A SURVEY ON SUPERVISED MACHINE LEARNING APPROACHES

[1]Y LATHA, [2]B SAI PRASAD

[1]Assistant professor, [2]Assistant professor
[1]Department of computer science and engineering,
[1]TKR College of Engineering and Technology, Hyderabad, Telangana, India

Abstract: This paper addresses the previous machine learning algorithms. Machine learning algorithms purpose like mining data, image processing, predictive analytics etc. main purpose of machine learning algorithm is, it learns what to do with data and works automatically.

Keywords: **Machine learning, deep learning, image processing.**

1. INTRODUCTION

Machine learning is used to learns machine with the systematic procedure, how to handle data more effectively and efficiently. Sometimes problems rise from, data interpretation and pattern extraction due to abundance of data set availability. Applications like many industries from medical filed to military activities machine learning is used to extract relevant information.
Previous techniques have done on how to machine understand by themselves. Mathematicians and programmers have been applied various approaches.
All the techniques of machine learning are explained in next section.

2. ISSUES OF SUPERVISED LEARNING ALGORITHMS

Learning from the past experiences is an attribute of humans while the computers do not have this ability. In supervised or Inductive machine learning, our main goal is to learn a target function that can be used to predict the values of a class. The process of applying supervised ML to a real-world problem is described in below figure.

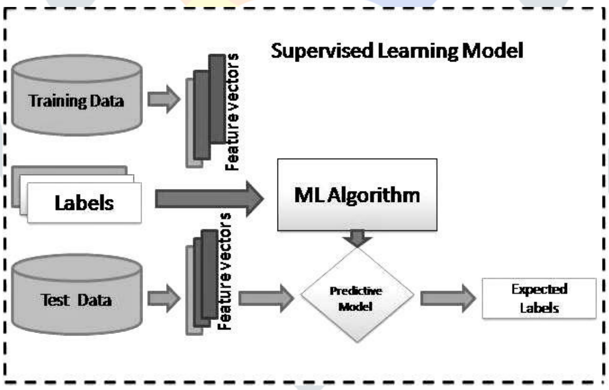


Fig.2. Supervised Machine Learning Model

In supervised learning the first step is dealing with dataset. In order to perform a better training on data set an appropriate expert could suggest better selection of features. If concerned expert is not in reach, then the other approach is "brute-force", which means measuring everything available in the hope that the right (informative, relevant) features can be isolated. However, a dataset collected by the "brute-force" method is not directly suitable for induction. Ultimately, in most cases it contains noise and missing feature values, and therefore requires significant pre-processing [1]. In the next step, data preparation and data preprocessing is a key function of researcher in Supervised Machine Learning (SML). A number of techniques have been introduced by different researchers to deal with missing data issue. Hodge & Austin [4] have conducted a survey of contemporary techniques for outlier (noise) detection. Karanjit

& Shuchita [5] have also discussed different outlier detection methods which are being used in different machine learning. H. Jair [6] has done comparison on 6 different outlier detection methods by performing experiment on benchmark datasets and a synthetic astronomical domain.

2.1 ALGORITHM SELECTION

The selection of algorithm for achieving good results is an important step. The algorithm evaluation is mostly judge by prediction accuracy. The classifier's (Algorithm) evaluation is most often based on prediction accuracy and it can be measured by given below formula

$$Accuracy = \frac{Number\ of\ Correct\ classifications}{Total\ number\ of\ test\ cases} \qquad (1)$$

There are number of methods which are being used by different researchers to calculate classifier's accuracy. Some researcher's splits the training set in such a way that, two-thirds retain for training and the other third for estimating performance. Cross-Validation (CV) or Rotation Estimation is another approach. CV provides a way to make a better use of the available sample. In k-fold cross-validation scheme, we divide the learning sample into k disjoint subsets of the same size, i.e.

$$ls = ls_{1}1 \cup ls_{2} \cup ls_{k} \qquad (2)$$

A model is then inferred by the learning algorithm from each sample $ls \backslash ls$, $i = 1,..,k$ and its performance is determined on the held out sample ls_i. Final performance is computed as the average performance over all these models. Notice that when k is equal to the number of objects in the learning sample, this method is called leave-one-out. Typically, smaller values of k (10 or 20) are however preferred for computational reasons [7].

The comparison between supervised ML methods can be done through to perform statistical comparisons of the accuracies of trained classifiers on specific datasets. For doing this we can run two different learning algorithms on samples of training set of size N, estimate the difference in accuracy for each pair of classifiers on a large test set[1]. For classification of data, a good number of techniques have been developed by researchers, such as logical statistics based techniques. In next sections, we will precisely discuss the most important supervised machine learning techniques, starting with logical techniques [1].

3. LOGIC BASED ALGORITHMS

In this section we will discuss two logical (symbolic) learning methods: decision trees and rule-based classifiers.

3.1 DECISION TREES

In machine learning domain the Decision Tree Induction [8, 9] is currently one of the most important supervised learning algorithms. In Artificial Intelligence (AI) field, Quinlan has contributed through his ID3 and C4.5 algorithms. C4.5 is one of the most popular and the efficient method in decision tree-based approach. Here C4.5 algorithm creates a tree model by using values of only one attribute at a time [10]. According to authors [7], the decision tree induction, which was initially designed to solve classification problems, has been extended to deal with single or multi-dimensional regression. The major benefits of decision trees are i) produce intensive results, ii) easy to understand, iii) and holds well-organized knowledge structure [28].

Decision Trees (DT) are trees that classify instances by sorting them based on feature values, where each node in a decision tree represents a feature in an instance to be classified, and each branch represents a value that the node can assume [1]. Instances are classified starting at the root node and sorted based on their feature values.

The Fig.3 is an example of a decision tree for the training set of Table.2. DT are extensively used is different computational fields to classify data. The reasons behinds the widely acceptability of DT learning algorithms are their flexibility to apply in wide range of problems. An interesting and important property of a decision tree and its resulting set of rules is that the tree paths or the rules are mutually exclusive and exhaustive. This means that every data instance/record/example/vector/case is covered by a single rule. According to Pierre et al. [7], DT algorithms combined with ensemble methods, can provide better results in terms of predictive accuracy and significantly in the context of high-throughput data sets, tree-based methods are also highly scalable from a computational point of view.

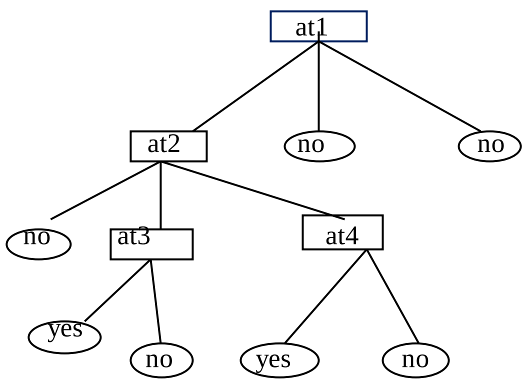


Fig.3. A Sample Decision Tree

By using the DT depicted in Fig.3 as an example, the instance (at1 = a1, at2 = b2, at3 = a3, at4 = b4) would sort to the nodes: at1, at2, and finally at3, which would classify the instance as being positive (represented by the values "Yes").

Table.2. Sample Training set

at1	at2	at3	at4	Class
a1	a2	a3	a4	Yes
a1	a2	a3	b4	Yes
a1	b2	a3	a4	Yes
a1	b2	b3	b4	No
a1	c2	a3	a4	Yes
a1	c2	a3	b4	No
b1	b2	b3	b4	No
c1	b2	b3	b4	No

The feature that best divides the training data would be the root node of the tree. There are different methods to extract the features that best divides the training data such as information gain [11] and gini index [12].

1. Check for base cases
2. For each attribute "a" calculate
   i. Normalized the information gain (IG) from splitting on attribute "a".
3. Find the best "a", attribute that has highest IG
4. Create a decision node: node that splits on best of "a"
5. Recurse on the sub-lists obtained by Splitting on a best and add those nodes as children of node

Fig.4. General pseudo-code for building decision trees

3.2 LEARNING SET OF RULES

It is also possible that decision trees can be translated into a set of rules by creating a separate rule for each path from the root to a leaf in the tree [13]. However, rules can also be directly induced from training data using a variety of rule-based algorithms. In [14], the author has provided an excellent overview of existing work in rule-based methods. The classification rules represent each class by Disjunctive Normal Form (DNF). A statement is in DNF if it is a disjunction (sequence of ORs) consisting of one or more disjuncts, each of which is a conjunction (AND) of one or more literals. Below is an example of disjunctive normal forms.

A k-DNF expression is of The form: $((A_1 \wedge A_2 \wedge A_n) \vee (A_{n+1} \wedge A_{n+2} \wedge A_{2n}) \vee \vee)$ $(A_{(k-1)n+1} \wedge A_{(k-1)n+2} \wedge \wedge A_{kn})$, where k is the number of disjunctions, n is the number of conjunctions in each disjunction, and A_n is defined over the alphabet $A_1, A_2, , A_j, \vartheta \sim A_1, \sim A_2 A_j$. Here the objective is to build the smallest rule-set that is consistent with the training data [1]. A good number of learned rules is usually a positive sign that the learning algorithm is attempting to remember the training set, instead of discovering the assumptions that govern it. A separate-and-conquer algorithm (recursively breaking down a problem into sub-problems) search for a rule that explains a part of its training instances, separates these instances and recursively conquers the remaining instances by learning more rules, until no instances remain [1]. In below Fig.5, a general pseudo-code for rule learners is presented.

1. Initialize rule set to a default
2. Initialize examples to either all available examples or all examples not correctly handled by rule set.
3. Repeat
   (a) Find best, the best rule with respect to examples.
   (b) If such a rule can be found
   i. Add best to rule set.
   ii. Set examples to all examples not handled correctly by rule set.
4. Until no rule best can be found

Fig.5. A general Pseudo code for rule learners

The core difference between heuristics for rule learning algorithms and heuristics for decision trees algorithms is that the latter evaluate the average quality of a number of disjointed sets, while rule learners only evaluate the quality of the set of instances that is covered by the candidate rule [1]. One of the most useful characteristic of rule based classifiers is their comprehensibility. In order to achieve better performance, even though some rule-based classifiers can deal with numerical features, some experts propose these features should be discredited before induction, so as to reduce training time and increase classification accuracy [15].

4. STATISTICAL LEARNING ALGORITHMS

Statistical learning is a framework for machine learning drawing from the fields of statistics and functional analysis [16].

Statistical learning theory deals with the problem of finding a predictive function based on data and it has a good number of applications in the field of AI. The major of goal of statistical learning algorithms is to provide a framework for studying the problem of inference that is obtaining knowledge, making predictions and making decision by constructing model from a set of data [17].

Bayesian networks are the most well-known representative of statistical learning algorithms. A good source for learning Bayesian Networks (BN) theory is [18], where readers can learn applications of BN.

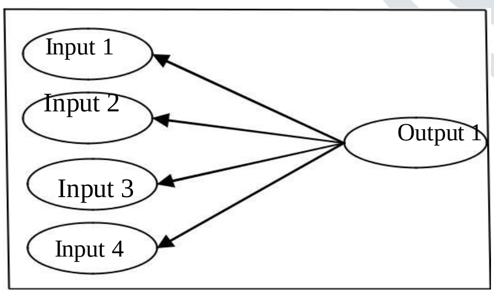
Statistical methods are characterized by having an explicit underlying probability model, which provides a probability that an instance belongs in each class, rather than simply a classification. Linear Discriminate Analysis (LDA),which was developed in 1936, and the related Fisher's linear discriminate are famous methods used in statistics and machine learning to retrieve the linear combination of features which best separate two or more classes of object [1]. The purpose of discriminate analysis is to classify objects (nations, people, customers…) into one of two or more groups based on set of features that describe the objects (e.g. gender, marital status, income, height, weight...). The another method for estimating probability distributions from data is maximum entropy. According to the base theory of maximum entropy, if nothing is known about a distribution except that it belongs to a certain class, then the distribution with the largest entropy should be chosen as the default.

4.1 NAIVE BAYES CLASSIFIERS

Bayesian networks are widely used to perform classification tasks. Naive Bayesian Networks (NBN) are very simple Bayesian networks which are composed of directed acyclic graphs with only one parent (representing the unobserved node) and several children (corresponding to observed nodes) with a strong assumption of independence among child nodes in the context of their parent [21]. According to author [20] the independence model (Naive Bayes) is based on estimating:

$$\frac{P(i|X)}{P(j|X)} = \frac{P(i)P(X|i)}{P(j)P(X|j)} = \frac{P(i)P\prod P(X_r|i)}{P(j)\prod P(X_r|j)} \qquad (3)$$

Here comparing these two probabilities, the larger probability indicates that the class label value that is more likely to be the actual label (if $R>1$: predict i else predict j) [1]. As shown in the below figure, the links in a Naive Bayes model are directed from output to input, which gives the model its simplicity, as there are no interactions between the inputs, except indirectly via the output.



An advantage of the Naive Bayes classifier is that it requires a small amount of training data to estimate the parameters necessary for classification.

4.2 BAYESIAN NETWORKS

Bayesian Networks (BN) are graphical models that are used to illustrate relationships between events or ideas to infer probabilities or uncertainties associated with those ideas or events. Information retrieval, predictions based on limited input or recognition software is some main applications of BN.

The Bayesian network structure S is a directed acyclic graph (DAG) and the nodes in S are in one-to-one correspondence with the features X. The arcs represent casual influences among the features while the lack of possible arcs in S encodes conditional independencies. Moreover, a feature (node) is conditionally independent from its non-descendants given its parents (X1 is conditionally independent from X2).

The below example shows that there are two events which could cause grass to be wet i.e. either the sprinkler is on or it's raining. Additionally here we also, suppose that the rain has a direct effect on the use of the sprinkler (namely that when it rains, the sprinkler is usually not turned on). Then the situation can be modeled with a Bayesian network. All three variables have two possible values, T (for true) and F (for false) [22].

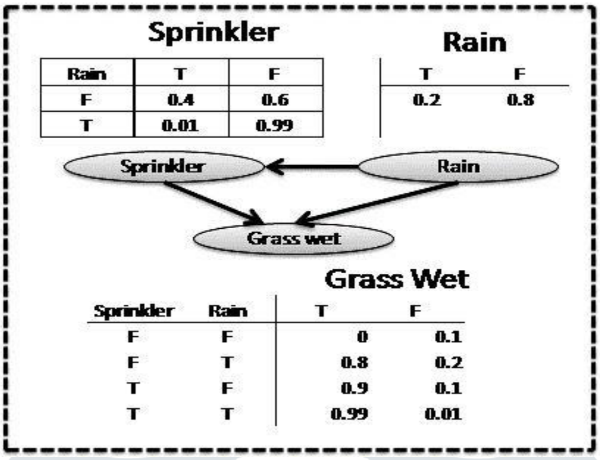


Fig.7. Bayesian network with conditional probability tables

The below is a joint probability function:

$$P(G, S, R) = P\big(G\ S, R\big)P\big(S\ R\big)P(R) \qquad (4)$$

where, the names of the variables have been abbreviated to:
G = Grass wet (yes/no)
S = Sprinkler turned on (yes/no)
R = Raining (yes/no). Cheng et al. draw the attention of a problem of BN classifiers that it is not suitable for datasets with many features. The reason for this is that trying to construct a very large network is simply not feasible in terms of time and space [23]. The pseudo code of training BN is shown in below figure:

```
Initialize an Empty Bayesian Network G containing n nodes (i.e., a BN with n nodes but no edges)
  1)  Evaluate the score of G: Score (G)
  2)  G' = G
  3)  for i = 1 to n do
  4)        for j = 1 to n do
  5)            if  i • j then
  6)            if there is no edge between the modes i and j in G● then
  7)            Modify  G' by adding an edge between the nodes i and j in G. Such that i is a parent of j: (i●j)
  8)            if the resulting G' is a DAG then
  9)            if (Score(G') > Score (G)) then
  10)           G = G'
  11)           end if
  12)           end if
  13)           end if
  14)           end if
  15)         G' = G
  16)           End for
  17)           End for
```

Fig.8. Pseudo-code for training of BN

5. INSTANCE-BASED LEARNING

About this learning scheme, the author [24] describes it as lazy-learning algorithms, as they delay the induction or generalization process until classification is performed. These algorithms require less computational time during the training phase than other eager-learning algorithms (such as decision trees, neural and Bayes nets) but need more computation time during the classification process. Nearest Neighbor algorithm is an example of instance-based learning algorithms [1]. Aha [25] and De et. al [26] discussed the instance-based learning classifiers.

k-Nearest-Neighbour (kNN) classification is one of the most widely used method for a classification of objects when there is little or no prior knowledge about the distribution of the data. KNN is a good choice to perform discriminate analysis when reliable parametric estimates of probability densities are unknown or difficult to determine [27].

kNN is a example of supervised learning algorithm in which the result of new instance query is classified based on majority of k-nearest neighbour category. The core function of algorithm is to classify a new object based on attributes and training samples. Here the classification is using majority vote among the classification of the *k* objects. For example we have conducted a survey on consumption of any particular item to know it's worth in the market. Below is a sample training table.

The outcome "Yes" or "No" is depended on the variable values of X1 and X2, so if we want to know the outcome of that combination which is not available in data table, for example, when x1 = 4, and x2 = 8 then without doing lengthy exercise of conducting surveys, we can predict the results by using kNN classification method.

The below pseudo code is an example for the instance base learning methods.

```
Procedure InstanceBaseLearner (Testing Instances)
for each testing instance
{

find the k most nearest instances of the training set according to a distance metric

Resulting Class: most frequent class label of the k nearest instances

}
```

Fig.9. Pseudo-code for instance-based learners

6. SUPPORT VECTOR MACHINES

Support Vector Machines (SVMs) are a set of supervised learning methods which have been used for classification, regression and outlier's detection. There are number of benefits for using SVM such as: i) It is effective is high dimensional space, ii) Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient, iii) It is versatile because holds different kernel functions can be specified for the decision function. Common kernels are provided, but it is also possible to specify custom kernels.

Most real-world problems involve non-separable data for which no hyper plane exists that successfully separates the positive from negative instances in the training set. One good solution to this inseparability problem is to map the data onto a higher dimensional space and define a separating hyper plane there. This higher-dimensional space is called the transformed feature space, as opposed to the input space occupied by the training instances [1].

Space in which the training set instance will be classified. Some new kernels are being proposed by researchers but given bellow is list of some popular kernels

- Linear: $K\left(X_i, X_j\right) = X_i^T X_j$
- Polynomial: $K\left(X_i, X_j\right) = \left(\gamma X_i^T X_j + r\right)^d, \gamma > 0$
- Radial Basis Function (RBF): $K\left(X_i, X_j\right) = \exp\left(-\gamma \left\| X_i - X_j \right\|^2\right), \gamma > 0$
- Sigmoid: $K\left(X_i, X_j\right) = \tanh\left(\gamma X_i^T X_j + r\right)$

Here γ, r and d are the kernel parameters. Where, X_i is a training vector and mapped into a high dimensional space by the function ϕ and $K\left(X_i, X_j\right) \equiv \phi\left(X_j\right)$ is known as kernel function.

7. DEEP LEARNING

The use of deep artificial neural networks has gain popularity for the last few years in pattern recognition and machine learning. Most of the popular Deep Learning Techniques are built from Artificial Neural Network (ANN). Deep learning can be defined as a model (e.g., neural network) with many layers, trained in a layer- wise fashion. Deep learning has had a tremendous impact on various applications such as computer vision, speech recognition, natural language processing [29], and crawling deep web [30]. Samy et al. [29] have discussed challenges and new applications of deep learning in their study.

The Table.4 summarizes the current progress in deep learning algorithms. It has been observed that different deep learning technologies [32-36] required huge computational resources to achieve significant results.

8. CONCLUSION

In this paper surveys on various machine learning algorithms. Now a days most of the areas using machine learning algorithms. In our paper we discussed most popular algorithms.

ACKNOWLEDGEMENT

Deep learning has also been successfully implemented in industry products that ultimately take advantage of the large volume of data. Top Information Technology (IT) companies like Microsoft, Google, Apple, Yahoo, Baidu, Amazon and Facebook, who collect and analyze massive amounts of data on a daily basis, have been investing a good share on finances on deep learning related projects. For example, Apple's Siri and Google Voice Search offer a wide variety of services including weather reports, sport news, answers to user's questions, and reminders etc., by utilizing deep learning algorithms [31]. Currently, these two applications support wide range spoken languages.

I would like to express my gratitude to my teacher, Dr. Wang Hongjun, whose expertise and guidance added considerably to my graduate experience. I appreciate his vast knowledge and his consistent assistance in completing this work. I would also like to thank the other PhD Scholars of my school, Mr. Amjad Ahmed, and Mr. Mehtab Afzal for the assistance they provided to understand machine learning. Very special thanks goes to Dr. Zhu Yan, without whose motivation and encouragement, I confess that it would be difficult for me to move forward in my PhD Program.

REFERENCES

[1] S. B. Kotsiantis, "Supervised Machine Learning: A Review of Classification Techniques", *Informatica*, Vol. 31, No. 3, pp. 249-268, 2007.

[2] James Cussens, "Machine Learning", *IEEE Journal of Computing and Control*, Vol. 7, No. 4, pp 164-168, 1996.

[3] Richard S. Sutton and Andrew G. Barto, "*Reinforcement Learning: An Introduction*", Cambridge, MA: MIT Press, 1998.

[4] Victoria J. Hodge and Jim Austin, "A Survey of Outlier Detection Methodologies", *Artificial Intelligence Review*, Vol. 22, No. 2, pp. 85-126, 2004.

[5] Karanjit Singh and Shuchita Upadhyaya, "Outlier Detection: Applications and Techniques", *International Journal of Computer Science Issues*, Vol. 9, Issue. 1, No. 3, pp. 307-323, 2012.

[6] Hugo Jair Escalante, "A Comparison of Outlier Detection Algorithms for Machine Learning", *CIC-2005 Congreso Internacional en Computacion-IPN*, 2005.

[7] Pierre Geurts, Alexandre Irrthum, Louis Wehenkel, "Supervised learning with decision tree-based methods in computational and systems biology", *Molecular BioSystems*, Vol. 5, No. 12, pp. 1593-1605, 2009.

[8] L. Breiman, J. Friedman, R. A. Olsen and C. J. Stone, "Classification and Regression Trees", *Belmont, California: Wadsworth International Group*, 1984.

[9] J. Quinlan, "*C4.5: Programs for machine learning*", San Francisco, CA: Morgan Kaufmann, 1986.

[10] Masud Karim and Rashedur M. Rahman, "Decision Tree and Naïve Bayes Algorithm for Classification and Generation of Actionable Knowledge for Direct Marketing", *Journal of Software Engineering and Applications*, Vol. 6, No. 4, pp. 196-206, 2013.

[11] Earl B. Hunt, Janet Marin and Philip J. Stone, "*Experiments in Induction*", New York: Academic Press, 1966.

[12] Leo Breiman, Jerome Friedman, Charles J. Stone and R. A. Olshen, "*Classification and Regression Trees (Wadsworth Statistics/Probability)*", Chapman and Hall/CRC, 1984.

[13] Steven L. Salzberg, "Book Review: C4.5: Programs for Machine Learning by J. Ross Quinlan. Inc., 1993", *Machine Learning*, Vol. 16, No. 3, pp. 235-240, 1994.

[14] Johannes Fürnkranz, "Separate-and-Conquer Rule Learning", *Artificial Intelligence Review*, Vol. 13, pp. 3-54, 1999.

[15] Aijun An and Nick Cercone, "Discretization of continuous attributes for learning classification rules", *Third Pacific-Asia Conference on Methodologies for Knowledge Discovery & Data Mining*, Vol. 1574, pp. 509-514, 1999.

[16] Mehryar Mohri, Afshin Rostamizadeh and Ameet Talwalkar, "*Foundations of Machine Learning*", One Rogers Street Cambridge MA: The MIT Press, 2012.

[17] Olivier Bousquet, St´ephane Boucheron and G´abor Lugosi, "Introduction to Statistical Learning Theory", *Lecture Notes in Computer Science*, Vol. 3176, pp. 175-213, 2004.

[18] Olivier Pourret, Patrick Naim and Bruce Marcot, "*Bayesian Networks: A Practical Guide to Applications*", Wiley Publishers, 2008.

[19] Kamal Nigam, John Lafferty and Andrew McCallum, "Using Maximum Entropy for Text Classification", *Workshop on Machine Learning for Information Filtering*, pp. 61-67, 1999.

[20] N. J. Nilsson, "*Learning Machines: Foundations of Trainable Pattern-Classifying Systems*", First Edition, New York: McGraw-Hill, 1965.

[21] Isidore Jacob Good, "*Probability and the Weighing of Evidence*", The University of Wisconsin - Madison: Charles Griffin, 1950.

[22] Shiliang Sun, Changshui Zhang and Guoqiang Yu, "A Bayesian Network Approach to Traffic Flow Forecasting", *IEEE Transactions on Intelligent Transportation Systems*, Vol. 7, No. 1, pp. 124-132, 2006.

[23] Jie Cheng, Russell Greiner, Jonathan Kelly, David Bell and Weiru Liu, "Learning Bayesian networks from data: An information-Theory based approach", *The Artificial Intelligence Journal*, Vol. 137, pp. 43-90, 2002.

[24] Tom M. Mitchell, "*Machine Learning: A Guide to Current Research*", The Springer International Series in Engineering and Computer Science Series, McGraw Hill, 1997.

[25] D. Aha, “*Lazy Learning*”, Dordrecht: Kluwer Academic Publishers, 1997.

[26] Ramon Lopez De Mantaras and Eva Armengol, “Machine learning from examples: Inductive and Lazy methods”, *Data and Knowledge Engineering*, Vol. 25, No. 1-2, pp. 99-123, 1998.

[27] Hamid Parvin, Hoseinali Alizadeh and Behrouz Minati, “A Modification on K-Nearest Neighbor Classifier”, *Global Journal of Computer Science and Technology*, Vol. 10, No. 14 (Ver.1.0), pp. 37-41, 2010.

[28] Yen-Liang Chen and Lucas Tzu-Hsuan Hung, “Using decision trees to summarize associative classification rules”, *Expert Systems with Applications*, Vol. 36, No. 2, Part 1, pp. 2338-2351, 2009.

[29] Samy Bengio, Li Deng, Hugo Larochelle, Honglak Lee, and Ruslan Salakhutdinov, “Guest Editors’ Introduction: Special Section on Learning Deep Architectures”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 35, No. 8, pp. 1795-1797, 2013.

[30] Qinghua Zheng, Zhaohui Wu, Xiaocheng Cheng, Lu Jiang and Jun Liu, “Learning to crawl deep web”, *Information Systems, Vol.* 38, No. 6, *pp.* 801-819, 2013.

[31] Xue-Wen Chen and Xiaotong Lin,” Big Data Deep Learning: Challenges and Perspectives”, *IEEE Access Practical Innovations: Open Solutions and Access and IEEE*, Vol. 2, pp. 514-525, 2014.

[32] Rajat Raina, Anand Madhavan and Andrew Yg, “Large-scale Deep Unsupervised Learning using Graphics Processors”, *26th International Conference on Machine Learning*, pp. 609-616, 2009.

[33] Alex Krizhevsky, Ilya Sutskever and Geoffrey E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks”, *Advances in Neural Information Processing System*, pp. 1106-1114, 2012.

[34] Jeffrey Dean, Greg S. Corrado and Rajat Monga Kai, “Large Scale Distributed Deep Networks”, *Advances in Neural Information Processing System*, pp. 1232-1240, 2012.

[35] Quoc V. Le, Marc’Aurelio Ranzato, Rajat Monga, Matthieu Devin, Kai Chen, Greg S. Corrado, Jeffrey Dean, and Andrew Y. Ng, “Building High-level Features Using Large Scale Unsupervised Learning”, *Proceedings of the 29th International Conference on Machine Learning*, 2012.

[36] A. Coats and B. Huval, “Deep Learning with COTS HPS systems”, *Journal of Machine Learning Research*, Vol. 28, No. 3, pp. 1337-1345, 2013.