IJCSE International Journal of Computer Sciences and Engineering Open Access

Research Paper Volume-5, Issue-10 E-ISSN: 2347-2693

Designing a Classifier Using Unsupervised Learning and Rough Set Theory

Vairaprakash Gurusamy[1*], K. Nandhini[2]

[1*]Department of Computer Applications, School of IT, Madurai Kamaraj University, Madurai, India
[2]Technical Support Engineer, Concentrix India Pvt Ltd, Chennai, India

*Corresponding Author: vairaprakashmca@gmail.com

Available online at: www.ijcseonline.org

Received: 17/Sep/2017, Revised: 30/Sep/2017, Accepted: 13/Oct/2017, Published: 30/Oct/2017

Abstract--- Dataset collected from multiple sources is often inconsistent and generates different label of decisions for the same conditional attribute values. A method for handling inconsistency has been proposed here using Kohonen Self organizing neural network, an unsupervised learning approach. After removing inconsistency, the minimum subset of attributes in the dataset called reducts are selected using Rough Set Theory, which effectively reduces dimensionality of the dataset. Unlike most of the existing reduct generation algorithms where all attributes are examined, here evaluation of all attributes is not required and therefore, time complexity has been improved considerably. In the next step, considering core attribute as root node of a decision tree, all possible rules are generated which are pruned based on information entropy and coverage of the rule set. The classifier is built using the reduced rule set demonstrating comparable results with the classifier consisting of all attributes.

Keywords— Inconsistency, Rough Set, Unsupervised Neural Network

I. Introduction

Nowadays data management in the World Wide Web considers very large Knowledge Database (KDB) which has little possibility of being consistent. The existing consistency checking algorithms and systems fail to analyze KDBs of such a huge size. Moreover, modern medical decision processes are generally based on patient data collected from different sources which are archived by a multi terminal or distributed computer systems. Error occurs due to handling of such data by different people and/or instruments which directly record the data. Detection of data inconsistencies at the global level of every

Patient record can help in tracing systematic as well as spurious errors in the data acquisition process. In 1982, Pawlak[1] introduced theory of Rough Sets, a new mathematical tool for handling vague and inconsistent data sets. As Rough Set theory considers data dependency solely based on data, many researchers tried to investigate attribute dependency in algebraic aspects [2], or in statistical aspects [3]. ROSETTA [4] and RSES are some data mining tools that try to find decision rules from databases. Due to time complexity there is some size limitation of input data for the systems. Rough Set Theory is used by some researchers by combining it with other well known theories. Ytow et al. [5] combined formal concepts having objects and attributes with rough sets to have upper and lower approximations, and Guo and Tanaka [6] showed similarity between possibility theory and rough set theory. In the paper, instead of set approximation approach, a scheme for removing

inconsistency in data sets has been proposed using Kohonen Self organizing Map [7]. The conditional attribute values of the inconsistent objects are replaced by the trained weights connected to the winning node, corresponding to a decision attribute value. As a next step the reducts of the consistent decision table is evaluated by employing a tree whose root node is formed taking all the condition attributes. The attribute dependency between condition and decision attributes are evaluated by gradually removing one attribute at a time and expanding the tree. If the dependency of the new set of attributes is same as that of the root then there is a possibility of finding reducts in that path, otherwise that path is aborted. The process is repeated unless all the branches from the root node is traversed. This gives the minimum set of attributes of the information system. Unlike the Quick reduct algorithm [8] here removal and evaluation for all attributes are not required and therefore, time complexity is improved considerably. A new classification scheme based on decision tree algorithm is proposed taking core as root of the tree and generating rules from the core. The paper is divided into seven sections. In the second section the fundamentals of Rough Set Theory has been presented. The subsequent sections deal with Inconsistency Removal Algorithm, Reduct generation, Building of classification Rules, Experimental results and Conclusions respectively.

II. Rough set theory basics

The fundamental concepts of Rough Sets are given below:

A. Knowledge Base

In Rough Set Theory an information system consisting of rows (objects) and columns (attributes), is represented as $I = (U, A)$ where U is a nonempty finite set of objects and A is a nonempty finite set of attributes such that $a{:}U \rightarrow V_a$ for every $a \in A$. V_a is the set of values that attribute a may take. A decision system is defined as $D=\{A \cup Q\}$ where A is the set of conditional attributes and Q is the set of decision attributes.

It is of interest here to find the minimum subset of attributes which would generate the same equivalence classes as all the attributes taken together. In order to find such minimum attributes the Indiscernibility relation of the attributes is to be evaluated.

B. Indiscernibility Relation

Let $P \subseteq A$ be a subset of attribute. A binary relation IND(P), called the Indiscernibility relation, is defined as follows:
$\text{IND}(P) = \{(x, y) \in U^2 \mid \forall a \in P, a(x) = a(y)\}$ for which two objects (x and y) are equivalent if and only if they have the same attribute values with respect to attributes in P. The partition of U, is determined by IND(P) and is denoted by U/IND(P).

The definition of Rough Set is derived from the concept of inconsistency in information system I. Let X be a target set of objects. In general X cannot be expressed exactly, because the set may include and exclude objects which are indistinguishable on the basis of attributes of P. Hence the concept of lower and upper approximations is useful.

C. Lower and Upper Approximations

Lower approximation is a set of objects that are known with certainty to belong to the set of interest, say, X with respect to the attribute subset say, E and defined as $\underline{E}X=\{x \in U \mid [x]_E \subseteq X\}$. Upper approximation is a description of the objects that possibly belong to the subset say, E and defined as $\bar{E}X=\{x \in U \mid [x]_{E \cap X \neq \phi}\}$. $[x]_E$ denotes the equivalence class of objects with respect to E. The boundary region is the difference between the upper and lower approximation sets ($BN_E(X) = \bar{E}X - \underline{E}X$), if the boundary region is nonempty the set is rough not exact. The Lower Approximation is also known as positive region denoted as $POS_E(X)$.The lower approximation of condition attributes with respect to decision attributes helps to detect whether the information system is consistent or not. The inconsistency of the information system can be dealt with, after its detection.

After removing the inconsistency from the data set it is required to evaluate the minimum attribute set which would generate the same equivalence classes as the overall condition attributes taken together. This minimum attribute set is called the reduct, which is defined below.

D. Reduct and Attribute Dependencies

Reduct is the minimal set of attributes which represent the information table by maintaining the indiscernibility relation. For data analysis another important concept of RST is to find out dependencies between the attributes. If a set of attributes Q depends on a set of attributes P, dependency $\gamma_P(Q)$ is defined with a degree k $\left(0 \le k \le 1\right)$, denoted as $P \Rightarrow_k Q$, where $k=\gamma_P(Q)=\sum \underline{P}Q_i$ and $i = 1..N$, number of values of attribute Q. A reduct K is defined as a subset of minimal cardinality K_{min} of the conditional attribute set A such that $\gamma_K(Q) = \gamma_A(Q)$, where Q is the decision attribute.

III. INCONSISTENCY REMOVAL ALGORITHM

Let Q be the decision attributes and A be the conditional attributes, and $\gamma_Q(A) \neq 1$ indicates that the information table is inconsistent. A data set is inconsistent when the attributes values for any two instances are same, but the decision attribute values are different. The inconsistency is removed by applying unsupervised learning approach using Kohonen Self Organizing Map(SOM).

SOM is a kind of nonlinear unsupervised learning technique developed in 1980 [7] which simulate basic characteristics of human brain. There is a mapping relation of input space and output space using continuous learning, adjusting weights and generates trained weight sets to classify objects. In the paper, the Kohonen network is trained with the training data sets consisting of inconsistent attribute values obtained from the decision table. Number of input nodes of the network equal to the number of conditional attributes while the numbers of output nodes equal to the different decision attribute values for which the data set is inconsistent. During training, the weights which are closest to the training data set is adjusted according to the learning rule. After training, the trained weight sets are substituted in place of the conditional attribute values of one of the inconsistent object for which the difference between the calculated output at any output node and the corresponding decision attribute value is minimum. Hence, the new conditional attribute values don't deviate much from their original values, maintaining integrity of the data set and at the same time remove inconsistency.

The algorithm for identifying and removing inconsistency in a data set using unsupervised learning is given below:-

```
Algorithm: (Input: Inconsistent Data Set,
            Output: Consistent Data Set)
  1. Input a Training data Set corresponding to the
     inconsistent conditional attribute values of an object to
     the Kohonen Self Organizing Map neural network.
```

2. Adjust the weights according to the learning rule.
3. Repeat step1 and step2 until the weights are saturated.
4. Replace the inconsistent conditional attribute values of one of the inconsistent objects, by trained weights of the winning node, for which the difference between the calculated output at any output node and the corresponding decision attribute value is minimum. The decision attribute value of that object should correspond to that of the winning node.

IV. Reduct generation

After removing the inconsistency from the data set it is required to evaluate the minimum attribute set which would generate the same equivalence classes as the overall condition attributes taken together. This minimum attribute set is called the reduct, the reducts of the decision table is evaluated using tree approach. The root node consists of all the condition attributes. Then gradually one of the attributes is taken off and subsequently the dependency of remaining attributes is evaluated. If the dependency is same to that of root node then there is a possibility of reduct in that path otherwise, that path is aborted and the attribute is replaced.

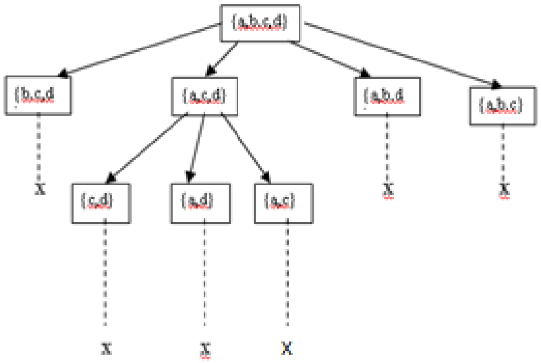


Figure 1: Showing Reduct Generation

The figure above gives an insight into the algorithm. Here assume {a, b, c, d} to be condition attributes, which is kept at root node. Its dependency is evaluated. Next the attribute 'a', is removed and the dependency of the attributes {b, c, d} is evaluated. Assume the dependency is not equal to that of the root node; hence that path is aborted as indicated by (X). Then remove attribute 'b' and assume that the dependency of remaining attributes is same as that of root node, so proceed further in that path and check the dependency of {c,d},{a,d},{a,c}. Assume the dependency of these attributes is not equal to that of the root node, so abort the process and {a,c,d} is a reduct. Further it is assumed that the dependency of {a,b,d}and{a,b,c} are not equal to that of root node, the path is aborted.

Algorithm:

i) Evaluate the Indiscernibility of Decision attributes
$U/IND_{\{Q\}}$.
ii) Evaluate $U/IND_{\{A\}}$ and $POS_A(Q)$.
iii) Calculate the dependency for root node
iv) Remove one attribute and evaluate the dependency of remaining attribute set.
v) If the dependency is same as that of root node then proceed further and repeat step 4, else abort the path.

Steps iii) to v) is repeated until all the branches from root node have been traversed.

This method generates the same reduct set as the Quick attribute reduction method [16] but it is more comprehensible than the Quick reduct method, moreover in this approach there is no need to traverse the path whose dependency is not equal to that of the root node so the complexity is effectively reduced. The next step is to look for minimum reducts and it forms the basis of the classifier. The core of the reduct is also evaluated.

V. Building of classification rules

The classification rules are generated using decision tree classification taking core of the reducts as root. A Tree Classification algorithm is used to compute a decision tree. The most important feature of decision tree classifier is their capability to break down a complex decision-making process into a collection of simpler decisions, thus providing a solution which is often easier to interpret. Decision trees are easy to understand and modify, and the model developed can be expressed as a set of decision rules. This algorithm scales well, even where there are varying numbers of training examples and considerable numbers of attributes in large databases. The core forms the root of the decision tree. The core is then split into its different values. The other attributes of the reduct are then split at the next level. Finally, the rules are generated by traversing from root to the leaf node. Initially the data is split into training and testing data set. The system is first trained on training data set and the rules are generated. Then the system is tested on testing data set. The accuracy of the classifier is then evaluated.

Algorithm: Function tree
Input: (C: set of condition attributes, D: a set of decision attributes, S: training data set)

Begin

- If S is empty, return a single node with value Failure;
- If S consists of records all with the same value for the target attribute, return a single leaf node with that value;
- Evaluate the core A of the reduct
- Let {aj| *j*=1,2, .., m} be the values of attribute A;
- Let {Sj|*j*=1,2, .., m} be the subsets of S Consisting respectively of records with value aj for A;
- Return a tree with root labeled A and arcs

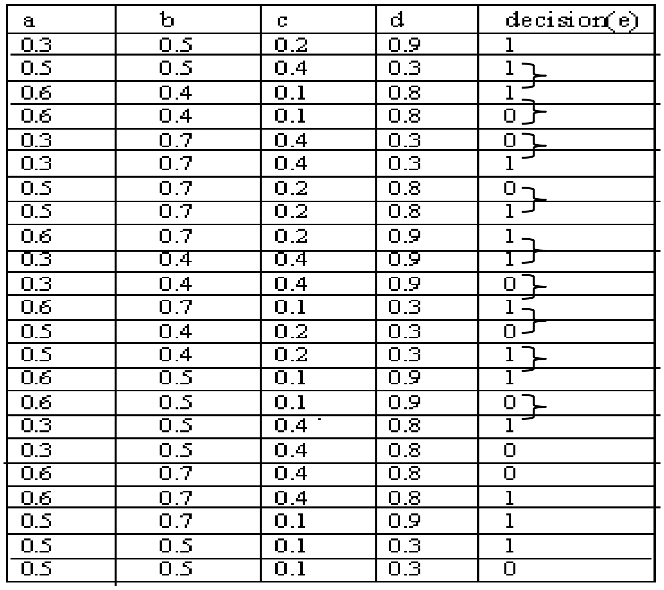
```
        labeled a1, a2, .., am going respectively to the
        trees (tree(R-{A}, D, S1), tree(R-{A}, D, S2),
        .....,tree(R-{A}, D, Sm);
    •   Recursively apply tree to subsets {Sj| j=1,2, .., m}
        until they are empty
End
```

Since in general more than one reduct is generated so number of rules are large which are required to be pruned for building the efficient classifier. The rules are pruned using coverage and Information Entropy Info$_A$(D)=|D_j|/|D| Info(D_j), where D is the set of all tuples of data set, D_j is the tuple with value j and A is the attribute of the reduct. Coverage is the number of objects covered by the rule.

Finally those rules are selected whose coverage is greater than a selected threshold and Information Entropy is less. Thus significant rules are only included in designing the classifier.

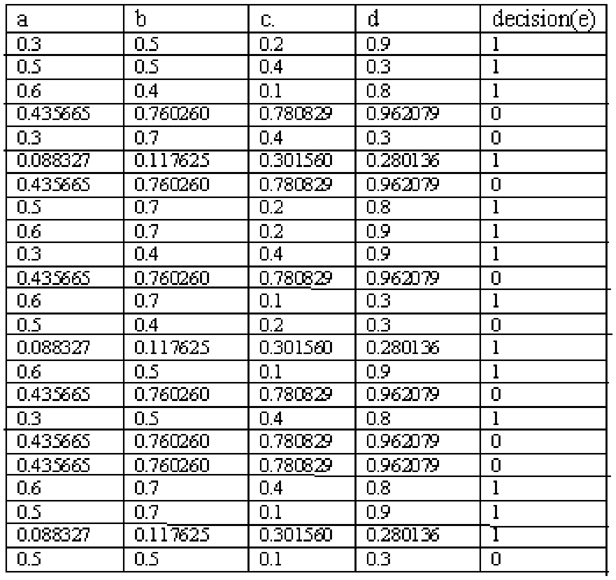
Table 1(a) below shows inconsistent data set. The data set contains {a,b,c,d} as condition attributes and{e} as decision attributes. The inconsistent objects of the table(3,4),(5,6),(7,8),(10,11),(13,14),(15,16),(17,18),(19,20),(22,23) are marked. The inconsistent attribute value was replaced with the trained weights of the Neural Network and the consistent data set obtained is shown in table 1(b). The consistent data set is then discretized and the discretized data set is shown in table 1(c).

Table: 1(a) Inconsistent data set



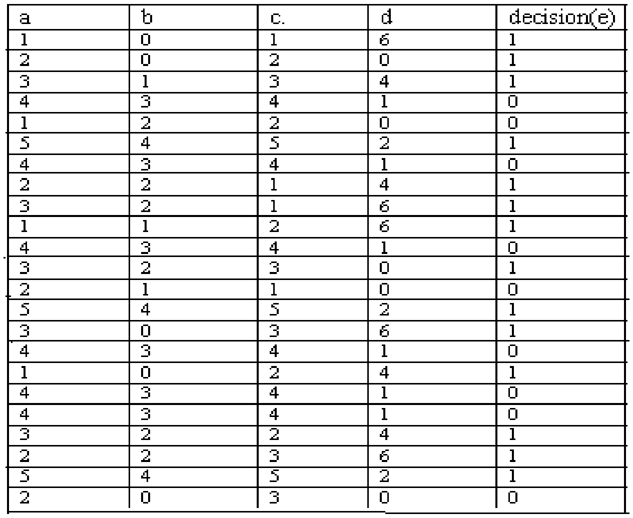
a	b	c	d	decision(e)
0.3	0.5	0.2	0.9	1
0.5	0.5	0.4	0.3	1
0.6	0.4	0.1	0.8	1
0.6	0.4	0.1	0.8	0
0.3	0.7	0.4	0.3	0
0.3	0.7	0.4	0.3	1
0.5	0.7	0.2	0.8	0
0.5	0.7	0.2	0.8	1
0.6	0.7	0.2	0.9	1
0.3	0.4	0.4	0.9	1
0.3	0.4	0.4	0.9	0
0.6	0.7	0.1	0.3	1
0.5	0.4	0.2	0.3	0
0.5	0.4	0.2	0.3	1
0.6	0.5	0.1	0.9	1
0.6	0.5	0.1	0.9	0
0.3	0.5	0.4	0.8	1
0.3	0.5	0.4	0.8	0
0.6	0.7	0.4	0.8	0
0.6	0.7	0.4	0.8	1
0.5	0.7	0.1	0.9	1
0.5	0.5	0.1	0.3	1
0.5	0.5	0.1	0.3	0

Table: 1(b) Consistent Data set



a	b	c.	d	decision(e)
0.3	0.5	0.2	0.9	1
0.5	0.5	0.4	0.3	1
0.6	0.4	0.1	0.8	1
0.435665	0.760260	0.780829	0.962079	0
0.3	0.7	0.4	0.3	0
0.088327	0.117625	0.301560	0.280136	1
0.435665	0.760260	0.780829	0.962079	0
0.5	0.7	0.2	0.8	1
0.6	0.7	0.2	0.9	1
0.3	0.4	0.4	0.9	1
0.435665	0.760260	0.780829	0.962079	0
0.6	0.7	0.1	0.3	1
0.5	0.4	0.2	0.3	0
0.088327	0.117625	0.301560	0.280136	1
0.6	0.5	0.1	0.9	1
0.435665	0.760260	0.780829	0.962079	0
0.3	0.5	0.4	0.8	1
0.435665	0.760260	0.780829	0.962079	0
0.435665	0.760260	0.780829	0.962079	0
0.6	0.7	0.4	0.8	1
0.5	0.7	0.1	0.9	1
0.088327	0.117625	0.301560	0.280136	1
0.5	0.5	0.1	0.3	0

Table: 1(c) Discretized table



a	b	c.	d	decision(e)
1	0	1	6	1
2	0	2	0	1
3	1	3	4	1
4	3	4	1	0
1	2	2	0	0
5	4	5	2	1
4	3	4	1	0
2	2	1	4	1
3	2	1	6	1
1	1	2	6	1
4	3	4	1	0
3	2	3	0	1
2	1	1	0	0
5	4	5	2	1
3	0	3	6	1
4	3	4	1	0
1	0	2	4	1
4	3	4	1	0
4	3	4	1	0
3	2	2	4	1
2	2	3	6	1
5	4	5	2	1
2	0	3	0	0

VI. EXPERIMENTAL RESULTS

The algorithm is implemented on a sample inconsistent data set shown in table 1(a). The inconsistent attribute values were replaced with the trained weights of the Neural Network and the consistent data set is shown in table 1(b). The consistent data set is then discretized and its reducts are evaluated. The discretized data set is shown in table1(c) is further used to generate the classification rules. The accuracy of the classifier shown in Fig.2 demonstrates better performance than existing algorithms.

© 2017, IJCSE All Rights Reserved

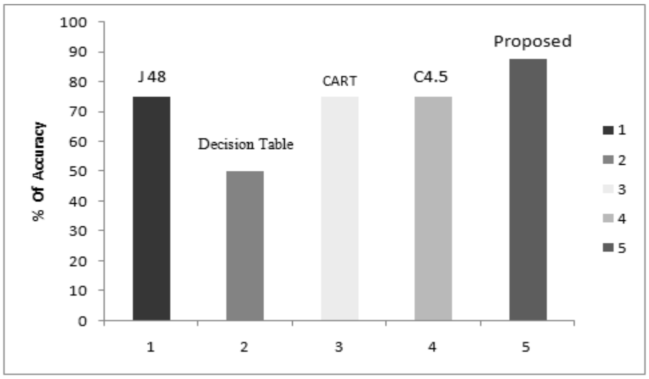


Figure 2: Showing accuracy of different classifiers

VII. CONCLUSION

The algorithm combines the efficiency of Rough set and Decision Tree induction. Inconsistency handling was the key issue which is resolved using SOM. The classification scheme can efficiently classify the new consistent data set with a minimum set of rules. The accuracy of the new classification scheme is better than other algorithms and comparable to that of Naïve Bayes classifier.\

VIII. REFERENCES

[1]. Zdzislaw Pawlak, "*Rough sets"*, International Journal of Computer and Information Sciences, 11, 341-356, 1982.

[2]. I. Düntsch, and G. Gediga, "*Algebraic aspects of attribute dependencies in information systems*", Fundamental Informaticae, Vol. 29, 1997, pp. 119-133.

[3]. A. Øhrn, "*Discernibility and rough sets in medicine: tools and applications*", PhD thesis, Department of Computer and information science, Norwegian University of Science and Technology, 1999.

[4]. J.G. Bazan, M.S. Szczuka, and J. Wroblewski, "*A new version of rough set exploration system*," Lecture notes in artificial intelligence, Vol.2475, 2002, pp. 397-404.

[5]. N.Ttow, D.R. Morse, and D.M. Roberts, "*Rough set approximation as formal concept*," Journal of advanced computational intelligence and intelligent informatics, Vol.10, No.5, 2006, pp. 606-611.

[6]. P.Guo, and H. Tanaka, "*Upper and lower possibility distributions with rough set concepts*,", In Rough set theory and granular computing, Springer, 2002, pp. 243-250.

[7]. Kohonen, T. "*Self-Organizing Maps*", 3rd edition, Berlin: Springer-Verlag, 2001.

[8]. Zhangyan Xu, Liyu Huang, Wenbin Qian, Bingru Yang, "*Quick Attribute Reduction Algorithm Based on Improved Frequent Pattern Tree*".

[9]. Ganesan G, Raghavendra Rao C., Latha D., "*An overview of rough sets"*, proceedings of the National Conference on the emerging trends in Pure and Applied Mathematics, Palayamkottai, India, pp: 70-76, 2005

[10]. [Han, 2001] Han J and Kamber M, "*Data Mining: Concepts and Techniques*", Morgan Kaufmann, 2001, 279-325.

[11]. C.R.Rao and P.V.Kumar. "*Functional Dependencies through Val"*,

[12]. ICCMSC '99, India, TMH publications 116-123, 1999.

[13]. Pawlak,1991] Zdzislaw Pawlak, "*RoughSets- Theoretical Aspects and Reasoning about Data"*, Kluwer Academic Publications, 1991

[14]. Quinlan J.R. "*Induction of decision trees*". Machine Learning 181-106.

[15]. Ramadevi Y, C.R.Rao, *"Knowledge Extraction Using Rough Sets –Gpcr – Classification*",International conference on Bioinformatics and diabetes mellitus, India, 2006.

[16]. [[Starzyk, 1999] Starzyk J, Nelson D.E., SturtzK, "*Reduct Generation in Information Systems"*,Bulletin of International Rough Set Society, 3(1/2), 1999.

Authors Profile

Mr. Vairaprakash Gurusamy pursed MCA from Bharathidasan University, Trichy in 2010. He is currently pursuing Ph.D from Madurai Kamaraj University, Madurai, India. He has published more than 10 research papers in reputed international journals like Scopus Indexed, UGC approved, SCI Indexed, Web of Science, Thomson Reuters etc. His main research work focuses on Bid Data Analytics, Distributed System, Artificial Intelligence, NLP, Cloud Computing, Data Mining and IOT. He has 3 years of Industry Experience and 4 years of Research Experience.

Ms. K. Nandhini pursed B.Tech (ECE) from Kalasalingam University, Krishnan Kovil, India in 2014. She has published more than 10 research papers in reputed international journals like Scopus Indexed, UGC approved, SCI Indexed, Web of Science, Thomson Reuters etc. His main research work focuses on Bid Data Analytics, Distributed System, Artificial Intelligence, NLP, Cloud Computing, Data Mining and IOT. He has 3 years of Industry Experience.