Building of Informatics, Technology and Science (BITS)
Volume 8, No 1, June 2026 Page: 550−559
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v8i1.10046

Effect of Word Embedding on Indonesian Social Media Hate Speech Classification Using Hybrid CNN-SimpleRNN

Mas Muhammad Rizqi Adiguna*, Yuliant Sibaroni

Informatic, School of Computing, Telkom University, Bandung, Indonesia
Email: [1,*]adigunaa@student.telkomuniversity.ac.id, [2]yuliant@telkomuniversity.ac.id
Email Penulis Korespondensi: adigunaa@student.telkomuniversity.ac.id
Submitted: **23/05/2026**; Accepted: **30/06/2026**; Published**: 30/06/2026**

Abstract−The rapid rise in the number of people using social media platforms, specifically platform X, poses great difficulties for spotting any possible harm from hate speech. In particular, due to the high informality of discourse of Indonesian internet users, expressed through slang, abbreviation, and distorted spelling, automatic moderation becomes even more complicated. The current study attempts at classifying hate speech on social media platform X, proposing a hybrid model of CNN combined with a Bidirectional SimpleRNN architecture, alongside a comparative study on word embedding approaches. The combination between CNN and SimpleRNN is chosen because it allows for exploiting both CNN's capability to extract local spatial features by finding toxic n-grams and the strength of Bidirectional SimpleRNN for capturing long contextual dependency in the sequence of text data. Given that the problem of OOV is highly significant, TF-IDF, FastText, and Word2Vec have been rigorously tested not only separately but also combined in different ways. Compared to the baseline configuration using standalone TF-IDF (which achieved 84.56% accuracy), the results show that the use of the hybrid TF-IDF + FastText provided the best performance with average accuracy of 86.49%, average precision of 86.32%, recall of 86.80% and F1 score of 86.55%. Conversely, the combination of multiple dense semantic vectors (Word2Vec and FastText) led to semantic drift and feature ambiguity; this created feature overlap and computational noise that obscured classification decision boundaries, resulting in redundancy and poor performance. It shows that the combination of lexico-statistical significance and semantic subword context greatly contributes to achieving better results in the Indonesian language setting and is very resistant to slang and OOV words found in digital settings. This study contributes to the field of natural language processing by providing a lightweight, highly accurate, and computationally efficient lexico-semantic framework tailored for moderating highly unstructured Indonesian social media text.

Keywords: Hate Speech; CNN; SimpleRNN; FastText; TF-IDF; Hybrid Model; Indonesian Language

1. INTRODUCTION

The proliferation of offensive content on microblogging sites like platform X has driven researchers to explore state-of-the-art machine learning and deep learning methodologies to automatically mitigate digital toxicity [1]. Recent systematic reviews in Natural Language Processing emphasize the urgent need for robust automatic hate speech detection systems capable of handling the highly unstructured linguistic complexities of social media platforms [2]. Thus, there has been an immediate need for designing automated and extremely accurate tools for moderating and identifying hate speech. Previous studies on the classification of hate speech and cyberbullying on social media have demonstrated significant performance improvements when employing advanced Deep Learning-based models [3], [4].

The automated detection of hate speech, hence, has become the topic of a number of studies across the realms of Natural Language Processing and Artificial Intelligence. The early approaches were primarily distinguished by their reliance on manually created features and conventional machine learning models, both of which struggled to handle the complexity of human language semantics. Text classification has been improved by deep neural network models. Previous research has used techniques based on Convolutional Neural Networks (CNN), which were first used in computer vision, to identify hate speech on Twitter. CNNs' strength is their capacity to extract spatial features; using one-dimensional convolution operations, they are particularly good at spotting local patterns and particular word combinations (n-grams) that strongly suggest abusive language. As a result, when compared to traditional methods, CNNs yield very competitive performance outcomes and quick calculation speed [5]. Recurrent Neural Networks (RNNs), on the other hand, are specifically made to handle sequence data. RNNs have the advantage of being able to store knowledge about the past in a hidden state, which enables them to comprehend context in ways that go beyond simple word-by-word analysis. As a result, RNN might be better at deciphering the meaning of a complex statement, but CNN might be able to identify an insult in a single word. Many attempts were made to combine RNN and CNN models because of their complementing features. Researchers demonstrated the capacity of such combinations to comprehend subtleties of social situations by combining the benefits of CNN models for local feature detection with those of RNNs[6]. Applying these models to Indonesian material on social media platform X poses a special and difficult problem despite these architectural breakthroughs. The biggest barrier is the language used by Indonesian internet users, which is very casual, unstructured, and dynamic. Users commonly employ a vast array of slang (*bahasa gaul*), regional dialects, acronyms, code-mixing, and intentionally altered spellings often used either to express raw emotion or to deliberately evade automated censorship filters. When an NLP model attempts to extract text features from such noisy data, it inevitably encounters a high rate of Out-of-Vocabulary (OOV) terms, which are words that the model has never seen during its training phase and thus cannot understand. To mitigate this pervasive problem, the selection of robust word embedding techniques is absolutely essential. Word embeddings translate textual data into numerical vectors that machines can process. Term Frequency-Inverse Document Frequency (TF-IDF) has been

Building of Informatics, Technology and Science (BITS)
Volume 8, No 1, June 2026 Page: 550−559
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v8i1.10046

utilized in numerous earlier studies to statistically highlight critical keywords; it assigns high weights to words that are frequent in a specific document but rare across the entire corpus, making it excellent for identifying highly discriminative swear words [7]. Semantic embeddings, on the other hand, capture a word's contextual meaning. Word2Vec groups words with similar meanings closer together by creating a dense semantic vector space based on a word's surrounding context [8]. However, OOV words are a major challenge for conventional Word2Vec. Other methods use FastText, a sophisticated embedding technique that separates words into character n-grams (sub-words) before vectorizing them, to overcome this. FastText is extremely resistant to typographical errors because of this sub-word information, which enables it to computationally infer the meaning of new, misspelled, or slang vocabulary based on their structural similarities to established terms [9]. The utilization of word embedding methods such as FastText has also proven highly effective in handling non-standard text and Out-of-Vocabulary (OOV) issues within Indonesian Twitter data [10].

Recent advancements in Natural Language Processing have been heavily dominated by Transformer-based architectures, such as BERT and its Indonesian-specific variant, IndoBERT. While these State-of-the-Art (SOTA) models exhibit exceptional accuracy in text classification tasks by capturing deep bidirectional context, their implementation poses significant challenges. Transformer models are notoriously resource-intensive, requiring massive computational power, high memory (VRAM) capacity, and extensive training times, making them less accessible for standard hardware deployments or real-time moderation systems with limited resources. Consequently, there is a compelling need to develop lightweight yet highly robust alternatives. While previous studies have explored various text representation techniques, the synergy of combining statistical methods (TF-IDF) with sub-word semantic embeddings (FastText) within a lightweight hybrid CNN-SimpleRNN architecture remains an underexploited area for achieving the optimal balance between computational efficiency and classification accuracy in the Indonesian language context. Therefore, this study aims to bridge this research gap. The primary purpose of this study is to systematically compare and assess the classification of hate speech on social media platform X using TF-IDF, Word2Vec, and FastText, evaluating their performance both as standalone single embeddings and in various hybrid combinations. By rigorously testing these configurations, this research seeks to identify a synergistic model that not only avoids the heavy computational burden of Transformer models but also provides an efficient, robust solution for comprehending the highly unstructured, OOV-heavy text typical of Indonesian social media. Furthermore, hybrid architectures that combine the local feature extraction capabilities of CNNs with the sequential context capturing of RNNs or LSTMs have been widely explored due to their superior performance compared to standalone models [11],[12]. The main contribution of this research lies in establishing an optimal, lightweight hybrid lexico-semantic text representation framework that surpasses standard dense embeddings in handling the unique Out-of-Vocabulary (OOV) challenges and morphological noise inherent in Indonesian social media, providing a practical blueprint for real-time digital moderation.

2. RESEARCH METHODOLOGY

2.1 Research Phases

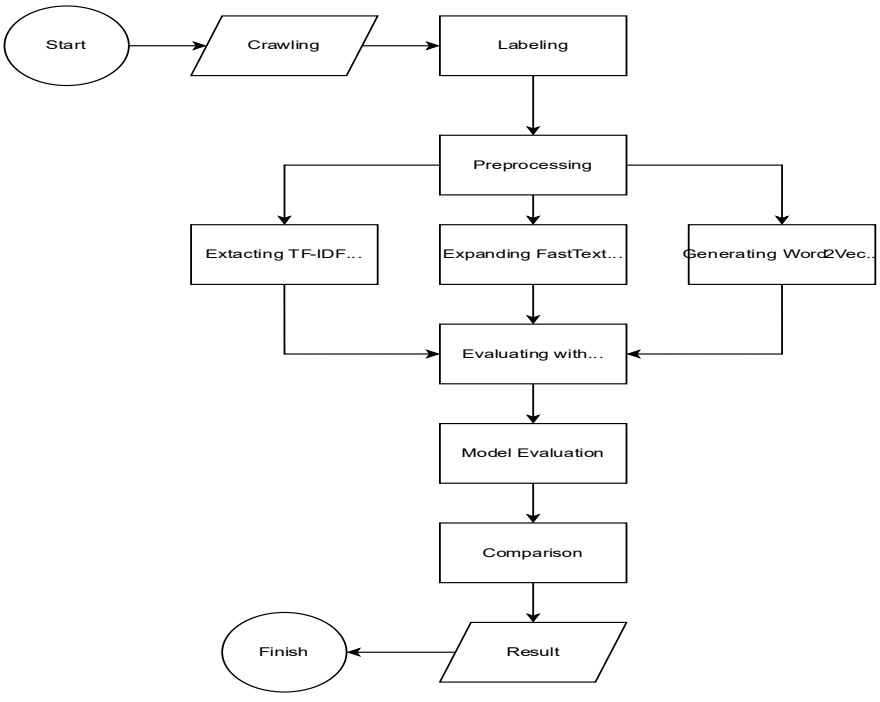

Figure 1. Flowchart of hate speech classification based on Hybrid CRNN and Word Embedding

Building of Informatics, Technology and Science (BITS)
Volume 8, No 1, June 2026 Page: 550−559
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v8i1.10046

This study employs a systematic methodology to categorize hate speech on the social media platform X. The research workflow begins with data collection via crawling, followed by manual data labeling and a rigorous preprocessing stage to clean the text. Once the data is prepared and partitioned, feature extraction is conducted using Term Frequency-Inverse Document Frequency (TF-IDF), FastText, and Word2Vec. These extracted features subsequently serve as the input for developing a hybrid Convolutional Recurrent Neural Network (C-RNN) model. Finally, the model's performance is independently evaluated across different feature extraction scenarios using standard metrics namely accuracy, precision, recall, and F1-score. This comprehensive evaluation allows for a rigorous comparison of the results to identify and report the optimal approach for hate speech classification.

Figure 1 **i**llustrates the systematic research workflow designed to address the challenges of hate speech classification on social media platform X. The process begins with data crawling and manual labeling, followed by a rigorous preprocessing stage to handle the unstructured nature of Indonesian digital discourse. A key feature of this methodology is the parallelized feature extraction phase, where the dataset is processed using three distinct methods: TF-IDF, FastText, and Word2Vec. These extracted features serve as the foundational inputs for the hybrid CNN-SimpleRNN model, which is configured to leverage both local spatial pattern recognition and sequential contextual dependency. Finally, the model undergoes a comparative evaluation phase to rigorously assess the performance of each embedding scenario, ensuring that the classification results are both accurate and reliable.

2.2 Crawling Dataset

Data was collected via web scraping methods on social media platform X. The procedure of gathering data took place in October and November of 2025. The dataset focused on both ordinary conversations and tweets in Indonesian that could be hate speech. A total of 30,000 tweets were collected. This collection ensured that the model would be trained using a variety of user experiences and perspectives that appropriately matched typical language use in the real world. The particular keywords used to gather the pertinent data throughout the crawling operation are shown in Table 1. These specific keywords were selected because they represent the most common highly offensive slurs and profanities frequently used by Indonesian netizens to demean or insult others, making them strong baseline indicators for scraping potential hate speech data.

Table 1. Sample of Dataset

Keyword	Sentence
Tolol	Ini lagi TOLOL nya bukan maen yg GOBLOK orang nya yg disalahin agama . Lagian masyarakat pada begitu juga karna sistem pendidikan Konoha masih jauh dari kata bagus dan layak alias masih perlu pembenahan. dan juga penyebabnya TIDAK BECUS NYA regime memperbaiki pendidikan Konoha wkwk opini org tolol yg suka liat org sebarin jejak digital org lain ke publik makanya gausah naro ekpetasi ke org yg mendadak tenar dgn pemikiran dia ga pernah melakukan kesalahan di masa lalu
Goblok	Ini lagi TOLOL nya bukan maen yg GOBLOK orang nya yg disalahin agama . Lagian masyarakat pada begitu juga karna sistem pendidikan Konoha masih jauh dari kata bagus dan layak alias masih perlu pembenahan. dan juga penyebabnya TIDAK BECUS NYA regime memperbaiki pendidikan Konoha wkwk opini org tolol yg suka liat org sebarin jejak digital org lain ke publik makanya gausah naro ekpetasi ke org yg mendadak tenar dgn pemikiran dia ga pernah melakukan kesalahan di masa lalu
Bangsat	BANGSAT LU THANAWIN SIALAN BABI LU KIRA GUE KUAT LU KIRA GUE SANGGUP PAKE LAGI NGGAK TUH BAJU
Anjing	DEMI ALLAH GUA GEDEGGG BANGET ANJINGGG. otaknya pada digade apa yak tolol smoga berdampak juga deh ke keluarga ppanya biar mreka ngerasain tuh ajg. gua nangisin ini dri malem kmren krna gua tau rasanya dan gua prnh ada diposisi kaya gini tp ga smpe nikah posisi gua masi kecil.

2.3 Data Labelling

Data labeling is one of the most crucial stages in creating a machine learning model. This study utilized a balanced dataset of 30,000 tweets, consisting of 15,000 positive hate speech instances (labeled as 1) and 15,000 negative non-hate speech instances (labeled as 0). This balanced distribution ensures that the model does not develop a bias toward a majority class during the training phase. Due to resource constraints, the data annotation was conducted by a single annotator. To mitigate subjective bias and ensure the reliability of the classification, the labeling process rigorously adhered to a strict set of criteria and definitions for hate speech established by the validated framework proposed by [13]. The labeling was performed with careful manual inspection, cross-referencing against common abusive indicators and context-specific slang, to ensure adherence to the defined binary classification. This method maintains the integrity of the ground truth data used for training. Table 2 provides examples of the grouped reviews

Table 2. Data Labelling

Label	Rating	Sentence	Description
1	Positive (Hate Speech)	Ini lagi TOLOL nya bukan maen yg GOBLOK orang nya yg disalahin agama Lagian masyarakat pada begitu juga karna sistem pendidikan Konoha	The text contains explicit offensive slurs ("TOLOL", "GOBLOK") and derogatory remarks aimed at

Building of Informatics, Technology and Science (BITS)
Volume 8, No 1, June 2026 Page: 550−559
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v8i1.10046

Label	Rating	Sentence	Description
		masih jauh dari kata bagus dan layak alias masih perlu pembenahan. dan juga penyebabnya TIDAK BECUS NYA regime memperbaiki pendidikan Konoha	degrading others, fulfilling the criteria for hate speech.
0	Negative (Non-Hate Speech)	wkwk opini org tolol yg suka liat org sebarin jejak digital org lain ke publik makanya gausah naro ekpetasi ke org yg mendadak tenar dgn pemikiran dia ga pernah melakukan kesalahan di masa lalu	The sentence attacks individuals using abusive terms ("tolol") to insult their opinions, clearly expressing hostility.

2.4 Preprocessing

At the initial stage of the methodology, the crawled dataset from social media platform X is inherently unstructured, raw, and highly noisy due to the informal nature of user interactions. To construct a reliable classification model, a rigorous preprocessing pipeline is strictly necessary to standardize the text and eliminate irrelevant computational noise before feature extraction. The preprocessing phase begins with extensive data cleaning, which specifically targets and removes non-textual clutter such as numbers, punctuation marks, special symbols, URLs, and emoticons that do not contribute to the semantic meaning of the tweet. The examples of the data components removed during this cleansing process are illustrated in Table 3. Once the text is free from external noise, the pipeline proceeds to case folding, a transformation process that converts all uppercase letters to lowercase, ensuring structural uniformity and preventing the model from treating identical words with different capitalizations as distinct entities. Following this, the tokenization process systematically divides the continuous string of text into individual, isolated words or tokens. This tokenized format is a fundamental prerequisite for stopword removal, where the Natural Language Toolkit (NLTK) library is utilized to filter out common but insignificant conjunctions and prepositions such as "saya", "sangat", and "di" that offer no discriminative value for hate speech detection, as shown in Table 4. Finally, to address the morphological complexity of the Indonesian language, a stemming process is implemented utilizing the Sastrawi library. This step algorithmically strips prefixes, suffixes, and infixes to reduce complex words to their fundamental root forms, as exemplified in Table 5. Through this comprehensive and consolidated procedure, the initially chaotic digital discourse is successfully transformed into highly structured, machine-readable data, which is optimally prepared for the subsequent feature extraction and neural network modeling phases.

Table 3. Sample of Cleaning

Data Component	Example
Numbers	100, Rp50.000
Symbols & Punctuation	!!!, ?, ,, .
URLs	www.tokobagus.com, http://...
Mentions	@promo, @username

Table 4. Sample of Stopword Removal

"saya", "sangat", "di", "ini", "karena", "itu", "banget", "kok"

Table 5. Sample of Stemming

Compound words	Base word
tololnya	tolol
orangnya	orang
disalahin	salah
penyebabnya	sebab
becusnya	Becus
pendidikan	Didik
memperbaiki	Baik
lagian	lagi
pembenahan	benah

2.5 Extracting TF-IDF Features

Feature extraction comes after the preprocessing phase. TF-IDF, a technique for assigning weights to each word (term) in a document, is the feature extraction approach employed in this work. The TF-IDF algorithm is then used to weight the data that has completed the previous step. This procedure yields a numerical representation of the text that may be fed into the subsequent modeling phase. During the implementation in this study, the TF-IDF vectorizer was configured with a maximum feature limit (*max_features*) of 5000 to manage the vocabulary size efficiently. This setting resulted in a sparse matrix with a dimensionality of 5000 representing the statistical weight of each discriminative term across the dataset. The Term Frequency-Inverse Document Frequency (TF-IDF) method is a highly reliable statistical lexical weighting technique for highlighting discriminative n-gram features, and it remains

Building of Informatics, Technology and Science (BITS)
Volume 8, No 1, June 2026 Page: 550−559
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v8i1.10046

highly relevant for short text classification on social media platforms [7]. Combining term-weighting schemes like TF-IDF with word embeddings has been shown to significantly enhance the semantic representation of text by prioritizing discriminative terms, which guides the neural network in focusing on critical abusive vocabulary [14].

2.6 Expanding FastText Features

The technique of enhancing text by adding semantic information to make the document representation more comprehensive is known as feature expansion. The goal is to help the model find terms that are not evident in the data, especially in non-standard text. In this study, features were expanded using the FastText method. This method can understand linguistic variances such as acronyms, variant spellings, and non-standard words by breaking words down into character n-grams. Because each word is represented as a collection of consecutive character fragments, the model is able to detect word similarities. By using this technique, FastText may find words with similar forms or meanings and then supplement the word vector representation with additional relevant information. This process helps overcome vocabulary limitations and improves the quality of feature representation in text. The FastText model was trained in the experimental scenario using a context window size of five and a vector dimensionality of 300. Additionally, in order to handle out-of-vocabulary (OOV) phrases in the dataset, the model was set up to evaluate character n-grams ranging from 3 to 6, allowing for very robust sub-word feature extraction.

2.7 Generating Word2Vec Features

In order to identify missing or previously unidentified words, this study used Word2Vec feature extension. In a word list or corpus created using the Word2Vec model, words with comparable or semantically related meanings are used to represent a word with zero weight in feature extraction [15]. This process is known as feature expansion. As discussed in Section 2.8, there are two kinds of Word2Vec algorithms: Skip-Gram and Continuous Bag-of-Words (CBOW). For this implementation, the Continuous Bag-of-Words (CBOW) architecture was selected. The Word2Vec model parameters were set to generate dense semantic vectors with a dimension of 300 and a context window size of 5. This configuration allows the model to accurately capture deep semantic relationships between words within the defined context boundary. Deep learning approaches for hate speech detection rely heavily on the quality of semantic representation, where pre-trained dense embeddings like Word2Vec and FastText are essential for managing the informal and highly distorted nature of social media texts [16].

2.8 CRNN

This study uses the TensorFlow framework to create a single hybrid model for text categorization by combining two models, CNN and RNN. Both local characteristics and sequential patterns in text are intended to be captured by this model.

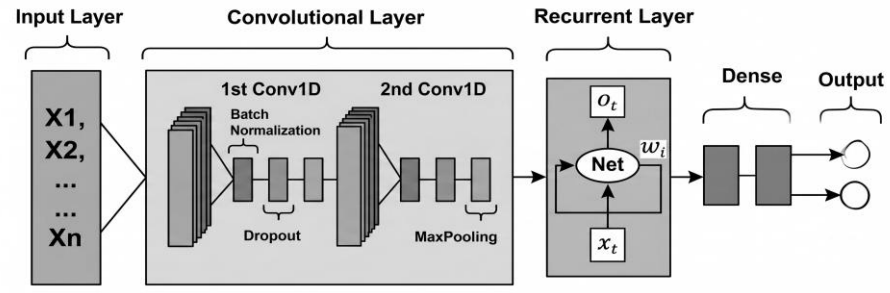

Figure 2. Hybrid CRNN Archiecture

The classification model is built utilizing a hybrid Convolutional Recurrent Neural Network (C-RNN) architecture, as depicted in Figure 2, which merges CNN's ability to capture local spatial features with RNN's capability to store long-term sequence memory. The feature matrix is created, normalized, and then reshaped into a three-dimensional structure to meet the model input specifications. The architecture starts with a one-dimensional convolutional layer (Conv1D) that uses the ReLU activation function and applies 128 filters with a kernel size of 3 (or 1). This layer is intended to record brief word combinations and local textual patterns that frequently reflect derogatory phrases. The Conv1D output is then processed through Batch Normalization and MaxPooling1D layers to increase training stability and decrease dimensional complexity. n text classification tasks, deep learning architectures particularly Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) have demonstrated significant advantages over traditional shallow models by effectively capturing both local spatial features and long-term dependencies [17].

The recovered spatial representations are then fed into a 64-unit Bidirectional SimpleRNN layer. While SimpleRNN theoretically suffers from the vanishing gradient problem, its application is justified here as social media text is inherently short-form (microblogging). Within these sequences, semantic linkages are effectively captured with lower computational overhead compared to LSTM/GRU. its application is justified in this study because social media text on platform X is inherently short-form (microblogging) and heavily constrained by character limits. Within these

Building of Informatics, Technology and Science (BITS)
Volume 8, No 1, June 2026 Page: 550−559
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v8i1.10046

short sequences, deeper semantic linkages inside a sentence can still be effectively captured by the model thanks to the bidirectional mechanism, which enables the network to examine the sequence from both immediate past and future contexts with significantly lower computational overhead.

2.9 Model Performance Measure

In machine learning, performance measurement is a notion that offers details about the classification system's predictions and actual classification results [18]. By taking into account the identity of subsequent classification outcomes, performance assessment can be used to assess a classification model's total performance [19]. True Positive (TP) refers to the number of positive data items that are classified as correct; False Positive (FP) refers to the number of positive data items that are classified as incorrect; True Negative (TN) refers to the number of negative data items that are classified as correct; and False Negative (FN) refers to the number of negative data items that are classified as incorrect [15],[18].

Performance measurement yields a number of metrics, including Accuracy, Precision, Recall, and F1-score, that can be used to evaluate a classification model's performance. The ratio of accurate predictions (True Positives and True Negatives) to the total amount of data is known as accuracy. The following formula is used to determine accuracy:

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)} \quad (1)$$

The ratio of genuine positives to all positive predictions, including both true and false positives, is known as precision. The following is an expression for the precision formula:

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

Precision is a ratio of Recall, which is the ratio of True Positives to all positive data, including False Negatives and True Positives. The ratio of True Positives to the total number of anticipated positives, including both True Positives and False Positives, is the formula for recall. The following is an expression for the Precision formula:

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision\ \_+ Recall} \quad (3)$$

3. RESULT AND DISCUSSION

3.1 Training Parameters and Testing *Scenario*

Prior to evaluating feature extraction performance, all models were tested using a uniform computing environment to ensure comparative validity. The Hybrid Convolutional Recurrent Neural Network (C-RNN) architecture was trained using the Adam optimizer with an initial learning rate of 0.0001. Training was conducted for a maximum of 30 epochs with a batch size of 64. To prevent overfitting and optimize computational time, this study implemented callback functions such as EarlyStopping (with a patience of 5 epochs to return the best weights) and ReduceLROnPlateau (a factor of 0.5 and a patience of 3), which dynamically reduces the learning rate when the validation loss metric no longer shows improvement.

The testing focused on comparing six different feature extraction and expansion scenarios, ranging from a single method (lexical or semantic) to a hybrid method. Each scenario was executed three times (3 runs) to minimize bias from random weight initialization, and the results reported are the average values across all trials.

Table 6. Testing Scenario

Testing Scenario	Description
Scenario 1	The C-RNN model uses TF-IDF statistical features as a baseline.
Scenario 2	The C-RNN model utilizes word embeddings generated by the Word2Vec model.
Scenario 3	The C-RNN model employs FastText embeddings for feature representation.
Scenario 4	The hybrid method combines Word2Vec and FastText models.
Scenario 5	The hybrid method integrates TF-IDF statistical features with Word2Vec embeddings.
Scenario 6	The hybrid method integrates TF-IDF statistical features with FastText embeddings.

3.2 Scenario 1 Testing Result

In scenario 1, the model's performance was evaluated solely through statistical feature extraction (TF-IDF) without the use of interword meaning understanding. Table 8 displays the overall outcomes of all testing scenarios.

Table 7. Overall Performance of Feature Extraction Methods

Method	Accuracy	Precision	Recall	F1-score
CRNN + TF-IDF	0.8456	0.8447	0.8414	0.8431
CRNN + Word2Vec	0.8103	0.8135	0.7983	0.8058

Building of Informatics, Technology and Science (BITS)
Volume 8, No 1, June 2026 Page: 550−559
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v8i1.10046

Method	Accuracy	Precision	Recall	F1-score
CRNN +_FastText	0.8234	0.8060	0.8497	0.8257
CRNN + Word2Vec + FastText	0.8080	0.8300	0.7678	0.7976
CRNN + TF-IDF + Word2Vec	0.8523	0.8526	0.8467	0.8497
CRNN + TF-IDF + FastText	0.8649	0.8632	0.8680	0.8655

The application of the TF-IDF feature produced a respectable accuracy of 84.56% based on the outcomes of scenario 1. This high accuracy result shows that a key marker in identifying hate speech classification on social media X is the frequency of occurrence of swear words or harsh phrases, which are typically assigned a high central weight by TF-IDF since they seldom appear in neutral corpuses.

3.3 Scenario 2 Testing Result

The Word2Vec architecture was used in Scenario 2 to assess the model's performance when focusing solely on semantic vector space. In comparison to the baseline scenario, the results presented in **Table 7** demonstrate a drop in accuracy to 81.03%. The tendency of pure semantic representation to generalize every word in a dense vector space distracts the CNN-SimpleRNN architecture's attention from highly discriminative, yet sparse, abusive terms. The tendency of pure semantic representation to generalize every word in a dense vector is the cause of this. Because of this, the cluster of neutral words in a tweet frequently distracts or drowns out the C-RNN architecture's attention to objectionable words.

3.4 Scenario 3 Testing Result

A FastText implementation that can split words into character n-grams (subwords) is tested in Scenario 3. Two corpora, IndoNews and Tweets, are combined in this test. According to the results presented in **Table 7**, the FastText implementation outperformed Word2Vec with an accuracy of 82.34%. This is because FastText effectively addresses vocabulary limitations through character n-gram sub-word modeling, which is crucial for handling the frequent typos, abbreviations, and slang spellings prevalent on social media platform X. This is due to the fact that FastText is made to deal with vocabulary issues that are beyond of its scope. FastText can successfully recreate the sub-word meanings of slang curse words, which Word2Vec cannot, because tweets on social media X are rife with typos, abbreviations, and slang spellings.

3.5 Scenario 4 Testing Result

In Scenario 4, Word2Vec and FastText vectors are combined to evaluate a purely semantic hybrid strategy. As shown in Table 7, this configuration yielded the lowest accuracy of 80.80% and a recall of 76.78%. Unlike the statistical features in Scenario 1, the concatenation of two dense semantic models (totaling 600 dimensions) does not provide complementary information. Instead, it introduces semantic drift and feature ambiguity; because both models are trained on similar language structures, their concatenation creates high feature correlation rather than true dimensional complexity. Without the lexico-statistical guidance provided by TF-IDF, the hybrid CNN-SimpleRNN model struggles to discern discriminative patterns amidst this redundant semantic space.

3.6 Scenario 5 Testing Result

Scenario 5 evaluates a hybrid lexico-semantic strategy that combines semantic vectors from Word2Vec with statistical indicators from TF-IDF. As indicated in **Table 7**, accuracy increased to 85.23% in Scenario 5. This hybrid lexico-semantic approach proved effective as Word2Vec provided structural sentence context, while the TF-IDF matrix acted as a statistical "guide," highlighting the presence of highly discriminative hate speech keywords. in hate speech, while the TF-IDF matrix acted as a "guide" emphasizing their presence.

3.7 Scenario 6 Testing Result

The best-fit test setup is Scenario 6, which combines FastText's semantic subword understanding with TF-IDF lexical characteristics. With an accuracy of 86.49% and an F1-score of 86.55% **Table 7**, Scenario 6 achieved the best classification performance. This configuration leverages the synergy between FastText's sub-word morphological adaptation and TF-IDF's statistical weighting. The results demonstrate that combining these techniques provides the most robust solution for handling linguistic anomalies on platform X, confirming the effectiveness of the hybrid CNN-SimpleRNN model compared to single-component architectures. The synergy of complementing techniques is the foundation of this accomplishment. While the character n-gram design of FastText corrects the Out-of-Vocabulary gap created by profanity abbreviated or changed by social media users, the TF-IDF matrix separates and assigns significant weight to hate speech words. These findings demonstrate that, when it comes to handling linguistic anomalies on platform X, the Scenario 6 method is the most effective and reliable feature extraction configuration.

3.8 Discussion

The experimental findings reveal profound insights into how different feature extraction and word embedding techniques fundamentally alter the learning dynamics of the hybrid CNN-SimpleRNN architecture when processing

Building of Informatics, Technology and Science (BITS)
Volume 8, No 1, June 2026 Page: 550−559
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v8i1.10046

unstructured Indonesian social media text. To visualize these variations in classification efficacy across lexical, semantic, and hybrid feature representations, the experimental results are summarized in Figure 3.

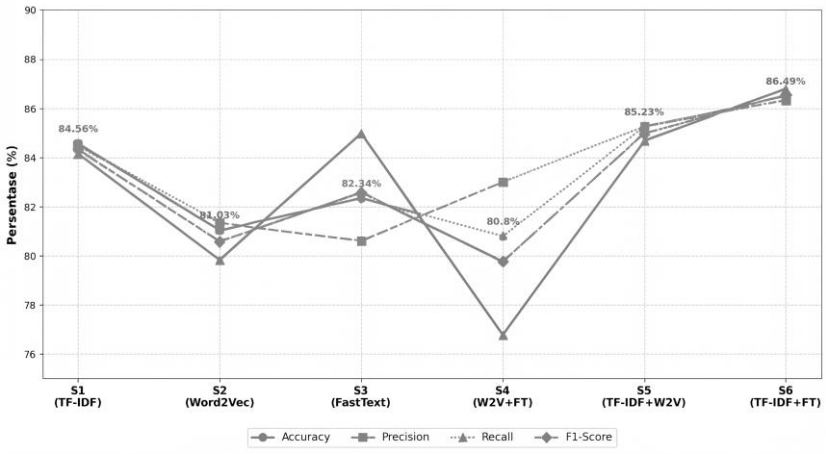

Figure 3. Performance Comparison Across Various Scenarios

As observed in Figure 3, while the model demonstrated robust learning capabilities across all configurations, a detailed comparative analysis highlights significant variations in performance. Scenario 1, which utilized TF-IDF as a standalone lexical baseline, achieved an impressive accuracy of 84.56%. This result highlights a crucial aspect of hate speech classification: the presence of highly offensive terms, such as profanity, serves as a potent marker for identifying harmful content. Because it gives more weight to uncommon but significant phrases, TF-IDF is quite effective in this situation, enabling the CNN-SimpleRNN model to highlight words with strong negative intent.

On the other hand, when used separately, standalone semantic-based methods demonstrated a number of drawbacks. Word2Vec's accuracy in Scenario 2 was 81.03%, while FastText's accuracy in Scenario 3 was 82.34% (Figure 3). The nature of dense semantic representations, which map words into vector spaces based on contextual associations without taking explicit statistical value into account, can be blamed for this poorer performance. Because they are combined with many neutral and frequently used terms in tweets, crucial hate-related signs may become less obvious. However, FastText offers structural advantages; its character n-gram system effectively handles spelling variants, informal Indonesian slang, Out-of-Vocabulary (OOV) phrases, and purposefully changed words, allowing it to infer the meaning of distorted abusive statements by analyzing character-level fragments.

The model performed the worst overall in Scenario 4, which included Word2Vec and FastText without using TF-IDF (80.80% accuracy, as shown in Figure 3). When these two kinds of dense semantic vectors are combined, the model suffers from feature ambiguity and semantic redundancy. The high feature correlation creates confusion in the CNN-SimpleRNN network, resulting in increased classification noise and poorly defined decision boundaries.

Scenario 6, which combined TF-IDF with FastText trained on the IndoNews and Tweets corpora, showed the biggest improvement. With an accuracy of 86.49% and an F1-Score of 86.55% (Figure 3), this setup produced the best results. The neural network was better able to concentrate on hate-related indications since TF-IDF served as a weighting mechanism that highlighted significant offending keywords. Simultaneously, FastText provided morphological adaptation and semantic understanding, enabling the model to decipher slang, misspelled words, abbreviations, and modified abusive statements. The CNN layer was better able to recognize local offensive n-gram patterns with this enriched representation, while the Bidirectional SimpleRNN layer was able to capture more general contextual dependencies. These results confirm that combining statistical lexical characteristics from TF-IDF with sub-word semantic representations from FastText is the most efficient and dependable method for classifying hate speech in highly unstructured social media environments.

To further validate the architectural contribution of this study, an ablation study was conducted to compare the proposed hybrid CNN-SimpleRNN model against standalone CNN and SimpleRNN architectures. This comparison utilized the best-performing feature configuration (Scenario 6: TF-IDF + FastText).

As presented in Table 8, the hybrid structure significantly outperforms single-architecture models. The standalone CNN achieved an accuracy of 84.27%, while the standalone SimpleRNN only reached 80.140%. This empirical evidence directly addresses the architectural dynamics: the CNN layer effectively localizes abusive local patterns (n-grams), while the Bidirectional SimpleRNN captures the broader sequential context of the tweet. This crucial synergy is lost when employing a standalone architecture, thus confirming the superiority of the hybrid C-RNN approach for hate speech classification.

Table 8. Performance Comparison of Standalone

Method	Accuracy	Precision	Recall	F1-score
CNN	0.8427	0.8485	0.8343	0.8413
RNN	0.8040	0.8098	0.7947	0.8022

Building of Informatics, Technology and Science (BITS)
Volume 8, No 1, June 2026 Page: 550−559
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v8i1.10046

Furthermore, when comparing these findings with previous studies, the proposed hybrid TF-IDF and FastText model demonstrates significant advantages. For instance, the baseline CNN model proposed by Pangestuti and Agustian [5] achieved competitive results but struggled with high rates of slang without semantic expansion. Similarly, while the hybrid CNN-BiLSTM approach by [12] utilized FastText, the computational overhead was considerably higher due to the complex LSTM layer. The findings of this study indicate that replacing LSTM with SimpleRNN and injecting TF-IDF statistical weights not only yields an optimal accuracy of 86.49% but also reduces structural complexity. This proves that the proposed model is a highly efficient alternative for Indonesian microblogging texts compared to purely semantic models explored in prior research [20], [10].

4. CONCLUSION

This study demonstrates that the effectiveness of hate speech classification on social media platform X is strongly influenced by the feature representation used to model textual data. The findings indicate that combining lexical-statistical and semantic representations can significantly improve classification performance, particularly for noisy and highly unstructured Indonesian social media text. The integration of TF-IDF and FastText proved effective in capturing both important hate-related keywords and contextual semantic relationships, enabling the Hybrid C-RNN model to achieve strong classification performance. These results suggest that hybrid feature representation plays a more critical role than architectural complexity alone in improving hate speech detection. However, the findings should be interpreted with caution due to several methodological limitations. The relatively small and imbalanced dataset raises concerns regarding model generalization and robustness, especially when applied to broader real-world scenarios across diverse digital platforms. This limitation significantly affects the reliability of the reported performance and indicates that stronger dataset preparation and validation strategies should have been implemented earlier in the research process. The main contribution of this study lies in proving that a hybrid lexico-semantic approach (TF-IDF combined with FastText) paired with a C-RNN architecture provides a highly efficient and robust solution for handling Indonesian OOV terms. This serves as a lightweight and reliable alternative to resource-intensive models for real-time hate speech moderation. Furthermore, although the Hybrid C-RNN architecture produced competitive results, the rapid advancement of Natural Language Processing indicates that this approach is no longer state-of-the-art. Recent Transformer-based models such as IndoBERT have demonstrated superior capability in capturing contextual and semantic information, particularly for complex linguistic patterns in Indonesian text. This suggests that while the proposed method provides useful baseline insights, more modern architectures are likely to deliver better performance and stronger generalization. Therefore, future research should focus on utilizing larger and more balanced datasets, improving validation strategies, and adopting more advanced Transformer-based approaches to build more reliable hate speech detection systems for Indonesian digital platforms.

REFERENCES

[1] F. E. Ayo, O. Folorunso, F. T. Ibharalu, and I. A. Osinuga, “Machine Learning Techniques For Hate Speech Classification Of Twitter Data: State-Of-The-Art, Future Challenges And Research Directions,” *Comput. Sci. Rev.*, vol. 38, p. 100311, 2020, doi: 10.1016/j.cosrev.2020.100311.

[2] M. S. Jahan and M. Oussalah, “A Systematic Review Of Hate Speech Automatic Detection Using Natural Language Processing,” *Neurocomputing*, vol. 546, p. 126232, 2023, doi: 10.1016/j.neucom.2023.126232.

[3] Salsabila, R. Sarno, I. Ghozali, and K. R. Sungkono, “Improving Cyberbullying Detection Through Multi-Level Machine Learning,” *Int. J. Electr. Comput. Eng.*, vol. 14, no. 2, pp. 1779–1787, 2024, doi: 10.11591/ijece.v14i2.pp1779-1787.

[4] A. M. El Koshiry, E. H. I. Eliwa, T. A. El-Hafeez, and M. Khairy, “Detecting Cyberbullying Using Deep Learning Techniques: A Pre-Trained Glove And Focal Loss Technique,” *PeerJ Comput. Sci.*, vol. 10, pp. 1–33, 2024, doi: 10.7717/peerj-cs.1961.

[5] I. Pangestuti and S. Agustian, “Klasifikasi Komentar Abusive Dan Hate Speech Teks Twitter Menggunakan Metode Convolutional Neural Network,” *Pros. Semin. Nas. Teknoka*, vol. 7, no. 2502, pp. 23–30, 2022, [Online]. Available: https://journal.uhamka.ac.id/index.php/teknoka/article/view/11000

[6] F. LAGRARI and Y. ELKETTANI, “Customized BERT With Convolution Model: A New Heuristic Enabled Encoder For Twitter Sentiment Analysis,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 10, p. 2025, 2020, doi: 10.14569/IJACSA.2020.0111053.

[7] M. Isnan, G. N. Elwirehardja, and B. Pardamean, “Sentiment Analysis For TikTok Review Using VADER Sentiment And SVM Model,” *Procedia Comput. Sci.*, vol. 227, pp. 168–175, 2023, doi: 10.1016/j.procs.2023.10.514.

[8] K. W. Church, “Emerging Trends: Word2Vec,” *Nat. Lang. Eng.*, vol. 23, no. 1, pp. 155–162, 2017, doi: 10.1017/S1351324916000334.

[9] J. Xu and Q. Du, “A Deep Investigation Into FastText,” *Proc. - 21st IEEE Int. Conf. High Perform. Comput. Commun. 17th IEEE Int. Conf. Smart City 5th IEEE Int. Conf. Data Sci. Syst. HPCC/SmartCity/DSS 2019*, pp. 1714–1719, 2019, doi: 10.1109/HPCC/SmartCity/DSS.2019.00234.

[10] F. N. Puteri, Y. Sibaroni, and F. F., “Hate Speech Detection in Indonesia Twitter Comments Using Convolutional Neural Network (CNN) and FastText Word Embedding,” *J. Media Inform. Budidarma*, vol. 7, no. 3, p. 1154, 2023, doi: 10.30865/mib.v7i3.6401.

[11] A. J. Andika, Y. Kristian, and E. I. Setiawan, “Deteksi Komentar Cyberbullying Pada YouTube Dengan Metode Convolutional Neural Network – Long Short-Term Memory Network (CNN-LSTM),” *Teknika*, vol. 12, no. 3, pp. 183–188, 2023, doi: 10.34148/teknika.v12i3.677.

Building of Informatics, Technology and Science (BITS)
Volume 8, No 1, June 2026 Page: 550−559
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v8i1.10046

[12] M. A. S. Nasution and E. B. Setiawan, "Enhancing Cyberbullying Detection on Indonesian Twitter: Leveraging FastText for Feature Expansion and Hybrid Approach Applying CNN and BiLSTM," *Rev. d'Intelligence Artif.*, vol. 37, no. 4, pp. 929–936, 2023, doi: 10.18280/ria.370413.
[13] M. O. Ibrohim and I. Budi, "Multi-label Hate Speech and Abusive Language Detection in Indonesian Twitter," *Proc. Third Work. Abus. Lang. Online*, vol. 1, no. 1, pp. 46–57, 2019, doi: 10.18653/v1/w19-3506.
[14] A. Onan, "Sentiment Analysis On Product Reviews Based On Weighted Word Embeddings And Deep Neural Networks," *Concurr. Comput. Pract. Exp.*, vol. 33, no. 23, pp. 1–12, 2021, doi: 10.1002/cpe.5909.
[15] M. N. Hasan, K. S. Sakib, T. T. Preeti, J. Allohibi, A. A. Alharbi, and J. Uddin, "OLF-ML: An Offensive Language Framework for Detection, Categorization, and Offense Target Identification Using Text Processing and Machine Learning Algorithms," *Mathematics*, vol. 12, no. 13, 2024, doi: 10.3390/math12132123.
[16] M. Corazza, S. Menini, E. Cabrio, S. Tonelli, and S. Villata, "A Multilingual Evaluation for Online Hate Speech Detection," *ACM Trans. Internet Technol.*, vol. 20, no. 2, 2020, doi: 10.1145/3377323.
[17] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 33, no. 12, pp. 6999–7019, 2022, doi: 10.1109/TNNLS.2021.3084827.
[18] M. Kamyab, G. Liu, A. Rasool, and M. Adjeisah, "ACR-SA: Attention-Based Deep Model Through Two-Channel CNN And Bi-RNN For Sentiment Analysis," *PeerJ Comput. Sci.*, vol. 8, pp. 1–29, 2022, doi: 10.7717/peerj-cs.877.
[19] D. Chicco and G. Jurman, "The Advantages Of The Matthews Correlation Coefficient (MCC) Over F1 Score And Accuracy In Binary Classification Evaluation," *BMC Genomics*, vol. 21, no. 1, pp. 1–13, 2020, doi: 10.1186/s12864-020-6950-5.
[20] J. S. Malik, H. Qiao, G. Pang, and A. van den Hengel, "Deep Learning for Hate Speech Detection: A Comparative Study," Arxiv, pp. 1–18, 2022, [Online]. Available: http://arxiv.org/abs/2202.09517