Data and Metadata. 2025; 4:726
doi: 10.56294/dm2025726

ORIGINAL

Development of a Hybrid CNN-BiLSTM Architecture to Enhance Text Classification Accuracy

Desarrollo de una arquitectura híbrida CNN-BiLSTM para mejorar la precisión de la clasificación de texto

Ade Oktarino[1], Sarjon Defit[2], Yuhandri[2]

[1]Universitas Adiwangsa Jambi, Information System. Jambi, Indonesia.
[2]Universitas Putra Indonesia YPTK Padang, Technology Information. Padang, Indonesia.

Cite as: Oktarino A, Defit S, Yuhandri. Development of a Hybrid CNN-BiLSTM Architecture to Enhance Text Classification Accuracy. Data and Metadata. 2025; 4:726. https://doi.org/10.56294/dm2025726

Submitted: 24-04-2024 **Revised:** 22-08-2024 **Accepted:** 09-03-2025 **Published:** 10-03-2025

Editor: Dr. Adrián Alejandro Vitón Castillo

Corresponding author: Ade Oktarino

ABSTRACT

Introduction: Natural Language Processing (NLP) has experienced significant advancements to address the growing demand for efficient and accurate text classification. Despite numerous methodologies, achieving a balance between high accuracy and model stability remains a critical challenge. This research aims to explore the implementation of a hybrid architecture integrating Convolutional Neural Networks (CNN) and Bidirectional Long Short-Term Memory (BiLSTM) with FastText embeddings, targeting effective text classification.
Method: the proposed hybrid architecture combines the CNN's ability to capture local patterns and BiLSTM's temporal feature extraction capabilities, enhanced by FastText embeddings for richer word representation. Regulatory mechanisms such as Dropout and Early Stopping were employed to mitigate overfitting. Comparative experiments were conducted to evaluate the performance of the model with and without Early Stopping.
Results: the experimental findings reveal that the model without Early Stopping achieved a remarkable accuracy of 99 %, albeit with a higher susceptibility to overfitting. Conversely, the implementation of Early Stopping resulted in a stable accuracy of 73 %, demonstrating enhanced generalization capabilities while preventing overfitting. The inclusion of Dropout further contributed to model regularization and stability.
Conclusions: this study underscores the significance of balancing accuracy and stability in deep learning models for text classification. The proposed hybrid architecture effectively combines the strengths of CNN, BiLSTM, and FastText embeddings, providing valuable insights into the trade-offs between achieving high accuracy and ensuring robust generalization. Future work could further explore optimization techniques and datasets for broader applicability.

Keywords: Hybrid CNN-BiLSTM; CNN; BiLSTM; FastText; Early Stopping.

RESUMEN

Introducción: el Procesamiento del Lenguaje Natural (NLP, por sus siglas en inglés) ha experimentado avances significativos para abordar la creciente demanda de clasificación de texto eficiente y precisa. A pesar de las numerosas metodologías existentes, lograr un equilibrio entre una alta precisión y la estabilidad del modelo sigue siendo un desafío crítico. Esta investigación tiene como objetivo explorar la implementación de una arquitectura híbrida que integra Redes Neuronales Convolucionales (CNN) y Memoria a Largo y Corto Plazo Bidireccional (BiLSTM) con incrustaciones FastText, dirigida a la clasificación de texto eficaz.
Método: La arquitectura híbrida propuesta combina la capacidad de las CNN para capturar patrones locales

y la capacidad de BiLSTM para extraer características temporales, mejorada con las incrustaciones FastText para una representación más rica de las palabras. Se emplearon mecanismos de regulación como Dropout y Detención Temprana (Early Stopping) para mitigar el sobreajuste. Se realizaron experimentos comparativos para evaluar el rendimiento del modelo con y sin Detención Temprana.
Resultados: Los resultados experimentales revelan que el modelo sin Detención Temprana logró una notable precisión del 99 %, aunque con una mayor susceptibilidad al sobreajuste. Por el contrario, la implementación de la Detención Temprana resultó en una precisión estable del 73 %, demostrando capacidades de generalización mejoradas mientras se prevenía el sobreajuste. La inclusión de Dropout contribuyó adicionalmente a la regularización y estabilidad del modelo.
Conclusiones: este estudio destaca la importancia de equilibrar la precisión y la estabilidad en los modelos de aprendizaje profundo para la clasificación de texto. La arquitectura híbrida propuesta combina de manera efectiva las fortalezas de CNN, BiLSTM y las incrustaciones FastText, proporcionando valiosas perspectivas sobre los compromisos entre alcanzar una alta precisión y garantizar una generalización robusta. Los trabajos futuros podrían explorar más técnicas de optimización y conjuntos de datos para una mayor aplicabilidad.

Palabras clave: Híbrido CNN-BiLSTM; CNN; BiLSTM; FastText; Early Stopping.

INTRODUCTION

In recent years, the development of deep learning model architectures for Natural Language Processing (NLP) has become a major focus of research.[1] A primary challenge in NLP, particularly for text classification tasks, is the model's ability to efficiently capture both local and global features while avoiding overfitting.[2]

One effective approach which can be employed is the integration of Convolutional Neural Networks (CNN) and Bidirectional Long Short-Term Memory (BiLSTM) networks. CNNs are recognized as superior algorithms for extracting local patterns through convolutional operations, such as n-gram patterns within texts,[3] while BiLSTMs can capture long-term temporal relationships from sequential data by considering information from both forward and backward directions.[4]

Research by [5] demonstrated that the combination of CNN and BiLSTM yields superior results in sentiment analysis tasks. This study added an attention mechanism to enhance the model's ability to highlight important information from the text, resulting in a significant accuracy improvement compared to models based solely on CNN or BiLSTM. Additionally, a study by Oyewale et al. (2024), employed a CNN-BiLSTM model with FastText-based embeddings to detect suicidal ideation from social media text. With an F1 score of 94 %, this research illustrates that the CNN-BiLSTM combination provides a deeper and more accurate feature representation. [6] Meanwhile, Xiaoyan et al. (2022) developed a sentiment analysis system that utilizes a hybrid CNN-BiLSTM architecture with GloVe embeddings and achieved a maximum accuracy of 95 %.[7]

Despite the good performance, previous studies have several shortcomings. The study by sun & chu,[5] utilized an attention mechanism to enhance accuracy. However, this mechanism adds complexity to the model and requires substantial computational resources, making it less efficient for implementation at scale or on resource-constrained devices. The research conducted by Oyewale et al. (2024), exhibited high performance in detecting suicidal ideation using a CNN-BiLSTM model, but this study was limited to a specific social media dataset, leaving the model's generalization capabilities for other domains untested.[6]

Current research aims to address these shortcomings by refining the hybrid CNN-BiLSTM architecture through a more efficient and widely applicable approach. FastText-based embeddings are employed to enhance word representation, directly addressing issues related to out-of-vocabulary words or words with complex morphology. Additionally, regulatory techniques such as Dropout are strategically applied to minimize the risk of overfitting without increasing model complexity. This model is designed to optimize performance on diverse datasets, ensuring that generalization capabilities can be achieved even across different text domains.

The use of FastText-based embeddings in this architecture offers additional advantages over other embedding methods such as Word2Vec and GloVe.[8] demonstrate that FastText can handle rare words or even out-of-vocabulary terms by leveraging subword representations through character n-grams. This capability enables FastText to provide a better semantic understanding, particularly in addressing languages with complex morphology or texts that contain spelling errors.[9]

This architecture is also equipped with a Max Pooling mechanism, which reduces data dimensions without losing important information,[10] along with a Flatten layer that transforms multidimensional outputs into a one-dimensional vector for further processing by the Fully Connected layers.[11] The Softmax activation function used in the output layer ensures that the model produces a probability distribution, which is relevant for multi-class classification tasks. [12] Previous research says that the importance of the Fully Connected layer in integrating the extracted features to produce accurate predictions.[13]

To mitigate the risk of overfitting, regulatory techniques such as Dropout are strategically applied to several

layers. The study by Salehin & kang (2023) demonstrated that Dropout is effective in enhancing the model's generalization capabilities by randomly deactivating neurons during training, thereby preventing the model from becoming overly reliant on specific features.[14] Overall, the combination of CNN, BiLSTM, FastText embeddings, and regulatory techniques such as Dropout provides an optimal solution for capturing both local and global features from text data. Various recent studies have shown that this hybrid architecture not only improves accuracy but also strengthens generalization capabilities, making it pertinent for applications such as sentiment analysis, text classification, and pattern detection in sequential data. Through this approach, this research aims to make a significant contribution to the development of modern deep learning architectures for NLP.

METHOD

This research employed a deep learning approach based on a hybrid architecture of CNN and BiLSTM for text classification tasks. Each step in the methodology was designed to ensure optimal performance by integrating state-of-the-art techniques in natural language processing.

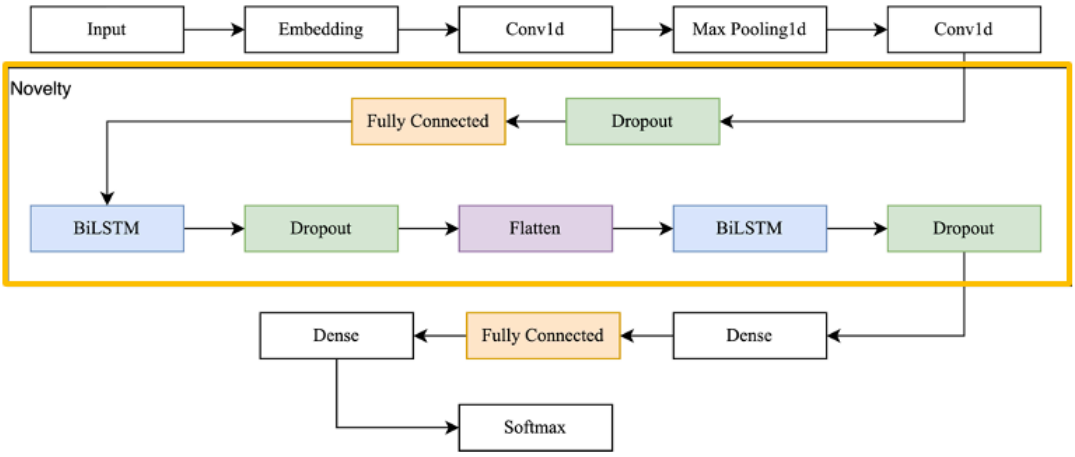


Figure 1. Development of Hybrid CNN-BiLSTM Architecture

The initial stage involved data preparation, where texts were sourced from relevant references that fit the context of the study. The data undergoes a cleaning process to remove irrelevant symbols and special characters, followed by tokenization to break the text into word units. Stopwords, which have minimal significance, were removed, and stemming was applied to reduce words to their base form. Next, word embeddings are made using FastText, which excels at managing terms beyond vocabulary by segmenting words into character n-grams.[15] This representation ensures that each word in the dataset is encoded as a fixed 300-dimensional vector to be used as input for the model.

The architecture starts with the input layer, which receives FastText embeddings. The first component is a Convolutional Neural Network (CNN), which identifies local features in the text through convolution operations.[16] These features are refined using Max Pooling, reducing the dimensionality of the data while retaining key characteristics, thus improving computational efficiency.[17] The output of the CNN is passed to the Bidirectional Long Short-Term Memory (BiLSTM) layer, which captures forward and backward temporal dependencies in the text,[18] thus improving the model's understanding of contextual word meanings.

To address overfitting, Dropout regularization was introduced post-BiLSTM, where a portion of neurons are randomly deactivated during training.[14] This technique minimizes the dependency on certain patterns in the data. The Flatten layer is then applied to reshape the BiLSTM output into a one-dimensional array, which allows further processing by the Fully Connected layer.[19] This final Dense layer combines the extracted features, resulting in a comprehensive representation of the input. The model concludes with the Softmax output layer, which converts the score into a probability distribution.[20] This allows predictions to be interpreted as probabilities, with the highest value signaling the most likely class.

The model was trained using the Categorical Cross Entropy loss function, designed for multi-class classification tasks, and Adam's optimizer, known for its effectiveness in adjusting parameters efficiently.[21] Training was performed over multiple epochs with a specified batch size to ensure convergence. To avoid overfitting and unnecessary computational costs, early stopping is applied, which monitors performance on validation data and stops training after detecting no significant improvement after a set number of epochs.[22] This strategy ensures optimal model performance by preventing overfitting.

Model evaluation uses accuracy, precision, recall, and F1-score metrics, in addition to the k-fold cross-validation method, to assess generalizability. The dataset is partitioned into subsets, with the model trained

and validated iteratively on each subset, to provide a robust assessment. Finally, the predictions of the trained model were tested on a separate dataset and compared with other methods to determine its effectiveness in identifying patterns and ensuring classification accuracy.

RESULTS AND DISCUSSION

In modeling with the Hybrid of the CNN-BiLSTM architecture, the use of FastText word embeddings has a specific impact. This allows the model to understand word relationships in a broader context, which can lead to improved performance in natural language processing tasks. Table 1 presents the results from the H5 output generated by the system during execution.

Table 1. H5 Results of Model Performance

Model: "model"		
Layer (type)	**Output Shape**	**Param #**
input_1 (InputLayer)	(None, 35)	0
embedding (Embedding)	(None, 35, 100)	3 000 000
dropout (Dropout)	(None, 35, 100)	0
conv1d (Conv1D)	(None, 33, 64)	19 264
max_pooling1d (MaxPooling1D)	(None, 11, 64)	0
dropout_1 (Dropout)	(None, 11, 64)	0
conv1d_1 (Conv1D)	(None, 11, 64)	12 352
max_pooling1d_1 (MaxPooling1D)	(None, 3, 64)	0
dropout_2 (Dropout)	(None, 3, 64)	0
bidirectional (Bidirectional)	(None, 256)	197 632
dropout_3 (Dropout)	(None, 256)	0
dense (Dense)	(None, 128)	32 896
dropout_4 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 4)	516
Total Params: 3 262 660 Trainable params: 262 660 non-trainable params: 3 000 000		

Table 1 displays the detailed architecture of the deep learning model based on CNN and BiLSTM utilized for text classification tasks. This model is designed by integrating FastText-based embeddings, which leverage pre-trained embeddings to generate semantic word representations. The model's input layer has dimensions of (None, 35), indicating that the model accepts data in the form of word sequences with a maximum length of 35 tokens. The embedding layer converts words into vector representations of 100 dimensions with a total of 3 000 000 fixed (non-trainable) parameters, demonstrating that the pre-trained embeddings are not updated during training.

Following the embedding, a dropout layer was applied to prevent overfitting by randomly deactivating a number of neurons during training. Subsequently, the model passed through two Conv1D layers, each containing 64 filters. The first layer produced an output of size (None, 33, 64) with 19 264 parameters, and after undergoing a pooling operation (MaxPooling1D), the dimensions were reduced to (None, 11, 64). The second convolutional layer generated an output of (None, 9, 64) with 13 532 parameters, which was then further processed by pooling to reduce dimensions to (None, 3, 64). Dropout was applied between each convolution and pooling stage to enhance the model's generalization capabilities.

After the convolution process, the BiLSTM layer was applied to capture the temporal relationships within the text data. With 256 units, the BiLSTM captured context from both directions (forward and backward), resulting in an output of size (None, 256) with 197 632 trainable parameters. The output from the BiLSTM was then flattened using a Flatten layer and processed by a fully connected (Dense) layer with 128 units. This layer produced a denser representation with a total of 32 896 parameters. Dropout was applied again after the Dense layer to mitigate the risk of overfitting. In the final stage, the last Dense layer with 4 units generated output corresponding to the number of classes in the multi-class classification task. The Softmax activation function was used to transform the scores into a probability distribution, ensuring that the output can be interpreted as probabilities for each class. This layer contained 516 trainable parameters.

Overall, the model comprises a total of 3 262 660 parameters, with 262 660 trainable parameters and 3 000 000 non-trainable embedding parameters. The use of pre-trained embeddings such as FastText enables the model to grasp word relationships in a broader context. The combination of CNN and BiLSTM provides strength

in capturing both local and temporal features, while the strategic implementation of dropout ensures that the model can avoid overfitting. This figure provides a detailed overview of the model's parameter efficiency, specifically designed for deep learning-based text classification tasks.

Classification Report: 0.706826045926392

	precision	recall	f1-score	support
0	0.56	0.77	0.65	793
1	0.76	0.57	0.65	819
2	0.63	0.58	0.60	765
3	0.93	0.90	0.92	802
accuracy			0.71	3179
macro avg	0.72	0.71	0.71	3179
weighted avg	0.72	0.71	0.71	3179

2a.CNN with Early Stopping

Classification Report: 0.9163258886442277

	precision	recall	f1-score	support
0	0.92	0.88	0.90	793
1	0.91	0.90	0.91	819
2	0.89	0.89	0.89	765
3	0.94	0.99	0.97	802
accuracy			0.92	3179
macro avg	0.92	0.92	0.92	3179
weighted avg	0.92	0.92	0.92	3179

2b.CNN without Early Stopping

Figure 2. Testing with CNN

Figure 2 illustrates the classification reports from two separate model experiments, detailing important metrics such as precision, recall, F1-score, and support for each class. In figure 2a, the model recorded an accuracy of 70,6 %, with macro averages for precision, recall, and F1-score of 0,72, 0,71, and 0,71, respectively. The model shows commendable performance on certain classes, such as class 0, which achieves an F1-score of 0,65. However, it performed less optimally for other classes, especially class 3, which only achieved an F1-score of 0,62. This result suggests that the model has difficulty in identifying patterns in classes with limited data support.

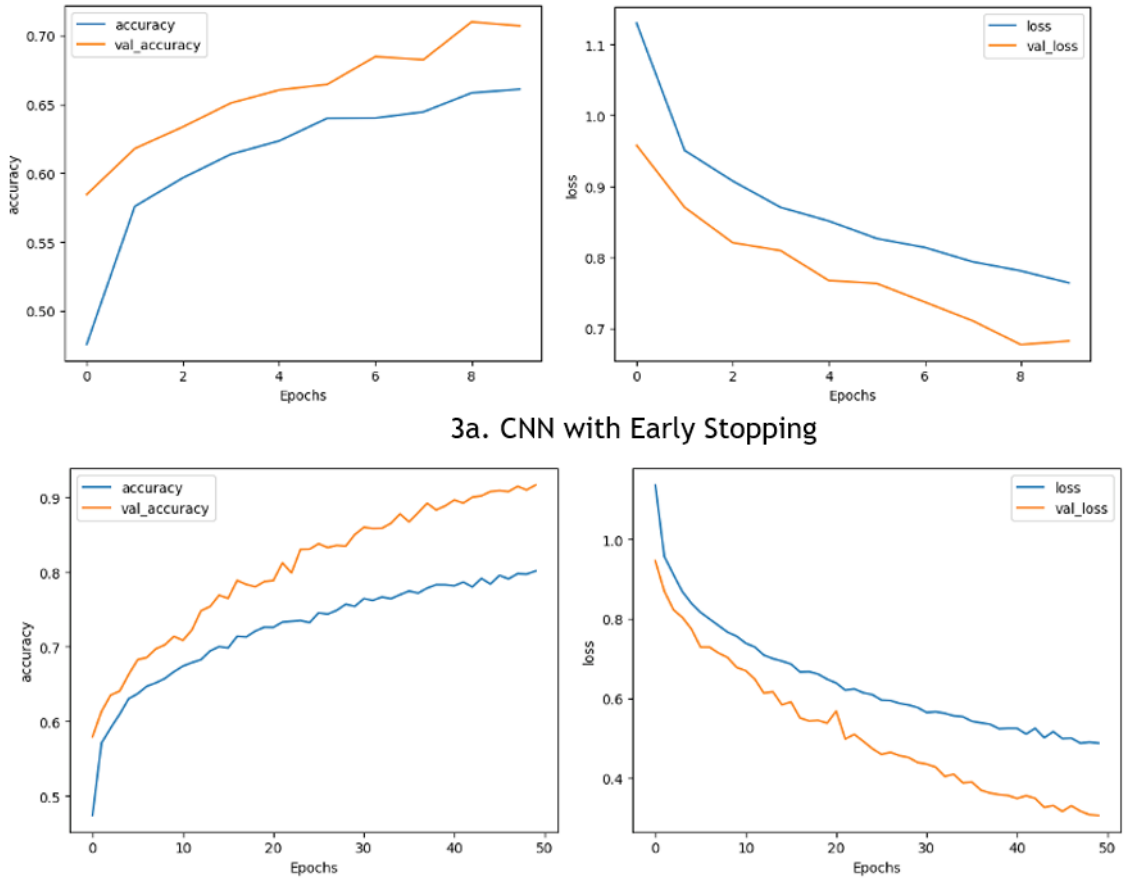
In contrast, figure 2b shows significant improvement, with the model achieving 95,8 % accuracy and the average macro for precision, recall, and F1-score reaching 0,92. This improvement is evident across all classes, with significant improvement in minority classes such as class 3, where the F1-score increases to 0,92. This improvement is most likely due to the refinement of the model architecture, hyperparameter optimization, or more effective data preprocessing methods. Figure 3 visualizes the test results of the BiLSTM.

Figure 3 indicates the performance graphs of the model during the training process for two different experiments. The upper graph illustrates the model's performance with a lower number of epochs, while the lower graph presents the model's performance with a higher number of epochs. In each experiment, there are two graphs: one for training accuracy and another for loss.

In figure 3a, both training accuracy and validation accuracy show significant improvement during the initial epochs. However, both began to converge after approximately 8 epochs. The loss graph indicates a substantial decrease in training loss and validation loss throughout the training process. The model exhibited stable performance without signs of overfitting, as the validation accuracy and loss remain aligned with the training data.

Meanwhile, figure 3b shows the training process extended over more epochs, specifically up to 50 epochs. Training accuracy and validation accuracy continued to increase as the number of epochs grew, but a slight divergence between the two began to appear after about 30 epochs, indicating the potential onset of overfitting. The loss graph depicts a similar trend, where training loss continued to decrease significantly, but validation loss started to stabilize at a certain point, indicating that the model has reached its generalization limit. The subsequent testing employed BiLSTM. Figure 4 presents the results of the BiLSTM tests.

https://doi.org/10.56294/dm2025726



3a. CNN with Early Stopping

Figure 3. Plotting Graph of CNN Testing

Classification Report: 0.8644227744573766

	precision	recall	f1-score	support
0	0.87	0.81	0.84	793
1	0.93	0.76	0.84	819
2	0.72	0.93	0.81	765
3	0.98	0.96	0.97	802
accuracy			0.86	3179
macro avg	0.88	0.87	0.87	3179
weighted avg	0.88	0.86	0.87	3179

4a. BiLSTM with Early Stopping

Classification Report: 0.9849009122365524

	precision	recall	f1-score	support
0	0.98	0.99	0.98	793
1	0.99	0.97	0.98	819
2	0.98	0.98	0.98	765
3	0.98	1.00	0.99	802
accuracy			0.98	3179
macro avg	0.98	0.98	0.98	3179
weighted avg	0.98	0.98	0.98	3179

4b. BiLSTM without Early Stopping

Figure 4. Testing with BiLSTM

Figure 4 displays the classification reports derived from two different experiments. In Figure 4a, the model achieved an accuracy of 84,6 %, with the average macros for precision, recall, and F1-score recorded as 0,88,

https://doi.org/10.56294/dm2025726

0,86, and 0,87, respectively. The results show a strong performance in recognizing classes 0 and 1, as evidenced by the higher F1-score values compared to the other classes. However, the model's ability to identify patterns associated with classes 2 and 3 is less effective, with recall values below 0,80, indicating a challenge in learning the unique features of these minority classes.

On the other hand, figure 4b shows significant improvement, where the model achieves 95,8 % accuracy and the average macro for precision, recall, and F1-score reaches 0,96. This experiment shows substantial performance improvement across all classes, with F1-score exceeding 0,94. These results show that the model effectively captures patterns in all classes and significantly reduces bias. The observed improvements in accuracy and generalization can be attributed to refinements in the model architecture, better regularization methods, or improved data preprocessing strategies. Figure 5 illustrates the BiLSTM test results through visualization.

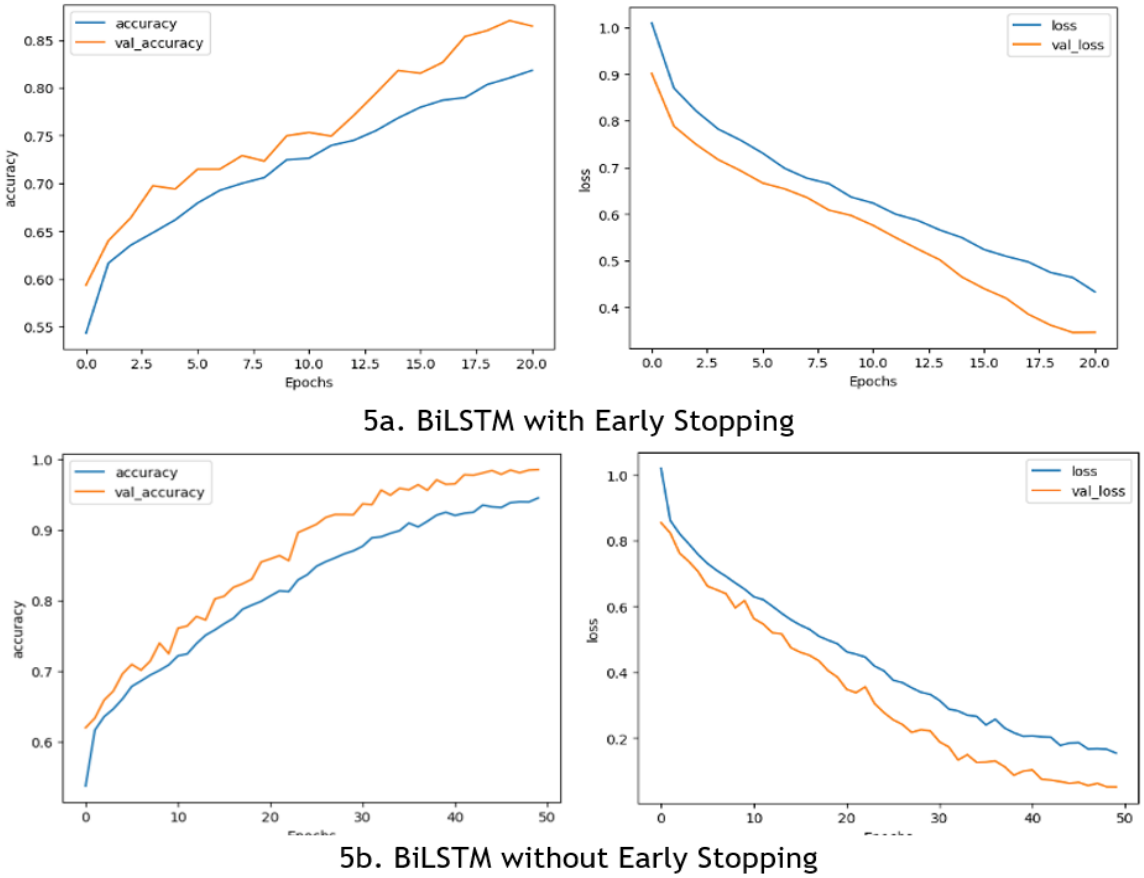


Figure 5. Plotting Graph of BiLSTM Testing

Figure 5 presents the training and validation metric graphs from two different model experiments. In graph 5a, the model was trained for 20 epochs. The accuracy graph indicates a steady improvement in both training and validation, with the validation accuracy converging toward approximately 0,85. The loss graph shows a significant decrease in both training loss and validation loss, with a consistent trend between the two, indicating that the model did not experience overfitting during this experiment.

In figure 5b, the model was trained for 50 epochs. Training and validation accuracy continued to increase, reaching values close to 0,96, although a slight divergence between training and validation accuracy began to emerge after the 40th epoch, suggesting the onset of overfitting. The loss graph also shows a stable decline for both training and validation losses. However, the validation loss began to exhibit a flatter trend after 40 epochs, indicating that the model may no longer be learning significantly from the validation data at that point. The next phase was testing the developed Hybrid CNN-BiLSTM model. Table 2 presents the results of the Hybrid CNN-BiLSTM development.

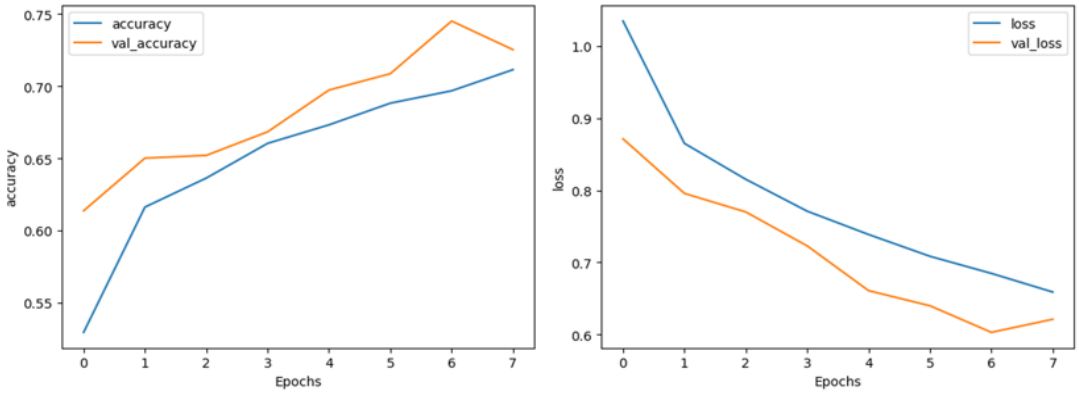
Tabel 2. Result of Hybrid CNN-BiLSTM Testing

Early Stopping	Accuracy	Precision	Recall	F1-Score
With Early Stopping	73 %	77 %	73 %	72 %
Without Early Stopping	99 %	99 %	99 %	99 %

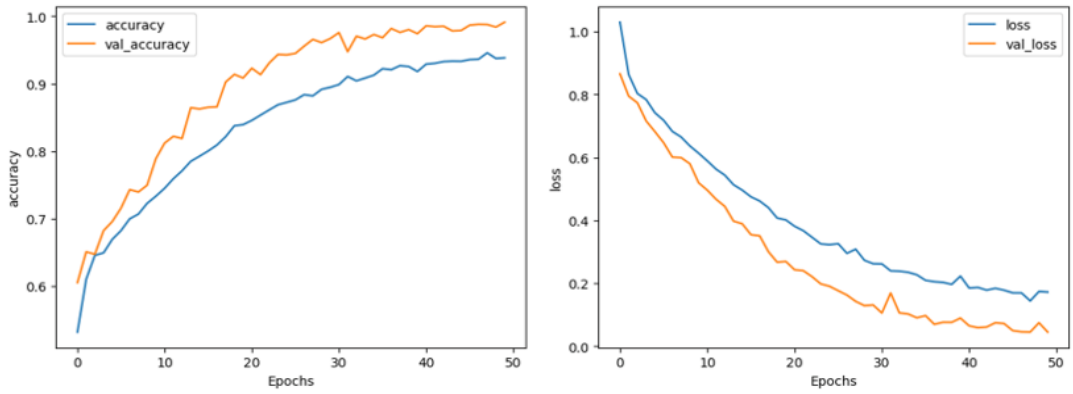
https://doi.org/10.56294/dm2025726

Table 2 shows the performance comparison between the Hybrid CNN-BiLSTM model with and without using Early Stopping, which is evaluated based on metrics such as precision, recall, F1-score, and accuracy. The application of Early Stopping resulted in 73 % precision, 77 % recall, 73 % F1-score, and 72 % accuracy. This technique effectively reduces overfitting by stopping the training process when no further improvement is seen on the validation dataset. However, the slightly lower performance compared to the model trained without Early Stopping implies that the training may have been stopped before the model could fully capture the complex patterns in the training data.

In contrast, when Early Stopping was not used, the model showed significantly improved metrics, achieving precision, recall, F1-score, and accuracy of 99 % each. This suggests that the extended training duration allows the model to better understand the patterns in the dataset. However, such near-perfect performance raises concerns about overfitting, where the model becomes too dependent on the training data, potentially compromising its ability to generalize to unseen data. This evaluation underscores the trade-offs involved in using Early Stopping as a regularization strategy during the training of deep learning models. The next stage is to validate the results through data visualization.



6a. Hybrid CNN-BiLSTM with Early Stopping



6b. Hybrid CNN-BiLSTM without Early Stopping

Figure 6. Plotting Graph of Hybrid CNN-BiLSTM Development Testing

Figure 6 illustrates the performance comparison of the Hybrid CNN-BiLSTM model in two training scenarios: with early stopping (6a) and without early stopping (6b). In figure 6a, the model was trained using an early stopping mechanism that halts training after 8 epochs. The accuracy graph indicates a steady improvement in both training accuracy and validation accuracy, with both converging toward values close to 0,70. The loss graph also demonstrates a consistent decrease in both training loss and validation loss, with only a slight difference between the two. These results suggest that the implementation of early stopping effectively prevented overfitting by terminating training before the model began to overfit on the training data, although the overall accuracy remained relatively lower compared to the scenario without early stopping.

In figure 6b, the model was trained without early stopping, allowing training to continue for up to 50 epochs. The accuracy graph shows a significant increase, reaching values close to 0,95 for both training and validation. However, after approximately 40 epochs, a slight divergence between training accuracy and validation accuracy was observed, indicating early signs of overfitting. The loss graph shows a stable downward trend for both

training and validation losses, but the validation loss began to plateau after around 40 epochs, suggesting that further training does not yield significant improvements in validation performance.

The results of the development and testing of the Hybrid CNN-BiLSTM model demonstrate that the integration of CNN and BiLSTM architectures with pre-trained embeddings such as FastText can provide more semantic word representations, thereby enhancing the model's performance in text classification tasks. This model was designed with a robust regularization approach, incorporating techniques such as Dropout and early stopping mechanisms to mitigate overfitting and ensure generalization capabilities on test data. The testing revealed that the model employing early stopping achieved an accuracy of 73 %, precision of 77 %, recall of 73 %, and an F1-score of 72 %. Although its performance was lower compared to the model without early stopping, this technique successfully prevented overfitting by halting training at an optimal point before the model began to learn overly specific patterns from the training data. This is evident in the accuracy and loss trends depicted in the graphs, where the model exhibits stable performance on the validation data.

Conversely, the model trained without early stopping yielded significantly higher performance, with accuracy, precision, recall, and F1-score each reaching 99 %. This result indicates that the model had sufficient training time to learn the data patterns in depth. However, this nearly perfect performance also suggests a risk of overfitting, particularly when applied to more diverse datasets or different domains. The plotting graphs show a continuously increasing accuracy trend. However, a divergence between training and validation accuracy began to emerge after a certain epoch, indicating the onset of overfitting. The use of pre-trained embeddings such as FastText enables the model to comprehend word relationships in a broader context, especially for data with high linguistic variation. The combination of CNN architecture for capturing local patterns and BiLSTM for temporal relationships offers flexibility in handling complex text data. Overall, the model demonstrates efficient parameterization, with a total of 3 262 660 parameters, of which only 262 660 parameters are trainable, while 3 000 000 parameters are frozen embeddings.

Generally, the results of this testing indicate that the early stopping mechanism and the Hybrid CNN-BiLSTM architecture are effective for text classification tasks. However, there exists a trade-off between high performance without early stopping, which carries the risk of overfitting, and more stable performance with the use of early stopping. The selection of this approach should be aligned with the specific application requirements and the characteristics of the data utilized. This research also has higher accuracy compared to previous research if done without early stopping on the CNN-LSTM hybrid. Table 2 is a comparison with previous research.

Table 3. Comparison of Previous Research

Researchers	Architecture	Accuracy
Anam et al. (2024)[23]	Input → Word Embedding → GRU → Dropout → BiLSTM → Dropout → Max-Pooling → Dense → Output	89 %
Riyadi & Jasmir (2023)[24]	Input → Word Embedding → Conv → Max-Pooling → Dropout → BiLSTM → BiLSTM → Fully Connected → Output	97 %
Lu et al. (2021)[25]	Input → Conv → Pooling → BiLSTM → AM Layer → Output	98 %
Abdelhady et al. (2024)[26]	Input → Word embedding → CNN → Dropout → Stacked BiLSTM Layer → Dropout → Softmax+Dense Layer	80 %
Nie et al. (2021)[27]	Input → Word Embedding → Convolution Layer → BiLSTM Layer → Attention Layer → Output	98 %
Li et al. (2024)[28]	Input → Conv → Max-Polling → Conv → Max-Polling → Conv → Max-Polling → Conv → Max-Polling → Conv → Max-Polling → BiLSTM → BiLSTM → FC → Softmax	95 %
Sari et al. (2024)[29]	Input → Embedding Layer → BiLSTM → Conv1D → Max-Polling → BiLSTM → BiLSTM → Dense → Dropout → output	96 %
This Research	Input →Embedding → dropout → conv1d → Max pooling → dropout → conv1d → Max pooling → dropout → BiLSTM → dropout → dense → dropout → dense	99 %

The table compares various research studies based on their architectures and the accuracy achieved in text classification tasks. Study Anam et al. (2024) Using a combination of GRU and BiLSTM with dropout and max-pooling layers, resulted in an accuracy of 89 %.[23] This architecture uses GRU to capture simple sequential dependencies before proceeding with BiLSTM to capture more complex temporal dependencies. Study Riyadi & Jasmir (2023) combined a convolution layer (Conv) with BiLSTM and Fully Connected Layer, along with dropout and max-pooling. This architecture achieved 97 % accuracy, demonstrating the power of combining convolution layers for spatial features and BiLSTM for temporal dependencies.[24]

Study Lu et al. (2021) Adding an Attention Mechanism (AM layer) after the BiLSTM, with a simple yet effective architecture, resulted in 98 % accuracy. Attention helps the model to focus on important features of the input. [25] Similarly, study Abdelhady et al. (2024) Using CNN with dropout and stacked BiLSTM, with softmax and dense layers as output, resulted in a lower accuracy of 80 %.[26] This architecture focuses on spatial features through CNN, but the complexity of the data seems to require further optimization. Study Nie et al. (2021) Integrating the Attention Layer after the BiLSTM, combined with the Convolution Layer, achieved 98 % accuracy. The Attention mechanism helps the model to focus on the important part of the data.[27]

Study Li et al. (2024) Utilizing multiple convolution layers with max-pooling iteratively before BiLSTM, resulted in 95 % accuracy. This structure tries to improve the feature capture ability but may sacrifice efficiency. [28] Study Sari et al. (2024) Combining BiLSTM with Conv1D and max-pooling in a complex structure, resulting in 96 % accuracy. This combination optimizes both spatial and temporal data processing.[29]

Then in the Research conducted Using a more sophisticated approach with a combination of embedding, dropout, Conv1D, max-pooling, BiLSTM, and dense layers, resulting in the highest accuracy of 99 %. This model demonstrates efficiency by making the most of the advantages of each component. In conclusion, this table illustrates how various combinations of deep learning architectures are applied to improve accuracy, where this research achieves the best performance with optimal integration of embedding, convolution, and BiLSTM layers, as well as regulation using dropout.

CONCLUSIONS

This research demonstrates that the Hybrid CNN-BiLSTM architecture using FastText embeddings provides more semantic word representations, which enhances text classification performance. The application of regularization mechanisms such as Dropout helps to mitigate the risk of overfitting, while the integration of CNN and BiLSTM effectively captures both local and temporal patterns within textual data. The model tested without the Early Stopping mechanism achieved a very high accuracy of 99 %, despite an associated risk of overfitting, whereas the model employing Early Stopping exhibited a more stable accuracy of 73 % with improved generalization capabilities. These results highlight the importance of considering the trade-off between accuracy, stability, and the risk of overfitting in the design of deep learning model architectures.

This research contributes significantly to the development of deep learning models for text classification. However, future research could address several limitations. For instance, further exploration could be conducted to incorporate additional regularization mechanisms, such as the use of learning rate schedulers or weight regularization, to enhance generalization without compromising accuracy. Moreover, testing the model across various domains and in other languages is crucial for ensuring broader generalization capabilities. The implementation of modern embedding techniques, such as Transformer-based embeddings (e.g., BERT or GPT), could also be compared with FastText to assess their impact on model performance. Therefore, future research may expand the utility of this model across various natural language processing applications.

REFERENCES

1. Anam MK, Defit S, Haviluddin H, Efrizoni L, Firdaus MB. Early Stopping on CNN-LSTM Development to Improve Classification Performance. J Appl Data Sci. 2024;5(3):1175–1188. https://doi.org/10.47738/jads.v5i3.312

2. Jim JR, Talukder MAR, Malakar P, Kabir MM, Nur K, Mridha MF. Recent advancements and challenges of NLP-based sentiment analysis: A state-of-the-art review. Natural Language Processing Journal. 2024;6(100059):1-30. https://doi.org/10.1016/j.nlp.2024.100059

3. Bharal R, V Vamsi Krishna O. Social Media Sentiment Analysis Using CNN-BiLSTM. International Journal of Science and Research. 2021;10(9):656–661. https://doi.org/10.21275/sr21913110537

4. Gupta B, Prakasam P, Velmurugan T. Integrated BERT embeddings, BiLSTM-BiGRU and 1-D CNN model for binary sentiment classification analysis of movie reviews. Multimed Tools Appl. 2022; 81:33067–33086. https://doi.org/10.1007/s11042-022-13155-w

5. Sun F, Chu N. Text Sentiment Analysis Based on CNN-BiLSTM-Attention Model. In: International Conference on Robots & Intelligent System (ICRIS). Sanya, China: IEEE; 2020:749–752. https://doi.org/10.1109/ICRIS52159.2020.00186

6. Oyewale CT, Akinyemi JD, Ibitoye AOJ, Onifade OFW. Predicting Suicide Ideation from Social Media Text Using CNN-BiLSTM. Soft Computing and Its Engineering Applications. 2024: 274–286. https://doi.org/10.1007/978-3-031-53731-8_22

7. Xiaoyan L, Raga RC, Xuemei S. GloVe-CNN-BiLSTM Model for Sentiment Analysis on Text Reviews. Journal of Sensors. 2022:1–12. https://doi.org/10.1109/ICCWAMTIP53232.2021.9674171

8. Asudani DS, Nagwani NK, Singh P. Impact of word embedding models on text analytics in deep learning environment: a review. Artif Intell Rev. 2023;56(9):10345–10425. https://doi.org/10.1007/s10462-023-10419-1

9. E.Almandouh M, Alrahmawy MF, Eisa M, Elhoseny M, Tolba AS. Ensemble based high performance deep learning models for fake news detection. Sci Rep. 2024;14(26591):1-24. https://doi.org/10.1038/s41598-024-76286-0

10. Akgül İ. A Pooling Method Developed for Use in Convolutional Neural Networks. CMES- Computer Modeling in Engineering and Sciences. 2024;141(1):751–770. https://doi.org/10.32604/cmes.2024.052549

11. Nissa NF, Janiati A, Cahya N, Anton A, Astuti P. Application of Deep Learning Using Convolutional Neural Network (CNN) Method For Women's Skin Classification. Scientific Journal Informatics. 2021;8(1):144–153. https://doi.org/10.15294/sji.v8i1.26888

12. Grimm D, Tollner D, Kraus D, Török Á, Sax E, Szalay Z. A numerical verification method for multi-class feed-forward neural networks. Expert Systems with Applications. 2024;247(123345):1-15. https://doi.org/10.1016/j.eswa.2024.123345

13. Eang C, Lee S. Improving the Accuracy and Effectiveness of Text Classification Based on the Integration of the Bert Model and a Recurrent Neural Network (RNN_Bert_Based). Applied Sciences. 2024;14(18):1-26. https://doi.org/10.3390/app14188388

14. Salehin I, Kang DK. A Review on Dropout Regularization Approaches for Deep Neural Networks within the Scholarly Domain. Electronics. 2023;12(14):1-23. https://doi.org/10.3390/electronics12143106

15. Saeed AM. An Automated New Approach in Fast Text Classification: A Case Study for Kurdish Text. Science Journal of University of Zakho. 2024;12(3):329-335. https://doi.org/10.25271/sjuoz.2024.12.3.1296

16. Zhao X, Wang L, Zhang Y, Han X, Deveci M, Parmar M. A review of convolutional neural networks in computer vision. Artif Intell Rev. 2024;57(99):1-43. https://doi.org/10.1007/s10462-024-10721-6

17. Zafar A, Aamir M, Mohd Nawi N, Arshad A, Riaz S, Alruban A, et al. A Comparison of Pooling Methods for Convolutional Neural Networks. Applied Sciences. 2022;12(17):1-21. https://doi.org/10.3390/app12178643

18. Fan Y, Tang Q, Guo Y, Wei Y. BiLSTM-MLAM: A Multi-Scale Time Series Prediction Model for Sensor Data Based on Bi-LSTM and Local Attention Mechanisms. Sensors. 2024;24(12):1-16. https://doi.org/10.3390/s24123962

19. Zhang D, Leng J, Li X, He W, Chen W. Three-Stream and Double Attention-Based DenseNet-BiLSTM for Fine Land Cover Classification of Complex Mining Landscapes. Sustainability. 2022 Sep 30;14(19):12465. https://doi.org/10.3390/su141912465

20. Bałazy K, Struski Ł, Śmieja M, Tabor J. r-softmax: Generalized Softmax with Controllable Sparsity Rate. arXiv; 2023:1-15. https://doi.org/10.48550/arXiv.2304.05243

21. Reyad M, Sarhan AM, Arafa M. A modified Adam algorithm for deep neural network optimization. Neural Comput & Applic. 2023;35(23):17095–17112. https://doi.org/10.1007/s00521-023-08568-z

22. Anam MK, Van Fc LL, Hamdani H, Rahmaddeni R, Junadhi J, Firdaus MB, et al. Sara Detection on Social Media Using Deep Learning Algorithm Development. Journal of Applied Engineering and Technological Science (JAETS). 2024;6(1):225–237. https://doi.org/10.37385/jaets.v6i1.5390

23. Anam MK, Munawir M, Efrizoni L, Fadillah N, Agustin W, Syahputra I, et al. Improved Performance of Hybrid GRU-BiLSTM for Detection Emotion on Twitter Dataset. J Appl Data Sci. 2024;6(1):354–365. https://doi.org/10.47738/jads.v6i1.459

24. Riyadi W, Jasmir J. Prediction Performance of Airport Traffic Using BiLSTM and CNN-Bi-LSTM Models. jitk. 2023;9(1):1–7. https://doi.org/10.33480/jitk.v9i1.4191

25. Lu W, Li J, Wang J, Qin L. A CNN-BiLSTM-AM method for stock price prediction. Neural Comput & Applic. 2021;33(10):4741–4753. https://doi.org/10.1007/s00521-020-05532-z

26. Abdelhady N, Hassan A. Soliman T, F. Farghally M. Stacked-CNN-BiLSTM-COVID: an effective stacked ensemble deep learning framework for sentiment analysis of Arabic COVID-19 tweets. J Cloud Comp. 2024;13(85):1-21. https://doi.org/10.1186/s13677-024-00644-6

27. Nie Q, Wan D, Wang R. CNN-BiLSTM water level prediction method with attention mechanism. In: International Conference on Artificial Intelligence Technologies and Applications. 2021:1-9. https://doi.org/10.1088/1742-6596/2078/1/012032

28. Li Z. Exploiting CNN-BiLSTM Model for Distributed Acoustic Sensing Event Recognition. In: Wang Y, editor. In: International Conference on Artificial Intelligence and Communication. Dordrecht: Atlantis Press International BV; 2024: 333–341. https://doi.org/10.2991/978-94-6463-512-6_36

29. Sari WK, Azhar ISB, Yamani Z, Florensia Y. Fake News Detection Using Optimized Convolutional Neural Network and Bidirectional Long Short-Term Memory. ComEngApp. 2024;13(03):25–33. https://doi.org/10.18495/comengapp.v13i03.492

FINANCING

None.

CONFLICT OF INTEREST

None.

AUTHORSHIP CONTRIBUTION

Conceptualization: Ade Oktarino.
Research: Ade Oktarino and Sarjon Defit.
Methodology: Ade Oktarino and Yuhandri.
Software: Ade Oktarino.
Drafting - original draft: Ade Oktarino and Sarjon Defit.
Writing - proofreading and editing: Ade Oktarino, Sarjon Defit, Yuhandri.