Assessing Students Programming Proficiency Using Machine Learning Algorithms – An Empirical Approach

Adepegba O.A., Ameen A.O., & Akinola S.O
Department of Computer Science
[1]Adeleke University, Ede, Nigeria
[2]University of Ilorin, Ilorin, Nigeria
[3]University of Ibadan, Ibadan, Nigeria
E-mail: adepegbafunmilola01@gmail.com; ahmedameenyk4@yahoo.com; solom202@yahoo.co.uk

ABSTRACT

This study adopted an empirical technique to perform a predictive analysis of the relationship between Senior Secondary Certificate Examination (O'Level) results in (physics and mathematics) of Computer Science Students of Adeleke University Ede, and their corresponding Programming Practical experiments scores using a hybridized machine learning technique. The two data mining classification techniques used were Decision Tree (ID3, C4.5/ J48) and Artificial Neural Network (multilayer perceptron). The results obtain from this research showed that most students' with good grades in their mathematics performed very well in their programming tasks followed by Physics and it's been observed that their O-level subjects is able to predict their programming proficiency. Furthermore, The Decision Tree (ID3, C4.5/ J48), and a Neural network (multilayer perceptron) technique/algorithms created models whose accuracy results were 60.5042%, 96.6387%,99.1597% respectively.The performance measures used in evaluating the model in multilayer perception gave the best predictive model. The research showed the best predictive model as a frame work which could be used for the prediction of students' performance.

Keywords: Programming proficiency, Data mining, Data mining techniques, Empirical approach, O-Level

iSTEAMS Multidisciplinary Conference Proceedings Reference Format
Adepegba O.A., Ameen A.O., & Akinola S.O (2019): Assessing Students Programming Proficiency Using Machine Learning Algorithms – An Empirical Approach. Proceedings of the 20th iSTEAMS Multidisciplinary Trans-Atlantic Conference, KEAN University, New Jersey, United States of America. 10th – 12th October, 2019. Pp 219-230. www.isteams.net/usa2019 - DOI Affix - https://doi.org/ 10.22624/AIMS/iSTEAMS-2019/V20N1P17

1. INTRODUCTION

The importance of computer programming cannot be underestimated in the field of computer science, but it is been observed that most computer science graduates knows very little about programming, most students of computer science do not really have interest in programming so students performs poorly in programming courses. Factors affecting these students' performance need to be identified and addressed properly. This would in a way serve as a tool for stakeholders such as students, parents, management and policy makers, to assist them in their decision making.

1.1 Background

Data mining is one of the processes of discovering meaningful new correlations, patterns and trends by sifting through large amounts of data stored in repositories, using pattern recognition technologies as well as statistical and mathematical techniques" (Bharadwaj (2011).

Some studies have showed that data mining techniques played major role in evaluating students' performances. For example, Al- Radiadeh et al (2015), Applied the data mining techniques, particularly classification to help in improving the quality of higher educational system by evaluating students' data to study the main attributes that may affect the students performances in their courses. Ayesha et al (2012) used K-means clustering algorithm as data mining technique to predict students learning activities, by using class quizzes, mid and final examination and assignments. The correlated information was conveyed to the class teachers to reduce the failing ratio by taking appropriate steps at a time and improving students' performances. Also, In a quest to improve performance of students offering programming in the university, several researches have been carried out to gain insights into how students proficiency in programming can be improved and as to the factors affecting their performance. Mrwan et al (2018) conducted a survey that explores if students' weakness in the English language affects the ability of the students to understand programming with respect to factors such as computer Lab facilities, Lecturer, mathematics, and logical thinking. This study took place in two universities in Libyan results of the survey showed that 37%, 38% and 25% of students stated that their programming abilities were negatively affected by English, Computer Lab and Lecturer respectively.

2. RELATED WORKS

Akinola et al (2014), explains that pair programming has been widely acclaimed the best way to go in computer programming. Also, collaboration involving more subjects has been shown to produce better results in programming environments. The work seeks to inculcate and acquaint the students involved in the study with the spirit of team work in software projects and to empirically determine the effective (optimum) team size that may be desirable in programming/learning real life environments. The challenge in this study is that the optimum group size needed for the collaboration has not been adequately addressed.

Hilal et al (2018) proposed a study that measures the level and difference of self-efficacy of computer in programming for non- technical students using scratch programming. The questionnaires were used in this work, the data obtained were analysed descriptively using a Statistical package (SPSS). The findings show that t-test analysis show that there is no statistically significant difference in self-efficacy based on gender and the background of the program study. Correlation analysis also found that there was no self-efficacy relationship between computers in scratch programming with academic achievement. There will be a very clear cut if the same data can be retrieved for a data mining technique(s) to be used in other to compare the result with the statistical package. Also, the use of the questionnaire approach seems to lack validity; there is no way to tell how truthful a respondent is being.

Akinola (2016), Computer programming skill and gender difference: An empirical study. In this study, an experimental proof approach was employed to verify the wide acclaimed gender difference in computing. Two parallel student-gender groups (male and Female) were subjected to two different computer programming problems. The problems were given to them to solve, starting from the analysis phase to the final implementation. Their outcomes measured in terms of the accuracy and efficiency of codes turned out from the experiments was then compared statistically. Here, possible challenge here could be the issue of gender equality among the student.

3. PROPOSED SYSTEM

3.1 Research Methodology (Knowledge Discovery Processes Steps)

1. **Problem Definition:** the stage entails defining the problem at hand.
2. **Data collection and gathering**: the data for the study was collected from the result of the students' experiment practical and the departmental student database at Adeleke University, Ede Nigeria.
3. **Data Selection**: the selection of the target and input variables and the data samples pertinent to the research**.**
4. **Data Preprocessing** (Data Integration/cleansing)**:** Handling noise, and missing data. The synchronization of data from their heterogeneous sources; merging of tables.
5. **Data Transformation:** the feature selection to reduce the effective number of variables, Resampling, discretization etc.
6. **The data mining task:** the data mining task used here is the classification task; the objective is the development of a predictive model.
7. **Weka Tool:** the data mining open source software tool used is WEKA.
8. **Methods used:** the two classification methods used for this research are the Decision Tree, and ANN. The algorithms used under these classifications ar**e;** ID3, C4.5 and Multilayer perceptron respectively.
9. **Result evaluation:** Interpretation of extracted patterns, removal of redundancy and visualization.
10. **Knowledge representation:** this is the representation of knowledge it pertains documentation and reporting. The checking and resolve potential conflicts with previously believed knowledge.

3.2 Experimental Design And Procedures

3.2.1 Data Acquisition

3.2.1.1 Method of Data Acquisition

Empirical cycle methods was used, it consists of following stages:

1. Formulation of an hypothesis or Questions to test
2. Observation of a situation
3. Abstracting observations into data
4. Analyzing and testing the hypothesis with new empirical data
5. Evaluating and drawing conclusions with respect to the tested hypothesis

The above empirical cycle can be achieved by carrying out a hypothesis using students' experiments;

3.3 Experiments

Students practical classes were been utilized for this study. Students' experiments was adopted where about 200 students were been tested and are divided into groups according to their levels. 35 students use the Laboratory at a time and it's a student to a personal computer (PC) and students' were keenly been observed. Students were given computer programming questions in at least two (2) different languages (C++ and Java) at different occasions and it is expected of the students' to write and run their codes within 40 minutes otherwise, an extra time of 5-10 minutes was given to debug and rerun codes. Students' were been accessed according to the timing (from problem definition to implementation, accuracy, correctness, first time error, and the time at which errors are debugged and recompiled. Using this technique, each students are scored and scores are based against their corresponding physics and mathematics scores, these was therefore used to determine the hidden relationship between the known O-level physics and mathematics scores (converted to key variables) and the corresponding results in their programming test tasks (in key variables) in order to really test for the correlation.

The data used in this study is the CSC (Computer Science) and CIS (Computer Information Science) students' data of the computer science and mathematics department of Institution. The data used comprises of two aspects the first is the mathematics and physics O' level grades of the students' collected from the Department and the second is the results of five (5) experiments conducted during their practical classes from each levels as stated above. The data include two basic categories of variables, the data consists records of the students' data variables as shown in table 1.

Table 1: Representation of Students' data

S/N	VARIABLE NAME	VARIABLE FORMAT	VARIABLE TYPE
1	Mat no	1,2,3…	Numerical
2	Maths	A1, B2, C4…	Categorical
3.	Physics	A1, B2, C4…	Categorical
4.	Practical 1	10, 20, 30…	Numerical
5.	Practical 2	10, 20, 30…	Numerical
6.	Practical 3	10, 20, 30…	Numerical
7.	Practical 4	10, 20, 30…	Numerical
8.	Practical 5	10, 20, 30…	Numerical
9.	Practical 6	10, 20, 30…	Numerical
10.	Practical 7	10, 20, 30…	Numerical
11.	Practical 8	10, 20, 30…	Numerical
12.	Practical 9	10, 20, 30…	Numerical
13	Practical10	10, 20, 30…	Numerical

The proposed system framework aggregates four components: The first component is the data acquisition and storage, responsible for storing students' data, gathered from different data sources proposed in a data warehouse. The second component is the model building, responsible for obtaining knowledge about the student, through appropriates classification models. Different classification algorithms are proposed in search for the best model with high predictive accuracy. The third component is for mapping the pattern in the rules generated with the student data to predict performance and the fourth component is the recommendation, responsible for recommending necessary action to be carried out on individual students' based on the prediction from the intelligent evaluation system as shown in figure 1

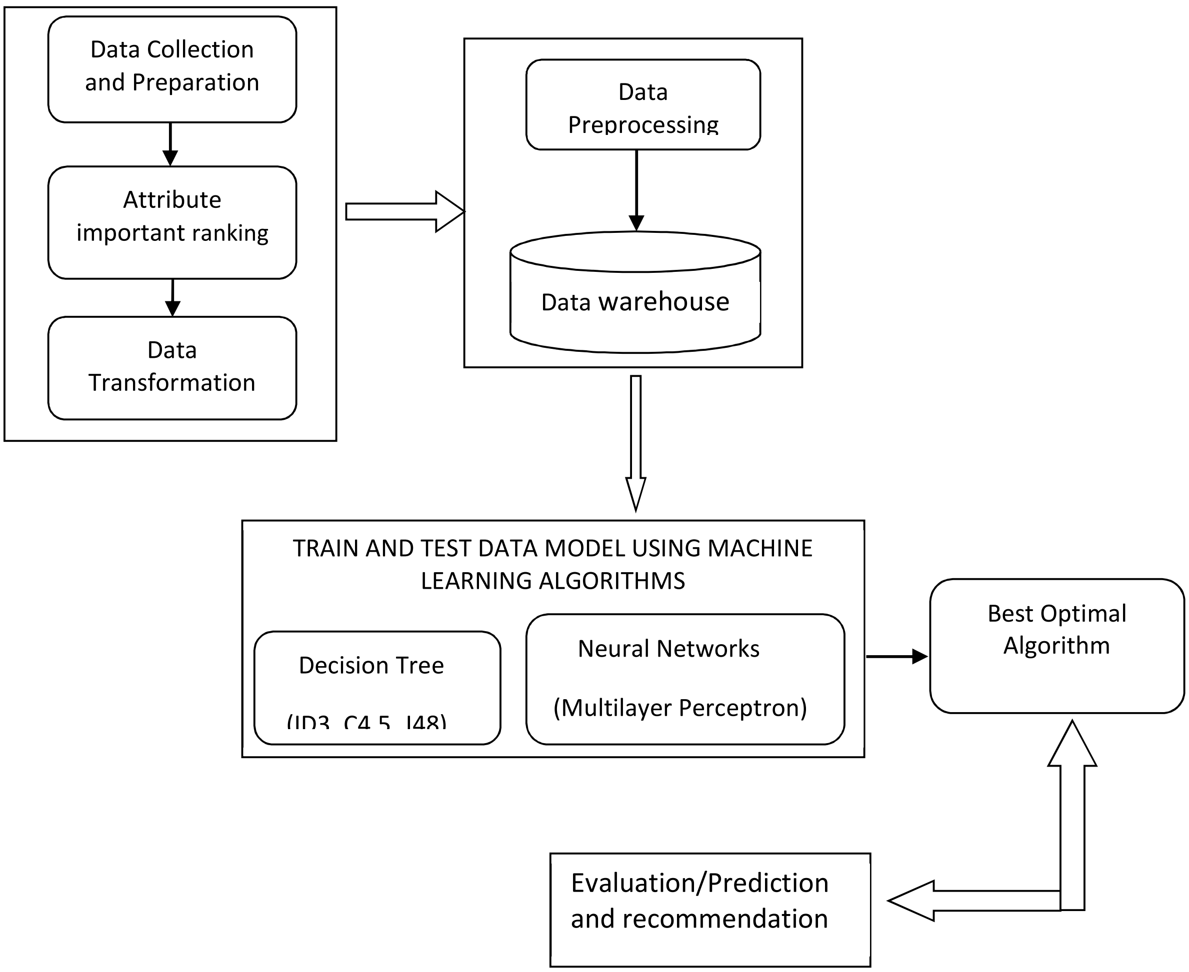

Figure 1: Architecture of the proposed data mining model

3.4. Performance Evaluation Measures

3.4.1 Evaluation Metrics

In selecting the appropriate algorithms and parameters that best model the students' performance, the following metrics were used.

i. Time: This referred to the time taken to complete a training or modeling of a dataset. It is represented in seconds.
ii. Percentage of Correct/Incorrect Classification: This is measured by the difference between the actual and predicted variable.
iii. TN (True Negative): This is the Number of correct predictions that an instance is irrelevant.
iv. TP (True positive): This is Number of correct predictions that an instance is relevant.
v. FN (False Negative): This is Number of incorrect predictions that an instance is irrelevant.
vi. FP (False Positive): This Number of incorrect predictions that an instance is relevant.
vii. Accuracy: The proportion of the total number of predictions that were correct; $\frac{TN+TP}{TN+FN+FP+T}$

viii. Precision: Refers to the proportion of the predicted relevant pages that were correct; $\frac{TP}{FP+TP}$

ix. Recall: It is the proportion of the relevant pages that were correctly identified; $\frac{TP}{FN+TP}$

x. F-Measure: Derives from precision and recall values. A combined measure from precision and recall; $2 \times Precision \times \frac{Recall}{Precision+Recall}$

xi. Kappa Statistics: It is used to measure the concordance level between categorical data during prediction. Cohen's kappa measures the agreement between two raters that each classifies *N* items into *C* mutually exclusive categories. The equation for κ is: $\kappa = \frac{\Pr(a) - \Pr(e)}{1-\Pr(e)}$, where Pr(a) is the relative observed agreement among raters, and Pr(e) is the hypothetical probability of chance agreement, using the observed data to calculate the probabilities of each observer randomly saying each category. If the raters are in complete agreement the κ = 1. If there is no agreement among the raters (other than what would be expected by chance) the κ ≤ 0.

xii. Correlation Coefficient: This is the correlation coefficient measures the statistical correlation between the predicted and actual values. This method is unique in that it does not change with a scale in values for the rest cases as the previously mentioned measures do. Also note that here a higher number means a better model, with a 1 meaning perfect statistical correlation and a 0 meaning there is no correlation at all. This performance measure is only used for numerical input and output.

xiii. Support: Percentage of training data for which the L.H.S of the rule is true. The measure of how widely applicable.

3.4.2 Error Metrics

i. Absolute Error: Absolute error is the amount of physical error in a measurement.

ii. Relative Error: Relative error gives an indication of how good a measurement is relative to the size of the thing being measured.
Absolute Error = Δx
Relative Error = $\frac{Absolute}{Measured\ value}$

iii. Mean Squared Error: The mean-squared error is one of the most commonly used measures of success for numeric prediction. This value is computed by taking the average of the squared differences between each computed value and its corresponding correct value. The root mean-squared error is simply the square root of the mean-squared error. The root mean-squared error gives the error value the same dimensionality as the actual and predicted values.

iv. Mean Absolute Error: Mean absolute error is the average of the difference between predicted and actual value in all test cases; it is the average prediction error.

v. Root Relative Squared Error: Relative squared error is the total squared error made relative to what the error would have been if the prediction had been the average of the absolute value. As done with the root mean-squared error, the square root of the relative squared error is taken to give it the same dimensions as the predicted values themselves. Also, just like root mean-squared error, this exaggerates the cases in which the prediction error was significantly greater than the mean error.

vi. Relative Absolute Error: relative absolute error refers to the total absolute error made relative to what the error would have been if the prediction simply has been the average of the actual values. If $y1, y2, \dots, yn$ are n observed values and $\widehat{y1}$,$\widehat{y2}\hat{y}1$

4. RESULT PRESENTATION & DISCUSSION

The goal of this study is to adopt an empirical technique to investigate and predict the correlation between student's O'level courses and their programming skills using data mining techniques from an empical study angle. A classification technique was adopted to develop the predictive model. The model was built with three machine learning algorithms: C4.5 Decision Tree Classification Algorithm, ID3 Decision Tree Classification Algorithm and MLP Neural Network algorithm using Weka, a machine learning software tool. A comparative analysis of the performance of the models was carried out.

4.1. The Dataset

The Data set is a set of data items. The data set used for this study was stored in a Microsoft Excel spreadsheet named "students'Data.csv". For easy usage, the data was converted into ARFF format, that is, the file with (.arff extension) named as "studentResult.arff". The data was loaded into Weka and the attributes are recognized.

The ARFF File Format requires declarations of @relation, @attribute and @data:
1. @relation declaration associates a name with the dataset e.g. @relation studentResult
2. @attribute declaration specifies the name and type of an attribute e.g. @attribute <attribute-name><datatype>. Data type can be numeric, nominal, string or date

- @relation studentResultTest
- @attribute math {A1, B2, B3, C4, C5, C6}
- @attribute physics {A1, B2, B3, C4, C5, C6}
- @attribute programming_score numeric
- @attribute result {excellent, good, pass, fail}

3. @DATA declaration is a single line denoting the start of the data segment e.g. A1, B2, B3, C4, C5, C6, 20, excellent, good, pass, fail. Missing values are represented by using the question mark (?).

4.1.1 Description of the Dataset

Each record in the dataset corresponds to a single student data and results which were collected during data acquisition.

Table 2: Attributes and their descriptions

S/N	ATTRIBUTE	DATA TYPE
1.	maths	Nominal
2.	physics	Nominal
3.	Programming_score	Numeric
4.	result	Nominal

4.1.2 Training Data set

The training set was used to enable the system to observe relationships between input data and the resulting outcomes. This allows the system to learn and develop a relationship between the input and the expected output. 57.97% of the dataset was used for training making 119 instances. The data used to train the model was named "studentResultTrain.arff".

4.1.3 Testing Data set

In this phase the model was applied to new instances or dataset which was made upof 87 instances; (42.69%) of records of the students' datasets in order to predictstudents' performance result class. The data used to test the model was named "studentData_Test.arff".

4.2 Data Pre-Processing

The preprocess section enables data to be loaded into the software tool. Data can be imported from a file in various formats: ARFF, CSV, C4.5, binary, it can also be read from a URL or from an SQL database (using JDBC). In this study, the data was loaded into the software tool using the ARFF (Attribute-Relation File Format) as this is the easiest and the most common way of getting data into tool.

Viewer

Relation: studentResultTest

No.	math Nominal	physics Nominal	programming_score Numeric	result Nominal
1	A1	C6	44.1	good
2	B2	B3	44.0	good
3	C5	C6	31.3	good
4	B3	A1	45.0	excellent
5	C6	C4	28.5	pass
6	C6	C6	27.4	pass
7	C4	B3	40.7	good
8	B3	C5	42.8	good
9	C5	C6	30.4	good
10	C4	C6	28.0	pass
11	C5	C6	31.3	good
12	C4	C5	27.5	pass
13	C6	C5	27.0	pass
14	B3	C6	39.0	good
15	B3	B3	44.5	excellent
16	C4	C4	27.5	pass
17	C5	C6	28.5	pass
18	C5	C4	33.0	good
19	B3	B3	47.9	excellent
20	A1	B3	48.4	excellent
21	C5	C4	33.3	good
22	B2	B3	48.4	excellent
23	C5	B3	37.5	good
24	C6	C5	31.5	good
25	B3	C6	37.9	good
26	C6	C6	26.4	pass
27	C6	C4	31.0	good
28	B2	B3	43.5	good
29	C6	C6	27.1	pass
30	C5	C5	29.5	pass
31	C5	C6	43.5	good

Undo OK Cancel

Figure 2: Preprocessed data

4.3 Feature Selection Methods

Attribute importance analysis was first carried out to rank the attributes used in this work by significance using Wrapper method using a BESTFirst Search technique and Filter method using RANKER's Search Technique.

4.4 Model Building Using ID3 Decision Tree

The raw data, that is, the training data set was selected and pre-processed by the ARFF converter. The ID3 decision tree model for the system was then generated from the studentResultTrain.arff. The ARFF pre-processed data was then trained by the implementation tool. The data is classified using the ID3 algorithm under the classify panel of the tool.

Table 3: Results of modeling data on ID3 algorithm

METRIC	VALUE
Time take to build the model	0 seconds
Correctly classified instances	60.5042%
Incorrectly classified instances	2.521%
Kappa statistics	0.9095
Mean Absolute Error	0.02
Root Mean Squared Error	0.0.1414
Relative Absolute Error	12.2338%
Root Relative Squared Error	52.2442%

Table 4: Detailed performance measures for ID3 algorithm

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.933	0.033	0.875	0.933	0.903	0.7	excellent
	0.963	0.048	0.981	0.963	0.972	0.861	good
	1	0	1	1	1	0.688	pass
	0	0	0	0	0	0	Fail
Weighted Avg.	0.96	0.041	0.961	0.96	0.96	0.815	

4.4 Model Building Using J48 Decision Tree (C4.5) Algorithm

C4.5 algorithm is an algorithm that can handle numeric attributes, in contrast to the ID3 algorithm from which C4.5 has evolved therefore; there was no need to discretize any of the attributes. The training data set is classified using the C4.5 algorithm under the classify panel on the tool.

Table 5: Results of modeling data on C4.5 (J48) algorithm

METRIC	VALUE
Time take to build the model	0.03 seconds
Correctly classified instances	96.6387%
Incorrectly classified instances	3.3613%
Kappa statistics	0.9392
Mean Absolute Error	0.0246
Root Mean Squared Error	0.1292
Relative Absolute Error	8.6915
Root Relative Squared Error	34.5225%

Table 6: Detailed Performance measures for C4.5 (J48) algorithm

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.939	0.012	0.969	0.939	0.954	0.962	excellent
	0.986	0.061	0.958	0.986	0.972	0.96	good
	00.938	0	1	0.938	0.968	0969	pass
	0	0	0	0	0	0	fail
Weighted Avg.	0.966	0.039	0.967	0.966	0.966	0.962	

Figure 3 shows the decision tree constructed by the J48 classifier. This indicates how the classifier uses the attributes to make a decision. The leaf nodes indicate the outcome of a test, and each leaf node holds a class label and the topmost node is the root node. Nine Rules generated from the decision tree are illustrated in Figure

4.5 Model Building Using The Multilayer Perceptron Ann Algorithm

The classifier for ANN algorithm was selected, that is, from the GUI ➔Click explorer ➔Open file. Apply filter then chooses classify button to select Multilayer Perceptron, where some parameter must be set for better result to be achieved.

Table 7: Results of modeling data on Multilayer Perceptron ANN algorithm.

METRIC	VALUE
Time take to build the model	8.47 seconds
Correctly classified instances	99.1597%
Incorrectly classified instances	0.88403%
Kappa statistics	0.9849
Mean Absolute Error	0.0105
Root Mean Squared Error	0.0659
Relative Absolute Error	3.7307%
Root Relative Squared Error	17.6128%

4.6 Model Comparison

The Weighted averages of the models were compared using different performance measures. The best model was then selected using Tables 4.7a, Table 4.7b and Figure 4.8. The performances of these models were evaluated based on these criteria:

- Prediction accuracy
- Time taken to build the model and
- Error rate.

These are illustrated in figure 4.9. MPL algorithm predicts better than the ID3 and C4.5 algorithms since its accuracy is the highest compared to others.

Table 8: Performance Summary of the models.

Algorithms	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
ID3	0.96	0.041	0.961	0.96	0.96	0.815
C4.5	0.966	0.039	0.967	0.966	0.966	0.962
MLP	0.992	0.012	0.992	0.992	0992	0.995

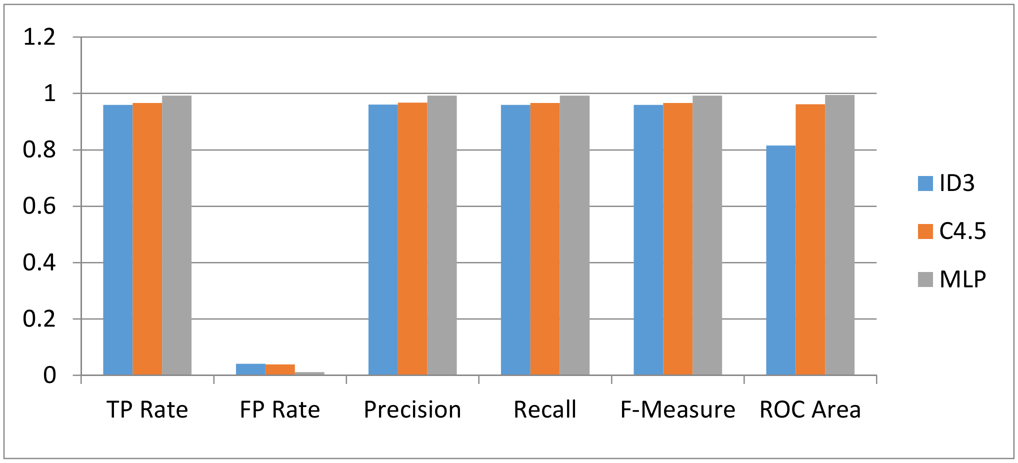

Figure 3: Comparison between performance measure parameter

Table 9: Comparative analysis on the models

METRIC	ID3	C4.5	MLP
Time take to build the model	0 second	0.03 seconds	8.47 seconds
Correctly classified instances	60.5042%	96.6387%	99.1597%
Incorrectly classified instances	2.521%	3.3613%	0.88403%
Kappa statistics	0.9095	0.9392	0.9849
Mean Absolute Error	0.02	0.0246	0.0105
Root Mean Squared Error	0.0.1414	0.1292	0.0659
Relative Absolute Error	12.2338%	8.6915	3.7307%
Root Relative Squared Error	52.2442%	34.5225%	17.6128%

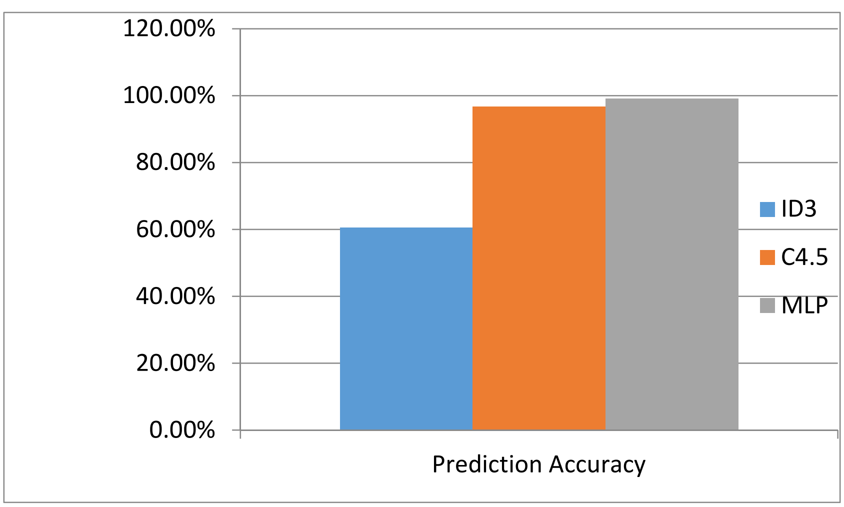

Figure 4: Prediction Accuracy

5. CONCLUSION

This study was able to combine and bring information science side by side with computer science by adopting an empirical approach together with data mining techniques to determining students' programming performance. Carrying out an experimental programming practical, it gives a new mind set as it is a real event towards data gathering to give a more valid and substantial results. The use of the questionnaire approach too lacks validity; there is no way to tell how truthful a respondent is being, and respondents may read differently into each questions and reply based on their own interpretations of the question which could not be generally accepted. In the feature selection method using the wrapper method and the BestFirst search technique, the result shows that students' performance in mathematics highly predicts their programming proficiency performance followed by physics. Also the multilayer algorithm has the best overall accuracy and this shows that it is the modeled with a superlative result compared to the ID3 and J48 models. Therefore using the empirical approach together with the data mining algorithms, it is concluded that a factual result is generated. On the other hand, the study also shows that student programming proficiency performances increase at subsequent levels of their programming test tasks, except for other factors like lack of consistent personal studying of programming, interest or attitudes towards programming may however be responsible for the low representation of other students in programming.

6. REFERENCE

1. Akinola, S.O. (2016) Computer programming skill and gender difference: An empirical study. *American Journal of Scientific and industrial research* ISSN:2153-649X,doi:10.5251/ajsir.2016.7.1.1.9
2. Akinola, S.O., Akinkunmi B.O., and Alo, T.S. (2012) A Data Mining Model for Predicting Computer Programming Proficiency of Computer Science Undergraduate Students. *Afr J. of Comp &ICTs Vol 5, No.1* pp 43-52
3. Akinola, S. O. and Nosiru, K. A (2014) Factors Influencing Students' Performance in Computer Programming: A Fuzzy Set Operations Approach. *International Journal of Advances in Engineering &Technology,* Sept., 2014.ISSN: 22311963.
4. Al-Radaideh Q. A., Al-Shawakfa E. W, and Al-Najjar M. I., (2016) "Mining student data using decision trees", *International Arab Conference on Information Technology*(ACIT'2016), Yarmouk University, Jordan, 2016.
5. Ayesha S., Tasleem M., AhsanRazaSattar, M. Inayat K.,(2010) "Data mining model for higher education system", *Europen Journal of Scientific Research*, Vol.43, No.1, pp.24-29, 2010.
6. Bharadwaj B.K and Pal. S (2016) "Data Mining: A prediction for performance improvement using classification", International Journal of Computer Science and Information Security (IJCSIS), Vol. 9, No. 4, pp. 136-140, 2011 .
7. Hilal, M. M, Anas A.S, and Zulhazlin A. (2018) Computer Self Efficacy in Programming Language for Non-Technical Students using Scratch Programming *International Journal of Pure and Applied Mathematics* ISSN:1311-8080Volume 118 No.18 2018,3381-3388
8. Mrwan, B.I and Hanny A. (2018) "The Correlation between Arabic Students' English Proficiency and their computer programming Ability at the University Level" *International Journal of Managing Public Sector Information and Communication Technologies (IJMPICT)* Volume 9. No 1,2018.
9. Pandey,U. K.. and Pal, S. 2011) "A Data mining view on class room teaching language", (IJCSI) *International Journal of Computer Science* Issue*,* Vol. 8, Issue 2, pp. 277-282, ISSN:1694-0814, 2011.