Towards Intelligent E-Learning Systems: A Hybrid Model for Predicatingthe Learning Continuity in Iraqi Higher Education

Mohammed K. Kadhim*
Department of Computer Science, University of Technology, Baghdad, Iraq.
E-mail: mohammedbabylon84@gmail.com

Alia K. Hassan
Department of Computer Science, University of Technology, Baghdad, Iraq.
E-mail: 110018@uotechnology.edu.iq

Received June 02, 2020; Accepted August 04, 2020
ISSN: 1735-188X
DOI: 10.14704/WEB/V17I2/WEB17023

Abstract

E-Learning system gains a great attention in the past years; with advance of the internet and the information exchange techniques the importance to merge the traditional learning means with the internet-based learning methods became a must especially in Iraq, the Iraqi higher education is now coping with the new information and communication technologies and adopting a modern methods for upgrading their education and learning ways. There are great efforts to blend E-Learning systems with the educational process, in order to fulfill this purposes the proposed research is advancing E-Learning systems by suggesting a hybrid method that combines two Artificial Intelligence Techniques (AI) inside the design and the development of an intelligent E-Learning system for computer science department at university of technology. The utilization of Artificial Neural Networks algorithm (ANNs) especially Recurrent Neural Networks (RNN) is a way of implementing deep learning technique to predict the students' final out comes in virtual class room based on their grades and their learning behaviors. RNN is optimized by utilizing ADAM optimizer to lift the accuracy of the proposed algorithm, the dataset are gathered and processed to suite the education purposes and was divided into80% for training the model and 20% for testing the model, the results of the hybrid model are compared with other machine learning methods like Multi-Layer Perceptron (MLP), decision tree, naïve Bayesian, and random forest using WEKA environment, the results of the proposed model showed a promising accuracy when compared with the mentioned machine learning algorithms.

Keywords

E-Learning, Learning Continuity, Artificial Intelligence, Artificial Neural Networks, Recurrent Neural Network, ADAM, and WEKA.

Introduction

Creating intelligent online environment in learning and education field is required especially when there is no direct interaction between the learner and the instructor, although E-Learning systems have a great impact on advancing the learning process, but the feeling of isolation that the students' are experience is wide. Therefore, an intelligent E-Learning system that capable on tracing the students' grades, outcomes, and learning behaviors inside the system is proposed, through this the drop-out prevention of the courses will be eliminated and the reasons behind the students' fail in a given class room will be translated and noticed so that the supervisors and faculty members can overcome these situations. Deep Learning [1][2][3] is sub-field of machine learning that has been applied for long time to solve many realistic problems and it is processed in different fields like health care, robotics, natural language processing (NLP) and Artificial Neural Networks (ANN), in this work the algorithms are made from Recurrent Neural Networks (RNN).Deep learning model is processed by various levels, input, output and hidden levels, the model is a network like and it is trying to mimic the human brain and how the reasoning process is taken, the input level contains the input nodes which transform the all the required data into the hidden level where the hidden nodes lies, hidden nodes make all the processing equations then pass the results to more abstract form to achieve deeper form, finally the network passes the results to the output level. In this paper, Artificial Neural Networks (ANNs) algorithms especially RNNs and ADAM optimizer are utilized to predict the outcomes of the students a given course, then the results obtained from the model is then analyzed to target the students that are most likely to drop the course in a process called drop-out interaction, this process will be implemented by two stages, first one is the utilization of Recurrent Neural Networks (RNN) to train the classifier based on students grades from their learning profiles inside the proposed, then the results of the previous stage will be improved by using adaptive momentum (ADAM) optimizer to optimize the accuracy of (RNN) algorithm which will be our contribution to this work.

Literature Review

A considerable number of researches have used the Neural Network method in the context of E-Learning system. For example, in [17], the authors have been discussed that the electronic course that has the properties required in the view of Smart education is an effective tool for non-formal and formal learning, in which most motivated students are interested now for obtaining high-quality knowledge. In [18], the authors have been discussed the definition of smart education and presents a conceptual framework. A four-tier framework of smart pedagogies and ten key features of smart learning environments are proposed for foster smart learners who need master knowledge and skills of the 21st century learning. The smart pedagogy framework includes class-based differentiated instruction, group-based collaborative learning, individual-based personalized learning and mass-based generative learning. A Smart Cloud E-Learning System with Social Networking based on architecture level of social networking and E-Learning cloud system to support E-Learning interactions in worldwide environments have been presented in [19], the researchers suggested a prototype of Cloud ELearning System with Social Networking is developed on cloud, and several applications features are described based on the proposed architecture to demonstrate the effectiveness of the architecture. The researcher in [20], deals with the application of the concepts of Internet of Things and its application in creating smart environments. The specific goal is to design a smart environment for enhancing the teaching and learning processes at universities. In [21], a model of students classification to predict the students' performance were presented, the research have shown that academic performance related to the first-year bachelor students in Computer Science course. Data have been acquired for a period of eight years from (Jul 2006/2007- Jul 2013/2014) which contain the family backgrounds, previous academic records, and demographics of students. Rule-based, Naïve Bayes, and Decision Tree approaches have been utilized to the data of students for the purpose of the research. In this research used decision tree along with neural network algorithm to assess the rate of the accuracy for the presented prediction into the academic students' achievement of the students, then the study analyzed the factor that affects the achievements of the student that affects the prediction for under-graduated studies. The data set comprised of 1,600 student records with 22 attributes of students registered between year 2001 and 2011. Results show that the decision tree classifier achieves high accuracy of 85.188%, which is higher than that of neural network classifier by 1.313%. In presented a novel study on signature verification based on Recurrent Neural Networks (RNNs), the study goal were to learn the differences metric among pair of signatures. It is considered as the first study in the on-line signature verification, both Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) systems with a Siamese architecture. In addition, a bidirectional scheme (which is able to access both past and future context) is considered for both LSTM- and GRU-based systems. The utilized database comprised of 400 users who contributed a total of 11,200 signatures in four separated acquisition sessions. Results achieved had the state-of-the-art on-line signature verification systems using the same database. In [24], a study in the field of deep learning were conducted, in this context, It is proposed to build a Deep Learning model using various Optimizers (Adagrad, RmsProp, Adam), Loss functions (mean squared error, binary cross entropy) and Dropout concept for the Convolutional neural networks and Recurrent neural networks and verify the performance such as Accuracy and Loss of the model. Multi optimizers are applied to the mentioned methods, when used ADAM optimizer a highest accuracy of the model have achieved.

Case Study

In order to train and test the proposed algorithm, first learning dataset needs to be gathered and organized, and this will be provided from creating and implementing a virtual learning environment which is called "smart education" this is provided by a case study about at university of technology especially computer science department and the courses that are embedded in the system are the same as the curriculum of artificial intelligence branch of computer science department which are: (artificial intelligence course, computer networks course, database course, operating systems course, computer security course, and software engineering course) along with extra course which is English language course, and an electronic library to involve all books about the courses. In order to distinguish the students from the teachers a special registration form will be used to register both of the users. The learning journey of each student from the moment that he/she registers inside the platform is captured by the system, and saved into a students' profile that is directly connected with the system database so that the models can be trained on the given dataset of the students' learning. The courses are divided into levels so that the students can enter the level that suites their knowledge best, the division of the levels gives the scene of the students that the proposed system mimics their learning abilities. The process of courses division will be programmed inside the system so that whenever a new student registers the system will direct him/her towards the suitable level after taking a pretest, this test will assess the level of the student knowledge and scientific background about the course. Pseudo code of this process in the following:

```
Start
Initialize: K: pretest score, L: level of the course where H: advance part, I: medium part, and J: easy part.
Input: Exam Score
Output: Part of the subject
Step1: Start a for loop
M= 1 // * from the first question in the exam
For (N=M, N<=10, N++) //* each exam has ten questions
Compute C //* List of questions
End for
Step2: If K <= 100 && K>=80 Then L = H
```

```
Else If K <= 70 && K>=50 Then L = I
Else If K <= 40 && K>=0 Then L = J
Else "The Student fails" // *must retake the exam to be evaluated
Step3: Save all the scores into the database, and update user profile, then suitable level will be showed to the student.
End
```

The above process is called that he system recommends parts of the course to the students, a great part of The available internet-based learning environments have fixed learning curriculum and don't meet the standard of the advanced E-Learning systems, more dynamic aspects are needed to be used in there environments to allow the learners to meet more personalized learning patterns [4]. In the proposed system a virtual class at University of Technology (UoT) in Computer Science Department is implemented; the system is designed especially to give electronic courses for the fourth class studies and to cover the curriculum of six different courses. Recommender system will provide the students the flexibility in learning and to reduce the time they needed to finish the course, plus these courses will be previewed within basic standards. Figure (1.1) shows recommender system architecture as the following:

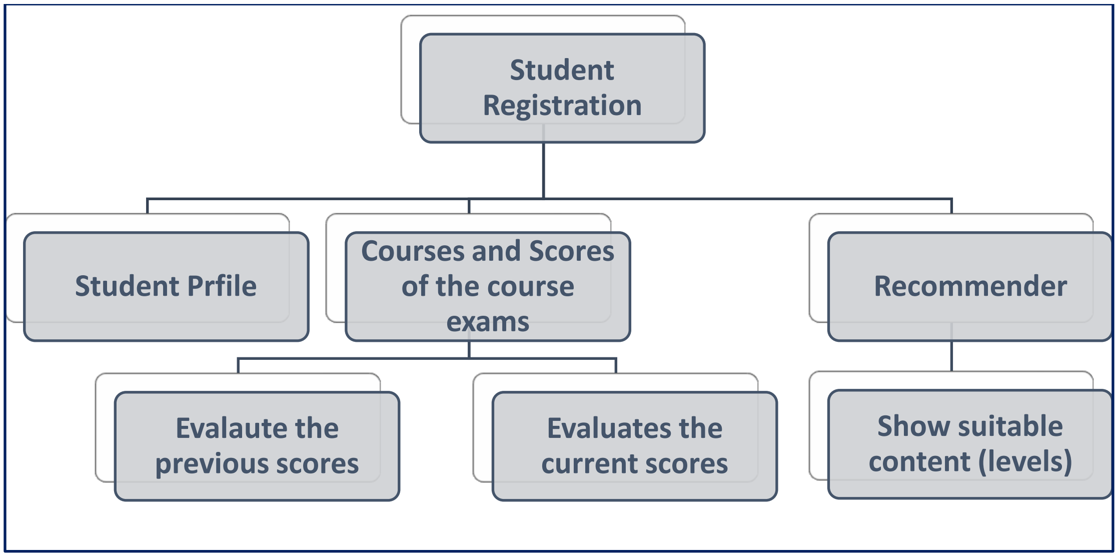

Student Registration
Student Prfile
Courses and Scores of the course exams
Recommender
Evalaute the previous scores
Evaluates the current scores
Show suitable content (levels)

Figure 1.1 Structure of recommendation system

The proposed E-Learning system will evaluate the student's overall knowledge of the course and create a unique student profile that retains all the scores as mentioned, while simultaneously displaying the exact content that meets each student's needs. Table (1.1) shows the customization of the recommender system:

Table 1.1 Recommendation Table

Score	Level
From 0 to 40	Level-1-
From 50 to 80	Level-2-
From 80 to 100	Level-3-

After the students have engaged inside the system, their grades and behaviors will be saved and analyzed then extracted to be as an input for the intelligent predication to know who will drop the course and who will pass and complete the course. The dataset were used to train and test the proposed algorithm is special learning dataset conducted from (Jason file) from the system's database. The database is important in order to save the system data like: (courses, students' scores, students' profile, teachers' evaluation, users' accounts, and other information). The dataset will be divided into 80% for training part and 20% for testing part. Selecting the dataset has two processes, one for selecting the desired rows and columns S: stands for selected feature, U: Stands for unselected feature, then a specific weight is assigned to each selected and un-selected feature. This process is also repeated to take all the attributes of the dataset it will be illustrated in details as table (1.2):

Table 1.2 Features selection from the dataset

No.	Feature	Details	Selection	Weight
1.	Id	Student ID	U	0
2.	First Name	Client name	U	0
3.	Last Name	Client family name	U	0
4.	University	University name	U	0
5.	E-mail	E-mail address	U	0
6.	Password	Unique password	U	0
7.	CP	Continuity prediction	S	10
8.	SQ1	score in 1st exam	S	10
9.	SQ2	score in 2nd exam	S	10
10.	SQ3	score in 3rd exam	S	10
11.	CL	Course Level	S	8
12.	PL	Pass Level	S	6
13.	CR	Course Repeating	S	4
14.	CC	Course certification	S	2

Figure (1.2) and (1.3) shows the stages required to implement the training phase.

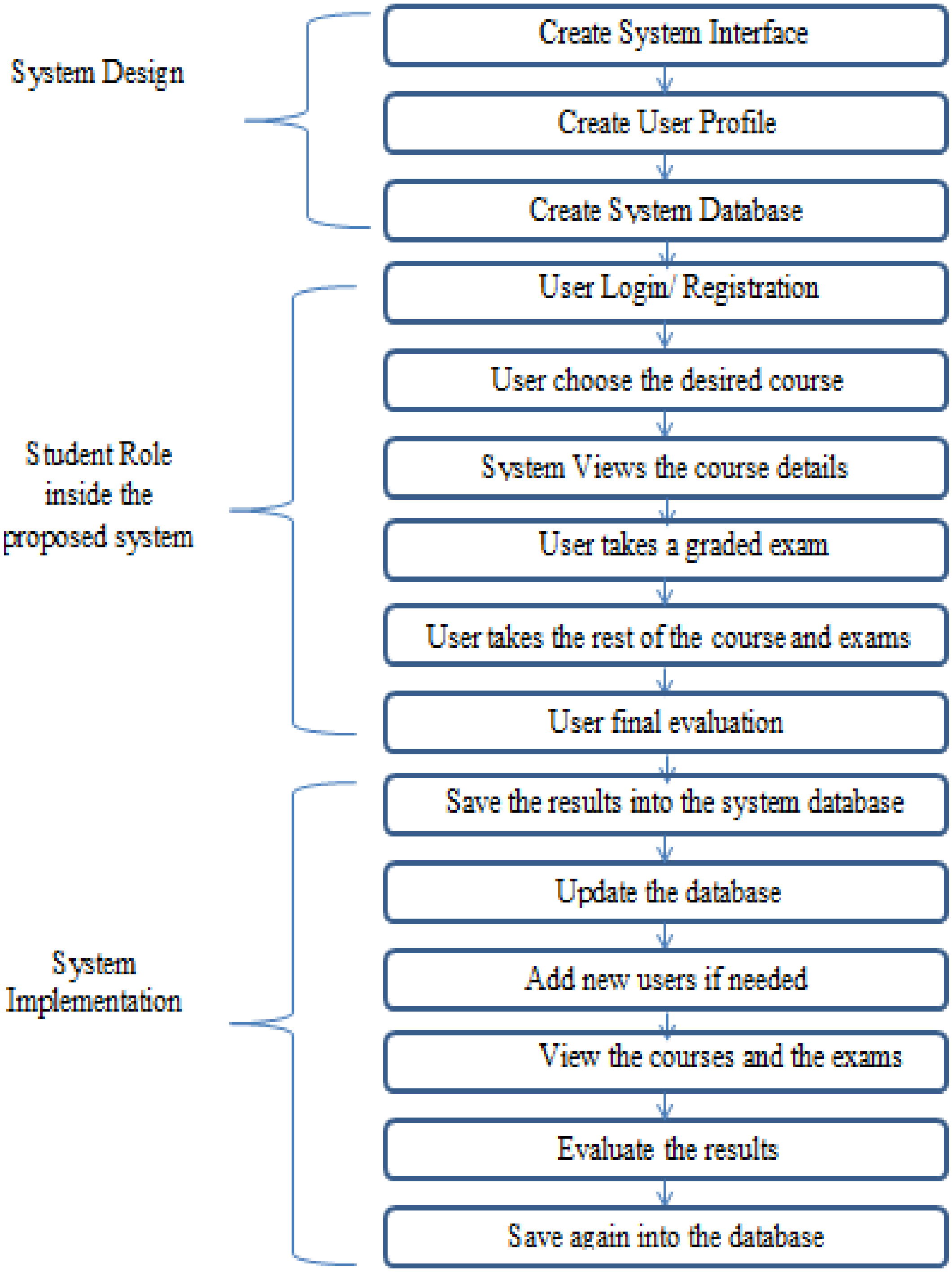

System Design
Create System Interface
Create User Profile
Create System Database
Student Role inside the proposed system
User Login/ Registration
User choose the desired course
System Views the course details
User takes a graded exam
User takes the rest of the course and exams
User final evaluation
System Implementation
Save the results into the system database
Update the database
Add new users if needed
View the courses and the exams
Evaluate the results
Save again into the database

Figure 1.2 The stages required to implement the training phase

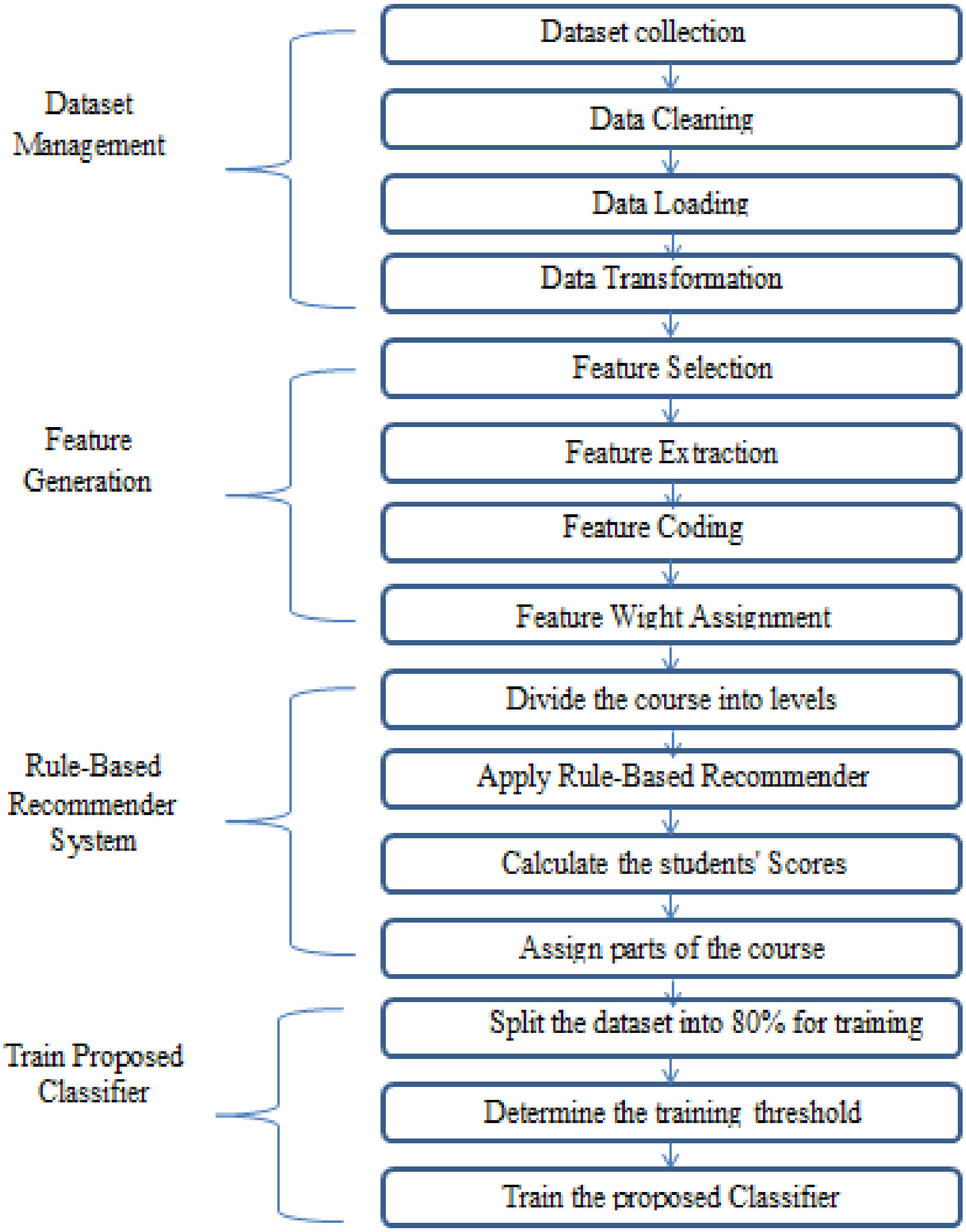

Dataset Management
Dataset collection
Data Cleaning
Data Loading
Data Transformation
Feature Generation
Feature Selection
Feature Extraction
Feature Coding
Feature Wight Assignment
Rule-Based Recommender System
Divide the course into levels
Apply Rule-Based Recommender
Calculate the students' Scores
Assign parts of the course
Train Proposed Classifier
Split the dataset into 80% for training
Determine the training threshold
Train the proposed Classifier

Figure 1.3 The stages required to implement the training phase

Practical Work

The contribution of this paper is to create an intelligent E-learning system that can predict the final students' learning continuity in an online class room environment and E-courses space, to reach this target supervised learning will be used to implement deep learning algorithm and then optimize the results to be more accurate. Recurrent Neural Networks will be used and optimized with Adaptive momentum optimizer**.** Figure (1.4) views the overall structure of this part:

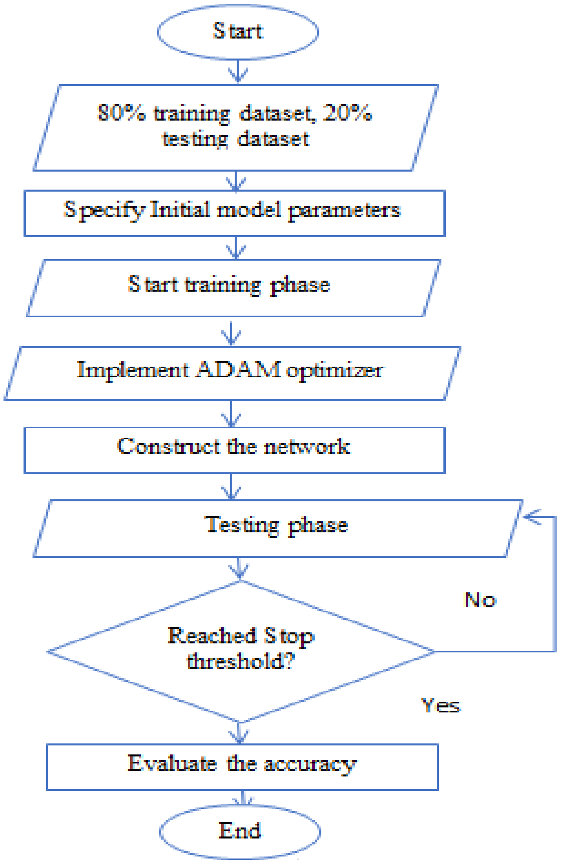


Figure 1.4 Flowchart of practical work

In this paper RNN algorithm will be used along with Adaptive Momentum (ADAM) optimizer to predict the learning continuity of the students in the proposed learning environment. System (Continue or abandon the course) based on the previously prepared dataset, while (ADAM) algorithm will optimize the final results of (RNN) algorithm by minimizing the error rate. Recurrent Neural Networks (RNN) [5] has the ability to make multiple processes from diverts input length of sequenced data by recursively implementing a special function to the hidden layer vector this function is called a transition function. The main concept behind RNN [6][7][8] is the it can process a sequence of inputs and creates a special memory to retrain these processed values in the memory after implementing the activation function, this step is preserved at each phase of the processing between the layers (input, hidden and the output) layers. To make a prediction the network should has the output of the previous iteration and the current iteration, the previous iteration is kept in its special storage until there is no input in the sequence, this process is called long short term memory (LSTM) [9]. Figure (1.5) shows the difference between RNN and ordinary neural network:

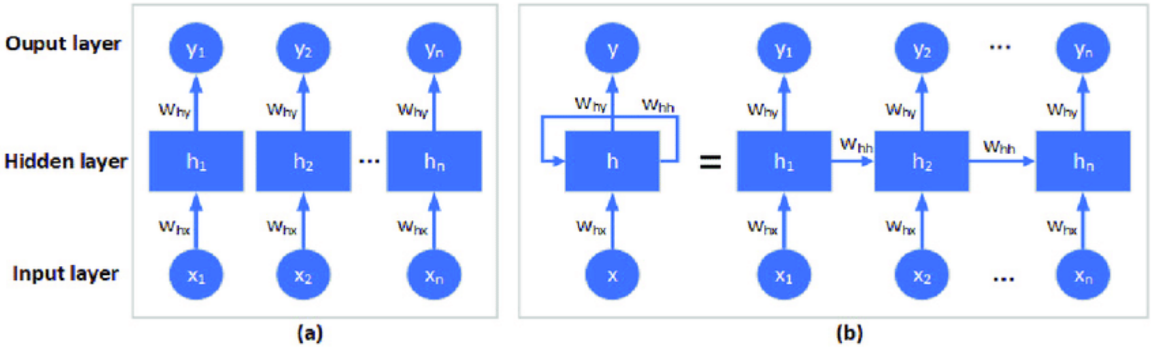


Figure 1.5 The architecture of (a) Traditional Artificial Neural Network (ANN) and (b) Recurrent Neural Networks (RNN). [10]

in figure (1.6) where all the required parameters will be explained then the functions and the equations will be detailed as the main training phase classifier, to keep in mind that the optimizer ADAM will not be used at this point, only RNN algorithm will be trained as the main classifier for this phase.

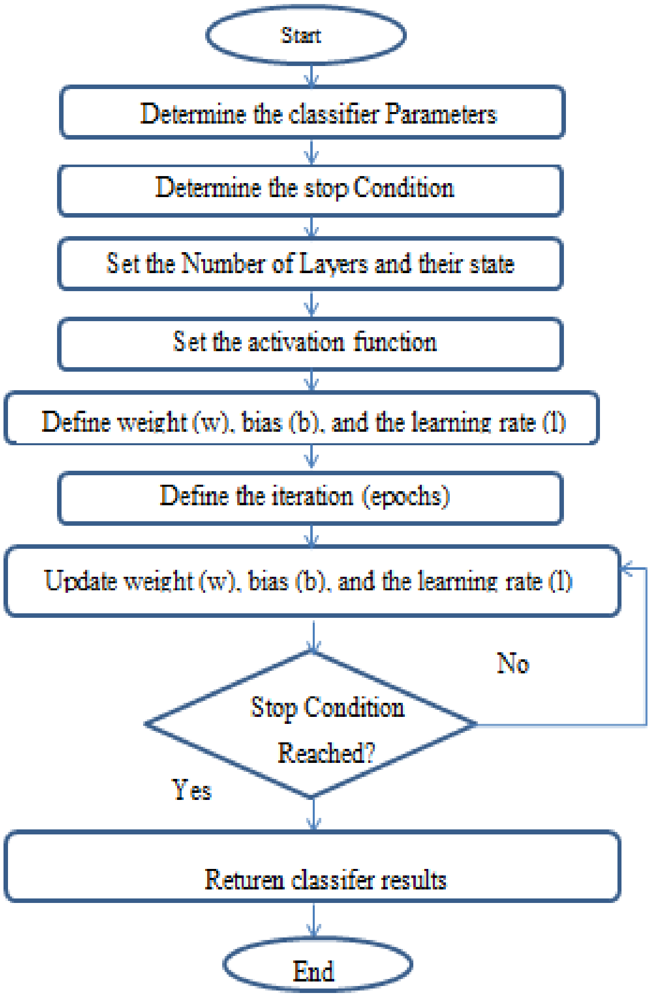

Start
Determine the classifier Parameters
Determine the stop Condition
Set the Number of Layers and their state
Set the activation function
Define weight (w), bias (b), and the learning rate (l)
Define the iteration (epochs)
Update weight (w), bias (b), and the learning rate (l)
No
Stop Condition Reached?
Yes
Returen classifer results
End

Figure 1.6 Recurrent Neural Network Training

The parameters that is utilized to set the RNN algorithm are:X_i: set of inputs in our case it is the students information, Y: output value which is the prediction of final course continuity, W_i: set of initial weights, b: the bias value, h_i: the states of the layers, *tan*: the activation function, l: the learning rate, t: number of iteration. The functions and all the formulas needed to calculate the input, output, and activations are in the following pseudo code:

```
Start
Initialize the parameters
Step1: determine the required values
$X$ //*Input value (student scores)
$Y$ //*Output value (prediction o final rate)
$w$ //*Weight
$b$ //*Bias
h1, h2, …,hn//* State of the layers
Tan//* Activation Function
L //* Learning rate
Epochs//* Number of Iteration
Step2: calculating current state
$h_t = f\ (h_{t-1}, x_t)$ Where:
ht //* current state
ht-1//* previous state
xt //* input state
$f$ //* Calculation function
Step3:
Applying Activation function (tanh)
$h_t\ =\ tanh\ (W_{hh} h_{t-1} +\ W_{xh} X_t)\ where$:
$W_{hh}$ //*weight at recurrent neuron
$W_{xh}$ //* weight at input neuron
$X_t$// *input state
Step4: Repeat the steps until the stop condition reached, then Classify the class label with its values
End
```

After training the classifier using only RNN.The main concept of deep learning optimization is to use gradient descent which is a well-known optimization method used in various Artificial Neural Networks (ANNs) algorithms. To digest the idea of the gradient descent, it is easier to consider some arbitrary point at the top of K mountain and this point must go down to reach the bottom of this mountain, the best way to descend is to choose a surface which tend to be more descended, in the same situation here, the best learning rate is the point and the algorithm space is the mountain surface and the great descent is the way to be used to reach the best possible prediction value [11][12][13]. Basically deep learning algorithms has utilized different gradient descent algorithms. In this paper ADAM optimizer is used with for Recurrent Neural Networks models. Figure (1.7) shows the gradient descent with ADAM optimizer:

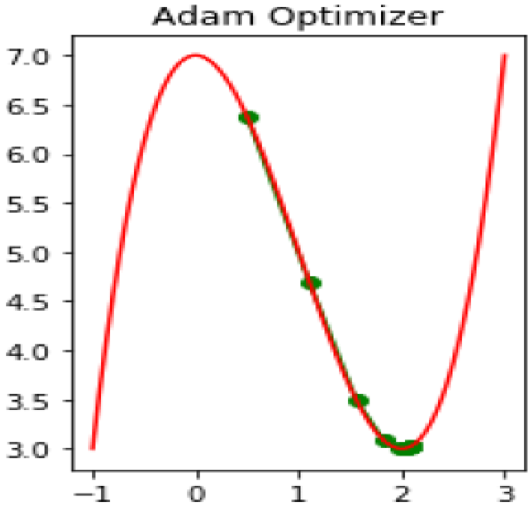


Figure 1.7 ADAM optimizer in the gradient descent [14]

ADAM Optimizer considers an efficient way utilized to improve the learning rate of ANN algorithms that computes the learning rate for each parameter in set of parameters [15]. ADAM considers the exponentially decaying average of gradients (like momentum) and squared gradients and they are termed as first moment and the second moment respectively and hence the name Adaptive Moment (ADAM) [16]. It utilizes the squared gradients to estimate the learning rate of the first and second moments of a gradient to change the rate for each weight of the network. Figure (1.8) ADAM optimizer steps.

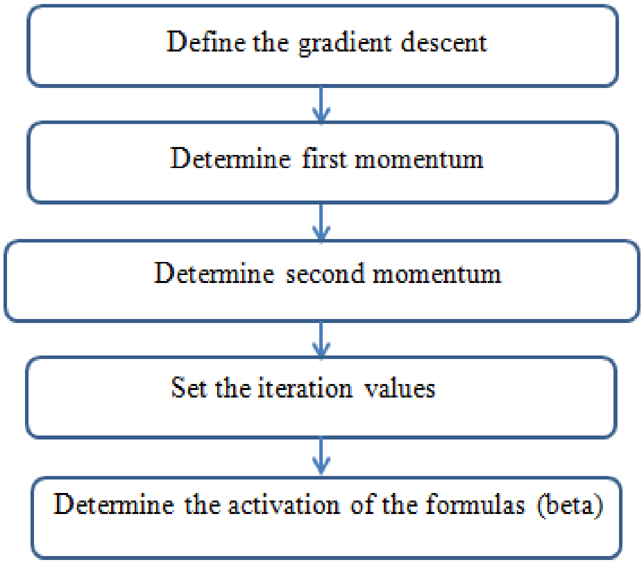

Define the gradient descent
Determine first momentum
Determine second momentum
Set the iteration values
Determine the activation of the formulas (beta)

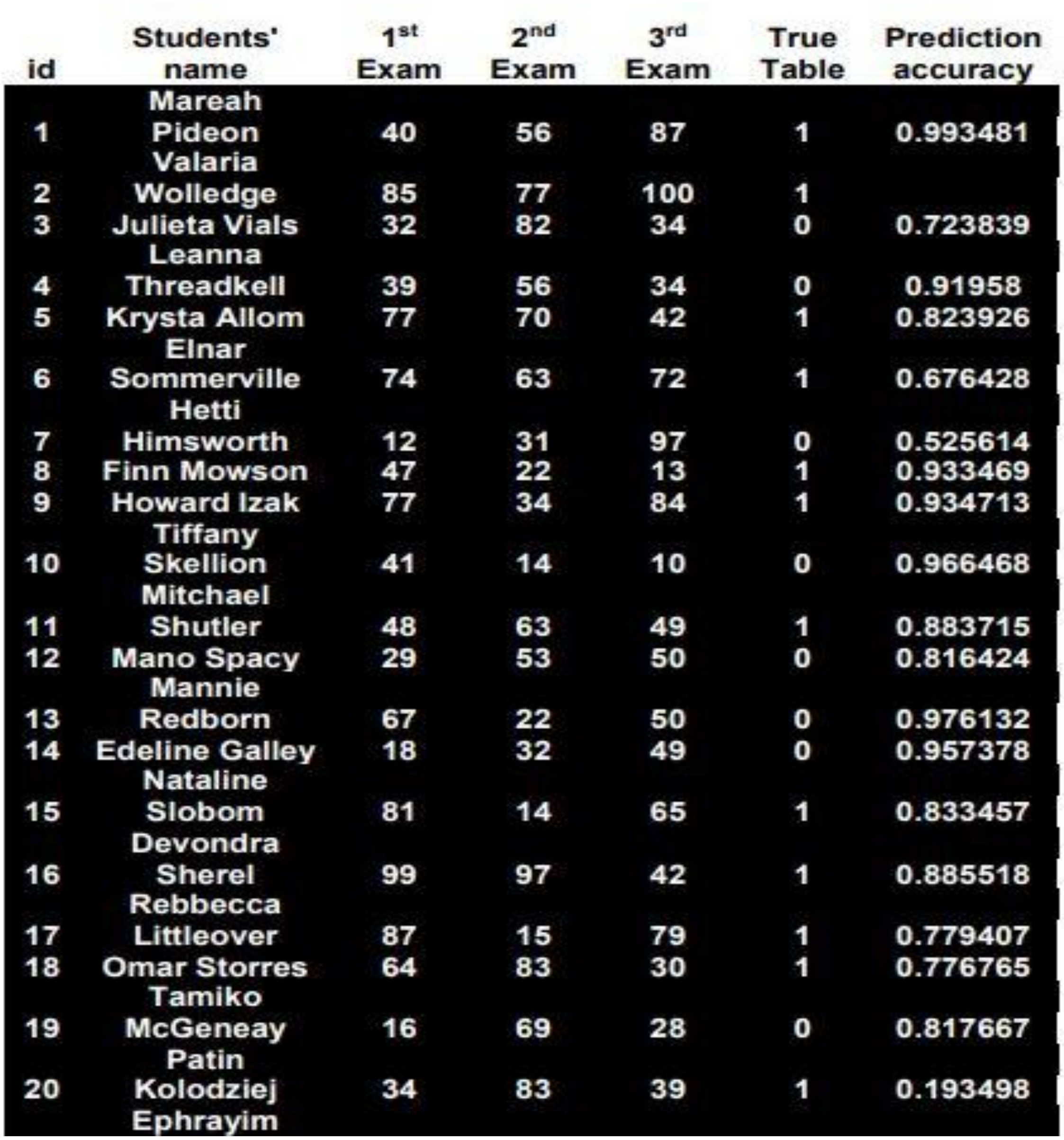
Figure 1.8 ADAM Optimizer Steps

The following parameters are required to construct the optimizer: $G\ (x, y)$: Gradient descent values, M: First moment, V: Second moment, t: number of iteration, β:activation of the formula. The functions and all the formulas needed to calculate the ADAM are in the following pseudo code:

```
Start
Step1: Determine $1st$ and $2nd$ moment$inthidcase(M, V)$
Step2: Start for loop in each iteration$t$ to calculate the number of the times that the network needs to be learned.
Step3: Compute the gradient equation:
    $g = \nabla \sum_{i=1}^{n} (x_i, y_i)$
Step4: Update bias from the first moment equation
    $m = \beta_1 * m + (1 - \beta_1) * g$
Step5: Update bias from the second moment equation:
    $v = \beta_2 * v + (1 - \beta_2) * g^2$
Step6: Calculate the derivative of the first moment equations
    $m^{'} = \frac{m}{(1 - \beta_1, t)}$
Step7: Calculate the derivative of the second moment equations:
    $v^{'} = \frac{v}{(1 - \beta_2, t)}$
Step8: Calculate the weight with respect to new derivative
    $w^{'} = w - l * m^{'} / v^{'}$
Step9: ApplyUpdate
    $w_{t+1} = w_t - \nabla w_{t+1}$
End for
Classify the class label after hiding its values
End
```

Experiment the Results

The result of the proposed algorithm will be visualized to the teachers of the system at a special control panel after the students have finished the course and each student will have a special (user learning profile), the final result will return the Student's information, students E-mail, and the prediction of continuity, "0" means that the student will drop the course, "1" means the student will complete the course. The page is automatically generated. Figure (1.9) shows the prediction window.



id	Students' name	1st Exam	2nd Exam	3rd Exam	True Table	Prediction accuracy
1	Mareah Pideon	40	56	87	1	0.993481
2	Valaria Wolledge	85	77	100	1	
3	Julieta Vials	32	82	34	0	0.723839
4	Leanna Threadkell	39	56	34	0	0.91958
5	Krysta Allom	77	70	42	1	0.823926
6	Elnar Sommerville	74	63	72	1	0.676428
7	Hetti Himsworth	12	31	97	0	0.525614
8	Finn Mowson	47	22	13	1	0.933469
9	Howard Izak	77	34	84	1	0.934713
10	Tiffany Skellion	41	14	10	0	0.966468
11	Mitchael Shutler	48	63	49	1	0.883715
12	Mano Spacy	29	53	50	0	0.816424
13	Mannie Redborn	67	22	50	0	0.976132
14	Edeline Galley	18	32	49	0	0.957378
15	Nataline Slobom	81	14	65	1	0.833457
16	Devondra Sherel	99	97	42	1	0.885518
17	Rebbecca Littleover	87	15	79	1	0.779407
18	Omar Storres	64	83	30	1	0.776765
19	Tamiko McGeneay	16	69	28	0	0.817667
20	Patin Kolodziej	34	83	39	1	0.193498
	Ephrayim					

Figure 1.9 Prediction Results

In the above figure1.9 records of random students have been extracted, the system has five columns (id, student name, and three columns the indicated the scores of all exams in a specific course), theses information are extracted from the students own profile in the database. The system will set a prediction and add two new columns which are the true table that gives either "0" or "1" as explained, the main focus is on the true tables results where the student's with "0" result is more likely to drop the course, his/her dataset, grades, behavior inside the system must be reviewed to target the reasons behind the failing results, and the prediction accuracy in which RNN-ADAM algorithm have been launched to make a prediction upon the students' scores.

Calculating the Accuracy

After training and testing the model, now the accuracy will be extracted from the model (RNN-ADAM) the accuracy will be calculated as the following: how many features was true and the model has predicted them as false, and the range must be as low as the possible, where the number of the iteration (epochs) was 100 to 200 try, and the learning rate started at 0.01. Table (1.3) and (1.4) shows the results of the accuracy of the given algorithm.

Table 1.3 The results of testing phase

Training Phase	Number of instance	Number of rows	Correctly Classified Scores	Incorrectly Classified Scores	Run Time
RNN	80%	920	719	201	5.06 sec.
RNN-Adam	80%	920	908	12	3.01 sec.

Table 1.4 The results of training phase

Testing Phase	Number of instance	Number of rows	Correctly Classified Scores	Incorrectly Classified Scores	Run Time
RNN-Adam	20%	80	78	2	1.001 sec.

A comparison between the proposed models with other supervised learning model is made to observe the accuracy and the goodness of the used model based on the same dataset from the proposed system. System, the software that is used to compare the accuracy is called (WEKA), which is tried and tested open source machine learning software that can be accessed through a graphical user interface, it contains a plethora of built-in tools for standard machine learning tasks [16]. Our proposed algorithm showed a highest accuracy among other classifiers with 99.1% accuracy rate, then Multi-layer perceptron algorithm (MLP) showed 97.99% accuracy, while decision tree indicates 64.78% accuracy rate, random forest on the other hand viewed 77.96% accuracy, finally naïve Bayesian classifier gives 89.34% accuracy rate. Table (1.5) shows the accuracy measure of algorithm performance prediction using WEKA environment.

Table 1.5 Accuracy Measure Results at testing phase only

No.	Model (supervised learning)	Training dataset	Testing dataset	Accuracy
1	Proposed Algorithm RNN-ADAM	80%	20%	99.1%
2	MLP	80%	20%	97.99%
3	Decision tree	80%	20%	64.78%
4	Random Forest	80%	20%	77.96%
5	Naïve Bayesians	80%	20%	89.34%

Conclusion

This research showed how RNN algorithm is optimized with ADAM optimizer using students dataset extracted from E-Learning system. We explained the main the parts of the system then we showed how the recommendation system of E-courses work, after that we display RNN algorithm and how it is optimized with using ADAM algorithms and viewed the pseudo code to embed them in the system programming part (back-end) and to show the results in the (front-end) to be visualized. Then we used WEKA software to compare our proposed algorithm with other machine learning algorithms of supervised part only, we used 1000 rows of the proposed dataset and we divide them to 80 for training and 20% for testing, our algorithm, showed highest accuracy results among four types of different algorithms. This paper introduces the following: (i): proposing intelligent online learning systems to create and extract students' dataset to be used later in classification and prediction. (ii) Implement Recurrent Neural Networks (RNN) as Deep learning classifier. (iii) Optimize the RNN with (ADAM) optimizer to minimize the error rate. (iv) Run the code and visualize the results. (v) Compare the proposed algorithms with other machine learning models.

References

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., & Torralba A. (2016). Learning deep features for discriminative localization. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 2921-2929.

https://www.Superdatascience.Com/Deeplearning/

Cong, P.H., & Robles-Kelly, A. (2008). A NURBS-based spectral reflectance descriptor with applications in computer vision and pattern recognition. *IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK*, 1-8.

Sultan, T., Nasr, M., & Saied, M. (2011). Recommender system role in e-Learning environment *(Doctoral dissertation, Faculty of Computers and Information, Helwan University)*.

Elman, Jeffrey, L.E. (1990). *Finding structure in time.* Cognitive science, *14*(2), 179–211.

Ruben, T., Ruben, V.R., Julian, F., & Javier, O.G. (2018). Exploring Recurrent Neural Networks for On-Line Handwritten Signature Biometrics. *IEEE Access*, *6*, 5128-5138.

Graves, A., & Jaitly, N. (2014).Towards end-to-end speech recognition with recurrent neural networks. *In International conference on machine learning*, 1764-1772.

Hochreiter, S., Bengio, Y., Frasconi, P., & Schmidhuber, J. (2001). Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. *In A Field Guide to Dynamical Recurrent Networks, S.C. Kremer and J.F. Kolen, Eds*.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computing*, *9*(8), 1735–1780.

https://www.researchgate.net/figure/The-architecture-of-a-a-traditional-Artificial-Neural-Network-ANN-and-b-a-Recurrent_fig1_325271784

Sebastian, R. An Overview of Gradient Descent Algorithms. *Cornell University Library*, Arxiv: 1609.04747[Cs. Lg]

Anirban, S., Aditya, C., Prantik, H., Balasubramanian V. Grad-Cam++: Generalized Gradient Based Visual Explanations For Deep Convolutional Networks. *Proceedings of IEEE Winter Conference on Applications of Computer Vision (Wacv'18)* 2018.

Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. *In Proceedings of the IEEE international conference on computer vision*, 618-626.

Deepak, B. (2018). Adam Optimization algorithms in Deep Learning.https://www.tech-quantum.com/adam-optimization-algorithms-in-deep-learning/

Zhang, Z. (2018). Improved adam optimizer for deep neural networks. *In IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*, 1-2.

https://www.cs.waikato.ac.nz/ml/weka/

Morze, N.V., Glazunova, O.G., & Grinchenko, B. (2013). What Should be E-Learning Course for Smart Education. *In ICTERI*, 411-423.

Zhu, Z.T., Yu, M.H., & Riezebos, P. (2016). A research framework of smart education. *Smart learning environments*, *3*(1).

Veeramanickam, M.R.M., & Radhika, N. (2014). A Smart E-Learning System for Social Networking. *International Journal of Electrical & Computer Engineering*, *4*(3), 447-455.

Simic, K., Despotovic-Zrakic, M., Đuric, I., Milic, A., & Bogdanovic, N. (2015).A model of smart environment for e-learning based on crowd sourcing. *RUO. Revijaza Univerzalno Odlicnost*, *4*(1), A1-A10.

Ahmad, F., Ismail, N.H., & Aziz, A.A. (2015).The prediction of students' academic performance using classification data mining techniques. *Applied Mathematical Sciences*, *9*(129), 6415-6426.

Pimpa, C. (2013). Predicting Student Academic Achievement by Using the Decision Tree and Neural Network Techniques. *International Multi Conference of Engineers and Computer Scientists*.