BBCS-Net: A BERT-Based CNN-BiLSTM-SVM Hybrid Architecture for Fine-Grained Cyberbullying Detection

Kunal Bhalerao, Vanita Mane
Department of Computer Engineering, Ramrao Adik Institute of Technology, D Y Patil Deemed to be University, Navi Mumbai, Maharashtra, India
E-mail: kunal.bhalerao@rait.ac.in, vanita.mane@rait.ac.in

Keywords: Cyberbullying detection, hybrid model, deep learning, CNN, BERT, bi-LSTM, SVM

Received: June 16, 2025

Cyberbullying detection remains challenging due to diverse linguistic patterns used by offenders. The inductive biases of deep learning architectures add to this challenge. Single-model approaches often capture only part of abusive language. This results in distinct but partially overlapping error spaces, limiting their effectiveness in real-world scenarios. To address this issue, we propose BBCS-Net. This is a modular hybrid framework that leverages diverse inductive biases. It combines Convolutional Neural Networks (CNNs), Bidirectional Long Short-Term Memory (BiLSTM) networks, and a margin-based Support Vector Machine (SVM) classifier. In the proposed framework, a Bidirectional Encoder Representations from Transformers (BERT) model is first fine-tuned on a publicly available fine-grained cyberbullying dataset consisting of approximately 47,000 samples across six classes. This generates contextualized word representations. CNN and BiLSTM models are then trained independently. These models learn complementary feature representations. CNNs capture localized contextual cues. BiLSTMs capture long-range sequential dependencies. Penultimate-layer features from both models are fused at the feature level and classified using an SVM. The proposed approach achieves an average accuracy and F1 Score of about 98.87% using nested cross-validation for model selection. On a held-out test set, it achieves 94.75% accuracy and a macro-F1 score of 94.75%. Error-space analysis shows that BBCS-Net recovers many model-specific errors. It corrects about 70.37% of CNN-only errors and 57.69% of BiLSTM-only errors. This highlights the effectiveness of feature-level hybridization in overcoming inductive-bias-driven failure modes. Latency analysis confirms the framework's practical feasibility. It achieves a single-sample inference latency of 17.75 ms. During batch processing, the ONNX runtime reduces this to about 4.18 ms per sample. This supports near real-time cyberbullying moderation.

Povzetek: Predlagali smo hibridni okvir BBCS-Net za odkrivanje kibernetskega ustrahovanja z združevanjem konvolucijskih nevronskih mrež (CNN), dvosmernega dolgega kratkoročnega spomina (Bi-LSTM) in klasifikatorjev podpornih vektorjev (SVM) z uporabo fino prilagojenih vgradenj BERT za izboljšanje predstavitev besedila.

1 Introduction

Social media is an essential part of our lives. we use it in different circumstances, such as entertainment, education, personal growth, and professional networking. Since it is widely available, all types of people can access it from smartphones, computers, and tablets. There are no age restrictions on many social media platforms, which negatively affects the lives of children and teenagers. Teenagers are not fully developed because they are overly dependent on social media for social interactions. When we use social media to exhibit our negative behaviors, cyberbullying develops[1]. Cyberbullying is a form of online harassment that involves the use of digital platforms like social media to insult, threaten, or troll individuals or communities. It's different from traditional bullying in its reach, persistence, and anonymity, with common types including flaming, harassment, dissing, and trolling [16]. These forms can be direct or indirect, involving bullying messages that include insults, and others using sarcastic or rude expressions [18].

The majorly affected group of people from cyberbullying is among teenagers, who generally suffer from psychological and emotional consequences such as anxiety, depression, low self-esteem, and even suicidal ideation. According to a survey, nearly 40% of social media users have experienced some form of online harassment, with teenagers being mostly vulnerable [16]. Victims of cyberbullying can normally experience both immediate and long-term effects on mental health, resulting in poor academic performance, social relationships, and overall well-being [17]. This serious trend underscores the need for a reliable, robust automatic cyberbullying detection system.

An automated cyberbullying detection system is required for timely detection and intervention to reduce the harm

caused to victims. Due to the massively exponential growth of social media data, manual detection is impractical and undoubtedly ineffective. An automated system is required to monitor digital platforms for bullying and harmful instances and effectively prevent escalation [19]. Additionally, Governments and social media have highlighted the need for a people-friendly environment by improving hate speech detection methods [20]. Studies are available in the literature on cyberbullying detection. They have utilized one of two approaches: a text-based approach and a multimodal approach. The text-based approach utilizes text-only data, whereas the multimodal approach utilizes more than one form of data, such as text, images, etc. [22].

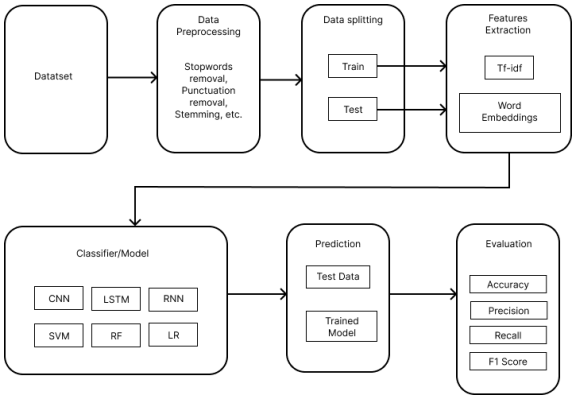

Figure 1: Cyberbully detection workflow

The general workflow for cyberbullying detection, as illustrated in Figure 1, typically consists of a sequence of key stages. First, raw textual data undergoes preprocessing to normalize and clean the input, ensuring consistency for downstream tasks. This is followed by partitioning the dataset into training and test subsets to enable model development and evaluation.

Next, feature extraction techniques are applied to transform textual data into numerical representations suitable for machine learning models. These representations may include traditional methods, such as TF-IDF, or advanced contextual embeddings, such as BERT. Subsequently, classification models ranging from conventional machine learning algorithms to deep learning architectures are trained to learn patterns associated with cyberbullying behavior.

Finally, the trained model is used for prediction, and its performance is evaluated using standard metrics such as accuracy, precision, recall, and F1-score.

However, building such a system has several challenges. One major issue is the availability of benchmark datasets for cyberbullying detection. Most existing datasets are relatively small, exhibit class imbalance, and are limited to specific languages, making it difficult to develop robust, generalizable models. Additionally, the nature of bullying language evolves continuously, rendering static datasets progressively obsolete. Furthermore, the subtle and context-dependent nature of bullying language makes it difficult for existing models to accurately detect nuanced instances of cyberbullying [18, 19].

Beyond data related challenges, the effectiveness of cyberbullying detection models is also influenced by their underlying architectural inductive biases. Recent studies have highlighted that inductive biases play a crucial role in determining what patterns deep learning models can learn and generalize [29]. In natural language processing, different architectures exhibit distinct inductive biases, leading to varied performance across linguistic phenomena [30]. For instance, convolutional models are well-suited for capturing local contextual patterns, whereas recurrent architectures are designed to model long-range dependencies. Consequently, models relying on a single architectural paradigm often learn only partial representations of complex linguistic behavior, resulting in systematic and partially overlapping error patterns. This motivates the need for hybrid approaches that integrate complementary representations to improve error coverage and robustness.

Motivated by these limitations, this work focuses on addressing inductive bias driven error spaces rather than fully resolving fine-grained class confusion at this stage. In particular, we investigate whether combining complementary feature representations learned by different neural architectures can improve error space coverage while maintaining strong generalization and deployment feasibility. The objective of this study is to design and evaluate a leakage free hybrid cyberbullying detection framework that integrates complementary inductive biases by combining CNN and BiLSTM based feature representations over task-adapted BERT embeddings, and to assess whether feature level fusion with a margin based SVM classifier can mitigate inductive bias induced error spaces under a strict train validation test evaluation protocol. We hypothesize that CNN and BiLSTM models, due to their respective inductive biases toward local contextual patterns and long-range sequential dependencies, produce complementary and partially overlapping error spaces in multi class cyberbullying detection. By fusing their learned feature representations and applying a margin based SVM classifier, the proposed framework is expected to recover a meaningful subset of both overlapping and model specific errors, while preserving stable generalization performance and practical inference efficiency. The remainder of this paper is organized as follows. Section II reviews related work spanning traditional machine learning, deep learning, and transformer based approaches for cyberbullying detection. Section III presents the proposed hybrid framework, while Section IV details the experimental setup and evaluation protocol. Section V discusses the empirical results and error space analysis, and Section VI concludes the paper with limitations and directions for future research.

2 Literature review

Cyberbullying detection has received considerable attention in recent years, driven by the rapid expansion of social media platforms and the growing prevalence of harmful online interactions. Existing approaches encompass traditional machine learning techniques, deep learning architectures, transformer based models, and hybrid frameworks. Although these methods have demonstrated promising performance, fundamental challenges persist, particularly in capturing nuanced linguistic patterns and ensuring robust generalization across diverse contexts.

Early studies primarily relied on traditional machine learning algorithms combined with handcrafted features. Mahmud et al. [1] utilized models such as LightGBM, XGBoost, and Random Forest with TF-IDF representations, achieving competitive performance on large-scale Twitter datasets. Similarly, Mahesh et al. [2] proposed a real-time cyberbullying detection system based on ensemble classifiers, demonstrating its practical applicability in streaming environments. Wiranto et al. [10] further enhanced traditional approaches through hyperparameter optimization techniques. Despite their effectiveness, these methods rely on sparse, context-independent feature representations, limiting their ability to capture the semantic and contextual nuances inherent in cyberbullying language.

To address these limitations, deep learning approaches have been widely adopted for their capacity to learn hierarchical representations directly from data. Singh et al. [7] investigated sequence-based architectures such as LSTMs and GRUs, demonstrating improved performance over traditional methods in modeling contextual dependencies. Kagithala et al. [11] utilized BiLSTM architectures to capture bidirectional contextual information, while Joseph et al. [8] conducted a comparative analysis of CNN, BiLSTM, and GRU models, highlighting their complementary strengths: CNNs for local pattern extraction and BiLSTMs for modeling long-range dependencies. However, these approaches are typically evaluated as standalone models within a single architectural paradigm, which may restrict their ability to capture the diverse linguistic characteristics present in cyberbullying content.

More recently, transformer-based models have advanced the state of the art by leveraging contextual embeddings. Chinivar et al. [3] compared embedding techniques including Word2Vec, GloVe, and BERT, demonstrating the effectiveness of contextual embeddings for fine-grained classification tasks. Tapaopong et al. [9] evaluated transformer variants, including BERT, RoBERTa, and DistilBERT, and reported strong performance across multiple classes. Jamjoom et al. [15] introduced a RoBERTa-based framework augmented with additional embeddings, achieving high classification performance. Although transformer models provide rich contextual representations, they primarily operate within a single modeling paradigm and do not explicitly incorporate complementary inductive biases from other architectures.

To overcome the limitations of single-model approaches, several studies have explored hybrid and multi-model frameworks. Balaji et al. [5] combined CNN and BiLSTM architectures, while Sarkale et al. [14] incorporated attention mechanisms into CNN-based models to enhance feature extraction. Bokolo et al. [6] evaluated combinations of deep learning and traditional classifiers, such as BiLSTM with SVM. While these approaches indicate the potential of hybridization, most implementations rely on loosely coupled architectures or ensemble strategies and do not explicitly perform structured feature-level fusion to exploit complementary representations.

Another important direction focuses on interpretability and transparency. Gongane et al. [13] integrated BiLSTM models with LIME to provide explanations for predictions, improving model transparency in sensitive applications. Additionally, multimodal approaches have been explored to incorporate diverse information sources. Ea et al. [12] investigated combined text and image-based cyberbullying detection. However, such approaches often employ loosely defined integration strategies and do not systematically address core representation challenges in textual modeling.

Despite these advancements, several key gaps remain. First, traditional machine learning methods lack contextual understanding, while deep learning and transformer-based models are typically constrained within a single architectural paradigm. Second, existing hybrid approaches do not explicitly leverage complementary inductive biases through structured feature-level fusion. Third, most studies focus on improving aggregate performance metrics without analyzing the distribution of errors across different models. As a result, inductive bias driven error spaces, where different models fail in distinct yet overlapping regions, remain insufficiently explored in the context of cyberbullying detection.

To address these limitations, this work proposes BBCS-Net, a hybrid framework that explicitly leverages inductive bias diversity by integrating CNN and BiLSTM architectures through structured feature-level fusion, followed by margin-based classification using SVM. Unlike existing approaches, the proposed framework not only achieves strong classification performance but also provides insights into model specific and overlapping error patterns, thereby improving robustness in detecting nuanced cyberbullying language.

Table 1 presents a comparative summary of the reviewed studies, including their methodologies, feature representations, performance metrics, and associated limitations.

3 Proposed framework

The proposed BBCS-Net (BERT-based BiLSTM, CNN, and SVM) hybrid framework for cyberbullying detection, shown in Figure 2, is designed to address the limitations of single architecture models by leveraging complementary

Table 1: Comparative analysis of reviewed papers (limitations are analytically inferred based on model design and reported methodology)

Authors	Model	Performance	Feature Representation	Limitation
[1]	LightGBM, XGBoost, RF	Accuracy: 85.5% F1-Score: 84.49%	TF-IDF	Relies on sparse features; limited contextual understanding
[2]	RF, AdaBoost	Accuracy: 94.06%	TF-IDF	Depends on manual feature engineering; limited semantic modeling
[3]	RoBERTa + SVM	Accuracy: 92.2%	Compared: BERT, GloVe, FastText	Models evaluated independently; lacks structured hybrid integration
[4]	BERT, BiLSTM, BiGRU	Accuracy: ∼96% (BERT)	BERT embeddings	Operates within single-model settings; does not explore complementary learning
[5]	CNN-BiLSTM, RF	Accuracy: ∼94% (RF)	GloVe	Hybridization is loosely coupled; no explicit feature-level fusion
[6]	BiLSTM, SVM, NB	Accuracy: ∼93% (SVM)	TF-IDF	Limited contextual representation; lacks deep feature integration
[7]	GRU, LSTM	Accuracy: ∼92% F1-Score: ∼92%	GloVe	Primarily models sequential dependencies; limited explicit local pattern extraction
[8]	CNN, BiLSTM, GRU	Accuracy: 83.1% (CNN)	TF-IDF	Comparative study only; no integrated hybrid framework
[9]	DistilBERT, RoBERTa	Accuracy: 94.36%	BERT variants	Limited to transformer-based representations; does not explore cross-architecture fusion
[10]	RF, SVM, LR	Accuracy: 94%	TF-IDF	Performance dependent on feature engineering; lacks deep contextual modeling
[11]	BiLSTM	Accuracy: 88.3%	Pre-trained embeddings	Focuses on sequential modeling; limited exploitation of local features
[12]	SVM (text), CNN (image)	Accuracy: 81.4%	TF-IDF + CNN	Multimodal integration is loosely defined; limited deep text representation
[13]	BiLSTM + LIME	Accuracy: 93.42%	BERT	Focuses on interpretability; does not address error reduction mechanisms
[14]	CNN + Attention	Accuracy: 80.9%	Attention-based CNN	Primarily enhances local feature extraction; limited sequential modeling capability
[15]	RoBERTa + GloVe	Accuracy: 95.9%	Transformer + GloVe	Does not explicitly perform feature-level fusion; lacks analysis of error-space diversity

inductive biases. The framework consists of three stages: text preprocessing, feature extraction using Convolutional Neural Networks (CNNs) and Bidirectional Long Short-Term Memory (BiLSTMs), and feature-level fusion, followed by classification using a Support Vector Machine (SVM). Unlike conventional approaches that rely on a single model, the proposed architecture integrates multiple learning paradigms to capture both local contextual patterns and long-range dependencies, thereby improving representation richness and mitigating inductive bias driven error spaces.

3.1 Text preprocessing

Text preprocessing is an important step in preparing The data for feature extraction and classification. Since this process has already been discussed in the introduction section, we provide a brief outline of it here. Text preprocessing helps clean raw text by removing noise, normalizing words, and preparing the data suitable for further model training. This includes text's lowercasing, URLs, and special characters removal within data, tokenizing, removing stop words, and lemmatizing.

3.1.1 Tokenization and contextual embedding generation using fine-tuned BERT

Traditional word embedding techniques like Word2Vec, GloVe, and TF-IDF struggle to capture contextual semantics in text. Cyberbullying often appears in social media, such as chats and comments, where nuanced expressions, local slang, sarcasm, and passive aggression are common. Accurate detection requires models that understand underlying contextual meaning. To address this, we utilize BERT to generate contextualized word embeddings. Unlike traditional embeddings, BERT produces representations based on the surrounding context of each word, enabling a deeper understanding of nuanced linguistic patterns. Its bidirectional architecture allows it to consider both preceding and succeeding words, making it particularly effective for capturing the semantics of cyberbullying related language. Furthermore, BERT employs subword tokenization (WordPiece), which helps handle out of vocabulary and misspelled words commonly found in social media text. In this work, BERT is first fine tuned on the training split of the cyberbullying dataset to adapt its representations to the target domain. The fine tuned BERT model is then used to extract contextualized embeddings, which serve as input features for downstream CNN and BiLSTM models for further feature extraction.

3.2 Feature extraction with CNN and bi-LSTM

The design of this feature extraction stage is motivated by the complementary inductive biases of CNN and BiLSTM architectures. CNNs are well suited for capturing local con-

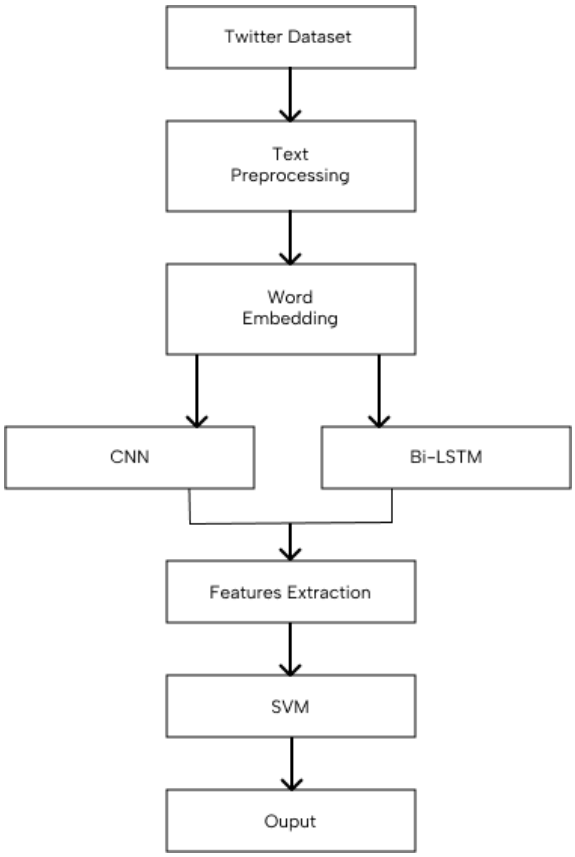

Figure 2: Proposed BBCS-net hybrid framework of cyberbullying detection system

textual patterns, such as n-grams, whereas BiLSTMs are effective at modeling long range sequential dependencies in text. Individually, these models capture only partial aspects of cyberbullying language, resulting in distinct and partially overlapping error spaces.

To address this limitation, the proposed BBCS-Net hybrid architecture integrates both CNN and BiLSTM to learn a more comprehensive representation of textual data. The CNN component extracts local contextual features, whereas the BiLSTM component captures long-range dependencies and sequential context. This combination enhances the model's ability to detect nuanced linguistic patterns, including sarcasm, implicit aggression, and context dependent expressions commonly observed in cyberbullying.

3.2.1 Convolutional neural networks (CNN)

As discussed in Section 3.2, CNNs are incorporated in the proposed architecture to capture local contextual patterns in textual data. In this work, the CNN operates on contextualized embeddings obtained from the fine-tuned BERT model, enabling it to learn meaningful local features from semantically rich representations.

The CNN architecture consists of a sequence of layers, including a convolutional layer, a non-linear activation function, pooling, and a fully connected layer. The input to the CNN is a sequence of contextual embeddings, where each token is represented as a dense vector.

The convolutional layer applies multiple filters (kernels) that slide over the input sequence to capture local n-gram features. This operation enables the model to identify important local patterns indicative of cyberbullying language. A non-linear activation function, such as ReLU, is then applied to introduce non-linearity and enhance the model's capacity to learn complex representations.

Subsequently, a max-pooling layer is used to downsample the feature maps by selecting the most salient features, thereby reducing dimensionality while retaining important information. The resulting feature maps are flattened and passed to a fully connected layer for further processing.

A 1D convolution operation is defined as:

$$C_i = f\left(\sum_{j=0}^{k-1} w_j x_{i+1} + b\right) \tag{1}$$

where:

- x is the input sequence,
- w represents convolutional filters,
- b is the bias term,
- f is the ReLU activation function.

The max-pooling operation is given by:

$$P_i = \max\left(C_{i:i+m}\right) \tag{2}$$

where m is the pooling window size.

3.2.2 Bidirectional long short-term memory (bi-LSTM)

Complementing the CNN component, BiLSTM is employed to capture long range sequential dependencies in textual data, which are essential for understanding context dependent and nuanced expressions in cyberbullying language. While CNNs focus on local patterns, BiLSTMs enable the model to learn dependencies across the entire sequence.

In this work, the BiLSTM operates on contextualized embeddings obtained from the fine-tuned BERT model. A BiLSTM consists of two LSTM networks that process the input sequence in forward and backward directions, respectively. This bidirectional processing enables the model to incorporate both past and future contextual information, resulting in a more comprehensive representation of the input sequence.

The BiLSTM architecture comprises forward and backward LSTM layers, followed by a hidden-state combination mechanism. At each time step, the hidden states from both directions are combined (e.g., via concatenation) to form a unified representation, which is then passed to subsequent layers.

Each LSTM cell is governed by gating mechanisms that regulate the flow of information:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{3}$$

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{4}$$

$$\tilde{C}_t = \tanh(W_c[h_{t-1}, x_t] + b_c) \tag{5}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{6}$$

$$o_t = \sigma(W_o[ht-1, x_t] + b_o) \tag{7}$$

$$h_t = o_t \odot \tanh(C_t) \tag{8}$$

where:

- f_t, i_t, o_t are the forget, input, and output gates,
- C_t is the cell state,
- h_t is the hidden state,
- W and b are weight matrices and biases,
- σ is the sigmoid function.

The final BiLSTM representation is obtained by combining the forward and backward hidden states:

$$H_t = [\overrightarrow{h_t}; \overleftarrow{h_t}] \tag{9}$$

where $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$ are the forward and backward hidden states.

3.3 Feature concatenation and SVM classifier

3.3.1 Feature concatenation

To derive a rich, complementary feature representation, the penultimate layer features from both the CNN and BiLSTM models are concatenated at the feature level. Unlike traditional hybrid approaches that rely on model ensembling or sequential stacking, this feature-level fusion enables the integration of complementary representations learned from different inductive biases. This allows the model to capture both localized and sequential patterns simultaneously, thereby improving coverage of model specific and overlapping error regions. The concatenation operation is defined as:

$$F = [F_{\text{CNN}}; F_{\text{BiLSTM}}] \tag{10}$$

where F_{CNN} is the feature representation generated by the CNN, and F_{BiLSTM} is the feature representation provided by the Bi-LSTM model.

3.3.2 Support vector machine (SVM) classifier

The concatenated feature vector lies in a high dimensional representation space, where conventional neural classifiers such as Softmax rely on probabilistic normalization, which can lead to overfitting, particularly when feature distributions are complex and non uniform. In contrast, Support Vector Machines (SVM) construct a maximum margin decision boundary, which enhances generalization by minimizing structural risk. Recent studies have demonstrated that integrating deep feature extraction with SVM classifiers improves robustness and classification performance compared to standard neural classifiers, particularly in hybrid architectures that combine multiple feature representations [27]. Furthermore, empirical evidence from hybrid deep learning frameworks indicates that SVM based classifiers can outperform Softmax based classifiers when applied to learned feature representations, especially in high dimensional and heterogeneous feature spaces [28]. This is because SVM focuses on maximizing the margin rather than probability estimation, making it more stable under complex feature distributions arising from feature level fusion. Therefore, in the proposed framework, SVM is employed as the final classifier to effectively leverage the fused feature space and improve generalization, particularly in mitigating classification errors arising from model-specific inductive biases. The decision function for SVM is given by:

$$f(x) = w^T x + b \tag{11}$$

where:

- w is the weight vector,
- x is the input feature vector,
- b is the bias term.

SVM minimizes the following objective function:

$$\min_{w,b} \frac{1}{2}||w||^2 + C\sum_{i=1}^{n} \max(0, 1 - y_i(w^T x_i + b)) \tag{12}$$

where C is the regularization parameter, and y_i is the true label of the i-th sample.

4 Experimental setup

A brief Experiment has been conducted to validate the effectiveness of the proposed framework.

4.1 Dataset

The dataset used in this study is the Fine-Grained Balanced Cyberbullying Dataset proposed by Wang et al. [21] and made publicly available via IEEE Dataport [23]. The dataset contains approximately 47,000 tweets annotated into multiple cyberbullying categories, including age,

ethnicity, gender, religion, other-cyberbullying, and non-cyberbullying. Each class is balanced, with approximately 8,000 samples per category, ensuring minimal class imbalance during training and evaluation. In this work, the other-cyberbullying class was excluded from the dataset due to its highly ambiguous nature and significant semantic overlap with other categories. Preliminary experimental analysis indicated that this overlap introduces inconsistent labeling and unstable decision boundaries, leading to increased misclassification across classes. To ensure reliable evaluation and stable model learning, the study focuses on well-defined and semantically distinct categories.

4.2 Preprocessing

To prepare the dataset for model training, a preprocessing pipeline consistent with BERT-based text encoding was employed. Unlike traditional preprocessing techniques that aggressively remove linguistic elements, the proposed approach preserves contextual information to align with the requirements of transformer based models. Specifically, all text samples were converted to lowercase to match the bert-base-uncased configuration. URLs were removed as they do not contribute meaningful semantic information for classification. User mentions were normalized to a special token rather than removed, allowing the model to retain interaction related context. Hashtags were processed by removing the hash symbol while preserving the underlying textual content. Importantly, punctuation, stopwords, and emojis were retained, as these elements often carry significant semantic and emotional cues in cyberbullying related text. This preprocessing strategy ensures that the contextual richness of social media language is preserved, enabling more effective representation learning through BERT embeddings.

4.3 Implementation details

The proposed framework was implemented in Python using standard data science and deep learning libraries. Data handling and preprocessing operations were performed using pandas and NumPy. Text preprocessing was performed using a custom BERT compatible pipeline, as described in Section 4.2.

Contextual word embeddings were generated using the Hugging Face Transformers library, specifically the bert-base-uncased model. The CNN and BiLSTM architectures were implemented and trained using PyTorch. Feature extraction from the penultimate layers of these models was performed to obtain learned representations, which were subsequently used for classification. The final classification stage was performed using a Support Vector Machine (SVM) implemented with the scikit-learn library.

To ensure reproducibility, all experiments were conducted using fixed random seeds. Specifically, a seed value of 42 was used across Python, NumPy, and PyTorch. Additionally, deterministic behavior was enforced by disabling non-deterministic CUDA optimizations (torch.backends.cudnn.deterministic = True and torch.backends.cudnn.benchmark = False).

All experiments were conducted on a system equipped with an Intel i5 (10th generation) processor, 16 GB RAM, and an NVIDIA GPU with 4 GB memory. CUDA acceleration was utilized to improve computational efficiency.

4.4 Model training setup

To ensure a fair, leakage free evaluation, the dataset was split globally into training, validation, and test sets at a 70:15:15 ratio. This split was consistently maintained across all components of the proposed framework. The training set was used for model learning, the validation set for model selection and hyperparameter tuning, and the test set was strictly held out for final evaluation.

The BERT model was first fine tuned on the training set and validated on the validation set to obtain task-adapted contextual embeddings. These embeddings were subsequently used as input for both CNN and BiLSTM models, which were trained independently using the same training and validation splits.

After training, features were extracted from the penultimate layers of both CNN and BiLSTM models for the training set. These features were concatenated to form a unified feature representation. To determine optimal SVM hyperparameters, a nested six fold cross validation strategy was applied on the training set, ensuring unbiased model selection. The SVM classifier was then trained using the optimal hyperparameters derived from this process.

Finally, the trained SVM model was evaluated on the held-out test set, which was not used during any stage of model training or hyperparameter tuning, thereby ensuring a reliable assessment of generalization performance.

The optimal training configurations obtained through empirical evaluation are listed below.

4.4.1 CNN training configuration

1. Batch size: 16
2. Learning rate: $1e^{-5}$
3. Optimizer: Adam
4. Loss function: Cross-entropy loss
5. Number of epochs: 5

4.4.2 BiLSTM training configuration

1. Batch size: 16
2. Learning rate: $1e^{-5}$
3. Optimizer: Adam
4. Loss function: Cross-entropy loss
5. Number of epochs: 5

4.4.3 SVM training configuration

Optimal hyperparameters obtained for SVM are:

1. Kernel Type: 'rbf'
2. Regularization Parameter (C): '10'
3. Gamma Value: 'scale'

4.4.4 Model architecture details

Model architecture details, along with its input and output dimensions, are summarised in the following tables.

CNN Architecture:

Table 2: CNN model's dimension

Layer	Input	Output
Input	(16, 80, 768)	(16, 80, 768)
Permute	(16, 80, 768)	(16, 768, 80)
Conv Block (Kernel = 2)		
Conv1D (K=2, P=1)	(16, 768, 80)	(16, 64, 81)
ReLU	(16, 64, 81)	(16, 64, 81)
AdaptiveMaxPool1D	(16, 64, 81)	(16, 64, 1)
Squeeze	(16, 64, 1)	(16, 64)
Conv Block (Kernel = 3)		
Conv1D (K=3, P=1)	(16, 768, 80)	(16, 64, 80)
ReLU	(16, 64, 80)	(16, 64, 80)
AdaptiveMaxPool1D	(16, 64, 80)	(16, 64, 1)
Squeeze	(16, 64, 1)	(16, 64)
Conv Block (Kernel = 4)		
Conv1D (K=4, P=2)	(16, 768, 80)	(16, 64, 81)
ReLU	(16, 64, 81)	(16, 64, 81)
AdaptiveMaxPool1D	(16, 64, 81)	(16, 64, 1)
Squeeze	(16, 64, 1)	(16, 64)
Conv Block (Kernel = 5)		
Conv1D (K=5, P=2)	(16, 768, 80)	(16, 64, 80)
ReLU	(16, 64, 80)	(16, 64, 80)
AdaptiveMaxPool1D	(16, 64, 80)	(16, 64, 1)
Squeeze	(16, 64, 1)	(16, 64)
Concatenation & Classification		
Concatenate (4 convs)	$4 \times (16, 64)$	(16, 256)
Dropout	(16, 256)	(16, 256)
Fully Connected	(16, 256)	(16, 5)

Bi-LSTM Architecture

Table 3: Bi-LSTM model's dimensions

Layer	Input	Output
Input	(16, 80, 768)	(16, 80, 768)
Bidirectional LSTM Layer		
Bi-LSTM (Forward)	(16, 80, 768)	(16, 80, 128)
Bi-LSTM (Backward)	(16, 80, 768)	(16, 80, 128)
Concatenation (Outputs)	$2 \times (16, 80, 128)$	(16, 80, 256)
Hidden State Extraction		
Hidden States	–	(2, 16, 128)
Forward Last Hidden	(2, 16, 128)	(16, 128)
Backward Last Hidden	(2, 16, 128)	(16, 128)
Concatenation	$2 \times (16, 128)$	(16, 256)
Classification Head		
Dropout	(16, 256)	(16, 256)
Fully Connected	(16, 256)	(16, 5)

Concatenated Feature Vector for SVM Features from the penultimate layer of CNN and Bi-LSTM have been extracted and concatenated into a single feature vector to train the SVM model.

1. CNN output features: (16, 256)
2. Bi-LSTM output features: (16, 256)
3. Concatenated feature shape: (16, 512)

4.4.5 Evaluation metrics

Performance evaluation metrics such as accuracy, precision, recall, and f-1 score have been used to evaluate our model's performance on the given dataset.

1. Accuracy: It indicates how the model correctly identifies the instances to the total instances by taking the ratio between them.
2. Precision: It indicated the percentage of positive instances that were positive themselves, hence reducing false positives.
3. Recall: It indicates the percentage of actual positive instances that are identified as positive by the model, hence reducing false negatives.
4. F1-score: It indicates the balance of precision and recall. And it's derived from the harmonic mean of precision and recall

5 Results

This section presents a comprehensive performance evaluation of the proposed BBCS-Net hybrid framework and its comparison with baseline models such as CNN and Bi-LSTM using standard performance metrics, including accuracy, precision, recall, and F1-score. In addition to conventional evaluation metrics, the analysis further incorporates error-space examination to understand the complementary behavior of individual models and the effectiveness of the hybrid approach in mitigating model-specific errors. Furthermore, latency analysis is conducted to assess the computational efficiency and the feasibility of real time deployment of the proposed framework under different execution settings.

5.1 CNN model's performance

The Convolutional Neural Network (CNN) model was trained for five epochs using BERT based embeddings as input features. The training process showed a steady decrease in training loss from 0.0738 to 0.0475, while validation accuracy remained consistently above 95% across all epochs. The best validation performance was achieved in the first epoch, with a validation accuracy of 95.08% and a validation loss of 0.2517. The average training time per epoch was approximately 195 seconds, with validation time around 40 seconds per epoch.

On the validation set, the CNN model achieved an overall accuracy of 95.08%, with macro-averaged precision, recall, and F1-score of 95.09%, 95.08%, and 95.07%, respectively. The class-wise performance of the model is presented in Table 4.

The model demonstrates strong performance across most classes. The age class achieved a precision of 98.58%, a recall of 98.08%, and an F1-score of 98.33% across 1,199 samples. Similarly, the ethnicity class achieved a precision of 98.58%, a recall of 98.91%, and an F1-score of 98.75% across 1,194 samples. The religion class also achieved high performance, with an F1-score of 96.71% across 1,199 samples.

In contrast, relatively lower performance was observed for the gender class and non-cyberbullying class. These results indicate variability in class-wise performance despite overall strong accuracy.

The class wise performance metrics are shown in Table 4.

Table 4: Class wise performance of the CNN model

Class	Precision	Recall	F1-Score	Samples
Age	0.9858	0.9808	0.9833	1199
Ethnicity	0.9858	0.9891	0.9875	1194
Gender	0.9470	0.9114	0.9288	1196
Non-Bully	0.8828	0.8909	0.8868	1192
Religion	0.9530	0.9817	0.9671	1199
Accuracy	0.9508			

The confusion matrix for the CNN model is shown in Figure 3, where the rows correspond to the true class labels and the columns to the predicted class labels. The diagonal elements indicate correctly classified instances, while off diagonal elements represent misclassifications. The model demonstrates strong classification performance across age, ethnicity, and religion, as reflected by higher values along the diagonal. However, relatively higher misclassification is observed in classes such as gender and not-cyberbullying, indicating variability in class wise prediction performance.

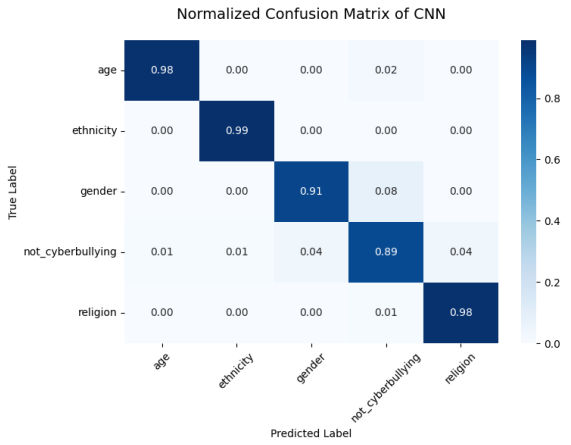

Figure 3: Confusion matrix of the CNN model

Figure 4 illustrates the class wise F1-scores for the CNN model. It can be observed that the model achieves higher F1-scores for classes such as age and ethnicity, while achieving comparatively lower F1-scores for gender and not-cyberbullying. This variation highlights differences in model performance across different categories despite high overall accuracy.

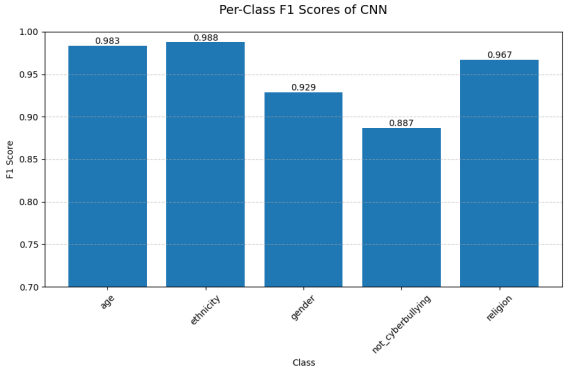

Figure 4: F1 score bar chart of the CNN model

5.2 Bi-LSTM model's performance

The Bidirectional Long Short-Term Memory (Bi-LSTM) model was trained for five epochs using BERT based embeddings as input features. The training process showed a consistent decrease in training loss from 0.0560 to 0.0383, while validation accuracy remained above 95% across all epochs. The best validation performance was achieved in the first epoch, with a validation accuracy of 95.42% and a validation loss of 0.1808. The average training time per epoch was approximately 200 seconds, with validation time around 40 seconds per epoch. On the validation set, the Bi-LSTM model achieved an overall accuracy of 95.42%, with macro-averaged precision, recall, and F1-score of 95.43%, 95.41%, and 95.42%, respectively. The class wise performance of the model is presented in Table 5. The model demonstrates strong performance across multiple classes. The age class achieved a precision of 98.66%, a recall of 98.42%, and an F1-score of 98.54%. The ethnicity class achieved a precision of 99.08%, a recall of 98.83%, and an F1-score of 98.95%. The religion class also performed well, achieving an F1-score of 96.77%. Relatively lower performance was observed for the gender class and the not-cyberbullying class. These results indicate variation in class wise performance while maintaining high overall accuracy.

Table 5: Class wise performance of the bi-LSTM model

Class	Precision	Recall	F1-Score	Samples
Age	0.9866	0.9842	0.9854	1199
Ethnicity	0.9908	0.9883	0.9895	1194
Gender	0.9458	0.9197	0.9326	1196
Non-Bully	0.8885	0.9027	0.8955	1192
Religion	0.9598	0.9758	0.9677	1199
Accuracy	0.9542			

The confusion matrix for the Bi-LSTM model is shown in Figure 5, where rows represent the true labels and columns represent the predicted labels. The matrix indicates strong classification performance for age, ethnicity, and religion, as evidenced by higher values along the diagonal. However, relatively higher misclassification is observed in the gender and not-cyberbullying classes, which

is consistent with the lower F1-scores observed for these categories. The class wise F1-score distribution for the

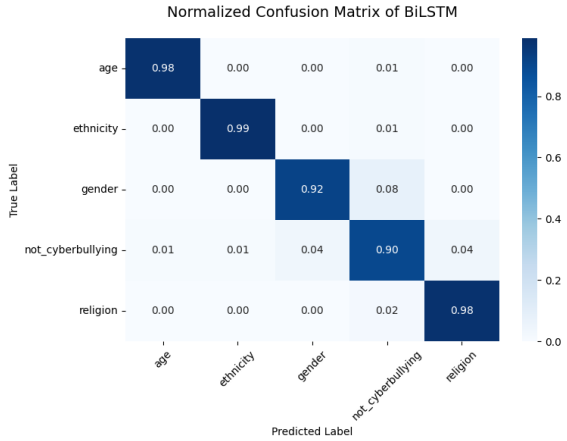

Figure 5: Confusion matrix of the bi-LSTM model

Bi-LSTM model is illustrated in Figure 6. The model achieves higher F1-scores for age and ethnicity, while comparatively lower F1-scores are observed for gender and not-cyberbullying, highlighting differences in model performance across categories.

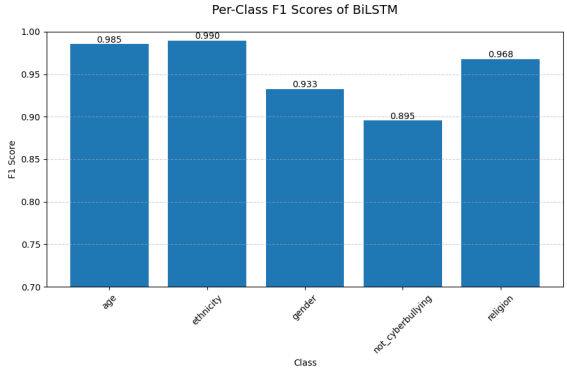

Figure 6: F1 score bar chart of the bi-LSTM model

5.3 Proposed BBCS-net frameworks performance

The performance of the proposed BBCS-Net framework was evaluated using a combination of nested cross validation on the training set and final evaluation on a held out test set. The framework integrates feature representations extracted from CNN and BiLSTM models, then classifies using a Support Vector Machine (SVM).

To determine optimal SVM hyperparameters, a nested six fold cross validation strategy was employed on the training set. The overall cross validation performance yielded an average accuracy of 98.87% ± 0.13 and an F1-score of 98.87% ± 0.13, indicating stable performance across different data splits. The optimal hyperparameters obtained were an RBF kernel with a regularization parameter (C) of 10 and gamma set to scale.

Using these optimal hyperparameters, the SVM classifier was trained on the full training set and evaluated on the held-out test set. The proposed framework achieved an overall accuracy of 94.75%, with macro-averaged precision, recall, and F1-score of 94.75%, 94.74%, and 94.75%, respectively. The class-wise performance is summarized in Table 6.

The age class achieved a precision of 98.99%, a recall of 98.00%, and an F1-score of 98.49%. The ethnicity class achieved an F1-score of 98.03%, while the religion class achieved an F1-score of 96.42%.

Relatively lower performance was observed for the gender and the non-cyberbullying class. These results indicate variability in class wise performance across categories. The

Table 6: Class wise performance of the BBCS-net framework

Class	Precision	Recall	F1-Score	Samples
Age	0.9874	0.9791	0.9832	1199
Ethnicity	0.9783	0.9799	0.9791	1194
Gender	0.9305	0.9181	0.9242	1196
Non-Bully	0.8825	0.8884	0.8855	1192
Religion	0.9589	0.9717	0.9652	1200
Accuracy	0.9475			

confusion matrix for the proposed BBCS-Net framework is shown in Figure 7, where rows represent the true labels and columns represent the predicted labels. The matrix shows a strong diagonal concentration of values, indicating correct classification across most categories. However, some degree of misclassification persists in classes such as gender and not-cyberbullying.

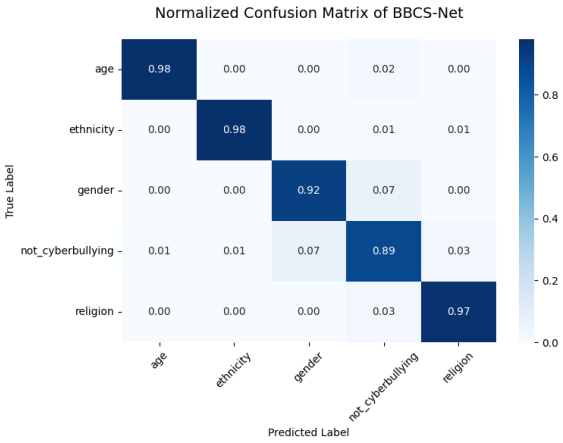

Figure 7: Confusion matrix of the proposed BBCS-net framework

The class wise F1-score distribution for the proposed framework is illustrated in Figure 8. The model achieves higher F1-scores for age, ethnicity, and religion, while comparatively lower F1-scores are observed for gender and not-cyberbullying, highlighting differences in performance across categories.

Table 7 presents a comparative evaluation of the base-

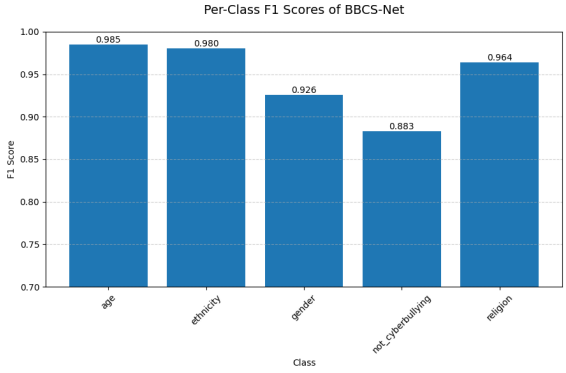

Figure 8: F1 score bar chart of the proposed BBCS-net framework

line models and the proposed BBCS-Net framework. While the individual CNN and BiLSTM models achieve slightly higher overall accuracy and F1 Scores, the proposed framework demonstrates competitive performance by integrating complementary feature representations from both architectures.

The marginal reduction in overall accuracy is accompanied by improved robustness, as the hybrid framework captures a significant portion of model specific errors, as further analyzed in the subsequent section. This highlights that overall accuracy alone may not fully reflect the model's effectiveness in handling diverse and nuanced cyberbullying patterns.

Table 7: Comparative performance of models

Model	Accuracy (%)	F1-score (%)
CNN	95.08	95.07
BiLSTM	95.42	95.42
BBCS-Net (Proposed)	**94.75**	**94.77**

5.4 Error-space analysis

To further evaluate the behavior of individual models and the proposed hybrid framework, an error-space analysis was conducted to examine misclassification patterns across the CNN, BiLSTM, and BBCS-Net models.The CNN model misclassified 322 samples, while the BiLSTM model misclassified 321 on the test set. Among these, 295 errors were overlapping, indicating instances where both models failed to correctly classify the same samples. Additionally, 27 errors were unique to the CNN model, and 26 to the BiLSTM model, reflecting differences in their prediction behavior. The proposed BBCS-Net framework resulted in 314 misclassifications on the test set. Among the overlapping errors, the hybrid model successfully corrected 29 samples, leaving 266 uncorrected. This corresponds to a correction ratio of approximately 9.83% for overlapping errors. In contrast, the hybrid framework demonstrated higher effectiveness in addressing model specific errors. It corrected approximately 70.37% of CNN-only errors and 57.69% of BiLSTM-only errors, indicating its ability to capture complementary information from both models. These results highlight the distribution of errors across individual and hybrid models and demonstrate the proposed framework's ability to recover a substantial portion of model-specific misclassifications while maintaining overall competitive performance.

5.5 Latency and deployment analysis

To evaluate the computational efficiency and real time applicability of the proposed BBCS-Net framework, latency analysis was conducted across single sample, repeated, and batch inference settings in both local and cloud based GPU environments.

Local Machine (RTX 2050 GPU):
For single sample inference, the total latency was measured at 17.75 ms, with the majority of computation attributed to the BERT encoding stage i.e 9.27 ms. When a single sample was evaluated repeatedly over 100 runs, the average latency was 15.11 ms, with a standard deviation of 0.66 ms, indicating stable inference performance. For batch processing, the model was evaluated on the full test set with a batch size of 16, resulting in a total processing time of 49,177.30 ms and a per sample latency of 8.22 ms. Further optimization using the ONNX Runtime reduced the total batch processing time to 24,977.26 ms, achieving a per sample latency of 4.18 ms and demonstrating significant improvement in inference efficiency.

Cloud Environment (Kaggle P100 GPU):
On the Kaggle P100 GPU, the single sample inference latency was 15.27 ms, with BERT encoding contributing 10.14 ms. Repeated inference over 100 runs yielded an average latency of 11.64 ms with a standard deviation of 0.68 ms, indicating consistent performance. Batch processing of the full test set with a batch size of 16 resulted in a total processing time of 19,637.38 ms, corresponding to a per sample latency of 3.28 ms. Using the ONNX runtime in the cloud environment resulted in a total processing time of 19,805.38 ms and a per sample latency of 3.31 ms.

These results provide a detailed breakdown of inference latency across different components and execution environments, demonstrating the computational feasibility of the proposed framework for large scale evaluation.

6 Discussion

The experimental results demonstrate that while individual models such as CNN and BiLSTM achieve slightly higher overall accuracy, the proposed BBCS-Net framework provides a more comprehensive and robust representation of cyberbullying patterns through feature level hybridization. The marginal reduction in overall performance is accompanied by improved coverage of model specific error regions, highlighting that evaluation based solely on aggregate metrics may not fully capture model effectiveness in nuanced classification tasks.

A key observation from the results is the presence of distinct, partially overlapping error spaces between the CNN and BiLSTM models. This behavior can be attributed to their differing inductive biases: CNN models are inherently suited for capturing local contextual patterns, while BiLSTM models are more effective in modeling long range dependencies in sequential data. As a result, each model learns a different representation of the input space and exhibits unique failure modes. The proposed hybrid framework leverages this complementarity by combining learned feature representations, enabling it to recover a substantial proportion of model specific errors, as demonstrated in the error space analysis.

The behavior of the proposed framework can be further interpreted through principles of adaptive and robust control systems. In such systems, adaptive mechanisms are employed to handle uncertainties and nonlinear dynamics by iteratively minimizing system error through feedback driven adjustments [24]. For example, adaptive backstepping control methods recursively stabilize uncertain nonlinear systems by reducing the tracking error at each stage, while output feedback control approaches estimate hidden system states under uncertainty using indirect observations [25]. Similarly, in the proposed framework, CNN and BiLSTM models act as complementary representation learners, analogous to multiple observers capturing different aspects of system dynamics. The training process, driven by backpropagation, serves as a feedback mechanism that refines model parameters based on prediction error.

Furthermore, integrating feature representations followed by SVM based classification can be viewed as a robust decision making layer. Margin based classifiers such as SVMs are designed to maintain stable decision boundaries in high dimensional spaces, analogous to robust optimal control strategies that ensure system stability in the presence of disturbances and modeling uncertainties [26]. This combination of complementary feature extraction and robust classification enhances the model's ability to generalize across diverse and evolving linguistic patterns.

From a practical perspective, the latency analysis demonstrates that the proposed framework is computationally feasible for real world deployment. The model achieves 17.75 ms single sample inference on local hardware and shows further improvements through ONNX optimization, indicating its suitability for near real time cyberbullying detection systems. Such performance is critical for applications in online content moderation and mental health monitoring, where timely intervention is essential.

However, cyberbullying language is highly dynamic and continuously evolving, which introduces challenges in maintaining model performance over time. Inspired by adaptive control strategies, the proposed framework can be extended to incorporate adaptive retraining mechanisms that periodically update the model using newly available data. Feedback driven updates based on misclassified instances could further enhance robustness in dynamic environments and improve the model's ability to handle emerging forms of cyberbullying.

Despite its strengths, the proposed framework has several limitations. First, the hybrid architecture introduces additional computational complexity due to multiple model components. Second, the dataset used in this study is limited to English language text, which may restrict generalization to multilingual scenarios. Third, while the model improves coverage of model specific errors, it does not fully resolve overlapping error regions, particularly for ambiguous or context dependent instances.

Overall, the findings suggest that combining models with complementary inductive biases offers a promising approach to improving robustness in cyberbullying detection. Rather than focusing solely on maximizing aggregate performance metrics, future research should emphasize understanding and mitigating model specific error patterns to build more reliable and adaptive systems.

Table 8: Comparison with state of the art methods

Study	Accuracy	Cross-Validation
Jamjoom et al. (2024)	95.9	95.07
Proposed (BBCS-Net)	94.75	98.87 ± 0.13

In comparison with recent state of the art transformer based approaches such as RoBERTaNET [15], which report approximately 95% test accuracy and closely match cross validation performance, it is important to interpret these results in the context of the evaluation methodology. While such performance is competitive, near identical cross validation and test results may suggest that evaluation splits are not fully independent or that feature extraction is performed prior to dataset partitioning. In contrast, the proposed BBCS-Net framework enforces a strict global data split (70–15–15) before any feature extraction or model training, thereby preventing data leakage. As a result, although the proposed framework reports a slightly lower held out test accuracy (94.75%), it achieves a consistently high cross validation performance (98.87%) with low variance, indicating strong generalization across unseen folds. This gap between cross validation and test performance reflects a more realistic evaluation scenario in which the model is exposed to truly unseen data distributions. Therefore, the proposed framework emphasizes robustness and evaluation integrity over optimistic performance estimates, providing a more reliable assessment of real world applicability.

7 Conclusion

Cyberbullying remains a critical challenge in online environments due to the diverse and evolving nature of abusive language, requiring robust and reliable automated detection systems. In this work, we proposed BBCS-Net, a hybrid cyberbullying detection framework that leverages complementary inductive biases by integrating BERT based contextual embeddings with CNN and BiLSTM architectures,

followed by an SVM classifier for margin based decision making.

Experimental results demonstrate that the proposed framework achieves competitive performance on fine grained cyberbullying classification, with 94.75% accuracy and a macro-F1 score of 94.75% on a held out test set, along with strong cross validation performance of approximately 98.8% with low variance. Beyond aggregate performance metrics, the study emphasizes the importance of understanding model behavior through error space analysis. The proposed framework effectively recovers a substantial proportion of model specific errors, correcting approximately 70.37% of CNN-only errors and 57.69% of BiLSTM-only errors, demonstrating the advantage of feature level hybridization in mitigating inductive bias driven failure modes.

In addition, latency analysis confirms the practical feasibility of the proposed system. The framework achieves 17.75 ms single sample inference latency, which is further reduced to approximately 4.18 ms per sample using ONNX runtime, enabling efficient large scale processing. This highlights the suitability of the proposed approach for near real time deployment in automated content moderation and online safety monitoring systems.

Despite these contributions, the proposed framework has certain limitations. The model is evaluated on a monolingual (English) dataset, which may limit its applicability in multilingual environments. Furthermore, the current approach does not explicitly address challenges such as sarcasm detection, passive aggressive language, or adversarial manipulation, which are prevalent in real world cyberbullying scenarios. The hybrid architecture also introduces additional computational complexity compared to single-model approaches.

Future work will focus on extending the framework to multilingual and cross domain datasets, and on incorporating multimodal inputs, such as text and images, to enhance contextual understanding. Additionally, integrating adaptive retraining strategies and explainable AI techniques will be explored to improve robustness and interpretability in dynamic online environments. Evaluating the system in real world deployment settings, such as live social media platforms, will further validate its practical effectiveness.

Overall, this study demonstrates that leveraging complementary inductive biases via hybrid architectures offers a promising approach for building more robust, reliable, and deployment ready cyberbullying detection systems.

References

[1] Mahmud M. I., Mamun M., and Abdelgawad A. (2022) "A Deep Analysis of Textual Features Based Cyberbullying Detection Using Machine Learning," *IEEE Global Conference on Artificial Intelligence and Internet of Things (GCAIoT)*, IEEE, Alamein New City, Egypt, pp. 166–170. https://doi.org/10.1109/GCAIoT57150.2022.10019058

[2] Mathur S. A., Isarka S., Dharmasivam B., and J. C. D. (2023) "Analysis of Tweets for Cyberbullying Detection," *International Conference on Secure Cyber Computing and Communication (ICSCCC)*, IEEE, Jalandhar, India, pp. 269–274. https://doi.org/10.1109/ICSCCC58608.2023.10176416

[3] Chinivar S., M. S R., J. S A., and K R V. (2022) "Comparison of Varied Embedding and Machine Learning Classifiers for Fine Grained Offensive Content Identification," *IEEE International Conference for Women in Innovation, Technology & Entrepreneurship (ICWITE)*, IEEE, Bangalore, India, pp. 1–6. https://doi.org/10.1109/ICWITE57052.2022.10176223

[4] Chow D. V., Natania F., Nathanael O. T., Setiawan K. E., and Hasani M. F. (2023) "Cyberbullying Detection: An Investigation into NLP and Machine Learning Techniques," *5th International Conference on Cybernetics and Intelligent System (ICORIS)*, IEEE, Pangkalpinang, Indonesia, pp. 1–6. https://doi.org/10.1109/ICORIS60118.2023.10352294

[5] Balaji P. G., Katariya P. P., Sruthi S., and Venugopalan M. (2024) "Cyberbullying Detection on Multiclass Data Using Machine Learning and A Hybrid CNN-BiLSTM Architecture," *ICKECS*, IEEE, Chikkaballapur, India, pp. 1–6. https://doi.org/10.1109/ICKECS61492.2024.10616957

[6] Bokolo B. G. and Liu Q. (2023) "Cyberbullying Detection on Social Media Using Machine Learning," *IEEE INFOCOM WKSHPS*, IEEE, Hoboken, NJ, USA, pp. 1–6. https://doi.org/10.1109/INFOCOMWKSHPS57453.2023.10226114

[7] Singh N. K., Singh P., and Chand S. (2022) "Deep Learning Based Methods for Cyberbullying Detection on Social Media," *ICCCIS*, IEEE, Greater Noida, India, pp. 521–525. https://doi.org/10.1109/ICCCIS56430.2022.10037729

[8] Joseph V. A., Prathap B. R., and Kumar K. P. (2024) "Detecting Cyberbullying in Twitter: A Multi-Model Approach," *ICDECS*, IEEE, Bangalore, India, pp. 1–6. https://doi.org/10.1109/ICDECS59733.2023.10502699

[9] Tapaopong W., Charoenphon A., Raksasri J., and Samanchuen T. (2024) "Enhancing Cyberbullying Detection on Social Media Using Transformer Models," *TIMES-iCON*, IEEE, Bangkok, Thailand, pp. 1–5. https://doi.org/10.1109/TIMES-iCON61890.2024.10630719

[10] Wiranto, Harjito B (2023) "Enhancing machine learning performance in cyberbullying detection through

hyperparameter optimization," *Proceedings of the IEEE International Conference on Technology, Engineering, and Computing Applications (ICTECA)*. https://doi.org/10.1109/ICTECA60133.2023.10490843

[11] Lakshminadh K., Sravanthi V., Koushik K., and Bhaskar C. S. (2023) "Enhancing Profanity Detection in Textual Data Using BiLSTM," *ICSSAS*, IEEE, Erode, India, pp. 1–6. https://doi.org/10.1109/ICSSAS57918.2023.10331695

[12] Ea P., Xiang J., Salem O., and Mehaoua A. (2023) "Evaluating Cyberbullying Detection Algorithm Performance in Text and Image Analysis," *MLCR*, IEEE, Nanjing, China, pp. 30–35. https://doi.org/10.1109/MLCR61158.2023.00015

[13] Gongane V. U., Munot M. V., and Anuse A. (2023) "Explainable AI for Reliable Detection of Cyberbullying," *PuneCon*, IEEE, Pune, India, pp. 1–6. https://doi.org/10.1109/PuneCon58714.2023.10450132

[14] Sarkale D. G., Gabani V. J., Zhang W., and Akilan T. (2023) "NLP-driven Content Classification Towards Fake News and Bully Detection," *AIBThings*, IEEE, Mount Pleasant, MI, USA, pp. 1–5. https://doi.org/10.1109/AIBThings58340.2023.10292460

[15] Jamjoom A. A., Karamti H., Umer M., Alsubai S., Kim T.-H., and Ashraf I. (2024) "RoBERTaNET: Enhanced RoBERTa Transformer Based Model for Cyberbullying Detection With GloVe Features," *IEEE Access*, vol. 12, pp. 58950–58959. https://doi.org/10.1109/ACCESS.2024.3386637

[16] Obaid M. H., Guirguis S. K., and Elkaffas S. M. (2023) "Cyberbullying Detection and Severity Determination Model," *IEEE Access*, vol. 11, pp. 97391–97399. https://doi.org/10.1109/ACCESS.2023.3313113

[17] Al-Hashedi M., Soon L.-K., Goh H.-N., Lim A. H. L., and Siew E.-G. (2023) "Cyberbullying Detection Based on Emotion," *IEEE Access*, vol. 11, pp. 53907–53918. https://doi.org/10.1109/ACCESS.2023.3280556

[18] Teng T. H. and Varathan K. D. (2023) "Cyberbullying Detection in Social Networks: ML vs Transfer Learning, *IEEE Access*," vol. 11, pp. 55533–55560. https://doi.org/10.1109/ACCESS.2023.3275130

[19] Elsafoury F., Katsigiannis S., Pervez Z., and Ramzan N. (2021) "When the Timeline Meets the Pipeline: A Survey on Automated Cyberbullying Detection," *IEEE Access*, vol. 9, pp. 103541–103563. https://doi.org/10.1109/ACCESS.2021.3098979

[20] Mansur Z., Omar N., and Tiun S. (2023) "Twitter Hate Speech Detection: A Systematic Review," *IEEE Access*, vol. 11, pp. 16226–16249. https://doi.org/10.1109/ACCESS.2023.3239375

[21] Wang J., Fu K., and Lu C.-T. (2020) "SOSNet: A Graph Convolutional Network Approach to Fine-Grained Cyberbullying Detection," *IEEE Big Data*, IEEE, Atlanta, GA, USA, pp. 1699–1708. https://doi.org/10.1109/BigData50022.2020.9378065

[22] Philipo A G, et al. (2025) "Cyberbullying detection: exploring datasets, technologies and challenges," *ACM Computing Surveys*. https://doi.org/10.1145/3785654

[23] Wang J., Fu K., and Lu C.-T. (2020) "Fine-Grained Balanced Cyberbullying Dataset," *IEEE Dataport*, November 13. https://doi.org/10.21227/kn1c-zx22

[24] Farouk Z, Kamel B S, Mohamed B (2013) "Adaptive backstepping control for a class of uncertain single input single output nonlinear systems," *Proceedings of the 10th International Multi-Conference on Systems, Signals & Devices (SSD)*, Hammamet, Tunisia, March 18–21, 2013. https://doi.org/10.1109/SSD.2013.6564134

[25] Boulkroune A, Hamel S, Zouari F, Boukabou A, Ibeas A (2017) "Output-feedback controller based projective lag-synchronization of uncertain chaotic systems in the presence of input nonlinearities," *Mathematical Problems in Engineering*, Vol. 2017, Article ID 8045803. https://doi.org/10.1155/2017/8045803

[26] Rigatos G, Abbaszadeh M, Sari B, Siano P, Cuccurullo G, Zouari F (2023) "Nonlinear optimal control for a gas compressor driven by an induction motor," *Results in Control and Optimization*, Vol. 11, Article 100226. https://doi.org/10.1016/j.rico.2023.100226

[27] Charabi I, Abidine M B, Fergani B, Oussalah M (2025) "DeepF-SVM: a new hybrid deep learning model for enhanced sensor-based human activity recognition," *Cluster Computing*, Vol. 28, Article 910. https://doi.org/10.1007/s10586-025-05636-y

[28] Nuanmeesri S (2021) "A hybrid deep learning and optimized machine learning approach for rose leaf disease classification," *Engineering, Technology & Applied Science Research*, vol. 11, no. 5, pp. 7678–7683. https://doi.org/10.48084/etasr.4455

[29] Goyal A, Bengio Y (2022) "Inductive biases for deep learning of higher-level cognition," *Proceedings of*

the Royal Society A, Vol. 478, Article 20210068. https://doi.org/10.1098/rspa.2021.0068

[30] Kishore S, He H (2024) "Unveiling divergent inductive biases of LLMs on temporal data," *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pp. 220–228, Mexico City, Mexico. https://doi.org/10.18653/v1/2024.naacl-short.20