SINTECHCOM: Science, Technology and Communication Journal
Vol. 6, No. 3, June 2026, pp. 271-282, DOI: 10.59190/stc.v6i3.384

Enhancing Indonesian hadith classification through multi-word embedding and support vector machine

Mila Hastati*, Junadhi, Susi Erlinda, Agustin
Department of Informatics Engineering, Universitas Sains dan Teknologi Indonesia, Pekanbaru 28299, Indonesia

ABSTRACT

Hadith classification plays an important role in supporting the organization and retrieval of Islamic knowledge in digital environments. However, the increasing volume of digital hadith collections presents challenges for manual classification, making automated approaches increasingly necessary. This study proposes a hadith text classification framework based on support vector machine (SVM) and a Multi-Word Embedding approach. The dataset used in this study was obtained from the Kaggle hadith dataset repository and consists of 34,441 hadith records. The textual data were preprocessed through case folding, noise removal, stopword removal, and stemming before feature extraction. Three embedding strategies were evaluated, namely Word2Vec, FastText, and the proposed multi-word embedding, which combines Word2Vec and FastText representations through vector concatenation. The generated feature vectors were subsequently classified using SVM and evaluated using accuracy, precision, recall, and F1-score. Experimental results show that the proposed multi-word embedding approach achieved the best performance, obtaining an accuracy of 75.58%, precision of 75.68%, recall of 75.58%, and F1-score of 75.46%. These results outperform Word2Vec + SVM and FastText + SVM, demonstrating that the integration of contextual semantic and subword-level information produces richer feature representations and improves classification effectiveness. The findings indicate that multi-word embedding is a promising approach for automated hadith text classification and can contribute to the development of intelligent Islamic information systems.

ARTICLE INFO

Article history:

Received Jun 12, 2026
Revised Jun 14, 2026
Accepted Jun 15, 2026

Keywords:

FastText
Hadith Classification
Multi-Word Embedding
Support Vector Machine
Word2Vec

* **Corresponding Author**
E-mail address: 2417052802124@usti.ac.id

1. INTRODUCTION

Hadith is one of the primary sources of Islamic teachings after the Qur'an, serving as an essential reference for religious practices, legal rulings, ethical values, and daily life guidance for Muslims. The rapid growth of digital Islamic repositories has resulted in the availability of large collections of hadith texts in electronic formats [1]. While these digital resources provide broader access to religious knowledge, they also create challenges in organizing, searching, and categorizing hadith documents efficiently. Manual classification of hadith collections requires extensive expertise in hadith sciences and considerable time, making automated classification approaches increasingly important for supporting Islamic digital libraries and educational applications [2, 3].

In hadith studies, authenticity classification plays a crucial role in determining the reliability of a narration. Traditionally, hadiths are categorized into several classes, including Shahih (authentic), Hasan (good), and Dhaif (weak), based on the evaluation of both the chain of narrators (sanad) and the textual content (matan) [4]. However, the increasing volume of digitized hadith collections has made manual classification impractical for large-scale datasets. Consequently, researchers have begun

exploring computational approaches capable of assisting the classification process through automated text analysis techniques [5].

Recent advances in Natural Language Processing (NLP) and Machine Learning have significantly improved the ability of computers to analyze and classify textual information. Various machine learning algorithms have been successfully applied to text classification tasks, including Naïve Bayes, Decision Tree, Random Forest, K-Nearest Neighbor, and Support Vector Machine (SVM) [6, 7]. Among these methods, SVM has demonstrated strong performance in high-dimensional text classification problems due to its capability to construct optimal decision boundaries and maintain good generalization performance even with limited training data. Previous studies have also reported competitive results when applying SVM to religious text classification tasks, including Qur'anic verse categorization and hadith classification [8, 9].

The effectiveness of machine learning models in text classification largely depends on how textual information is represented. Traditional approaches such as Bag-of-Words (BoW) and Term Frequency–Inverse Document Frequency (TF-IDF) primarily focus on word occurrence frequencies and often fail to capture semantic relationships between words [10, 11]. To address this limitation, word embedding techniques have been widely adopted. Word embedding transforms textual data into dense vector representations that preserve semantic and syntactic relationships among words. Among the most widely used embedding methods are Word2Vec and FastText. Word2Vec effectively captures contextual semantic information, whereas FastText incorporates subword information, enabling better handling of rare words and morphological variations [12].

Several studies have investigated the application of Word2Vec or FastText individually for text classification tasks. Although these approaches have achieved promising results, relying on a single embedding model may not fully capture the complexity of linguistic information contained in textual data [13]. Each embedding technique possesses unique strengths and limitations in representing semantic and morphological characteristics. Consequently, using only one embedding representation may result in the loss of complementary information that could improve classification performance [14].

A review of previous studies reveals that most hadith classification research still employs single feature representation techniques such as TF-IDF, Word2Vec, or FastText independently. Furthermore, research exploring the integration of multiple embedding representations for Indonesian hadith classification remains limited [15, 16]. This gap indicates the need for a more comprehensive feature representation strategy capable of combining the advantages of different embedding methods. By integrating multiple embeddings, richer textual representations can be generated, potentially improving the ability of machine learning models to distinguish between hadith categories.

Therefore, this study proposes a hadith text classification framework based on Multi-Word Embedding and Support Vector Machine. The proposed approach combines Word2Vec and FastText representations through a feature fusion strategy to enrich semantic and morphological information extracted from Indonesian hadith texts. The resulting feature vectors are subsequently utilized as inputs to an SVM classifier for categorizing hadiths into three authenticity classes: Shahih, Hasan, and Dhaif. The contributions of this study are threefold: (1) developing a multi-word embedding representation for Indonesian hadith texts, (2) evaluating the effectiveness of SVM for hadith authenticity classification, and (3) comparing the performance of multi-word embedding against single embedding approaches. The findings are expected to contribute to the advancement of NLP applications in Islamic studies and provide a practical solution for automated hadith classification systems.

2. RESEARCH METHODS

This study proposes a hadith text classification framework based on Support Vector Machine (SVM) and a Multi-Word Embedding approach. The objective is to investigate whether combining multiple embedding representations can improve the classification performance of Indonesian hadith texts compared to single-embedding approaches. The overall research workflow consists of six main stages: dataset preparation, text preprocessing, feature representation using Word2Vec, feature representation using FastText, Multi-Word Embedding generation, classification using SVM, and performance evaluation. The complete research framework is illustrated in Figure 1.

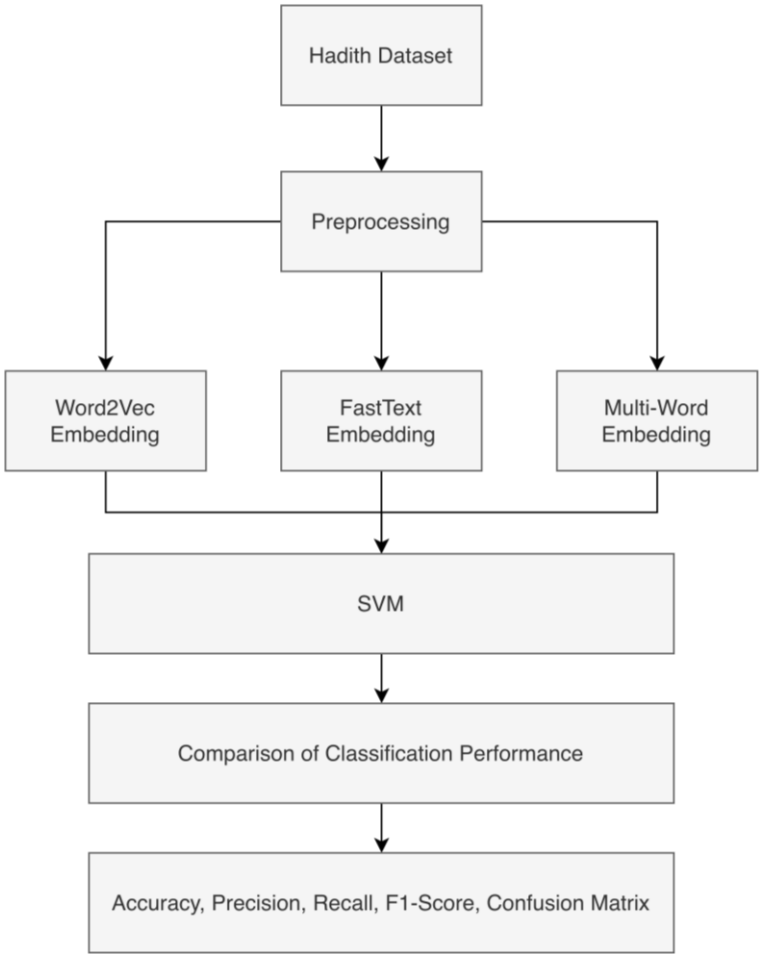

Figure 1. Proposed classification workflow.

2.1. Dataset

The dataset used in this study was obtained from the publicly available Hadith Dataset provided on the Kaggle platform. The dataset was developed by Fahd09 and can be accessed through the Kaggle repository. It contains a large collection of hadith texts along with associated metadata describing their source, chapter information, and narration structure. The availability of both textual content and metadata makes the dataset suitable for Natural Language Processing (NLP) and text classification tasks. The dataset consists of 34,441 hadith records and 9 attributes, namely id, hadith_id, source, chapter_no, hadith_no, chapter, chain_indx, text_ar, and text_en. The attributes provide information regarding the unique identity of each hadith, its source collection, chapter details, narrator chain identifiers, and textual content in both Arabic and English. Table 1 presents the description of each attribute contained in the dataset.

Table 1. Description of dataset attributes attribute.

Attribute	Description
id	Unique identifier of each record
hadith_id	Unique identifier of the hadith
source	Source collection of the hadith
chapter_no	Chapter number
hadith_no	Hadith number within the chapter
chapter	Chapter title
chain_indx	Narrator chain identifiers
text_ar	Original Arabic hadith text
text_en	English translation of the hadith text

In this study, the text_en attribute was employed as the primary textual feature for classification, while the remaining attributes were used solely for descriptive and identification purposes. The dataset contains 34,441 hadith records, providing a substantial corpus for training and evaluating the proposed classification model.

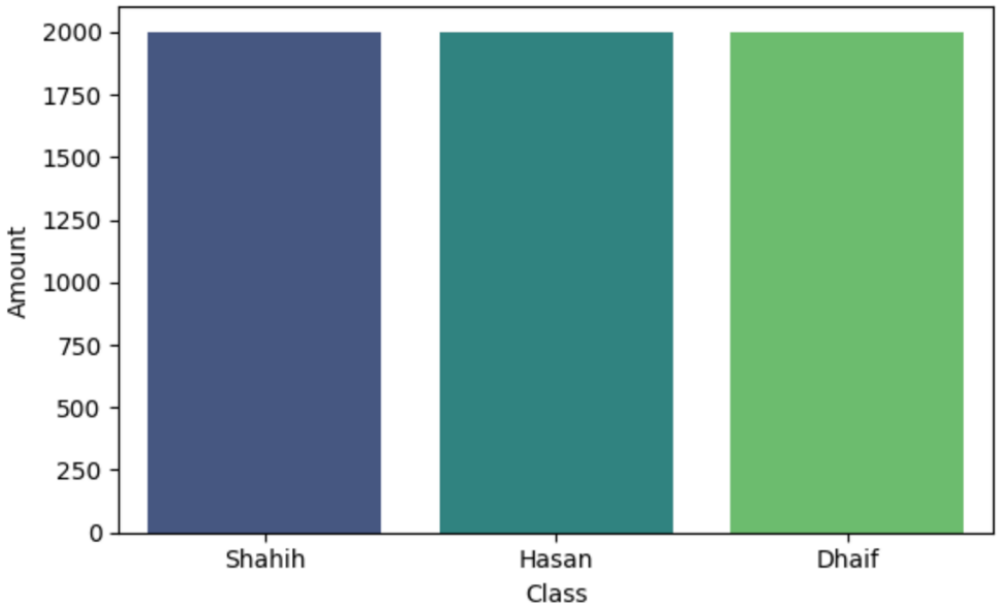

Figure 2. Distribution of hadith classes.

Figure 1 presents the distribution of hadith classes used in this study. The dataset consists of three categories, namely Shahih, Hasan, and Dhaif, with approximately 2,000 instances in each class. The relatively balanced class distribution minimizes potential bias during model training and enables a more reliable evaluation of classification performance.

2.2. Text Preprocessing

Text preprocessing was conducted to transform raw hadith texts into a clean and standardized format suitable for machine learning analysis. This stage aims to reduce noise, eliminate irrelevant information, and normalize textual representations before feature extraction. The preprocessing pipeline consisted of case folding, noise removal, whitespace normalization, stopword removal, and stemming.

2.2.1. Case Folding

Case folding converts all characters into lowercase letters to ensure consistency across textual data and reduce vocabulary redundancy [17]. Given a document D containing a sequence of characters.

$$D = \{c_1, c_2, \dots, c_n\} \tag{1}$$

the case folding operation is defined as:

$$D_{cf} = Lower(D) \tag{2}$$

where:
D = represents the original text document,
$Lower(\cdot)$ = denotes the lowercase transformation function,
D_{cf} = represents the transformed document.

2.2.2. Noise Removal

After case folding, non-alphabetic characters such as numbers, punctuation marks, and special symbols are removed to retain only meaningful textual information [18]. The cleaning process can be represented as:

$$D_{nr} = \{w_i \in D_{cf} \mid w_i \in [a-z]\} \quad (3)$$

where:
D_{nr} = denotes the cleaned document,
w_i = represents a character token,
$[a-z]$ = indicates valid alphabetic characters.

2.2.3. Whitespace Normalization

Multiple consecutive spaces generated during the cleaning process are normalized into a single space. The normalization operation is defined as:

$$D_{wn} = Normalize\ Space(D_{nr}) \quad (4)$$

where:
D_{wn} = denotes the normalized document.

2.2.4. Stopword Removal

Stopword removal eliminates commonly occurring words that contribute little semantic value to classification performance [19].

$$S = \{s_1, s_2, \dots, s_m\} \quad (5)$$

be the set of stopwords. The resulting document after stopword removal is:

$$D_{sr} = D_{wn} - S \quad (6)$$

where:
D_{sr} = denotes the filtered document,
S = represents the stopword dictionary.

2.2.5. Stemming

Stemming reduces inflected or derived words into their root forms to decrease vocabulary dimensionality while preserving semantic meaning [20]. The stemming function can be expressed as:

$$D_{stem} = Stem(D_{sr}) \quad (7)$$

where:
$Stem(\cdot)$ = denotes the stemming function,
D_{stem} = represents the stemmed document.

The output of the preprocessing stage is a normalized corpus represented as:

$$D_{final} = Stem(Remove\ Stopword(Normalize\ Space(Remove\ Noise(Lower(D))))) \quad (8)$$

This final corpus serves as the input for the subsequent Word2Vec and FastText embedding generation processes.

2.3. Word2Vec Embedding

Word2Vec is employed to capture semantic relationships among words based on their contextual usage within the corpus. This technique projects words into a continuous vector space,

where semantically similar words tend to be located closer to each other [21]. The probability of predicting a target word from its surrounding context can be expressed as:

$$P(w_t \mid Context) = \frac{\exp(v_{w_t}^T h)}{\sum_{i=1}^{V} \exp(v_i^T h)} \tag{9}$$

where v_{w_t} denotes the vector representation of the target word, h represents the hidden-layer output, and V is the vocabulary size. The resulting embeddings provide dense semantic representations that facilitate the identification of latent relationships among hadith terms and concepts.

2.4. FastText Embedding

FastText extends the Word2Vec architecture by incorporating character-level information through subword modeling. Instead of treating a word as a single unit, FastText decomposes words into character n-grams and learns vector representations for each subword component [13]. The vector representation of a word is defined as:

$$v_w = \sum_{g \in G_w} z_g \tag{10}$$

where G_w represents the set of character n-grams associated with word w, and z_g denotes the vector representation of n-gram g. This approach enables FastText to effectively represent rare words, out-of-vocabulary terms, and morphological variations commonly found in Indonesian textual data.

2.5. Multi-Word Embedding

To enrich textual representation, the embeddings generated by Word2Vec and FastText are combined using a feature fusion strategy. The fusion process aims to leverage the complementary strengths of both methods, allowing the model to simultaneously capture semantic and morphological information. The Multi-Word Embedding representation is constructed through vector concatenation:

$$V_{MWE} = [V_{W2V}; V_{FT}] \tag{11}$$

where V_{W2V} represents the Word2Vec vector, V_{FT} denotes the FastText vector, and [;] indicates the concatenation operator. If each embedding model produces a 300-dimensional vector, the resulting Multi-Word Embedding representation contains 600 dimensions. This richer representation is expected to provide more discriminative features for the classification task.

2.6. Support Vector Machine Classification

The classification stage utilizes Support Vector Machine (SVM), a supervised learning algorithm widely recognized for its effectiveness in high-dimensional text classification problems. SVM seeks to identify an optimal hyperplane that maximizes the separation margin between classes. The decision function of SVM is defined as:

$$f(x) = w^T x + b \tag{12}$$

where w represents the weight vector, xdenotes the input feature vector, and bis the bias term. The optimization objective is formulated as:

$$\min \frac{1}{2} \| w \|^2 \tag{13}$$

subject to:

$$y_i(w^T x_i + b) \geq 1 \tag{14}$$

where y_i corresponds to the class label of sample i.

A linear kernel is employed in this study due to its computational efficiency and proven effectiveness for text classification tasks involving high-dimensional feature spaces.

2.7. Classification Performance Comparison

To evaluate the effectiveness of the proposed approach, three experimental scenarios are conducted:

1. Word2Vec + SVM
2. FastText + SVM
3. Multi-Word Embedding + SVM

The classification results obtained from these scenarios are compared to determine the contribution of embedding fusion toward classification performance improvement.

2.8. Performance Evaluation

The performance of the classification model is evaluated using a confusion matrix and several widely used evaluation metrics, including Accuracy, Precision, Recall, and F1-Score. Accuracy measures the proportion of correctly classified instances:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{15}$$

Precision measures the reliability of positive predictions:

$$Precision = \frac{TP}{TP+FP} \tag{16}$$

Recall evaluates the ability of the model to identify relevant instances:

$$Recall = \frac{TP}{TP+FN} \tag{17}$$

F1-Score provides a balanced assessment of Precision and Recall:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision+Recall} \tag{17}$$

where TP denotes True Positive, TN denotes True Negative, FP denotes False Positive, and FN denotes False Negative. The experimental results obtained from the three classification scenarios are analyzed and compared based on these evaluation metrics to identify the most effective embedding strategy for Indonesian hadith text classification.

3. RESULTS AND DISCUSSIONS

3.1. Experimental Setup

This study evaluated three feature representation approaches for hadith text classification, namely Word2Vec, FastText, and the proposed Multi-Word Embedding. All feature representations were classified using the Support Vector Machine (SVM) algorithm under the same experimental settings to ensure a fair comparison. The performance of each approach was assessed using Accuracy, Precision, Recall, and F1-Score.

Table 2. Classification performance comparison.

Method	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Word2Vec + SVM	0.7442	0.7461	0.7442	0.7418
FastText + SVM	0.7008	0.7004	0.7008	0.6992
Multi-Word Embedding + SVM	0.7558	0.7568	0.7558	0.7546

The results indicate that the proposed Multi-Word Embedding approach achieved the best performance across all evaluation metrics. The model obtained an Accuracy of 75.58%, Precision of 75.68%, Recall of 75.58%, and F1-Score of 75.46%, outperforming both Word2Vec and FastText when used individually. These findings suggest that combining multiple embedding representations can provide a richer feature space and improve the effectiveness of hadith text classification.

3.2. Analysis of Word2Vec Performance

The Word2Vec-based model achieved an Accuracy of 74.42% and an F1-Score of 74.18%. These results demonstrate that Word2Vec effectively captures contextual semantic relationships among words within the hadith corpus. By learning word representations from surrounding contexts, Word2Vec enables the classifier to identify meaningful textual patterns associated with different hadith categories. However, Word2Vec represents each word as a single vector regardless of its morphological variations. Consequently, words with similar roots but different forms may be treated as separate entities, potentially limiting the model's ability to capture certain linguistic characteristics present in the dataset.

3.3. Analysis of FastText Performance

FastText produced an Accuracy of 70.08% and an F1-Score of 69.92%, which were lower than those obtained by Word2Vec. Although FastText is designed to capture subword information through character n-grams, its effectiveness may vary depending on the characteristics of the corpus. The relatively lower performance indicates that the morphological information captured by FastText alone was insufficient to represent the semantic relationships required for accurate hadith classification. Since the dataset consists of English-translated hadith texts, contextual semantics appear to play a more dominant role than character-level information in distinguishing between classes.

3.4. Impact of Multi-Word Embedding

The proposed Multi-Word Embedding approach achieved the highest performance among all evaluated methods. Compared to Word2Vec, the proposed method increased Accuracy from 74.42% to 75.58%, corresponding to an improvement of approximately 1.16 percentage points. Similarly, the F1-Score improved from 74.18% to 75.46%. The improvement demonstrates that the fusion of Word2Vec and FastText successfully combines complementary linguistic information. Word2Vec contributes contextual semantic knowledge, whereas FastText provides morphological and subword-level representations. By integrating both embeddings into a single feature space, the classifier gains access to more comprehensive textual information. The results suggest that the two embedding techniques complement each other rather than compete with one another. Consequently, the resulting feature representation becomes more discriminative, allowing the SVM classifier to identify class boundaries more effectively.

3.5. Confusion Matrix Analysis

Figure 3 presents the confusion matrix of the best-performing model, namely the Multi-Word Embedding combined with SVM. The matrix provides a detailed overview of the classification outcomes by comparing the actual and predicted classes.

As illustrated in Figure 3, most instances are concentrated along the main diagonal of the matrix, indicating that the majority of hadith texts were correctly classified into their respective categories. The model correctly identified 257 Dhaif, 320 Hasan, and 330 Shahih instances, demonstrating its capability to learn discriminative textual patterns from the proposed embedding representation. Among the three classes, the Shahih category achieved the highest number of correct predictions, followed by Hasan and Dhaif. This result suggests that the linguistic characteristics of Shahih hadiths are relatively more distinguishable within the dataset. Conversely, the Dhaif class exhibited the largest number of misclassifications, particularly toward the Hasan category, indicating a degree of semantic similarity between these classes.

The confusion matrix also reveals that misclassifications mainly occurred between neighboring authenticity categories, especially between Dhaif and Hasan. This finding suggests that certain hadith texts share overlapping semantic features, making the classification task more challenging. Nevertheless, the relatively high concentration of predictions along the diagonal confirms

that the proposed Multi-Word Embedding approach effectively captures relevant semantic information for hadith classification. Overall, the confusion matrix analysis supports the quantitative evaluation results presented in Table 2. The dominance of correctly classified instances demonstrates that combining Word2Vec and FastText representations provides a richer feature space, enabling the SVM classifier to achieve superior performance compared with single-embedding approaches.

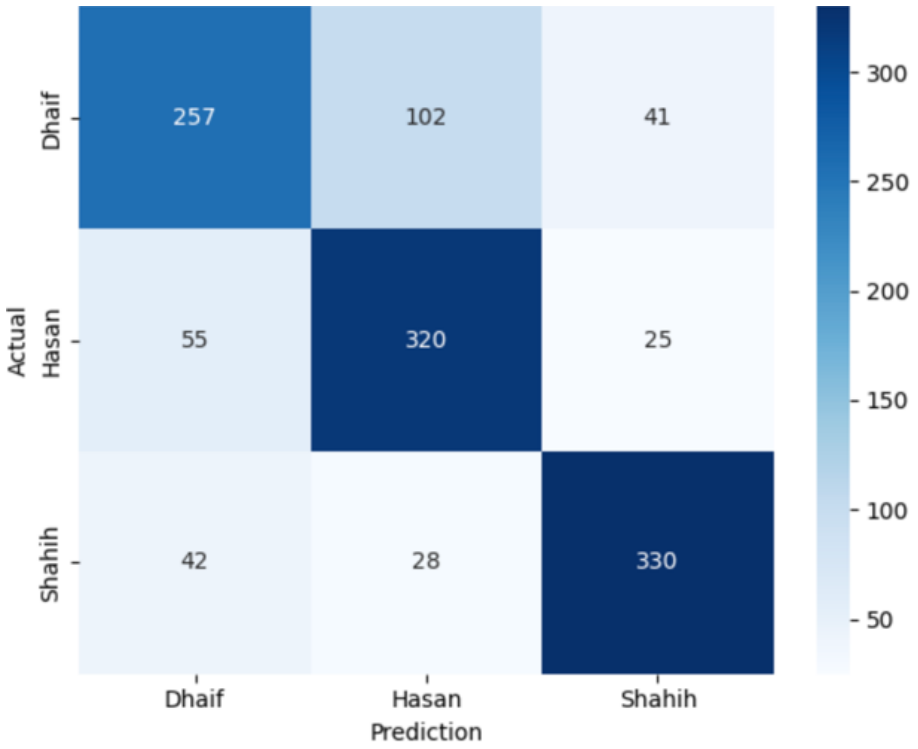

Figure 3. Confusion matrix of the proposed multi-word embedding and SVM model.

3.6. Comparative Discussion

The experimental results demonstrate that the choice of feature representation has a significant impact on hadith text classification performance. Among the three evaluated approaches, the proposed Multi-Word Embedding combined with SVM achieved the highest performance, obtaining an Accuracy of 75.58%, Precision of 75.68%, Recall of 75.58%, and F1-Score of 75.46%. These results outperform both Word2Vec + SVM and FastText + SVM, indicating that the integration of multiple embedding representations can enhance classification effectiveness. The Word2Vec-based model achieved competitive performance with an Accuracy of 74.42% and an F1-Score of 74.18%. This result suggests that contextual semantic information plays an important role in distinguishing hadith categories. Word2Vec learns word relationships based on surrounding contexts, enabling the model to capture meaningful semantic patterns from the hadith corpus.

However, Word2Vec represents each word as a single vector and does not explicitly consider subword information, which may limit its ability to handle morphological variations and infrequent terms. In contrast, FastText achieved the lowest performance among the evaluated methods, with an Accuracy of 70.08% and an F1-Score of 69.92%. Although FastText is capable of incorporating character-level information through n-gram representations, its reliance on subword features alone may not be sufficient to capture the broader contextual semantics required for hadith classification. Since the dataset consists of English-translated hadith texts, semantic context appears to contribute more significantly to classification performance than morphological information.

The superior performance of the proposed Multi-Word Embedding approach can be attributed to its ability to combine the complementary strengths of Word2Vec and FastText. Word2Vec contributes contextual semantic representations, while FastText provides additional morphological and subword-level information. The fusion of these representations produces a richer and more informative feature space, enabling the SVM classifier to learn more discriminative decision boundaries. As a

result, the model is better equipped to differentiate between hadith categories that share similar linguistic characteristics. Furthermore, the confusion matrix analysis revealed that most classification errors occurred between the Dhaif and Hasan categories.

This observation suggests that certain hadith texts belonging to these classes exhibit overlapping semantic patterns, making them inherently more difficult to distinguish. Nevertheless, the Multi-Word Embedding approach reduced the impact of such ambiguities by leveraging multiple sources of linguistic information, thereby improving overall classification accuracy. The findings of this study are consistent with previous research indicating that embedding fusion strategies can improve text classification performance by capturing diverse linguistic characteristics that may not be fully represented by a single embedding model. The results confirm that integrating contextual and morphological information leads to more robust text representations and enhances classification effectiveness.

From a practical perspective, the proposed framework can support the development of intelligent Islamic information systems capable of automatically organizing and categorizing large-scale hadith collections. In addition, the proposed Multi-Word Embedding strategy may be extended to other religious text mining applications, such as Qur'anic verse classification, Islamic document categorization, and semantic retrieval systems. Therefore, the proposed approach provides both methodological and practical contributions to the application of Natural Language Processing and Machine Learning in Islamic studies.

4. CONCLUSION

This study proposed a hadith text classification framework based on Support Vector Machine (SVM) and a Multi-Word Embedding approach that combines Word2Vec and FastText representations. The objective was to investigate whether the integration of multiple embedding techniques could improve the classification performance of hadith texts compared with single-embedding approaches. The experimental results demonstrated that feature representation plays a crucial role in determining classification effectiveness. Among the evaluated methods, the proposed Multi-Word Embedding approach achieved the best performance, obtaining an Accuracy of 75.58%, Precision of 75.68%, Recall of 75.58%, and F1-Score of 75.46%. These results outperformed both Word2Vec + SVM and FastText + SVM, indicating that the fusion of contextual semantic information and subword-level features produces a more informative representation of hadith texts.

The confusion matrix analysis further confirmed that the majority of instances were correctly classified, demonstrating the capability of the proposed model to distinguish among the Dhaif, Hasan, and Shahih categories. The findings suggest that combining multiple embedding representations can enhance the discriminative power of textual features and improve classification performance. Therefore, the proposed Multi-Word Embedding framework offers a promising solution for automated hadith classification and contributes to the advancement of Natural Language Processing applications in Islamic studies. Future research may explore the integration of contextualized embedding models such as BERT, RoBERTa, or transformer-based architectures to further improve classification accuracy. In addition, expanding the dataset with more diverse hadith collections and investigating advanced embedding fusion strategies may provide deeper insights into the automatic classification of religious texts.

ACKNOWLEDGMENTS

The authors would like to thank Universitas Sains dan Teknologi Indonesia (USTI) for its support and facilities provided throughout this research. The authors also appreciate all parties who contributed directly or indirectly to the completion of this study.

REFERENCES

[1] Audya, P. N. & Febriansyah, D. S. (2025). The Role of Hadith in Developing Contemporary Islamic Education Teaching Methodologies. Proceeding International Conference on Religion, Science and Education, **4**, 113–119.

[2] Kamran, A. B., Butt, N. A., & Basharat, A. (2026). Semantic Enrichment of Hadith Corpus—Knowledge Graph Generation From Islamic Text. *Semantic Web*, **17**(2), 22104968261431425.
[3] Umanah, R. (2024). The digital era of hadith: Challenges of authenticity and opportunities for innovation. *Al-Iftah: Journal of Islamic Studies and Society*, **5**(2), 136–148.
[4] Akbar, N. & Ali, M. (2025). Hadis sahih, hasan, daif dan maudu'. *Mahad Aly Journal of Islamic Studies*, **4**(1), 58–85.
[5] Alayed, A. (2024). Machine learning and deep learning approaches for arabic sign language recognition: A decade systematic literature review. *Sensors*, **24**(23), 7798.
[6] Chaid, A. M., Abdulrazzaq, Z. A., Sadoon, R. N., & Aljabery, M. A. (2025). Comparative analysis of innovative machine learning algorithms: Advancements in natural language processing. *Journal of Information Systems Engineering and Management*, **10**(14s), 648–668.
[7] Shin, J., Rahman, W., Ahmed, T., Mazrur, B., Mia, M. M., Ekfa, R. I., Rana, M. S., & Kim, P. (2025). Exploring the effectiveness of machine learning and deep learning algorithms for sentiment analysis: A systematic literature review. *Computers, Materials & Continua*, **84**(3), 4105–4153.
[8] Marsiani, E. S., Natsir, F., Sihombing, R. A., Izzatillah, M., & Rajiansyah, R. (2025). Support Vector Machine Based Machine Learning for Sentiment Analysis of User Reviews of the Bibit Application on Google Play Store. *JICO: International Journal of Informatics and Computing*, **1**(2).
[9] Wang, Q. (2022). Support vector machine algorithm in machine learning. *2022 IEEE international conference on artificial intelligence and computer applications (ICAICA)*, 750–756.
[10] Dzisevič, R. & Šešok, D. (2019). Text classification using different feature extraction approaches. *2019 Open Conference of Electrical, Electronic and Information Sciences (eStream)*, 1–4.
[11] Suzen, N., Gorban, A., Levesley, J., & Mirkes, E. (2026). Predicting the impact of scientific articles through semantic analysis of abstracts. *Scientometrics*, 1–72.
[12] Li, C., Xie, Z., & Wang, H. (2025). Short text classification based on enhanced word embedding and hybrid neural networks. *Applied Sciences*, **15**(9), 5102.
[13] Pertiwi, A., Azhari, A., & Mulyana, S. (2025). Fast2Vec, a modified model of FastText that enhances semantic analysis in topic evolution. *PeerJ Computer Science*, **11**, e2862.
[14] Patil, R., Boit, S., Gudivada, V., & Nandigam, J. (2023). A survey of text representation and embedding techniques in nlp. *IEEE Access*, **11**, 36120–36146.
[15] Athallah, M. R. & Lhaksmana, K. M. (2025). Hadith text classification based on topic using convolutional neural network (CNN) and TF-IDF. *Journal of Renewable Energy, Electrical, and Computer Engineering*, **5**(1), 30–36.
[16] Razaka, A. & Lhaksmana, K. (2025). Collaboration between Convolutional Neural Network and Semantic Search for English Hadith Search Using Automatic Topic Classification, TF-IDF, and Sentence-BERT. *Building of Informatics, Technology and Science (BITS)*, **7**(3), 2017–2024.
[17] Pradha, S., Halgamuge, M. N., & Vinh, N. T. Q. (2019). Effective text data preprocessing technique for sentiment analysis in social media data. *2019 11th international conference on knowledge and systems engineering (KSE)*, 1-8.
[18] Saputro, T. H. & Hermawan, A. (2021). The Accuracy Improvement of Text Mining Classification on Hospital Review through The Alteration in The Preprocessing Stage. *International Journal of Computer and Information Technology*, **10**(4), 140–146.
[19] Mashtalir, S. V. & Nikolenko, O. V. (2023). Data preprocessing and tokenization techniques for technical Ukrainian texts. *Applied Aspects of Information Technology*, **6**(3), 318–326.

[20] Siino, M., Tinnirello, I., & La Cascia, M. (2024). Is text preprocessing still worth the time? A comparative survey on the influence of popular preprocessing methods on Transformers and traditional classifiers. *Information Systems*, **121**, 102342.
[21] Nurdin, A., Aji, B. A. S., Bustamin, A., & Abidin, Z. (2020). Perbandingan kinerja word embedding word2vec, glove, dan fasttext pada klasifikasi teks. *J. Tekno Kompak*, **14**(2), 74.