5 (for example the base certainty value for an effectively characterized instance), the class instance space can be isolated into a few regions dependent on the certainty of the network in classifying the instances For instance, the instance space for some class,

say class A, might be isolated into state, five regions: region 1, 2, 3, 4 and 5 utilizing the certainty values in the scope of [0.5, 0.6], [0.6, 0.7], [0.7, 0.8] [0.8, 0.9] and [0.9, 1.0] individually. Expecting the network outputs are in the interim [0.5, 1.0] for instances that are arranged into class An, instances from regions 5 are model or most commonplace instances while instances from region 1 are atypical or less run of the mill instances. Sizes of every one of these regions may not be equivalent and regularly the appropriation of the instances among these regions is skewed and hence may not be uniform.

6.3 Design issues

The instance selection algorithm must be nonexclusive so it very well may be utilized with any dataset, procedure and apparatus. Following design issues were considered while designing the proposed instance selection algorithm.

- The algorithm ought to be free of
- Nature of the dataset o Types of attributes o Number of attributes and instances o Number of classes o Presence or absence of missing values
- Internal architecture of the base classifier
- Preprocessing tasks
- Data mining tool

7. Comparison of SIS

7.1 Comparison of SIS with WEKA's unsupervised instance selection algorithm

WEKA gives an unsupervised instance selection algorithm (referred as WEKAUSU-MC in this thesis). It enables client to choose any of the accessible base classifier. It is helpful for the location of anomalies. The essential distinction between the two techniques is that 62 WEKA-USU-MC is unsupervised and center upon location and evacuation of exception instances, while as algorithm SIS utilizes supervised learning and spotlights on recognition of anomalies as well as chooses a small amount of instances from various regions of the class instance spaces. Consequently, the amount of storage reduction if there should be an occurrence of the models built utilizing instances chosen by WEKA-USU-MC channel relies upon number of exception instances in the training set; vast quantities of anomalies will result in higher storage reduction. Amount of storage reduction if there should be an occurrence of SIS depends on number of exceptions as well as the value of p . It is normal that the models constructed utilizing instances chosen by SIS should utilize lesser number of training instances than that of models constructed utilizing the instances chosen by WEKA-USU-MC without corrupting the classification accuracy. Subsequently, the comparison can be made between three neural network classifiers: classifier built utilizing all instances, for example with no instance selection, classifier built utilizing instances chosen by WEKA-USU-MC and the classifier built utilizing the instances chosen by SIS

One of the design issues in designing SIS is to design an instance selection algorithm to be methodology free, for example instances chosen by SIS utilizing neural network as a base classifier ought to likewise have the capacity to develop compact classifiers utilizing any procedure other than neural network classifiers, for example, decision tree or lazy classifiers. Hence the performance of the decision trees and lazy classifier models constructed utilizing instances chosen by SIS ought to be measured and contrasted and the decision

trees and lazy classifier models constructed utilizing every one of the instances. In light of the predisposition of every system, accuracy of the decision tree and lazy classifiers constructed utilizing instances chosen by SIS may debase

7.2 Comparison of SIS with iDA's supervised typicality-based instance selection algorithm

As idea of instance ordinariness can likewise be utilized to build compact models. Instance ordinariness of an instance as the average similitude of/to alternate individuals from its class highly run of the mill data instances inside a class are called class models and instances with low typicality scores speak to hopeful anomalies. It is seen that by choosing delegate training instances from a pool of accessible instances and restricting the incorporation of atypical instances, test set classification accuracy can be enhanced and results into compact supervised classification models. In this way there appears to have closeness between the idea of instance commonality and certainty values of the network. Instance classifiers, for example, ESX in the iDA instrument, registers 63 typicality of each training instance amid the model building process. Subsequently a comparison can be made between the neural network and lazy classifier models constructed utilizing all instances, and neural network and lazy models constructed utilizing instances chosen by SIS and instance ordinariness. Just stage 1 in the algorithm SIS is necessitated that sorts the records. When the records are arranged, wanted number of instances having higher certainty values is chosen from each class.

7.3 Comparison of SIS with DROP 3, DROP 5 and EXPLORE

As expressed before, a few instance selection algorithms have been accounted for in the writing that are utilized if there should arise an occurrence of instance-based or lazy classifiers, for example, DROP-3, DROP-5, EXPLORE. The conspicuous inquiry is the reason not to utilize these algorithms for building compact classifiers? one is that due to the inclination of every procedure, the accuracy of the compact models build utilizing algorithms, for example, DROP3, DROP5 and EXPLORE may debase when contrasted with the accuracy of the models constructed utilizing instances chosen by SIS. Exploratory comparison subsequently can affirm this hypothesis.

7.4 Comparison of SIS with sampling

Sampling strategies are quicker and versatile. In this manner, these strategies are regularly utilized for instance selection. Be that as it may, in light of their intrinsic randomness in choosing instances, accuracy of the models constructed utilizing such randomly chosen instances may debase with unseemly sample measure. Instance selection algorithms, for example, SIS that chooses instances dependent on specific criteria in this manner ought to give lesser corruption in accuracy. Nonetheless, such strategies have higher time complexities and may not be adaptable. Along these lines sampling techniques are quicker however the accuracy of the models constructed utilizing random sampling of instances with unseemly sample size may debase. The decision of choosing a procedure for instance selection, for example, sampling or SIS, along these lines might be a tradeoff

between training time and likelihood of debasement in classification accuracy and could be checked experimentally

8. Conclusion

Artificial neural network learning gives a practical method to adapting genuine valued and vector-valued functions over ceaseless and discrete-valued attributes, in a way that is strong to robust in the training data. The BACKPROPAGATION algorithm is the most widely recognized network learning strategy and has been effectively connected to an assortment of learning undertakings, for example, penmanship acknowledgment and robot control. When the instances are chosen, a model is retrained with the diminished set of instances that gives a compact and accurate model

SIS might be adjusted in order to naturally settle on number of instances to be chosen from every one of the classes from the training set by picking each effectively

characterized instance from each class with unmistakable certainty value, for example rather than picking a default value of 20% or enabling client to pick some other value, the algorithm can choose this value dependent on number of unmistakable certainty values of the base neural network for classifying every one of the training instances. The changed form in this way will choose lesser number of instances if there are less particular certainty values when contrasted with the quantity of instances chosen by SIS. With less number of particular certainty values, the changed variant may give higher storage reduction, conceivably without corrupting classifier accuracy. Over fitting the training data is a vital issue in ANN learning. Over fitting outcomes in networks that sum up inadequately to new data in spite of incredible performance over the training data. Cross-approval strategies can be utilized to appraise a suitable halting point for gradient plummet inquiry and consequently to limit the danger of over fitting.

References

- [1]. Kantardzic, M. (2002). Data Mining: Concepts, Models, Methods, and Algorithms: WileyIEEE Press.
- [2]. Tan, P.-N., Steinbach, M., & Kumar, V. (2005). Introduction to Data Mining, (First Edition): Addison-Wesley Longman Publishing Co., Inc.
- [3]. Cao, C. R., Wegman, E. J., & Solka, J. L. (2005). Handbook of Statistics: Data Mining and Data Visualization: Elsevier North-Holland.
- [4]. V. Rihani and S. Garg, "Neural networks for the prediction of stock market", IETE Technical Review, Vol. 23 No. 2, pp. 113-118, 2006 L. Yu, S. Wang, K. K. Lai, "An integrated data preparation scheme for neural network data analysis", IEEE Trans, on Knowledge and Data Engineering, Vol. 18, No. 2, pp. 217-230, 2006.
- [5]. Liu, H., & Motoda, H. (2007). Computational Methods of Feature Selection (Chapman & Hall/Crc Data Mining and Knowledge Discovery Series): Chapman & Hall/CRC
- [6]. Chow, T. W. S. (2007). Neural Networks and Computing: Learning Algorithms and Applications: World Scientific Publishing Co., Inc.
- [7]. H.J. Kim, J.H. Jang, H. Choi, T. Lee, J. Ko, M. Yoon and H.J. Kim. Inorg. Chem. 2008, 47, 2411-2415.
- [8]. JS. Patil, N. Subbareddy, "Neural network based system for script identification in Indian documents", Sadhana, Vol. 27, Part 1, pp. 83-97, 2009.
- [9]. S. Palaniappan, C. Saravanan. J. App. Poly. Sci. 2010, 118, 518-524.
- [10]. M. BalaMurali Krishna, V. Praveen Kumar, N. Venkatramaiah, R. Venkatesan and D. Narayana Rao. Appl. Phy. Lett. 2011, 98, 081106. (Virtual Journal of Nano scale Science & Technology, 2011, 23(9)).