❏   199

An LSTM-based prediction model for gradient-descending optimization in virtual learning environments

Edi Ismanto[1], Noverta Effendi[2]
[1]Department of Informatics Education, Universitas Muhammadiyah Riau, Pekanbaru, Indonesia
[2]Department of Electronic Engineering Vocational Education, Universitas Muhammadiyah Riau, Pekanbaru, Indonesia

Article Info

ABSTRACT

A virtual learning environment (VLE) is an online learning platform that allows many students, even millions, to study according to their interests without being limited by space and time. Online learning environments have many benefits, but they also have some drawbacks, such as high dropout rates, low engagement, and students' self-regulated behavior. Evaluating and analyzing the students' data generated from online learning platforms can help instructors to understand and monitor students learning progress. In this study, we suggest a predictive model for assessing student success in online learning. We investigate the effect of hyperparameters on the prediction of student learning outcomes in VLEs by the long short-term memory (LSTM) model. A hyperparameter is a parameter that has an impact on prediction results. Two optimization algorithms, adaptive moment estimation (Adam) and Nesterov-accelerated adaptive moment estimation (Nadam), were used to modify the LSTM model's hyperparameters. Based on the findings of research done on the optimization of the LSTM model using the Adam and Nadam algorithm. The average accuracy of the LSTM model using Nadam optimization is 89%, with a maximum accuracy of 93%. The LSTM model with Nadam optimisation performs better than the model with Adam optimisation when predicting students in online learning.

Corresponding Author:

Edi Ismanto
Department of Informatics Education, Universitas Muhammadiyah Riau
Tuanku Tambusai Street, Pekanbaru, Riau, Indonesia
Email: edi.ismanto@umri.ac.id

1. INTRODUCTION

The design and development of virtual learning environments (VLE) and learning management systems (LMS), as well as other online learning platforms, have rapidly improved, eliminating not only the constraints of time and place but also lowering the cost and facilitating access to education. Evaluating and analyzing the students' data generated from online learning platforms can help instructors to understand and monitor students learning progress [1]. The earlier the students' performance is detected in the VLEs, the better it is for the instructor to persuade and warn students for keeping them on the right track. Therefore, it is challenging to create a predictive model that can precisely identify students' in-course learning behaviors by looking at behavior data.

In previous research, machine learning (ML) techniques have been extensively used in the development of predictive models to illustrate student learning behavior in VLE [2]–[6]. However, there are some limitations to the use of ML techniques in the development of predictive models. For example, there are limitations on the features selected and the ML models that are used [4]–[8]. The advancement of deep learning methodologies will allow prediction models to perform more accurately [9]–[13]. In an online

learning environment where a lot of data is produced every day. One of the best deep learning algorithms for handling issues with time series data is long short-term memory (LSTM) [14], [15].

The LSTM architecture is an enhanced recurrent neural network (RNN) that works well for long-term dependability in time series sequential data [16]. There are many hyperparameters available for LSTMs, including learning rates, the number of hidden units, input length, and batch sizes [17], [18]. Hyperparameters are parameters that are specifically defined to regulate how the model learns [19]. The model's output is significantly impacted by its hyperparameters [20]. Determining the right combination of models and hyperparameters is often a challenge. We want to investigate how hyperparameters affect LSTM. Hyperparameter selection and optimization frequently distinguish the outcomes from model accuracy. To fine-tune the hyperparameters, we used the adaptive moment estimation (Adam) and Nesterov-accelerated adaptive moment estimation (Nadam) optimization algorithms. Adam and Nadam, are the two most effective gradient descent optimization algorithms [21], [22].

The following is a review of a number of prior research studies that addressed the use of the LSTM algorithm to forecast online learning. The attention-based multi-layer (AML) LSTM, which combines clickstream data and student demographic data for thorough analysis, is suggested in this article [23] as a method for predicting students. The outcomes demonstrate that, from week 5 to week 25, the proposed model can increase accuracy for the four-class classification task by 0.52% to 0.85%. According to Alsabhan [24], the LSTM model performs better in terms of accuracy for the prediction of withdrawal in a VLE than both the logistic regression algorithm and neural networks. When detecting student cheating in higher education, LSTM with dropout layers, dense layers, and Adam optimizer [25] achieves 90% better accuracy than ML algorithms.

The LSTM model was improved in [26] research for predicting student performance using the Adam and root mean square propagation (RMSprop) algorithms. When compared to the RMSprop algorithm, the LSTM model with Adam's algorithm performs better. According to Bock and Weiß [27], Adam and Nadam outperformed adaptive learning rate delta (AdaDelta), adaptive gradient descent (AdaGrad), or RMPProp in terms of setting optimization parameters, as determined by the perceptual loss function and visual perception. In this study, the Adam and Nadam optimisation algorithm was used to test the LSTM algorithm model in order to determine the algorithm's optimal performance.

We suggest an LSTM algorithm model for predicting student learning outcomes in a VLE that has been improved with Adam and Nadam. The Adam and Nadam optimization algorithm is used to test each model. Then, the accuracy, recall, precision, and F1-score of each model are assessed in order to compare the outcomes. A stochastic gradient descent technique called Adam optimization is based on the adaptive estimation of first and second-order moments [28]. When dealing with complex problems involving a large number of variables or data, the method is incredibly effective. Adam is a fusion of the 'gradient descent with momentum algorithm' and the 'RMSprop' algorithm. The Adam and RMSprop methods have their respective strengths, and Adam optimizer builds on those strengths to produce a gradient descent that is more optimized.

The Nadam algorithm is a sophisticated gradient descent optimization method that raises the quality and convergence rate of neural networks [29]. Nadam alters the momentum component of Adam while maintaining an adaptive learning rate that is a pure amalgamation of Adam and Nesterov's accelerated gradient (NAG). Nadam converges faster and outperforms NAG and Adam on some types of data sets. Our research makes use of two hyperparameter optimization algorithms specifically Adam and Nadam. The parameters that we use to construct the LSTM model include learning rates, the number of hidden units, the length of input, batch sizes, and dropout. The following queries are what this essay aims to address: i) RQ1: how do hyperparameter optimization techniques LSTM as well as compare with each other? and ii) RQ2: which LSTM model is the most effective after assessing how well the optimization method worked?

2. METHOD

The research methodology used to compare the LSTM model with the gradient descent optimization method in order to forecast student performance in a VLE is shown in Figure 1. The initial stages of a research project are data gathering, data comprehension, and data processing [30]. Afterward, carry out the data preparation for the LSTM models. The data is separated into training, validation, and testing data.

2.1. Datasets

This study makes use of the open university VLE dataset. The open university learning analytics dataset (OULAD) dataset that was acquired includes the demographic information, login patterns, and assessment behavior of 32,593 students over the course of nine months. It consists of seven modules, or courses, each of which is taught at least twice a year at different times. The student performances are broken down into four groups, with 9% receiving distinctions, 38% receiving passes, 22% receiving failures, and

31% discontinuing their studies. The acquired raw data set consists of files that contain data on student demographics, clickstream data that shows how students interact with the online environment, assessments, quiz results, and module information.

Data about both students and courses are included in the dataset. The OULAD dataset contains data for seven courses. Data from the course BBB were the subject of our study. BBB is the course code. A total of 7,909 students are enrolled in the course's focus on social sciences, which has the highest enrollment of any other subject.
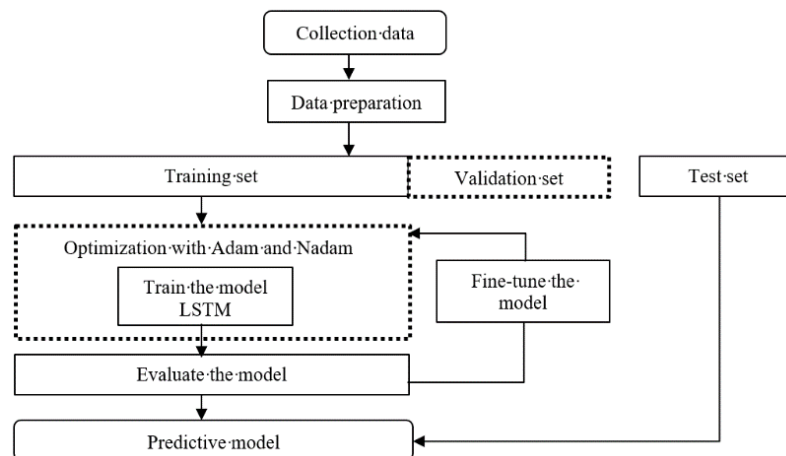

Figure 1. The phases of the research methodology used

2.2. Preparation of data

Data preparation is the collection, combination, cleaning, and transformation of raw data for ML projects in order to make accurate predictions. The dataset is preprocessed to select the BBB course features that will be used to train and test the model. The features that have been chosen and will be put to use are the module code, presentation code, student ID, clicks, assignment assessment, average assignment assessment, and final results.

There are 1,565,580 lines of BBB courses after preprocessing. There are two presentation codes or semester codes in the BBB course: "B" begins in February, while "J" begins in October. The presentation code used in the BBB course is shown in Figure 2. The data for the BBB course are divided: 60% for training, 20% for validation, and 20% for testing.
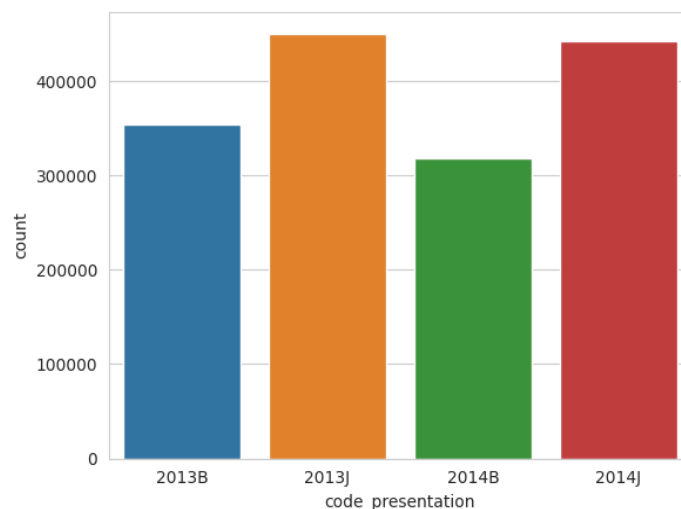

Figure 2. The BBB course's presentation code

2.3. The architecture of the designed long short-term memory model

LSTM is one of the RNN variants [14]. LSTM fills the gap left by RNN's inability to predict words based on previously learned information that has been stored for a long time. The fundamental distinction between LSTM and RNN architectures is that the hidden layer of the LSTM is a gated unit or gated cell [15]. It is made up of four layers that work together in some way to produce both the cell state and the output of that cell. Then, these two items are transferred to the following hidden layer. In contrast to RNNs, which only have one tanh layer, LSTMs have three logistic, sigmoid gates, and one tanh layer.

The LSTM model, which was created to predict a VLE, makes use of three input layers, two output layers with one node each and sigmoid activation functions, one hidden layer with sixteen nodes, and a hyperbolic tangent activation function to solve the non-linear function. Then, to enhance the LSTM model, a dropout layer with a 50% setting in each training step is included. The LSTM model was trained using batch size 32, with the back-propagation method. Figure 3 displays a design for the LSTM architecture.
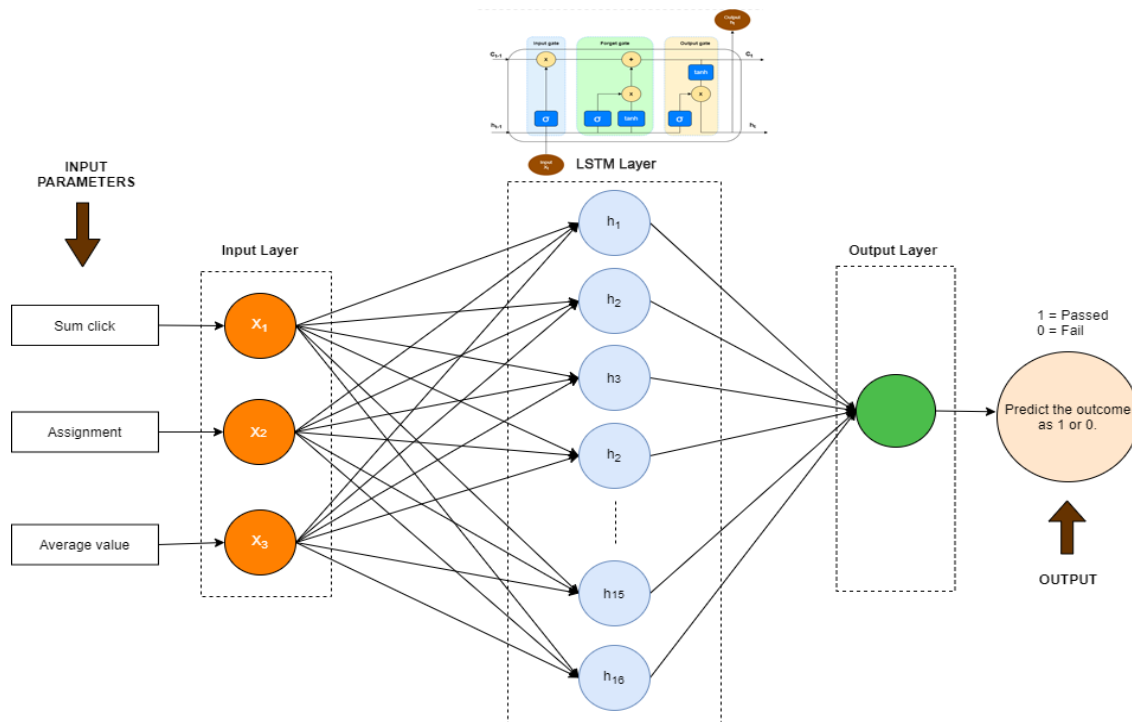

Figure 3. The architecture of the designed LSTM model

2.4. Optimization algorithms using gradient descent

The process of gradient descent is used to enhance neural network models [31]. The Adam and Nadam algorithms were used in this study as gradient-based optimization algorithms. The gradient descent algorithm requires that both the target function and its derivative function be optimized. The gradient descent optimization algorithm used in the study is as:

2.4.1. Adam optimizer

In contrast to the more traditional stochastic gradient descent approach, Adam is an optimization algorithm that can be used to iteratively update weights based on training data [21], [28]. Adam can be characterized as a stochastic gradient descent with momentum and the RMSprop model. Adam is a technique of the adaptive learning rate that lowers individual learning rates for various parameters.

2.4.2. Nadam optimizer

The NAG and Adam algorithms were combined to create the Nadam algorithm [22], [29]. Nadam performs a momentum update for the value of $\hat{m}_t$ [32]. The update rule has the following format:

$$\theta_{t+1} = \theta_t - \frac{n}{\sqrt{\hat{v}_t + \epsilon}} \left(\beta_1 \hat{m}_t + \frac{(1 - \beta_t) g_t}{1 - \beta_1^t} \right) \tag{1}$$

2.5. Performance evaluation of the model

The model's effectiveness was measured using a confusion matrix, accuracy (CA), precision, recall, and F1-score (F1) [33]. The confusion matrix depicts the present state of the dataset as well as the number of accurate and wrong model predictions [34]. The proportion of accurate predictions to all predictions is measured by accuracy, which is a crucial and intuitive metric. Precision measures the percentage of correctly predicted positive outcomes to the total number of correctly predicted positive outcomes. The recall is the ratio of true positive predictions compared to the total number of true positive data. A weighted comparison of the average precision and recall is called an F1-score (F1).

$$Accuracy = \frac{\sum True\ positives(TP) + \sum True\ negatives(TN)}{\sum Total\ population} \tag{2}$$

$$Recall = \frac{\sum True\ positives\ (TP)}{\sum True\ positives(TP) + \sum False\ negatives(FN)} \tag{3}$$

$$Precission = \frac{\sum True\ positives(TP)}{\sum False\ positives(FP) + \sum True\ positives(TP)} \tag{4}$$

$$F1\ Score = 2 * \frac{\sum Recall * \sum Precission}{\sum Recall + \sum Precission} \tag{5}$$

Formally, positives denote students who really fail, whereas negatives denote students who actually pass, while true denotes a valid prediction, and false denotes an incorrect forecast. A true positive value is TP, a true negative value is TN, a false negative value is FN, and a false positive value is FP. Table 1 illustrates the confusion matrix associated with various combinations of actual and predicted.

Table 1. The confusion matrix

Actual	Predicted	
	Positive (1)	Negative (0)
Positive (1)	TP	FP
Negative (0)	FN	TN

3. RESULTS AND DISCUSSION

In this study, Python programming was used for model training and testing. The architectural performance parameters were developed using 10 different combinations, and validation tests were carried out from 20% of the training dataset samples. The Adam and Nadam optimization algorithms were used to refine the models' hyperparameters. The following provides an explanation of the outcomes of the LSTM models' performance assessment.

3.1. Performance analysis of the long short-term memory model

We assess the accuracy, recall, precision, and F1-score of the Adam and Nadam algorithm-optimized LSTM model's performance. We compare the outcomes of our model performances to determine which is the best. The LSTM and Adam models were tested and trained in our first experiment. The second experiment went on to train and test the LSTM and Nadam models. Table 2 displays the measurement outcomes of the LSTM model with hyperparameter settings applied using Adam's algorithm.

Table 2. LSTM model results with Adam optimisation

	LSTM+Adam optimizer			
Decile	Accuracy	Recall	Precision	F1-score
0	0.75	0.76	0.57	0.65
1	0.60	0.60	0.67	0.62
2	0.63	0.64	0.67	0.65
3	0.71	0.72	0.71	0.72
4	0.78	0.78	0.79	0.78
5	0.82	0.82	0.82	0.82
6	0.87	0.88	0.88	0.87
7	0.88	0.88	0.88	0.88
8	0.89	0.90	0.90	0.89
9	0.91	0.92	0.92	0.92
10	0.92	0.92	0.92	0.92

The average accuracy value for the LSTM model using Adam's algorithm optimization is 87%, with the lowest accuracy value being obtained at 60%, and the highest accuracy value being obtained at 92%. The highest recall value is 92%, the lowest recall value is 60%, and the average recall value is 88%. Table 3 displays the measurement outcomes of the LSTM model with hyperparameter settings using Nadam's algorithm.

The average accuracy value for the LSTM model using the Nadam algorithm optimization is 89%, with the highest accuracy value obtained being 93%, and the lowest accuracy value obtained being 60%, according to experimental results. The average recall percentage is 89%, with the lowest recall percentage being 60% and the highest recall percentage being 93%. We visualize the accuracy results of the LSTM-Adam and LSTM-Nadam models and compare them. Figure 4 displays the performance visualization of the measurement outcomes from the LSTM model. The results of the analysis show that the LSTM-Nadam model outperforms the LSTM-Adam model in a number of accuracy domains.

Table 3. LSTM model results with Nadam optimisation

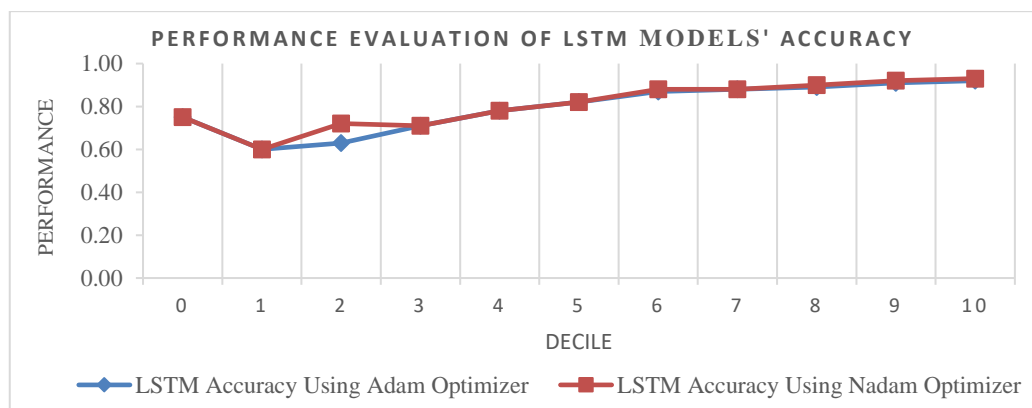| Decile	Accuracy	Recall	Precision	F1-Score
	LSTM+Nadam optimizer			
0	0.75	0.76	0.57	0.65
1	0.60	0.60	0.66	0.62
2	0.72	0.72	0.71	0.72
3	0.71	0.71	0.72	0.71
4	0.78	0.78	0.78	0.78
5	0.82	0.82	0.82	0.82
6	0.88	0.88	0.87	0.87
7	0.88	0.88	0.88	0.88
8	0.90	0.90	0.90	0.89
9	0.92	0.92	0.92	0.91
10	0.93	0.93	0.92	0.92

Figure 4. Comparison of the LSTM models' level of accuracy

3.2. Results from the long short-term memory model for prediction

Evaluation of the LSTM model's performance in foretelling final student data in a VLE. A total of 1,521 records from testing data are used to evaluate the LSTM model. Table 4 displays the outcomes of the LSTM model prediction using the Adam optimization algorithm.

The LSTM model's classification results used the Adam optimization algorithm, which produced the best classification outcomes; in the decile 0 data, 1,149 students were correctly categorized under the pass category. In addition, 369 data have classification results that are incorrect but still pass, despite the fact that they do not pass. The classification of students who actually failed was zero, this is in accordance with the actual data. There were three instances where data on students who did not pass were classified incorrectly and were actually students who did pass.

The same data testing is used in the LSTM model's prediction using the Nadam optimization algorithm. In Table 5, the outcomes of the Nadam optimization algorithm's prediction of the LSTM model are displayed. The LSTM model with the Nadam optimization algorithm has some higher accuracy values.

Table 4. LSTM model prediction outcomes using Adam optimization

Models	Course BBB Decile 0	Predicted	Predicted Fail	Pass	Accuracy
LSTM+Adam optimizer	Actual	Fail	0	369	75%
		Pass	3	1,149	
	Decile 1		Fail	Pass	
	Actual	Fail	164	205	60%
		Pass	402	750	
	Decile 2		Fail	Pass	
	Actual	Fail	147	222	63%
		Pass	333	819	
	Decile 3		Fail	Pass	
	Actual	Fail	148	221	71%
		Pass	209	943	
	Decile 4		Fail	Pass	
	Actual	Fail	208	161	78%
		Pass	167	985	
	Decile 5		Fail	Pass	
	Actual	Fail	227	142	82%
		Pass	130	1,022	
	Decile 6		Fail	Pass	
	Actual	Fail	201	168	87%
		Pass	18	1,134	
	Decile 7		Fail	Pass	
	Actual	Fail	266	103	88%
		Pass	77	1,075	
	Decile 8		Fail	Pass	
	Actual	Fail	260	109	89%
		Pass	44	1,110	
	Decile 9		Fail	Pass	
	Actual	Fail	283	86	91%
		Pass	37	1,115	
	Decile 10		Fail	Pass	
	Actual	Fail	283	109	92%
		Pass	39	1,113	

Table 5. LSTM model prediction outcomes using Nadam optimization

Models	Course BBB Decile 0	Predicted	Predicted Fail	Pass	Accuracy
LSTM+Nadam optimizer	Actual	Fail	0	369	75%
		Pass	0	1,152	
	Decile 1		Fail	Pass	
	Actual	Fail	166	203	60%
		Pass	411	741	
	Decile 2		Fail	Pass	
	Actual	Fail	135	234	72%
		Pass	188	964	
	Decile 3		Fail	Pass	
	Actual	Fail	159	210	71%
		Pass	227	925	
	Decile 4		Fail	Pass	
	Actual	Fail	209	160	78%
		Pass	175	977	
	Decile 5		Fail	Pass	
	Actual	Fail	220	149	82%
		Pass	123	1,029	
	Decile 6		Fail	Pass	
	Actual	Fail	223	146	88%
		Pass	42	1,110	
	Decile 7		Fail	Pass	
	Actual	Fail	257	112	88%
		Pass	72	1,080	
	Decile 8		Fail	Pass	
	Actual	Fail	260	109	90%
		Pass	20	1,132	
	Decile 9		Fail	Pass	
	Actual	Fail	272	97	92%
		Pass	30	1,112	
	Decile 10		Fail	Pass	
	Actual	Fail	260	109	93%
		Pass	11	1,141	

The LSTM model with the Nadam optimization algorithm generates the best classification outcomes at decile 0 by classifying the 1,152 passing students. Additionally, 369 data have classification outcomes that are inaccurate but still pass even though they do not pass. The classification of students who actually failed was zero, this is in accordance with the actual data. There are 0 students who do not pass and are correctly classified.

4. CONCLUSION

Based on analysis done to categorize student performance in a VLE using the LSTM model optimized with the Adam and Nadam optimization algorithm. The average accuracy of the LSTM model using Nadam optimization is 89%, with a maximum accuracy of 93%, while Adam's optimization-based LSTM model has a maximum accuracy of 92% and an average accuracy of 87%. The LSTM model with the Nadam optimization algorithm performs better than Adam's optimization algorithm in the prediction problem for VLE. The contribution of this study is the performance improvement of the LSTM model through hyperparameter optimization using the Adam and Nadam algorithm, which can be used as a reference when developing prediction systems based on LSTM. For further research and development, testing can be done using the meta-heuristic optimization algorithm and assessing the performance of the resulting model.

REFERENCES

[1] Y. Zheng, Z. Gao, Y. Wang, and Q. Fu, "MOOC dropout prediction using FWTS-CNN model based on fused feature weighting and time series," *IEEE Access*, vol. 8, pp. 225324–225335, 2020, doi: 10.1109/ACCESS.2020.3045157.

[2] K. A. Mayahi and M. Al-Bahri, "Machine learning based predicting student academic success," in *2020 12th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, Brno, Czech Republic: IEEE, Oct. 2020, pp. 264–268, doi: 10.1109/ICUMT51630.2020.9222435.

[3] E. S. Bhutto, I. F. Siddiqui, Q. A. Arain, and M. Anwar, "Predicting students' academic performance through supervised machine learning," in *2020 International Conference on Information Science and Communication Technology (ICISCT)*, KARACHI, Pakistan: IEEE, Feb. 2020, pp. 1–6, doi: 10.1109/ICISCT49550.2020.9080033.

[4] E. B. M. Magalhaes, G. A. Santos, F. C. D. Molina, J. P. J. da Costa, F. L. L. de Mendonca, and R. T. de Sousa, "Student dropout prediction in MOOC using machine learning algorithms," in *2021 Workshop on Communication Networks and Power Systems (WCNPS)*, Brasilia, Brazil: IEEE, Nov. 2021, pp. 1–6, doi: 10.1109/WCNPS53648.2021.9626227.

[5] M. B. Shah, M. Kaistha, and Y. Gupta, "Student performance assessment and prediction system using machine learning," in *2019 4th International Conference on Information Systems and Computer Networks (ISCON)*, Mathura, India: IEEE, Nov. 2019, pp. 386–390, doi: 10.1109/ISCON47742.2019.9036250.

[6] H. Zeineddine, U. Braendle, and A. Farah, "Enhancing prediction of student success: automated machine learning approach," *Computers and Electrical Engineering*, vol. 89, Jan. 2021, doi: 10.1016/j.compeleceng.2020.106903.

[7] B. Mounika and V. Persis, "A comparative study of machine learning algorithms for student academic performance," *International Journal of Computer Sciences and Engineering*, vol. 7, no. 4, pp. 721–725, Apr. 2019, doi: 10.26438/ijcse/v7i4.721725.

[8] A. Nabil, M. Seyam, and A. Abou-Elfetouh, "Deep neural networks for predicting students' performance," in *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*, Virtual Event USA: ACM, Mar. 2021, pp. 1380–1380, doi: 10.1145/3408877.3439685.

[9] H. Karimi, J. Huang, and T. Derr, "A deep model for predicting online course performance," *Association for the Advancement of Artificial Intelligence*, 2020, pp. 1–6.

[10] Y. Zhang, L. Chang, and T. Liu, "MOOCs dropout prediction based on hybrid deep neural network," in *2020 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, Chongqing, China: IEEE, Oct. 2020, pp. 197–203, doi: 10.1109/CyberC49757.2020.00039.

[11] S. Li and T. Liu, "Performance prediction for higher education students using deep learning," *Complexity*, vol. 2021, pp. 1–10, Jul. 2021, doi: 10.1155/2021/9958203.

[12] N. M. Aslam, I. U. Khan, L. H. Alamri, and R. S. Almuslim, "An improved early student's academic performance prediction using deep learning," *International Journal of Emerging Technologies in Learning*, vol. 16, no. 12, Jun. 2021, doi: 10.3991/ijet.v16i12.20699.

[13] J. Hu, X. Wang, Y. Zhang, D. Zhang, M. Zhang, and J. Xue, "Time series prediction method based on variant LSTM recurrent neural network," *Neural Process Letters*, vol. 52, no. 2, pp. 1485–1500, Oct. 2020, doi: 10.1007/s11063-020-10319-3.

[14] S. Qu, K. Li, B. Wu, K. Tang, and L. Weimin, "Multilayer LSTM with global access gate for predicting students performance using online behaviors," *Australian Journal of Intelligent Information Processing Systems*, vol. 17, no. 2, pp. 40–48, 2019.

[15] Y. Xiao, H. Yin, Y. Zhang, H. Qi, Y. Zhang, and Z. Liu, "A dual-stage attention-based Conv-LSTM network for spatio-temporal correlation and multivariate time series prediction," *International Journal of Intelligent Systems*, vol. 36, no. 5, pp. 2036–2057, May 2021, doi: 10.1002/int.22370.

[16] Z. Yu *et al.*, "A LSTM network-based learners' monitoring model for academic self-efficacy evaluation using EEG signal analysis," in *2020 5th IEEE International Conference on Big Data Analytics (ICBDA)*, Xiamen, China: IEEE, May 2020, pp. 154–159, doi: 10.1109/ICBDA49040.2020.9101313.

[17] K. Ramanathan and B. Thangavel, "Minkowski sommon feature map-based densely connected deep convolution network with LSTM for academic performance prediction," *Concurrency and Computation*, vol. 33, no. 13, Jul. 2021, doi: 10.1002/cpe.6244.

[18] J.-Y. Kim and S.-B. Cho, "Evolutionary optimization of hyperparameters in deep learning models," in *2019 IEEE Congress on Evolutionary Computation (CEC)*, Wellington, New Zealand: IEEE, Jun. 2019, pp. 831–837, doi: 10.1109/CEC.2019.8790354.

[19] E. Domingos, B. Ojeme, and O. Daramola, "Experimental analysis of hyperparameters for deep learning-based churn prediction in the banking sector," *Computation*, vol. 9, no. 3, Mar. 2021, doi: 10.3390/computation9030034.

[20] S. Bock and M. Weis, "A proof of local convergence for the adam optimizer," in *2019 International Joint Conference on Neural Networks (IJCNN)*, Budapest, Hungary: IEEE, Jul. 2019, pp. 1–8, doi: 10.1109/IJCNN.2019.8852239.

[21] Z. Zhixuan and H. Zaien, "Research and application of rectified-nadam optimization algorithm in data classification," *American Journal of Computer Science and Technology*, vol. 4, no. 4, p. 106, 2021, doi: 10.11648/j.ajcst.20210404.13.

[22] Y. Xie, "Student performance prediction via attention-based multi-layer long-short term memory," *Journal of Computer and Communications*, vol. 09, no. 08, pp. 61–79, 2021, doi: 10.4236/jcc.2021.98005.

[23] S. Hassan, H. Waheed, N. R. Aljohani, M. Ali, S. Ventura, and F. Herrera, "Virtual learning environment to predict withdrawal by leveraging deep learning," *International Journal of Intelligent Systems*, vol. 34, no. 8, pp. 1935–1952, Aug. 2019, doi: 10.1002/int.22129.

[24] W. Alsabhan, "Student cheating detection in higher education by implementing machine learning and LSTM techniques," *Sensors*, vol. 23, no. 8, Jan. 2023, doi: 10.3390/s23084149.

[25] T. T. Dien, S. Hoai, N. Thanh-Hai, and N. Thai-Nghe, "Deep learning with data transformation and factor analysis for student performance prediction," *International Journal of Advanced Computer Science and Applications(IJACSA)*, vol. 11, no. 8, 2020, doi: 10.14569/IJACSA.2020.0110886.

[26] A. Geng, A. Moghiseh, C. Redenbach, and K. Schladitz, "Comparing optimization methods for deep learning in image processing applications," *tm - Technisches Messen*, vol. 88, no. 7–8, pp. 443–453, Jul. 2021, doi: 10.1515/teme-2021-0023.

[27] S. Bock and M. Weiß, "Non-convergence and limit cycles in the adam optimizer," in *Artificial Neural Networks and Machine Learning – ICANN 2019: Deep Learning*, I. V. Tetko, V. Kůrková, P. Karpov, and F. Theis, Eds., in Lecture Notes in Computer Science, vol. 11728. Cham: Springer International Publishing, 2019, pp. 232–243, doi: 10.1007/978-3-030-30484-3_20.

[28] L. Li, W. Xu, and H. Yu, "Character-level neural network model based on Nadam optimization and its application in clinical concept extraction," *Neurocomputing*, vol. 414, pp. 182–190, Nov. 2020, doi: 10.1016/j.neucom.2020.07.027.

[29] J. Sultana, U. R. Macigi, and H. Farquad, "Student's performance prediction using deep learning and data mining methods," *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 8, no. 1S4, pp. 1018–1021, Jun. 2019.

[30] I. Ahmadianfar, O. Bozorg-Haddad, and X. Chu, "Gradient-based optimizer: a new metaheuristic optimization algorithm," *Information Sciences*, vol. 540, pp. 131–159, Nov. 2020, doi: 10.1016/j.ins.2020.06.037.

[31] Q. Zhang *et al.*, "Boosting adversarial attacks with nadam optimizer," *Electronics*, vol. 12, no. 6, Mar. 2023, doi: 10.3390/electronics12061464.

[32] W. W. Piegorsch, "Confusion matrix," in *Wiley StatsRef: Statistics Reference Online*, N. Balakrishnan, T. Colton, B. Everitt, W. Piegorsch, F. Ruggeri, and J. L. Teugels, Eds., 1st ed.Wiley, 2020, pp. 1–4, doi: 10.1002/9781118445112.stat08244.

[33] D. Krstinić, M. Braović, L. Šerić, and D. Božić-Štulić, "Multi-label classifier performance evaluation with confusion matrix," in *Computer Science & Information Technology*, AIRCC Publishing Corporation, Jun. 2020, pp. 01–14, doi: 10.5121/csit.2020.100801.

[34] X. Zhou and A. Del Valle, "Range based confusion matrix for imbalanced time series classification," in *2020 6th Conference on Data Science and Machine Learning Applications (CDMA)*, Riyadh, Saudi Arabia: IEEE, Mar. 2020, pp. 1–6, doi: 10.1109/CDMA47397.2020.00006.

BIOGRAPHIES OF AUTHORS

Edi Ismanto [ID] [G] [SC] [O] completed education bachelor's degree in the Department of Informatics Engineering, State Islamic University of Sultan Syarif Kasim Riau and master's degree in Master of Computer Science at Putra Indonesia University Padang. Currently, he works as a lecturer at the Department of Informatics, University Muhammadiyah of Riau. His research interests are in the field of machine learning algorithms and AI. He can be contacted at email: edi.ismanto@umri.ac.id.

Noverta Effendi [ID] [G] [SC] [O] completed education bachelor's degree in the Department of Electronics Engineering, at the State Islamic University of Sultan Syarif Kasim Riau and master's degree in Master of Information Technology at Putra Indonesia University Padang. Currently, he works as a lecturer in the Department of Computer Science, University Muhammadiyah of Riau. His research interests are in the field of machine learning algorithms and AI. He can be contacted at email: nover@umri.ac.id.