Prediction of Academic Performance Applying NNs: A Focus on Statistical Feature-Shedding and Lifestyle

Shithi Maitra[1], Sakib Eshrak[2], Md. Ahsanul Bari[3],
Abdullah Al-Sakin[4], Rubana Hossain Munia[5], Nasrin Akter[6], Zabir Haque[7]
Dept. of Computer Science and Engineering, Ahsanullah University of Science and Technology, Dhaka, Bangladesh[1,2,3,4,5,7]
Dept. of Electronics and Telecommunication Engineering, Daffodil International University, Dhaka, Bangladesh[6]

***Abstract*—Automation has made it possible to garner and preserve students' data and the modern advent in data science enthusiastically mines this data to predict performance, to the interest of both tutors and tutees. Academic excellence is a phenomenon resulting from a complex set of criteria originating in psychology, habits and according to this study, lifestyle and preferences–justifying machine learning to be ideal in classifying academic soundness. In this paper, computer science majors' data have been gleaned consensually by surveying at *Ahsanullah University*, situated in Bangladesh. Visually aided exploratory analysis revealed interesting propensities as features, whose significance was further substantiated by statistically inferential Chi-squared (χ^2) independence tests and independent samples *t*-tests for categorical and continuous variables respectively, on median/mode-imputed data. The initially relaxed *p-value* retained all exploratorily analyzed features, but gradual rigidification exposed the most powerful features by fitting neural networks of decreasing complexity i.e., having 24, 20 and finally 12 hidden neurons. Statistical inference uniquely helped shed off weak features prior to training, thus optimizing time and generally large computational power to train expensive predictive models. The *k*-fold cross-validated, hyper-parametrically tuned, robust models performed with average accuracies wavering between 90% to 96% and an average 89.21% F1-score on the optimal model, with the incremental improvement in models proven by statistical *ANOVA*.**

***Keywords*—Educational Data Mining (EDM); Exploratory Data Analysis (EDA); median and mode imputation; inferential statistics; t-test; Chi-squared independence test; ANOVA-test**

I. Introduction

The research field of Educational Data Mining (EDM) applies statistics and machine learning to information stemming from educational environments and is thus contributing to educational psychology. EDM leverages precise, fine-grained data to discover types of learners, examine effectiveness/suggest improvements of instructional learning environments, predict students' learning behavior and advance learning sciences. Baker, Yacef [1] critically identified learners, educators, researchers and administrators to be the four stakeholders of EDM.

The bulk of the academic literature, while addressing problems from the domain of EDM, has taken past academic credentials into account. Fewer academicians resorted to mental health and personality traits. However, the application of features related to students' lifestyle and preferences, as done in this study to predict academic excellence, is a novel approach to the field. In this study, we choose ten such features and apply an evidential function—mapping them to students' expertise in the respective field. The study shows that attributes apart from academic track-records alone can predict academic success which can help institutions to foresee the aptitude of the graduates they are producing, admitting, strategizing for hiring or educating.

Systematic collection of educational data and ML methodologies enable researchers to explore the similarities and dissimilarities among academically sound and unsound students. Recent such researches in the EDM arena have gained momentum using Neural Networks (NNs). NNs are surpassing traditional learning models such as Logistic Regression, Support Vector Machines in performance—characteristically having multiple hidden layers with different activation functions. NNs are versed in fitting complex functions spread through many dimensions featuring multiple independent variables. Back-propagation allows refinement of its initial parameters through numerous epochs, with derivatives showing the direction and learning rate indicating the magnitude of refinement. The weights represent a hierarchical mapping from lower (learns comparatively simpler features) layers to the higher (learns sophisticated features) layers.

The research work addresses a binary classification problem in categorizing final-year Computer Science (CS) students from *Ahsanullah University, Bangladesh* as of their academic performance, following the four EDM phases [2]:

- It is generally held that if a CS student is able to maintain a $\mathbf{CGPA} \geq \mathbf{3.40}$ until the final semester, he/she is faring academically well. First, we exploratorily choose unconventional, unique features by finding their consistent relations with CGPA.
- Then the best use of available data is made by imputing both categorical and continuous variables.
- Third, NN models are proposed to predict academic status.
- The models and features are statistically cross-validated and finer conclusions are drawn.

The sequencing of this paper renders the second section as a review of existing literature, the third section as descriptions of methods followed, the fourth section as a depiction of experimental results and the final section as concluding notes.

II. Related Works

Artificial intelligence-based and statistically analytical methods (Fig. 1) applied in classifying academic performance can be discussed in light of three prototypical dimensions as below.

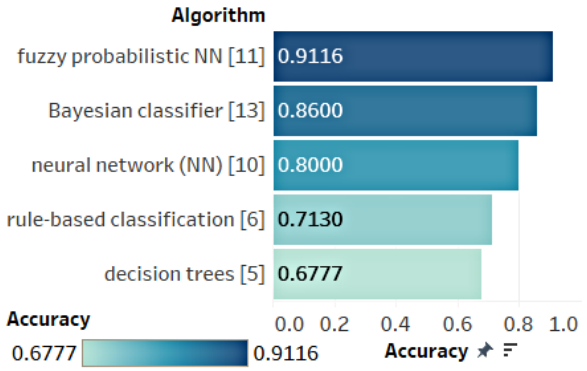

Fig. 1. Comparison among related researches.

A. Conventional Statistics and Decision Trees

Wilkinson, Zhang et al. [3] conducted a study on 706 undergraduate medical students in three consecutive years at the University of Queensland with their objective of modestly determining how precisely each of prior academics, admission tests and interviews accounted for students' performance at post-graduation. These altogether served as the selection criteria which accounted for 21.9% variation in overall scores. They explored GPA to correlate most strongly with performance (p-*value* < 0.001), followed by interviews (p-*value* $= 0.004$) and admission tests (p-*value* $= 0.08$), respectively.

Chamorro-Premuzic et al. [4] established through two longitudinal studies (sample size, $n = 70$, 75 respectively) that personality-measures could testify for students' academic ability. The setting examined students over three academic years at two British universities along academic behavior and personality traits. Sample-1 proved that neuroticism negatively and conscientiousness positively impacted students' academics, accounting for 10% variance. Sample-2 used EPQ-R showing three personality factors were instrumental in predicting academic performance and accounted for 17% variance.

Yadav et al. [5] explored C4.5, ID3 and CART decision trees on engineering students' data to predict final exam's scores. They obtained a true positive rate (TPR) of 0.786 on the 'fail' class using ID3 and using C4.5 decision trees, the highest accuracy of 67.77%. Ahmad et al. [6] proved the impact of demographic information of students spanning eight educational years in predicting academic success. They found rule-based classification techniques to fit the data best with 71.3% accuracy.

B. Unsupervised Clustering Approaches

Oyelade et al. [7] analyzed students' data at a private Nigerian institution using k-means clustering. The cluster analysis was combined with standard statistical methods and a deterministic model was k = 3-fold cross-validated using different cluster sizes. The study clustered students labeling them in 5 categories depending on marks' thresholding. However, the study utilized typical academic indicators. Shovon et al. [8] utilized k-means clustering to analyze learning behavior in terms of quizzes, mids and finals in three classes.

C. Supervised, Parametric Learning Approaches

Bhardwaj et al. [9] applied a Naive Bayes classifier on the data of 300 students by preprocessing and transforming the features of raw data. They selected features with probabilities > 0.5. They classified among four classes: first, second, third and fail. The study succeeded in finding interesting features such as living location, mother's qualifications etc. Naser et al. [10] devised an NN based on multilayer perceptron topology and trained it using sophomores' data of five consecutive engineering intakes. They considered high school scores, scores at math and circuitry-based courses during freshman-year, gender among the predictors—gaining 80% accuracy on test-set.

Arora et al. [11] proposed a fuzzy probabilistic NN model for generating personalized prediction which outperformed traditional ML models. The personalized results showed cross-stream generalization capabilities and produced 90%, 96% and 87.5% accuracies on three ranks upon training over 570 instances. The model converged to an error of 0.0265 and included interest, belief, family etc. among eighteen features. Taylan et al. [12] designed an adaptive neuro-fuzzy inference system (ANFIS), a combination of NN and fuzzy systems, to enhance speed and adaptability. The new trend in soft computing produced predictions of students' academics with crisp numerics. Mueen et al. [13] took into account academic participation and scores of two courses and modeled them to Naive Bayes, NN and decision tree—finding the Bayesian classifier to provide the highest accuracy of 86%.

III. Implemented Methodology

Ethical collection of students' data, followed by exploratory analysis, preprocessing, predictive modeling and methodical estimation of metrics led to interesting findings (Fig. 2).

A. Preparation of AUST CS Students' Data

1) Collection of Final Semester's Data:

- **Questionnaire:** Students' responses were gathered via a survey containing questions of multifarious forms including numerical entries, multiple choices and sentential expressions.
- **Environmental setting:** The subjects were surveyed using *Google forms* and the responses were recorded as structured data. There were multiple phases of data-collection either in the labs of AUST or within the comfort of home. No time-constraint allowed subjects to amply think before responding.

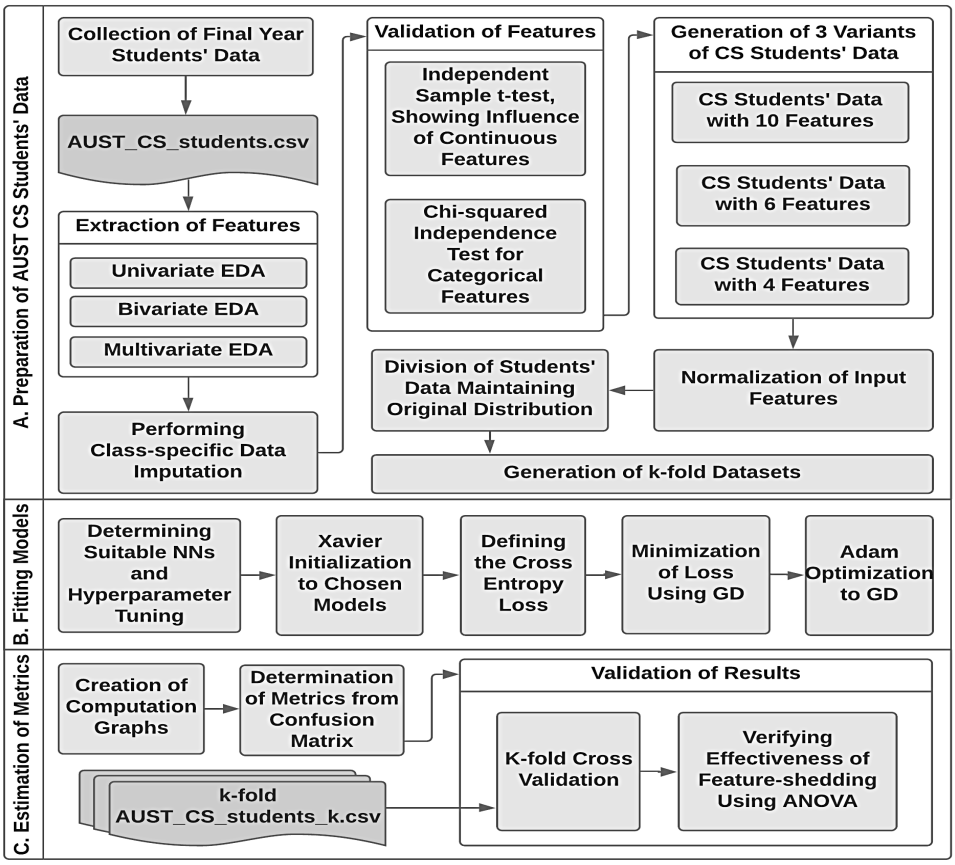

Fig. 2. Workflow of the proposed prediction of academic performance

- **Representativeness:** The sample size (also the population) of 103 subjects represent the whole CS-batch and thus the findings may be generalized among educated youth.
- **Consensual usage:** A pattern recognition lab-project was afoot and students contributed with conscious knowledge and consent to any research thereof.

2) Extraction of Features using Exploratory Data Analysis (EDA): EDA is the statistical process of summarizing tendencies within different attributes of a dataset, assisted by visualizations. The outcome of data-collection, *AUST_CS_students.csv*, had above 30 features and EDA extracted insights beyond predictive analysis to hypothesize features underpinned by data.

A bivariate exploratory visualization (Fig. 3) exposes that pupils with a high attendance rate are the top-scorers (CGPA: 3.3069) and this gradually falls along low and medium attendance. A multivariate observation shows that learners with the strongest passion for both sessional and theory are the highest achievers (CGPA: 3.4398) in terms of academia.

A univariate box-and-whisker exploration (Fig. 4) shows that learners have a median CGPA of 3.25 and programmers investing five or more hours daily in coding are rare. Interestingly, seniors with lower-than-threshold CGPA tend to spend more time (2.261 hours) on social media than their counterparts. The lighter shades of violet tell that either family

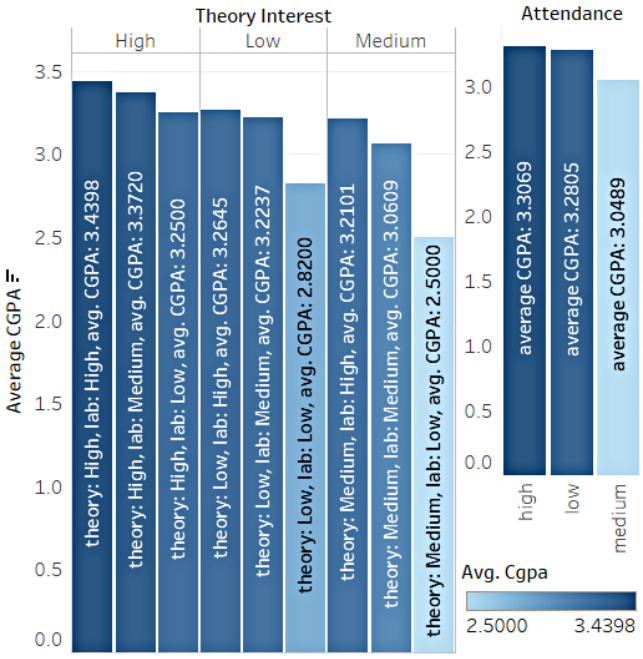

Fig. 3. Relationship of interest in theoretical/sessional CS and attendance (both categorical) with CGPA (continuous)

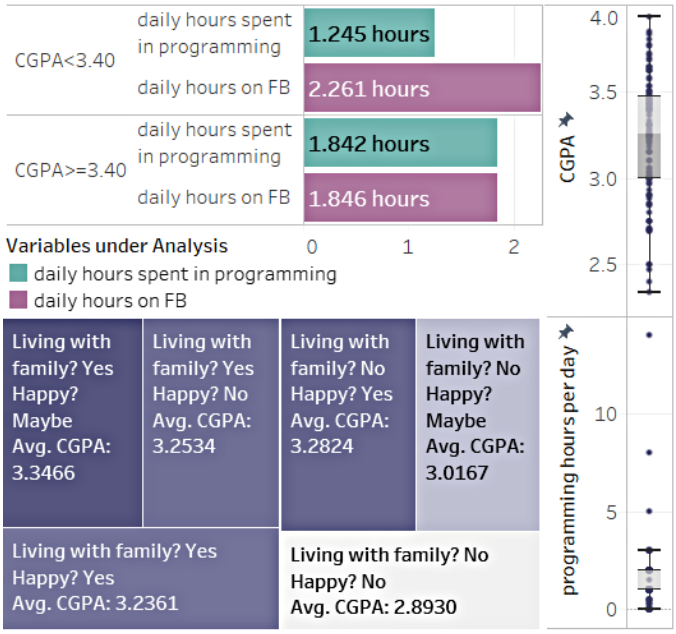

Fig. 4. Univariate and multivariate analyses of lifestyle-factors with CGPA

or happiness should probably be present for a brighter CGPA.

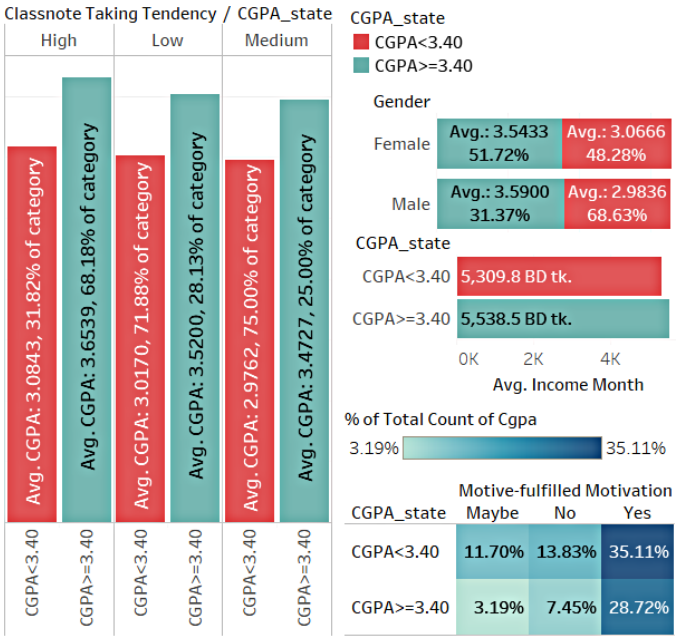

Fig. 5. Thresholded CGPA with respect to preferences (class note, motivation), facts (gender), figures (income)

In another discovery (Fig. 5), class-note taking shows promise in not only that this being high holds the highest CGPA-holders but also in that even the lower-than-threshold students are the highest scorers in their respective category (CGPA < 3.40). More than half (51.72%) of the females hold high CGPA, contrary to their male counterparts.

It is a tendency among students to engage in tutoring and other part-time jobs for self-sufficiency. We find that academically high-achievers tend to earn more than their peers (Fig. 5). Another unintuitive but intriguing cross-tabular finding is that lower-threshold students assert to remain more loyal to their passion (35.11%) even if motive (money, parents' satisfaction, social status) is fulfilled in some other way.

The analyzed attributes clearly show correlations with academic performance and are thus initially justified as features. Data has been visualized according to the best practices, admitting that statistical findings may not always map absolute reality.

3) Performing Class-specific Data Imputation: The statistical process of assigning inferred values to absent fields in accordance with existing fields and summary of the dataset is known as imputation. The *AUST_CS_students.csv* file had numerous blank entries both at categorical and continuous fields, which were eventually filled with class-specific modes and medians respectively (Fig. 6).

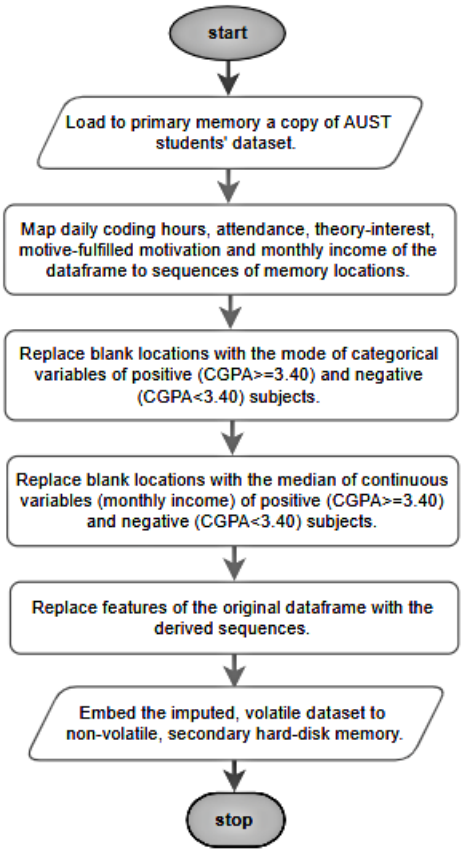

Fig. 6. Class-specialized mode/median imputation algorithm

4) Feature-validation and Generation of Three Variants of Dataset: Inferential statistics is generating statistical models to test hypotheses about a population by producing additional data and eventually deducing propositions using the said model. Most statistical inferences signify *p-value* < 0.05 (a 95% probability of the alternative hypothesis being true), we, however, relax this condition initially and gradually solidify.

To determine if the association between two qualita-

TABLE I. INFERRED STATISTICAL SIGNIFICANCE OF FEATURES

Pearson's χ^2-test			
discrete features	χ^2	**degrees of freedom**	**p-value**
daily hours on FB, state of CGPA	45.254	1	1.73E-11
classnote-taking tendency, state of CGPA	18.553	2	9.36E-05
interest in theory, state of CGPA	4.956	2	8.39E-02
living with family, state of CGPA	2.7991	1	9.43E-02
interest in sessional, state of CGPA	2.7272	2	2.56E-01
attendance in class, state of CGPA	1.978	2	3.72E-01
gender, state of CGPA	0.2086	1	6.48E-01
motive fulfilled motivation, state of CGPA	0.59718	2	7.42E-01
Welch Two Sample t-test			
continious feature	***t*-score**	**degrees of freedom**	**p-value**
daily programming hours, state of CGPA	0.21972	36.864	8.27E-01
monthly income, state of CGPA	-0.63789	24.137	5.30E-01

tive variables is statistically significant, we conduct the χ^2-independence test. Firstly, we define the null hypothesis, $H_\circ$: *no significant association exists between daily hours spent on social media and CGPA*. Conversely, the alternative hypothesis is H_a. To find evidence against $H_\circ$, we compare the observed counts with the expected counts using,

$$\chi^2 = \sum \frac{(observed - expected)^2}{expected} \tag{1}$$

Looking up 45.254 in the χ^2-table for 1 degree of freedom, we find the *p-value*: 1.731E-11, highly statistically significant. Other features are analyzed the same way (Table I). The independent samples *t*-test is a test to determine whether the difference between two groups' (CGPA above or below 3.40) means are significant. If so, an attribute can constitute a feature, where the *t*-statistic is calculated as:

$$t = \frac{\bar{x_1} - \bar{x_2}}{\sqrt{\frac{{s_1}^2}{n_1} + \frac{{s_2}^2}{n_2}}} \tag{2}$$

where,
$\bar{x_1}$ = mean of sample-1
$\bar{x_2}$ = mean of sample-2
n_1 = number of subjects in sample-1
n_2 = number of subjects in sample-2
${s_1}^2$ = variance of sample-1 = $\frac{\sum (x_1 - \bar{x_1})^2}{n_1}$
${s_2}^2$ = variance of sample-2 = $\frac{\sum (x_2 - \bar{x_2})^2}{n_2}$

Not all exploratorily extracted features show a strong rejection of the null hypothesis. We start out by retaining all features and gradually drill down to the more significant ones (e.g., *p-value* < 0.4 and *p-value* < 0.1), thus generating three variants.

5) Normalization of Input Features: Preprocessing mandates inputs and parameters to belong to the same range and scale for fair comparison and for the gradient descent to converge following an aligned orientation.

$$X_{new} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{3}$$

The above formula rescaled all numerics (both categorical: gender, attendance, interest, etc. and continuous: income, daily hours) within the range $[0, 1]$.

6) Maintained Division of Data and k-fold Datasets: Standard ML practices have been followed by assigning a larger set of 80% (83 examples) of total examples for training and the rest 20% (20 examples) for cross-validation. The original distribution of data, i.e., 22.33% positive and 77.67% negative examples, have been maintained throughout training and test data, in order to eliminate any bias during training or cross-validation (Fig. 7).

```
# setting working directory
setwd("F:/4.2/pattern recognition lab/project")

# reading data into a dummy dataframe
dummy <- read.csv("3.4_threshold_after_imputation.csv")

# separating and shuffling subjects with CGPA>=3.40
dummy_ones <- dummy[dummy$cgpa_status==1, ]
dummy_ones <- dummy_ones[sample(1:nrow(dummy_ones)), ]

# separating and shuffling subjects with CGPA<3.40
dummy_zeros <- dummy[dummy$cgpa_status==0, ]
dummy_zeros <- dummy_zeros[sample(1:nrow(dummy_zeros)), ]

# preparing test-set with 20% data
# 77.67% of test data
test_zeros <- dummy_zeros[1:16, ]
# 22.33% of test data
test_ones <- dummy_ones[1:4, ]
test <- rbind(test_zeros, test_ones)

# preparing training set with 80% data
# 77.67% of training data
train_zeros <- dummy_zeros[17:80, ]
# 22.33% of training data
train_ones <- dummy_ones[5:23, ]
train <- rbind(train_zeros, train_ones)

# assembling prepared dataset
dummy2 <- rbind(train, test)
write.csv(dummy2, "3.4_threshold_div.csv", row.names=FALSE)
```

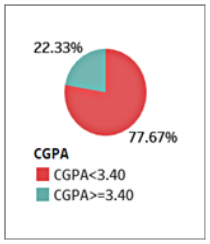

Fig. 7. R script to divide data into an 80%-20% ratio, with the original distribution as inset

K-fold cross-validation is an independent analysis of a model's consistent performance on *k* different training and validation sets. Running the *R* script *k* times provided *k* differently permuted datasets due to shuffle before each binding, thereby allowing the generation of *k*-fold data.

B. Fitting the Models

1) Determining Suitable NNs and Hyperparameter Tuning: Continuous and categorical features' numeric representations were fed to the input layer, with weighted inputs eventually propagating through two *ReLU*-activated hidden layers to the probabilistic *SoftMax* output layer (Fig. 8).

Hyperparameters, upon which the most favorable outcome of a learning model depends besides learnable weights, have been tuned to the following values.

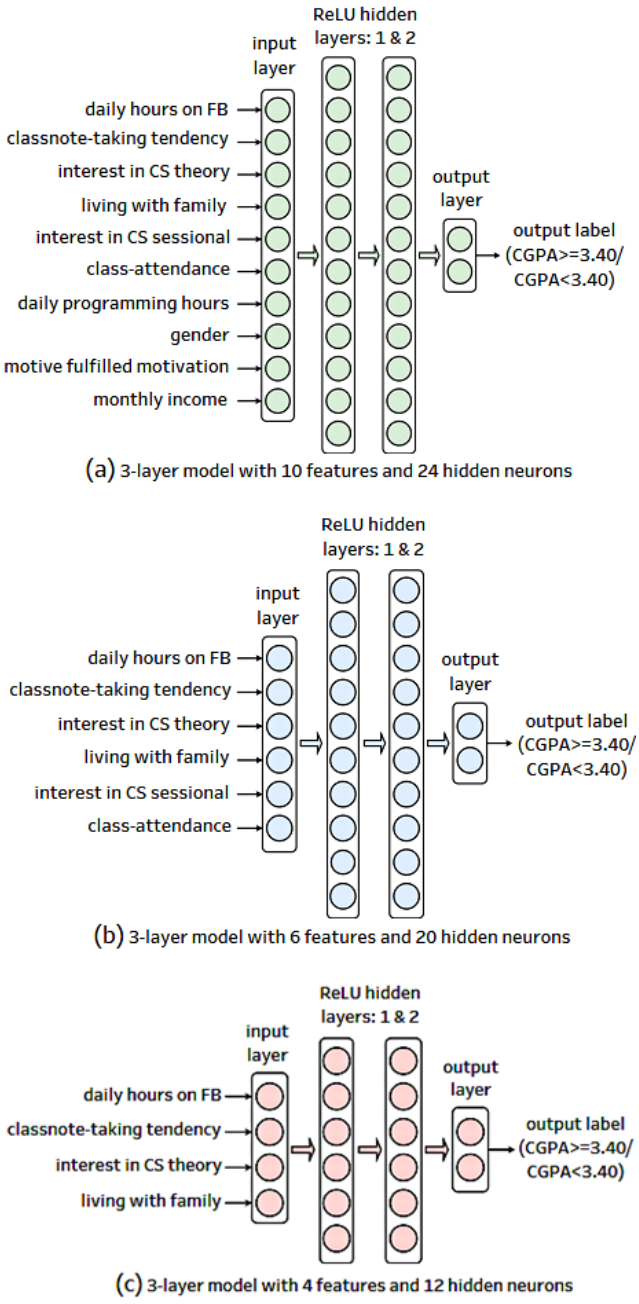

Fig. 8. Proposed three-layer neural network models

- **Number of layers, neurons:** A scarce 83 training examples demanded a simple two hidden-layered network to avoid overfitting. An identical number of hidden neurons were chosen to preclude underfitting. Narrowing the scope to features of greater significance, the complexity reduces; e.g. from 24 (Fig. 8(a)) to 20 (Fig. 8(b)), 12 (Fig. 8(c)).
- **Number of epochs:** 150 for models Fig 8(a, b) and a larger 550 for Fig. 8(c), to converge to an optimum set of parameters.
- **Learning rate:** Depending on epochs, 0.02 for models Fig. 8(a, b) and as small as 0.001 for Fig. 8(c), in order to avoid overshooting across minima.
- **Size of minibatch:** Given the availability of 3.78 GB physical memory, batch gradient descent has been used.

2) Xavier Initialization of Chosen Models: *Xavier* initialization was used for delicate initialization of weights in order to keep them reasonably ranged across multiple layers as:

$$Var(W) = \frac{1}{n_{in}} \tag{4}$$

Where W is the initialization distribution with zero mean for the neuron in question and n_{in} is the number of neurons feeding in. The distribution is typically *Gaussian* or uniform.

3) Defining the Cross-Entropy Loss Function: The cross-entropy loss has been optimized for the classification problem with a view to obtaining most optimally refined parameters. Here we represent the precise cross-entropy [14], summed over all training examples:

$$\begin{aligned} -logL(\{y^{(n)}\},\{\hat{y}^{(n)}\}) &= \sum_n [-\sum_i y_i \, log \, \hat{y}_i^{(n)}] \\ &= \sum_n H(y^{(n)}, \hat{y}^{(n)}) \end{aligned} \tag{5}$$

where n denotes the number of training examples, $y^{(n)}$ indicates the ground-truth for a separate example, $\hat{y}^{(n)}$ is prediction generated by the model and i renders the sequence of activation within a layer.

4) Minimization of Loss using Gradient Descent: A set of parameters θ was to be selected in order to minimize loss $J(\theta)$. Gradient descent algorithm [14] initialized θ, then repeatedly performed the following update.

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta) \tag{6}$$

This update was parallelly performed for all features, i.e., $j = 0, 1, ..., n$ with α being the learning rate. This is a quite natural algorithm that iteratively took steps towards the steepest decrease of $J(\theta)$. Its implementation required the partial derivative term to be computed. Considering only one training example (x, y), we have:

$$\begin{aligned} \frac{\partial}{\partial \theta_j} J(\theta) &= \frac{\partial}{\partial \theta_j} \frac{1}{2} (h_\theta(x) - y)^2 \\ &= 2.\frac{1}{2}(h_\theta(x) - y).\frac{\partial}{\partial \theta_j}(h_\theta(x) - y) \\ &= (h_\theta(x) - y).\frac{\partial}{\partial \theta_j}(\sum_{i=0}^{n} \theta_i x_i - y) \\ &= (h_\theta(x) - y)x_j \end{aligned}$$

$$Therefore, \quad \theta_j := \theta_j + \alpha(y^{(i)} - h_\theta(x^{(i)}))x_j^{(i)} \tag{7}$$

To modify the above for a set of more than one examples, the statement should be replaced by the algorithm below:

Repeat until convergence {

$$\theta_j := \theta_j + \alpha \sum_{i=1}^{m} (y^{(i)} - h_\theta(x^{(i)}))x_j^{(i)} \quad (for\ every\ j) \qquad (8)$$

}

5) Adam Optimization to Gradient Descent: Adam is a first-order gradient-based optimization algorithm for stochastic objective functions, using adaptive estimates of lower-order moments. The parameters used for *Adam* in this study are as follows:

- $\boldsymbol{\alpha}$ **:** The learning rate or step size, whose decay is permissible for *Adam*, but has not been used.
- $\boldsymbol{\beta_1}$ **:** The exponential decay for first-order moment estimates (e.g. 0.9).
- $\boldsymbol{\beta_2}$ **:** The exponential decay for second-order moment estimates (e.g. 0.999).
- $\boldsymbol{\epsilon}$ **:** An infinitesimal number to prevent division by 0 in the implementation (e.g. 10E-8).

C. Estimation of Metrics

1) Creation of Computation Graphs: A computation graph is a collective mathematical function represented using the frameworks of graph theory. The round nodes indicate operations while the rectangular ones denote operands, with the directed edges delineating the sequence of mathematical operations performed.

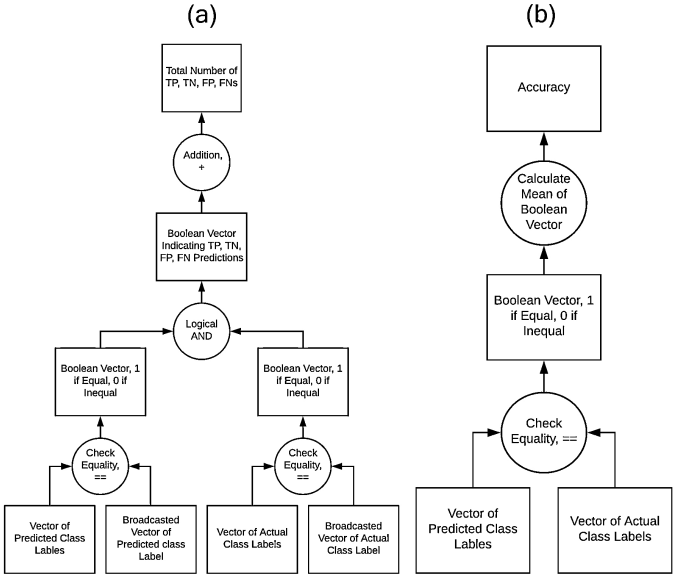

Fig. 9. (a) Generalized computational graph to determine entries associated with confusion matrix; (b) Computation graph portraying computation of accuracy.

TensorFlow's NN framework requires a computation graph to be devised before running a session to refine numerics. The one-hot Boolean representation of class labels has been used to concoct two bottom-up graphs in order to determine entries associated (Fig. 9(a)) with confusion matrices and accuracy on cross-validation set.

After equality-checking, the boolean vector of outputs gave 'high's against the examples identified correctly and 'low's against the converse as to having a CGPA above the threshold. The mean of this data structure rendered the fraction of correct identification (Fig. 9(b)).

2) Determination of Metrics from Confusion Matrix: In the domain of statistical classification, a confusion matrix (Fig. 10(a)) is a special type of contingency table with identical sets of classes in both dimensions—used to account for the performance of a classification model on cross-validation data for which the actual labels are available.

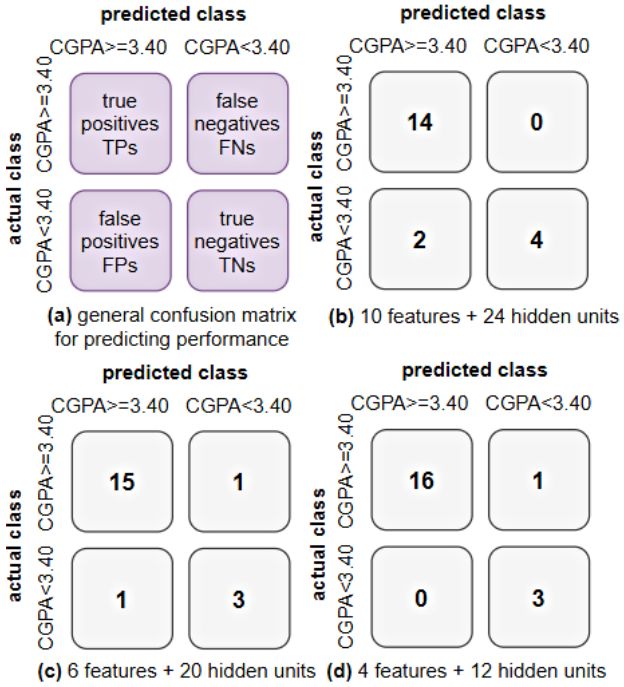

Fig. 10. Confusion matrices of our models for some random *k*-th cross-validation

Rows of the tabular layout (Fig. 10) represent instances in an actual class and columns represent predicted labels. The name originates from its making viable to verify if the system is confusing the classes. For our binary classification, we select the popular accuracy, precision, recall and F1-score as evaluative metrics.

- **Accuracy:** proportion of actually correct predictions (both upper and lower-threshold),

$$accuracy = (TP + TN)/(P + N)$$

- **Precision:** proportion of actually correct CGPA>=3.40 identifications,

$$precision = TP/(TP + FP)$$

- **Recall:** proportion of actual CGPA>=3.40 was identified correctly,

$$recall = TP/(TP + FN)$$

- **F1-score:** a trade-off between accuracy and precision, their harmonic mean,

$$F1\text{-}score = (2 * TP)/(2 * TP + FP + FN)$$

3) K-fold, ANOVA-tested Validation of Improvement in Models: Hypothesis-testing technique *ANOVA* (Analysis of Variance) tested the incremental improvement of proposed models' mean accuracies by examining their variances (each having k = 5 instances). The samples are random and independent, to the fulfillment of *ANOVA*'s assumptions.

Equality of all sample means is the null hypothesis of *ANOVA*. Hence, $H_\circ$: $\mu_1 = \mu_2 = \mu_3$. Thus, the alternative hypothesis is given as, H_a: *The mean accuracies are reliably unequal*. It essentially calculates the ratio:

$$F = variance\ between\ groups\ /\ variance\ within\ groups$$

The greater the ratio, the more the likelihood of rejection of $H_\circ$. The results of *ANOVA* is written in the format $F(b, w)$ where b and w are degrees of freedoms between and within groups, respectively.

Here,
b = number of groups – 1
w = total number of observations – number of groups

IV. Experimental Results and Discussion

The results originating from fitting three models using the CS students' data are transcending in that even the lowest achieved accuracy surpassed orthodox learning algorithms reviewed in the literature. Application of class-specific median and mode imputation ensured no shrinkage of the already small dataset of 103 tuples, leading to the best use of already existing and further inferred data. Features have been cut down and models' complexity has been gradually reduced, all statistically validated.

For some random k, the cross-entropy loss fell with each epoch during training the first two models through 150 epochs with a learning rate of 0.02 (Fig. 11(a, b)). The training was stopped when the error plateaued to a reasonably small value. The third model was trained for 550 epochs with a 0.001 learning rate, whose k-fold (k = 5) cooling down of error from warmer-shaded greater errors are shown in (Fig. 11(c)).

Firstly, we present the 5-fold consistent results fitting the 10-feature model (Fig. 8(a)) on different cross-validation sets (Fig. 12). The k = 5 cases of a consistent 90% test-accuracy can be differentiated by optimized training errors. The model seems to fit training data impressively and is already surpassing traditional models in accuracy (Fig. 14). All cross-validations are consistently giving promising F1-scores (greater or equal to 0.75).

Secondly, we fit another model (Fig. 8(b)) with the same hyperparameters except that now we extract out 6 most significant features as per Table I instead of retainment of all exploratorily discovered features. This scaled down the model's complexity from 24 hidden units to 20. The 5-fold cross-validations resemble training and testing accuracies closely,

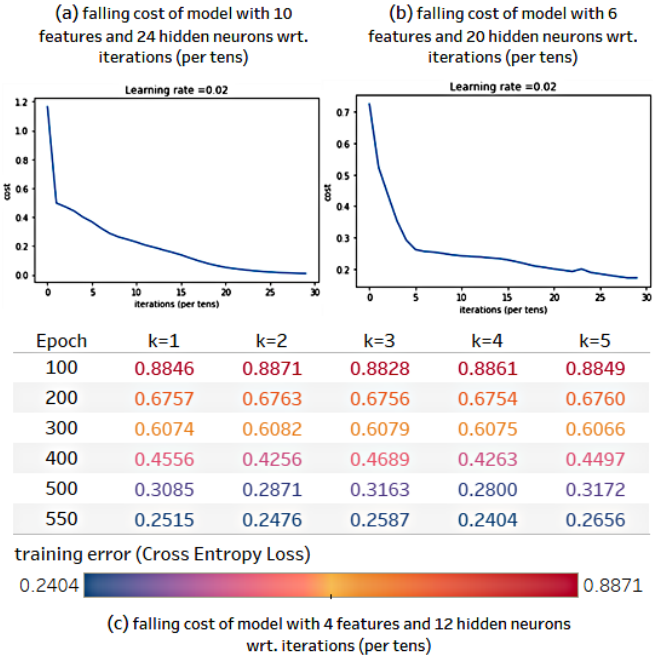

Fig. 11. 10, 6-Feature models' learning curves and 4-feature model's lessening of error with epochs

TABLE II. 5-Fold cross-validated results upon training the 4-featured 3-layer final model

k-fold	optimized training loss	test accuracy	precision	recall	F1-score
1	0.251543	0.95	1	0.75	0.857143
2	0.247627	1	1	1	1
3	0.258734	0.95	1	0.75	0.857143
4	0.24039	0.95	0.8	1	0.888889
5	0.265609	0.95	1	0.75	0.857143

leading to perfect fitting with test-accuracies as impressive as the former model (Fig. 12).

Finally, we become more selective by cherrypicking features with more stringent *p-values* < 0.1 (90% chance of the alternative hypothesis to be true). The network (Fig. 8(c)) thus deprecated its complexity to just 12 hidden neurons, yielding comparatively the most promising (Fig. 13) and consistent (Fig. 12) metrics.

TABLE III. ANOVA-test results verifying the incremental improvement of models

ANOVA (Analysis of Variance) test metrics	Values
degrees of freedom for numerator (ind)	2
degrees of freedom for denominator (residuals)	12
sum of squares of numerators (ind)	0.012
sum of squares of denominators (residuals)	0.002
mean of squares of numerators (ind)	0.006
mean of squares of denominators (residuals)	0.000167
analysed value	36
p-value, Pr(>F)	8.50E-06

Applying *ANOVA* on test-accuracy data from Fig. 12 and Table II, we attempt to test whether the mean accuracies of

F1-score
0.7500 0.9474

NN architecture + 'n' features	k-fold	optimized training loss	training accuracy	test accuracy	true negatives, TNs	false negatives, FNs	false positives, FPs	true positives, TPs	precision	recall	F1-score
3-layer NN model with 6 features	1	0.1750	0.9157	0.9000	15	1	1	3	0.7500	0.7500	0.7500
	2	0.1698	0.9157	0.9000	15	1	1	3	0.7500	0.7500	0.7500
	3	0.1392	0.9277	0.9000	14	0	2	4	0.6667	1.0000	0.8000
	4	0.1863	0.9157	0.9000	14	0	2	4	0.6667	1.0000	0.8000
	5	0.1807	0.9036	0.9000	14	0	2	4	0.6667	1.0000	0.8000
3-layer NN model with 10 features	1	0.0393	1.0000	0.9000	14	0	2	4	0.6667	1.0000	0.8000
	2	0.1443	1.0000	0.9000	15	1	1	3	0.7500	0.7500	0.7500
	3	0.0524	1.0000	0.9000	16	2	0	2	1.0000	0.9000	0.9474
	4	0.0686	0.9880	0.9000	15	1	1	3	0.7500	0.7500	0.7500
	5	0.1211	0.9880	0.9000	15	1	1	3	0.7500	0.7500	0.7500

Fig. 12. 5-Fold cross-validated results upon training 10, 6-featured 3-layer models (6-featured 3-layer model better fitting the data by overcoming overfitting)

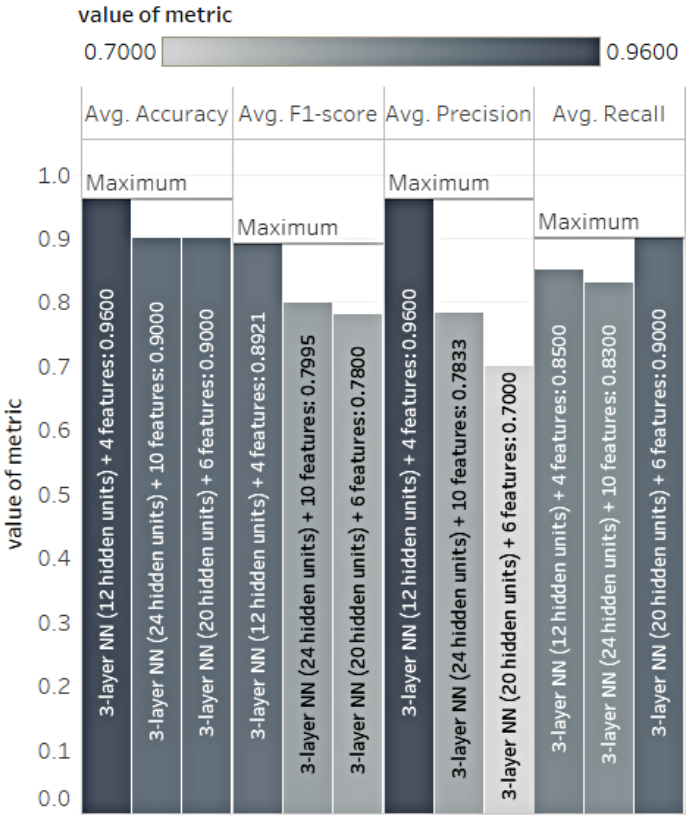

Fig. 13. Comparison among proposed models' average performance measures

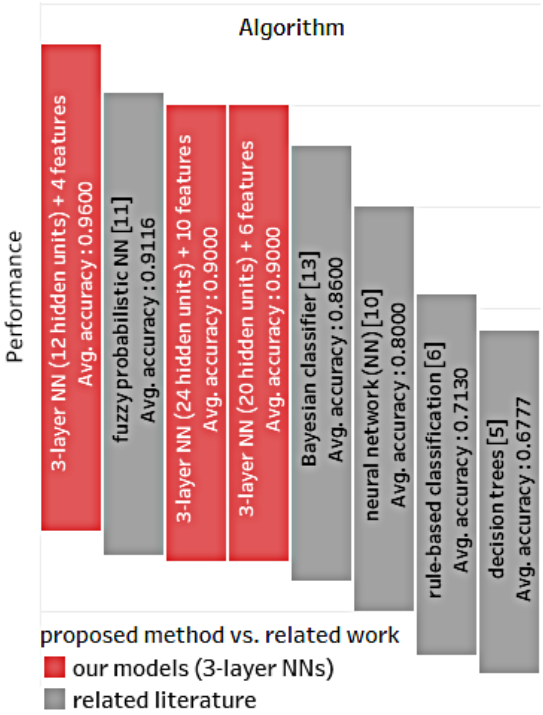

Fig. 14. Comparison between our methodology and reviewed literature

the architectures are systematically different or are just due to sampling errors. The *ANOVA* results (Table III) show:

$$F(2, 12) = 36,\ p\text{-}value = 8.50\text{E-}06 < 0.05,$$

leading us to safely conclude, the models have a systematic effect on the accuracy and similar results can be expected if further data-points are added.

A comparative analysis (Fig. 13) reveals that the most optimized model does brilliantly in accuracy, precision and F1-score. The 6-feature model performs best in terms of average recall. Deployment of the suitable model should be done carefully as different models excel differently. Another comparative study (Fig. 14) manifests that the 3-layer NNs proposed in this paper outsmart many existing methods utilized to solve similar problems.

V. Conclusion

The curious problem of predicting students' performance has, till date, been addressed using direct predictive modeling—this paper proves the effectiveness of visually exploratory and statistical analysis prior to that objective, leading to the following landmarks.

- The study avoids random, carefree, holistic selection of features by first examining their relevance through hypothesis testing, thus establishing the importance of statistical preprocessing.
- The research endorses data-engineered median and mode imputation in handling missing values, introducing no outside noise to training data.
- The paper testifies robustness of the incrementally developed proposed models through k-fold cross-validated, *ANOVA*-tested, significant results.

It is recognized that setting the threshold to a CGPA of 3.40 may not epitomize aptitude, which depends on factors external to the scope of this endeavor. However, this study approves and incentivizes further researches to consider lifestyle and personal preferences as useful features towards that end.

REFERENCES

[1] Ryan SJD Baker and Kalina Yacef. The state of educational data mining in 2009: A review and future visions. *JEDM— Journal of Educational Data Mining*, 1(1):3–17, 2009.

[2] Cristóbal Romero and Sebastián Ventura. Educational data mining: a review of the state of the art. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 40(6):601–618, 2010.

[3] David Wilkinson, Jianzhen Zhang, Gerard J Byrne, Haida Luke, Ieva Z Ozolins, Malcolm H Parker, and Raymond F Peterson. Medical school selection criteria and the prediction of academic performance. *Medical journal of australia*, 188(6):349–354, 2008.

[4] Tomas Chamorro-Premuzic and Adrian Furnham. Personality predicts academic performance: Evidence from two longitudinal university samples. *Journal of research in personality*, 37(4):319–338, 2003.

[5] Surjeet Kumar Yadav and Saurabh Pal. Data mining: A prediction for performance improvement of engineering students using classification. *arXiv preprint arXiv:1203.3832*, 2012.

[6] Fadhilah Ahmad, Nur Hafieza Ismail, and Azwa Abdul Aziz. The prediction of students' academic performance using classification data mining techniques. *Applied Mathematical Sciences*, 9(129):6415–6426, 2015.

[7] OJ Oyelade, OO Oladipupo, and IC Obagbuwa. Application of k means clustering algorithm for prediction of students academic performance. *arXiv preprint arXiv:1002.2425*, 2010.

[8] Md Hedayetul Islam Shovon and Mahfuza Haque. Prediction of student academic performance by an application of k-means clustering algorithm. *International Journal of Advanced Research in Computer Science and Software Engineering*, 2(7), 2012.

[9] Brijesh Kumar Bhardwaj and Saurabh Pal. Data mining: A prediction for performance improvement using classification. *arXiv preprint arXiv:1201.3418*, 2012.

[10] S Abu Naser, Ihab Zaqout, Mahmoud Abu Ghosh, Rasha Atallah, and Eman Alajrami. Predicting student performance using artificial neural network: In the faculty of engineering and information technology. *International Journal of Hybrid Information Technology*, 8(2):221–228, 2015.

[11] Nidhi Arora and JR Saini. A fuzzy probabilistic neural network for student's academic performance prediction. *International Journal of Innovative Research in Science, Engineering and Technology*, 2(9):4425–4432, 2013.

[12] Osman Taylan and Bahattin Karagözoğlu. An adaptive neuro-fuzzy model for prediction of student's academic performance. *Computers & Industrial Engineering*, 57(3):732–741, 2009.

[13] Ahmed Mueen, Bassam Zafar, and Umar Manzoor. Modeling and predicting students' academic performance using data mining techniques. *International Journal of Modern Education and Computer Science*, 8(11):36, 2016.

[14] Ng, A., 2000. CS229 Lecture notes. CS229 Lecture notes, 1(1), pp.1-3.