JOIV : Int. J. Inform. Visualization, 8(1) - March 2024 231-238

INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION

journal homepage : www.joiv.org/index.php/joiv

Comparison of Adam Optimization and RMSprop in Minangkabau-Indonesian Bidirectional Translation with Neural Machine Translation

Fadhli Almu'iini Ahda [a,b,*], Aji Prasetya Wibawa [a], Didik Dwi Prasetya [a], Danang Arbian Sulistyo [a,b]

[a] *Electrical Engineering and Informatics, Universitas Negeri Malang, Jl. Semarang No.5, Malang, 65145, Indonesia*
[b] *Institut Teknologi dan Bisnis Asia Malang, Rembuksari No.1A, Malang, Indonesia*
*Corresponding author: *adhi32286@gmail.com*

***Abstract*— Language is a tool humans use to establish communication. Still, the language used is one language and between regions or nations with their languages. Indonesia is a country that has a diversity of second languages and is the fourth most populous country in the world. It is recorded that Indonesia has nearly 800 regional languages, but research activities in natural language processing are still lacking. Minangkabau is an endangered language spoken by the Minangkabau people in Indonesia's West Sumatra province. According to UNESCO, the Minangkabau language is listed as a language that is "definitely endangered," with only around 5 million speakers worldwide. This study uses neural machine translation (NMT) to create a formula based on this information. Neural machine translation, in contrast to conventional statistical machine translation, intends to build a single neural network that can be built up to achieve the best performance. Because it can simultaneously hold memory for a long time, comprehend complicated relationships in data, and provide information that is very important in determining the outcome of translation, LSTM is one of the most powerful machine-learning techniques for translating languages. The BLUE score is utilized in the NMT evaluation. The test results use 520 Minangkabau sentences, conducting tests based on the number of epochs ranging from 100-1000, resulting in optimization using Adam being better than optimization RMSprop. This is evidenced by the results of the best BLUE-1 score of 0.997816 using 1000 epochs.**

***Keywords*— LSTM; machine translation; optimization; BLUE.**

Manuscript received 8 May. 2023; revised 3 Jul. 2023; accepted 2 Oct. 2023. Date of publication 31 Mar. 2024.

I. INTRODUCTION

Indonesia is a country that has a diversity of second languages and is the fourth most populous country in the world. It is recorded that Indonesia has almost 800 regional languages, but research activities in natural language processing still need to be improved [1]. It is time to build a local Indonesian natural language processing dataset, first focusing on developing a framework and creating, collecting, and classifying datasets reflecting this situation [2]**.** The top 10 regional languages in Indonesia can be seen in Figure 1. Minangkabau is an endangered language spoken by the Minangkabau people in the West Sumatra province of Indonesia. According to UNESCO, Minangkabau is listed as a "definitely endangered" language, with only around 5 million speakers worldwide [3].

Neural machine translation is the newest language translation method widely applied worldwide [4]. This technique allows machines to learn from source and target language data to produce precise and accurate translations. In language translation, optimization is one of the critical factors affecting the performance of the neural engine. Several optimization techniques are used in neural machine training, including RMSProp and Adam optimization [5].

In translating Minangkabau into Indonesian, few studies have compared the effectiveness of RMSProp and Adam optimization [6]. Therefore, this study aims to compare the performance of RMSProp and Adam optimization in neural machine translation from Minangkabau to Indonesian. In this study, we will use existing translation data to train a neural machine translation model and then compare the model's performance using RMSProp and Adam optimization [7]. We will analyze the results of both optimization techniques and evaluate translation performance based on standard evaluation metrics [8]. This research is expected to understand better the use of RMSProp and Adam optimization in translating Minangkabau into Indonesian. The results of this study can help develop more effective and efficient neural machine translation models in the future [9].

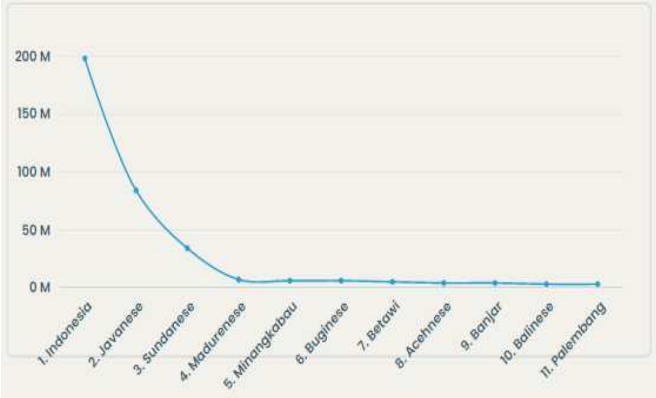

Fig. 1 The top 10 regional languages in Indonesia

Various other languages, including Malay, Javanese, and Arabic, have influenced the Minangkabau language. It is a unique language that uses a matrilineal system of inheritance and is often called the "world's largest matrilineal society" [10]. Several factors have contributed to the endangerment of the Minangkabau language. One of the main reasons is the increasing use of Indonesian as the national language of Indonesia, which has resulted in a decline in the use of regional languages like Minangkabau. Additionally, there has been a lack of support for the preservation and promotion of Minangkabau, including a lack of language education resources [3], [11].

II. Material and Methods

The first stage in this research is data collection. After the data is collected, the next step is modeling, data processing, separation, and evaluation.

A. Data Collection

The data used is Minangkabau and Indonesian language corpus data in .txt format, accessed on data. The data used is 520 Minangkabau and Indonesian language sentences in rhymes. The last remaining step for the Minang-Indonesian dataset is to reverse the dataset, which can be seen in Table I.

TABLE I
An example of the dataset used Indonesia-Minang

Indonesian	Minangkabau
pagi-pagi menuai padi	sajak pagi manuai padi
selesai sudah pukul satu	sudahnyo lah pukua satu
yang masak saja yang dituai	nan masak sajo nan dituai
saya menangis dalam hati	denai manangih dalam hati
siapalah orang yang akan tahu	sialah urang nan katau
jika tidak orang yang merasai	jikok ndak urang nan marasai

B. Neural Machine Translation

Since the early 2000s, neural machine translation (NMT) has grown significantly and matured. The most popular machine translation method still needs to perform better on language pairs with low resources compared to their high-resource counterparts because no massive parallel corpora are available [11], [12]. Despite the neural machine translation (NMT) success in performance testing, the absence of significant parallel corpora is a practical challenge for many language combinations [14]. Many solutions have been proposed to address this problem, such as triangulation and semi-supervised learning approaches, but they still need a potent cross-lingual signal [15], [16].

C. Preprocessing

This section removes all unnecessary characters in compiling the dataset, from punctuation to symbols [17]. Table 2 shows that the process starts by loading the data-separated dataset. After the refinement process, proceed with compiling and defining the regex of character sequences based on the dataset [18]. The process is continued by preparing the translation table to remove all readable punctuation marks in the dataset. The process includes normalizing Unicode characters, creating tokens for spaces, converting the dataset into lowercase, removing unnecessary characters and numbers, and ending with saving a list of clean sentences [19]. Then, the following process is to display the dataset in separate forms in Minang and Indonesian, and the last is to clean up the sentence again and save the dataset [20].

TABLE II
Pseudocode for preprocessing

```
Pseudocode for the preprocessing process using a Python script
for cleaning datasets.
    load doc into memory
    , and split a loaded document into sentences.
    clean a list of line
        prepare regex for char filtering.
        prepare a translation table for removing punctuation.
            make Unicode characters more readable.
            tokenize using blank spaces.
            lowercase the letters.
            Take the punctuation out of every token.
            Take the non-printable characters out of
each token.
            Remove all tokens that contain numerals.
    save a list of clean sentence to file
    load dataset
    split into Minang-Indonesia
    clean sentences
    save clean pairs to file
    spot check
```

D. LSTM (Long Short-Term Memory)

In this study, cell A was given to typical LSTM cells. Because of its superior performance, LSTM is a widely utilized RNN in the learning domain. Memory modules are another name for LSTM hidden layer modules [21], [22]. It has three gates that handle reading, storing, and writing accordingly. These are input, forgetting, and output gates [23], [24]. The three gates are valve-like devices that open and close to convey data to the neurons. They establish how much information is used to calculate a neuron's currents and how much is transferred to the cell after it [24]. Figure 2 represents the LSTM's internal architecture.

1) The LSTM model's function, as indicated in Figure 2, is a logistic and unique function. It is an activation function to enable the LSTM model's nonlinearity. The input and candidate gates create a new cell, Ct, which is then sent to the next stage as an updated cell. The input candidate gate utilizes the hyperbolic tangent of the function (mt) as the activation function, while the input gate uses the sigmoid function (it) as the activation function. Another unique feature of the LSTM is the tan function, indicated by "tanh" in Figure 2.

2) *The forgetting vector*: The value ranges from 0 to 1, produced by the forgetting gate. The ft function operates as forgetting multiplied by the previous cell value [25].

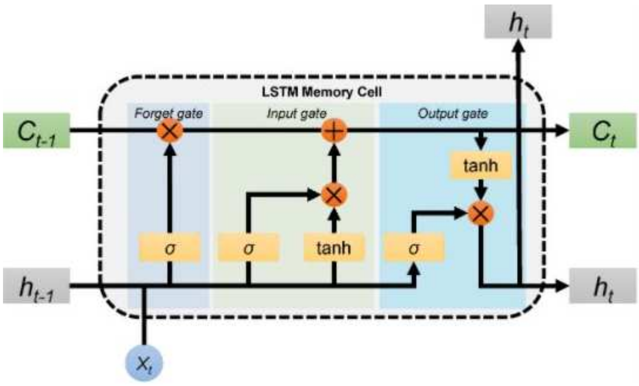

Fig. 2 Basic mechanism work of LSTM [26]

The following is the LSTM (Long Short-Term Memory) formula that is commonly used in artificial neural networks [27]

1. The forget gate is defined as

$$f\_t = (W\_f. [h\_t\text{-}1, x\_t] + b\_f) \quad (1)$$

2. Input Gate:

$$i\_t = (W\_i. [h\_t\text{-}1, x\_t] + b\_i) \quad (2)$$

The formula for _t is tanh

$$(W\_c. [h\_t\text{-}1, x\_t] + b\_c) \quad (3)$$

3. Change the cell as follows:

$$C\_t = f\_t * C\_t\text{-}1 + i\_t * \_t \quad (4)$$

4. The output gate is:

$$o\_t = (W\_o. [h\_t\text{-}1, x\_t] + b\_o)$$
$$o\_t * \tanh (C\_t) = h\_t \quad (5)$$

In the modeling methodology in Figure 3, there are five steps involved in the optimization process.

1) *Preparing Data*: The first input was the Minang-Indonesia and Indonesia-Minang corpus datasets during this process. It was then preprocessing, where the dataset was cleaned of punctuation so that the normalization process could occur.

2) *Generating a List Containing Combination* Parameter: The following process provides a list of parameter combinations to measure datasets: providing hidden layers, number of neurons, learning rate, and number of epochs.

3) *Transforming Data*: In grouping datasets, how many will be used as training data, and how many will be utilized as testing data.

4) *Training Models*: This process is the essence of the problem, how to choose the model to be used. The selected model is The LSTM model, which has two optimization models: Adam optimization and RMSprop.

5) *Selecting the Best Models*: This process is the final process in research where the results and predictions are already visible as a BLUE Score so that the best model can be selected.

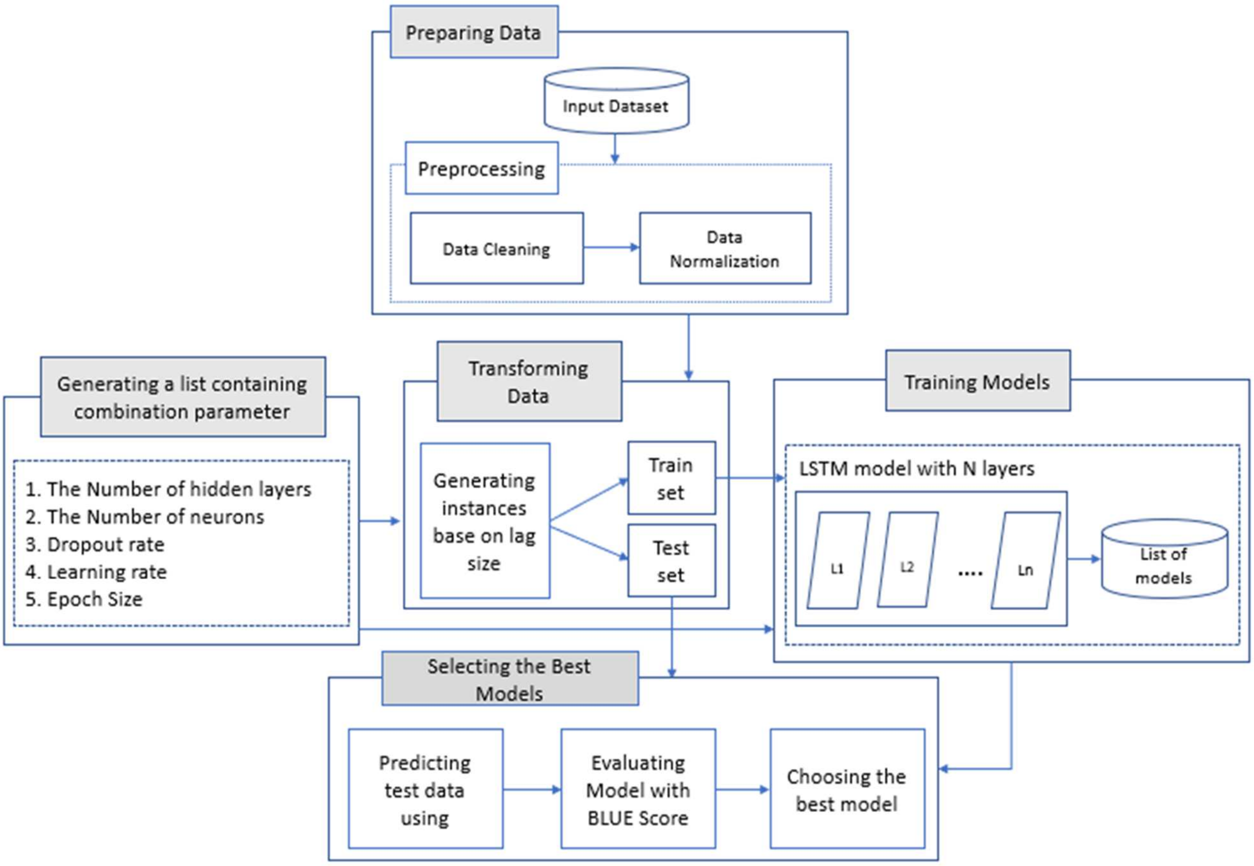

Fig. 3 The proposed methodology for Minangkabau into Indonesian bidirectional translation using LSTM

E. RMSprop Optimization

The stochastic gradient descent algorithm RMSprop (Root Mean Square Propagation) modifies the learning rate for each weight based on the average magnitude of recent gradients for that weight[28] [29]. It uses an exponentially decaying average to calculate this magnitude, giving more weight to recent gradients than older ones. This helps prevent the learning rate from oscillating too much and enables faster convergence to the optimal solution. After calculating the gradient, the squared gradient is accumulated to apply RMSprop: [30]

$$r \leftarrow \rho r + (1 - \rho) g \circledS g \tag{6}$$

where the rate of deterioration is. The following is how the parameter update is calculated and used:

$$\Delta\theta = -\frac{\epsilon}{\sqrt{\delta + r}} \circledS g \tag{7}$$

$$\theta \leftarrow \theta + \Delta\theta \tag{8}$$

F. Adam Optimization

Adam (Adaptive Moment Estimation) optimization is an extension of RMSProp that includes a momentum term [31]. The momentum term helps the optimizer to continue moving in the same direction as previous gradients, which can help overcome areas of low gradient magnitude or saddle points. Additionally, Adam uses bias-correction terms to adjust the estimates of the first and second moments of the gradient, which can help with convergence when the gradients are sparse [32]. The Adam method needs the first and second instant variables m and u. Biased first- and second-moment estimates are updated at time step t following gradient computation, respectively: [30]

$$m_t \leftarrow \rho_1 m_{t-1} + (1 - \rho_1) g_t \tag{9}$$

$$u_t \leftarrow \rho_2 u_{t-1} + (1 - \rho_2) g \circledS g \tag{10}$$

The bias is then adjusted in the first and second moments after that. Updates to parameters are calculated and made using the updated moment estimates:

$$\theta_t \leftarrow \theta_{t-1} + \Delta\theta \tag{11}$$

Adam has many benefits. It first needs to be slightly tuned for the learning rate. It is also an easy strategy to use and is not affected by the diagonal scaling of gradients [33]. It requires less memory and is highly computationally efficient. Additionally, Adam is suitable for non-stationary goals and issues with extremely noisy and sparse gradients [34].

G. K-Fold Cross Validation

K-fold cross-validation is a machine learning method for assessing a model's performance on a given dataset. The dataset is split up into k subsets, or nearly equal-sized folds. The model is then tested on the final fold after being trained on k-1 folds. Each fold is the test set once during this process, which is repeated k times [35]. The k-fold results are averaged to obtain a final performance statistic. This method helps to reduce the variance of the estimated performance compared to a simple train-test split, as all data points are used for training and testing [36]. K-fold cross-validation is often used to tune hyperparameters of a model, such as the regularization parameter, by evaluating the performance of the validation set. It can also help detect overfitting by providing a more robust estimate of the model's generalization performance.

III. Results and Discussion

By using a dataset that is still limited, namely 520 sentences from Minang to Indonesian, and vice versa. Figure 4 and Figure 5 are the results of optimization testing using LSTM.

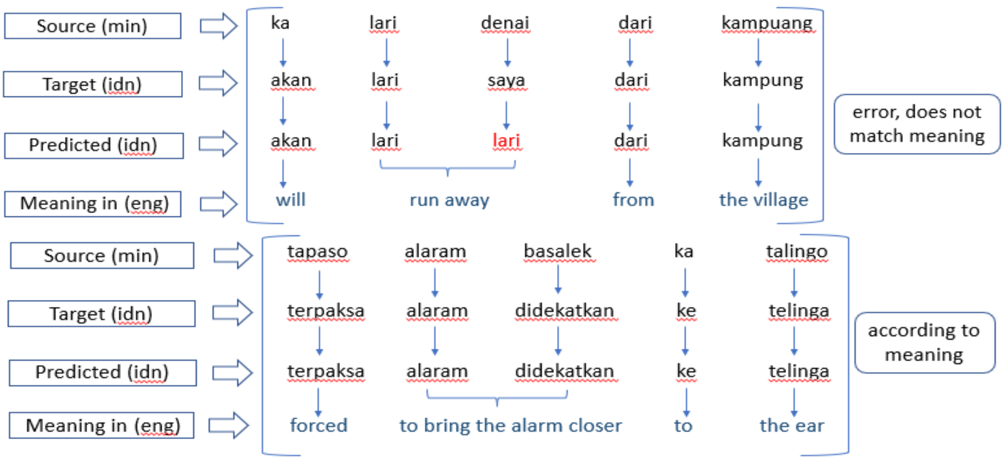

Fig. 4 Minang - Indonesian translation test results

In Figure 4 for the first Minang-Indonesian translation, there was an error in translation where it should have been the target sentence (I will run away from the village) with the meaning "I will leave this village," but in the predicted results, the sentence became (I will run away from the village) with the meaning "I will run around the village" so that it does not match the meaning of the translated sentence. The second experiment, starting from the target and prediction, follows the meaning of the desired translation sentence.

Fig. 5 Indonesian - Minang translation test results

In Figure 5 for the reverse translation, Indonesian-Minang, two experimental examples are shown with the same sentence from the previous experiment. As seen in the picture above, experiments 1 and 2 of the target and prediction of the translated sentence all match the actual meaning of the sentence. So, from the test results of the two translations, there are differences in the results, where the results of the Indonesian-Minang translation test get better results than the Minang-Indonesian translation test.

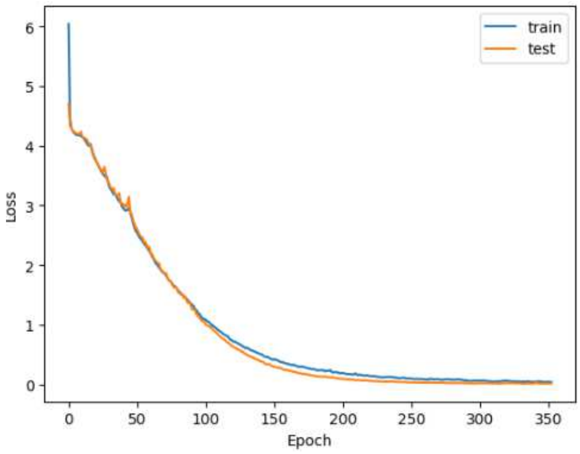

Fig. 6 Minang-Indonesia with Optimization Adam

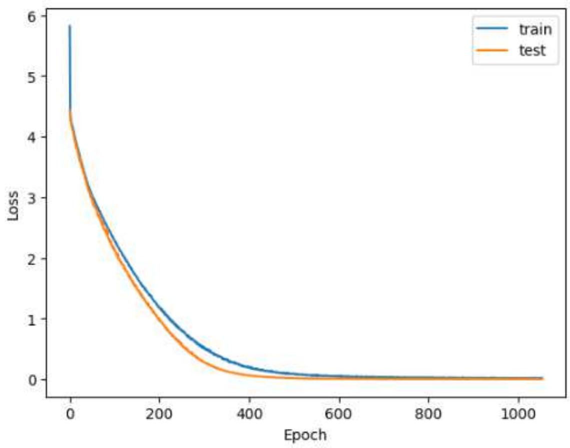

Fig. 7 Minang-Indonesia with Optimization RMSprop

From the experimental results in Figures 6 and 7 for the Minang-Indonesian translation, it can be seen that Adam's optimization only requires 358 epochs to display the test results graph. In comparison, RMSprop's optimization requires 1122 epochs. Adam's optimization testing results display a more accurate graph than the RMSprop optimization.

TABLE III
MINANG-INDONESIAN OPTIMIZATION TEST RESULTS

Fold	Adam	RMSProp
	BLUE -1	BLUE -1
1	0.995575	1
2	1	0.995614
3	0.995614	1
4	1	1
5	1	1
6	1	0.995708
7	1	1
8	0.995614	0.995434
9	1	0.995745
10	1	0.995495
Average	**0.9986803***	**0.9977996**

In this test, the k-fold cross-validation model was carried out by experimenting with split data with a ratio of 90:10 [37]. This means using 90% of the dataset for training and 10% of the dataset for testing. They performed ten times with split data from a total of 520 datasets. The k-fold test for Minang-Indonesian translation was performed ten times for Blue Score-1 using Adam Optimization and RMSprop. Obtaining test results using Adam is better than RMSprop optimization, with an average BLUE Score of 0.9983023 for Adam optimization and 0.9982868 for RMSprop optimization. The difference between the BLUE Score values between optimizations is 0.0008807.

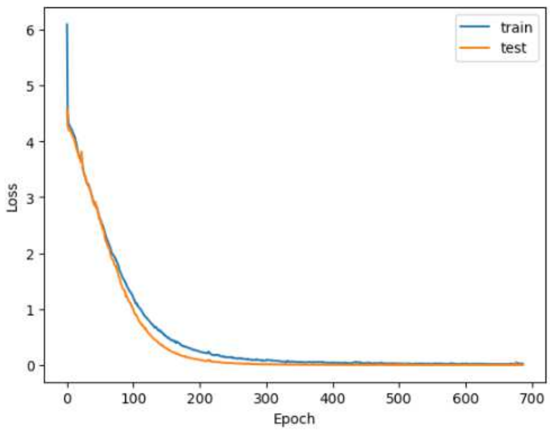

Fig. 8 Indonesia-Minang with Optimization Adam

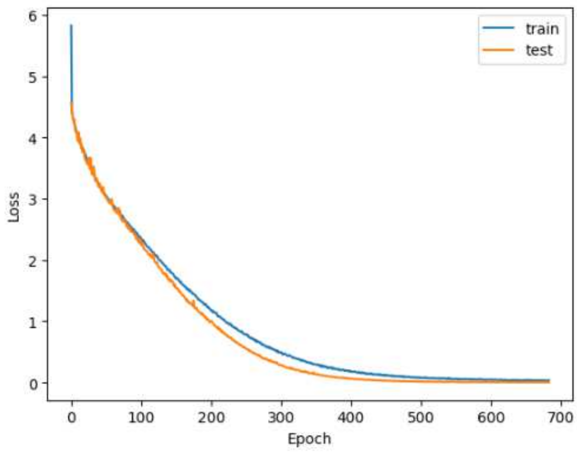

Fig. 9 Indonesia-Minang with Optimization RMSprop

From the results of the testing experiments in Figures 8 and 9 for the Indonesian-Minang translation, the Epoch results are similar between Adam's optimization and RMSprop 689 and 673 Epoch. However, Adam's optimization is still superior to RMSprop's for translation accuracy.

TABLE IV
INDONESIAN-MINANG OPTIMIZATION TEST RESULTS

Fold	Adam	RMSProp
	BLUE -1	**BLUE -1**
1	1	0.995763
2	0.987333	1
3	1	0.995726
4	1	1
5	1	1
6	1	1
7	1	0.991379
8	0.99569	1
9	1	1
10	1	1
Average	**0.9983023***	**0.9982868**

As for the fold test for Indonesian-Minang translation, the same process was carried out with ten tests for BLUE Score 1, obtaining test results of 0.9983023 for Adam optimization and 0.9982868 for RMSprop optimization. So, the difference between optimizations is 0.0000155, assuming that Adam's optimization is still better than RMSprop optimization in this case study.

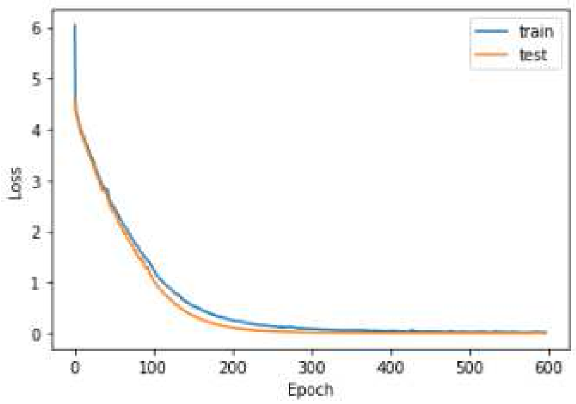

Fig. 10 Adam Optimization Result Graph

The graphical results using Adam optimization show that the training and testing lines look almost perfectly integrated, which means the results are close to 1. It can also be seen that testing and training iterations require 598 epochs with 128 layers, which only takes 317 seconds.

TABLE V
ADAM OPTIMIZATION BLUE RESULTS

Epoch	BLUE-1	BLUE-2	BLUE-3	BLUE-4
100	0.708150	0.599884	0.543484	0.423315
200	0.990834	0.985237	0.981014	0.960261
300	0.996942	0.995424	0.994433	0.979995
400	0.996943	0.995363	0.994093	0.979148
500	0.997380	0.996147	0.994803	0.980072
600	0.996942	0.995424	0.994433	0.979995
700	0.997380	0.995581	0.994223	0.979256
800	0.997379	0.996209	0.995144	0.980920
900	0.996943	0.995363	0.994093	0.979148
1000	**0.997816***	**0.996710**	**0.995686**	**0.981704**

The table above describes the results of 10 experiments for Adam optimization using a comparison of the number of epochs ranging from 100-1000. This experiment finds the best optimization regarding the time used to make predictions. Because in optimization, what is seen is good accuracy and the length of time used in optimization [38]. Table V above shows Adam's optimization's best accuracy results in the last test with a BLUE-1 value of 0.997816 with several epochs of 1000. From the graphical results in Figure 11 using RMSProp optimization, the training and testing lines show that there is still a distance between the blue and orange lines. At the same time, the looping time takes longer than Adam's optimization, namely 822 epochs with 431 seconds.

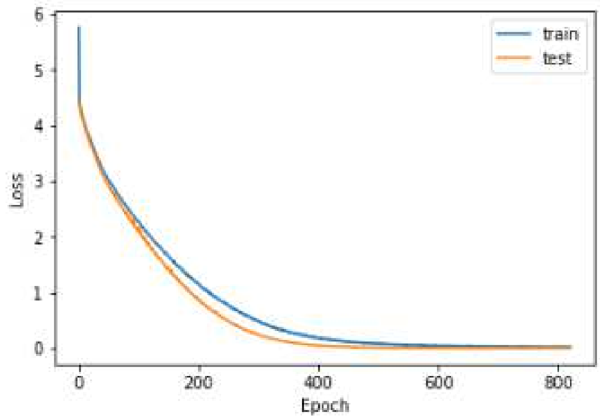

Fig. 11 RMSprop Optimization Result Graph

TABLE VI
RMSPROP OPTIMIZATION BLUE RESULTS

Epoch	BLUE-1	BLUE-2	BLUE-3	BLUE-4
100	0.278127	0.154305	0.126766	0.078917
200	0.787241	0.715970	0.679447	0.585099
300	0.987336	0.981788	0.978701	0.959029
400	0.996507	0.994296	0.992730	0.977350
500	0.997379	**0.996209**	0.994903	0.980381
600	**0.997380***	0.996147	0.994803	0.979732
700	0.996943	0.995363	0.994093	0.979148
800	**0.997380***	0.996147	0.994803	0.980072
900	0.996942	0.995424	0.994433	0.979995
1000	0.997379	0.995926	**0.994975**	**0.980780**

The test results using the RMSprop optimization are shown in Table VI above by using a test based on the number of epochs from 10 times of testing, the results for BLUE Score-1, the best score occurred in experiments 6 and 8 with the same value, namely 0.997380.

TABLE VII
ACCURACY RESULTS IN ADAM AND RMS PROB

Accuracy Results		
Evaluation	**Adam**	**RMSProp**
BLUE-1	**0.997816***	**0.997380**
BLUE-2	0.996710	0.996209
BLUE-3	0.995686	0.994975
BLUE-4	0.981704	0.980780

From the results of the BLUE score, the accuracy for optimization using Adam is still better than optimization using RMSprop. It is proven that the highest BLUE values are at BLUE Score 1, namely 0.997816 and 0.997380, respectively.

IV. CONCLUSION

Based on the study's results, it can be concluded that Adam's optimization is more effective than RMSProp in translating the neural machine from Minangkabau to Indonesian and vice versa. Although both optimization techniques give relatively good results, Adam's optimization performs better in achieving convergence and accelerating the training model.

To further the progress of Natural Language Processing (NLP), developing an extensive bilingual corpus encompassing the Minangkabau and Indonesian languages presents a substantial opportunity. This highly significant resource is positioned to function as the fundamental knowledge for advancing comprehensive Minangkabau genealogy techniques. With the foresight of the potential of this research, it is envisaged that the next endeavors would result in significant enhancements in the performance of machine translation alignment. Moreover, it is anticipated that these technological improvements will considerably decrease the amount of data storage required, enhancing machine translation systems' effectiveness.

The following research endeavors to advance the current state of Natural Language Processing (NLP) and has the potential to bring about a significant transformation in the area. Additionally, these initiatives will create opportunities for the development of innovative applications in the domain of machine translation. The development, as mentioned earlier, will introduce a novel phase of stemming methodologies specifically designed to meet the requirements of the continuously changing linguistic environment, addressing the necessities of forthcoming language processing advancements.

REFERENCES

[1] A. F. Aji *et al.*, "One Country, 700+ Languages: NLP Challenges for Underrepresented Languages and Dialects in Indonesia." arXiv, Mar. 24, 2022. Accessed: Oct. 13, 2022. [Online]. Available: http://arxiv.org/abs/2203.13357

[2] M. Amien, C. Feng, and H. Huang, "Location-based Twitter Filtering for the Creation of Low-Resource Language Datasets in Indonesian Local Languages." arXiv, Jun. 14, 2022. Accessed: Oct. 13, 2022. [Online]. Available: http://arxiv.org/abs/2206.07238

[3] D. Bradley and M. Bradley, *Language Endangerment*. Cambridge University Press, 2019.

[4] M.-T. Luong, H. Pham, and C. D. Manning, "Effective Approaches to Attention-based Neural Machine Translation." arXiv, Sep. 20, 2015. doi: 10.48550/arXiv.1508.04025.

[5] S. H. Haji and A. M. Abdulazeez, "Comparison of Optimization Techniques Based on Gradient Descent Algorithm: A Review," *PalArchs J. Archaeol. Egypt Egyptol.*, vol. 18, no. 4, Art. no. 4, Feb. 2021.

[6] G. Singh, S. Sharma, V. Kumar, M. Kaur, M. Baz, and M. Masud, "Spoken Language Identification Using Deep Learning," Computational Intelligence and Neuroscience, vol. 2021, pp. 1–12, Sep. 2021, doi: 10.1155/2021/5123671.

[7] D. K. R. Gaddam, M. D. Ansari, S. Vuppala, V. K. Gunjan, and M. M. Sati, "A Performance Comparison of Optimization Algorithms on a Generated Dataset," ICDSMLA 2020, pp. 1407–1415, Nov. 2021, doi:10.1007/978-981-16-3690-5_135.

[8] J. Wieting, T. Berg-Kirkpatrick, K. Gimpel, and G. Neubig, "Beyond BLEU: Training Neural Machine Translation with Semantic Similarity." arXiv, Sep. 14, 2019. doi: 10.48550/arXiv.1909.06694.

[9] S. A. Mohamed, A. A. Elsayed, Y. F. Hassan, and M. A. Abdou, "Neural machine translation: past, present, and future," Neural Computing and Applications, vol. 33, no. 23, pp. 15919–15931, Jul. 2021, doi: 10.1007/s00521-021-06268-0.

[10] C. Mahfud, R. Astari, A. Kasdi, M. A. Mu'ammar, M. Muyasaroh, and F. Wajdi, "Islamic cultural and Arabic linguistic influence on the languages of Nusantara; From lexical borrowing to localized Islamic lifestyles," Wacana, vol. 22, no. 1, p. 224, May 2021, doi: 10.17510/wacana.v22i1.914.

[11] J. Santoso, E. I. Setiawan, F. X. Ferdinandus, G. Gunawan, and L. Hernandez, "Indonesian Language Term Extraction using Multi-Task Neural Network," Knowledge Engineering and Data Science, vol. 5, no. 2, p. 160, Dec. 2022, doi: 10.17977/um018v5i22022p160-167.

[12] S. Ranathunga, E.-S. A. Lee, M. Prifti Skenduli, R. Shekhar, M. Alam, and R. Kaur, "Neural Machine Translation for Low-resource Languages: A Survey," ACM Computing Surveys, vol. 55, no. 11, pp. 1–37, Feb. 2023, doi: 10.1145/3567592.

[13] "Natural Language Processing in Higher Education | Bulletin of Social Informatics Theory and Application." Accessed: Oct. 03, 2023. [Online]. Available: https://pubs.ascee.org/index.php/businta/article/view/593

[14] Md. A. Islam, Md. S. H. Anik, and A. B. M. A. A. Islam, "Towards achieving a delicate blending between rule-based translator and neural machine translator," Neural Computing and Applications, vol. 33, no. 18, pp. 12141–12167, Mar. 2021, doi: 10.1007/s00521-021-05895-x.

[15] M. Artetxe, G. Labaka, E. Agirre, and K. Cho, "Unsupervised Neural Machine Translation." arXiv, Feb. 26, 2018. doi:10.48550/arXiv.1710.11041.

[16] J. Santoso, E. I. Setiawan, C. N. Purwanto, and F. Kurniawan, "Indonesian Sentence Boundary Detection using Deep Learning Approaches," Knowledge Engineering and Data Science, vol. 4, no. 1, p. 38, Aug. 2021, doi: 10.17977/um018v4i12021p38-48.

[17] Z. Jianqiang and G. Xiaolin, "Comparison Research on Text Pre-processing Methods on Twitter Sentiment Analysis," IEEE Access, vol. 5, pp. 2870–2879, 2017, doi: 10.1109/access.2017.2672677.

[18] A. P. Wibawa, H. K. Fithri, I. A. E. Zaeni, and A. Nafalski, "Generating Javanese Stopwords List using K-means Clustering Algorithm," Knowledge Engineering and Data Science, vol. 3, no. 2, p. 106, Dec. 2020, doi: 10.17977/um018v3i22020p106-111.

[19] Rianto, A. B. Mutiara, E. P. Wibowo, and P. I. Santosa, "Improving the accuracy of text classification using stemming method, a case of non-formal Indonesian conversation," Journal of Big Data, vol. 8, no. 1, Jan. 2021, doi: 10.1186/s40537-021-00413-1.

[20] S. M. Tauhid and Y. Ruldeviyani, "Sentiment Analysis of Indonesians Response to Influencer in Social Media," 2020 7th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE), Sep. 2020, doi:10.1109/icitacee50144.2020.9239218.

[21] Le, Ho, Lee, and Jung, "Application of Long Short-Term Memory (LSTM) Neural Network for Flood Forecasting," Water, vol. 11, no. 7, p. 1387, Jul. 2019, doi: 10.3390/w11071387.

[22] D. A. Sulistyo, "LSTM-Based Machine Translation for Madurese-Indonesian," Journal of Applied Data Sciences, vol. 4, no. 3, pp. 189–199, Sep. 2023, doi: 10.47738/jads.v4i3.113.

[23] D. Janardhanan and E. Barrett, "CPU workload forecasting of machines in data centers using LSTM recurrent neural networks and ARIMA models," 2017 12th International Conference for Internet Technology and Secured Transactions (ICITST), Dec. 2017, doi:10.23919/icitst.2017.8356346.

[24] Y. Deng, H. Fan, and S. Wu, "A hybrid ARIMA-LSTM model optimized by BP in the forecast of outpatient visits," Journal of Ambient Intelligence and Humanized Computing, vol. 14, no. 5, pp. 5517–5527, Oct. 2020, doi: 10.1007/s12652-020-02602-x.

[25] W. Lu, J. Li, Y. Li, A. Sun, and J. Wang, "A CNN-LSTM-Based Model to Forecast Stock Prices," Complexity, vol. 2020, pp. 1–10, Nov. 2020, doi: 10.1155/2020/6622927.

[26] J. Long and K. Furati, *Identification and prediction of time-varying parameters of COVID-19 model: a data-driven deep learning approach*. 2021.

[27] P. Kumari and D. Toshniwal, "Long short term memory–convolutional neural network based deep hybrid approach for solar irradiance forecasting," Applied Energy, vol. 295, p. 117061, Aug. 2021, doi:10.1016/j.apenergy.2021.117061.

[28] W. Wang, C. M. Lee, J. Liu, T. Colakoglu, and W. Peng, "An empirical study of cyclical learning rate on neural machine translation," Natural Language Engineering, vol. 29, no. 2, pp. 316–336, Feb. 2022, doi:10.1017/s135132492200002x.

[29] P. Nimmani, S. Vodithala, and V. Polepally, "Neural Network Based Integrated Model for Information Retrieval," 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS), May 2021, doi: 10.1109/iciccs51141.2021.9432241.

[30] D. Soydaner, "A Comparison of Optimization Algorithms for Deep Learning," International Journal of Pattern Recognition and Artificial Intelligence, vol. 34, no. 13, p. 2052013, Apr. 2020, doi:10.1142/s0218001420520138.

[31] E. Okewu, S. Misra, and F.-S. Lius, "Parameter Tuning Using Adaptive Moment Estimation in Deep Learning Neural Networks," Lecture Notes in Computer Science, pp. 261–272, 2020, doi: 10.1007/978-3-030-58817-5_20.

[32] J. S. John, "AdamD: Improved bias-correction in Adam." arXiv, Oct. 22, 2021. doi: 10.48550/arXiv.2110.10828.

[33] A. Pranolo, Y. Mao, A. P. Wibawa, A. B. P. Utama, and F. A. Dwiyanto, "Optimized Three Deep Learning Models Based-PSO Hyperparameters for Beijing PM2.5 Prediction," KnowledgeEngineering and Data Science, vol. 5, no. 1, p. 53, Jun. 2022, doi: 10.17977/um018v5i12022p53-66.

[34] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization." arXiv, Jan. 29, 2017. doi: 10.48550/arXiv.1412.6980.

[35] Z. Xiong, Y. Cui, Z. Liu, Y. Zhao, M. Hu, and J. Hu, "Evaluating explorative prediction power of machine learning algorithms for materials discovery using k-fold forward cross-validation," *Comput. Mater. Sci.*, vol. 171, p. 109203, Jan. 2020, doi:10.1016/j.commatsci.2019.109203.

[36] J. J. Salazar, L. Garland, J. Ochoa, and M. J. Pyrcz, "Fair train-test split in machine learning: Mitigating spatial autocorrelation for improved prediction accuracy," Journal of Petroleum Science and Engineering, vol. 209, p. 109885, Feb. 2022, doi: 10.1016/j.petrol.2021.109885.

[37] S. Saud, B. Jamil, Y. Upadhyay, and K. Irshad, "Performance improvement of empirical models for estimation of global solar radiation in India: A k-fold cross-validation approach," Sustainable Energy Technologies and Assessments, vol. 40, p. 100768, Aug. 2020, doi: 10.1016/j.seta.2020.100768.

[38] G. I. Diaz, A. Fokoue-Nkoutche, G. Nannicini, and H. Samulowitz, "An effective algorithm for hyperparameter optimization of neural networks," IBM Journal of Research and Development, vol. 61, no. 4/5, pp. 9:1-9:11, Jul. 2017, doi: 10.1147/jrd.2017.2709578.