

Intelligent classification and performance prediction of multi-text assessment with recurrent neural networks-long short-term memory

Tukino Paryono^{1,2}, Eko Sedyono¹, Hendry¹, Baenil Huda^{1,2}, April Lia Hananto²,
Aviv Yuniar Rahman^{3,4}

¹Department of Computer Science, Faculty of Information Technology, Universitas Kristen Satya Wacana, Salatiga, Indonesia

²Department of Information System, Faculty of Computer Science, Universitas Buana Perjuangan, Karawang, Indonesia

³Department of Informatics Engineering, Faculty of Engineering, Universitas Widyagama Malang, Malang, Indonesia

⁴Department of Information & Communication Technology, School of Graduate Studies, Asia e University, Selangor, Malaysia

Article Info

Article history:

Received Nov 28, 2023

Revised Feb 21, 2024

Accepted Mar 1, 2024

Keywords:

Assessment

Long sort term memory

Optimizer

Recurrent neural network

Tokenizer

ABSTRACT

The assessment document at the time of study program accreditation shows performance achievements that will have an impact on the development of the study program in the future. The description in the assessment document contains unstructured data, making it difficult to identify target indicators. Apart from that, the number of Indonesian-based assessment documents is quite large, and there has been no research on these assessment documents. Therefore, this research aims to classify and predict target indicator categories into 4 categories: deficient, enough, good, and very. Learning testing of the Indonesian language assessment sentence classification model using recurrent neural networks-long short-term memory (RNN-LSTM) using 5 layers and 3 parameters produces performance with an accuracy value of 94.24% and a loss of 10%. In the evaluation with the Adamax optimizer, it had a high level of accuracy, namely 79%, followed by stochastic gradient descent (SGD) of 78%. For the Adam optimizer, Adadelta, and root mean squared propagation (RMSProp) have an accuracy rate of 77%.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Tukino Paryono

Department of Computer Science, Faculty of Information Technology, Universitas Kristen Satya Wacana
Salatiga, Indonesia

Email: 982021005@student.uksw.edu

1. INTRODUCTION

Assessment is a critical process for evaluating an institution's performance and quality, relying on systematic evaluation using predefined indicators [1]. The accreditation of academic programs is pivotal in determining accreditation status, measuring performance achievements against key indicators and other relevant criteria [2]. Institutional accreditation assesses overall performance using defined indicators, categorizing assessments as unaccredited or accredited with varying ratings, such as excellent, good, fair, and poor, valid for a specific period [3]–[6].

Achieving superior accreditation results contributes not only to enhancing graduate competitiveness but also to elevating institutional reputation and fostering excellence in service delivery [7], [8]. Assessment sheets obtained from accreditation processes play a crucial role in pinpointing areas for improvement [7], [9]. However, these collected sheets frequently encompass various academic domains, presenting challenges in accurately discerning assessment categories [7], [10].

Efforts to improve service provision include utilizing assessment documents in the native language, such as Indonesian. Despite promising results from sentiment analysis in English, optimizing Bahasa Indonesia assessment documents remains an opportunity for further investigation [5], [11]–[13]. Considerable research endeavors have been invested in studying comparative performance analysis, with a special emphasis on using recurrent neural networks (RNN) with long short-term memory (LSTM) for the analysis of textual and image data sets [14], [15]. Across these studies, a consistent finding emerged: image-focused deep learning models demonstrate greater efficiency compared to sentiment-focused counterparts, evidenced by enhanced task completion times and improved classification accuracies [16].

Additionally, substantial research efforts have been directed towards predicting learning gain. The primary aim is to develop robust computational models capable of forecasting student progress at an early stage. These efforts entail comparing various models, including Bayesian knowledge tracking, vanilla models, and LSTM models, which have led to significant strides in achieving highly accurate predictions with minimal training data [17].

This paper aims to propose an intelligent methodology for classifying and predicting assessment categories using LSTM-RNN. Starting with program assessment documents, we outline a framework for preprocessing unstructured data, transforming word collections into vectors, and employing LSTM and softmax methods for classification. Utilizing training data from Bahasa Indonesia datasets, we optimize parameters and algorithms for enhanced performance. Furthermore, we underscore the importance of visualizing information through diagrams or graphs to facilitate user understanding and interaction [7]. Through these efforts, we aim to contribute to advancing intelligent assessment methodologies, particularly in institutional performance evaluation contexts.

2. METHOD

Based on the research that will be carried out, the researcher prepares a test flow aimed at Figure 1. The method used to search for, handle, examine and categorize values uses assessment sheets as a data source to identify value categories. Because score sheets are unstructured data and include dynamic topics, special character symbols, and abbreviations, extracting and analyzing important information from score sheets can be a challenge. Therefore, the proposed methodological framework is divided into several modules: data preprocessing, text representation, text labeling, classification, and prediction. This research proposes a flow diagram for creating an assessment classification model which is divided into four stages: i) text preprocessing; ii) model design; iii) model training; and iv) testing and prediction.

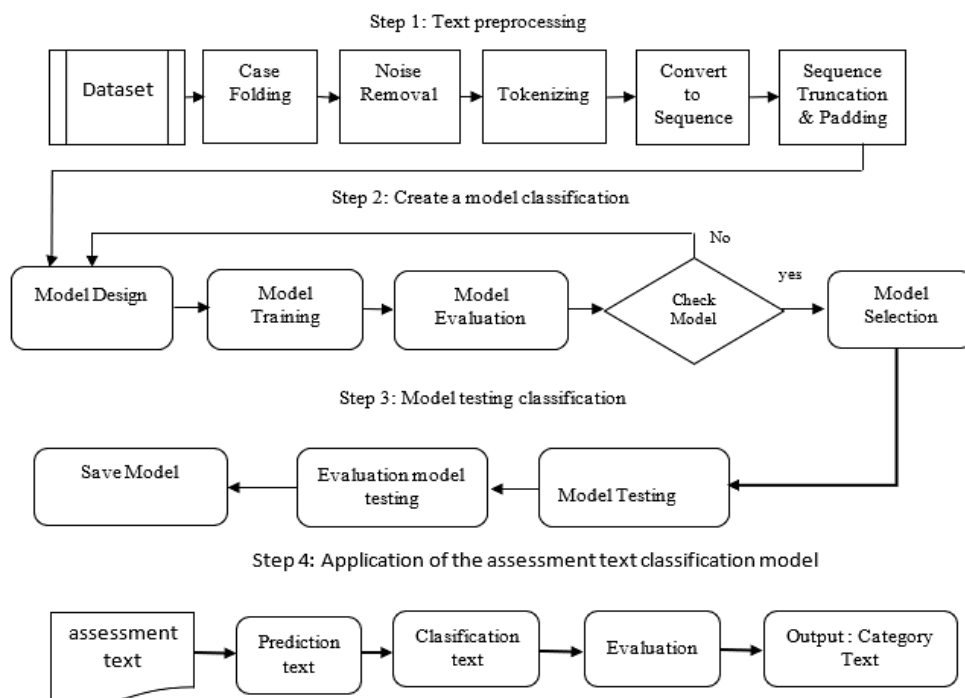


Figure 1. Design of assessment classification models

Figure 1 explains several methods used to search, handle, examine, and categorize values. The proposed system architecture uses assessment sheets as a data source to identify value categories. Because score sheets are unstructured data and include dynamic topics, special character symbols, and abbreviations, extracting and analyzing important information from score sheets can be a challenge. Therefore, the proposed methodological framework is divided into several modules: data preprocessing, text representation, text labeling, classification, and prediction.

The primary objective of the suggested framework is to employ LSTM-based text classification for the automated identification and examination of grade sheets. Accredited institutions provide the collected mark sheets to start the work phase. After document collection, text mining techniques are used for sentiment analysis. Score sheets should be labeled after the information has been thoroughly and carefully identified. By using sentiment extraction and analysis methods, the score sheet is categorized as deficient, enough, good, and very.

Next, dimensionless vectors are used to represent the assessment text using a word embedding model known as word index or word2vec. In the end, the mark sheet is classified based on the type of grade category and predicted using LSTM trained with softmax regression. You can automatically identify and estimate different grade categories by checking the grade sheet.

2.1. Data processing

Data processing is the process of converting an unstructured assessment text into a structured one, as shown in Figure 1. The assessment description explains the conditions for meeting performance targets based on standards and indicators in a lengthy text format. Special characters, numbers, and words are included in the assessment description [17], [18]. It's crucial to use text mining techniques to remove these characters from the scoring sheet before adding them to the classification model [19], [20]. The assessment sheet is written in paragraphs, so we begin by employing a sentence segmentation technique to divide the paragraph into individual sentences [21], [22].

2.2. Tokenization

The process of tokenizing a complex text is to reduce it to a list of words known as tokens. Word spacing, punctuation, hashtags, and non-text characters are typically found in complex text [23], [24]. The suggested system breaks each text segment into word (bag of words representation) by removing non-alphanumeric characters using an n-gram tokenization technique. Each text is then represented as a word sequence for processing after this step.

2.3. Convert to sequence

Large assessments can be difficult to categorize because the assessment form contains formal descriptions with frequently used words in lowercase or capital letters based on the sentence's position. When writing capital and lowercase letters inconsistently. The utilization of the stemming technique offers a solution to circumvent this issue. Through this approach, each word within the text is mapped into a unified feature space [25], [26].

2.4. The recurrent neural networks model

An artificial neural network architecture named a RNN uses the principle of repetition to store output using internal memory based on specific layers [27], preserving previously learned information. It then forwards the output to become input in the subsequent iteration, predicting the output of the subsequent layer, sequentially, due to the lengthy input circuit. Because of this, RNNs are particularly good at predicting continuous data sequences.

There are two types of RNN architecture [28]–[30]: time step RNN and conventional RNN. A traditional RNN, like the one in Figure 2, only processes input x during the training phase and generates output y . It lacks the concept of time. Conversely, Figure 3 represents an unrolled version of a traditional RNN with multiple inputs. Time is represented by the symbol t on the horizontal axis and processing proceeds from left to right to express as a time step.

In this case, case the text "organizing an article writing workshop" is denoted by the letter x . The letters "organizing", "workshop", "writing", and "article" are processed one by one in this order. Until the final data input, the hidden layer continuously transfers data on a time scale (t) to the next hidden layer. The words represented by the symbol y ($y_1, y_2, y_3, y_4, \dots, y_t$) are the output of the RNN. The internal state, represented by S_t , is provided from a series of time steps to the next, stored by the RNN for each processing. $U, V,$ and W are the parameters of each RNN box. RNN has a competitive advantage in text processing, machine translation, and language modeling with this model.

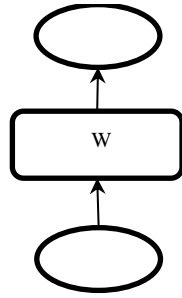


Figure 2. RNN conventional

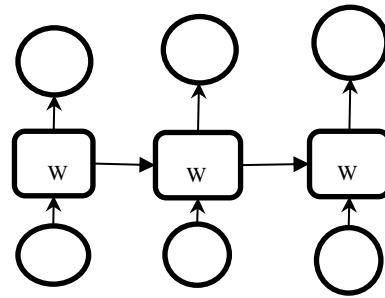


Figure 3. RNN unrolled

2.5 Long short-term memory

These method models have the same basic architecture, depicted in Figure 4, where the distinctive capability of LSTM to calculate hidden states capable of retaining long-term dependencies is highlighted. Additionally, sending sequential data is well suited to the neural network architecture known as LSTM [31]–[33]. A forward propagation process is required for classification, especially to perform the softmax activation function.

In Figure 4, the RNN-LSTM model is presented which is divided into preprocessing, input layer, and output layer. We propose this method to categorize and estimate values. Before reaching the output layer, the RNN-LSTM model performs cleaning on the score sheet by removing redundant words and converting them to a word vector. This vector is then used as input for the input layer, LSTM, and hidden layers.

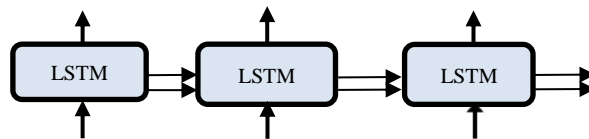


Figure 4. Model LSTM

2.6. Softmax regression

Softmax regression [34]–[36] is a model for generating probabilities that represent outcomes from several classes. The input for softmax regression is still a d-dimensional vector, h_T , which is the output of the LSTM model. As mentioned previously, we use N words to predict the polarity of each sentence in document D, which is discussed as in (1).

$$p(pol|X) = Softmax(X) = \frac{\exp(X)}{\sum_{j=0}^i (X_i)}, X = W_s h_T + b_s \tag{1}$$

$$Pol = \text{arg max}_p(pol|X_k) \tag{2}$$

$$Loss = \frac{1}{M} \sum_{s=1}^M Y_s \cdot \text{log}_p(pol_s|X_s) \tag{3}$$

In the equation, the input time step j and sentiment category are denoted by x and i , respectively. Polarity prediction is based on input data with maximum probability, as explained in (2). To evaluate the model's effectiveness, a loss function is employed, which evaluates each labeled sentence using cross entropy, as depicted in (3).

Figure 5 illustrates a classification model evaluation using RNN-LSTM. This model utilizes a RNN integrating LSTM cell types regarding categorizing text based on its evaluation. Within this model, the assessment text supplied, such as "The facilities and infrastructure are adequate to carry out educational activities," is broken down into pertinent tokens, including "facilities," "and," "infrastructure," "adequate," and "educational activities". Each token is then converted into a vector that can be processed by the model. The next step involves using an LSTM layer to understand the sequence of these tokens. LSTM has the ability to store long-term information in data sequences, making it very suitable for text processing. Following the LSTM layer, the resultant vector for each token is inputted into the fully connected (FC) layer. The primary task of

this FC layer is to further manipulate these vectors to facilitate their interpretation for classification objectives. The model output is in the form of rating categories, for example "poor", "medium", "good", or "very". This categorization is determined based on the model's understanding of the assessment text that it has processed. For example, based on the text provided, the model can produce "sufficient" output because the facilities and infrastructure owned are considered adequate to carry out educational activities.

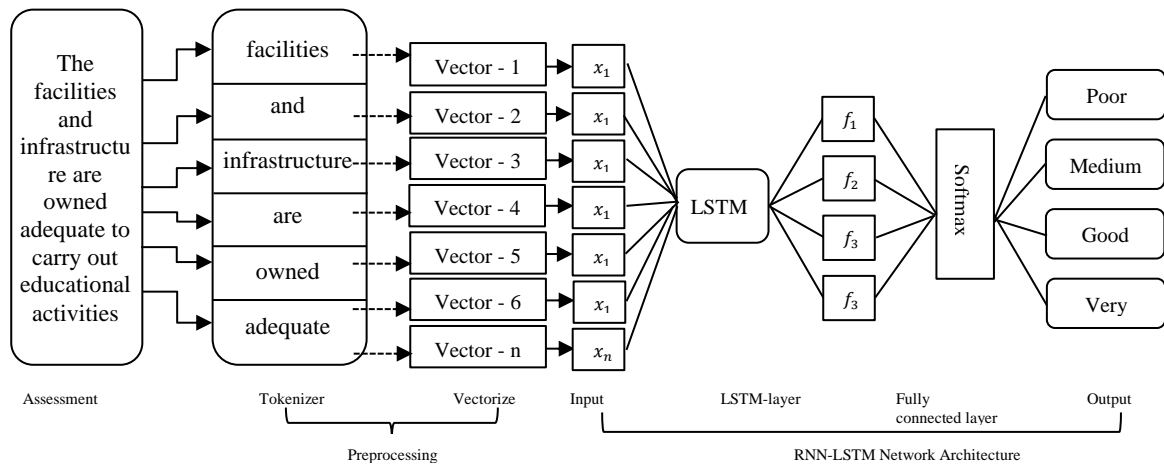


Figure 5. Assessment classification model using RNN-LSTM

2.7. LSTM training process design

In an effort to create the best classification model, development using the LSTM model considers four different variations with unique parameter settings. In addition, four different optimization algorithms were applied, which are detailed in Table 1, to obtain optimal performance in carrying out classification. This process allows extensive exploration of combinations of parameter values and optimization methods to achieve optimal results.

Table 1. Variations of learning model training

No	Adam (lr, bs, ep)	Adamax (lr, bs, ep)	Adadelta (lr, bs, ep)	SGD (lr, bs, ep)	RMSProp (lr, bs, ep)
1	0.01,16,10	0.01,16,25	0.5, 16, 33	0.1, 16, 46	0.01, 32, 26
2	0.001, 32,19	0.001, 32,31	0.5, 16, 43	0.1, 16,51	0.001, 32, 32
3	0.005,32, 22	0.005,32,14	1.01, 32, 31	0.1, 32, 32	0.01, 128, 23
4	0.005,32,46	0.005,64,36	1.0, 4, 40	0.1, 32, 35	-

3. RESULTS AND DISCUSSION

The results and discussion in smart classification and prediction: multi-text assessment institutional performance review using RNN-LSTM have been carried out and are addressed in sub-section 3.1. The assessment form is a performance assessment that aims to achieve targets that are very good, good, fair, and not good. Because the assessment form is unstructured text, the preprocessing discussed in subsection 3.2 must be converted into structured text. In this study justifies the classification and prediction methods, and this section uses test data from institutions to justify the evaluation methods.

3.1. Data set and parameters

Using repeated experiments, the LSTM model is used to predict optimal categories and classify target categories for performance. Assessment forms are collected from various fields, including computers, management, education, engineering, and health. A total of 1,500 records were collected from institutions. Preprocessing models are employed for the analysis of the gathered data. Following preprocessing, the data was normalized, with 70% allocated for training and 30% for testing classification. This dataset is then utilized to train and evaluate LSTM models using word index. The suggested LSTM model is contrasted with RNN, random forest (RF), and support vector machine (SVM) classification models for document assessment. The optimization algorithms, including Adamax, Adadelta, SGD, Adam, and root mean squared propagation

(RMSProp), are applied to test data to enhance performance models in classification seeks to achieve accuracy, precision, recall, and f1 scores. Five layers make up the LSTM model for text classification: two layers dense, spatial dropout, embedding, and LSTM. The model parameter values that produce the smallest loss function values are found using an optimizer algorithm. The learning rate of an optimization algorithm determines how many steps it takes to find the smallest value.

3.2. Performance evaluation

In this section, we delve into the training outcomes to assess the on the evaluation dataset. The extent to which word embedding models utilizing Word index influence the classification of assessment criteria categories was also assessed. The effectiveness of the suggested LSTM text classification model is contrasted with that of alternative classification techniques. LSTM and five optimizer algorithms are used in this experiment to test the scoring text classification model. Experimental findings show that learning rate, batch size, and epoch parameter values impact the accuracy rate. The initial experiment involved the combination of three parameters and was conducted four times utilizing the Adam optimizer algorithm along with the LSTM model detailed in Table 2. Achieving learning rate 0.005, batch size 32, and epoch 22, the third experimental scenario demonstrates the highest accuracy rate at 93.08%.

Table 2. LSTM-Adam training results

Learning rate	Batch_size	Epoch	Training result			
			Acc	Val_acc	Loss	Val_loss
0.01	16	10	91.85	79.65	17.15	92.94
0.001	32	19	93.08	80.81	13.26	85.65
0.005	32	22	92.11	80.23	15.50	80.23
0.005	32	46	92.63	78.49	20.12	85.76

The second experiment involved using an LSTM model with the Adamax optimization algorithm and three different parameters, namely learning rate, batch size, and epoch, as detailed in Table 3. Four tests were conducted, each utilizing different parameter combinations. The results encompassed evaluations of training accuracy (Acc), validation accuracy (Val_acc), training loss (Loss), and validation loss (Val_loss). This table illustrates fluctuations in model performance based on the specific parameter combinations utilized. Achieving learning rate 0.01, batch size 16, and epoch 25, the third experimental scenario demonstrates the highest accuracy rate at 94.24%.

Table 3. LSTM-adamax training results

Learning rate	Batch_size	Epoch	Training result			
			Acc	Val_acc	Loss	Val_loss
0.01	16	25	94.24	86.05	10.10	94.32
0.001	32	31	90.36	83.14	25.50	71.71
0.005	32	14	92.04	83.30	17.72	81.19
0.005	64	36	94.18	82.56	12.54	88.40

Table 4 showcases the outcomes of four experiments conducted to enhance the LSTM model utilizing the Adadelta algorithm, encompassing three separate sets of parameters. The results show variations in model performance, where the highest accuracy was achieved in the fifth experiment with a value of 93.60%, using batch size 4 and epoch 40. Meanwhile, the second highest result was achieved in the fourth experiment, reaching 92.95% accuracy, with batch size 16, epoch 43, and learning rate 0.5. This analysis shows the importance of experimenting with parameter combinations in improving the performance of classification models.

Table 4. LSTM-adadelta training results

Learning rate	Batch_size	Epoch	Training result			
			Acc	Val_acc	Loss	Val_loss
0.5	16	33	91.14	81.98	19.94	68.30
0.5	16	43	92.95	83.70	17.54	67.13
1.01	32	31	92.50	83.14	17.40	86.06
1.0	4	40	93.60	81.98	15.99	93.54

The fourth experiment entailed executing four distinct test scenarios using the LSTM-SGD model, concentrating on parameters including learning rate, batch size, and epochs. Table 5 documents the outcomes for these parameter arrangements. Significantly, the highest accuracy of 91.85% was achieved in the initial scenario, which utilized a batch size of 16 with 46 epochs. In the second scenario, with a batch size of 32 and 35 epochs, the second highest accuracy was achieved, reaching 91.40%.

Table 5. LSTM-SGD training results

Learning rate	Batch_size	Epoch	Training result			
			Acc	Val_acc	Loss	Val_loss
0.1	16	46	91.85	82.56	20.19	89.12
0.1	16	51	91.53	79.65	91.53	90.93
0.1	32	32	90.17	82.56	22.94	88.66
0.1	32	35	91.40	79.65	18.27	11.51

Table 6 outlines the test outcomes from the ultimate experiment employing the LSTM-RMSProp model in three distinct experimental scenarios, featuring adjustments in learning rate, batch size, and epoch parameters. The testing revealed that the LSTM-RMSProp model reached its highest accuracy, reaching 94.57%, with a batch size of 128 and 36 epochs. Moreover, an accuracy of 93.60% was achieved with a batch size of 32 and 26 epochs. These findings offer insights into the model's proficiency in handling diverse parameter configurations.

Table 6. LSTM-RMSProp training results

Learning rate	Batch_size	Epoch	Training result			
			Acc	Val_acc	Loss	Val_loss
0.01	32	26	93.60	83.14	11.70	1.60
0.001	32	32	93.47	81.40	13.15	83.37
0.01	128	23	92.58	72.73	13.13	15.33

The classification learning model uses five optimization algorithms and learning rate parameters to produce a relatively stable level of accuracy. The lower bound of the accuracy of the optimization algorithm is based on the current epoch and the upper bound on the loss function extending from the initial epoch to the final epoch that contributes to convergence. The Adamax, Adelta, Adam, SGD, and RMSProp optimization algorithms provide better results, as in Table 7.

Table 7. Comparison of optimization algorithm test results

Optimization	Epoch	Acc (%)	Loss (%)
Adamax	25/100	94.24	10
Adadelta	40/100	93.60	16
Adam	19/100	93.08	13
SGD	46/100	91.85	20
RMSProp	26/100	92.89	10

In the initial trial setup, the highest accuracy rate recorded was 94.24%, achieved with a learning rate set at 0.01, a batch size of 25, and 41 epochs. The second-highest accuracy, at 93.60%, was observed in the fourth scenario, which utilized a learning speed of 1.0., a batch size of 4, alongside 40 epochs. The slight variance in accuracy between these two situations implies that their testing performance was similarly efficient.

Figure 6 displays a graph showing the progression of accuracy over time for the LSTM-Adam model during training, using optimal scenarios on the evaluation dataset. Initially, the model's accuracy stood at 58.67%, and the validation accuracy was recorded at 63.95%. The training process reached its peak at epoch 22, with validation accuracy of 80.81% and accuracy of 93.08%. Model training ended at the 19 and epoch out of a total of 100 epochs because the model no longer experienced an increase in accuracy in subsequent epochs and the model also did not appear to show overfitting.

By using test data that has never been tested, the LSTM-Adam classification model has an accuracy rate of 93.08% and is kept for testing. The confusion matrix, as shown in Figure 7, is used to evaluate it based on the prediction results. The model's capability to categorize test data into each class is validated, with average macro precision, recall, and F1 score values stand at 67%, 63%, and 66%, respectively.

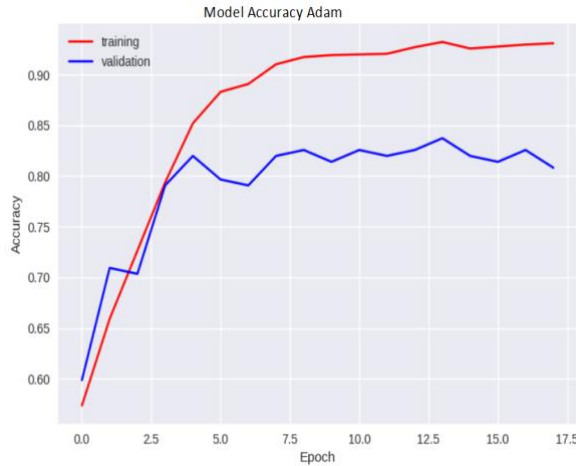


Figure 6. Training accuracy of the LSTM-Adam

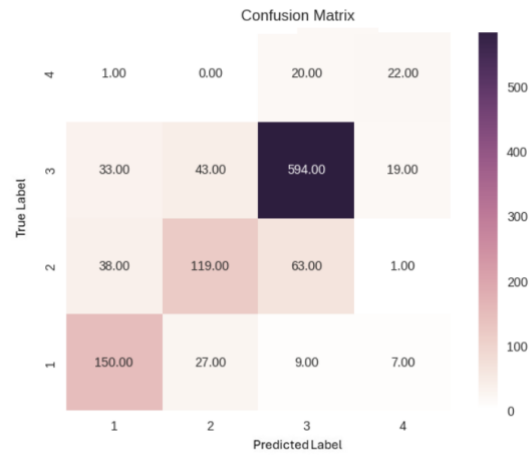


Figure 7. Confusion matrix of the LSTM-Adam

The model can classify data into four different categories. First, the “not predicted” category has 193 data, with 78% accurate and 22% inaccurate. Second, the "fairly predictable" category has 221 data, with an accuracy rate of 54%. Third, the "good" category has 689 data, with an accuracy rate of 86% and 14% inaccurate. Finally, the "very good" category has 43 data, with an accuracy rate of 51%.

Figure 8 shows the development of accuracy LSTM-Adamax training results with the optimal scenario on the assessment dataset over time. The model accuracy in the first epoch was 62.61% with validation accuracy of 69.10%. The training process reached its peak at epoch 25, with validation accuracy of 86.05% and accuracy of 94.24%. Model training ended at epoch 25 out of a total of 100 epochs because the model no longer experienced an increase in accuracy in the next epoch. Moreover, overfitting does not seem to occur in this model.

By using test data that has never been tested, the optimal scenario LSTM-Adamax classification model with an accuracy rate of 94.24% is saved for testing. As shown in Figure 9, the confusion matrix is used to evaluate it based on the prediction results. The model has demonstrated its capability to categorize test data into distinct classes, with the mean macro figures representing precision, recall, and F1 score being 66%, 64%, and 65%, respectively.

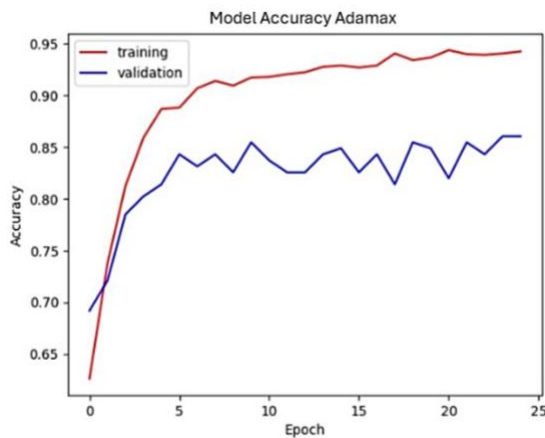


Figure 8. Training accuracy of the LSTM-Adamax

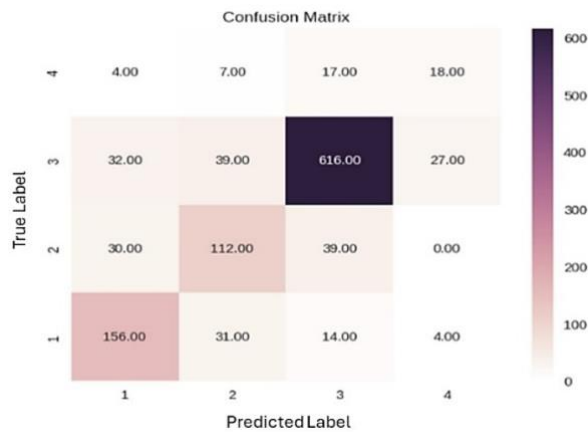


Figure 9. Confusion matrix of the LSTM-Adamax

The training graph shows the changes at each epoch being trained. Figure 10 shows the results of using the best scenario to train the LSTM-Adadelta model on the assessment dataset. In epoch 1 the accuracy value was 59.70%; it increases with each epoch and reaches a maximum of 93.60% in epoch 40; validation accuracy was 81.98%. Model training ends at epoch 40 out of a total of 100 epochs.

The LSTM-Adadelta classification model attained an accuracy of 93.60%, demonstrating effective category prediction. Subsequently, a more thorough evaluation was conducted using a confusion matrix, depicted in Figure 11. The analysis revealed a precision rate of 64%, a recall rate of 69%, and an F1 score of 65%. Such an analysis offers a comprehensive insight into the model's capability in category prediction and reinforces the reliability of the obtained results. The model is capable of accurately classifying the following categories: i) the category was not predicted with 169 records and was not predicted with 72 records (70% appropriate); ii) the category was quite correctly predicted with 116 records and incorrectly predicted with 73 records (61% appropriate); iii) the good category was correctly predicted with 573 records and 63 records incorrectly (90% appropriate); and iv) the very good category was correctly predicted with 26 records and incorrectly predicted with 54 records (80% appropriate).

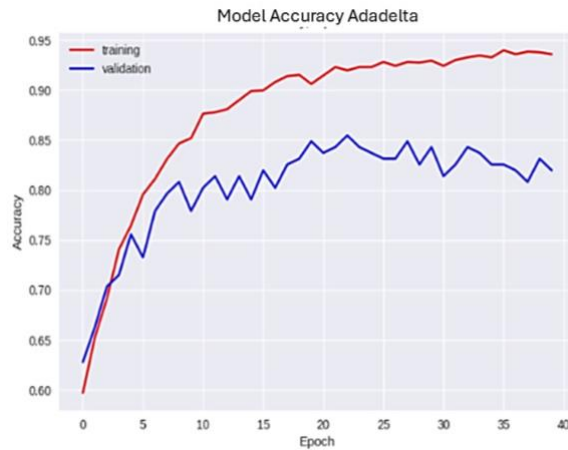


Figure 10. Training accuracy of the LSTM-Adadelta

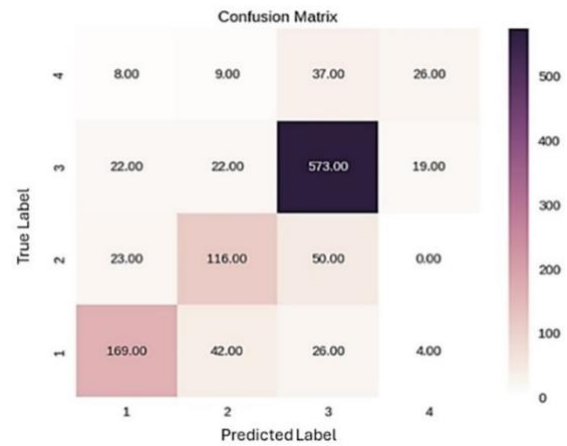


Figure 11. Confusion matrix of the LSTM-Adadelta

Figure 12 presents the training results of the LSTM-SGD model, illustrating the change in accuracy at each epoch. The initial epoch accuracy value was 57.82%; after that, it consistently increased at each epoch and peaked at 91.85% at the 46 epoch and 82.56% for validation. At epoch 48 out of a total of 100 epochs, model training ends because it has reached its maximum.

The best-performing LSTM-SGD classification model, which attained an accuracy of 91.58%, was chosen and reserved for additional assessment. A confusion matrix, illustrated in Figure 13, was used to evaluate this model. The assessment resulted in a precision rate of 68%, a recall rate of 63%, and an F1-score of 65%.

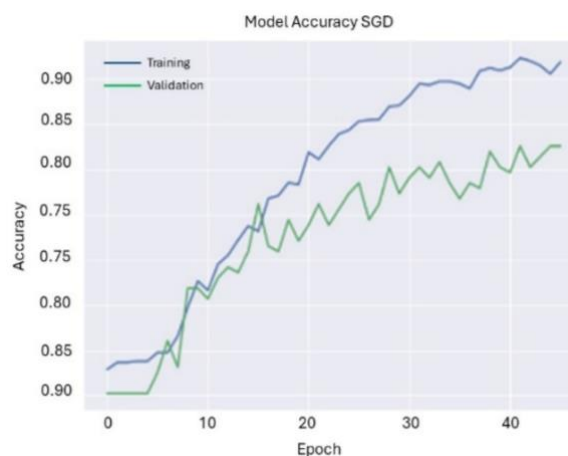


Figure 12. Training accuracy of the LSTM-SGD

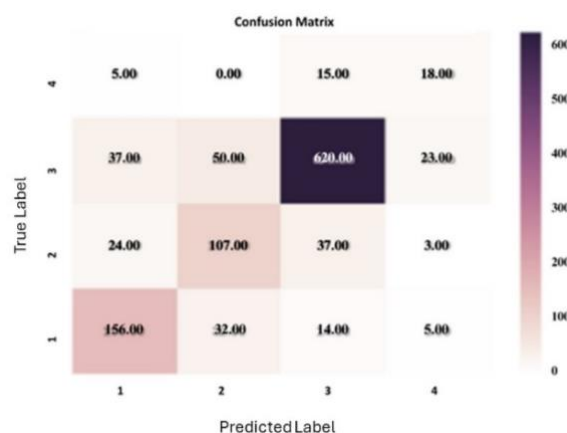


Figure 13. Confusion matrix of the LSTM-SGD

The model successfully categorized various classes, such as non-predicted, moderately predicted, good, and very good. It incorrectly predicted 156 data points, attaining an accuracy rate of 75%, and somewhat accurately predicted 107 data points, with a 63% accuracy rate. Moreover, it correctly classified 620 data points as good with an 85% accuracy level, and 18 data points as very good, achieving a 47% accuracy level. This analysis offers a detailed overview of the model's proficiency in classifying data into the correct categories.

Figure 14 illustrates the level of training accuracy of the LSTM-RMSProp model by showing variations in accuracy at each epoch stage. The accuracy value was 60.16% in the first epoch, and continued to increase with each epoch, reaching a maximum of 93.60% in the 26th epoch, and 83.14% for validation. Model training ends at epoch 26 out of a total of 100 epochs because the model no longer becomes more accurate. The LSTM-RMSProp classification model is the best scenario with an accuracy of 93.60% and evaluated using a confusion matrix as in Figure 15 with a precision value of 65%, recall 69%, and F1-score 66%.

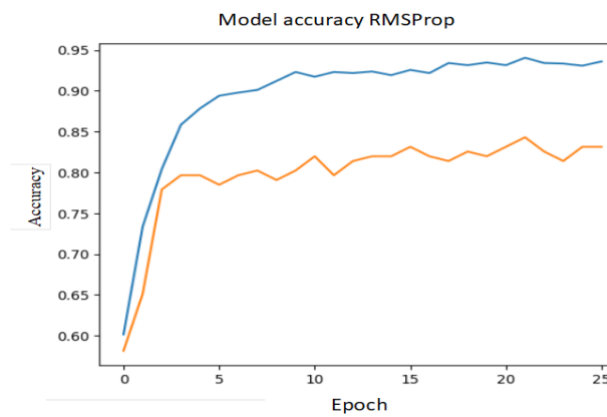


Figure 14. Training accuracy of the LSTM-RMSProp



Figure 15. Confusion matrix of the LSTM-RMSProp

The model demonstrated its capacity to classify categories at different levels of accuracy. It incorrectly predicted 165 data points, yielding a 73% accuracy rate, while it predicted 121 data points with moderate accuracy, achieving a 68% accuracy rate. Furthermore, it accurately classified 579 data points as well, with an 89% accuracy rate, and very accurately classified 27 data points, attaining a 31% accuracy rate. This analysis offers an extensive view of the model's competency in performing classifications across various existing categories.

The application of various optimization algorithms such as Adamax, Adadelta, Adam, SGD, and RMSProp allows for achieving optimal performance in scoring category predictions. The performance evaluation of the classification models recorded in Table 8 presents an analysis of how effective each algorithm is in producing optimal results in predicting assessment categories. This provides a clearer picture of the strengths and weaknesses of each algorithm in the context of rating category predictions.

Table 8. Comparison of LSTM and optimizer model evaluations

Optimizer	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Adam	77	67	63	66
Adamax	79	66	64	65
Adadelta	77	64	69	65
SGD	78	68	63	65
RMSProp	77	65	69	66

The paragraph describes the analysis of model performance based on experiments conducted during model training for categorizing assessment criteria. The highest accuracy results, as depicted in Figure 16, demonstrate a significant success rate in classification tasks. This indicates the model's ability to accurately categorize assessment criteria. Additionally, Figure 17 illustrates a low loss value, suggesting that the model can generate predictions with a high degree of accuracy, further validating its effectiveness in the classification process.

The paragraph provides an analysis of the performance achieved using different optimization algorithms. The Adamax optimizer attained the highest accuracy rate of 79%, indicating their effectiveness in accurately classifying assessment criteria. Similarly, SGD optimizers achieved the highest precision rate of

68% indicating their ability to minimize false positives in the classification process. In contrast, the Adadelata optimizer exhibited the highest recall rate of 69%, implying its proficiency in capturing relevant instances of assessment criteria. Additionally, Adam and RMSProp optimizer achieved the top F1 score rate of 66%, demonstrating its well-balanced performance in precision and recall. This highlights the importance of choosing the right optimization algorithms for specific situations, as they have a significant impact on determining the efficiency of the classification model.

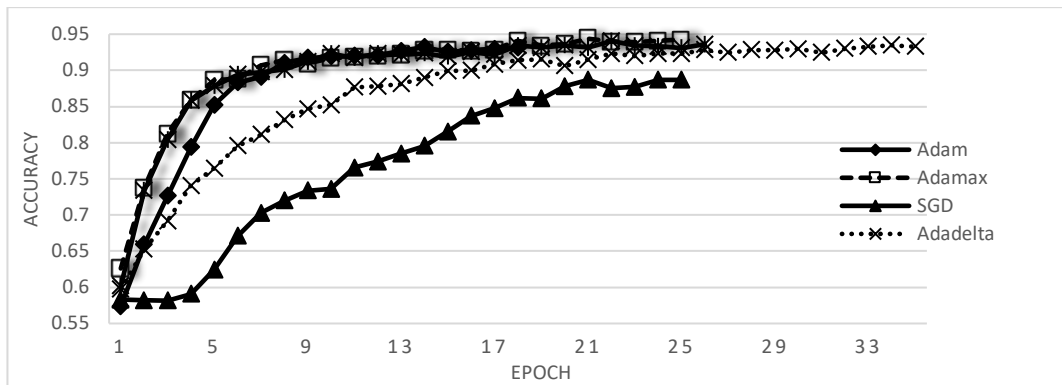


Figure 16. Comparison of training accuracy data

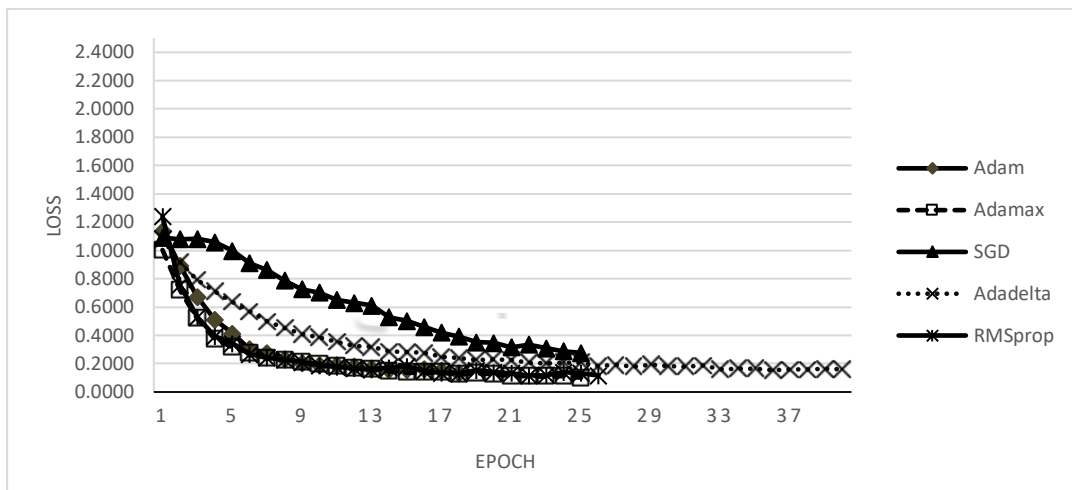


Figure 17. Comparison of training loss data

Table 9 presents a comparison between the results of previous studies and those obtained using the suggested approach, highlighting significant variances in accuracy, precision, recall, and F1 scores across different research. Earlier studies showed that the accuracy of LSTM-RNN ranged between 70% and 85.69%, while the accuracy for LSTM alone was reported at 53%. On the other hand, the new method demonstrated a substantially higher accuracy rate of 94.24%, along with a maximum F1 score of 94.24%. These findings suggest that the proposed method surpasses existing techniques in predicting research categories.

Furthermore, research has also been carried out on the comparison of predicting learning. The results in testing can be said to be lower than previous research, namely an accuracy of 70%. In this research, it is stated that the F1-score cannot be accepted or produced because the amount of data is different, causing the average to not be detected.

Research has also been carried out in the LSTM model taking the topic of deep learning models for prediction of initial student performance in massive open online courses (MOOCs). This research obtained accuracy results of 53%. In this case, the results received are very small and minimal compared to previous research. Therefore, scientists propose the intelligent classification and prediction method, which involves using multiple texts assessment of institutional performance reviews through RNN-LSTM. This approach achieved a recall

rate with a 66% recall rate and an F1 score of 94.24%. These outcomes are considered a significant improvement in the effectiveness of LSTM and RNN techniques compared to previous research.

Table 9. Comparison table of research that has been carried out with proposed research

Researcher	Method	Accuracy (%)	Precision	Recall	F1-score
[37]	LSTM-RNN	85.69	-	-	-
[38]	LSTM-RNN	70	-	-	-
[39]	LSTM	53	-	-	-
Proposed Method	LSTM-RNN	94.24	66%	66%	94.24%

4. CONCLUSION

This study used RNN-LSTM, a six-layer, three-parameter classification model for assessment sentences in Indonesian. The best learning rate was achieved using the Adamax algorithm, produce an accuracy of 94.24% and a 10% loss. Adamax proves to be a beneficial algorithm for iteratively selecting random samples of data from one or more training data segments within a single iteration. It corrects this randomly sampled data using a first gradient rule to gauge changes in both the input value and the function. Thus, the suggested classification model's evaluation results produce an accuracy level of 79%. Future research needs to identify scientific field topics in assessment datasets.

ACKNOWLEDGEMENTS

The study took place at Universitas Buana Perjuangan Karawang in Indonesia, sponsored by the LPPM Research Fund of the same institution. Its objective was to facilitate the advancement of Computer Science Ph.D. programs at Universitas Kristen Satya Wacana, Salatiga, Indonesia. This collaborative effort between the two universities aims to foster substantial progress in the field of computer science within higher education.




REFERENCES

- [1] S. Bosse, "Learning damage event discriminator functions with distributed multi-instance RNN/LSTM machine learning-mastering the challenge," *Procedia Manufacturing*, vol. 52, no. 2019, pp. 193–202, 2020, doi: 10.1016/j.promfg.2020.11.034.
- [2] C. W. Lee and D.-G. Yoo, "Development of leakage detection model and its application for water distribution networks using RNN-LSTM," *Sustainability*, vol. 13, no. 16, Aug. 2021, doi: 10.3390/su13169262.
- [3] G. Abbas, M. Nawaz, and F. Kamran, "Performance comparison of NARX RNN-LSTM neural networks for LiFePO₄ battery state of charge estimation," *2019 16th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, pp. 463–468, 2019, doi: 10.1109/IBCAST.2019.8667172.
- [4] D. Kaur, R. Kumar, N. Kumar, and M. Guizani, "Smart grid energy management using RNN-LSTM: A deep learning-based approach," *2019 IEEE global communications conference (GLOBECOM)*, pp. 1–6, 2019, doi: 10.1109/GLOBECOM38437.2019.9013850.
- [5] A. Sherstinsky, "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network," *Physica D: Nonlinear Phenomena*, vol. 404, 2020, doi: 10.1016/j.physd.2019.132306.
- [6] S. Hansun and J. C. Young, "Predicting LQ45 financial sector indices using RNN-LSTM," *Journal of Big Data*, vol. 8, no. 1, 2021, doi: 10.1186/s40537-021-00495-x.
- [7] M. A. Kausar, A. Soosaimanickam, and M. Nasar, "Public sentiment analysis on twitter data during COVID-19 outbreak," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 2, pp. 415–422, 2021, doi: 10.14569/IJACSA.2021.0120252.
- [8] M. Dua, D. Makhija, P. Y. L. Manasa, and P. Mishra, "A CNN-RNN-LSTM based amalgamation for alzheimer's disease detection," *Journal of Medical and Biological Engineering*, vol. 40, no. 5, pp. 688–706, 2020, doi: 10.1007/s40846-020-00556-1.
- [9] F. Liu, Z. Chen, and J. Wang, "Video image target monitoring based on RNN-LSTM," *Multimedia Tools and Applications*, vol. 78, no. 4, pp. 4527–4544, 2019, doi: 10.1007/s11042-018-6058-6.
- [10] K. Kumar and M. T. U. Haider, "Enhanced prediction of intra-day stock market using metaheuristic optimization on RNN-LSTM network," in *New Generation Computing*, vol. 39, no. 1, pp. 231–272, 2021.
- [11] E. Yuniarti, N. Nurmaini, B. Y. Suprpto, and M. Naufal Rachmatullah, "Short term electrical energy consumption forecasting using RNN-LSTM," in *2019 International Conference on Electrical Engineering and Computer Science (ICECOS)*, Oct. 2019, pp. 287–292, doi: 10.1109/ICECOS47637.2019.8984496.
- [12] A. Khan and A. Sarfaraz, "RNN-LSTM-GRU based language transformation," *Soft Computing*, vol. 23, no. 24, pp. 13007–13024, 2019, doi: 10.1007/s00500-019-04281-z.
- [13] A. M. Alayba and V. Palade, "Leveraging Arabic sentiment classification using an enhanced CNN-LSTM approach and effective Arabic text preparation," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 10, pp. 9710–9722, 2022, doi: 10.1016/j.jksuci.2021.12.004.
- [14] S. Gajendran, M. D, and V. Sugumaran, "Character level and word level embedding with bidirectional LSTM – Dynamic recurrent neural network for biomedical named entity recognition from literature," *Journal of Biomedical Informatics*, vol. 112, 2020, doi: 10.1016/j.jbi.2020.103609.
- [15] T. Najafi, R. Jaafar, R. Remli, and W. A. W. Zaidi, "A classification model of EEG signals based on RNN-LSTM for diagnosing focal and generalized epilepsy," *Sensors*, vol. 22, no. 19, 2022, doi: 10.3390/s22197269.
- [16] N. Rai, D. Kumar, N. Kaushik, C. Raj, and A. Ali, "Fake news classification using transformer based enhanced LSTM and BERT,"




- International Journal of Cognitive Computing in Engineering*, vol. 3, pp. 98–105, 2022, doi: 10.1016/j.ijcce.2022.03.003.
- [17] S. N. Mohanty, E. L. Lydia, M. Elhoseny, M. M. G. Al Otaibi, and K. Shankar, “Deep learning with LSTM based distributed data mining model for energy efficient wireless sensor networks,” *Physical Communication*, vol. 40, 2020, doi: 10.1016/j.phycom.2020.101097.
- [18] M. D. P. P. Goonathilake and P. P. N. V. Kumara, “CNN, RNN-LSTM based hybrid approach to detect state-of-the-art stance-based fake news on social media,” in *2020 20th International Conference on Advances in ICT for Emerging Regions (ICTer)*, Nov. 2020, pp. 23–28, doi: 10.1109/ICTer51097.2020.9325477.
- [19] A. F. Ganai and F. Khursheed, “Predicting next word using RNN and LSTM cells: stastical language modeling,” *2019 fifth international conference on image information processing (ICIIP)*, vol. 2019, pp. 469–474, 2019, doi: 10.1109/ICIIP47207.2019.8985885.
- [20] X. Liu, H. Shi, X. Hong, H. Chen, D. Tao, and G. Zhao, “Hidden states exploration for 3D skeleton-based gesture recognition,” *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1846–1855, 2019, doi: 10.1109/WACV.2019.00201.
- [21] P. Prakash, S. K. Y. Hanumanthaiah, and S. B. Mayigowda, “CRNN model for text detection and classification from natural scenes,” *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 13, no. 1, pp. 839–849, Mar. 2024, doi: 10.11591/ijai.v13.i1.pp839-849.
- [22] Y. Yu, J. Cao, and J. Zhu, “An LSTM short-term solar irradiance forecasting under complicated weather conditions,” *IEEE Access*, vol. 7, pp. 145651–145666, 2019, doi: 10.1109/ACCESS.2019.2946057.
- [23] V. S. Rathore, J. M. R. S. Tavares, V. Piuri, and B. Surendiran, “Emerging trends in expert applications and security,” *Proceedings of 2nd ICETEAS 2023*, vol. 841, Springer Singapore, 2023.
- [24] Y. Mintz and R. Brodie, “Introduction to artificial intelligence in medicine,” *Minimally Invasive Therapy & Allied Technologies*, vol. 28, no. 2, pp. 73–81, 2019, doi: 10.1080/13645706.2019.1575882.
- [25] P. Wang, “On defining artificial intelligence,” *Journal of Artificial General Intelligence*, vol. 10, no. 2, pp. 1–37, 2019, doi: 10.2478/jagi-2019-0002.
- [26] A. Holzinger, G. Langs, H. Denk, K. Zatloukal, and H. Müller, “Causability and explainability of artificial intelligence in medicine,” *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 9, no. 4, pp. 1–13, 2019, doi: 10.1002/widm.1312.
- [27] C. Zhang and Y. Lu, “Study on artificial intelligence: The state of the art and future prospects,” *Journal of Industrial Information Integration*, vol. 23, 2021, doi: 10.1016/j.jii.2021.100224.
- [28] S. Ghose, “Forecasting bitcoin price considering macro economic factors and media influence using bidirectional LSTM and random forest regressor as ensemble model,” *B.Sc. Thesis*, Department of Computer Science and Engineering, Brac University, Dhaka, Bangladesh, 2023.
- [29] L. S. Moonlight, B. R. Trilaksono, B. B. Harianto, and F. Faizah, “Implementation of recurrent neural network for the forecasting of USD buy rate against IDR,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 13, no. 4, pp. 4567–4581, 2023, doi: 10.11591/ijece.v13i4.pp4567-4581.
- [30] S. Dash and P. Parida, “Text recognition using CRNN models based on temporal classification and interpolation methods,” in *Machine Learning in Medical Imaging and Computer Vision*, 2023, doi: 10.1049/pbhe049e.
- [31] A. Boukhalfa, A. Abdellaoui, N. Hmina, and H. Chaoui, “LSTM deep learning method for network intrusion detection system,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 3, pp. 3315–3322, Jun. 2020, doi: 10.11591/ijece.v10i3.pp3315-3322.
- [32] U. Verma, P. Tyagi, and M. Kaur, “Single input single head CNN-GRU-LSTM architecture for recognition of human activities,” *Indonesian Journal of Electrical Engineering and Informatics (IJEI)*, vol. 10, no. 2, pp. 410–420, 2022, doi: 10.52549/ijece.v10i2.3475.
- [33] C. Tian, J. Ma, C. Zhang, and P. Zhan, “A deep neural network model for short-term load forecast based on long short-term memory network and convolutional neural network,” *Energies*, vol. 11, no. 12, 2018, doi: 10.3390/en1123493.
- [34] Y. Zheng, T. Wang, B. Xin, T. Xie, and Y. Wang, “A sparse autoencoder and softmax regression based diagnosis method for the attachment on the blades of marine current turbine,” *Sensors*, vol. 19, no. 4, 2019, doi: 10.3390/s19040826.
- [35] L. Munkhdalai, K. H. Ryu, O. E. Namsrai, and N. T. -Umpon, “A partially interpretable adaptive softmax regression for credit scoring,” *Applied Science*, vol. 11, no. 7, 2021, doi: 10.3390/app11073227.
- [36] S. A. E. -Mienye, E. Esenogho, and T. G. Swart, “Integrating enhanced sparse autoencoder-based artificial neural network technique and softmax regression for medical diagnosis,” *Electronics*, vol. 9, no. 11, pp. 1–13, 2020, doi: 10.3390/electronics9111963.
- [37] A. S. Gaafar, J. M. Dahr, and A. K. Hamoud, “Comparative analysis of performance of deep learning classification approach based on LSTM-RNN for textual and image datasets,” *Informatica*, vol. 46, no. 5, pp. 21–28, 2022, doi: 10.31449/inf.v46i5.3872.
- [38] C. Lin and M. Chi, “A comparisons of BKT, RNN and LSTM for learning gain prediction,” *Artificial Intelligence in Education: 18th International Conference, AIED 2017*, vol. 10331, pp. 536–539, 2017, doi: 10.1007/978-3-319-61425-0_58.
- [39] F. A. Al-azazi and M. Ghurab, “ANN-LSTM: A deep learning model for early student performance prediction in MOOC,” *Heliyon*, vol. 9, no. 4, 2023, doi: 10.1016/j.heliyon.2023.e15382.

BIOGRAPHIES OF AUTHORS






Tukino Paryono    is presently engaged in doctoral studies within the Computing Division of the Faculty of Information Technology at Universitas Kristen Satya Wacana, located in Salatiga, Indonesia. He attained both his undergraduate and graduate qualifications in information systems from Universitas Gunadarma, Indonesia. Presently, he serves as a faculty member in the Department of Information Systems at Universitas Buana Perjuangan in Karawang, Indonesia, specializing in machine learning and natural language processing. For further details or inquiries. He can be contacted at email: tukino@ubpkarawang.ac.id.






Eko Sedyono    is a professor at Universitas Kristen Satya Wacana in Salatiga, Indonesia, in the Division of Informatics Engineering within the FTI. The University of Indonesia is where he obtained his Master's and Doctorate degrees in computer science. Apart from his position at UKSW Salatiga, he is an assessor at BAN PT and LAM Infokom, and he is also actively involved in the Research, Innovation, and Entrepreneurship Division. His work primarily focuses on machine learning, artificial neural networks, and artificial intelligence. To communicate with him. He can be contacted at email: eko@uksw.edu.






Hendry    completed his undergraduate studies in 1983, earning a Bachelor of Science in Information Technology at the Technology School located in Surabaya, Indonesia. He furthered his studies at Institut Teknologi Sepuluh Nopember (ITS), Surabaya, achieving a Master of Science in Information Technology. In 2009, he achieved his Doctorate in Information Management with distinction from Chaoyang University of Technology (CYUT) located in Taichung, Taiwan. Between 2012 and 2014, he served as the director at the Business and Technology Incubator of Universitas Kristen Satya Wacana in Salatiga, Indonesia. Later, he spent time as a researcher after completing their doctoral studies in CYUT's Department of Information Management between 2018 and 2019. Currently, he is a lecturer at Universitas Kristen Satya Wacana, engaging in research related to knowledge engineering, domain ontology, recommendation systems, and the practical implementation of artificial intelligence. For communication purposes. He can be contacted at email: hendry@uksw.edu.






Baenil Huda    earned a Bachelor of Engineering (S.T) degree from STMIK Kharisma Karawang in the period 2000 to 2005. He then continued his studies and obtained Master of Computer (M.Kom.) degree from STMIK Likmi Bandung in the period 2012 to 2015. Currently, he serves as Lecturer in the Information Systems Study Program at UBP Karawang since 2015. Apart from that, he is also studying for a Doctoral Degree in Computer Science at SWCU Salatiga from 2023. Given his extensive educational background and professional experience, he is actively engaged in the realms of education and research within the domain of Information Systems and Computer Science. He can be contacted at email: baenil88@ubpkarawang.ac.id.



April Lia Hananto    having obtained his Doctor of Computer Science degree from Universiti Teknologi Malaysia in 2022, he possesses a strong educational foundation. He earned his bachelor's degree from STMIK Rosma in 2005 and subsequently obtained his M.Kom. degree from Sekolah Tinggi Teknik Informatika Benarif Indonesia in 2007. Currently, he holds the position of associate professor in the Information Science Department at Universitas Buana Perjuangan Karawang, Indonesia. He is interested in research areas such as computer vision, machine learning, data mining, and information systems. He has contributed significantly to his field. He has written more than 50 papers that have been published internationally in journals and at conferences July 2005 to December 2023. For more information. He can be contacted at email: apriliah@ubpkarawang.ac.id.



Aviv Yuniar Rahman    a distinguished scholar, completed his Bachelor of Science in Electrical Engineering, specializing in Computer Engineering and Telematics at ITS Surabaya in 2013. He furthered his education at the same institution, obtaining a Master's in Electrical Engineering, focusing on Intelligent Multimedia Networks, in 2017. Since 2014, he has consistently served as a committed member of the Informatics Engineering Department at Universitas Widyagama Malang. Concurrently, he began pursuing his Ph.D. in ICT at AeU Malaysia in 2022. He boasts an extensive portfolio, having authored and published more than 70 articles, making significant contributions to his field. He can be contacted at email: aviv@widyagama.ac.id.