



# Kahramanmaraş Sutcu Imam University

## Journal of Engineering Sciences



Geliş Tarihi : 18.04.2025  
Kabul Tarihi : 05.12.2025

Received Date : 18.04.2025  
Accepted Date : 05.12.2025

### TRI-EMBEDDINGS: A NOVEL APPROACH FOR DETECTING ABUSIVE LANGUAGE ON SOCIAL MEDIA

### TRI-EMBEDDINGS: SOSYAL MEDYADA SALDIRGAN DİL TESPİTİ İÇİN YENİ BİR YAKLAŞIM

Rezan BAKIR<sup>1\*</sup> (ORCID: 0000- 0002-4373-2231)

<sup>1</sup> Sivas University of Science and Technology, Department of Computer Engineering, Sivas, Turkey

\*Sorumlu Yazar / Corresponding Author: Rezan BAKIR, rezan.bakir@sivas.edu.tr

#### ABSTRACT

The rise in global digital communication has led to an increase in hate speech and offensive language, posing significant threats to societal well-being. While AI presents cybersecurity challenges, it also plays a crucial role in addressing these issues. Researchers must develop AI with a multidisciplinary approach, mitigating algorithmic misuse and ensuring cybersecurity. This study introduces Tri-Embeddings, an innovative method for detecting abusive language using AI-powered text analysis, applied to Twitter data. The method combines pre-trained models such as Word2Vec, FastText, and Universal Sentence Encoder (USE). Additionally, this study explores the impact of integrating a large language model, DistilBERT, into the proposed unified embedding framework. The findings demonstrate the method's effectiveness, with high precision, recall, and F1 scores, showing its potential to reduce the spread of offensive and hateful language. This approach helps mitigate ethical breaches and creates a safer, more inclusive online space.

**Keywords:** Abusive language, hate speech, machine learning, natural language processing, social networks analysis.

#### ÖZET

Küresel dijital iletişimin artışı, nefret söylemi ve uygunsuz dil içeren metinlerin yayılmasına yol açmış ve bu durum toplumsal refahı tehdit etmektedir. Yapay zekâ, siber güvenlik zorlukları yaratırken, aynı zamanda bu sorunların çözülmesinde de kritik bir rol oynamaktadır. Araştırmacılar, algoritmaların yanlış kullanımını azaltmak ve siber güvenliği sağlamak için çok disiplinli bir yaklaşım benimsemelidir. Bu çalışmada, Twitter verisi üzerinde yapay zekâ destekli metin analiziyle uygunsuz kelimeler içeren metni tespit etmek amacıyla Tri-Gömülü Temsiller (Tri-Embeddings) adı verilen yenilikçi bir yöntem sunulmaktadır. Bu yöntem, Word2Vec, FastText ve Universal Sentence Encoder (USE) gibi önceden eğitilmiş modelleri birleştirir. Semantik incelikleri yakalamak ve modelin uygunsuz kelimeler içeren metindeki ince farkları tespit etme yeteneğini artırmak için yapılan bu entegrasyon, doğruluk ve sağlamlık açısından önemli gelişmeler sağlamıştır. Ek olarak, DistilBERT gibi büyük bir dil modelinin önerilen yöntemde dâhil edilmesinin etkisi de incelenmiştir. Bulgular, yüksek doğruluk, duyarlılık ve F1 skoru değerleriyle yöntemin etkinliğini göstermektedir. Bu yaklaşım, etik ihlalleri azaltarak daha güvenli ve kapsayıcı bir çevrimiçi ortam yaratmaya yardımcı olmaktadır.

**Anahtar Kelimeler:** Saldırgan dil, nefret söylemi, makine öğrenimi, doğal dil işleme, sosyal ağ analizi.

#### INTRODUCTION

Freedom of expression is the right to voice one's thoughts, but it is not a free pass to say anything without considering its potential harm to others. Responsible communication entails sharing ideas while being mindful not to hurt others. In the dynamic world of social media, individuals come together to express their thoughts, share experiences, and engage in various conversations. However, the prevalence of abusive language is a significant concern. Abusive language detection has emerged as a vital domain within natural language processing (NLP), addressing the growing prevalence of harmful communication in online platforms. Abusive language is an umbrella term encompassing

diverse forms of harmful speech, including hate speech, offensive language, and other related subcategories like threats or harassment. By focusing on detecting and mitigating abusive language, researchers aim to foster safer and more inclusive digital environments. Abusive language is broadly defined as any form of communication that is disrespectful, harmful, or disruptive, affecting the well-being of individuals or groups. Studies such as (Fortuna & Nunes, 2018a) describe abusive language as including hate speech, offensive language, and other harmful expressions, emphasizing its detrimental impact on social cohesion and individual safety. Similarly, the authors in (Waseem & Hovy, 2016) study highlighted the need for effective detection mechanisms to address the increasing prevalence of such language in user-generated content. Hate speech and offensive language, while distinct in certain aspects, share considerable overlap. Hate speech is defined as language that expresses prejudice or hostility toward individuals or groups based on attributes such as race, religion, gender, or sexual orientation (Davidson et al., 2017). It is typically characterized by targeted and ideological intent. Offensive language, on the other hand, refers to disrespectful, vulgar, or rude expressions that may or may not target specific individuals or groups. While less ideologically driven, offensive language contributes to a hostile and unproductive communication environment (Waseem & Hovy, 2016).

Building on these observations, the decision was made to combine hate speech and offensive language in this study. This integration is intended to reflect their shared linguistic characteristics and overlapping detrimental effects on online communication. By addressing both categories within a unified framework, a more robust detection system is aimed to be developed, capable of tackling a wider range of harmful language and fostering safer, more inclusive digital environments. Both categories pose challenges for detection models due to their context-dependent nature and the use of implicit or coded language. As noted by Davidson et al. (2017), distinguishing between hate speech and offensive language can often be ambiguous, with some statements fitting into both categories depending on the context and interpretation. Combining hate speech and offensive language datasets under the broader framework of abusive language detection offers several key advantages. First, it provides a unified focus, as both hate speech and offensive language contribute to toxic online environments. Addressing them together aligns with the overarching goal of fostering safer communication spaces. Additionally, the shared linguistic features between the two categories enhance detection capabilities, as both often involve overlapping linguistic constructs, such as slurs, profanities, and offensive terms. A unified approach enables models to generalize better across these patterns. And finally, integrating these datasets simplifies the detection pipeline by reducing the need for separate preprocessing or annotation strategies, as demonstrated in studies like (Zampieri et al., 2019).

By adopting the abusive language detection framework, this study leverages a unified dataset of hate speech and offensive language to develop robust detection models. This approach aligns with previous research emphasizing the need for comprehensive detection systems that account for diverse forms of harmful communication.

Technological advancements, such as NLP and Machine Learning (ML), provide opportunities to develop effective content moderation strategies. Despite extensive research, current detection methods often fail to accurately identify subtle and context-dependent abusive language (Davidson et al., 2019a; Fortuna & Nunes, 2018b; Mozafari et al., 2022a). This study aims to address this gap by introducing a novel approach that leverages advanced NLP and machine learning techniques. This research not only highlights cybersecurity issues but also presents an innovative method for detecting abusive language on social networks. The contribution of this study to the academic domains can be abstracted as follows:

- A novel unified detection methodology for abusive language on social media platforms, which combines multiple NLP embeddings—Word2Vec, FastText, universal sentence encoder (USE), is introduced. This unified approach enhances the representation of linguistic nuances across various levels, significantly improving the model's ability to detect offensive and harmful speech. By merging these embeddings, the methodology offers a comprehensive semantic understanding, which results in more effective detection compared to models relying on individual embeddings.
- This study contributes by examining the effect of incorporating DistilBERT into the proposed Tri-Embeddings method. The analysis evaluates its impact and contribution to improving the overall performance and effectiveness of the model, providing insights into the benefits of integrating transformer-based embeddings with traditional techniques.
- The study introduces a distinctive contribution through the integration of hard voting classifiers. It explores how ensemble methods, particularly voting classifiers, can synergize diverse models to achieve robust offensive speech detection. The integration between unified embedding in the feature extraction phase and the ensemble model in the detection phase establishes a symbiotic relationship that enhances the overall efficacy of the current approach. To

the best of our knowledge, the proposed method is novel and has not been previously explored in the literature for abusive language detection, including hate speech, offensive language, and related fields.

## RELATED WORK

Harmful online behavior detection has become an essential area of research due to its impact on digital communities and individual well-being. Existing studies have investigated various aspects of harmful communication, such as hate speech, offensive language, and abusive behavior, on social media platforms (Kowalski et al., 2014, 2018; Kokkinos et al., 2014).

Abusive language on social media platforms has become a pressing concern, as it can have negative effects, such as demeaning comments and hate speech (Tawalbeh et al., 2020). This research on abusive language detection aligns with broader endeavors for creating secure online spaces. Lessons from the advancements in spam detection, as demonstrated by (Ghanem et al., 2023; Ghanem & Erbay, 2020, 2023) studies guide the proposed approach in developing robust methods to identify and combat abusive language on social networks. With the exponential growth of online content, manually identifying hate and harmful texts has become a challenging task (Sharif et al., 2021). Therefore, developing automated information processing systems capable of effectively detecting and limiting the spread of harmful content is of paramount importance. To address this issue, researchers and stakeholders are turning to ML and NLP techniques. These techniques allow for the automated detection of abusive language, enabling platforms to take necessary actions to counteract its spread. By analyzing the textual content of messages, comments, and posts, ML models can be trained to identify specific types of abusive language, including offensive language, hate speech, racism, and cyberbullying (Oriola & Kotzé, 2020). One major challenge in detecting abusive language using NLP-based approaches is the intentional obfuscation of text by malicious users (Cécillon et al., 2021). To tackle this, researchers are exploring robust techniques that can overcome such intentional manipulation. Numerous studies have contributed to the understanding and development of robust detection systems across diverse linguistic and cultural contexts. The following literature review synthesizes key findings and methodologies from relevant research endeavors. For example, the (Rizwan et al., 2020) study focused on hate speech and offensive language detection in Roman Urdu. Their work involved the creation of a lexicon for hateful words, the development of an annotated dataset, and the exploration of transfer learning. Notably, a CNN-gram deep learning architecture was proposed, showcasing the effectiveness of transfer learning over training embeddings from scratch. The (Sigurbergsson & Derczynski, 2019) research study addressed offensive language and hate speech detection for Danish, utilizing a dataset of user-generated comments from Reddit and Facebook. Their study encompassed the development of four automatic classification systems for both English and Danish, achieving noteworthy F1-scores. The broader goal was to detect hate speech and cyberbullying, emphasizing the importance of language-specific datasets. On the other hand, Davidson et al. (2017) delved into the challenge of distinguishing hate speech from offensive language. Their work involved the use of a crowd-sourced hate speech lexicon to classify tweets into hate speech, offensive language, and none. Results indicated nuances in classification, highlighting the difficulty in differentiating between categories such as racist, homophobic, and sexist tweets.

Examining racial bias in Twitter data for abusive language detection, Davidson et al. (2019) found evidence of systematic bias. Classifiers trained on African-American English tweets exhibited higher rates of abuse, underscoring potential discrimination in abusive language detection systems and their impact on specific user groups. Moreover, Mozafari et al. (2022) proposed a meta-learning-based approach for Cross-Lingual Few-Shot hate speech and offensive language detection. Their study showcased the superiority of meta-learning models over transfer learning-based models across 15 datasets and 8 languages for hate speech and 6 datasets and 6 languages for offensive language.

Likewise, Roy et al. (2022) explored hate speech and offensive language detection in Dravidian languages, employing a deep ensemble framework. The study compared machine learning and deep learning approaches, ultimately proposing an ensemble model that outperformed existing models in detecting offensive language and hate speech on social networking platforms. In the same way, Nayel & Shashirekha (2019) presented DEEP at HASOC2019, an ML Framework for hate speech and offensive language detection. Their system targeted Indo-European languages, including English, German, and Hindi, employing classical ML approaches to classify posts based on various subtasks. Furthermore, Watanabe et al. (2018) proposed a pragmatic approach to detect hate speech on Twitter, achieving high accuracy percentages in binary and ternary classification. Their method relied on unigrams and patterns from a training set, demonstrating the effectiveness of leveraging language patterns for hate speech detection in dynamic social media environments.

The study (Farha & Magdy, 2020) investigated multitask learning for Arabic offensive language and hate speech detection. The SMASH team explored deep learning, transfer learning, and multitask learning approaches, submitting a model with a CNN-BiLSTM architecture trained to identify hate speech and forecast sentiment. On the other hand, Dorris et al. (2020) introduced HateDefender, a hate speech and offensive language defense system utilizing deep LSTM neural networks. Likewise, Wei et al. (2021) proposed a method for Offensive Language and Hate Speech Detection with Deep Learning and Transfer Learning. Leveraging BI-LSTM models, transfer learning, and hyperparameter tuning, their proposed classification module included text classification, sentiment checking, and data augmentation, showcasing the necessity of a holistic approach for robust detection in public tweet data.

Recently, work in the abusive/hate speech domain has continued to evolve. For example, (Y. Zhang et al., 2025) The study examines how instruction-tuned versus human-feedback tuned large language models (LLMs) differ in sensitivity and bias when detecting abusive language, showing that tuning choice strongly affects over-/under-prediction of abusive categories. (Gaim et al., 2025) propose a multi-task benchmark for abusive language detection in low-resource settings (Tigrinya), jointly modeling abusiveness, sentiment, and topic in social media comments. The study (Fetahi et al., 2025) focuses on low-resource languages and shows how transformer models combined with explainable AI (XAI) can improve detection and transparency. In a language-specific task, (Radha & Swathika, 2025) develop a hate speech detection method for caste- and migration-related content in Tamil social media, using multilingual BERT in a fine-tuning approach. Moreover, (Kapil & Ekbal, 2025) propose a multi-task, multi-modal transformer framework for hate speech detection combining text and other modalities. These works underscore ongoing challenges in bias, interpretability, multilingual coverage, and cross-modal modeling.

In this research, the focus is on combining different embedding methods—Word2Vec, FastText, USE, and DistilBert—during the feature extraction stage. This unique blend is intended to capture various levels of meaning, making the proposed model better at understanding subtle nuances in abusive language, including offensive language and hate speech. By leveraging the strengths of each embedding method, the model gains a thorough understanding of the intricate aspects of such communication. Additionally, this research introduces a vote classifier in the detection phase. This addition adds an extra layer of sophistication to the detection process. The collective decision-making enabled by the vote classifier aims to not only improve accuracy but also enhance overall robustness. This ensemble approach, involving multiple classifiers working together, is expected to result in a more nuanced and reliable system for identifying instances of offensive language and hate speech. The novelty of using combined embeddings in this study lies in its ability to harness the complementary strengths of multiple pre-trained models to create a richer, more nuanced feature representation.

## METHODOLOGY

### *Used Dataset*

The experiments in this study were conducted using the publicly available Hate Speech and Offensive Language Dataset introduced by Davidson et al. (2017). The dataset consists of approximately 24,783 English tweets, each manually annotated into one of three categories: hate speech (~1,430 samples), offensive language (~19,190 samples), and neither/neutral (~4,163 samples). This dataset has become a benchmark in abusive language detection research due to its linguistic diversity and the variety of expressions reflecting different forms of online hostility. In the present study, the hate speech and offensive language classes were merged into a single category labeled abusive language, while neither class was retained as non-abusive, resulting in a binary classification setup suitable for robust comparative evaluation. Analysis of the annotations revealed that only around 5% of tweets were identified as hate speech by the majority of coders, with unanimous agreement occurring in just 1.3% of cases. According to the authors, this highlights the limitations of relying on lexicon-based approaches, such as Hatebase, for precise hate speech detection. Furthermore, compared with prior Twitter-based studies reporting approximately 11.6% of tweets as hate speech, the lower percentage here can be attributed to the adoption of stricter classification criteria. The majority of the corpus was labeled as offensive language, accounting for about 76% with partial agreement and 53% with full agreement, while the remaining tweets were considered non-offensive (16.6% and 11.8% under the same agreement levels, respectively) (Davidson et al., 2017). After the preprocessing phase applied in this study, the used dataset comprised 24,752 distinct tweets. The original dataset contained three categories: hate speech (**label 0**, 1,429 tweets), offensive language (**label 1**, 19,166 tweets), and neither (**label 2**, 4,157 tweets).

To simplify the classification task and focus on distinguishing abusive from non-abusive content, the hate speech and offensive language categories were merged into a single class labeled abusive language. Following this transformation, the binary dataset consisted of 20,595 abusive tweets (label 1) and 4,157 non-abusive tweets (label 0). This merging strategy aligns with prior work in abusive language detection and ensures that the resulting dataset maintains a realistic class imbalance representative of real-world online discourse.

The preprocessing involved converting the text to lowercase, removing URLs and HTML tags, eliminating non-alphanumeric characters and newlines, excluding stopwords, applying stemming, discarding punctuation marks, and removing uncertain or unusable samples. These steps are essential to normalize the text and reduce noise, ensuring that the embeddings and classifiers can focus on meaningful patterns. The dataset was initially divided into 80% training and 20% hold-out test partitions. All baseline classifiers were trained once on this split, whereas the proposed unified model—built on concatenated Word2Vec, FastText, USE, and DistilBERT embeddings and finalised as a hard-voting ensemble—was subjected to an additional 5-fold stratified cross-validation inside the training set to probe for potential overfitting and verify its robustness.

### ***Proposed Feature Extraction Method***

Due to the critical role of feature extraction and word representation in text classification, the proposed method incorporates a powerful concept known as unified embedding to enhance text classification. Unified embedding, as seen in Algorithm 1, involves the combination of three key techniques: Word2Vec, FastText, and the USE. This approach aims to furnish a robust representation for each word, enabling effective handling and discrimination by ML classifiers.

#### ***Word2Vec***

Word2Vec (Mikolov et al., 2013) is a popular technique for generating word embeddings, which are dense vector representations of words in a continuous vector space. These embeddings capture semantic similarities between words based on their contextual usage in a large corpus of text. In the context of offensive language detection, Word2Vec embeddings can be leveraged to capture nuanced linguistic patterns and semantic relationships among words, facilitating the identification of offensive or inappropriate language.

The Word2Vec model is developed using an extensive text corpus (Li et al., 2018), such as social media posts, forum discussions, or comments sections, where abusive language is prevalent. The model learns to map words to high-dimensional vectors in such a way that semantically similar words are located close to each other in the vector space. During training, Word2Vec employs either the Continuous Bag of Words (CBOW) or Skip-gram architecture to predict the context words given a target word or vice versa (Al-Saqqa & Awajan, 2019). The model parameters are optimized using techniques like stochastic gradient descent (SGD) or negative sampling to maximize the likelihood of predicting context words accurately. Once trained, the Word2Vec model generates dense vector representations for each word in the vocabulary. These embeddings capture semantic similarities and relationships among words, enabling abusive language detection algorithms to identify patterns associated with abusive language usage. By leveraging Word2Vec embeddings, abusive language detection systems can effectively capture the contextual nuances and semantic meanings of words, enhancing their ability to accurately classify text as harmful or non-harmful.

#### ***Fasttext***

FastText is an extension of the Word2Vec model introduced by Facebook AI Research (Bojanowski et al., 2017). It enhances Word2Vec by incorporating subword information, enabling it to generate word embeddings not only for complete words but also for character n-grams. This approach is particularly useful for handling out-of-vocabulary words and capturing morphological variations of words, making it well-suited for tasks like offensive language detection.

Unlike Word2Vec, which generates embeddings only for complete words, FastText considers the internal structure of words by breaking them down into character n-grams (subwords) (Lumbantoruan et al., 2022). By considering these subword units, FastText is able to capture morphological information and handle unseen or misspelled words effectively.

Similar to Word2Vec, the FastText model is trained on a large corpus of text data using either the Continuous Bag of Words (CBOW) or Skip-gram architecture. However, in addition to predicting context words given a target word

or vice versa, FastText also considers subword information during training. During the training process, FastText constructs a vocabulary consisting of both complete words and character n-grams. It generates embeddings for both unit types, refining model parameters to predict context words using the integrated representations of entire words and their subword components. After training, FastText generates embeddings not only for complete words but also for character n-grams. These embeddings are concatenated or averaged to obtain the final representation of a word. This approach allows FastText to handle unseen words or morphological variations by leveraging the embeddings of their constituent subwords. By incorporating subword information, FastText embeddings offer enhanced robustness against misspellings, slang, and out-of-vocabulary words, thereby improving the performance of abusive language detection systems, particularly in noisy and informal text domains.

### ***Universal Sentence Encoder***

The USE is a state-of-the-art model developed by Google Research for generating fixed-size vector representations of sentences (Cer et al., 2018). Unlike traditional word embedding models like Word2Vec or FastText, which operate at the word level, the USE is capable of encoding entire sentences into dense vectors, capturing their semantic meanings and contextual information (Gupta et al., 2022). The USE model architecture is based on a deep neural network trained on a large corpus of text data using a Siamese encoder-decoder architecture (Reimers & Gurevych, 2019). The encoder component of the model processes input sentences and generates embeddings that encapsulate semantic similarities and relationships among sentences. The embeddings are learned in an unsupervised manner, leveraging techniques such as transfer learning and multi-task learning to generalize across diverse text domains (Reimers & Gurevych, 2019).

Google provides pretrained versions of the USE trained on various datasets and objectives. These pretrained models include both the original USE (used in this study), which encodes sentences into 512-dimensional vectors, and the USE-Large, which produces 1024-dimensional embeddings. Additionally, specialized versions of the USE are available, such as the USE-Multilingual, designed to handle text in multiple languages.

The output of the USE is a dense vector representation of the input sentence, where each dimension of the vector captures different aspects of the sentence's semantics. These embeddings are fixed-length and semantically meaningful, enabling downstream tasks such as abusive language detection to leverage the semantic information encoded in the sentences.

By utilizing the USE, abusive language detection systems can benefit from its ability to capture nuanced semantic meanings and contextual information, thereby improving the accuracy and effectiveness of the detection process.

### ***Unified Embedding***

The unified embedding method proposed in this study integrates the three previously mentioned NLP techniques. This approach aims to enhance the representation of text data by leveraging the complementary strengths of each embedding technique, thereby improving the performance of the ML classifiers in abusive language detection tasks. The combination of Word2Vec, FastText, and USE provides a comprehensive approach to capturing both semantic meaning and contextual nuances.

While both Word2Vec and FastText are distributional embedding models, they capture complementary aspects of linguistic representation. Word2Vec effectively models global semantic relationships between words, thereby learning contextual associations that reflect meaning across large corpora. In contrast, FastText represents each word as a combination of character-level n-grams, enabling the model to capture morphological variations, handle out-of-vocabulary terms, and remain robust to noisy or misspelled content—phenomena that are especially frequent in social media text. By concatenating these two embeddings, the proposed unified representation integrates both semantic coherence and subword-level morphological awareness, resulting in richer feature spaces and improved generalization in informal and context-dependent online communication.

The USE understands not only separate words but also captures the overall meaning of whole sentences, helping us get a complete understanding of language. This is crucial for tasks like abusive language detection, where the meaning of a word can change drastically depending on its context. Combining these models leverages the strengths of each: Word2Vec's semantic relationships, FastText's morphological awareness, and USE's sentence-level understanding. This multi-faceted approach enhances the model's ability to understand and classify text accurately,

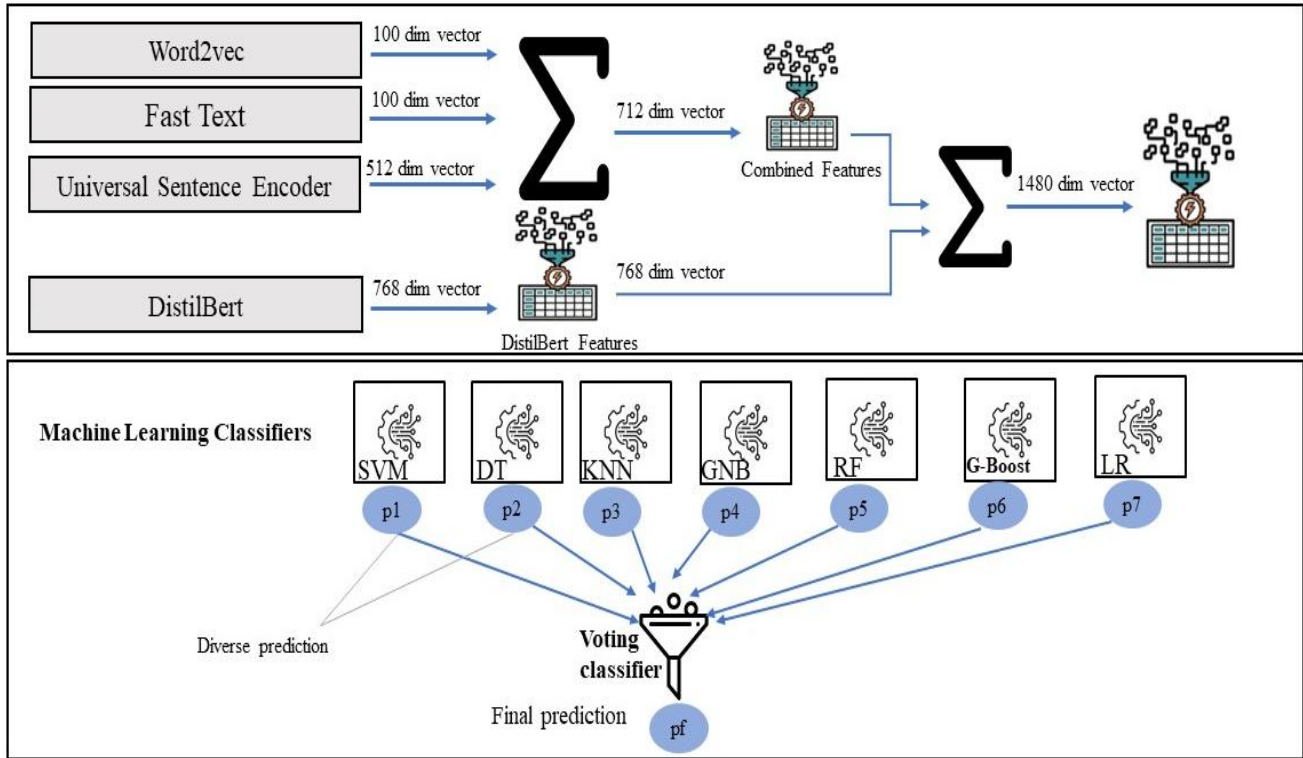
making it particularly effective for complex tasks like detecting abusive language. The specifics of the proposed methods are outlined in Algorithm 1.

### ***DistilBERT Integration***

In addition to introducing unified embeddings, this study primarily focused on integrating transformer-based model into the feature extraction phase to enhance performance in classification tasks. This was achieved by integrating DistilBERT, a distilled version of BERT, with a proposed embedding method. DistilBERT brings the power of transformer-based contextual embeddings to the mix. It captures deep contextual relationships and nuances by processing entire sentences and paragraphs, offering a richer and more dynamic representation of text. Equally important, DistilBERT is 40% smaller and roughly 60% faster than the original BERT while retaining  $\geq 97\%$  of its downstream accuracy (Sanh et al., 2018). This compactness lets us run five-fold cross-validation on a single mid-range GPU within practical time and memory budgets, and it reduces the risk of over-parameterisation (and therefore overfitting). By leveraging these diverse embedding sources, the study aimed to enrich the semantic understanding and feature representation capabilities, thereby improving the effectiveness of the integrated model in accurately classifying textual data. A secondary motivation for incorporating DistilBERT was to establish a like-for-like benchmark: by comparing the runtime and predictive quality of a pure transformer embedding against our unified (Word2Vec + FastText + USE) feature vector, we can quantify the time–accuracy trade-off and demonstrate that the proposed hybrid approach achieves comparable accuracy with markedly lower end-to-end processing time. Initially, DistilBERT was used as a standalone feature extractor, and the extracted features were fed into various ML models. Subsequently, these features were combined with those obtained from the unified embedding method and evaluated again using the same ML models. Texts were tokenized using the DistilBERT tokenizer from the transformers library, converting tweets into tokenized sequences suitable for input into the DistilBERT model. DistilBERT was configured using the DistilBert For Sequence Classification model architecture, initialized with pre-trained weights (distilbert-base-uncased).

### ***Classification Phase***

With the enriched word representations, ML classifiers were employed for text classification. Options include Support Vector Machines (SVMs), logistic Regression (LR), Gradient boosting (GB), Decision Tree (DT), K nearest neighbor (KNN), Random forest (RF). In the methodology, a voting classifier, specifically the Hard-Voting Classifier, was strategically incorporated as a key element in the approach to abusive speech detection. This ensemble technique involves combining predictions from multiple individual classifiers, each contributing a unique perspective to the decision-making process. The voting classifier aggregates the outputs of diverse models, including logistic regression, RF, and SVM. This ensemble approach aims to capitalize on the strengths and diversity of individual classifiers, fostering a more robust and accurate abusive language detection system. The hard-voting classifier was utilized in this study. The hard-voting classifier makes decisions based on a majority vote from its constituent classifiers. By considering the collective opinion of the ensemble, this mechanism enhances the reliability of predictions, particularly when individual models exhibit complementary strengths. As a result, the unified embeddings play a pivotal role in providing nuanced word representations, while the voting classifier excels in synthesizing these representations for effective decision-making. The proposed approach is illustrated in Figure 1. After the first stage, which involves preprocessing the raw data, the processed data proceeds to the feature extraction phase, where features are extracted using three proposed methods: Word2Vec, FastText, and USE. The resulting embedding vectors, with dimensions of 100, 100, and 512 for Word2Vec, FastText, and USE, respectively, are combined to form the final feature vector with a dimensionality of 712 at this stage. This vector is then passed to the classification stage, where six conventional learners—selected and tuned via an inner stratified 5-fold grid search—are trained with the following hyper-parameters: KNN ( $k = 5$ , Euclidean distance), RF (100 trees, Gini impurity,  $\max\_features = \text{sqrt}$ ), SVM (RBF kernel,  $C = 1.0$ ,  $\gamma = \text{“scale”}$ ), LR (L2 penalty,  $\text{class\_weight} = \text{balanced}$ ,  $\max\_iter = 1000$ ), DT (Gini criterion, no depth limit) and GB (100 estimators, learning-rate = 0.1,  $\max\_depth = 3$ ). The predictions produced by these six base models are subsequently aggregated by a hard-voting ensemble to yield the final decision. The final models were evaluated using different evaluation metrics. After that, the DistillBert pretrained model was used to extract features after preparing the input data to fit the model. The extracted feature vector was combined with the previous feature vector and fed into the ML classification phase, and the proposed models were evaluated again.



**Figure 1.** The Proposed Method

## EXPERIMENTAL RESULTS AND DISCUSSION

### *Experimental Setup and Evaluation Metrics*

The experiments were performed on a computational setup comprising an 11th Gen Intel(R) Core (TM) i7-11700 processor with a clock speed of 2.50GHz and 32 GB of RAM. The implementation of the experiments was carried out using the Python programming language. Standard evaluation metrics were utilized in this study, such as F1, Recall, and Precision to assess the performance of the proposed method.

Precision is a critical metric in the context of abusive language detection and other classification tasks. It evaluates the correctness of a model's positive predictions by determining the ratio of accurately identified positive cases to the total instances classified as positive. In the context of detecting abusive language in social networks, precision would indicate the effectiveness of the model in correctly identifying abusive content without misclassifying non-harmful content as abusive language. The equation for precision is defined as:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \quad (1)$$

Where:

True Positives (TP) are the instances correctly identified as positive (abusive) by the model.

False Positives (FP) are the instances incorrectly identified as positive by the model.

In other words, Precision quantifies the percentage of correctly detected abusive instances (TP) among all cases classified as abusive (TP + FP). It evaluates the reliability of the model's positive predictions. Recall, also known as sensitivity or true positive rate, is a crucial metric in classification tasks, including abusive language detection. It quantifies the ability of a model to correctly identify all positive instances from the total number of actual positive instances.

In the context of detecting abusive language in social networks, recall measures the model's effectiveness in capturing all abusive content without missing any. A high recall value indicates that the model is adept at identifying abusive language, minimizing false negatives.

The equation for recall is defined as:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \quad (2)$$

### Algorithm 1. Unified Feature-Extraction Pipeline

**Input** : tweet\_texts – list of raw tweet strings

**Output** : feature\_vectors – list of 1476-dimensional unified vectors

#### 1. Model Initialisation (run once)

- 1.1 Load pre-trained Word2Vec model (100 features)
- 1.2 Load pre-trained FastText model (100 features)
- 1.3 Load Universal Sentence Encoder (512 features)
- 1.4 Load DistilBERT tokenizer and encoder (768 features)

#### 2. Create an empty container

feature\_vectors ← [ ]

#### 3. For each tweet $t \in \text{tweet\_texts}$ do

- 3.1 Light cleaning – remove URLs, @mentions, hashtags, emojis, punctuation; convert to lower-case
- 3.2 Word2Vec embedding – tokenize  $t$ ; average in-vocabulary vectors  $\rightarrow w2v\_vec \in \mathbb{R}^{100}$
- 3.3 FastText embedding – sentence vector  $\rightarrow ft\_vec \in \mathbb{R}^{100}$
- 3.4 USE embedding – sentence vector  $\rightarrow use\_vec \in \mathbb{R}^{512}$
- 3.5 DistilBERT embedding –
  - a) tokenize  $t$  with DistilBERT tokenizer
  - b) pass tokens through the encoder
  - c) mean-pool token outputs  $\rightarrow db\_vec \in \mathbb{R}^{768}$
- 3.6 Late fusion – concatenate:  
 $v = [w2v\_vec \parallel ft\_vec \parallel use\_vec \parallel db\_vec] \in \mathbb{R}^{1476}$
- 3.7 Append  $v$  to feature\_vectors

#### 4. Return feature\_vectors

In other words, recall represents the proportion of correctly identified abusive instances (True Positives) out of all actual abusive instances (True Positives + False Negatives). It measures the model's ability to correctly identify all positive instances.

Finally, the F1 score is a metric that combines precision and recall into a single value, providing a balanced measure of a model's performance in classification tasks. It is calculated as the harmonic mean of precision and recall:

$$F_1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \quad (3)$$

The F1 score considers both false positives (precision) and false negatives (recall) and is particularly useful when there is an imbalance between positive and negative instances in the dataset. It rewards models that achieve high precision and recall simultaneously, making it a suitable metric for evaluating the overall effectiveness of a model in abusive language detection. The results obtained in this study are presented in the following Tables.

**Table 1.** Results of the Proposed Tri-embedding Method on the Davidson et al. (2017) Dataset

Classifier	Precision	Recall	F1-score
LR	0.8338	0.9990	0.9088
DT	0.8383	0.8279	0.8331
KNN	0.8378	0.9678	0.8981
RF	0.8349	0.9954	0.9080
GB	0.8344	0.9981	0.9090
SVM	0.8337	1.0000	0.9091
Hard voting Ensemble	0.8338	1.0000	0.9092

**Table 2.** DistilBERT-based Results on the Davidson et al. (2017) Dataset

Classifier	Precision	Recall	F1-score
LR	0.8351	0.9964	0.9086
DT	0.8361	0.8221	0.8290
KNN	0.8360	0.9685	0.8974
RF	0.8353	0.9944	0.9079
GB	0.8346	0.9987	0.9089
SVM	0.8334	1.0000	0.9091
Hard voting Ensemble	0.8341	0.9993	0.9093

**Table 3.** Results of the Proposed Tri-embedding Method Including DistilBERT Embedding

Classifier	Precision	Recall	F1-score
LR	0.8366	0.9913	0.9074
DT	0.8411	0.8211	0.8310
KNN	0.8379	0.9670	0.8978
RF	0.8350	0.9947	0.9079
GB	0.8349	0.9964	0.9085
SVM	0.8334	1.0000	0.9091
Hard voting Ensemble	0.8342	0.9988	0.9091

**Table 4.** Results of the Proposed Tri-embedding Method Including DistilBERT with 5-fold Cross Validation on the Davidson et al. (2017) Dataset

Classifier	Precision	Recall	F1-score
LR	0.8620	0.6415	0.7356
DT	0.8385	0.8200	0.8292
KNN	0.8372	0.9647	0.8964
RF	0.8340	0.9976	0.9085
GB	0.8344	0.9971	0.9085
SVM	0.8321	1.0000	0.9083
Hard voting Ensemble	0.8340	0.9983	0.9088

**Table 5** Results of the Proposed Tri-embedding Method in Detecting Offensive Language with 5-fold Cross Validation on the Davidson et al. (2017) Dataset

Classifier	Precision	Recall	F1-score
LR	0.9740	0.7916	0.8734
DT	0.8967	0.8905	0.8936
KNN	0.8916	0.9565	0.9229
RF	0.8996	0.9635	0.9305
GB	0.9197	0.9533	0.9362
SVM	0.9125	0.9641	0.9376
Hard voting Ensemble	0.9178	0.9572	0.9370

**Table 6.** Results of the Proposed Tri-Embedding Method in Detecting Hate Speech with 5-Fold Cross Validation on the Davidson et al. (2017) Dataset

Classifier	Precision	Recall	F1-score
LR	0.7337	0.9102	0.8125
DT	0.6749	0.6707	0.6728
KNN	0.8382	0.6671	0.7429
RF	0.7846	0.7780	0.7813
GB	0.8051	0.8323	0.8185
SVM	0.7873	0.8518	0.8183
Hard voting classifier	0.7929	0.8583	0.8243

To strengthen generalizability, additional experiments were conducted on the TweetEval Offensive Language benchmark, which contains annotated Twitter posts labeled as offensive or non-offensive. The training, validation, and test splits include 9490, 1013, and 860 samples, respectively, totaling 11363 tweets. Each tweet averages  $19.8 \pm 7.5$  tokens (median = 18) (Barbieri et al., 2020). The class distribution is moderately imbalanced, with 61.2 % non-offensive and 38.8 % offensive instances. For comparison, our original dataset contains 2820 examples (1,874 non-offensive, 946 offensive) with an average tweet length of  $17.4 \pm 6.2$  tokens. Table 8 shows the obtained results of the

proposed method when applied on TweetEval dataset. On the other hand, Table 9 presents a comparison between the performance of the proposed method and that of transformer-based models reported in the literature.

**Table 7.** Performance Comparison with Previous Studies on the Davidson et al. (2017) Dataset

Study	Feature extraction with classifier	Class	Results		
			precision	Recall	F1-score
(Davidson et al., 2017)	TF-IDF 1–3 grams + POS, 80 / 10 / 10 split / Logistic Regression	offensive language	0.91	0.90	0.90
		hate Speech	0.44	0.61	0.51
(Davidson et al., 2019b)	Regularised Logistic Regression, TF-IDF n-grams, 80 / 20 split, 5-fold CV inside training	offensive language	0.96	0.88	0.92
		hate Speech	0.32	0.53	0.40
(Z. Zhang et al., 2018)	Char-CNN with dynamic routing/ 10-fold CV	hate Speech	-	-	0.61
This study	Unified embedding, 5-fold CV / hard voting classifier	hate Speech	0.7929	0.8583	0.8243
This study	Unified embedding, 5-fold CV / LR	offensive language	0.9740	0.7916	0.8734
This study	Unified embedding, 5-fold CV / hard voting classifier	offensive language	0.9178	0.9572	0.9370
This study	Unified embedding , 5-fold CV/ hard voting classifier	abusive language (including offensive language and hatespeech)	0.8340	0.9983	0.9088
This study	Unified embedding 5-fold CV/ SVM	abusive language	0.8321	1.0000	0.9083

**Table 8.** Classification Results of the Proposed Tri-Embedding Method on the Tweeteval Offensive Language Dataset (Barbieri et al., 2020)

Classifier	Precision	Recall	F1-Score	Accuracy
LR	0.7392	0.6940	0.7058	<b>0.7596</b>
SVM	0.7686	0.6222	0.6212	0.7358
RF	0.7565	0.6180	0.6160	0.7316
GB	0.7156	0.6670	0.6772	0.7404
DT	0.6098	0.6103	0.6100	0.6514
KNN	0.5775	0.5702	0.5717	0.6348
Hard Voting Ensemble	0.7570	0.6359	0.6406	0.7408

**Table 9.** Comparison of the Proposed Tri-Embedding Model with State-of-the-Art Transformer Models on the Tweeteval Offensive Language Dataset

Model	Embedding Type	Precision	Recall	F1-Score	Accuracy	Reference
BERT-base-uncased	Transformer	0.815	0.792	<b>0.803</b>	0.803	Barbieri et al., 2020
RoBERTa-base	Transformer	0.828	0.807	<b>0.817</b>	0.817	Barbieri et al., 2020
DistilBERT	Transformer (Distilled)	0.742	0.707	<b>0.718</b>	0.765	This Study
<b>Proposed TriEmbedding</b> (Word2Vec + FastText + USE)	Classical + Sentence	0.739	0.694	<b>0.706</b>	0.760	This Study

## DISCUSSION

### Unified Methods

Before applying the combined Tri-Embedding approach, each embedding technique was evaluated independently to assess its individual contribution to text representation. The standalone Word2Vec, FastText, and Universal Sentence Encoder embeddings achieved comparable performance, indicating that each model captured complementary but partially overlapping linguistic information. Word2Vec provided stable results for frequent terms, while FastText slightly improved performance on out-of-vocabulary and morphologically rich words. USE, on the other hand, yielded stronger sentence-level coherence but did not substantially outperform the word-level embeddings. Given the minimal performance gaps across these single-embedding experiments, this study suggests the integration of the

three embeddings into a unified Tri-Embedding representation to leverage their combined semantic and contextual strengths.

The integration of multiple embeddings in the unified method, both with and without DistilBERT (Tables 1 and 3), produced the best overall performance. The unified method excluding DistilBERT achieved the highest F1-scores for ensemble classifiers (e.g., hard voting and gradient boosting), showcasing its ability to leverage the complementary strengths of Word2Vec, FastText, and USE. However, including DistilBERT demonstrated only marginal improvements over the method excluding DistilBERT. While DistilBERT offers advanced contextual understanding, the unified embeddings already capture substantial semantic information, leading to diminishing returns. Although the observed improvement is numerically modest, the unified representation demonstrates greater cross-classifier stability and resilience to data irregularities, highlighting its practical value for real-world abusive language detection.

### ***DistilBERT as a Standalone Method***

DistilBERT was integrated to benchmark the unified model against a transformer-based contextual embedding. Results show that the Tri-embedding model achieves comparable accuracy while requiring less computational time, demonstrating its efficiency and practical applicability.

The model provided competitive results (Table 2), particularly in terms of precision. However, its recall and F1-scores were slightly lower compared to the proposed methods. This indicates that while transformer-based models like DistilBERT excel in capturing deep contextual relationships, their computational overhead and minimal performance gain may not justify their standalone use in all scenarios.

### ***Classifiers and Ensemble Approaches***

The hard-voting classifier consistently outperformed others, particularly when combined with the unified embeddings. This demonstrates the advantage of ensemble approaches in leveraging the diverse strengths of individual classifiers. SVM also achieved excellent results across all feature extraction methods, particularly in recall, suggesting its effectiveness in binary classification tasks like abusive language detection.

### ***5-Fold Cross-Validation Evaluation***

To assess generalisation capacity and detect potential over-fitting, the unified representation augmented with DistilBERT was subjected to five-fold stratified cross-validation on the 80% training partition. The averaged results in Table 4 confirm three central findings.

1. The hard-voting ensemble achieves the highest F1 (0.9088), followed closely by its two tree-based constituents (RF, GB) and by the kernel SVM. The maximum F1 spread among these four models is  $\leq 0.0005$ , demonstrating that, once fed the 1476-dimensional unified vector, both heterogeneous voting and single margin-based classifiers extract virtually identical—and state-of-the-art—information.
2. KNN secures the second-best recall (0.9647) but at the cost of a lower precision (0.8372), yielding an F1 of 0.8964. The single decision tree exhibits balanced but comparatively modest values (0.8385 / 0.8200 / 0.8292), reflecting its limited representational capacity once ensemble gains are removed.

Finally, Tables 5 and 6 present the experiment in which the proposed unified model is evaluated on the original, three-label Davidson et al. (2017) corpus—without merging or discarding any class. The findings indicate that the unified approach effectively identifies offensive language across different classifiers. Notably, the SVM and Hard Voting Classifier exhibit the highest performance, both achieving an F1-score of approximately 0.937. This indicates their robustness and reliability in identifying offensive language with high precision and recall. Logistic Regression also performs exceptionally well, with a precision of 0.9567 and an F1-score of 0.9360, showcasing its effectiveness in this context. The RF classifier follows closely, achieving a precision of 0.9637 and an F1-score of 0.9298, indicating its strong capability in handling the classification task.

KNN and the DT classifiers, while still performing well, exhibit slightly lower F1-scores of 0.9198 and 0.8927, respectively. These results suggest that while these classifiers are effective, they may not capture the complexities of offensive language as accurately as the top-performing models. Overall, the high precision and recall values across all classifiers reflect the robustness of the proposed method.

On the other hand, the hate-speech experiment (Table 6) shows a noticeably harder classification problem than the offensive-language task, yet the unified representation still delivers a substantial performance margin over previously

published baselines ((Davidson et al., 2017):  $F1 \approx 0.51$ ). Across our models, the Hard-Voting ensemble attains the highest F1 (0.824), balancing a solid precision of 0.793 with a recall of 0.858. GB and the RBF-SVM form a close second tier ( $F1 \approx 0.818$ ), indicating that both margin-based and tree-ensemble learners can exploit the richer 1 476-dimensional vector. LR displays the highest recall (0.910) among all classifiers, at the expense of precision (0.734), suggesting that a linear boundary is able to flag most hate tweets but admits a larger share of false positives. Conversely, KNN yields the top precision (0.838), yet its lower recall (0.667) limits the F1 to 0.743, illustrating the usual recall-precision trade-off in instance-based methods. The single DT posts the weakest figures ( $F1 = 0.673$ ), confirming that hate-speech cues are too dispersed for a shallow tree to capture. Overall, every classifier trained on the unified embedding surpasses the TF-IDF logistic baseline by at least +0.26 absolute F1, with the ensemble registering a +0.31 gain. These results demonstrate that the proposed combination of Word2Vec, FastText, USE, and DistilBERT substantially enhances hate-speech detection while retaining a favorable precision–recall balance.

Building upon these findings, the proposed model was further validated on the TweetEval benchmark dataset. As shown in Table 8, the proposed Tri-Embedding model demonstrates competitive performance on the TweetEval benchmark. LR achieved the highest overall accuracy (75.96%) and F1-score (70.58%), while the Hard-Voting ensemble attained balanced precision (0.757) and strong recall (0.636). Although these results are slightly below the performance of large transformer-based models, they confirm that integrating classical embeddings (Word2Vec, FastText, USE) still yields effective representations for offensive language detection with reduced computational cost.

To further evaluate the robustness and generalizability of the proposed Tri-Embedding model, its performance was compared with several transformer-based models that represent the current state-of-the-art in offensive language detection. These include BERT-base, and RoBERTa, as reported in the original TweetEval benchmark study (Barbieri et al., 2020), as well as DistilBERT, which serves as a lighter yet competitive variant. While these models leverage contextual embeddings learned from large-scale pretraining, the proposed Tri-Embedding approach integrates classical distributed representations (Word2Vec and FastText) with the Universal Sentence Encoder, providing a balanced trade-off between accuracy and computational efficiency. The comparative results are summarized in Table 9.

The results presented in Table 9 clearly indicate that transformer-based architectures such as BERT and RoBERTa, achieve the highest F1-scores on the TweetEval dataset due to their deep contextual understanding of linguistic nuances. However, the proposed Tri-Embedding model achieves a competitive F1-score of 0.706 and accuracy of 0.760, closely approaching the performance of DistilBERT (0.718 F1, 0.765 accuracy) while maintaining a significantly lower computational footprint. The claim of computational efficiency is based on the structural properties of the embedding models used. Word2Vec and FastText are shallow neural architectures trained with efficient algorithms such as skip-gram with negative sampling, requiring only CPU-level operations. Similarly, the universal sentence encoder operates with a fixed encoder and does not require fine-tuning in our setup. These characteristics eliminate the need for GPU backpropagation and significantly reduce memory consumption compared to transformer-based models such as BERT and RoBERTa, which contain hundreds of millions of parameters. Therefore, the proposed Tri-Embedding model provides a lightweight and computationally efficient alternative, as supported by prior literature. Furthermore, because this study focuses primarily on accuracy comparison rather than runtime profiling, we rely on well-established references describing the computational efficiency of Word2Vec (Mikolov et al., 2013), FastText (Bojanowski et al., 2017), and the Universal Sentence Encoder (Cer et al., 2018), all of which are documented to operate with significantly lower resource consumption than transformer architectures. This highlights the model's suitability for environments with limited processing resources or real-time constraints, where transformer-based solutions may be impractical. Moreover, by combining Word2Vec, FastText, and the Universal Sentence Encoder, the Tri-Embedding model effectively captures both word-level semantics and sentence-level contextual relationships, demonstrating that hybrid embedding strategies can still provide robust results without extensive fine-tuning or large-scale pretraining. In future work, this approach can be further extended through task-specific fine-tuning or adaptive weighting of embeddings to bridge the remaining performance gap with large transformer models while preserving efficiency.

## CONCLUSION

The rapid expansion of the digital landscape has significantly amplified the impact of language, both positively and negatively. While the connectivity facilitated by the digital realm encourages collaboration and communication, it has also accelerated the troubling proliferation of hate speech and offensive language. Addressing these challenges

necessitates robust technological interventions for the detection and prevention of harmful content. This research endeavors to contribute to this critical effort by introducing a novel approach to abusive language detection. The proposed method integrates Word2Vec, FastText, and USE during the feature extraction phase. This integrated approach aims to capture semantic information at various levels, thereby enhancing the model's ability to discern subtle nuances in abusive language, including offensive language and hate speech. Moreover, this research incorporates a vote classifier in the detection phase, facilitating collective decision-making to improve both accuracy and robustness. Furthermore, the study employs the DistilBERT pretrained model—a transformer-based model—in the feature extraction phase to test the robustness of the proposed method. The results indicate that the proposed method outperforms DistilBERT when used as a standalone embedding, demonstrating more stable performance across classifiers. Interestingly, integrating DistilBERT features with the Tri-Embedding representation did not yield substantial additional gains, suggesting that Word2Vec, FastText, and USE already capture a rich and complementary spectrum of semantic and contextual information. Furthermore, when the proposed method was applied specifically to offensive language detection, it produced notably strong results. Although transformer-based architectures generally achieve the highest absolute accuracy, the proposed Tri-Embedding approach offers a practical and competitive alternative for real-world environments where computational constraints, latency requirements, or limited GPU availability make large transformer models less feasible. In conclusion, the proposed method offers a balanced combination of performance, interpretability, and efficiency, demonstrating strong potential for real-world abusive language detection in dynamic online environments.

Although the study does not directly measure ethical or societal impact, the improved precision and recall in detecting abusive language imply a tangible contribution to safer online interactions. By accurately identifying and limiting the spread of harmful content, the proposed model indirectly supports ethical communication and inclusivity on social platforms.

Future works could explore enhanced embedding techniques, including the incorporation of contextual information, and continuous model evaluation and improvement to adapt to evolving language patterns and emerging forms of abusive language. Future research could also focus on adaptive weighting strategies or lightweight fine-tuning mechanisms to further enhance the model's scalability and domain generalization.

Additionally, attention should be given to addressing limitations such as generalization to diverse languages and cultures, data bias and imbalance, and scalability and computational resources. By addressing these areas, future research endeavors can further advance the effectiveness and applicability of abusive language detection systems, contributing to the creation of a more positive and inclusive online environment for all users.

## **ACKNOWLEDGMENT**

The authors would like to thank (Davidson et al., 2017) for sharing their dataset.

## **STATEMENTS AND DECLARATIONS**

### ***Conflicts of Interest***

The author declares that there is no conflict of interest.

### ***Data Availability Statement***

The data supporting the findings of this study are accessible through the dataset provided in the paper by (Davidson et al., 2017) with additional information available from the corresponding author upon request.

### ***Competing Interests and Funding***

This research was conducted without external funding.

### ***Artificial Intelligence Contribution Statement***

The authors declare that artificial intelligence tools were used only for language editing, grammar correction, and formatting assistance during the preparation of this manuscript. All conceptualization, methodology development, experimental design, data analysis, interpretation of results, and final scientific decisions were conducted solely by the authors. The authors take full responsibility for the content of this study.

## REFERENCES

- Al-Saqqa, S., & Awajan, A. (2019). The use of word2vec model in sentiment analysis: A survey. *Proceedings of the 2019 International Conference on Artificial Intelligence, Robotics and Control*, 39–43. <https://doi.org/10.1145/3388218.3388229>
- Barbieri, F., Camacho-Collados, J., Neves, L., & Espinosa-Anke, L. (2020). TweetEval: Unified benchmark and comparative evaluation for tweet classification. *ArXiv Preprint ArXiv:2010.12421*. <https://doi.org/10.18653/v1/2020.findings-emnlp.148>
- Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5, 135–146. [https://doi.org/10.1162/tacl\\_a\\_00051](https://doi.org/10.1162/tacl_a_00051)
- Cécillon, N., Labatut, V., Dufour, R., & Linares, G. (2021). Graph embeddings for abusive language detection. *SN Computer Science*, 2, 1–15. <https://doi.org/10.1007/s42979-020-00413-7>
- Cer, D., Yang, Y., Kong, S., Hua, N., Limtiaco, N., John, R. S., Constant, N., Guajardo-Cespedes, M., Yuan, S., & Tar, C. (2018). Universal sentence encoder. *ArXiv Preprint ArXiv:1803.11175*. <https://doi.org/10.18280/ria.350404>
- Davidson, T., Bhattacharya, D., & Weber, I. (2019a). Racial bias in hate speech and abusive language detection datasets. *ArXiv Preprint ArXiv:1905.12516*. <https://doi.org/10.18653/v1/w19-3504>
- Davidson, T., Warmusley, D., Macy, M., & Weber, I. (2017). Automated hate speech detection and the problem of offensive language. *Proceedings of the International AAAI Conference on Web and Social Media*, 11(1), 512–515. <https://doi.org/10.1609/icwsm.v11i1.14955>
- Dorris, W., Hu, R., Vishwamitra, N., Luo, F., & Costello, M. (2020). Towards automatic detection and explanation of hate speech and offensive language. *Proceedings of the Sixth International Workshop on Security and Privacy Analytics*, 23–29. <https://doi.org/10.1145/3375708.3380312>
- Farha, I. A., & Magdy, W. (2020). Multitask learning for Arabic offensive language and hate-speech detection. *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, 86–90.
- Fetahi, E., Susuri, A., Hamiti, M., Kastrati, Z., Canhasi, E., & Misini, A. (2025). Enhancing social media hate speech detection in low-resource languages using transformers and explainable AI. *Social Network Analysis and Mining*, 15(1), 82. <https://doi.org/10.1007/s13278-025-01497-w>
- Fortuna, P., & Nunes, S. (2018a). A survey on automatic detection of hate speech in text. *ACM Computing Surveys (CSUR)*, 51(4), 1–30. <https://doi.org/10.1145/3232676>
- Fortuna, P., & Nunes, S. (2018b). A survey on automatic detection of hate speech in text. *Acm Computing Surveys (Csur)*, 51(4), 1–30. <https://doi.org/10.1145/3232676>
- Gaim, F., Song, H., Lee, H., Ko, C., Hwang, E. J., & Park, J. C. (2025). A Multi-Task Benchmark for Abusive Language Detection in Low-Resource Settings. *ArXiv Preprint ArXiv:2505.12116*.
- Ghanem, R., & Erbay, H. (2020). Context-dependent model for spam detection on social networks. *SN Applied Sciences*, 2, 1–8. <https://doi.org/10.1007/s42452-020-03374-x>
- Ghanem, R., & Erbay, H. (2023). Spam detection on social networks using deep contextualized word representation. *Multimedia Tools and Applications*, 82(3), 3697–3712. <https://doi.org/10.1007/s11042-022-13397-8>
- Ghanem, R., Erbay, H., & Bakour, K. (2023). Contents-Based Spam Detection on Social Networks Using RoBERTa Embedding and Stacked BLSTM. *SN Computer Science*, 4(4), 380. <https://doi.org/10.1007/s42979-023-01798-x>
- Gupta, V., Dixit, A., & Sethi, S. (2022). A Comparative Analysis of Sentence Embedding Techniques for Document Ranking. *Journal of Web Engineering*, 21(7), 2149–2185. <https://doi.org/10.13052/jwe1540-9589.2177>
- Kapil, P., & Ekbal, A. (2025). A transformer based multi task learning approach to multimodal hate speech detection. *Natural Language Processing Journal*, 11, 100133. <https://doi.org/10.1016/j.nlp.2025.100133>
- Kokkinos, C. M., Antoniadou, N., & Markos, A. (2014). Cyber-bullying: An investigation of the psychological profile of university student participants. *Journal of Applied Developmental Psychology*, 35(3), 204–214. <https://doi.org/10.1016/j.appdev.2014.04.001>

- Kowalski, R. M., Giumetti, G. W., Schroeder, A. N., & Lattanner, M. R. (2014). Bullying in the digital age: A critical review and meta-analysis of cyberbullying research among youth. *Psychological Bulletin*, 140(4), 1073–1137. <https://doi.org/10.1037/a0035618>
- Kowalski, R. M., Toth, A., & Morgan, M. (2018). Bullying and cyberbullying in adulthood and the workplace. *The Journal of Social Psychology*, 158(1), 64–81. <https://doi.org/10.1080/00224545.2017.1302402>
- Li, L., Xiao, L., Jin, W., Zhu, H., & Yang, G. (2018). Text Classification Based on Word2vec and Convolutional Neural Network. *Neural Information Processing: 25th International Conference, ICONIP 2018, Siem Reap, Cambodia, December 13–16, 2018, Proceedings, Part V 25*, 450–460. [https://doi.org/10.1007/978-3-030-04221-9\\_40](https://doi.org/10.1007/978-3-030-04221-9_40)
- Lumbantoruan, R., Siregar, R. U., Manik, I., Tambunan, N., & Simanjuntak, H. (2022). Analysis comparison of FastText and Word2vec for detecting offensive language. *2022 IEEE International Conference of Computer Science and Information Technology (ICOSNIKOM)*, 1–8. <https://doi.org/10.1109/icosnikom56551.2022.10034886>
- Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *ArXiv Preprint ArXiv:1301.3781*.
- Mozafari, M., Farahbakhsh, R., & Crespi, N. (2022a). Cross-lingual few-shot hate speech and offensive language detection using meta learning. *IEEE Access*, 10, 14880–14896. <https://doi.org/10.1109/access.2022.3147588>
- Mozafari, M., Farahbakhsh, R., & Crespi, N. (2022b). Cross-lingual few-shot hate speech and offensive language detection using meta learning. *IEEE Access*, 10, 14880–14896. <https://doi.org/10.1109/access.2022.3147588>
- Nayel, H. A., & Shashirekha, H. L. (2019). DEEP at HASOC2019: A Machine Learning Framework for Hate Speech and Offensive Language Detection. *FIRE (Working Notes)*, 336–343.
- Oriola, O., & Kotzé, E. (2020). Evaluating machine learning techniques for detecting offensive and hate speech in South African tweets. *IEEE Access*, 8, 21496–21509. <https://doi.org/10.1109/access.2020.2968173>
- Radha, N., & Swathika, R. (2025). SSN\_IT\_HATE@ LT-EDI-2025: Caste and Migration Hate Speech Detection. *Proceedings of the 5th Conference on Language, Data and Knowledge: Fifth Workshop on Language Technology for Equality, Diversity, Inclusion*, 84–89. <https://doi.org/10.18653/v1/2024.ltedi-1.29>
- Reimers, N., & Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. *ArXiv Preprint ArXiv:1908.10084*. <https://doi.org/10.18653/v1/d19-1410>
- Rizwan, H., Shakeel, M. H., & Karim, A. (2020). Hate-speech and offensive language detection in roman Urdu. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2512–2522. <https://doi.org/10.18653/v1/2020.emnlp-main.197>
- Roy, P. K., Bhawal, S., & Subalalitha, C. N. (2022). Hate speech and offensive language detection in Dravidian languages using deep ensemble framework. *Computer Speech & Language*, 75, 101386. <https://doi.org/10.1016/j.csl.2022.101386>
- Sharif, O., Hossain, E., & Hoque, M. M. (2021). Nlp-cuet@ dravidianlangtech-eacl2021: Offensive language detection from multilingual code-mixed text using transformers. *ArXiv Preprint ArXiv:2103.00455*.
- Sigurbergsson, G. I., & Derczynski, L. (2019). Offensive language and hate speech detection for Danish. *ArXiv Preprint ArXiv:1908.04531*.
- Tawalbeh, S. K., Hammad, M., & Al-Smadi, M. (2020). KEIS@ JUST at SemEval-2020 Task 12: Identifying multilingual offensive tweets using weighted ensemble and fine-tuned BERT. *ArXiv Preprint ArXiv:2005.07820*. <https://doi.org/10.18653/v1/2020.semeval-1.269>
- Waseem, Z., & Hovy, D. (2016). Hateful symbols or hateful people? predictive features for hate speech detection on twitter. *Proceedings of the NAACL Student Research Workshop*, 88–93. <https://doi.org/10.18653/v1/n16-2013>
- Watanabe, H., Bouazizi, M., & Ohtsuki, T. (2018). Hate speech on twitter: A pragmatic approach to collect hateful and offensive expressions and perform hate speech detection. *IEEE Access*, 6, 13825–13835. <https://doi.org/10.1109/access.2018.2806394>
- Wei, B., Li, J., Gupta, A., Umair, H., Vovor, A., & Durzynski, N. (2021). Offensive language and hate speech detection with deep learning and transfer learning. *ArXiv Preprint ArXiv:2108.03305*.

Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N., & Kumar, R. (2019). Predicting the type and target of offensive posts in social media. *ArXiv Preprint ArXiv:1902.09666*.

Zhang, Y., Hangya, V., & Fraser, A. (2025). LLM Sensitivity Challenges in Abusive Language Detection: Instruction-Tuned vs. Human Feedback. *Proceedings of the 31st International Conference on Computational Linguistics*, 2765–2780.

Zhang, Z., Robinson, D., & Tepper, J. (2018). Detecting hate speech on twitter using a convolution-gru based deep neural network. *European Semantic Web Conference*, 745–760. [https://doi.org/10.1007/978-3-319-93417-4\\_48](https://doi.org/10.1007/978-3-319-93417-4_48)