ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 2, August 2012

A Novel Approach for WEKA & Study On Data Mining Tools

M.Vijayakamal, Mulugu Narendhar

***Abstract* - Mining tools to solve large amounts of problems such as classification, clustering, association rule, neural networks, it is a open access tools directly communicates with each tool or called from java code to implement using this. In this paper we present machine learning data mining tool used for different analysis, Waikato Environment for Knowledge Analysis is introduced by university of New Zealand it has the capacity to convert CSV file to Flat file. Our work shows the process of WEKA analysis of file converts and selection of attributes to be mined and comparison with Knowledge Extraction of Evolutionary Learning not only analysis the data mining classifications but also the genetic, evolutionary algorithms is the best efficient tool in learning.**

***Keywords* – Data Mining Tools, Classification Techniques, Machine Learning, WEKA,KEEL.**

I. INTRODUCTION

The Waikato Environment for Knowledge Analysis (Weka) is a machine learning toolkit introduced by Waikato University, New Zealand. It is open source software written in Java (GNU Public License) and used for research, education and Projects. It can be run on Windows, Linux and Mac. It consists of collection of machine learning algorithms for implementing data mining tasks. GUI (data visualization) based tool mainly used for comprehensive set of preprocessing tools, evaluation methods and has an environment for comparing learning techniques. There are several versions of Weka like Weka 3.0 “book version” compatible with description in data mining book. WEKA 3.2: “GUI version” adds graphical user interfaces (book version is command-line only). WEKA 3.3: “development version” with lots of improvements. This talk is based on the latest snapshot of WEKA 3.5.This article gives a comparative study of open source tools of data mining available in the market and focuses on the vital role of Weka in comparison with other tools and its implementation in the real world scenario.

II. DATA MINING TOOLS

Open data mining tools are used to solve the data mining classification problems, here we discuss on classification, clustering, association, regression and mining tools the process of extracting patterns from data is called data mining. It is recognized as an essential tool by modern business since it is able to convert data into business intelligence thus giving an informational edge. At present, it is widely used in profiling practices, like surveillance, marketing, scientific discovery, and fraud detection.

A. Association Rule

Given a set of transactions, where each transaction is a set of items, an association rule [1] is an expression X/Y where X and Y are sets of items. The intuitive meaning of such a rule is that the transactions that contain the items in X tend to also contain the items in Y. An example of such a rule might be that 30% of transactions that contain beer also contain diapers; 2% of all transactions contain both these items". Here 30% is called the confidence of the rule, and 2% the support of the rule. The problem of mining association rules is to find all rules that satisfy a user-specified minimum support and minimum confidence. The association rule mining problem can be decomposed into two sub problems [2]: Find all combinations of items, called frequent item-sets, whose support is greater than minimum support and time consuming. Use the frequent item sets to generate the desired rules. The idea is that if, say, ABCD and AB are frequent, then the rule AB/CD holds if the ratio of support(ABCD) to support(AB) is at least as large as the minimum confidence. Note that the rule will have minimum support because ABCD is frequent.

B. Classification

In Classification, training examples are used to learn a model that can classify the data samples into known classes. The Classification process involves following steps:

a. Create training data set.
b. Identify class attribute and classes.
c. Identify useful attributes for classification (Relevance analysis).
d. Learn a model using training examples in Training set.
e. Use the model to classify the unknown data samples.

1. Decision Tree

Decision trees are a way of representing a series of rules that lead to a class or value. For example, you may wish to classify loan applicants as good or bad credit risks. Figure 7 shows a simple decision tree that solves this problem while illustrating all the basic components of a decision tree: the decision node, branches and leaves.

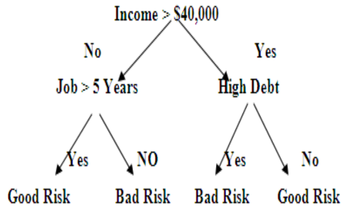


Fig: 1 Shows the Example of Decision Tree

The first component is the top decision node, or root node, which specifies a test to be carried out. The root node in this example is “Income > $40,000.” The results of this test cause the tree to split into branches, each representing one of the possible answers. In this case, the test “Income > $40,000” can be answered either “yes” or “no,” and so we get two branches. Depending on the algorithm, each node may have two or more branches. For example, CART generates trees with only two branches at each node. Such a tree is called a binary tree. When more than two branches are allowed it is called a multi way tree.

C. Clustering Technique

Cluster is a number of similar objects grouped together. It can also be defined as the organization of dataset into homogeneous and/or well separated groups with respect to distance or equivalently similarity measure. Cluster is an aggregation of points in test space such that the distance between any two points in cluster is less than the distance between any two points in the cluster and any point not in it. There are two types of attributes associated with clustering, numerical and categorical attributes. Numerical attributes are associated with ordered values such as height of a person and speed of a train. Categorical attributes are those with unordered values such as kind of a drink and brand of car. Clustering is available in flavors of

- Hierarchical
- Partition (non Hierarchical)

***1 Hierarchical*2:** In hierarchical clustering the data are not partitioned into a particular cluster in a single step. Instead, a series of partitions takes place, which may run from a single cluster containing all objects to n clusters each containing a single object [12]. Hierarchical Clustering is subdivided into *agglomerative* methods, which proceed by series of fusions of the n objects into groups, and *divisive* methods, which separate n objects successively into finer groupings for the partitional can be of K-means [15] & K-mediod.

2. K-Mean Algorithm:

1. Select k centroids arbitrarily (called as seed as shown in the figure) for each cluster C_i , i ε
[1, k]

2. Assign each data point to the cluster whose centroid is closest to the data point.

3. Calculate the centroid C_i of cluster $C_{i,}$ i ε [1, k] In short

4. Repeat steps 2 and 3 until no points change between clusters. A major disadvantage of K means is that one must specify the clusters in advance and further the algorithm is very sensitive of noise, mixed pixels and outliers. The definition of means limit the application to only numerical variables. We choose k-means because it is data driven with relatively few assumptions on the distributions of underlying data.

D. Regression:

It is a data mining function that predicts a number. Profit, sales, mortgage rates, house values, square footage, temperature, or distance could all be predicted using regression techniques. For example, a regression model could be used to predict the value of a house based on location, number of rooms, lot size, and other factors.

A regression task begins with a data set in which the target values are known. For example, a regression model that predicts house values could be developed based on observed data for many houses over a period of time. In addition to the value, the data might track the age of the house, square footage, number of rooms, taxes, school district, proximity to shopping centers, and so on. House value would be the target, the other attributes would be the predictors, and the data for each house would constitute a case. In the model build (training) process, a regression algorithm estimates the value of the target as a function of the predictors for each case in the build data. These relationships between predictors and target are summarized in a model, which can then be applied to a different data set in which the target values are unknown. Regression models are tested by computing various statistics that measure the difference between the predicted values and the expected values. The historical data for a regression project is typically divided into two data sets: one for building the model, the other for testing the model. Regression modeling has many applications in trend analysis, business planning, marketing, financial forecasting, time series prediction, biomedical and drug response modeling, and environmental modeling.

III. DM SOFTWARE

The five best open source data mining software that are Orange, Rapid Miner, Weka, JHepWork, and KNIME.

1.Orange

It is a component-based data mining and machine learning software suite that features friendly yet powerful, fast and versatile visual programming front-end for explorative data analysis and visualization, and Python bindings and libraries for scripting. It contains complete set of components for data preprocessing, feature scoring and filtering, modeling, model evaluation, and exploration techniques. It is written in C++ and Python, and its graphical user interface is based on cross-platform Qt framework.

2.RapidMiner

formerly called as YALE (Yet Another Learning Environment), is an environment for machine learning and data mining experiments that is utilized for both research and real-world data mining tasks. It enables experiments to be made up of a huge number of arbitrarily nestable operators, which are detailed in XML files and are made with the graphical user interface of Rapid Miner. Rapid Miner provides more than 500 operators for all main machine learning procedures, and it also combines learning schemes and attribute evaluators of the Weka learning environment. It is available as a stand-alone tool for data analysis and as a data-mining engine that can be integrated into your own products.

3.Weka

Written in Java, Weka (Waikato Environment for Knowledge Analysis) is a well-known suite of machine learning software that supports several typical data mining tasks, particularly data preprocessing, clustering, classification, regression, visualization, and feature selection. Its techniques are based on the hypothesis that the data is available as a single flat file or relation, where each data point is labeled by a fixed number of attributes. Weka provides access to SQL databases utilizing Java Database Connectivity and can process the result returned by a database query. Its main user interface is the Explorer, but the same functionality can be accessed from the command line or through the component-based Knowledge Flow interface.

4.JHepWork

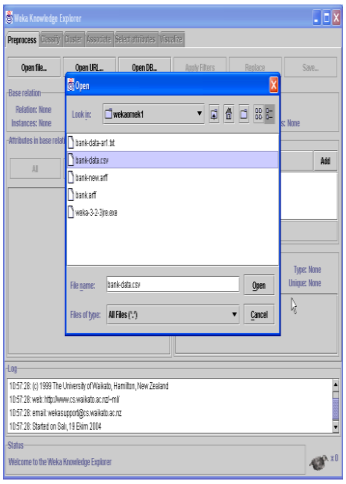
Designed for scientists, engineers and students, it is a free and open-source data-analysis framework that is created as an attempt to make a data-analysis environment using open-source packages with a comprehensible user interface and to create a tool competitive to commercial programs. It is specially made for interactive scientific plots in 2D and 3D and contains numerical scientific libraries implemented in Java for mathematical functions, random numbers, and other data mining algorithms. jHepWork is based on a high-level programming language Jython, but Java coding can also be used to call jHepWork numerical and graphical libraries.

5.KNIME

KNIME (Konstanz Information Miner) is a user friendly, intelligible, and comprehensive open-source data integration, processing, analysis, and exploration platform. It gives users the ability to visually create data flows or pipelines, selectively execute some or all analysis steps, and later study the results, models, and interactive views. KNIME is written in Java, and it is based on Eclipse and makes use of its extension method to support plugins thus providing additional functionality. Through plugins, users can add modules for text, image, and time series processing and the integration of various other open source projects, such as R programming language, Weka, the Chemistry Development Kit, and LibSVM.

IV. IMPLEMENTATION

WEKA has the capacity to read in ".csv" format files is fortunate since many databases or spreadsheet applications can save or export data into flat files in this format can be seen in the sample data file, the first row contains the attribute names (separated by commas) followed by each data row with attribute values listed in the same order (also separated by commas). In fact, once loaded into WEKA, the data set can be saved into ARFF format. Interested in converting a ".csv" file into WEKA's native ARFF, then the recommended approach is to use the following from the command line: **java weka.core.converters.CSVLoader filename.csv > filename.arff** Load the data set into WEKA, perform a series of operations using WEKA's attribute and discretization filters, and then perform association rule mining on the resulting data set. While all of these operations can be performed from the command line, we use the GUI interface for WEKA Knowledge Explorer. Initially (in the Preprocess tab) click "open" and navigate to the directory containing the data file (.csv or .arff). In this case we will open the above data file. This is shown in Figure 1.



Base relation
Relation: None
Instances: None

File name: bank-data.csv

Type: None
Unique: None

Log

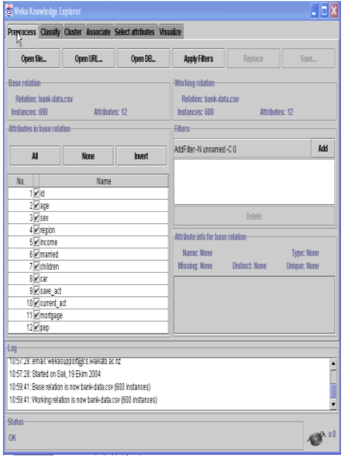
```
10:57:28: (c) 1999 The University of Waikato, Hamilton, New Zealand
10:57:28: web: http://www.cs.waikato.ac.nz/~ml/
10:57:28: email: wekasupport@cs.waikato.ac.nz
10:57:28: Started on Salı, 19 Ekim 2004
```

Status
Welcome to the Weka Knowledge Explorer

Fig 1. Screen 1- Loading the Data into WEKA

A. Choosing The Data From File

After data is loaded, WEKA will recognize the attributes and during the scan of the data will compute some basic statistics on each attribute. The left panel in Figure 2 shows the list of recognized attributes, while the top panels indicate the names of the base relation (table) and the current working relation.



Base relation
Relation: bank-data.csv
Instances: 600 Attributes: 12

Working relation
Relation: bank-data.csv
Instances: 600 Attributes: 12

Attributes in base relation

No.		Name
1	☑	id
2	☑	age
3	☑	sex
4	☑	region
5	☑	income
6	☑	married
7	☑	children
8	☑	car
9	☑	save_act
10	☑	current_act
11	☑	mortgage
12	☑	pep

Filters
AddFilter -N unnamed -C 0

Attribute info for base relation
Name: None Type: None
Missing: None Distinct: None Unique: None

Log

```
10:57:28: email: wekasupport@cs.waikato.ac.nz
10:57:28: Started on Salı, 19 Ekim 2004
10:59:41: Base relation is now bank-data.csv (600 instances)
10:59:41: Working relation is now bank-data.csv (600 instances)
```

Status
OK

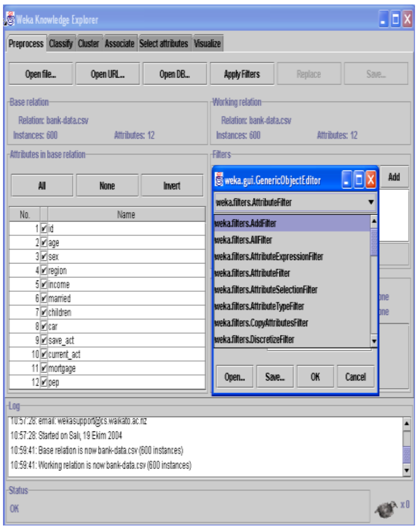
Fig 2. Screen 2 – Choosing the Data into File

Clicking on any attribute in the left panel will show the basic statistics on that attribute. For categorical attributes, the frequency for each attribute value is shown; while for continuous attributes we can obtain min, max, mean, standard deviation, etc.

B. Prepare the Data to Be Mined

1. Selecting Attributes

From sample data file, each record is uniquely identified by a customer id, need to remove this attribute before the data mining step and using the Attribute filter in WEKA. In the "Filters" panel, click on the filter button (to the left of the "Add" button). This will show a popup window with a list available filters. Scroll down the list and select "weka.filters.AttributeFilter" as shown in Figure 3.



Base relation — Relation: bank-data.csv, Instances: 600, Attributes: 12

Working relation — Relation: bank-data.csv, Instances: 600, Attributes: 12

No.		Name
1	☑	id
2	☑	age
3	☑	sex
4	☑	region
5	☑	income
6	☑	married
7	☑	children
8	☑	car
9	☑	save_act
10	☑	current_act
11	☑	mortgage
12	☑	pep

weka.filters.AttributeFilter

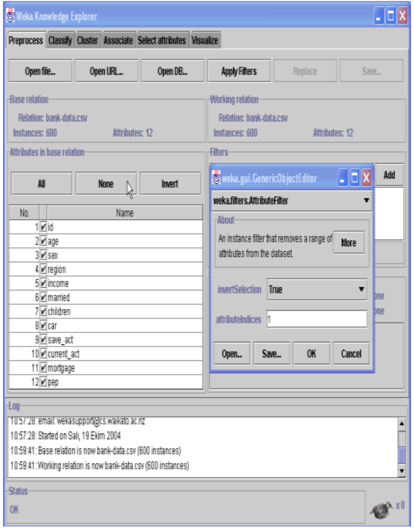
- weka.filters.AddFilter
- weka.filters.AllFilter
- weka.filters.AttributeExpressionFilter
- weka.filters.AttributeFilter
- weka.filters.AttributeSelectionFilter
- weka.filters.AttributeTypeFilter
- weka.filters.CopyAttributesFilter
- weka.filters.DiscretizeFilter

```
10:57:28: email: wekasupport@cs.waikato.ac.nz
10:57:28: Started on Salı, 19 Ekim 2004
10:59:41: Base relation is now bank-data.csv (600 instances)
10:59:41: Working relation is now bank-data.csv (600 instances)
```

Status: OK

Fig 3. Screen3 – Shows The Mining Process Of Selecting Attributes.

The resulting dialog box enter the index of the attribute to be filtered out (this can be a range or a list separated by commas). In this case, enter 1 which is the index of the "id" attribute (see the left panel). Make sure that the "invert Selection" option is set to false (otherwise everything except attribute 1 will be filtered). Then click "OK" (See Figure 4).



Base relation — Relation: bank-data.csv, Instances: 600, Attributes: 12

Working relation — Relation: bank-data.csv, Instances: 600, Attributes: 12

No.		Name
1	☑	id
2	☑	age
3	☑	sex
4	☑	region
5	☑	income
6	☑	married
7	☑	children
8	☑	car
9	☑	save_act
10	☑	current_act
11	☑	mortgage
12	☑	pep

weka.filters.AttributeFilter

An instance filter that removes a range of attributes from the dataset

invertSelection: True

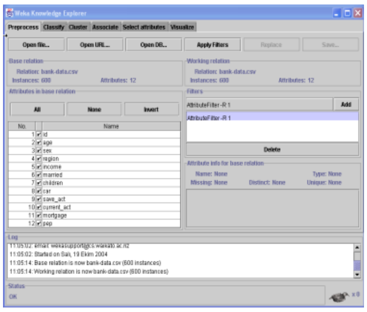
attributeIndices: 1

```
10:57:28: email: wekasupport@cs.waikato.ac.nz
10:57:28: Started on Salı, 19 Ekim 2004
10:59:41: Base relation is now bank-data.csv (600 instances)
10:59:41: Working relation is now bank-data.csv (600 instances)
```

Status: OK

Fig 4. Screen 4 Shows The Invert Selection Process.

In the filter box you will see "Attribute Filter -R 1". Click the "Add" button to add this to the selected list.



Base relation — Relation: bank-data.csv, Instances: 600, Attributes: 12

Working relation — Relation: bank-data.csv, Instances: 600, Attributes: 12

No.		Name
1	☑	id
2	☑	age
3	☑	sex
4	☑	region
5	☑	income
6	☑	married
7	☑	children
8	☑	car
9	☑	save_act
10	☑	current_act
11	☑	mortgage
12	☑	pep

Filters: AttributeFilter -R 1

AttributeFilter -R 1

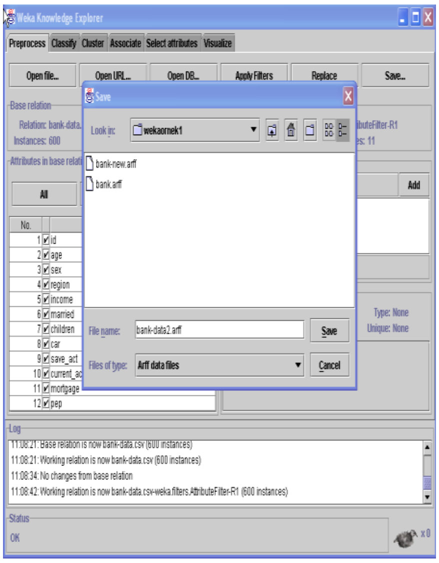
Name: None Type: None
Missing: None Distinct: None Unique: None

```
11:05:02: email: wekasupport@cs.waikato.ac.nz
11:05:02: Started on Salı, 19 Ekim 2004
11:05:14: Base relation is now bank-data.csv (600 instances)
11:05:14: Working relation is now bank-data.csv (600 instances)
```

Status: OK

Fig 5. Screen 5 – Represents Attributes is to be filtered

Finally, click the button "Apply Filters" on the top panel to apply the filter to the current working relation. You will notice that the "working relation" has now changed to the resulting data set containing the remaining 11 attributes. Note that it is possible to select several filters and apply all of them at once. However, in this example we will apply the different filters step-by-step. Also, it is possible now to apply additional filters to the new working relation. In this example, however, we will save our intermediate results as separate data files and treat each step as a separate WEKA session. To save the new working relation as an ARFF file, click on save button in the top panel. Here, as shown in the "save" dialog box below, save the new relation in the file "bank-data2.arff".



Relation: bank-data.
Instances: 600

ibuteFilter-R1
es: 11

No.		
1	☑	id
2	☑	age
3	☑	sex
4	☑	region
5	☑	income
6	☑	married
7	☑	children
8	☑	car
9	☑	save_act
10	☑	current_ac
11	☑	mortgage
12	☑	pep

Look in: wekaornek1

- bank-new.arff
- bank.arff

File name: bank-data2.arff

Files of type: Arff data files

Type: None
Unique: None

```
11:08:21: Base relation is now bank-data.csv (600 instances)
11:08:21: Working relation is now bank-data.csv (600 instances)
11:08:34: No changes from base relation
11:08:42: Working relation is now bank-data.csv-weka.filters.AttributeFilter-R1 (600 instances)
```

Status: OK

Fig 6 Screen 6 –Shows the ARFF Dataset Result save Process

V. COMPARATIVE ANALYSIS

Data mining tools are open source java software to assess evolutionary algorithms for problems in mining include data classification techniques. Knowledge extraction based on evolutionary learning contains collection of knowledge extraction algorithms preprocessing technique such as training selection feature discretization imputation methods for missing values.

Computational intelligence based learning algorithms including evolutionary rule learning based on different approaches and hybrid models such as genetic fuzzy systems evolutionary neural networks Main feature of KEEL developed to ensemble different data mining models of evolutionary learning algorithms with open source code in java which includes data preprocessing in specialized data transformation discretization training feature selection imputation methods for missing values and noisy data. KEEL provides an user friendly interface oriented to the analysis of algorithms to create in online that education supports to learn the operation of the algorithm and different evolutionary rule learning models have been implemented fuzzy rule with good trade-off between accuracy and interpretability, genetic programming algorithms that use tree representations for extracting knowledge and based on patterns subgroups discovery have been integrated to reduce the data evolutionary algorithms have included. Compare Kowledge extraction based evolutionary learning WEKA is collection of machine learning directly applied to dataset or called from java code. Weka helps in realizing goal of Data Mining, in this case by predicting missing values and validating that the predicted values are indeed correct. Weka Algorithms can be used via its API to build custom tools, applications and algorithms as well. WEKA system has been able to implement and evaluate a number of different Algorithms for different steps in the machine learning process. Output information provided by the package is sufficient for an expert in machine learning and results as displayed by the system show a detailed description of the flow and the steps involved in the entire machine learning process. The outputs provided by different algorithms are easy to compare and hence make the analysis easier.

ARFF dataset is one of the most widely used data storage formats for research databases, making this system easier for use in research oriented projects. This package provides and number of application program interfaces (API) which help novice Data miners build their systems using the "core WEKA system". First, major disadvantage is that the system is a Java based system and requires Java Virtual Machine installed for its execution. Since the system is entirely based on Command Line parameters and switches, it is difficult for an amateur to use the system efficiently. A Textual interface and output makes it all the more difficult to interpret and visualize the results. Important results such as the pruned trees, hierarchy based outputs cannot be displayed making it difficult to visualize the results. Although a commonly used dataset, ARFF is the only format that the WEKA system supports and all the current version i.e. 3.0.1 has some bugs or disadvantages, the developers are working on a better system and have come up with a new version which has a graphical user interface making the system complete.

VI. CONCLUSION

In this paper a general data mining tools for analysing classification, clustering, association rule, neural networks datasets and an explanation mechanism to explain the novel results was described. The specific approaches of mining tools learning are characterized, we developed the WEKA method is based on choosing the file and selecting attributes to convert .csv file to flat file and discussed features of KEEL and WEKA performance. Our work extends to utilize the implementation of dataset for each data mining tools present in the section III to achieve a high rate of accuracy in the case of unknown attacks in human-understandable can also improve the efficiency when analysing the complex dataset.

REFERENCES

[1] W. Lee, S. J. Stolfo Data Mining Approaches for Intrusion Detection.

[2] An Implementation of ID3: Decision Tree Learning Algorithm Wei Peng, Juhua Chen and Haiping Zhou Project of Comp 9417: Machine Learning University of New South Wales, School of Computer Science & Engineering, Sydney, NSW 2032, and Australia.

[3] D.P. Greene and S.F. Smith, Competition-based induction of decision models from examples, Machine Learning 3 (1993) 229-257.

[4] J. Bacardit and J.M. Garrell, "Bloat control and generalization pressure using the minimum description length principle for a Pittsburgh approach learning classifier system", in Advances at the frontier of Learning Classifier Systems, LNCS Vol. 4399 (2006) 61-80.

[5] www.junauza.com/2010/11/free-data-mining software.html

Author's Profile

M.Vijayakamal B.E CSE from Gulbarga University M.Tech Software Engineering from JNTU Hyderabad currently he working as Assoc Prof Department of IT in SriDevi Women's Engineering College. He is having 10 years of Academic Experience and member in Computer society of India & attended National Conference by Mahatma Gandhi Institute of Technology published journal in volume 2 Issue 6 June 2012 IJESS. His research areas include Data mining Information Retrieval Systems.

Mulugu Narendhar B.Tech CSIT from JNT University Hyderabad M.Tech Software Engineering from JNT University Hyderabad. He is currently Assoc Prof Department of IT at Bandri Srinivas Institute of Technology Chevella seven years of Academic experience. His research areas include Software engineering Data mining Web Technologies Computer Networks Testing & Project Management professional member in Computer Society of India.