Article **Journal High School**
June 2023, Vol.7 No.17 17-30

Predicting academic performance grades of control student using a LSTM neural network

Predicción de las notas de rendimiento académico de un alumno de control mediante una red neuronal LSTM

TORRES-RAMÍREZ, Dulce Esperanza†*, JIMÉNEZ-GONZÁLEZ, Fernando C., HERRERA-OGAZ, José Alberto and MENDOZA-PÉREZ, Miguel Ángel

Universidad Tecnológica de Ciudad Juárez, México.

ID 1st Autor: *Dulce Esperanza, Torres-Ramírez* / **ORC ID:** 0000-0001-8409-2691, **Researcher ID:** IST-3145-2023, **CVU CONAHCYT ID:** 322126

ID 1st Co-author: *Fernando C., Jiménez-González* / **ORC ID:** 0000-0002-4577-3408, **CVU CONAHCYT ID:** 443945

ID 2nd Co-author: *José Alberto Herrera-Ogaz* / **ORC ID:** 0000-0002-4282-0002

ID 3rd Co-author: *Miguel Ángel Mendoza-Pérez* / **ORC ID:** 0009-0008-3666-0293

DOI: 10.35429/JHS.2022.15.6.17.30 Received January 15, 2023; Accepted June 30, 2023

Abstract

The evolution of artificial intelligent (AI) allows the development of new neural networks models associated with academic process such as: research, teaching-learning, and evaluation. Typically, in evaluation, the teacher verifies the correct use of the competency by the students. Deep Neural Networks has a significant impact to automate grading process for certain types of subjects. In skills learning process, especially in control and automation, the student must develop professional tasks. In this paper, a LSTM recurrent neural model is presented as an evaluation tool to classify the level of the skill in sufficient, prominent, or automated, using a temporal series associated with task performance. The model contains 5 layers, with SoftMax activation function and Adam optimizer. A non-recurrent Neural Network (NN) with 5 layers, RELU activation function and Adam optimizer, is developed to compare results. The results show an improvement in the LSTM model with 91.7% accuracy, 92.4% precision and 60% less requirement in the training and validation process. The results allow the proposed LSTM model to be a tool-assistant for the evaluation of control and automation skills.

LSTM, SoftMax, Adam, Recurrent, Activation, Neural, Competency, Accuracy

Resumen

La evolución constante en Inteligencia Artificial (IA) permite el desarrollo de nuevos modelos de redes neuronales en procesos académicos de investigación, enseñanza-aprendizaje, y evaluación. Típicamente, en la evaluación, el docente verifica el correcto uso de la competencia por los estudiantes. Las redes neuronales profundas presentan un impacto significativo para automatizar el proceso de calificación para ciertos tipos de materias. En el proceso de aprendizaje de competencias, especialmente en control y automatización, el alumno debe desarrollar tareas profesionales. El presente trabajo, propone utilizar un modelo de red neuronal recurrente LSTM para la evaluación de estudiantes, mediante una secuencia de tiempo asociada a la realización de tareas, con el fin de clasificar la competencia en suficiente, destacado y autónomo. El modelo contiene 5 capas, función de activación SoftMax y optimizador Adam. Se desarrolló una Red Neuronal (NN) no recurrente de 5 capas, función de activación RELU y optimizador Adam, para comparar resultados. Los resultados muestran una mejora en el modelo LSTM con 91,7% de precisión, un 92,4% de precisión, y 60% menos de exigencia en el proceso de entrenamiento y validación. Los resultados permiten que el modelo LSTM propuesto sea una herramienta-asistente para la evaluación de habilidades de control y automatización.

LSTM, recurrente, competencia, Recurrente, Activación, Neuronal, Competencia, Exactitud

Citation: TORRES-RAMÍREZ, Dulce Esperanza, JIMÉNEZ-GONZÁLEZ, Fernando C., HERRERA-OGAZ, José Alberto and MENDOZA-PÉREZ, Miguel Ángel. Predicti30ng academic performance grades of control student using a LSTM neural network. Journal High School. 2023. 7-17:17-30.

* Correspondence from the Author: (E-mail: dulce_torres@utcj.edu.mx)
† Researcher contributing as first author.

Introduction

Nowadays, cognitive technologies **(Kuzior, 2022)** and artificial intelligence have revolutionized several fields, including education. Student assessment is a necessary aspect of the education system, providing information such as performance, understanding, and progress. Traditional assessment based on written exams and subjective grading has limitations in objectivity, consistency, and the ability to capture the veracity level of a student's knowledge. Typically, student evaluations are slow, subjective, and biased. Developments in artificial intelligence (AI) and machine learning have a high impact on evolving the assessment processes for students, which has garnered significant interest in recent decades **(Xieling, 2022)**.

Technology in classrooms has helped deliver instruction based on student learning behavior **(Cheng, 2021)**. AI measures and understands learning processes, such as student training and performance evaluation, in a more precise, efficient, and personalized manner **(Arun, 2022)**. With artificial intelligence technologies, it is possible to design adaptive assessment systems that align with the student's level and pace of learning. It is feasible to analyze individual student data, providing objective assessments. An algorithm can study responses, tasks, and evaluations without biases or human subjectivity **(Sadiku, 2022)**, **(Bravo, 2023)** thereby enhancing accuracy and efficiency, reducing the time and effort required by teachers, and ensuring impartiality in student performance evaluation **(Ahmad, 2022)**. AI provides continuous monitoring to generate favorable personalized feedback based on students' strengths, weaknesses, and learning styles.

AI algorithms can process large datasets to identify patterns, trends, progress, and learning gaps in adapting teaching strategies. There is a high impact on finding statistical methods for diagnostic classification students **(Cui, 2016)**, Studying and applying artificial neural networks (RNN).

Deep Learning implementation in student assessment processes presents significant challenges such as data availability, interpretation of results, adaptability, and personalization of assessment.

K-Nearest Neighbors, Naïve Bayes, and Support Vector Machine are machine learning techniques for classifying results **(Yahya, 2013)**. The validation data analysis, using learning algorithms like Deep Learning, confirms that the evaluation results related to student performance (Hou, 2021) will be reliable when grading students because it can discover hidden relationships and patterns in the data.

The article presents an evaluation method that involves a time-based test sequence to obtain a dataset of 600 samples. Moreover, the model uses a 5-layer LSTM algorithm. The model is compared with an RNN, a non-recurrent neural network model, resulting in 91.7% accuracy and 92.4% precision.

Section 2 contains the background, followed by the state of the art in Section 3, the design of the classification model in Section 4, tests, and results in Section 5, and finally, the conclusions in Section 6, along with references in Section 7.

Research Works

Neural Networks (NN)

Methods like traditional artificial neural networks (ANN) have been developed to resemble the connectivity patterns of human brain neurons to perform tasks with improved performance through learning, training, and continuous improvement **(Kardan, 2013)**. ANNs can predict and compare a given system **(Martinez, 2023)**. A neuron allows different inputs to apply to an output. NNs can characterize the connections between neurons (topology) and the method for obtaining the optimal connection weights (training and learning algorithm) and activation function **(Rodríguez, 2021)**. The mathematical model function of the NN is presented in Equation 1.

$$\widetilde{\boldsymbol{Y}} = f\left(\widetilde{\boldsymbol{X}}, \widetilde{\boldsymbol{W}}\right) \tag{1}$$

ISSN-2523-0336

Where $\tilde{Y}$ y $\tilde{X}$ are the input and output vectors. $\widetilde{W}$are the weight parameters that represent las internal connections between NN.

ANNs are organized into layers: an input layer, which collects data into a set of features, one or more hidden layers, which process the input values, and an open-loop output layer. These networks are used for classification problems and pattern recognition in non-sequential data. The neurons value j and vectors $\tilde{Y}$are computed by the weighted sum of input elements from x and w. Where w is updated recursively (equation 2).

$$y_j = \theta\left(\sum_{i=1}^{N_i} w_{ij} x_i\right) \quad (2)$$

θ is the activation function (transfer function), N_i is the total number of connections of the j neuron and x_i is the output value of the past layer i neuron. The hyperbolic tangent function is used as the activation function (θ) to transfer the value of the weighted sum of inputs to the output layer.

Recurrent Neural Network (RNN)

Recurrent Neural Networks (RNNs) are a type of artificial neural network designed to process sequential data. Recurrent connections create a form of internal memory or state that allows capturing dependencies and patterns in sequence with the length of the input variable (**Karim, 2019)**, **(Lau, 2021)**.

Pascanu et al. **(Pascanu, 2013)** propose an RNN to maintain a hidden vector **h** as a state that is updated at time ***t***, with a step activation function (equation 3 and 4).

$$\boldsymbol{h_t} = tanh(\boldsymbol{W}\boldsymbol{h_{t-1}} + \boldsymbol{I}\boldsymbol{x_t}) \quad (3)$$

$$\boldsymbol{y}_t = softmax(\boldsymbol{W}\boldsymbol{h}_{t-1}) \quad (4)$$

Where *tanh* is a hyperbolic tangent function, x_t is the input vector at time ***t****.* is the recurrent weight matrix, and **I** is the projection matrix. In the prediction y_t, **h** is a hidden state, **W**, is the weight matrix, and SoftMax normalizes the model's output. The sigmoid function is denoted as **σ** in equation 3. Deep RNNs can be formed by stacking the output of one RNN as the input to another (equation 5).

$$\boldsymbol{h}_t^l = \sigma(\boldsymbol{W}\boldsymbol{h}_{t-1}^l + \boldsymbol{I}\boldsymbol{h}_t^{l-1}) \quad (5)$$

The basic architecture of an RNN can be used to address various problems. One issue that can arise with an RNN is that they can be affected by vanishing gradients, which impacts the training of the sequential network (network weights become very small - vanish and very large - explode, making the network unstable). LSTM (Long Short-Term Memory) or a GRU (Gated Recurrent Unit) could solve the problem.

Long Short-Term Memory (LSTM)

LSTMs are used when computational resources are limited. They are recurrent architectures. Unlike a traditional RNN, LSTM cells have a memory unit called the memory cell or long-term memory. The memory allows the network to retain relevant information throughout the sequence. Equations 6 to 11 show the basic structure of an LSTM cell **(Song, 2020)**.

$$i_t = \sigma(W_{ri}R_t + U_{hi}h_{t-1} + b_i) \quad (6)$$

$$f_t = \sigma(W_{rf}R_t + U_{hf}h_{t-1} + b_f) \quad (7)$$

$$\tilde{c}_t = tanh(W_{ri}R_t + U_{hc}h_{t-1} + b_c) \quad (8)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \quad (9)$$

$$o_t = \sigma(W_{ro}R_t + U_{ho}h_{t-1} + b_o \quad (10)$$

$$h_t = o_t \tanh(c_t) \quad (11)$$

Where $\boldsymbol{\sigma}$ is the sigmoid activation function, $\boldsymbol{R_t}$ is the input feature matrix over time ***t****,* $\boldsymbol{W_{ri}}, \boldsymbol{W_{rf}}, \boldsymbol{W_{rc}}$ ***y*** $\boldsymbol{W_{ro}}$, denote the matrix of weights between the input layer and the input gate, the forget gate, the memory cell and the output gate , respectively. $\boldsymbol{U_{hi}}, \boldsymbol{U_{hf}}, \boldsymbol{U_{hc}}$ ***y*** $\boldsymbol{U_{ho}}$, denote the matrix of weights from the hidden layer to the input gate, the oblivion gate, the memory cell and the output gate, respectively. $\boldsymbol{b_i}, \boldsymbol{b_f}, \boldsymbol{b_c}$ ***y*** $\boldsymbol{b_o}$, denote the offset value of the input gate, forget gate, memory cell, and output gate, respectively.

Loss function

The loss function is used to quantify the discrepancy between the outputs predicted by the network and the actual values of the training data. The loss function is defined in equation 12 **(Akbari, 2021), (Chen, 2022)**.

June 2023, Vol.7 No.17 17-30

$$L(\hat{y}-y) = -ylog(\hat{y}) - (1-y(\log(1-\hat{y})) \quad (12)$$

Where $\boldsymbol{y}$, denotes the actual classification of the sample and $\hat{\boldsymbol{y}}$, represents the result of pattern recognition.

SoftMax

SoftMax is an activation function used in the output layers of a neural network for multiclass classification problems. Equation 13 and 14 shows the SoftMax description.

$$p\left(\frac{z}{a}\right) = softmax(w^{(s)}h_k + b^{(s)} \quad (13)$$

$$z = argmax\left(p\left(\frac{z}{x}\right)\right) \quad (14)$$

Where **z** is the probability that the prediction is correct, **x** is the input, **h** is the hidden state.

Adam optimizer

Adam is a first-order optimization algorithm for stochastic functions based on adaptive estimates of lower-order moments used in the training of deep neural networks **(Kingma, 2020)**. Adam maintains a set of moving averages of gradients, which are calculated during the training process and used to auto-adjust the learning rates of parameters. The algorithm combines the advantages of the SGD with the momentum gradient algorithm and RMSprop, which uses a window to consider the most recent gradients **(Son, 2020)**. Compute for the moment gradient at time **t** (15), updating the biased estimate of the first moment (16), updating the biased estimate of the second raw moment (17), calculation of the first moment estimates with bias correction (18), Compute of the second moment to estimate the bias correction (19), parameter update (20):

$$g_t = \nabla_\theta(\theta_{t-1}) \quad (15)$$

$$m_t = \beta_1 m_{t-1} + (1-\beta_1)g_t \quad (16)$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2)g_t^2 \quad (17)$$

$$\widehat{m_t} = \frac{m_t}{1-\beta_2^t} \quad (18)$$

$$\widehat{v_t} = \frac{v_t}{1-\beta_2^t} \quad (19)$$

$$\theta_t = \frac{\theta_{t-1} - \alpha * \widehat{mt}}{\sqrt{\widehat{v_t + \varepsilon}}} \quad (20)$$

Related work

Luo et al. **(Luo, 2022)** analyze a method for piano teaching implementation using a neural network (NN) model, which evaluates piano performance and simulates teachers to guide students in their exercises.

With the impact of the COVID-19 pandemic, teaching methods evolved to be remote, offering online classes to mitigate learning issues. To ensure that students participate effectively, Bhardwaj et al. **(Bhardwaj, 2021)** introduce an innovative algorithm based on Deep Learning, using Convolutional Neural Networks (CNN) to monitor students' emotions in real-time, including feelings like anger, disgust, fear, happiness, sadness, and surprise. In addition, the work includes algorithms like MES (Mean Engagement Score) for facial landmark detection to gauge feelings. Both methods enhance and innovate digital learning.

Nuha Alruwais et al. **(Alruwais, 2023)** conducted a study using RNNs to predict student interactions in the virtual classroom (e-learning), allowing the prediction of a student's engagement. A classification of machine learning (ML) algorithms is used for machine learning and student evaluation through cross-validation and different metrics to optimize ML models. Parameters such as accuracy, precision, recall, and AUC scores (Area Under the Curve) are evaluated, as well as the ROC curve (Receiver Operating Characteristic curve), used to display the classification model's performance, showing the true positive and false positive rates. The model predicts that a student has a high level of engagement, achieving an accuracy of 94.64%.

Evaluating students' cognitive abilities in academic contexts can provide information to identify and assess their cognitive profile and create personalized teaching approaches. Orsoni et al. **(Orsoni, 2023)** compare two clustering techniques for students based on their cognitive abilities. The first compared method was SOM (Kohonen's Self-Organizing Map), and the k-means method, while the second method was AdaBoost and ANN (Artificial Neural Network).

ISSN-2523-0336

The work demonstrates that the first method is better to solve the problem, but the ANN method provides better classification. They suggest hybrid techniques to improve the reliability of clustering and the effectiveness of the results.

Alsabhan et al. **(Alsabhan, 2023)** propose a method to identify potential cheating incidents in exams using machine learning approaches. LSTM (Long Short-Term Memory) for modeling sequential data, which retains information from the previous input, and the Adam optimizer to find an optimal set of parameters that minimize the loss function, enabling the neural network to make accurate predictions. The implemented model achieves an accuracy of 90% in training and 92% in system validation. The implemented method creates academic performance prediction models and identifies vulnerable students with problematic behaviors.

Author	DataSet/Application	Results	Precision
Al-Azazi, et al. (2023)	A model ANN-LSTM is proposed to predict student performance using artificial neurons, with a dataset of 32,593.	The results show that the ANN-LSTM model achieved the best results among the RNN and GRU models.	70% accuracy, a value above 53%, and 57% for the other models.
Jha, et al. (2022)	The work proposed SVM, LSTM, and BILSTM models to address traffic and tracking of complaints on the student portal through the implementation of Foul/Hate Detection using Machine Learning and Deep Learning technologies, with a dataset of 11,325.	The results show that the LSTM model outperformed the other SVM and BILSTM models.	84% Precision
Tao, et al. (2022)	The work proposed a deep memory network PI-DMN based on ABSA aspects to assess the mental health of students through LSTM using a dataset of 52,106.	The results show that the LSTM outperformed with values above other models.	Precision between 80 and 85%
Rahman, et al. (2021)	The work proposes a language model to evaluate and repair source code using a BiLSTM with a dataset of 2,482.	The BiLSTM model outperforms the unidirectional LSTM and RNN models.	97% precision
Minn, et al. (2020)	The work proposes a BKT-LSTM model that predicts student performance in temporal ability assessment through BKT evaluation, using a dataset of 58.	The performance of BKT-LSTM is significantly better than that of the more advanced models in the tested datasets such as DKT and DKVMM.	85% accuracy, with a 10% improvement compared to the other models.

Table 1 Related work

Evaluating whether students' skills and competencies meet current industry requirements is a topic that has evolved in companies that express an interest in hiring employees certified in the domain of activities related to their profession.

The literature review presents some areas of opportunity to develop competency assessment systems, such as using recurrent neural network models for grading students' performance in specific topic-related tasks. The rapid development of new sequential processes can demonstrate certification of competencies over time. The present work exploits these opportunity areas in the design of a new LSTM model for student evaluation.

Classification model design

The evaluation model for students provides an interactive, intelligent, and automatic design to assist professors in control competences evaluation. The method architecture presents three main blocks: **1) Test sequence development. 2) Creation of the time-series dataset. 3) Development of the LSTM deep learning model for classifying students as sufficient, outstanding, or autonomous.** The Figure 1 shows the overall architecture.

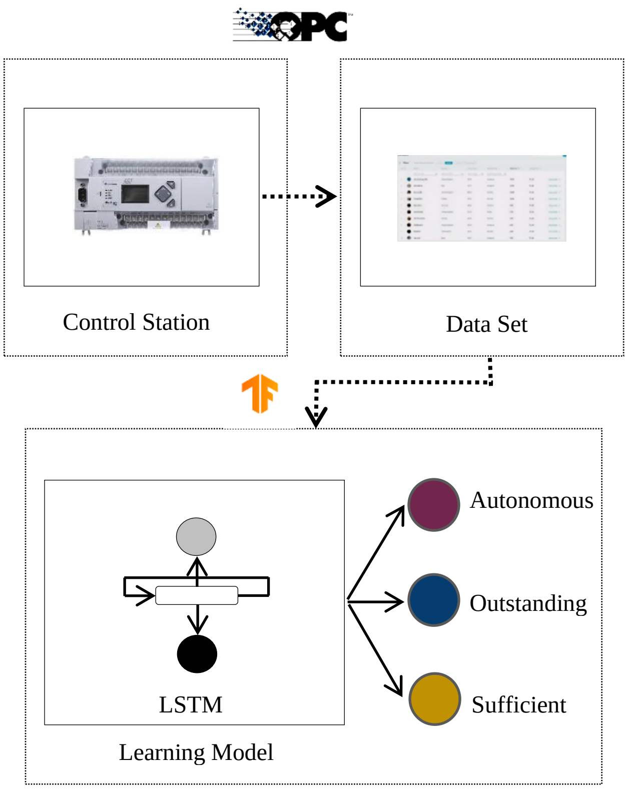


Figure 1 Overall system architecture

The test sequence (1) is a series of 5 main tasks in the control field and automation, which allows for obtaining a time sequence of the student's performance. In other words, the student develops progressive activities to accumulate time until completing the entire series. The tasks are related to PLC configuration, PLC programming, and troubleshooting in the automatic control system. The Table 2. shows the test configuration for data acquisition.

No.	Task	Evaluation	Equipment	Time Min
0	Sensor and actuators configuration	Equipment Installation and functional tests	Pneumatic valves and actuators, PNP and NPN sensors	5
1	Analog Sensor Configuration	Equipment Installation and functional tests	Voltage and Current Sensors	5
2	Troubleshooting	Quantity of fixed errors	PLC y CPU	10
3	PLC Programming	Sequential Programming process	PLC y CPU	20
4	Instrumentation Variables	Digital and Analog Monitor tags	PLC y CPU	20

Table 2 Tasks description

The completed tasks are monitored by time within the PLC program, which records a timestamp each time the student reaches the goals in the activities. In the first activity, the student has to configure two NPN inductive sensors and 2 PNP inductive sensors, and two bistable solenoid valves 5/3 with two pneumatic piston-type actuators. The student finishes the activity when the inputs and outputs signal proper function in the PLC program. The Figure 2 shows the completed configuration diagram of Task 1.

Figure 2 Student task 1

In the second activity, the student must connect an analog position signal and configure the signal acquisition filter. The task concludes when the student correctly scales the signal. The third activity is based on the student finding a wiring fault in the proposed configuration, presented by the evaluator. The task concludes when the student correctly identifies the fault or when the maximum time elapses. In the fourth activity, the student must program a pneumatic sequence equation using one of the step-by-step programming methods or a sequencer. The task concludes when the automatic mode of the sequence is successfully achieved. The Figure presents an example of the pneumatic sequence task.

In the final activity, the student performs an instrumentation of the analog position variable by implementing a Kalman filter and defined thresholds. The task concludes when the student programs the position levels: near, intermediate, and far. Figure 3 shows a student using a panel view Allen Bradley to develop the level instrumentation.

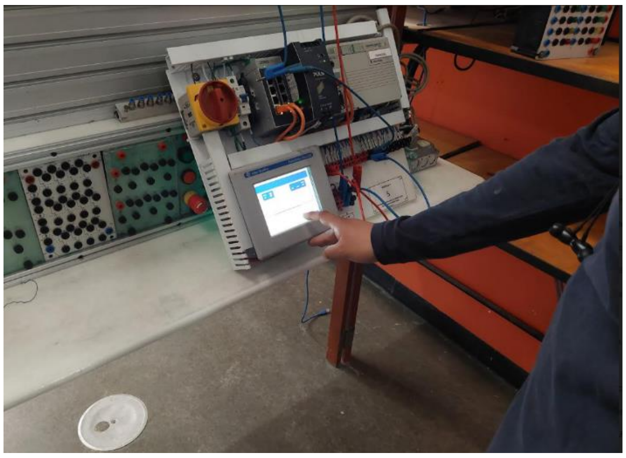


Figure 3 Student task 4

Typically, in deep learning LSTM, the creation of datasets for classification is of dynamic nature **(Ameur et al, 2020)**. The test sequence captures a time series of performance, describing the competency acquired by the student to solve control tasks. The Figure 4 shows three performance series of students categorized as sufficient, outstanding, and autonomous over time.

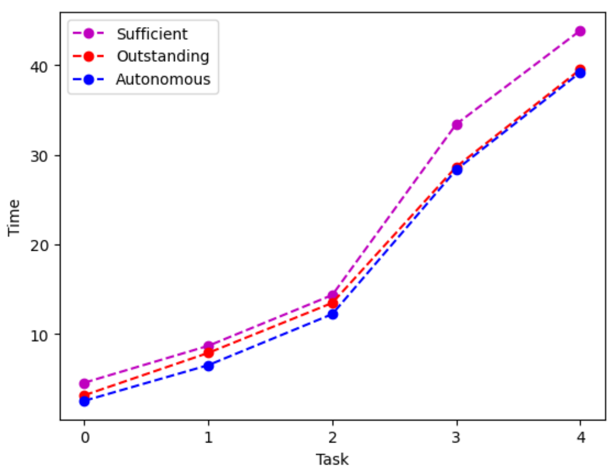
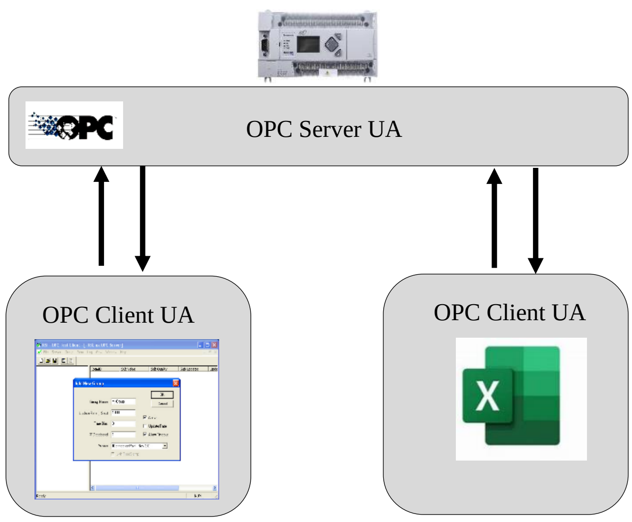


Figure 4 Performance series task

The timestamps and performance rates are sensitive to time-based information and exhibit the trend of the student's skills. The evaluation sequences describe past, present, and future features that capture long-term deendencies of time series data **(Nguyen et al, 2020)**.

For example, a sufficient-level class requires more time to implement control tasks versus the outstanding and autonomous levels. Additionally, the outstanding-level and autonomous-level classes exhibit similar performance indices. However, there is a trend in the time window markers to classify each area.

Dataset creation (2) transfers the timestamps into a database format through an OPC server (Object Linking Embedded for Process Control Server) communication provided by RSLinx OPC Server. An OPC server is object-model-based distributed software that enables direct connection with the PLC variables with data handling interfaces, reducing complexity in data capture and management **(Ahmad et al 2020).** The Figure 5 shows the communication scheme between PLC and the data interface.



The distribution of test sequence data was divided into 200 samples for each category: sufficient, outstanding, and autonomous, to obtain an initial dataset of 600 test sequences stored in CSV format. The deep-learning model uses the Dataset for the training and validation stages. Each feature corresponds to the completed time of the task. The table shows the design of the randomized dataset.

Record	Inputs					Outputs
1	T0	T1	T2	T3	T4	Sufficient
..	T0	T1	T2	T3	T4	Outstanding
600	T0	T1	T3	T3	T4	Autonomous

Table 3 Dataset structure

The deep learning LSTM model (M1) (3) assesses the student based on performance features over time in solving tasks. Typically, multi-label classification problems use recurrent sigmoid activation **(Lipton et al, 2015)**, and LSTM models have demonstrated high accuracy in multivariable classification **(Soufiane et al, 2021), (Xiao et al, 2020)**. The design of the model M1 presents a sigmoid recurrent activation function and five layers. The input layer presents a hidden layer with 64 LSTM neurons (tanh activation). In the middle, hidden layers use 64 LSTM neurons (SoftMax activation) and dropout layer at 20%. Finally, the output layer uses a SoftMax function. The Figure 6 illustrates the schematic of the proposed model in M1.

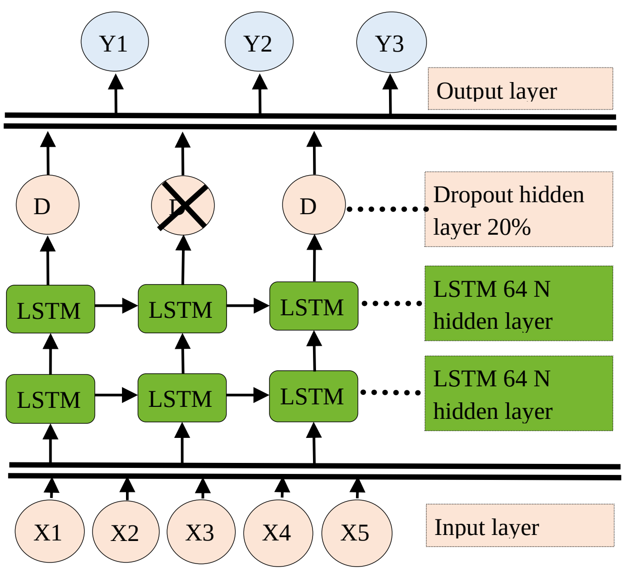


Figure 5 Model M1 structure

The behavior of the test sequences that students develop during the evaluation is from a non-linear nature, so a recurrent model of the LSTM type offers advantages in extracting series characteristics, as the mathematical model enables the LSTM cell to determine features of the previous state.

First, the model operates in the time state t**,** with the input set x**,** and the previous state h_{t-1}, to determine which information to retain or forget from the previous state f_t of the equation 21.

$$\mathrm{f_t} = \sigma\,(\mathrm{W_{ij}h_{t-1}} + \mathrm{W_{fx}x_t} + \mathrm{b_f}) \qquad (21)$$

Where σ is la logistic function, W is the weight matrix, y b describe the bias of the network **(Jeong et al, 2019)**. Model M1 uses the highly efficient rates of SoftMax function in multivariate classification, specifically in the final layers of a recurrent LSTM network **(Arbane et al, 2023)**. The Equation 22 shows the mathematical expression to define SoftMax cross-entropy loss function.

$$\hat{Y} = \frac{e^{xi}}{\sum_{i=1}^{k} e^{xi}} \qquad (22)$$

Where e^{xi} is the standard exponential function applied to the input value, k is number of classes, and $\sum_{i=1}^{k} e^{xi}$ ensures that the sum of the values results equals to 1 and that the values are within the range of (0 to 1). The Adam optimizer has an advantage as a multi-variable optimizer in LSTM classification problems.

The Adam algorithm uses adaptive gradients, which calculate the movement of gradient averages and squares to automatize the learning rate process **(Sakinah et al, 2019)**. A 20% dropout removes some neurons during training, which aids in the generalization of learning and helps prevent model overfitting. The training process uses 80 epochs and 80% of the samples for training, and 20% for validation.

An additional non-recurrent neural network model (M2) was developed to compare the results. M2 Model has 5 layers: the input layer, 1 hidden Dense layer with 64 neurons (Relu activation), 1 Dense layer with 64 neurons (SoftMax activation), 1 dropout layer at 20%, and the output layer. The Figure 7 shows the schematic of M2 model.

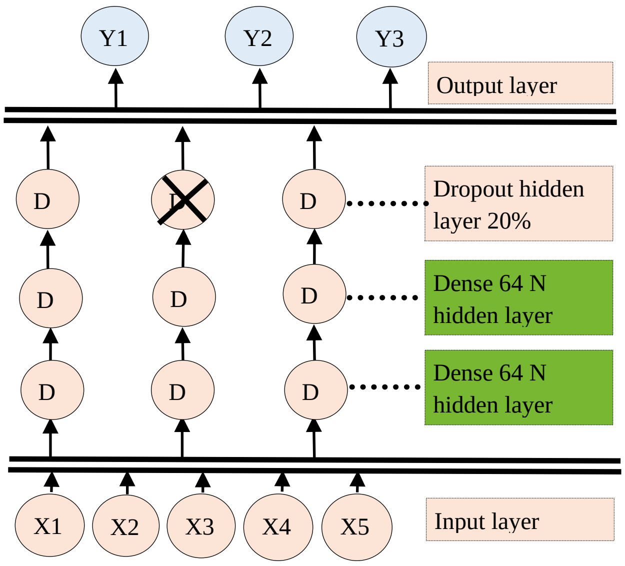


Figure 6 Model M2 structure

In M2 model, a SoftMax function was used in the last hidden layer, and a 20% dropout was applied for model response reliability. Additionally, it was optimized using the Adam algorithm in model compilation, and a training of 250 epochs was proposed for M2 model.

Model Evaluation

Several experiments evaluate the performance parameters of models M1 and M2. In the first level, the evaluation shows the overall performance percentages. The second level assesses the learning process in terms of time and graph characteristics. Finally, the confusion matrix shows the model error percentage.

Classification model performance

Typically, the parameters evaluated in LSTM models include accuracy and precision **(Althubiti et al, 2018).** Model M1 was compared with model M2 and two additional machine learning methods, Random Forest and KNN (K Nearest Neighbors). The table shows the results.

Method	Accuracy	Precision
LSTM – M1	91.70 %	92.40%
NN – M2	86.70%	88.30%
Random Forest	90.80%	90.10%
KNN	91.20%	90.54%

Table 4 Results model

The results show that model **M1** achieves a higher percent accuracy in determining whether the student is sufficient, outstanding, or autonomous in the test sequences. The results of **M1** versus **M2**, given the nature of the time-series input data, shows that **M1** achieves a higher accuracy and precision rate in learning while retaining the time-dependent characteristics of student performance.

Learning features

The evaluation of models **M1** and **M2** during the training and testing phases shows different behavior in the loss function and accuracy graphs. The figures 8 and 9 illustrate the behavior of the loss function for models **M1** and **M2**.

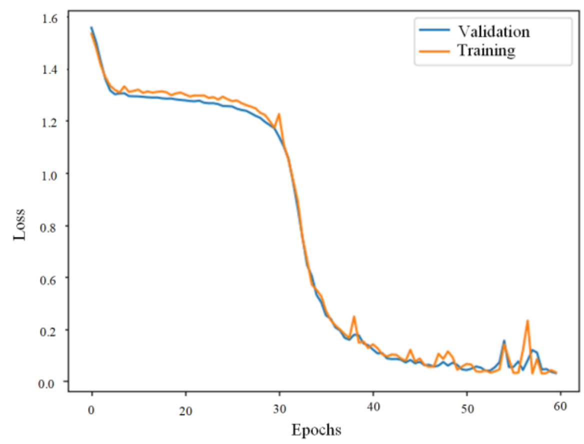


Figure 7 Model M1 Training and validation behavior

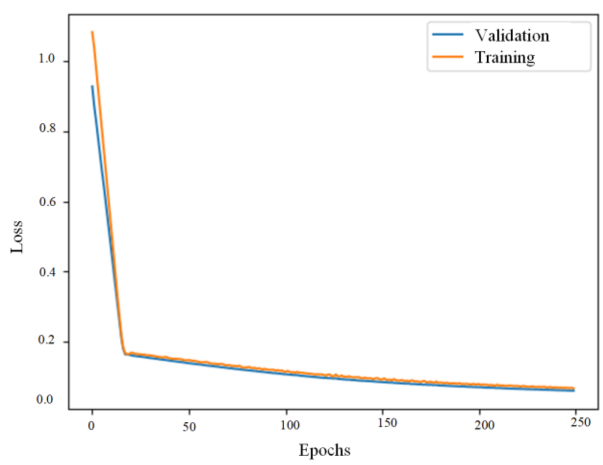


Figure 8 Model M2 training and validation behavior

ISSN-2523-0336

The figure shows the performance of model M1 in terms of the epochs required to achieve an accuracy of 91.70% and a loss function below 0.1 after epoch 80. Model M1 exhibits the typical noise behavior introduced by the 20% dropout. The model M2 in the figure shows a smooth behavior; however, it requires more than 225 epochs to reach a stable level in the loss function and an accuracy level of 86.70%.

The accuracy behavior for models M1 and M2 is shown under the same epoch conditions, where it can be observed that model M1 requires fewer epochs to achieve the accuracy percentage. The figures 10 and 11 illustrate the accuracy function behavior for models M1 and M2.

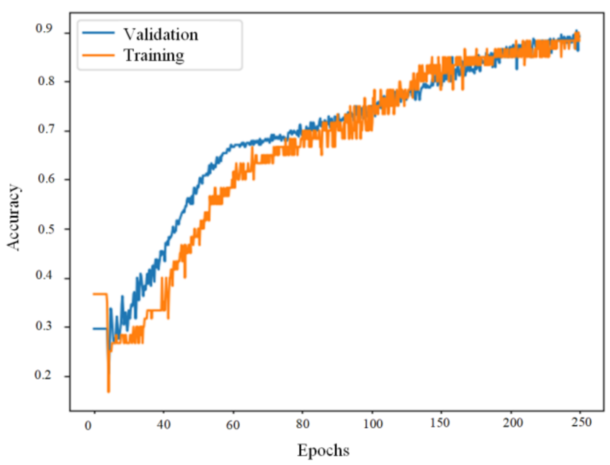


Figure 9 Model M1 accuracy behavior

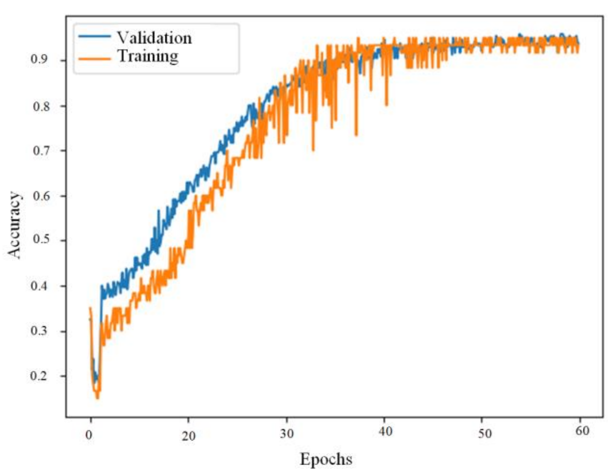


Figure 10 Model M2 accuracy behavior

The typical behavior of models associated with noise from a 20% dropout are shown in figures. The models get the accuracy percentages at different epochs. Model M1 gets its maximum accuracy at 60 epochs, while model M2 achieves it after 200 epochs.

Error evaluation

Usually, confusion matrices are used to determine the error of a deep learning model. **(Laghrissi et al, 2021).** The confusion matrix contains a summary of the predictions made by the classification model. The Figures 12 and 13 shows the confusion matrices for model M1 and M2.

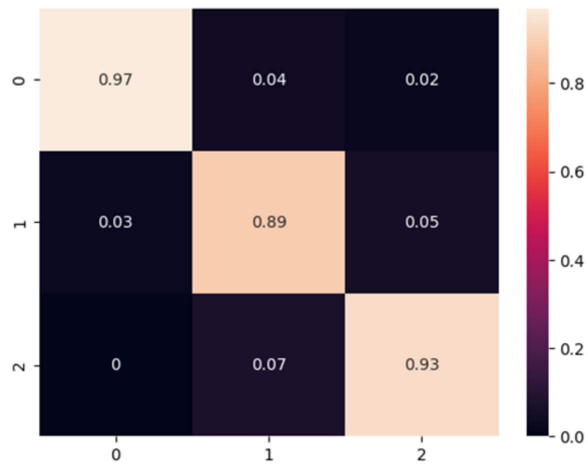


Figure 11 Model M1 confusion matrix

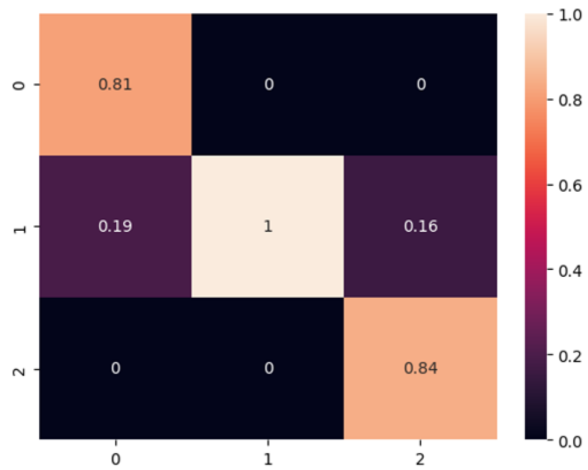


Figure 12 Model M2 confusion matrix

As seen in the Figure 12, model M1 has a higher error rate in identifying label 1 (outstanding), with a low probability of classifying it as 2 (autonomous). The highest accuracy rate is in label 0 (sufficient) and label 2 (autonomous). On the other hand, in the Figure 13, model M2 has the highest probability of making an error in label 0 (sufficient), and the highest accuracy rate is observed in label 1 (outstanding).

Conclusion

This article presents a classification model for evaluating students' competencies in control tasks to determine whether the student is sufficient, outstanding, or autonomous in solving activities related to automation and control. At times, assessment through exams can lead students to memorize information to pass an evaluation. Therefore, this method demands knowledge and requires the student to demonstrate professional abilities. Therefore, this method demands knowledge and requires the student to demonstrate professional abilities. The present paper creates an assessment of students based on a time series trying to make a relation between theoretical knowledge and practical abilities. In conclusion, a certified student can master the theory in the practice competencies by improving task times.

Model M1 presents the following improvements in the evaluation process:

- Model M1, when compared to model M2, exhibits a higher accuracy and precision rate in classifying the labels of 'sufficient,' 'outstanding,' or 'autonomous'.
- Model M1 has a better learning process in terms of the number of epochs required to achieve an effective classification percentage.
- Model M1 has lower error rates in confusion matrices, making it less likely to classify students incorrectly.
- Model M1 is a tool that allows complementing the theoretical evaluation of students by linking practice to an AI system to determine the category associated with their performance.

Finally, the project begins as an assessment alternative for a specific zone within the field of automation and control. However, this article demonstrates an opportunity approach for its implementation in various areas associated with engineering. Nowadays, academic institutions have a big challenge in developing evaluation methods that evaluate beyond theoretical concepts and focus on certifying students' practical skills.

The results demonstrate high levels of accuracy and precision, thus promoting further research in future work to continue enriching the dataset and establishing data acquisition systems associated with IoT. It is necessary to preserve the dynamic data structures and the generation of robust datasets to use them in competence certification processes.

References

Kuzior, A., & Kwilinski, A. (2022). Cognitive Technologies and Artificial Intelligence in Social Perception. Management Systems in Production Engineering, 30(2), 109-115. DOI 10.2478/mspe-2022-0014

Xieling, C. H. E. N., Di, Z. O. U., Haoran, X. I. E., Cheng, G., & Caixia, L. I. U. (2022). Two decades of artificial intelligence in education: Contributors, collaborations, research Topics, challenges, and future directions. Educational Technology and Society, 25(1), 28-47. https://scholars.ln.edu.hk/en/publications/two-decades-of-artificial-intelligence-in-education-contributors-

Cheng, Y., Cai, Y., Chen, H., Cai, Z., Wu, G., & Huang, J. (2021). A cognitive level evaluation method based on a deep neural network for online learning: from a bloom's taxonomy of cognition objectives perspective. Frontiers in psychology, 12, 661235. https://doi.org/10.3389/fpsyg.2021.661235

Arun Kumar, U., Mahendran, G., & Gobhinath, S. (2022). A review on artificial intelligence-based E-learning system. *Pervasive Computing and Social Networking: Proceedings of ICPCSN 2022*, 659-671. https://link.springer.com/chapter/10.1007/978-981-19-2840-6_50

Sadiku, M. N., Musa, S. M., & Chukwu, U. C. (2022). Artificial intelligence in education. iUniverse. https://books.google.es/books?hl=es&lr=&id=161cEAAAQBAJ&oi=fnd&pg=PT18&dq=sadiku+%2B+artificial+intelligence+in+education+iUniverse&ots=NqeS2o2BZ4&sig=WJXMAj4siI5pV67_aADZ768w74E#v=onepage&q=sadiku%20%2B%20artificial%20intelligence%20in%20education%20iUniverse&f=false

ISSN-2523-0336

Bravo, L. E. C., Bermúdez, G. M. T., & López, H. J. F. (2023). Analítica de datos y rendimiento académico. Editorial Universidad Distrital Francisco José de Caldas. https://books.google.es/books?hl=es&lr=&id=Ef7JEAAAQBAJ&oi=fnd&pg=PA1989&dq=Bravo,+L.+E.+C.,+Berm%C3%BAdez,+G.+M.+T.,+%26+L%C3%B3pez,+H.+J.+F.+(2023).Anal%C3%ADtica+de+datos+y+rendimiento+acad%C3%A9mico.+Editorial+Universidad+Distrital+Francisco+Jos%C3%A9+de+Caldas.&ots=Nh8owSSDSq&sig=zvB2D7qPDeh1gJi4Go0dF2ODKGs#v=onepage&q&f=false

Ahmad, S. F., Alam, M. M., Rahmat, M. K., Mubarik, M. S., & Hyder, S. I. (2022). Academic and administrative role of artificial intelligence in education. Sustainability, 14(3), 1101. https://doi.org/10.3390/su14031101

Cui, Y., Gierl, M., & Guo, Q. (2016). Statistical classification for cognitive diagnostic assessment: An artificial neural network approach. Educational Psychology, 36(6), 1065-1082. https://doi.org/10.1080/01443410.2015.1062078

Yahya, A. A., Osman, A., Taleb, A., & Alattab, A. A. (2013). Analyzing the cognitive level of classroom questions using machine learning techniques. Procedia-Social and Behavioral Sciences, 97, 587-595. https://doi.org/10.1016/j.sbspro.2013.10.277

Hou, Q. (2021). Research on the diagnosis of students' cognitive level based on deep learning. Security and Communication Networks, 2021, 1-6. https://doi.org/10.1155/2021/6106955

Kardan, A. A., Sadeghi, H., Ghidary, S. S., & Sani, M. R. F. (2013). Prediction of student course selection in online higher education institutes using neural network. Computers & Education, 65, 1-11. https://doi.org/10.1016/j.compedu.2013.01.015

MARTINEZ, Y. T. T. (2023). DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE EVALUACIÓN ADAPTATIVA BASADA EN MODELOS DE APRENDIZAJE AUTOMÁTICO PARA UNA COMPETENCIA DE ELECTRÓNICA ANALÓGICA (Doctoral dissertation, Universidad Tecnológica de Pereira). https://repositorio.utp.edu.co/server/api/core/bitstreams/a436336d-3d95-4bd5-b4f8-374207602cac/content

Ibrahim, Z., & Rusli, D. (2007, September). Predicting students' academic performance: comparing artificial neural network, decision tree and linear regression. In *21st Annual SAS Malaysia Forum, 5th September*. https://www.researchgate.net/profile/Daliela-Rusli/publication/228894873_Predicting_Students'_Academic_Performance_Comparing_Artificial_Neural_Network_Decision_Tree_and_Linear_Regression/links/0deec51bb04e76ed93000000/Predicting-Students-Academic-Performance-Comparing-Artificial-Neural-Network-Decision-Tree-and-Linear-Regression.pdf

Rodríguez-Hernández, C. F., Musso, M., Kyndt, E., & Cascallar, E. (2021). Artificial neural networks in academic performance prediction: Systematic implementation and predictor evaluation. Computers and Education: Artificial Intelligence, 2, 100018. https://doi.org/10.1016/j.caeai.2021.100018

Karim, F., Majumdar, S., & Darabi, H. (2019). Insights into LSTM fully convolutional networks for time series classification. IEEE Access, 7, 67718-67725. 10.1109/ACCESS.2019.2916828

Lau, E., Chai, K. K., Goteng, G. L., & Wijeratne, V. (2021). A Neural Network Modelling and Prediction of Students' Progression in Learning: A Hybrid Pedagogic Method. In CSEDU (1) (pp. 84-91). https://www.scitepress.org/Papers/2021/104056/104056.pdf

Pascanu, R., Gulcehre, C., Cho, K., & Bengio, Y. (2013). How to construct deep recurrent neural networks. arXiv preprint arXiv:1312.6026. https://doi.org/10.48550/arXiv.1312.6026

Song, R., Xiao, Z., Lin, J., & Liu, M. (2020). CIES: Cloud-based Intelligent Evaluation Service for video homework using CNN-LSTM network. Journal of Cloud Computing, 9, 1-9. https://link.springer.com/article/10.1186/s13677-020-0156-5

Akbari, A., Awais, M., Bashar, M., & Kittler, J. (2021, July). How does loss function affect generalization performance of deep learning? Application to human age estimation. In International Conference on Machine Learning (pp. 141-151). PMLR. https://proceedings.mlr.press/v139/akbari21a.html

Chen, X., Yu, R., Ullah, S., Wu, D., Li, Z., Li, Q., ... & Zhang, Y. (2022). A novel loss function of deep learning in wind speed forecasting. Energy, 238, 121808. https://doi.org/10.1016/j.energy.2021.121808

Kingma, D. P., & Ba, J. (2020). Adam: A Method for Stochastic Optimization, ArXiv14126980 Cs.(2017). arXiv preprint arXiv:1412.6980. https://doi.org/10.48550/arXiv.1412.6980

Luo, W., & Ning, B. (2022). Toward piano teaching evaluation based on neural network. Scientific Programming, 2022, 1-9. https://doi.org/10.1155/2022/6328768

Bhardwaj, P., Gupta, P. K., Panwar, H., Siddiqui, M. K., Morales-Menendez, R., & Bhaik, A. (2021). Application of deep learning on student engagement in e-learning environments. *Computers & Electrical Engineering*, *93*, 107277. https://doi.org/10.1016/j.compeleceng.2021.107277

Alruwais, N., & Zakariah, M. (2023). Student-Engagement Detection in Classroom Using Machine Learning Algorithm. *Electronics*, *12*(3), 731. https://doi.org/10.3390/electronics12030731

Orsoni, M., Giovagnoli, S., Garofalo, S., Magri, S., Benvenuti, M., Mazzoni, E., & Benassi, M. (2023). Preliminary evidence on machine learning approaches for clusterizing students' cognitive profile. *Heliyon*, *9*(3). https://www.cell.com/heliyon/pdf/S2405-8440(23)01713-9.pdf

Alsabhan, W. (2023). Student Cheating Detection in Higher Education by Implementing Machine Learning and LSTM Techniques. Sensors, 23(8), 4149.

Al-Azazi, F. A., & Ghurab, M. (2023). ANN-LSTM: A deep learning model for early student performance prediction in MOOC. Heliyon, 9(4). https://www.cell.com/heliyon/pdf/S2405-8440(23)02589-6.pdf

Tao, R. (2022). A Deep Learning-Based TE Method for MSs' Mental Health Analysis. Journal of Environmental and Public Health, 2022. https://doi.org/10.1155/2022/1420542

Rahman, M. M., Watanobe, Y., & Nakamura, K. (2021). A bidirectional LSTM language model for code evaluation and repair. Symmetry, 13(2), 247. https://doi.org/10.3390/sym13020247

Minn, S. (2020). BKT-LSTM: Efficient Student Modeling for knowledge tracing and student performance prediction. arXiv preprint arXiv:2012.12218. https://doi.org/10.48550/arXiv.2012.12218

Ameur, S., Khalifa, A. B., & Bouhlel, M. S. (2020). A novel hybrid bidirectional unidirectional LSTM network for dynamic hand gesture recognition with leap motion. Entertainment Computing, 35, 100373. https://doi.org/10.1016/j.entcom.2020.100373

Nguyen, H. P., Liu, J., & Zio, E. (2020). A long-term prediction approach based on long short-term memory neural networks with automatic parameter optimization by Tree-structured Parzen Estimator and applied to time-series data of NPP steam generators. Applied Soft Computing, 89, 106116. https://doi.org/10.1016/j.asoc.2020.106116

Ahmad, B., & Prajitno, P. (2020, April). Design of neural network and PLC-based water flow controller. In Journal of Physics: Conference Series (Vol. 1528, No. 1, p. 012065). IOP Publishing. DOI 10.1088/1742-6596/1528/1/012065

Lipton, Z. C., Kale, D. C., Elkan, C., & Wetzel, R. (2015). Learning to diagnose with LSTM recurrent neural networks. arXiv preprint arXiv:1511.03677. https://doi.org/10.48550/arXiv.1511.03677

Belagoune, S., Bali, N., Bakdi, A., Baadji, B., & Atif, K. (2021). Deep learning through LSTM classification and regression for transmission line fault detection, diagnosis and location in large-scale multi-machine power systems. Measurement, 177, 109330. https://doi.org/10.1016/j.measurement.2021.109330

Xiao, H., Sotelo, M. A., Ma, Y., Cao, B., Zhou, Y., Xu, Y., ... & Li, Z. (2020). An improved LSTM model for behavior recognition of intelligent vehicles. IEEE Access, 8, 101514-101527. 10.1109/ACCESS.2020.2996203

Jeong, J. H., Yu, B. W., Lee, D. H., & Lee, S. W. (2019). Classification of drowsiness levels based on a deep spatio-temporal convolutional bidirectional LSTM network using electroencephalography signals. Brain sciences, 9(12), 348. https://doi.org/10.3390/brainsci9120348

Arbane, M., Benlamri, R., Brik, Y., & Alahmar, A. D. (2023). Social media-based COVID-19 sentiment classification model using Bi-LSTM. Expert Systems with Applications, 212, 118710. https://doi.org/10.1016/j.eswa.2022.118710

Sakinah, N., Tahir, M., Badriyah, T., & Syarif, I. (2019, September). LSTM with adam optimization-powered high accuracy preeclampsia classification. In 2019 International Electronics Symposium (IES) (pp. 314-319). IEEE. 10.1109/ELECSYM.2019.8901536

Althubiti, S. A., Jones, E. M., & Roy, K. (2018, November). LSTM for anomaly-based network intrusion detection. In 2018 28th International telecommunication networks and applications conference (ITNAC) (pp. 1-3). IEEE. 10.1109/ATNAC.2018.8615300

Laghrissi, F., Douzi, S., Douzi, K., & Hssina, B. (2021). Intrusion detection systems using long short-term memory (LSTM). Journal of Big Data, 8(1), 65. https://link.springer.com/article/10.1186/s40537-021-00448-4