iRecruit Tool : A Marketing Application for Higher Educational Institution using C4.5 Classification Technique

Sheila A. Abaya, Bobby D. Gerardo, and Bartolome T. Tanguilig III

***Abstract*— This research focuses on the application of C4.5 classification technique to help the marketing department of Higher Educational Institutions identify probable secondary schools for recruitment. A mapping feature was also provided to show possible routes from the point of origin to the target institution/s. This is to maximize the visit for every probable school. In view of this tool, the cost incurred by these educational institutions can be eliminated, if not, alleviated. Likewise, the application helps the administration recognize some qualified secondary schools for School-to-School Promotion (STSP) and utilize other resources.**

***Index Terms*— Data mining, C4.5, classification, recruitment**

I. Introduction

CURRENTLY, the education domain has vast amount of misused data. This research primarily aims to develop an application that enables the marketing department of Higher Educational Institutions identify probable secondary schools for recruitment. This is possible through the business technology of Data Mining (DM). DM bridges the gap between the generation of data and the understanding of data [6]. Likewise, this paper presents the iRecruit tool that classifies the target school for enrollment with the routing feature. This is to lessen the travel time from one target school to another.

II. FILE USED

This section enumerates the files used relevant to the proposed application.

A. Historical Data

The data contains 1,497 secondary students who took the entrance examination for a particular Higher Education Institution (HEI). It has been validated to ensure correctness and reliability of the output. Fig. 1 shows the table of Historical Data used in this research. The field column specifies the attributes enumerated from the historical data of HEI. Each attribute is defined with its data type and length.

Field	Type
StudentName	varchar(50)
SchoolName	char(100)
SchoolYearEnrolled	int(11)
RadialDistance_	decimal(10,3)
Gwa_	decimal(4,2)
IncomeBracket_	int(11)
SchoolOwnership	char(25)
Enrolled	char(25)
RadialDistance	char(25)
Gwa	char(25)
IncomeBracket	char(25)

Fig. 1. The HistoricalData Relation in MySQL. The relation is used to store the information from HistoricalData.

The following are the definition of terms used in the field column of the above table: (1) *Student Name*, the identity of the student; (2) *School Name*, the name of the secondary institution where the student graduated at; (3) *School Year Enrolled*, the year when the student enrolled in the HEI; (4) *Radial Distance*, the remoteness of the school; (5) *GWA*, the general weighted average of the student; (6) *Income Bracket*, the salary grade of the student's parents; (7) *School Ownership*, the type of ownership of the school whether public, or private; and (8) *Enrolled field*, student enrollment confirmation. In view of the aforementioned fields, some were found irrelevant in searching for probable secondary schools. As a result, only four attributes were chosen and treated as relevant criteria: Radial Distance, GWA, Income Bracket, School Ownership, and the Enrolled field identify the confirmation of enrollment.

Manuscript submitted January 27, 2014; revised February 06, 2014. This work was supported and financed by the University of the East, Manila, Philippines.

S. A. Abaya is a doctoral student under the program of Doctor in Information Technology of the Technological Institute of the Philippines and currently a faculty of the Department of Computer Studies and Systems of the University of the East, Caloocan Philippines, phone: +63-9063025931; (e-mail: sheila_abaya@yahoo.com.ph).

B. D. Gerardo was with West Visayas State University in Iloilo City, Philippines. He is now the Vice President for Admin and Finance of the same institution (e-mail: bgerardo@wsu.edu.ph).

B. T. Tanguilig is currently the Dean of the Graduate School of the Technological Institute of the Philippines, Quezon City Philippines (e-mail: bttanguilig_3@yahoo.com).

ISBN: 978-988-19252-5-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

B. *iRecruit.names*

A file that defines the class and attributes based on the selected criteria implemented during the Query Tool feature of the application. The values for each class and attributes are also defined in this file. The class states the expected output for a certain query.

```
Enrolled, Did Not Enroll.

Gwa: 75-79.99, 80-84.99, 85-89.99, 90-94.99, 99-100.
IncomeBracket: 10-20K, 21-50K, 51K+.
RadialDistance: -10Km, 10-20Km, 20+Km.
SchoolOwnership: PRIVATE, PUBLIC.
```

Fig. 2. Value Assigned per Attribute. The class is the expected output and the four relevant attributes with its corresponding range values.

Fig. 2 defines the contents of iRecruit.names, which is dependent on the selected attributes in iRecruit Setup feature. These are as follows: (1) "*Enrolled, Did Not Enroll*," the expected outcome of the query; (2) *GWA*, the value is categorized from 75 – 100; (3) *Income Bracket*, ranges from PHP10, 000 to PHP51, 000 and above; (4) *Radial Distance*, less than 10 to 20 kilometers from the HEI; and *School Ownership*, the type of ownership of the school whether public, or private. However, the contents of iRecruit.names may vary depending on the selected criteria.

C. *iRecruit.data*

Another file is needed for C4.5 algorithm together with file iRecruit.names. This file lists all the instances extracted from the HistoricalData to meet the selected criteria in the Query Tool feature of the application. It identifies the attributes and its value as well as the expected class. Fig. 3 shows an excerpt of iRecruit.data based on the chosen attribute value.

```
90-94.99, 21-50K, -10Km, PUBLIC, Did Not Enroll
80-84.99, 21-50K, -10Km, PUBLIC, Enrolled
90-94.99, 21-50K, -10Km, PUBLIC, Did Not Enroll
90-94.99, 21-50K, -10Km, PUBLIC, Did Not Enroll
85-89.99, 21-50K, -10Km, PUBLIC, Did Not Enroll
75-79.99, 21-50K, -10Km, PUBLIC, Did Not Enroll
80-84.99, 21-50K, -10Km, PUBLIC, Did Not Enroll
90-94.99, 21-50K, -10Km, PUBLIC, Enrolled
80-84.99, 21-50K, -10Km, PUBLIC, Enrolled
80-84.99, 51K+, -10Km, PUBLIC, Enrolled
85-89.99, 21-50K, -10Km, PUBLIC, Did Not Enroll
85-89.99, 21-50K, -10Km, PUBLIC, Enrolled
80-84.99, 10-20K, -10Km, PUBLIC, Did Not Enroll
80-84.99, 21-50K, -10Km, PUBLIC, Did Not Enroll
75-79.99, 21-50K, -10Km, PUBLIC, Enrolled
85-89.99, 10-20K, -10Km, PUBLIC, Enrolled
```

Fig. 3. An Excerpt of data from HistoricalData. Values extracted from the HistoricalData meeting the four applicable attributes.

Since the four attributes were selected in iRecruit Set up feature, the file was created and retrieved from the HistoricalData demarcated by comma.

III. TECHNOLOGY USED

This section defines the tools/techniques used in the development of the iRecruit application. It also discusses the implementation process of the application.

A. *MySQL*

Open source software that logically fits the application written in Apache and PHP. The Database Management System software is capable of manipulating the data due to its efficiency and reliability. It uses C and Java as programming languages. The easy-to-use service of MySQL and its portability in different platforms [1] substantiates the use of back-end DBMS. It is compatible with C4.5 algorithm which was written in C language.

B. *PERL*

Program Extraction and Reporting Language is open source software, which is considered a practical and useful language and achieves the same result whether a person is a structured or unstructured developer. The language is user-friendly. It can solve variety of problems with few lines of code and can run in UNIX platform due to its portability. [2].The technology was used to extract the attributes from the decision tree (DT) generated by C4.5. The extracted attributes were used to construct the Structured Query Language (SQL) statements and create the expected query. This was used to produce dynamic Hypertext Markup Language (HTML) as User Interface of the application. This is compatible with the application since C4.5 runs in Ubuntu which is a GNU/Linux distribution.

C. *JAVA Script*

A Web programming language designed for Web browsers. This language works effectively within the elements of web pages due to its constructs "lives" [3]. The language was used in the page to access the Google Maps API (Application Program Interface) and to validate the input data in HTML forms.

D. *Apache Web Server*

Free software distributed by Apache software foundation used to aid web technologies. It is the most popular web server designed to work in UNIX environment. This research has adopted the aforesaid web server as the host of Common Gateway Interface (CGI) of iRecruit.

E. *Google Maps API*

A web based mapping service application provided by Google with the power of routing traveler. This application was used to identify possible routes heading to the target schools. Markers have been provided as names of the queried schools from the selected criteria.

ISBN: 978-988-19252-5-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

F. HTML

It is used to develop web pages. Hypertext Markup Language (HTML) is a language that consists of a set of characters or symbols. It defines the logical structure of HTML documents and specifies how these should be displayed or printed [4]. This was used to describe the appearance of the page. The user interface (UI) of iRecruit was derived from this technology.

G. Ubuntu

It is an open source operating system (OS) [5] which is a project of Mark Shuttleworth in 2004 distributed by GNU/Linux. The application runs in the same platform due to its speed, reliability, and flexibility as well as to conform to the requirement of C4.5 in the UNIX environment.

IV. THE APPLICATION IRECRUIT

This section presents the flow of iRecruit application in Web based form.

A. Step 1: Setting the Criteria

Once the application is launched, it allows the user to choose a desired attribute to meet the criteria in identifying a target school.

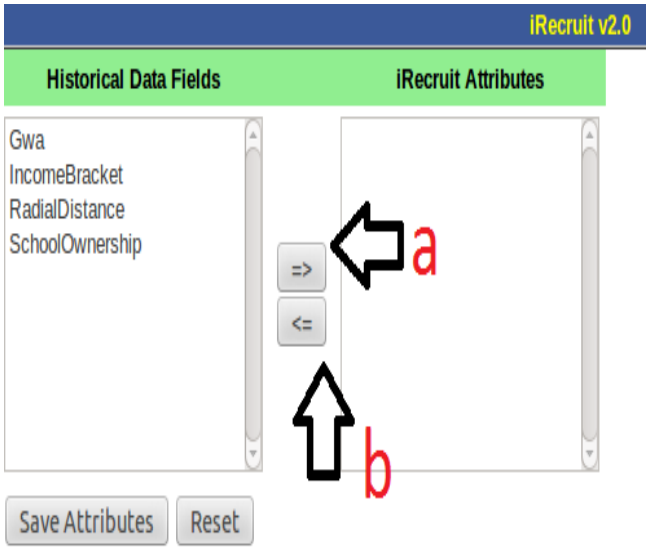

Fig. 4. Attribute Setup Interface. Initial screen of iRecruit for choosing the desired query criteria.

Fig. 4 shows the significant data fields listed on the Historical Data file. This feature allows the user to customize the query as well as to add or remove an attribute based on the criteria set for selecting secondary schools. Choosing (a) is considered as an attribute data field while choosing (b) eliminates a data field through iRecruit attribute list.

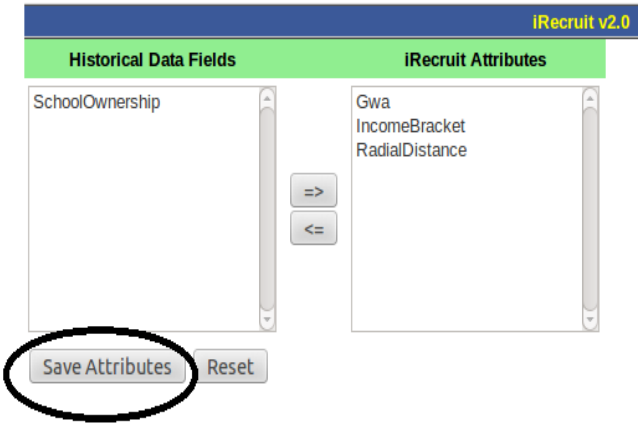

Fig. 5. iRecruit Attribute. Selected attributes are defined in the attributes list.

As shown on Fig. 5, selected attributes are moved on the list of iRecruit attributes. These criteria are considered parameters in generating the iRecruit.names while the "Save Attributes" button allows the SQL to store the selected attributes in database. On the other hand, the "Reset" button clears the iRecruit attributes column.

B. Step 2: Query Tool

The selected iRecruit attributes are displayed on the interface of Query Tool feature as seen on Fig. 6.

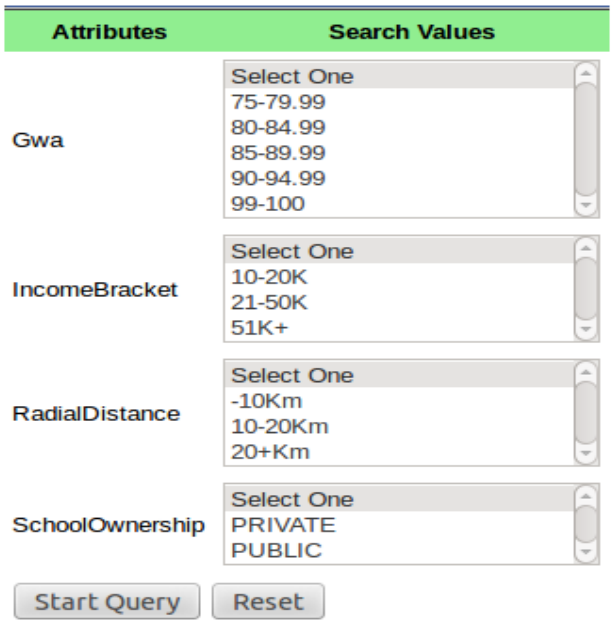

Fig. 6. Query Tool Interface. Selected attributes with possible values.

Since the user determined the four attributes, these elements have been placed together with corresponding values. This feature allows the user to select a value for a particular attribute.

ISBN: 978-988-19252-5-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

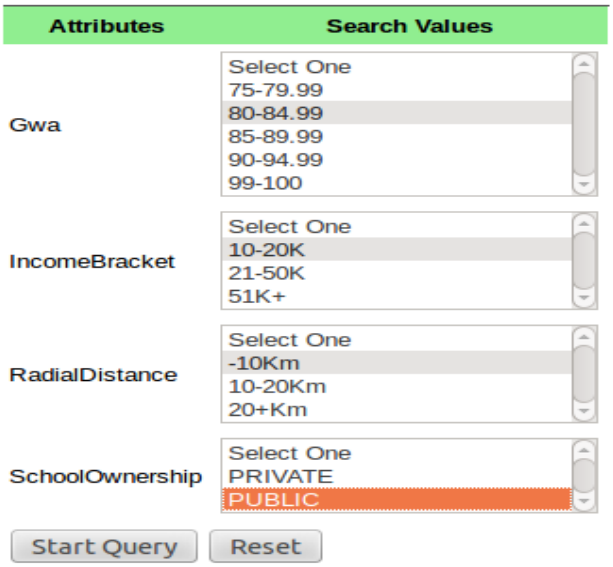

Fig. 7. Attribute with selected values. Preferred values per attribute are marked and to be used as criteria in executing the query.

The interface on Fig. 7 as highlighted shows the chosen attributes with corresponding values. Clicking the "Start Query" button triggers the application to generate iRecruit.data which contains instances derived from the HistoricalData.

C. *Step 3: C4.5 Algorithm*

iRecruit produces two files (iRecruit.names and iRecruit.data) needed by C4.5 algorithm upon access to MySQL. The application triggers the classifier to implement changes on the Decision Tree (DT) based on user's selected criteria and values. The criteria are stored on database tables. The classification technique identifies significant attributes from the chosen criteria in line with the computation of information gain.

```
RadialDistance = -10Km:
|   Gwa = 75-79.99: Enrolled (97.0/45.0)
|   Gwa = 99-100: Enrolled (85.0/38.0)
|   Gwa = 80-84.99:
|   |   SchoolOwnership = PRIVATE: Did Not Enroll (23.0/10.0)
|   |   SchoolOwnership = PUBLIC: Enrolled (395.0/195.0)
|   Gwa = 85-89.99:
|   |   SchoolOwnership = PRIVATE: Enrolled (21.0/7.0)
|   |   SchoolOwnership = PUBLIC: Did Not Enroll (278.0/131.0)
|   Gwa = 90-94.99:
|   |   SchoolOwnership = PRIVATE:
|   |   |   IncomeBracket = 10-20K: Enrolled (5.0/2.0)
|   |   |   IncomeBracket = 21-50K: Did Not Enroll (44.0/15.0)
|   |   |   IncomeBracket = 51K+: Did Not Enroll (0.0)
|   |   SchoolOwnership = PUBLIC:
|   |   |   IncomeBracket = 10-20K: Did Not Enroll (45.0/19.0)
|   |   |   IncomeBracket = 21-50K: Enrolled (342.0/166.0)
|   |   |   IncomeBracket = 51K+: Enrolled (8.0/3.0)
RadialDistance = 10-20Km:
|   Gwa = 75-79.99: Enrolled (4.0/2.0)
|   Gwa = 80-84.99: Did Not Enroll (16.0/4.0)
|   Gwa = 85-89.99: Did Not Enroll (16.0/5.0)
|   Gwa = 90-94.99: Enrolled (22.0/9.0)
|   Gwa = 99-100: Enrolled (3.0/1.0)
```

Fig. 8. Decision Tree (DT) of C4.5. The classification method generates a tree based on the selected criteria.

Based on the attributes selected as shown on Fig. 8, the Radial Distance becomes the root. It is followed by GWA, School ownership and Income Bracket. Considering the selected value, the DT is interpreted as:

If RadialDistance is < 10km
And GWA is from 80 – 84.99
And SchoolOwnership is public then
Out of 395 matched instances 200 students enrolled
And 195 students did not enroll in the HEI

The tree provides significant fields for SQL statements to execute the desired query.

D. *Step 4: Query Result*

This process prints the query output in tabular form as seen on Fig. 9 (a).

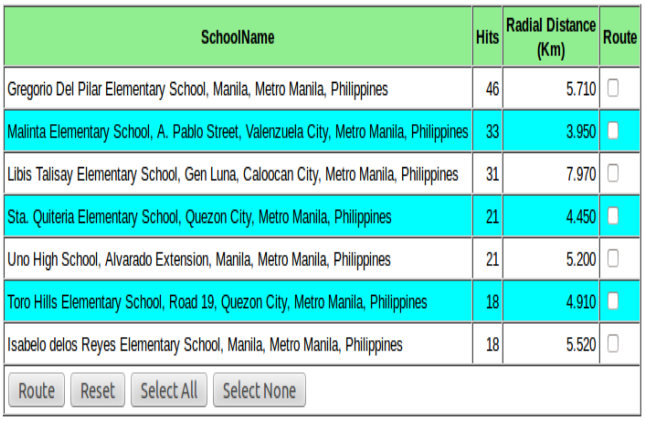

Fig. 9. (a) Query Result. The outcome of the query will be displayed on this interface.

The list of probable schools is arranged accordingly with the number of hits. It refers to the number of students who enrolled in the HEI. This helps the marketing team to decide whether the school still qualifies for the school-to-school promotion (STSP). The radial distance is also presented to know the remoteness of the institution. Clicking on the "Select All" button enables the route check box as seen on Fig. 9 (b).

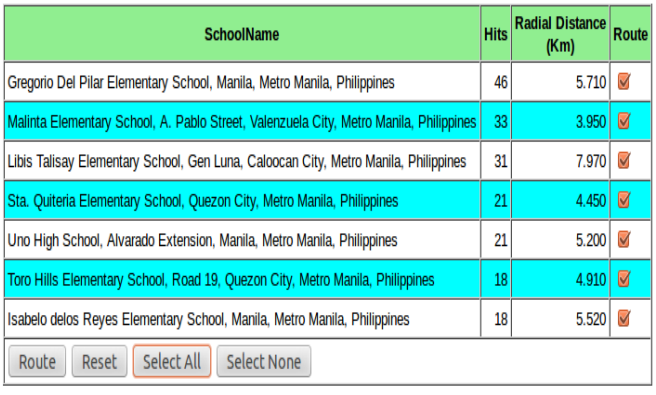

Fig. 9. (b) Query Result. The route check box enables the application to execute the waypoints of selected school.

ISBN: 978-988-19252-5-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

This feature allows the user to trace possible routes from the point of origin to the target school/s.

E. Step 5: Mapping/Routing

Once the "Route" option is clicked as seen on the interface of Fig. 9 (b), the application calls Google Maps API to generate the markers as presented on Fig. 10 (a).

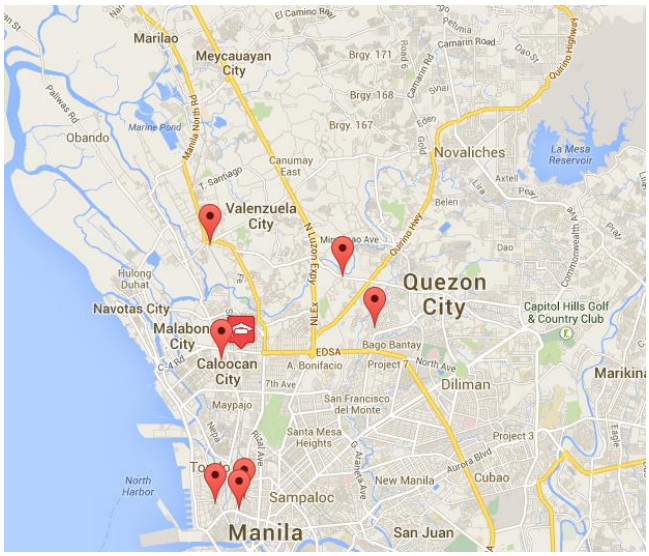

Fig. 10. (a) Google MapsAPI. Selected schools for mapping are named with marker.

The markers in the map represent the selected schools to be visited. iRecruit identifies the schools' latitude and longitude values.

Fig. 10 (b) presents the waypoints of several selected institutions. It shows the order of schools to be visited depending on the distance from the point of origin.

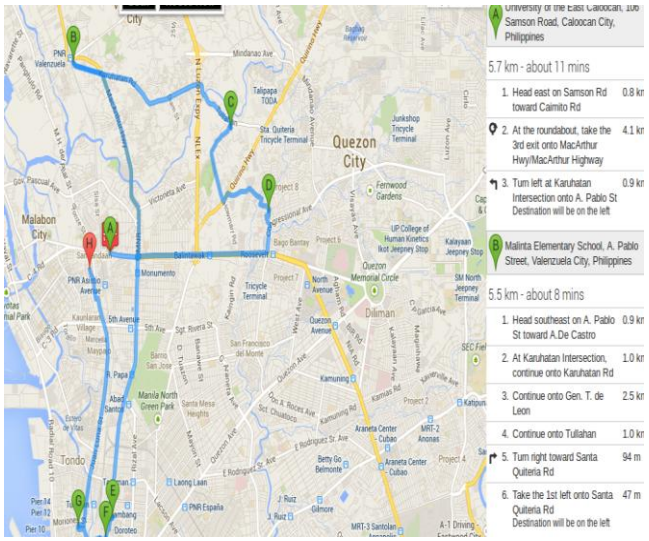

Fig. 10. (b) Google MapsAPI. Markers are drawn by lines as waypoints of selected school.

Google Maps API establishes the waypoints by optimizing the listed destination/s using the Shortest Path algorithm from the point of origin.

V. CONCLUSION

iRecruit is an effective application tool that enables the marketing department of several HEIs identify potential secondary schools for recruitment. It also alleviates the expenses of the department. Moreover, iRecruit can modify the chosen attributes based on the criteria set by the user. iRecruit.names and iRecruit.data contents can also create a dynamic tree generated by C4.5. The mapping aspect of the application identifies the appropriate route to enable the HEIs arrange the sequence of secondary schools to be visited in a particular time.

VI. RECOMMENDATION

Modifications on the algorithm of C4.5 technique is recommended as needed. Enhancement of the algorithm determines other possible ways to implement a Decision Tree (DT) efficiently. Lastly, regardless of the existence of C5 in the market, which is an improvement of C4.5, implementing the algorithm in other programming languages is highly recommended to identify the efficiency of the method in different platform.

ACKNOWLEDGMENT

S. A. Abaya would like to thank RBA and PLORS for the continued support in this endeavor. This project will not be completed without your continuous motivation.

REFERENCES

[1] D. Gosselin, D. Kokoska, and R. Easterbrooks, "PHP Programming with MySQL," Cengage Learning Aris Pte. Ltd., 2011, pp. 340 – 357.
[2] M. Brown, "DeBugging PERL. Troubleshooting for Programmers," McGraw-Hill Companies, 2001, pp. 3 – 17.
[3] D. Gosselin, "Javascript" 5th ed.,. Course Technology Cengage Learning, 2011, pp. 1, 22 – 35.
[4] D. Gosselin, "Principles of HTML, XHTML, and DHTML," Course Technology Cengage Learning, 2011, pp. 5 – 16.
[5] B.M. Hill, J. Bacon, et al, "The Official Ubuntu Book," Canonical Ltd., 2007, pp. 12 – 20.
[6] I. Witten, E. Frank and M. Hall, "Data Mining : Practical Machine Learning Tools and Techniques," 3rd ed., Elsevier Inc., 2011, pp. 4, 33.