

Comparative analysis of word embedding features to improve the performance of deep learning models on social media data

Jasmir Jasmir¹, Pareza Alam Jusia², Yulia Arvita², Gunardi Gunardi³

¹Department of Computer Engineering, Faculty of Computer Science, Universitas Dinamika Bangsa, Jambi, Indonesia

²Department of Informatic Engineering, Faculty of Computer Science, Universitas Dinamika Bangsa, Jambi, Indonesia

³Department of Information System, Faculty of Computer Science, Universitas Dinamika Bangsa, Jambi, Indonesia

Article Info

Article history:

Received Aug 25, 2024

Revised Apr 10, 2025

Accepted May 27, 2025

Keywords:

Deep learning

Feature extraction

Media social data

Sentiment analysis

Word embedding

ABSTRACT

In this study, we apply various deep learning methods incorporating word embedding features to evaluate their impact on improving classification performance in sentiment analysis. The methods employed include conditional random field (CRF), bidirectional long short term memory (BLSTM), and convolutional neural network (CNN). Our experiments utilize social media data from restaurant review. By testing different iterations of these deep learning techniques with various word embedding features, we found that the BLSTM algorithm achieved the highest accuracy of 80.00% before integrating word embedding features. After incorporating word embeddings, the BLSTM with the word2vec feature achieved an accuracy of 87.00%. Notably, the CNN showed a significant improvement with the FastText feature. Considering all evaluation metrics—accuracy, precision, recall, and F1-score—the BLSTM algorithm consistently demonstrated the best performance across different word embeddings.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Jasmir Jasmir

Department of Computer Engineering, Faculty of Computer Science, Universitas Dinamika Bangsa

St. Jendral Sudirman, Tehok, South Jambi, Jambi, Indonesia

Email: ijay_jasmir@yahoo.com

1. INTRODUCTION

A recent development that offers an alternative to conventional data collection methods is the inclusion of social media data [1], [2]. Social media data collection is seen to be efficient in a number of ways. This efficiency includes lower data gathering costs, real-time data availability, and the production of precise information that more closely represents public opinion [3], [4]. The process of examining public responses and opinions using social media data is known as sentiment analysis [5], [6].

In this increasingly technologically advanced condition, computer-based text data is also increasing and mushrooming and flooding our social media pages. Every second we get and witness the emergence of many new web pages, along with the emergence of news articles, magazines, and scientific papers that continue without stopping, especially on social media platforms. This surge has certainly resulted in a lot of textual content available in digital form and is very easy to get and access [7], [8]. With the large number of digital texts that can be accessed freely and the demand for flexible access continues to increase, the task of classifying texts becomes very important and very necessary [9]. However, there are several things that need to be studied, that one of the main challenges in such a task lies in the large dimensionality of the feature space [10]. Most of these features have proven to be no longer appropriate or can be detrimental to classification accuracy, which requires the identification and combination of more relevant features to improve performance [11].

Given the vast amount of unstructured information available on the web, collecting and organizing data is a challenging task. It necessitates the use of automated methods to assist researchers in gathering and

analyzing sentiment-related information [12]. The object of sentiment analysis can be speech, text and images. Here we use a restaurant review dataset which is usually presented in text form, so sentiment analysis in most papers focuses on text-based sentiment analysis.

Sentiment analysis, a branch of natural language processing (NLP), utilizes machine learning techniques to detect and extract factual information from text [13]. This involves recognizing emotional subtleties and assessing the overall sentiment—positive, neutral, or negative—expressed by the author. Applying sentiment analysis to large text datasets, like social media posts or user comments, enables thorough examination [14]–[16].

In NLP, computers do not naturally understand textual language, so methods for transforming words into vectors are essential for interpretation. The development of word vector representation continues to be a significant research focus. This representation is crucial because it greatly impacts the precision and effectiveness of the resulting learning models. Word representation techniques fall under the category of feature engineering. Due to the unstructured nature of text, feature engineering in textual data presents unique challenges. One widely adopted strategy for this is the word embedding feature [17], [18].

This word embedding feature is combined with various classification methods. Numerous classifiers are commonly employed for sentiment analysis, with machine learning [19], [20] and deep learning [21], [22] being frequently utilized. In this research, deep learning methods such as conditional random field (CRF) [23], bidirectional long short term memory (BLSTM) [24] and convolutional neural network (CNN) [25] are used. CRFs are used to construct probabilistic models for sequential data segmentation and labeling. Being conditional, CRFs facilitate easy inference and help avoid label bias issues. BLSTM is utilized to capture information from both past and future contexts. CNN, on the other hand, is employed to assess processing capabilities and evaluate classification performance on text data.

We assessed the effectiveness of various deep learning methods by evaluating their performance with different word embedding features. The word embedding features tested include Word2Vec [26], global vectors for word representation (GloVe) [27], and FastText [28]. Experiments were conducted on a sentiment analysis dataset derived from review restaurant data.

Subha *et al.* [29] conducted a study highlighting the importance of a deeper understanding of how information technology (IT) investments provide value to companies, given that many studies still focus on the correlation between IT spending and financial returns. To address this challenge, a methodology based on three main stages was developed: data preparation, feature selection, and model training. In the data preparation stage, irrelevant and less valuable features were cleaned. Feature selection was performed using optimized principal component analysis (PCA), while model training utilized a combination of CRF-BiLSTM-CNN. Compared to CNN and BiLSTM models separately, this approach showed superior performance with a success rate of 98.87%. Meanwhile, Khalid *et al.* [30] conducted a study developing an RNN-BiLSTM-CRF ensemble model to improve the accuracy of electricity theft detection, overcoming the limitations of conventional approaches that only rely on one-dimensional (1-D) data. This model utilizes a combination of 1-D and 2-D electricity consumption data, and integrates the strengths of RNN and BiLSTM architectures to capture complex consumption patterns. Experimental results show that this approach achieves an accuracy of 93.05%, outperforming previous methods and showing great potential in a more reliable electricity theft detection system. Gupta *et al.* [31] conducted a study proposing a hybrid CNN-LSTM model for fake news detection in Hindi, using FastText embedding and a combination of Conv1D and LSTM. The model achieved 97% accuracy on newly collected data and 89% F1-score on CONSTRAINT2021 data. The study also introduced a new dataset and pioneered the field of fake news detection in Hindi.

Luo and Xu [32] have conducted similar research, namely the use of deep learning methods on restaurant reviews during the COVID-19 era using BLSTM and simple embedding + average pooling, but have not specifically discussed the significant increase in value. This study examines changes in restaurant review trends on Yelp during the COVID-19 pandemic, but has several limitations related to data without visit dates, limited location coverage, the use of black-box deep learning models, and a short observation period. They recommend conducting further studies by expanding the observation period and location coverage, and evaluating the model on various platforms.

Referring to the above studies, and seeing the many opportunities and challenges in research on the use of word embedding features to improve the evaluation value of classification performance, we continue the research by using word embedding features that can improve the evaluation value of text classification performance for several deep learning methods which are also contributions to this research. This research uses a sentiment analysis dataset on restaurant reviews.

2. METHOD

To achieve research results that are in accordance with the research objectives, a series of important steps are prepared to develop this model to be better. The steps in question are presented in Figure 1. The process starts from collecting datasets taken from the kaggle.com site, then entering the preprocessing stage. After the preprocessing stage, we conducted 2 experiments, the first experiment, the process of entering the deep learning stage and the classification performance evaluation stage. The second experiment, before the deep learning stage, this process passes through the word embedding feature and is continued to the classification performance evaluation stage, at the end is to compare the results of the two experiments.

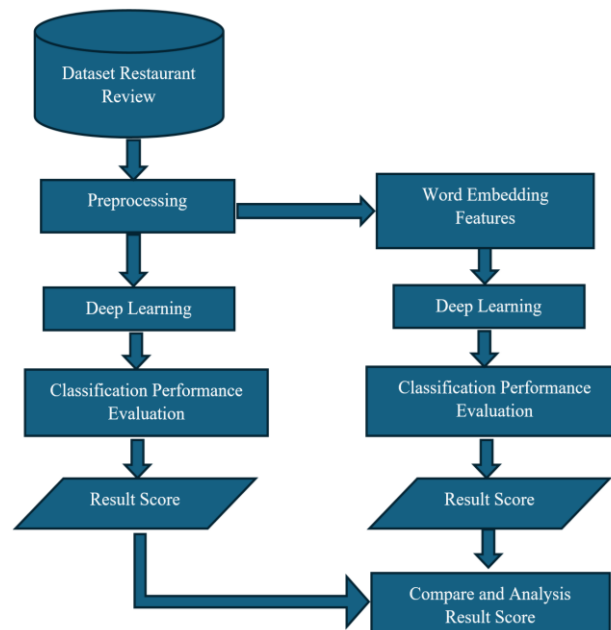


Figure 1. Research framework

2.1. Dataset

The selection of a dataset is determined by the type of data to be processed, which involves searching for existing data and obtaining any additional necessary data. Once collected, the data is integrated into the dataset. For this research, a restaurant review dataset was chosen. This dataset comprises 200 reviews, with an equal split of 100 positive reviews and 100 negative reviews [33].

2.2. Preprocessing

After getting data from restaurant reviews, the data is cleaned and transformed into the desired form before modeling. This step is important to ensure that the data used by the sentiment classification model is clean, structured, and ready for analysis. The dataset used consists of only 100 positive reviews and 100 negative reviews. This dataset undergoes preprocessing through three different processes: tokenization, stopwords removal, and stemming.

2.3. Word embedding

Each word is represented as a low-dimensional numerical vector. Word embedding allows for capturing semantic nuances from large text corpora, making it essential for various NLP tasks to achieve optimal word representation. Several algorithms are available for word embedding, including GloVe, Word2Vec, and FastText. In this research, we employ pre-trained models that incorporate all three features.

2.3.1. Global vectors for word representation

GloVe utilizes co-occurrence and matrix factorization to generate its vectors. It focuses on identifying statistical relationships between words. Initially, GloVe creates a large matrix of words and contexts, capturing information about their co-occurrence [34], [35]. In this case, once the co-occurrence matrix is formed, GloVe calculates the co-occurrence proportion to capture the contextual relationships between words. GloVe relies on the probability distribution of co-occurrence between words to understand the strength of semantic associations.

2.3.2. Word2Vec

Word2Vec leverages word occurrences in text to identify connections between them. Word2Vec operates in two ways: context prediction, which predicts surrounding words based on a given word, and context-based prediction (Bag-of-Words), which predicts words given a context. Essentially, Word2Vec takes a text corpus as input and produces word vectors as output [36], [37]. The process begins with a text corpus, where Word2Vec forms relationships between words within a given context window. The context window determines the number of words around a central word that will be considered for learning the word relationships. Next, the model is trained using the skip-gram or continuous bag of words (CBOW) algorithm. Both of these algorithms produce word vector representations that maximize the probability of relationships between words based on context. After training, Word2Vec produces a fixed-dimensional vector for each word. These vectors capture semantic relationships between words, where words that frequently appear together in the same context will have similar vector representations.

2.3.3. FastText

Similar to Word2Vec, FastText starts by forming a text corpus and defining a context window for each word. The context window determines how many words around the center word will be considered during training [27], [38]. FastText uses character n-grams to represent words. This means that each word is considered as a collection of several character n-grams. In this way, FastText can capture morphological information and the internal structure of words. After forming the n-grams, FastText trains the model using the skip-gram or CBOW approach, similar to Word2Vec. The model tries to predict the context word based on the center word (or vice versa), but taking into account the n-grams that make up each word. After training, FastText generates word vector representations based on a combination of character n-gram vectors. This provides the advantage of capturing the meaning of words that have similar suffixes or prefixes, as well as capturing more subtle patterns in the text.

2.4. Deep learning

To test this research, we used several deep learning methods. The deep learning methods we use are CRF, BiLSTM, and CNN. Information about each method can be seen in the following section.

2.4.1. Conditional random fields

CRF belong to a class of discriminative models ideally suited for classification tasks wherein the current classification is impacted by contextual factors or adjacent states [39], [40]. CRF finds application in named entity recognition [41], part-of-speech tagging, gene prediction, noise reduction, and object detection tasks. Discriminative models, also known as conditional models, are a subset of models commonly employed in statistical classification, particularly in supervised machine learning. Discriminative classifiers aim to model the observed data exclusively, learning classification from provided statistics. Approaches in supervised learning are typically classified into discriminative models or generative models. Discriminative models, in contrast to generative models, make fewer assumptions about distributions and place greater reliance on data quality [42], [43].

2.4.2. Bidirectional long short-term memory

Derived from the recurrent neural network (RNN), BLSTM enhances the RNN architecture by introducing a "gateway" mechanism to regulate the flow of data [44], [45]. Primarily, the LSTM architecture comprises memory cells along with input, output, and forget gates. These elements are structured into a chain-like arrangement composed of RNN modules, which enables the smooth transfer of memory cells along the chain. Moreover, three separate gates are integrated to oversee and regulate the inclusion or inhibition of information into the memory cell [46], [47].

2.4.3. Convolutional neural network

CNN is a form of regulated feed-forward neural network that autonomously learns feature engineering via the optimization of filters, also known as kernels. Unlike lower layer features, higher layer features are extracted from a broader context window. CNNs are sometimes called shift invariant or space invariant artificial neural networks (SIANN) because of their architecture, which involves convolution kernels or filters with shared weights moving across input features. This movement produces a feature map that is equivalent to translation. However, despite the terminology, many CNNs are not inherently translation invariant, mainly because of the down sampling operation applied to the input [48], [49].

3. RESULTS AND DISCUSSION

This section summarizes the results and discussions from experiments conducted based on the research framework described earlier. The experiments focus on analyzing social media text data using various deep learning methods combined with word embedding features, following an 80:20 validation split. The research involved testing deep learning models with different word embedding variations. The deep learning methods employed for sentiment classification in this study include CRF, BLSTM, and CNN. Three types of word embeddings were utilized: Word2Vec, GloVe, and FastText.

Table 1 explains the CRF confusion matrix with three word embedding features and one without features. In CRF without features, the results obtained are TP=82, FP=25, FN=20, and TN=73. The TP value is quite high, indicating the model's ability to detect positive data is relatively good. However, on the other hand, the FP value is quite high, indicating the model often incorrectly detects negative data as positive. FN is still quite a lot, meaning the model loses some positive data. Then CRF with Word2Vec, the results obtained are TP=84, FP=20, FN=23, and TN=73. Word2Vec as a feature representation can improve the accuracy of positive detection compared to without features. The FP value can be seen to decrease from 25 to 20, meaning the ability to distinguish negative data increases. While on the other hand, FN increases to 23, indicating more positive data loss than without features. Furthermore, CRF with GloVe has a value of TP=81, FP=20, FN=19, and TN=80. The model performance is more balanced in detecting positive and negative data. This can be seen in the FN value decreasing to 19, indicating an increase in the ability to detect positive data, the TN value increasing to 80, and the TP value slightly lower than Word2Vec (81 vs. 84), this means the ability to distinguish negative data is better. Then CRF with FastText obtained a value of TP=80, FP=18, FN=15, and TN=87. With this value, FastText produces the best results in reducing classification errors. The FN value is the lowest, meaning the model is better at capturing positive data. The FP value is also the lowest, indicating the model's ability to distinguish negative data very well. And the highest TN, confirming that there is an increase in accuracy for negative data. In this section, FastText is proven to be superior to other features because of its ability to capture better semantic relationships of words in classification.

Table 1. Confusion matrix of CRF

Experiment	TP	FP	FN	TN
CRF without feature	82	25	20	73
CRF with Word2Vec	84	20	23	73
CRF with GloVe	81	20	19	80
CRF with FastText	80	18	15	87

Figure 2 shows the performance evaluation of the CRF method with various word embedding and non feature features. The evaluation was carried out using the accuracy, precision, recall, and F1-score metrics. In non feature (without word embedding) accuracy=77.5% was obtained, this shows the basic performance of CRF without the help of features. Precision=76.64%, indicating the level of accuracy of the model in classifying positive data. Recall=80.39%, higher than precision, indicating that the model is better at capturing all positive data even though not all of its classifications are correct. F1-score=78.47%, illustrates the balance between precision and recall. In Word2Vec accuracy increases to=78.5%, indicating a positive contribution of Word2Vec to performance. Precision=80.77%, indicating that Word2Vec improves the model's ability to classify positive data more accurately than non feature. Recall=78.50%, slightly lower than non feature, indicating a slight decrease in model sensitivity. F1-score=79.62%, showing an improvement in the balance of precision and recall. Word2Vec improves the overall performance of CRF especially in precision and F1-score, although recall decreases slightly. In GloVe accuracy increases more significantly to 80.5%. Precision=80.20%, slightly lower than Word2Vec, but still better than non feature. Recall=81.00%, the highest compared to non feature and Word2Vec. F1-score=80.60%, better than Word2Vec and non feature. GloVe provides improvements in all metrics, especially recall, indicating that the model is better at capturing positive data without significantly reducing precision. In FastText accuracy reaches the highest value=83.5%, showing the best performance compared to other methods. Precision=82.86%, showing a high level of accuracy in classifying positive data. Recall=85.29%, also the highest, showing the best sensitivity in detecting positive data. F1-score=84.06%, the best value, indicating the most optimal balance of precision and recall.

In general, the word embedding feature is able to provide a more meaningful text representation, thereby improving the performance of CRF. However, FastText has the advantage of capturing word information, including words not found in the vocabulary, through the subword embedding approach. This helps the CRF model improve all evaluation metrics significantly. FastText provides the best balance between precision and recall, making it an optimal choice for CRF-based classification tasks.

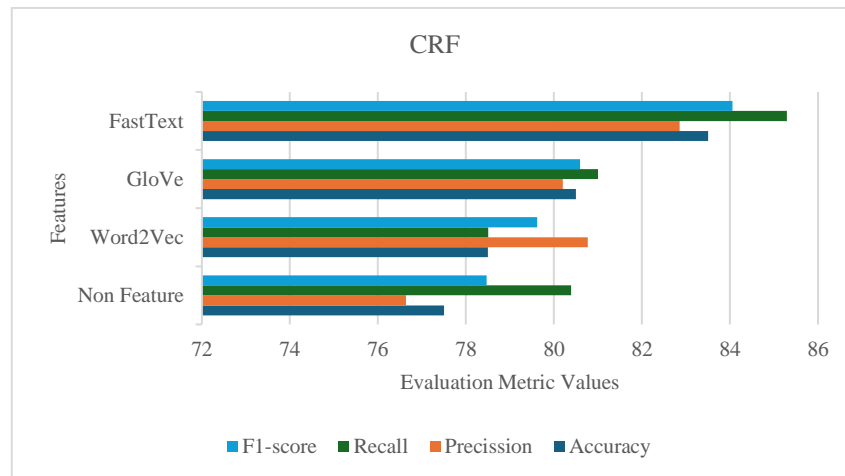


Figure 2. Comparison graph of CRF evaluation values with word embedding

Table 2 shows the experimental results using the BLSTM method with and without various word embedding features. In BLSTM without feature, the values TP=82, FP=18, FN=22, and TN=78 are obtained. This model does not use special feature representation (only utilizes word tokenization), so its performance depends on the ability to understand word sequences without semantic context. On the other hand, FP and FN are quite high, this indicates that the model often makes mistakes in recognizing classes, and has difficulty understanding the relationship between words because there is no context representation feature. In BLSTM with Word2Vec, the values TP=92, FP=14, FN=12, and TN=82 are obtained. Using Word2Vec for vector-based word representation, which captures semantic and contextual relationships between words. TP and FN show the best model in detecting positive classes compared to other methods. The decrease in FP also indicates that the model is more accurate in classification. Word2Vec provides an effective word representation for text data with clear context. In BLSTM with GloVe, the values obtained are TP=84, FP=15, FN=23, and TN=78. Using GloVe for word representation based on co-occurrence matrix, focuses more on global relationships between words. On the other hand, the FN results are higher than Word2Vec, indicating that the model is less than optimal in detecting positive classes. Its performance is slightly better than without features, but still inferior to Word2Vec and FastText. Furthermore, in BLSTM with FastText, TP=84, FP=14, FN=17, and TN=85 are obtained. FastText captures morphological features (prefixes, suffixes) so that it is able to recognize words that are rare or absent in the training data. TN is the highest, indicating that the model is better at detecting negative classes. The combination of TP, FP, FN, and TN results is quite balanced, making FastText a competitive choice. Overall, Word2Vec gives the best results because it has the highest TP and the lowest FN, indicating excellent positive detection. This indicates that Word2Vec semantic representation is very effective for understanding word context. The featureless model shows the worst performance, proving the importance of word representation for improving model accuracy.

Table 2. Confusion matrix of BLSTM

Experiment	TP	FP	FN	TN
BLSTM without feature	82	18	22	78
BLSTM with Word2Vec	92	14	12	82
BLSTM with GloVe	84	15	23	78
BLSTM with FastText	84	14	17	85

Figure 3 shows the evaluation of BLSTM performance in classification with and without using word embedding features. In BLSTM without features, accuracy=80% is obtained. Accuracy shows that only 80% of predictions are correct. This model does not use special feature representation, so it only utilizes word tokenization and linear relationships between words. Precision=82%: the model is quite accurate in predicting the positive class, but there is a significant error rate in FP predictions. Recall=78.85%: the ability to detect positive classes is quite limited, with many positive samples that fail to be recognized (high false negative). F1-score=80.39%: the combination of precision and recall shows adequate overall performance, but is less than optimal compared to the model with embedding. In BLSTM with Word2Vec, accuracy=87.00%. The highest accuracy, indicating that this model is very reliable in making correct predictions. Precision=86.79%:

Word2Vec is able to capture semantic relationships between words, thereby reducing FP predictions. A high precision value indicates that the model is very accurate in recognizing positive classes. Recall=88.46%: the highest recall among all methods, indicating that Word2Vec is very effective in detecting positive samples, with the least FN. F1-score=87.62%: the combination of precision and recall produces the highest F1-score, confirming that Word2Vec provides the best overall performance. In BLSTM with GloVe, accuracy=81% is obtained. Accuracy is slightly better than the model without features, but far behind Word2Vec and FastText. Precision=84.85%: GloVe is able to produce competitive precision, but is not as good as Word2Vec in reducing false positive predictions. Recall=78.50: low recall indicates that the model often fails to recognize positive samples, with a fairly high number of FNs. F1-score=81.55: the medium F1-score value indicates that although Precision is quite good, overall performance is hampered by low recall. In BLSTM with FastText, accuracy=84.50% is obtained. FastText provides the second highest accuracy, indicating solid overall performance. Precision=85.71: FastText precision shows very good accuracy in recognizing positive class, almost close to Word2Vec. Recall=83.17: FastText recall is better than GloVe and non feature, but still lower than Word2Vec. F1-score=84.42%: the combination of precision and recall produces a fairly high F1-score, close to the performance of Word2Vec.

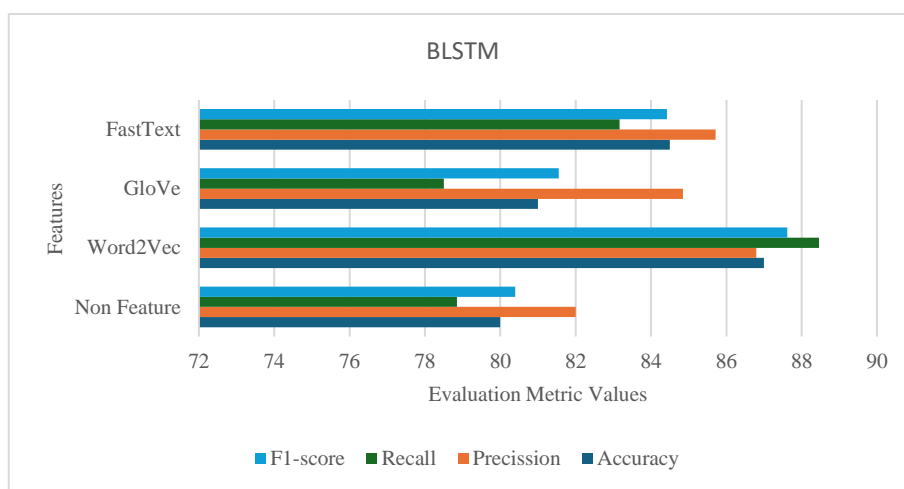


Figure 3. Comparison graph of BLSTM evaluation values with word embedding

Overall Word2Vec gives the best results in all evaluation metrics: strong local semantic representation improves accuracy, precision, recall, and F1-score. Non-feature models perform the lowest: limitations in understanding word relationships lead to significantly lower performance compared to embedding methods.

Table 3 shows the results of CNN evaluation with and without using word embedding features. CNN without features produces a value of TP=70, FP=31, FN=22, and TN=77. Has adequate baseline performance without additional features. Able to recognize 70 positive samples correctly. However, the number of FP is high, indicating that many negative samples are misclassified as positive. The number of FN is significant, indicating that the model often fails to recognize positive samples. CNN without features has limited baseline performance because it does not utilize semantic or morphological representation of words. CNN with Word2Vec produces a value of TP=82, FP=18, FN=21, and TN=79 is able to increase TP to 82, indicating a better ability to recognize positive samples. Reducing FP from 31 to 18, improves prediction accuracy. TN increases to 79, indicating a reduction in false negative predictions. However, FN decreases slightly, but is still higher than FastText. Word2Vec provides significant improvements with strong semantic representation, so that the model is better able to understand the context of the data. CNN with GloVe produces TP=76, FP=23, FN=27, and TN=74. TP increases to 76 compared to the featureless model. FP is lower than the baseline. However, FN increases, indicating many unrecognized positive samples. TN decreases, indicating more negative misclassifications. GloVe improves performance compared to the featureless model, but the results are less than optimal compared to Word2Vec or FastText, especially in detecting positive samples. CNN with FastText produces TP=87, FP=14, FN=19, and TN=80. The highest TP, indicating an excellent ability to recognize positive samples. The lowest FP, indicating that the model is very accurate in avoiding false positive predictions. The lowest FN, indicating the best ability to detect positive samples. The highest TN, indicating the best performance in correctly identifying negative samples. FastText produces the best performance due to its ability to capture word morphology and local context, providing a very rich feature representation.

Table 3. Confusion matrix of CNN

Experiment	TP	FP	FN	TN
CNN without feature	70	31	22	77
CNN with Word2Vec	82	18	21	79
CNN with Glove	76	23	27	74
CNN with FastText	87	14	19	80

Figure 4 shows the performance evaluation of CNN with and without using word embedding features. CNN without features produces accuracy=73.50%. Only 73.5% of the model's predictions are correct. This model uses a simple word representation without additional features, so its ability to understand the context of words is limited. Precision=69.31%: only about 69.31% of the positive class predictions are actually correct. The FP rate is quite high, indicating a prediction error on negative samples. Recall=76.09%: the model is able to recognize about 76.09% of all positive samples, but FN is still quite significant. F1-score=72.54%: the combination of precision and recall produces adequate overall performance, but much lower than the model with embedding. This featureless model has the lowest performance because it does not use semantic representation or contextual relationships between words. These results indicate that CNN requires a richer feature representation for optimal performance. CNN with Word2Vec produces accuracy=80.50%. This value shows that the model makes more correct predictions than the baseline. Precision=82.00%: Word2Vec provides a strong semantic representation, thus reducing the number of FPs and increasing the accuracy in predicting positive classes. Recall=79.61%: the ability to detect positive samples is close to optimal, with a reduced number of FNs compared to the baseline. F1-score=80.79%: the combination of precision and recall produces an excellent F1-score, indicating an overall performance improvement. Word2Vec helps CNN capture semantic relationships between words, significantly improving all evaluation metrics compared to the featureless model. CNN with GloVe produces accuracy=75.00%. Accuracy increases slightly compared to the baseline but is still lower than Word2Vec and FastText. Precision=76.77%: the ability to recognize positive classes increases, but is lower than Word2Vec and FastText. Recall=73.79%: the model is less effective in recognizing all positive samples than Word2Vec or FastText, with a still quite high number of FNs. F1-score=75.25%: the combination of precision and recall produces adequate but not optimal overall performance. GloVe improves performance over baselines, but not as well as Word2Vec or FastText. This may be due to the nature of GloVe which focuses more on global representation rather than local context of words. CNN with FastText produces accuracy=83.50%. Showing that this model is most reliable in making correct predictions. Precision=86.14%: FastText gives the highest precision, showing its excellent ability in avoiding FP. Recall=82.08%: high recall shows that the model is very effective in detecting positive samples with the least FN. F1-score=84.06%. The combination of precision and recall produces the highest F1-score, showing the most optimal overall performance.

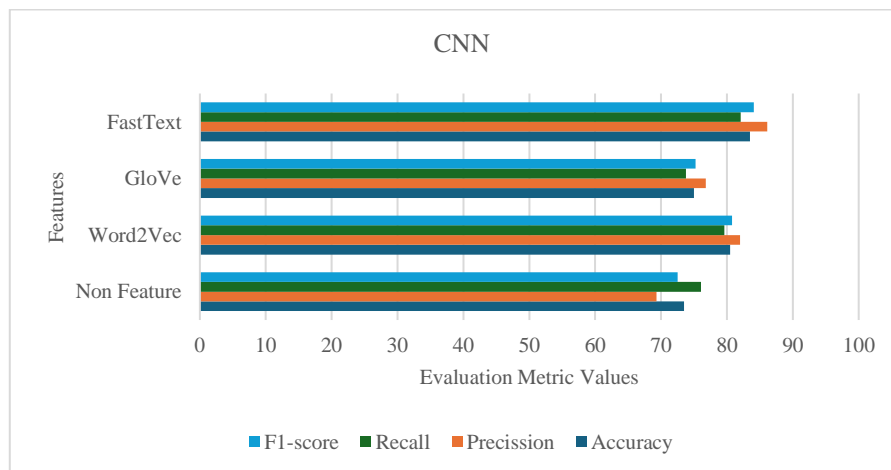


Figure 4. Comparison graph of CNN evaluation values with word embedding

Overall, FastText shows the best performance due to its ability to capture word morphological features (prefixes, suffixes) and local context, so that the model is better able to understand word relationships in the text. FastText (highest performance). Achieves the best results in all evaluation metrics. The ability to capture

word morphology makes FastText superior to other embeddings. Without features (baseline): the lowest performance in all evaluation metrics, showing the limitations of CNN without additional feature representation.

This study examines the impact of better performance, computationally CNN is very efficient, this is due to the use of shared weights and local connections, making it suitable for processing long texts and large datasets. With the word embedding feature, CNN can capture more interactions between features that may be overlooked by CRF and BLSTM. While previous studies have investigated the impact of other features of the same method, the study did not explicitly discuss its effect on computational performance.

In all experiments we found that the BLSTM algorithm achieved the highest accuracy rate of 80% while the CNN algorithm had the lowest accuracy of 73.5% before applying any word embedding features. After combining the word embedding features, the BLSTM algorithm correlated with the Word2Vec feature achieved the highest accuracy of 87%, while CNN with the GloVe feature had the lowest accuracy. The method proposed in this study tends to have a much higher proportion of computational performance compared to the use of other features. All tests still exhibit false positive and false negative errors, and all algorithms utilized their original parameters. This suggests potential for further research to reduce false positives or false negatives and to improve accuracy through hyperparameter tuning. Notably, before using features, CNN had the lowest performance among the three deep learning methods, whereas after feature application, the best results were obtained with the BLSTM using Word2Vec. This discrepancy might be due to the CNN's characteristics being less suited to text data, but its performance improves significantly with the right feature.

Our findings show that BLSTM excels in processing text by analyzing input data from both directions, resulting in strong performance. The proposed method can benefit significantly from the evaluation of classification performance, without affecting the main objective. Additionally, the experiment found that Word2Vec provided the best word embedding, achieving the highest score with the BLSTM. This result aligns well with the operational characteristics of each method, Word2Vec can produce the best accuracy values because of the characteristics of word2vec which is able to represent words into vectors and is very suitable when paired with BLSTM, because the characteristics of BLSTM are compatible algorithms for processing text data. Meanwhile, FastText gives the highest score for the CRF and CNN. While not achieving higher scores than BLSTM, CRF, and CNN benefit significantly from the inclusion of FastText. FastText is able to provide very significant improvements because fast text works based on a collection of sequential words in a text document which contains words, numbers, symbols, and punctuation.

Although improvements have been observed, all models continue to exhibit false positive and false negative results. Future research could address these issues by tuning hyperparameters and investigating advanced pre-processing methods. Additionally, expanding the training dataset with a broader range of samples may further enhance the model's robustness. These findings support the conclusion that FastText's efficiency and subword representation capabilities contribute significantly to its effectiveness in sentiment analysis, as demonstrated in this study.

4. CONCLUSION

Our study has highlighted the effectiveness of pre-trained word embedding models in sentiment analysis. Through a series of experiments, we have demonstrated the ability of these models to achieve high accuracy across a variety of textual datasets. In our evaluation, we tested various deep learning methods with different word embedding features. In this study, we found that the use of word embedding features such as Word2Vec, GloVe, and FastText significantly improved the model performance in text classification compared to no additional features (non feature). FastText consistently outperformed all models (CRF, BLSTM, and CNN) with the highest accuracy, precision, recall, and F1-score, indicating superior contextual representation of words. Word2Vec also provided significant performance improvements, especially favoring BLSTM with the highest accuracy and F1-score. Meanwhile, GloVe provided moderate improvements in all metrics compared to Non Feature but still lagged behind Word2Vec and FastText, likely due to its focus more on global representation rather than local relationships between words. Overall, the choice of word embedding features greatly influences the effectiveness of text classification. In further research, static embeddings (such as Word2Vec, GloVe, and FastText) can be compared or combined with dynamic embeddings such as BERT or GPT to see their impact on CRF, BLSTM, and CNN models.

ACKNOWLEDGEMENTS

The author would like to thank the Yayasan Dinamika Bangsa Jambi for its financial support and use of the laboratory, so that this research could be completed.

FUNDING INFORMATION

This research was funded by the Jambi Dinamika Bangsa Foundation. This financial support aims to help lecturers to be more enthusiastic in improving their functional positions as lecturers to become professors.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Jasmir Jasmir	✓	✓		✓	✓				✓	✓		✓		
Pareza Alam Jusia				✓	✓	✓	✓		✓	✓			✓	✓
Yulia Arvita		✓	✓						✓	✓	✓			
Gunardi Gunardi			✓			✓		✓		✓	✓			

C : **C**onceptualization

M : **M**ethodology

So : **S**oftware

Va : **V**alidation

Fo : **F**ormal analysis

I : **I**nvestigation

R : **R**esources

D : **D**ata Curation

O : **O** Writing - **O**riginal Draft

E : **E** Writing - Review & **E**editing

Vi : **V**isualization

Su : **S**upervision

P : **P**roject administration

Fu : **F**unding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

INFORMED CONSENT

We have obtained informed consent from all individuals included in this study.

ETHICAL APPROVAL

The research related to human use has been complied with all the relevant national regulations and institutional policies in accordance with the tenets of the Helsinki Declaration and has been approved by the authors' institutional review board or equivalent committee.

DATA AVAILABILITY

The data that support the findings of this study are openly available in Kaggle at <https://www.kaggle.com/datasets/vigneshwarsofficial/reviews/versions/1?resource=download> [33].

REFERENCES




- [1] J. Ohme *et al.*, "Digital Trace Data Collection for Social Media Effects Research: APIs, Data Donation, and (Screen) Tracking," *Communication Methods and Measures*, vol. 18, no. 2, pp. 124–141, 2024, doi: 10.1080/19312458.2023.2181319.
- [2] W. M. S. Yafooz, "Enhancing Arabic Dialect Detection on Social Media: A Hybrid Model with an Attention Mechanism," *Information*, vol. 15, no. 6, p. 316, 2024, doi: 10.3390/info15060316.
- [3] A. Babiker, S. Alshakhsi, C. Sindermann, C. Montag, and R. Ali, "Examining the growth in willingness to pay for digital wellbeing services on social media: A comparative analysis," *Heliyon*, vol. 10, no. 11, p. e32467, 2024, doi: 10.1016/j.heliyon.2024.e32467.
- [4] S. M. Fernández-Miguélez, M. Díaz-Puche, J. A. Campos-Soria, and F. Galán-Valdivieso, "The impact of social media on restaurant corporations' financial performance," *Sustainability*, vol. 12, no. 4, pp. 1–14, 2020, doi: 10.3390/su12041646.
- [5] T. A. Khan, R. Sadiq, Z. Shahid, M. M. Alam, and M. M. Su'ud, "Sentiment Analysis using Support Vector Machine and Random Forest," *Journal of Informatics and Web Engineering*, vol. 3, no. 1, pp. 67–75, 2024, doi: 10.33093/jiwe.2024.3.1.5.
- [6] A. Assiri, A. Gumaei, F. Mehmood, T. Abbas, and S. Ullah, "DeBERTa-GRU: Sentiment Analysis for Large Language Model," *Computers, Materials & Continua*, vol. 79, no. 3, pp. 1–10, 2024, doi: 10.32604/cmc.2024.050781.
- [7] A. He and M. Abisado, "Text Sentiment Analysis of Douban Film Short Comments Based on BERT-CNN-BiLSTM-Att Model," *IEEE Access*, vol. 12, pp. 45229–45237, 2024, doi: 10.1109/ACCESS.2024.3381515.
- [8] Z. A. Khan *et al.*, "Developing Lexicons for Enhanced Sentiment Analysis in Software Engineering: An Innovative Multilingual Approach for Social Media Reviews," *Computers, Materials & Continua*, vol. 79, no. 2, pp. 2771–2793, 2024, doi: 10.32604/cmc.2024.046897.
- [9] A. Palanivinaayagam, C. Z. El-Bayeh, and R. Damaševičius, "Twenty Years of Machine-Learning-Based Text Classification: A Systematic Review," *Algorithms*, vol. 16, no. 5, pp. 1–28, 2023, doi: 10.3390/a16050236.
- [10] Z. Zeng, A. Tang, S. Yi, X. Yuan, and Y. Zhu, "A Heuristic Radiomics Feature Selection Method Based on Frequency Iteration and Multi-Supervised Training Mode," *Computers, Materials & Continua*, vol. 79, no. 2, pp. 2277–2293, 2024, doi:

- 10.32604/cmc.2024.047989.
- [11] S. Nurmaini, R. U. Partan, W. Caesarendra, and T. Dewi, "An Automated ECG Beat Classification System Using Deep Neural Networks with an Unsupervised Feature Extraction Technique," *Applied Sciences*, vol. 9, no. 14, 2019, doi: 10.3390/app9142921.
 - [12] M. S. Rahman and H. Reza, "A Systematic Review Towards Big Data Analytics in Social Media," *Big Data Mining and Analytics*, vol. 5, no. 3, pp. 228–244, 2022, doi: 10.26599/BDMA.2022.9020009.
 - [13] M. Furqan and A. F. A. Nasir, "Big Data Approach To Sentiment Analysis in Machine Learning-Based Microblogs: Perspectives of Religious Moderation Public Policy in Indonesia," *Journal of Applied Engineering and Technological Science*, vol. 5, no. 2, pp. 955–965, 2024, doi: 10.37385/jaets.v5i2.4498.
 - [14] M. S. Islam *et al.*, "Challenges and future in deep learning for sentiment analysis: a comprehensive review and a proposed novel hybrid approach," *Artificial Intelligence Review*, vol. 57, no. 3, p. 62, 2024, doi: 10.1007/s10462-023-10651-9.
 - [15] C. Ouni, E. Benmohamed, and H. Ltifi, "Deep learning-based Soft word embedding approach for sentiment analysis," *Procedia Computer Science*, vol. 246, no. C, pp. 1355–1364, 2024, doi: 10.1016/j.procs.2024.09.720.
 - [16] K. Liu, X. Sun, and H. Zhou, "Big data sentiment analysis of business environment public perception based on LTP text classification Take Heilongjiang province as an example," *Heliyon*, vol. 9, no. 10, p. e20768, 2023, doi: 10.1016/j.heliyon.2023.e20768.
 - [17] Z. Zhuang, Z. Liang, Y. Rao, H. Xie, and F. L. Wang, "Out-of-vocabulary word embedding learning based on reading comprehension mechanism," *Natural Language Processing Journal*, vol. 5, pp. 1–6, 2023, doi: 10.1016/j.nlp.2023.100038.
 - [18] B. Wang, A. Wang, F. Chen, Y. Wang, and C. C. J. Kuo, "Evaluating word embedding models: Methods and experimental results," *APSIIPA Transactions on Signal and Information Processing*, vol. 8, pp. 1–14, 2019, doi: 10.1017/ATSIP.2019.12.
 - [19] S. Rapacz, P. Cholda, and M. Natkaniec, "A method for fast selection of machine-learning classifiers for spam filtering," *Electronics*, vol. 10, no. 17, 2021, doi: 10.3390/electronics10172083.
 - [20] R. Passarella, S. Nurmaini, M. N. Rachmatullah, H. Veny, and F. N. N. Hafidzoh, "Development of a machine learning model for predicting abnormalities of commercial airplanes," *Data Science and Management*, vol. 7, no. 3, pp. 256–265, 2024, doi: 10.1016/j.jsamd.2023.100613.
 - [21] R. Newbury *et al.*, "Deep Learning Approaches to Grasp Synthesis: A Review," *IEEE Transactions on Robotics*, vol. 39, no. 5, pp. 3994–4015, 2023, doi: 10.1109/TRO.2023.3280597.
 - [22] A. Tavanaei, M. Ghodrati, S. R. Kheradpisheh, T. Masquelier, and A. Maida, "Deep learning in spiking neural networks," *Neural Networks*, vol. 111, pp. 47–63, 2019, doi: 10.1016/j.neunet.2018.12.002.
 - [23] R. Cotterell and K. Duh, "Low-Resource Named Entity Recognition with Cross-lingual, Character-Level Neural Conditional Random Fields," *arXiv*, 2024, doi: 10.48550/arXiv.2404.09383.
 - [24] G. Liu and J. Guo, "Bidirectional LSTM with attention mechanism and convolutional layer for text classificatio," *Neurocomputing*, 2019, doi: 10.1016/j.neucom.2019.01.078.
 - [25] M. Krichen, "Convolutional Neural Networks: A Survey," *Computers*, vol. 12, no. 8, pp. 1–41, 2023, doi: 10.3390/computers12080151.
 - [26] J. Zhou, Z. Ye, S. Zhang, Z. Geng, N. Han, and T. Yang, "Investigating response behavior through TF-IDF and Word2vec text analysis: A case study of PISA 2012 problem-solving process data," *Heliyon*, vol. 10, no. 16, p. e35945, 2024, doi: 10.1016/j.heliyon.2024.e35945.
 - [27] A. Kumar, A. Q. Md. J. C. Jackson, and C. Iwendi, "Predicting and Curing Depression Using Long Short Term Memory and Global Vector," *Computers, Materials & Continua*, vol. 74, no. 3, pp. 5837–5852, 2023, doi: 10.32604/cmc.2023.033431.
 - [28] I. N. Khasanah, "Sentiment Classification Using fastText Embedding and Deep Learning Model," *Procedia CIRP*, vol. 189, pp. 343–350, 2021, doi: 10.1016/j.procs.2021.05.103.
 - [29] S. B., I. A. K. Shaikh, P. J. Patil, R. Sethumadhavan, M. Preetha, and H. Patil, "Predictive Analysis of Employee Turnover in IT Using a Hybrid CRF-BiLSTM and CNN Model," in *2023 International Conference on Sustainable Communication Networks and Application (ICSCNA)*, Theni, India, 2023, pp. 914–919, doi: 10.1109/ICSCNA58489.2023.10370093.
 - [30] A. Khalid, G. Mustafa, M. R. R. Rana, S. M. Alshahrani, and M. Alymani, "RNN-BiLSTM-CRF based amalgamated deep learning model for electricity theft detection to secure smart grids," *PeerJ Computer Science*, vol. 10, pp. 1–18, 2024, doi: 10.7717/peerj-cs.1872.
 - [31] R. K. Gupta, V. Sharma, R. K. Pateriya, V. Dehalwar, and P. Gupta, "Fake News Detection in Indian Languages: A Case Study with Hindi Using CNN-LSTM," *Procedia Computer Science*, vol. 259, pp. 150–160, 2025, doi: 10.1016/j.procs.2025.03.316.
 - [32] Y. Luo and X. Xu, "Comparative study of deep learning models for analyzing online restaurant reviews in the era of the COVID-19 pandemic," *International Journal of Hospitality Management*, vol. 94, p. 102849, 2021, doi: 10.1016/j.ijhm.2020.102849.
 - [33] Kaggle, "Restaurant Customer Reviews," www.kaggle.com. [Online]. Available: <https://www.kaggle.com/datasets/vigneshwarsofficial/reviews/versions/1?resource=download>, (Accessed: May. 13, 2024).
 - [34] A. George, H. B. B. Ganesh, M. A. Kumar, and K. P. Soman, *Significance of global vectors representation in protein sequences analysis*, vol. 31, 2019, doi: 10.1007/978-3-030-04061-1_27.
 - [35] N. Badri, F. Kboubi, and A. H. Chaibi, "Combining FastText and Glove Word Embedding for Offensive and Hate speech Text Detection," *Procedia Computer Science*, vol. 207, pp. 769–778, 2022, doi: 10.1016/j.procs.2022.09.132.
 - [36] C. A. N. Agustina, R. Novita, Mustakim, and N. E. Rozanda, "The Implementation of TF-IDF and Word2Vec on Booster Vaccine Sentiment Analysis Using Support Vector Machine Algorithm," *Procedia Computer Science*, vol. 234, pp. 156–163, 2024, doi: 10.1016/j.procs.2024.02.162.
 - [37] D. Jatnika, M. A. Bijaksana, and A. A. Suryani, "Word2vec model analysis for semantic similarities in English words," *Procedia Computer Science*, vol. 157, pp. 160–167, 2019, doi: 10.1016/j.procs.2019.08.153.
 - [38] M. Rizwan *et al.*, "Depression Intensity Classification from Tweets Using FastText Based Weighted Soft Voting Ensemble," *Computers, Materials & Continua*, vol. 78, no. 2, pp. 2047–2066, 2024, doi: 10.32604/cmc.2024.037347.
 - [39] D. Zhao, X. Chen, and Y. Chen, "Named Entity Recognition for Chinese Texts on Marine Coral Reef Ecosystems Based on the BERT-BiGRU-Att-CRF Model," *Applied Sciences*, vol. 14, no. 13, p. 5743, 2024, doi: 10.3390/app14135743.
 - [40] Q. Zhang, Y. Cao, and H. Yu, "Parsing citations in biomedical articles using conditional random fields," *Computers in biology and medicine*, vol. 41, pp. 190–194, 2011, doi: 10.1016/j.compbio.2011.02.005.
 - [41] W. Lee, K. Kim, E. Y. Lee, and J. Choi, "Conditional random fields for clinical named entity recognition: A comparative study using Korean clinical texts," *Computers in biology and medicine*, vol. 101, pp. 7–14, 2018, doi: 10.1016/j.compbio.2018.07.019.
 - [42] P. Corcoran, P. Mooney, and M. Bertolotto, "Linear street extraction using a Conditional Random Field model," *Spatial Statistics*, vol. 14, pp. 532–545, 2015, doi: 10.1016/j.spasta.2015.10.003.
 - [43] C. Jiang, M. Maddela, W. Lan, Y. Zhong, and W. Xu, "Neural CRF model for sentence alignment in text simplification," *arXiv*, 2020, doi: 10.48550/arXiv.2005.02324.
 - [44] S. M. Al-Selwi *et al.*, "RNN-LSTM: From applications to modeling techniques and beyond—Systematic review," *Journal of King*




- Saud University-Computer and Information Sciences*, vol. 36, no. 5, p. 102068, 2024, doi: 10.1016/j.jksuci.2024.102068.
- [45] K. S. Tai, R. Socher, and C. D. Manning, "Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks," *arXiv*, 2015, doi: 10.48550/arXiv.1503.00075.
- [46] W. Riyadi and J. Jasmir, "Prediction Performance of Airport Traffic Using Bilstm and Cnn-Bi-Lstm Models," *JITK (Jurnal Ilmu Pengetahuan dan Teknologi Komputer)*, vol. 9, no. 1, pp. 1–7, 2023, doi: 10.33480/jitk.v9i1.4191.
- [47] Z. Dai, X. Wang, P. Ni, Y. Li, G. Li, and X. Bai, "Named Entity Recognition Using BERT BiLSTM CRF for Chinese Electronic Health Records," *2019 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, Suzhou, China, 2019, pp. 1–5, doi: 10.1109/CISP-BMEI48845.2019.8965823.
- [48] J. Zhang, F. Liu, W. Xu, and H. Yu, "Feature fusion text classification model combining CNN and BiGRU with multi-attention mechanism," *Future Internet*, vol. 11, no. 11, p. 237, 2019, doi: 10.3390/fi11110237.
- [49] A. W. Salehi *et al.*, "A Study of CNN and Transfer Learning in Medical Imaging: Advantages, Challenges, Future Scope," *Sustainability*, vol. 15, no. 7, p. 5930, 2023, doi: 10.3390/su15075930.

BIOGRAPHIES OF AUTHORS






Jasmir Jasmir    is senior lecture at Universitas Dinamika Bangsa Jambi, Indonesia. He received his Bachelor in Computer Engineering in 1995 and Master degree in Information Technology in 2006 from Universitas Putra Indonesia YPTK Padang, Indonesia. He receives a Doctor in Informatics Engineering at Universitas Sriwijaya Palembang, Indonesia in 2022. His research interest is data mining, machine learning, and deep learning for natural language processing and its application. He can be contacted at email: ijay_jasmir@yahoo.com.






Pareza Alam Jusia    is a lecture at Universitas Dinamika Bangsa Jambi, Indonesia. He received his Bachelor in Informatics Engineering in Universitas Dinamika Bangsa Jambi in 2011 and Master degree in Computer Science in Universitas Budi Luhur Jakarta, Indonesia in 2015. His research interest is data mining, image processing, and decision support system. He can be contacted at email: parezaalam@gmail.com.



Yulia Arvita    is a lecture at Universitas Dinamika Bangsa Jambi, Indonesia. She received his Bachelor in Informatics Engineering in Universitas Dinamika Bangsa Jambi in 2013 and Master degree in Magister system Information in Universitas Dinamika Bangsa Jambi, Indonesia in 2015. Her research interest is data mining, database, and artificial intelligence. She can be contacted at email: yulia_arvita@yahoo.co.id.



Gunardi Gunardi    is a lecturer at Dinamika Bangsa University, he earned a Bachelor's degree in Computers at UNAMA Jambi in 2009 and a master's degree in information systems at UNAMA in 2014, Indonesia. His research interests are information systems, enterprise systems, knowledge management systems, and data mining. He can be contacted at email: gun4rdi.sj@gmail.com.