ILKOM Jurnal Ilmiah Vol. 16, No. 3, December 2024, pp. 271-282
Accredited 2nd by RISTEKBRIN No. 200/M/KPT/2020; E-ISSN 2548-7779 | P-ISSN 2087-1716

Research Article

Refining the Performance of Indonesian-Javanese Bilingual Neural Machine Translation Using Adam Optimizer

Fadia Irsania Putri [a,1]; Aji Prasetya Wibawa[a,2,*]; Leonel Hernandez Collante[b,3]
[a] *Electrical Engineering and Informatics, Engineering Faculty, Universitas Negeri Malang, Indonesia*
Institución Universitaria de Barranquilla IUB, Cl. 18 #39-100, Soledad, Atlántico, Colombia
[1] *fadia.irsania.2105356@students.um.ac.id;* [2]*aji.prasetya.ft@um.ac.id;* [3]*lhernandezc@itsa.edu.co*
** Corresponding author*

Article history: Received October 23, 2024; Revised November 07, 2024; Accepted December 10, 2024; Available online December 12, 2024

Abstract

This study focuses on creating a Neural Machine Translation (NMT) model for Indonesian and Javanese languages using Long Short-Term Memory (LSTM) architecture. The dataset was sourced from online platforms, containing pairs of parallel sentences in both languages. Training was performed with the Adam optimizer, and its effectiveness was compared to machine translation (MT) conducted without an optimizer. The Adam optimizer was utilized to enhance the convergence speed and stabilize the model by dynamically adjusting the learning rate. Model performance was assessed using BLEU (Bilingual Evaluation Understudy) scores to evaluate translation accuracy across different training epochs. The findings reveal that employing the Adam optimizer led to a significant enhancement in model performance. At epoch 2000, the model using the Adam optimizer achieved the highest BLEU score of 0.989957, reflecting very accurate translations, whereas the model without the optimizer showed lower results. Furthermore, translations from Indonesian to Javanese were found to be more precise than those from Javanese to Indonesian, largely due to the intricate structure and varying speech levels of the Javanese language. In summary, the implementation of the LSTM method with the Adam optimizer significantly improved the accuracy of bidirectional translations between Indonesian and Javanese. This research contributes notably to the advancement of local language translation technologies, supporting language preservation in the digital age and holding promise for applications in other regional languages.

Keywords: Adam Optimizer; BLEU Score; LSTM; Neural Machine Translation.

Introduction

Multilingualism in its various forms is the norm in Indonesia, as over 700 local languages are spoken throughout the archipelago [1]. Therefore, Indonesia appears to have two major problems regarding the local languages, including their fostering and protection. The Java Malay language has an estimated number of speakers of 68.2 million, according to Ethnologue's 2022 report [2]. Unfortunately, in present Indonesia, many of the Javanese, in general, due to the dominance of Bahasa as a national language, the urge to use Javanese in everyday context is dropping, especially in the younger generations when foreign languages become more widespread due to globalization, education, and mass media [3]. To a greater extent, the focalization, in this case, is dangerous since Javanese is closely attached to all cultural values and the local identity of the vast majority of Javanese ethnicities [4]. Hence, technology significantly impacts the context in which it has been implemented as a form of language preservation and modernization of the Javanese language [5]. On top of that, automatic translation, as a part of modern technology, can be utilized to increase knowledge and engagement with the Javanese language, particularly among the younger generations.

Most previous studies on language translation have focused on translating Indonesian into foreign languages, such as [6], which translates Indonesian into English. For regional languages in Indonesia, previous studies have explored translations such as Indonesian-Seramese [7], Indonesian-Sundanese [8], Lampung-Indonesian [9], Indonesian-Batak Toba [10], and Madurese-Indonesian [11]. In the context of Indonesian-Javanese translation, previous research [12], using SMT has shown that parallel and monolingual corpora augmentation improves the quality of phrase-based translation. SMT primarily relies on statistical models, such as trigram models, which provide reliable performance with small corpora [12], [13]. However, these models struggle with the linguistic complexity inherent in Javanese. Further research by [14], has explored strategies such as extending the n-gram model to 7-grams and incorporating monolingual corpora from external sources. While these enhancements offered incremental improvements, the overall

http://dx.doi.org/10.33096/ilkom.v16i3.2467.271-282

gains were limited due to the small size and quality of the corpora [13], [14]. This highlights SMT's inability to fully capture the rich linguistic features of Javanese.

This study continues the previous efforts by enhancing Javanese translation using Adam Optimizer. Adam optimizer is used because of its ability to speed up the convergence process and stabilize weight updates during translation training [15]. As highlighted in several investigations, the Adam optimizer has shown even more unique benefits for language pairs with structural and morphological characteristics in Neural Machine Translation (NMT) models. For instance, authors [11] demonstrated that the inclusion of the Adam optimizer in NMT models raised the evaluation metric values such as Bilingual Evaluation Understudy (BLEU), signifying an overall advancement in the quality of the translations. Based on this understanding, this study also uses the Long Short-Term Memory (LSTM) method, which has shown success in handling sequential data and complex linguistic structures in regional languages, such as Madurese [11]. LSTM is particularly effective in capturing long-range dependencies and maintaining context over sequences, which is crucial for modeling hierarchical languages like Javanese [16]. By combining the strengths of the Adam optimizer and the LSTM method, this study aims to address the linguistic complexities of the Javanese language, particularly its hierarchical speech levels, to improve translation quality effectively.

Method

In this chapter, the researcher describe the procedure for the creation of the NMT model for Indonesian – Javanese language pair based on LSTM architecture. The research process is explained in a sequence of steps. It is depicted in a flowchart in **Figure 1**, introducing fundamental research components in sequence, starting with data acquisition to the final aspect of model evaluation.

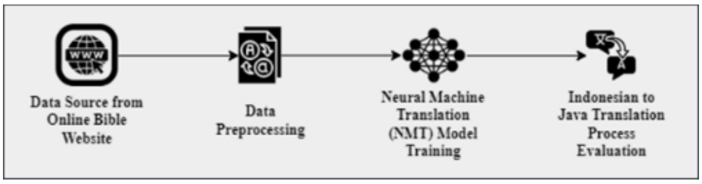


Figure 1. Design of Indonesian-Javanese Machine Translation System

The first procedure includes gathering information from an online Bible selected because it has ample sources of parallel Indonesian texts and Javanese. Then, this data proceeds to preprocessing, which involves tokenization, normalization, scrubbing, and sentence alignment [17]. These procedures help make the data accessible from noise and standardize the training. Tokens are macro units decomposed by the text. Normalization is the process of fixing any differences in the format of the components, and sentence alignment converges sentences from both languages to build a parallel corpus [18]. The next step is training the LSTM-based NMT model with this preprocessed training dataset. Translation tasks are LSTM's strong suit because they retain information over long sequences, which is vital in decoding the contextual embedding of the Indonesian and Javanese languages. Attention mechanisms are embedded in the encoder-decoder architecture to enable the model to concentrate on specific parts during translation, making the results more accurate [19]. The feedback loop manages model translation and feedback, and techniques such as Adam are also utilized for convergence acceleration and stability [20]. The last one includes a performance assessment of the model conducted through a computational algorithm called the BLEU score, which aims to use translation evaluation metrics [21]. Although BLEU certainly provides a means of determining the correctness of the translated text, some other evaluators are also used to evaluate the fluency of the translations and their contextual [22]. This assessment assures the system can generate contextually relevant and grammatically correct translations.

A. Dataset

The research used a dataset from the Online Bible, which consists of more than 31.000 sentence pairs in Indonesian and Javanese. The data was gathered using web scraping, particularly the Web Scraper extension, which enables the automatic gathering of data from web pages. This method collects and assembles large quantities of data quickly and accurately to provide a variety of data for training the translation model. The scraping process commenced with selecting the Bible version and identifying the target language. After successfully extracting data, it was exported in Excel file format (.xlsx). **Table 1** presents an example of the dataset extracted from the book of 2 Chronicles, verses 1-4. The dataset then underwent manual preprocessing.

Table 1. Indonesian-Javanese Dataset

Indonesian	Javanese

Rakyat negeri itu mengambil Yoahas, anak Yosia, dan menjadikan dia raja menggantikan ayahnya di Yerusalem.	*Rakyat ing nagara banjur ngaturi Pangeran Yoahas, putrane Sang Prabu Yosia, sarta kajumenengake ratu ana ing Yerusalem nggentosi kang rama.*
Yoahas berumur 23 tahun ketika dia menjadi raja. Dia memerintah di Yerusalem selama tiga bulan.	*Nalika jumeneng ratu iku Pangeran Yoahas yuswa telulikur taun, dene anggone ngasta paprentahan ana ing Yerusalem suwene telung sasi.*
Raja Mesir memecatnya di Yerusalem dan mendenda negeri itu sebanyak seratus talenta perak dan satu talenta emas.	*Sang Prabu nuli kalungsur saka anggone ngasta paprentahan ing Yerusalem dening Sang Nata ing Mesir, sarta nagara iku kadhendha satus talenta selaka lan satalenta emas.*
Lalu, raja Mesir mengangkat Elyakim, saudara Yoahas, untuk memerintah atas Yehuda dan Yerusalem, dan mengubah namanya menjadi Yoyakim. Namun, Nekho menawan Yoahas, saudaranya, dan membawa dia ke Mesir.	*Sang Nata ing Mesir banjur njumenengake ratu Pangeran Elyakim, sadhereke Sang Prabu Yoahas, ngratoni Yehuda lan Yerusalem, sarta nyantuni asmane dadi Yoyakim. Nanging Sang Prabu Yoahas, sadhereke mau, katawan dening Sang Prabu Nekho, kabekta menyang ing Mesir.*

B. Preprocessing

Following data collection via scraping, preprocessing was performed to ready the data for analysis. This process involved initial manual preprocessing and an automated approach using Python [23]. Manual preprocessing was necessary due to inconsistencies in the placement of translation texts between Indonesian and Javanese. Variations in scraping results were especially apparent in certain verses, where both texts sometimes merged multiple verses into a single entry due to more extended sentence structures. As illustrated in **Table 2**, the second verse shows differing scraping results; the Indonesian text remains uncombined, while the Javanese translation consolidates verses. A similar situation occurs in **Table 3**, where the Indonesian text for verse 18 is combined, but the Javanese translation is not. In this study, two strategies were implemented, separating translations when a verse contains more than three sentences and merging translations when a verse consists of fewer than three sentences.

Table 2. Differences in Verse Structure Javanese in the Indonesian-Javanese Dataset

Indonesian	Javanese
1. Sebab, aku tidak mau kamu tidak mengetahuinya, Saudara-saudara, bahwa para nenek moyang kita, semuanya berada di bawah awan dan semuanya melewati laut.	*1. Anadene karepku, para sadulur, supaya kowe padha ngreti, yen para leluhur kita padha kaauban ing mega, lan kabeh wis padha lumaku nratas patunggilane Nabi Musa sarana kabaptis ana ing mega lan ing sagara mau.*
2. Mereka semua dibaptis dalam Musa, di dalam awan dan di dalam laut.	2. (10:1)

Table 3. Differences in Verse Structure Indonesian in the Indonesian-Javanese Dataset

Indonesian	Javanese
17. Sesungguhnya, TUHAN akan melemparkanmu jauh-jauh, hai manusia! Dia akan memegangmu kuat-kuat, menggulungmu erat seperti bola, dan melemparkanmu ke tanah yang lapang. Di sanalah kamu akan mati dan di sanalah kereta-kereta kemuliaanmu akan tinggal, hai kamu yang menjadi aib di rumah tuanmu.	*17. Lah Pangeran Yehuwah bakal nguncalake sira nganti adoh banget, he manungsa! Sira bakal kacepeng kalawan kenceng,*
18. (22:17)	*18. lan bakal kagulung kenceng dadi gulungan sarta kaglundhungake kaya bal, menyang ing tanah kang amba; ana ing kono sira bakal mati, lan ana ing kono dununge kreta-kreta kamulyanira, he sira kang gawe wiranging brayate gustinira!*

Next, manual preprocessing involved adjusting numeric values and their formats. Inconsistencies were observed in using numeric and cardinal numbers in both texts. **Table 4** illustrates an example from the book of Exodus, where cardinal numbers in Javanese are maintained. In contrast, numeric values or combinations of numeric and cardinal numbers are removed to enhance translation efficiency. This step simplifies and clarifies translation outcomes, ensuring accuracy and consistency within the dataset.

Table 4. Adjustments to Number Formatting in the Indonesian-Javanese Dataset

Before		
No	***Indonesian***	***Javanese***

1.	*Panjang setiap tirai harus 28 hasta, dan lebar setiap tirai harus 4 hasta, dan ukuran semua tirai harus sama.*	*Dawane saben tendha wolulikur asta lan ambane saben tendha patang asta, tendha mau kabeh ukurane padha.*
	After	
1.	*Panjang setiap tirai harus dua puluh delapan hasta, dan lebar setiap tirai harus empat hasta, dan ukuran semua tirai harus sama.*	*Dawane saben tendha wolulikur asta lan ambane saben tendha patang asta, tendha mau kabeh ukurane padha.*

Once the manual preprocessing was finalized, the following action was to perform automated preprocessing with the aid of Python. Before automated preprocessing, the data stored in Microsoft Excel document (.xls) format was transformed into a document file (.txt) format. This automated preprocessing was required to ensure that the data was prepared efficiently and consistently, especially when there was a considerable amount.

The first task was Unicode normalization, where liaison characters were made uniform and used with different encoding schemes. This process included making characters uniformly addressable in Unicode, managing diacritics, and resolving script-specific glyphs to avoid misinterpreting symbols and special characters involving Javanese [24]. Tokenization followed, which is breaking down a sentence into tokens, the most minor units such as words and phrases [25]. This stage ensured that each unit was isolated to allow close examination, which is crucial in other further steps, such as translation. For this analysis, tokenization was modified to enable accurate plotting of Javanese word boundaries and punctuation marks, improving the quality of the analysis that came after.

Lowercasing denotes a procedure where all text is rendered to lowercase letters, allowing for uniform formatting. In this case, the utilization of capitalization was eliminated, which could have otherwise affected word perception and analysis accuracy [26]. Punctuation Removal, Cleansing of unnecessary symbols like [!-./:;ó?@_"#$%&'()*.'|] is not related to language processing, which sanitizes text and cuts down interference from the data in noisy situations [22]. Lastly, all non-printing characters, such as double spaces and invisible control characters, were deleted so that only readable text was left. This stage avoided including useless features in the clean dataset that could hinder the machine. The following **Table 5** outlines and briefly discusses the phases of automated preprocessing.

Table 5. Automated Preprocessing for Text Data

Step	Example Sentence
Raw Sentence	*Sang Yusuf banjur seda, yuswane satus sepuluh taun; layone banjur dijebadi kalebetake trebela ana ing tanah Mesir.*
Tokenization	*(Sang) (Yusuf) (banjur) (seda) (,) (yuswane) (satus) (sepuluh) (taun) (;) (layone) (banjur) (dijebadi) (kalebetake) (trebela) (ana) (ing) (tanah) (Mesir) (.)*
Lowercasing	*sang yusuf banjur seda, yuswane satus sepuluh taun; layone banjur dijebadi kalebetake trebela ana ing tanah mesir.*
Punctuation Removal	*sang yusuf banjur seda yuswane satus sepuluh taun layone banjur dijebadi kalebetake trebela ana ing tanah mesir*

C. Neural Machine Translation

In this study NMT is chosen as the main model, which combines encoder and decoder components in a language model [27]. NMT enables more advanced and accurate automatic translation as it manages to maintain the rather complex relationships of meaning between words without following a set of structural rules [28]. The encoder-decoder architecture best illustrates this methodology, as represented in **Figure 2**, where the encoder and the decoder operate together to translate sentences in a stepwise approach to the context [29].

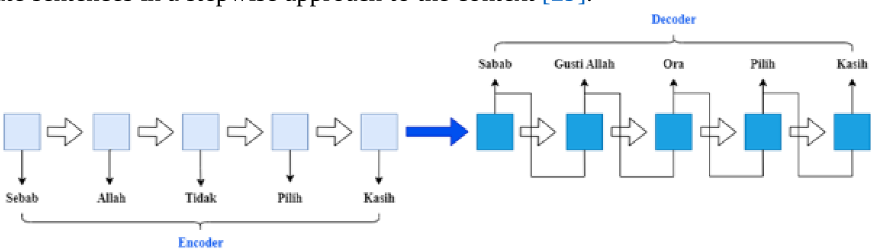


Figure 2. Encoder-Decoder Architecture Model

Within the encoder-decoder schema **Figure 2**, the encoder encodes the input phrase and generates a vector representing it the internal representation that has been composed [30]. This representation will be conveyed to the decoder, which processes the sentence in increments, using the state in the previous step to progress onto the following

[30]. Each step entails a translation whereby each output word relates to the next from a global perspective and helps to anchor the entire flow of the translation. The encoder and the decoder combination does not allow one element of the source sentence to be accurately interpreted, so seamless interpretation in terms of the message and the language of translation occurs [31]. Also, as illustrated in **Figure 3**, this model can be further improved using more sophisticated techniques.

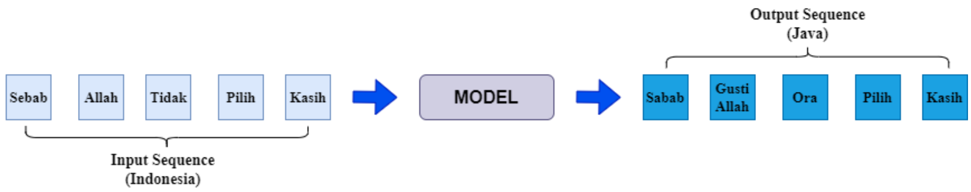


Figure 3. Machine Translation Model from Indonesian to Javanese

The machine Translation Model from the Indonesian Language to the Javanese Language is projected in **Figure 3**. The model interprets the input word sequence in Indonesian and outputs the same in Javanese, preserving the relevant links. Writers also stress the importance of parameters when training the translation model, as presented in **Table 6**.

Table 6. Parameter Settings

No.	Parameter	Value
1.	Activation function	Softmax
2.	Epoch	100, 500, 1000, 1500, 2000
3.	Batch size	64
4.	Verbose	2
5.	Dropout	0.2
6.	Optimizer	Adam, Non-Optimizer

Table 6 also shows the parameter settings for sample training of the NMT model, one of them being the softmax activation function, which transforms scores into class probabilities. The training was done through several epochs of 100, 500, 1000, 1500, and 2000 to polish the learning processes. It was fixed at 64, indicating the number of samples used in each iteration. A verbose level of 2 was set, which indicates a moderate amount of details. The dropout rate of 0.2 was employed to reduce the chances of overfitting, and convergence speed during the training phase was enhanced by employing the Adam optimizer.

D. Evaluation

One of the evaluation metrics used to evaluate the translation results is the BLEU, which compares the generated translated sentences with the translated reference sentences. BLEU is one of the first metrics to assess machine translation output, which is the translated text in the target language from a source language [31]. The statistic-based approach of this algorithm reproduces the correlation between the machine's translation of a text and its statutory translation [32]. As such, the over-approval of the precision score is indicated in overly brief translations by a score of BP, which earns its name from the 'Brevity Penalty' [33].

$$BP_{BLUE} = \{1, if\ c > re^{1\frac{r}{e}}, if\ c \leq r \tag{1}$$

$$P_n = \frac{\sum C \in corpus\ n - gram \in C \sum count\ clip^{(n-gram)}}{\sum C \in corpus\ n - gram \in C \sum count_{(n-gram)}} \tag{2}$$

$$BLEU = BP \times exp\left(\sum_{n=1}^{N} w_n \cdot log\ P_n\right) \tag{3}$$

The BLEU score integrates the precision of n-grams with the length of the translation compared to its reference, awarding a penalty when a significant disparity exists between the two. The brevity penalty (BP) is defined in Equation 1. It is used to scale a score based on the insertion of a period in a penultimate weak translation that is a fraction of the total number of insertions [33]. Equations 2 defines the n-gram precision as the ratio of the number of n-grams present in the translation to the number of such n-grams in the reference text of one to four words or phrases (n-grams). Finally, Equation 3 combines these two measures represented through BP and the N-th power of n-gram precision's average logarithm [12], [33]. The possible scores on the BLEU scale are 0-1, whereas a score of 1 would suggest that the translation corresponds entirely with the reference translation.

Results and Discussion

A. Indonesian-Javanese Translation Result

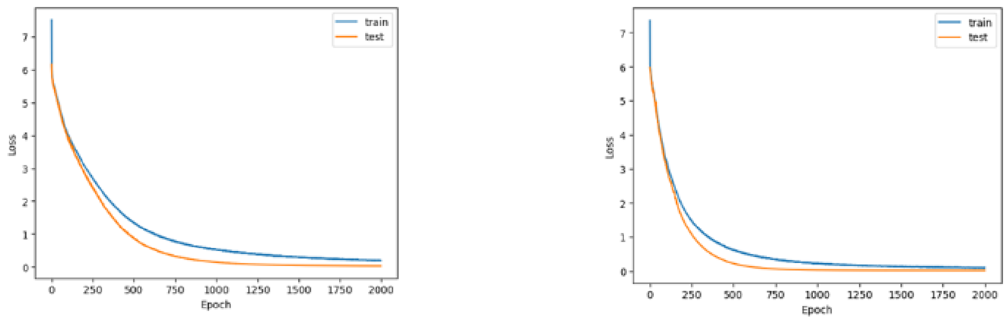
Table 7 presents the evaluation results of the translation model from Indonesian to Javanese. The assessment is based on various epochs and optimizers, utilizing BLEU scores at different n-gram levels (BLEU 1 to BLEU 4).

Table 7. BLEU Result from Indonesia to Java

Epoch	Optimizer	BLEU 1	BLEU 2	BLEU 3	BLEU 4
100	Non-Optimizer	0.231382	0.124957	0.111962	0.067686
500		0.792841	0.691121	0.631387	0.521109
1000		0.982038	0.973043	0.965825	0.947631
1500		0.994549	0.992815	0.991766	0.986541
2000		0.995317	0.993833	0.993056	0.987021
100	Adam	0.272185	0.152366	0.130950	0.078944
500		0.912569	0.865060	0.836199	0.774055
1000		0.994409	0.992425	0.991212	0.985528
1500		0.995457	0.993909	0.992821	0.987225
2000		0.995877	0.994855	0.993981	0.989957

Table 7 shows the BLEU measurement results of the automatic translation process from Indonesian to Javanese. This BLEU measurement method measures the model's accuracy in constructing a translated sentence similar to the reference sentence, where BLEU 1 to BLEU 4 indicates n-grams from one to four words. The higher the BLEU value, the more accurate the translation result. The table shows that the experiments were conducted with two settings, i.e., without optimizer and with Adam optimizer, and measured at several epoch counts, 100, 500, 1000, 1500, and 2000. In the first 100 epochs, both settings show low BLEU values, with no optimizer yielding lower scores than Adam (BLEU 4 of 0.067686 for no optimizer and 0.078944 for Adam). This shows that the model is still in the early stages of learning and produces less accurate translations at low epochs.

As the epochs increased, both without optimizer and with Adam showed significant improvement in all BLEU values. At epoch 500, the BLEU value without optimizer increased dramatically (BLEU 4 reached 0.521109), but Adam showed better performance with BLEU 4 of 0.774055. This performance continues to improve until epoch 2000, where the BLEU value without the optimizer reaches a value of 0.987021 for BLEU 4, while with Adam's optimizer, it reaches 0.989957 at BLEU 4. This indicates that Adam helps the model achieve more optimal results than without the optimizer. Overall, using Adam's optimizer improved the accuracy of the model in translation, especially at higher epochs. These results show that the optimized model can produce more accurate translations with better n-gram structure, which is essential in automatic translation and requires accuracy and contextual appropriateness.



(a.) (b.)

Figure 4. Epoch Graph (a.) Without Optimizer, (b.) With Adam

Figure 4 displays two graphs comparing the model's performance at epoch 2000, both with and without the Adam optimizer. The graph in Figure 4a illustrates the model's performance without an optimizer. Although the model performs well at epoch 2000, convergence occurs more slowly, exhibiting more significant fluctuations before reaching stability. This indicates that the model requires more time to reach an optimal point without an optimization mechanism, and the learning process tends to be unstable. Such instability may result from inconsistent weight update steps, leading to more significant fluctuations in loss before the model finds the optimal minimum.

In contrast, **Figure 4b** shows the model's performance using the Adam optimizer. Here, it is evident that employing the Adam optimizer assists the model in achieving convergence faster and with fewer fluctuations. The Adam optimizer adjusts the learning rate dynamically and accounts for momentum, allowing for more effective and efficient weight updates. At epoch 2000, the graph indicates that the model is already in an optimal state, with minimal loss differences between adjacent epochs. This signifies that the model has stabilized and no significant improvement can be gained by continuing training. Overall, the comparison between these two graphs illustrates that the Adam optimizer accelerates convergence and enhances training stability, making the model more efficient in achieving optimal performance.

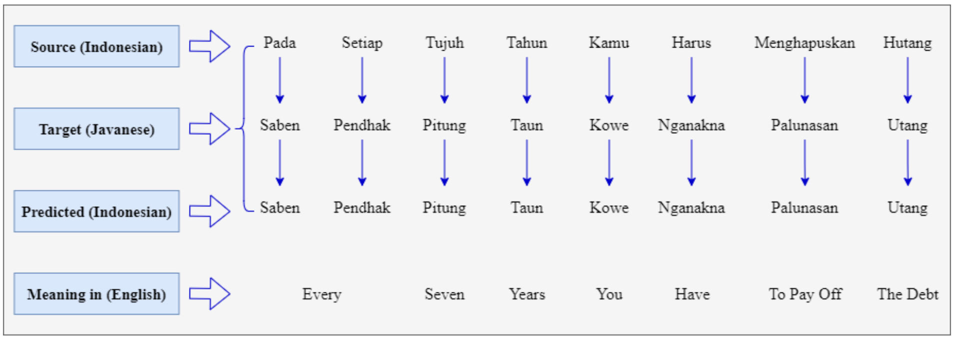


Figure 5. Conformity of Indonesian-Javanese Translation Structure

Figure 5 presents translating words from Indonesian to Javanese in the Source (Indonesian) and Target (Javanese) columns. Each word in Indonesian is accurately translated into Javanese according to its context, such as "Setiap" being translated to "*Saben*" and "*Tujuh*" becoming "*Pitung*." In the Prediction (Javanese) column, the model's predictions closely match the target translations without significant discrepancies. This indicates that the model can translate from Indonesian to Javanese correctly and consistently.

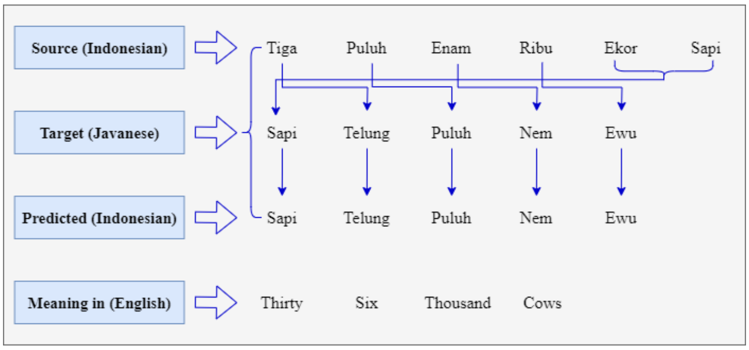


Figure 6. Conformity of Indonesian-Javanese Translation Structure

Figure 6 highlights the translation challenges between the two languages with differing sentence structures, specifically Indonesian and Javanese. The phrase "*Tiga Puluh Enam Ribu Ekor Sapi*" in Indonesian is translated into "Sapi Telung Puluh Nem Ewu" in the Prediction (Javanese) column. While this translation is close to the original meaning, Javanese's word order and structure do not conform to a more natural norm. The Target (Javanese) column shows identical results to the prediction; however, it still feels less ideal due to the word order not following the general pattern of Javanese. In Javanese, the order of numbers and nouns is typically arranged differently to create a more natural sentence. Nevertheless, the translation from Indonesian to Javanese is still considered accurate, as the results remain understandable and aligned with the intended meaning. This underscores that, despite structural differences, the translation into Javanese maintains clarity and contextual relevance.

B. Javanese-Indonesian Translation Result

Table 8 displays the evaluation results of the translation from Javanese to Indonesian using various epochs, comparing the use of the Adam optimizer with no optimizer.

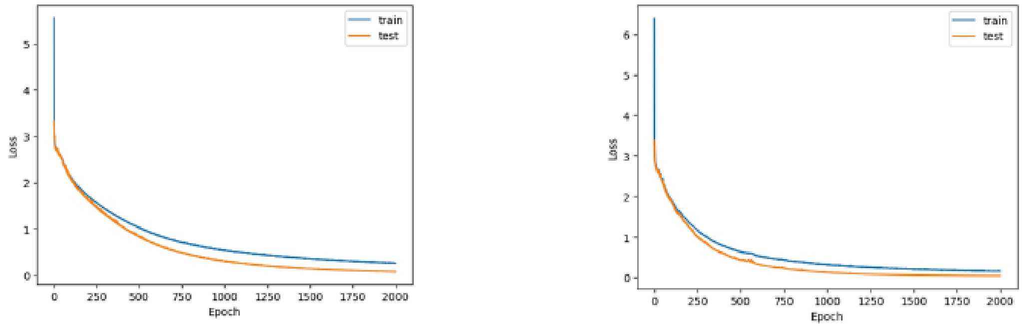
Table 8. BLEU Result from Javanese to Indonesian

Epoch	Optimizer	BLEU 1	BLEU 2	BLEU 3	BLEU 4

100	Non-Optimizer	0.179809	0.078498	0.054465	0.014978
500		0.607327	0.456503	0.386926	0.271104
1000		0.867322	0.792548	0.744126	0.652180
1500		0.955577	0.921072	0.915741	0.880595
2000		0.976625	0.965761	0.959379	0.942472
100	Adam	0.207298	0.109279	0.091552	0.051254
500		0.770584	0.649743	0.582392	0.466536
1000		0.951904	0.924813	0.907288	0.868806
1500		0.980331	0.971652	0.966609	0.952847
2000		0.985027	0.979342	0.976057	0.966987

Table 8 above shows the translation results from Javanese to Indonesian using the same method as the previous table but in the opposite translation direction. This table measures the model performance based on the BLEU value for n-grams (BLEU 1 to BLEU 4) at several epoch counts (100, 500, 1000, 1500, and 2000) with and without the Adam optimizer. At low epoch (100), the BLEU values without the optimizer are in the low range for all n-grams, with BLEU 4 at 0.014978. The use of Adam's optimizer at the same number of epochs increased the score, although it was still relatively low, indicating that the model was still in the primary learning stage. When the number of epochs increased to 500, the model without the optimizer started to show significant improvement, especially on BLEU 1 and BLEU 2. However, Adam gave higher results on all BLEU values, especially on BLEU 4 at 0.466536.

At epoch 1000 to 2000, both approaches show an increasing trend. With Adam's optimizer, the BLEU values increase faster and produce higher scores, especially on BLEU 4, which reaches 0.966987 at 2000 epochs. This value indicates that Adam's optimizer is more effective in improving the translation accuracy from Javanese to Indonesian, reflecting the higher similarity between the model translation and the reference sentence. However, the translation results from Javanese to Indonesian generally still show lower performance compared to translation from Indonesian to Javanese. This is due to the complexity of the Javanese language structure which add to the challenge of producing accurate and contextual translations.



(a) (b)

Figure 7. Epoch Graph (a.) Without Optimizer, (b.) With Adam

Figure 7 compares the performance of the model over 2000 epochs, both with and without the Adam optimizer, illustrating how the model learns and achieves convergence during training. **Figure 7a** displays the model's performance without an optimizer, where the convergence process is slower and initially shows more significant fluctuations. Although the model eventually reaches a lower loss value at the end of the epoch, the gap between training loss and testing loss remains significant initially. This indicates that the model requires more iterations to stabilize weight adjustments without an optimizer.

In contrast, **Figure 7b** indicates that the Adam optimizer model demonstrates faster convergence, with more minor fluctuations in loss values. The model learning process is faster because weights are adjusted correctly and initialized with the Adam optimizer. This shows up with a much faster learning curve and a reduction of the loss at a much earlier stage of training. Around the further epoch, that is, in 2000, the training loss and the testing loss values come almost to one point; these observations lead to the conclusion that the model with Adam optimizer has much more generalization and stability. This faster and more stable convergence leads us to assume that the model also converges much faster in achieving the optimal performance for any particular configuration. The text is translated from Javanese to Indonesian and contained in **Figures 8** and **9**. This translation process reveals that Javanese's word order and sentence length tend to be longer and more complex than in Indonesian.

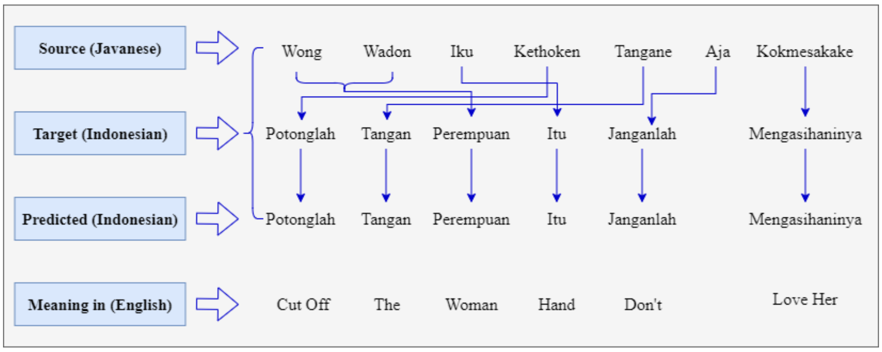


Figure 8. Conformity of Javanese-Indonesian Translation Structure

The Javanese sentence presented in **Figure 8** is "*Wong Wadon Iku Kethoken Tangane Aja Kokmesakake*." When translated directly into Indonesian, it becomes "*Potong tangan wanita itu, jangan tunjukkan rasa kasihannya*." The Javanese sentence includes additional words, such as "*Aja*" (don't) and "*Kokmesakake*" (show mercy), making it longer and sometimes more detailed. This indicates that the sentence structure in Javanese tends to be more complex, with additional words or phrases emphasizing specific meanings.

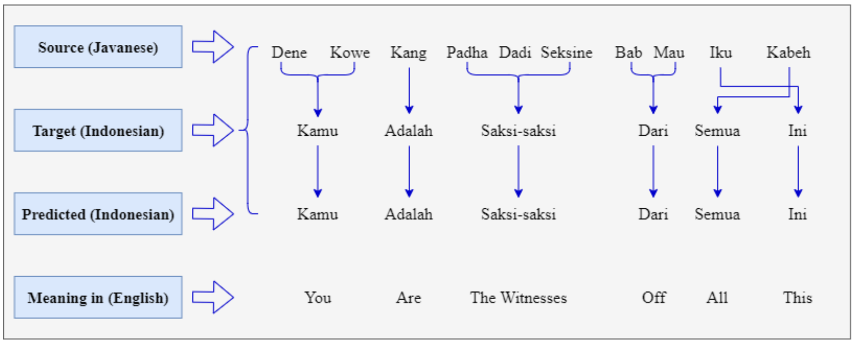


Figure 9. Conformity of Javanese-Indonesian Translation Structure

Figure 9 presents another Javanese sentence, "*Dene Kowe Kang Padha Dadi Seksine Bab Mau Iku Kabeh*." The translation into Indonesian is, "*Kamu adalah saksi dari semua ini*." The structure of this Javanese sentence is longer and contains more elements than its simpler Indonesian version. In this sentence, phrases like "Kang Padha Dadi" (who becomes) provide additional context that is absent in the more straightforward Indonesian version.

C. Factors Influencing Translation Outcomes

The results present that the translation from Indonesian into Javanese is more effective than the translation from Javanese into Indonesian. Many reasons may explain this variation, which is caused by the intricacies of both languages. The Indonesian language is more structured and systemic in grammar, which makes it easier for the model to handle [12]. On the other hand, in the Javanese language, there are varying degrees of speech that require a greater understanding of the context and the words used [34]. In general terms, the translation results have an exciting feature, as the BLEU-1 score has always been higher than the range of BLEU-2 through BLEU-4. The reason for this is that BLEU-1 measures the similarity between words without having to relate words to the context of the n-grams or the n-gram structure [33]. On the other hand, BLEU-2,3 and four work to measure the similarity of longer n-grams, but with consideration of the sequence and context [31]. The latter of these scores present the most stringent measure, which shows how well the model retained the overall sense of the sentences [32].

The Adam optimizer addresses the issues of slow convergence, instability, and inaccuracy in the translations with an improved approach [35]. Adam employs two strategies, which are; momentum and an adaptive learning rate. The use of momentum helps the model to still make strides toward finding the global minimum along with drastic changes in gradients. Alternatively, the adaptive learning rate adjusts the learning rate with the change of the gradient, allowing for more efficient parameter updating. This feature makes Adam more efficient than other optimization approaches, such as Stochastic Gradient Descent (SGD) [36], particularly in the case of the machine translation of complex languages like the Indonesian and Javanese languages. Effects of research have invariably shown that Adam has performed well in optimizing deep learning models for machine translation and has avoided errors brought about by varying rates of gradients [11]. The other equally important issue is the number of epochs in the case of selecting the training periods. It is essential to take note of the number of epochs, as too low can result in an underfitting model, while too many will yield an overfitting model [37]. Therefore, epoch 2000 represented the most efficient model with an accurate number of epochs that enabled the language translation task to be completed effectively while maintaining stability.

As shown in this study, there are specific issues regarding the level of translation of Javanese utterances that are likely to result in lower scores at lower epochs. In Javanese, there are unique linguistic levels of language, all of which indicate levels of respect and formality based on the speaker and audience [12], [34]. Each level also has not only different vocabulary but also various sentence structures, which makes translation very complicated. It is evident that in the low period, where the models are at the beginning and early stages of learning, most models will go beyond what is boasted at this level, as they do not have enough precision to avoid translations that do not fit the social context, which does not exist in this case. This difficulty highlights the fact that models of the sociocultural characteristics of the Javanese language need to be more focused which could otherwise be provided by richer and more graded training tools. Hence, such challenges add to the low BLEU scores recorded in Javanese-Indonesian translation.

To address the challenges identified in this study, future research could focus on incorporating more advanced techniques, such as speech level recognition, into the translation model. By explicitly training models to recognize and adapt to the varying levels of Javanese speech, translation accuracy can be significantly refined. Exploring alternative architectures, such as Transformer-based models or hybrid approaches that combine neural and statistical methods, could also improve the model's ability to handle linguistic complexities. Additionally, expanding and diversifying the training dataset to include more examples of different speech levels and social contexts would enable the model to better capture the nuances of the language. Fine-tuning pre-trained models with a specific emphasis on speech levels could further enhance translation quality. These advancements not only promise to improve the performance of Javanese-Indonesian translations but also contribute to the preservation and digitization of local languages, ensuring their relevance in the modern technological landscape.

Conclusion

The research on the Indonesian-Javanese bilingual NMT model demonstrates the significant impact of the Adam optimizer in enhancing model performance by improving convergence speed and stability, which are essential for addressing the complexities of translating between Indonesian and Javanese. The challenges posed by Javanese, such as its hierarchical language structure and multiple speech levels, require a nuanced understanding of context and word usage, making translation more intricate compared to Indonesian. The model's accuracy, particularly in translating from Indonesian to Javanese, was validated through rigorous evaluation using the BLEU score, indicating significant advancements in translation quality. However, some limitations persist, especially in handling word order and maintaining contextual coherence in Javanese translations. Future research could enhance translation performance by incorporating speech level recognition and exploring advanced architectures, such as Transformer-based models, to address the complexities of Javanese. Expanding and diversifying training datasets, along with fine-tuning pre-trained models focused on speech levels, could significantly improve contextual accuracy and translation quality, while also supporting the preservation of regional languages in the digital era.

References

[1] A. F. Aji *et al.*, "One Country, 700+ Languages: NLP Challenges for Underrepresented Languages and Dialects in Indonesia," *Proc. Annu. Meet. Assoc. Comput. Linguist.*, vol. 1, pp. 7226–7249, 2022, doi: 10.18653/v1/2022.acl-long.500.

[2] A. Fitriati, "The Family Language Policy of Javanese Multilingual Families," *J. Lang. Lit.*, vol. 23, no. 2, pp. 405–415, 2023, doi: 10.24071/joll.v23i2.7020.

[3] E. L. Zen, "Javanese Language As an Ethnic Identity Marker Among Multilingual Families in Indonesia," *Linguist. Indones.*, vol. 39, no. 1, pp. 49–62, 2021, doi: 10.26499/li.v39i1.195.

[4] A. D. S. Muhammad Rohmadi, Memet Sudaryanto, Dias Andris Susanto, Kundharu Saddhono, "Sociopragmatic study of Javanese oral discourse in big city communities as an effort to maintain indigenous language," *Res. J. Adv. Humanit.*, vol. 4, no. 4, 2023, doi: 10.58256/rjah.v4i4.1290 Section: Literature, Linguistics & Criticism.

[5] Z. Sakhiyya and N. Martin-Anatias, "Reviving the language at risk: a social semiotic analysis of the linguistic landscape of three cities in Indonesia," *Int. J. Multiling.*, vol. 20, no. 2, pp. 290–307, 2023, doi: 10.1080/14790718.2020.1850737.

[6] T. Wahyu Guntara, A. Fikri Aji, R. Eko Prasojo, and J. Kemang Raya No, "Benchmarking Multidomain English-Indonesian Machine Translation," *Proc. 13th Work. Build. Using Comp. Corpora*, no. May, pp. 35–43, 2020.

[7] A. W. Rukua, Y. A. Lesnussa, D. L. Rahakbauw, and B. P. Tomasouw, “Application of Neural Machine Translation with Attention Mechanism for Translation of Indonesian to Seram Language (Geser),” *Pattimura Int. J. Math.*, vol. 2, no. 2, pp. 53–62, 2023, doi: 10.30598/pijmathvol2iss2pp53-62.

[8] C. T. and K. H. B. D. Wijanarko, Y. Heryadi, D. F. Murad, “Recurrent Neural Network-based Models as Bahasa Indonesia-Sundanese Language Neural Machine Translator,” *Int. Conf. Comput. Sci. Inf. Technol. Eng. (ICCoSITE), Jakarta, Indones.*, pp. 951–956, 2023.

[9] P. Permata and Z. Abidin, “Statistical Machine Translation Pada Bahasa Lampung Dialek Api Ke Bahasa Indonesia,” *J. Media Inform. Budidarma*, vol. 4, no. 3, p. 519, 2020, doi: 10.30865/mib.v4i3.2116.

[10] I. A. and A. S. P. Z. Abidin, A. Junaidi, Wamiliana, F. M. Togatorop, “Direct Machine Translation Indonesian-Batak Toba,” *7th Int. Conf. New Media Stud. (CONMEDIA), Bali, Indones.*, pp. 82–87, 2023, doi: 10.1109/CONMEDIA60526.2023.10428332.

[11] D. A. Sulistyo, A. P. Wibawa, D. D. Prasetya, and F. A. Ahda, “LSTM-Based Machine Translation for Madurese-Indonesian,” *J. Appl. Data Sci.*, vol. 4, no. 3, pp. 189–199, 2023.

[12] A. E. P. Lesatari, A. Ardiyanti, and I. Asror, “Phrase based statistical machine translation Javanese-Indonesian,” *J. Media Inform. Budidarma*, vol. 5, no. 2, pp. 378–386, 2021.

[13] S. Saxena, S. Chauhan, P. Arora, and P. Daniel, “Unsupervised SMT: an analysis of Indic languages and a low resource language,” *J. Exp. Theor. Artif. Intell.*, pp. 865–884, 2022, doi: 10.1080/0952813X.2022.2115142.

[14] H. Sujaini, “Improving the role of language model in statistical machine translation (Indonesian-Javanese),” *Int. J. Electr. Comput. Eng.*, vol. 10, no. 2, pp. 2102–2109, 2020, doi: 10.11591/ijece.v10i2.pp2102-2109.

[15] M. Reyad, A. M. Sarhan, and M. Arafa, “A modified Adam algorithm for deep neural network optimization,” *Neural Comput. Appl.*, vol. 35, no. 23, pp. 17095–17112, 2023, doi: 10.1007/s00521-023-08568-z.

[16] A. Lopopolo, E. Fedorenko, R. Levy, and M. Rabovsky, “Cognitive Computational Neuroscience of Language: Using Computational Models to Investigate Language Processing in the Brain,” *Neurobiol. Lang.*, vol. 5, no. 1 Special Issue, pp. 1–6, 2024, doi: 10.1162/nol_e_00131.

[17] G. D. K. M. Kashina, I.D. Lenivtceva, “Preprocessing of unstructured medical data: the impact of each preprocessing stage on classification,” *Procedia Comput. Sci.*, vol. 178, pp. 284–290, 2020, doi: 10.1016/j.procs.2020.11.030.

[18] K. Dedes, A. B. P. Utama, A. P. Wibawa, A. N. Afandi, A. N. Handayani, and L. Hernandez, “Neural machine translation of Spanish-English food recipes using LSTM,” *JOIV Int. J. Informatics Vis.*, vol. 6, no. 2, pp. 290–297, 2022.

[19] A. Pranolo, Y. Mao, A. P. Wibawa, A. B. P. Utama, and F. A. Dwiyanto, “Robust LSTM With tuned-PSO and bifold-attention mechanism for analyzing multivariate time-series,” *Ieee Access*, vol. 10, pp. 78423–78434, 2022.

[20] N. Mathur, T. Baldwin, and T. Cohn, “Tangled up in BLEU: Reevaluating the evaluation of automatic machine translation evaluation metrics,” *arXiv Prepr. arXiv2006.06264*, 2020.

[21] S. Lee *et al.*, “A Survey on Evaluation Metrics for Machine Translation,” *Mathematics*, vol. 11, no. 4, pp. 1–22, 2023, doi: 10.3390/math11041006.

[22] A. P. Wibawa, I. T. Saputra, A. B. P. Utama, W. Lestari, and Z. N. Izdihar, “Long Short-Term Memory to Predict Unique Visitors of an Electronic Journal,” in *2020 6th International Conference on Science in Information Technology (ICSITech)*, IEEE, 2020, pp. 176–179.

[23] K. C. A. Thakkar, D. Mungra, A. Agrawal, “Improving the Performance of Sentiment Analysis Using Enhanced Preprocessing Technique and Artificial Neural Network,” *IEEE Trans. Affect. Comput.*, vol. 13, no. 4, pp. 1771–1782, 2022, doi: 10.1109/TAFFC.2022.3206891.

[24] Q. Dombrowski, M. Goregaokar, B. Joeng, and A. Kamran, “Encoding Multilingualism: Technical Affordances of Multilingual Publication from Manuscripts to Unicode and OpenType,” *J. Electron. Publ.*, vol. 27, no. 1, 2024.

[25] A. Rai and S. Borah, “Study of various methods for tokenization,” in *Applications of Internet of Things: Proceedings of ICCCIOT 2020*, Springer, 2021, pp. 193–200.

[26] G. V. A. Gutiérrez, “A comparative study of NLP and machine learning techniques for sentiment analysis and topic modeling on amazon,” *Int. J. Comput. Sci. Eng*, vol. 9, no. 2, pp. 159–170, 2020.

[27] M. Isik and H. Dag, “The impact of text preprocessing on the prediction of review ratings,” *Turkish J. Electr. Eng. Comput. Sci.*, vol. 28, no. 3, pp. 1405–1421, 2020.

[28] S. Edunov, A. Baevski, and M. Auli, “Pre-trained language model representations for language generation,” *arXiv Prepr. arXiv1903.09722*, 2019.

[29] A. Javaloy and G. García-Mateos, “Text normalization using encoder-decoder networks based on the causal feature extractor,” *Appl. Sci.*, vol. 10, no. 13, 2020, doi: 10.3390/app10134551.

[30] K. Aitken, V. Ramasesh, Y. Cao, and N. Maheswaranathan, “Understanding how encoder-decoder architectures attend,” *Adv. Neural Inf. Process. Syst.*, vol. 34, pp. 22184–22195, 2021.

[31] M. Stasimioti, V. Sosoni, K. L. Kermanidis, and D. Mouratidis, “Machine Translation Quality: A comparative evaluation of SMT, NMT and tailored-NMT outputs,” in *Proceedings of the 22nd annual conference of the European Association for Machine Translation*, 2020, pp. 441–450.

[32] E. Chatzikoumi, “How to evaluate machine translation: A review of automated and human metrics,” *Nat. Lang. Eng.*, vol. 26, no. 2, pp. 137–161, 2020.

[33] Z. Sun, J. M. Zhang, M. Harman, M. Papadakis, and L. Zhang, “Automatic testing and improvement of machine translation,” in *Proceedings of the ACM/IEEE 42nd international conference on software engineering*, 2020, pp. 974–985.

[34] S. H. Wijono, K. Azizah, and W. Jatmiko, “Canonical Segmentation For Javanese-Indonesian Neural Machine Translation,” *J. Eng. Sci. Technol.*, vol. 18, pp. 62–68, 2023.

[35] X. Jiang, B. Hu, S. Chandra Satapathy, S. H. Wang, and Y. D. Zhang, “Fingerspelling Identification for Chinese Sign Language via AlexNet-Based Transfer Learning and Adam Optimizer,” *Sci. Program.*, vol. 2020, 2020, doi: 10.1155/2020/3291426.

[36] N. Landro, I. Gallo, and R. La Grassa, “Mixing Adam and SGD: A combined optimization method,” *arXiv Prepr. arXiv2011.08042*, 2020.

[37] A. B. Shanmugavel, V. Ellappan, A. Mahendran, M. Subramanian, R. Lakshmanan, and M. Mazzara, “A Novel Ensemble Based Reduced Overfitting Model with Convolutional Neural Network for Traffic Sign Recognition System,” *Electron.*, vol. 12, no. 4, 2023, doi: 10.3390/electronics12040926.