Sentiment Analysis of Social Network Data Using Traditional and Hybrid Deep Learning Models

Abdimukhan Tolep
Khoja Akhmet Yassawi International Kazakh-Turkish University, Kazakhstan
tolepabdimukhan@gmail.com (corresponding author)

Satmyrza Mamikov
University of Friendship of People’s Academician A. Kuatbekov, Kazakhstan
aseltas@tuta.io

Zhansaya Yakhiya
University of Friendship of People’s Academician A. Kuatbekov, Kazakhstan
Zhansaya.yakhiya@mail.ru

Ainur Abdullayeva
International University of Tourism and Hospitality, Kazakhstan
abdullaeva.a@iuth.edu.kz

Received: 21 June 2025 | Revised: 21 July 2025 and 16 August 2025 | Accepted: 30 August 2025

DOI: https://doi.org/10.48084/etasr.12865

ABSTRACT

Social networking platforms generate vast volumes of short, informal user-generated text that encode opinions and attitudes. Organizations seek to analyze this content to derive actionable insights, motivating robust sentiment-analysis methods. This study presents an empirical comparison of deep-learning approaches for sentiment classification of social-media text. We evaluate six standalone models—Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Recursive Neural Network (RecNN), Long Short-Term Memory (LSTM), Bidirectional LSTM (BiLSTM), and Gated Recurrent Unit (GRU)—and three hybrid variants (CNN–RNN, CNN–LSTM, CNN–BiLSTM). The corpus undergoes standard preprocessing (cleaning, tokenization, lemmatization, stop-word removal) before classification. Across the tested settings, the CNN–BiLSTM consistently achieves the strongest overall performance on key metrics (e.g., recall, F1-score, AUC-ROC), outperforming both standalone architectures and the other hybrids. These findings indicate that combining localized n-gram feature extraction with bidirectional sequence modeling yields a favorable balance of accuracy and robustness for short, noisy posts, and they offer practical guidance for selecting architectures for social-media sentiment analysis.

Keywords-sentiment analysis; deep learning; hybrid models; classification; NLP; social media

I. INTRODUCTION

Sentiment analysis is an algorithmic approach used to analyze people's opinions, feelings, emotional responses, assessments, and preferences towards various entities, such as products, services, institutions, individuals, concerns, occurrences, subjects, and their attributes. The field of sentiment analysis emerged concurrently with the rise of social networking sites, which provide a vast volume of unorganized data in the form of feedback, discussion forums, blog posts, multimedia messaging, and other forms of online communication. Text mining is a relatively new subfield of data analysis that involves the process of extracting valuable knowledge from unstructured data. It has become an important instrument for extracting significant information from an enormous amount of data, given that today, approximately 2.5 exabytes of data are generated every single day [1]. Text categorization has emerged as a crucial research area, especially with the rise and rapid growth of online social networking sites and similar platforms. On these kinds of websites, people exchange their knowledge and opinions, which are then evaluated to determine what people like and dislike about various products. The process of extracting the audience's responses, evaluations, ideas, and comments for the products or services offered by a business is known as "sentiment analysis," and it has been implemented to accomplish this [2]. The retrieved sentiments are used to formulate and refine policies to increase sales and improve consumer services. The widespread use of social networking sites has resulted in a wealth of instructive data that can be used to extract meaningful information. The flexibility and depth of social networking libraries have contributed to a rise in the prevalence of these resources. A substantial body of work in Natural Language Processing (NLP) addresses emotion and sentiment identification [2, 3], and sentiment analysis on social networks has attracted significant interest [4, 5]. Extracting reliable information from the massive volume of user-generated text requires advanced computational methods. Deep learning (DL) underpins many modern sentiment-analysis systems [6-9] by learning task-specific representations that reduce manual feature engineering.

Despite these advances, systematic evaluations of hybrid architectures across heterogeneous social-media corpora remain limited. In particular, the comparative behavior of CNN–RNN, CNN–LSTM, and CNN–BiLSTM versus standalone models (CNN, RNN, RecNN, LSTM, BiLSTM, GRU) under varying styles, topics, and class distributions is not well established. To address this gap, we benchmark nine deep approaches—six standalone and three hybrids—on three public sentiment datasets (STS-Test, SS-Tweet, SemEval-2013) under identical preprocessing and evaluation protocols. Our contributions are: (1) a comprehensive empirical comparison of standalone and hybrid deep models; (2) an assessment of cross-dataset consistency and generalizability; and (3) practical guidance on trade-offs among model complexity, accuracy, and computational cost for deployable sentiment-analysis systems.

II. PROBLEM STATEMENT

Sentiment analysis of user-generated social-media content poses challenges that extend beyond conventional text classification. In contrast to formal prose, social-media posts are typically brief, weakly structured, and replete with slang, abbreviations, and emotive symbols, thereby complicating the reliable extraction of sentiment cues. Moreover, prior studies have predominantly emphasized lexicon-based approaches or isolated DL architectures without systematically articulating their probabilistic foundations in terms of conditional-probability formulations. To address these limitations, this study frames sentiment analysis as a conditional-likelihood estimation problem defined over sequences of social-media posts.

Let a sequence of interactions on a social network session be denoted as $S=\left\{x_1,x_2,...,x_{T|}\right\}$, where each message x_t is timestamped and associated with a sentiment label $y_t \in \{0,1,2\}$ (negative, neutral, positive). Formally, we seek parameters θ of a discriminative model for the conditional distribution of the sentiment label given the observed history:

$$P\left(y_t \mid x_{\le t}, S;\theta\right) \tag{1}$$

$$P\left(y_t \mid x_{\le t}, S;\theta\right) \tag{2}$$

where θ denotes the trainable parameters of the CNN-BiLSTM model, and $x_{\le t}$ represents the sequence history up to time *t*. The convolutional layers extract local n-gram features, while the BiLSTM encodes sequential dependencies across the session, thus ensuring that the softmax output layer incorporates both local contextual representations and historical session knowledge (S) into the likelihood estimation.

The model encodes the input sequence into hidden states $h_t = f_{BiLSTM}\left(f_{CNN}\left(x_t, h_{t-1}\right)\right)$. The softmax output defines the class probability distribution:

$$P(y_t = c|x_{\le t}, S;\theta) = \frac{exp(W_c h_t + b_c)}{\sum_{c'} exp(W_{c'} h_t + b_{c'})} \tag{3}$$

where $W_{c'}$ and b_c are class-specific parameters. Training optimizes the negative log-likelihood (cross-entropy loss):

$$L(\theta) = -\sum_{t=1}^{T} \log P\left(y_t \mid x_{\le t}, S;\theta\right) \tag{4}$$

This formulation clarifies that the CNN–BiLSTM is not merely an ensemble of architectures but a unified probabilistic model in which the predictions are conditioned jointly on message-level features and session-level history. By casting the task within a likelihood-based framework, the approach makes explicit the connection between the DL outputs and statistical inference, thereby strengthening the theoretical rigor and interpretability of context-aware sentiment classification.

A. Dataset

This study uses three well-known sentiment analysis datasets. The Stanford Twitter Sentiment Test Set (STS-Test) [10] contains 498 tweets in three classes: positive 182 (36.5%), negative 177 (35.5%), and neutral 139 (27.9%). The Sentiment Strength Twitter (SS-Tweet) dataset [11] includes 4,242 tweets: positive 1,252 (29.5%), negative 1,037 (24.5%), and neutral 1,953 (46.0%). The SemEval-2013 dataset [12] comprises 13,975 tweets: positive 5,349 (38.3%), negative 2,186 (15.6%), and neutral 6,440 (46.1%). Each dataset is treated as a three-class classification problem.

B. Deep Learning in Sentiment Analysis

The proposed model consists of the preprocessing stage, the feature extraction stage, the categorization stage, and the evaluation stage. In this portion, focus is placed on conducting a more in-depth analysis of each stage. DL employs multilayer neural architectures that learn hierarchical representations in their hidden layers. Unlike traditional ML pipelines, which rely on hand-crafted features and feature selection [13], DL learns task-specific representations directly from data and couples representation learning with classification. For short, noisy social-media text, this joint optimization often yields improved robustness and accuracy, provided sufficient data and regularization. Model hyperparameters (e.g., learning rate, batch size, dropout) are tuned via automated search procedures such as grid or random search with early stopping under a shared budget [14]. Consequently, ML and DL follow different routes to sentiment-polarity classification: the former separates feature engineering from the classifier, whereas the latter learns them jointly. Figure 1 contrasts these pipelines.

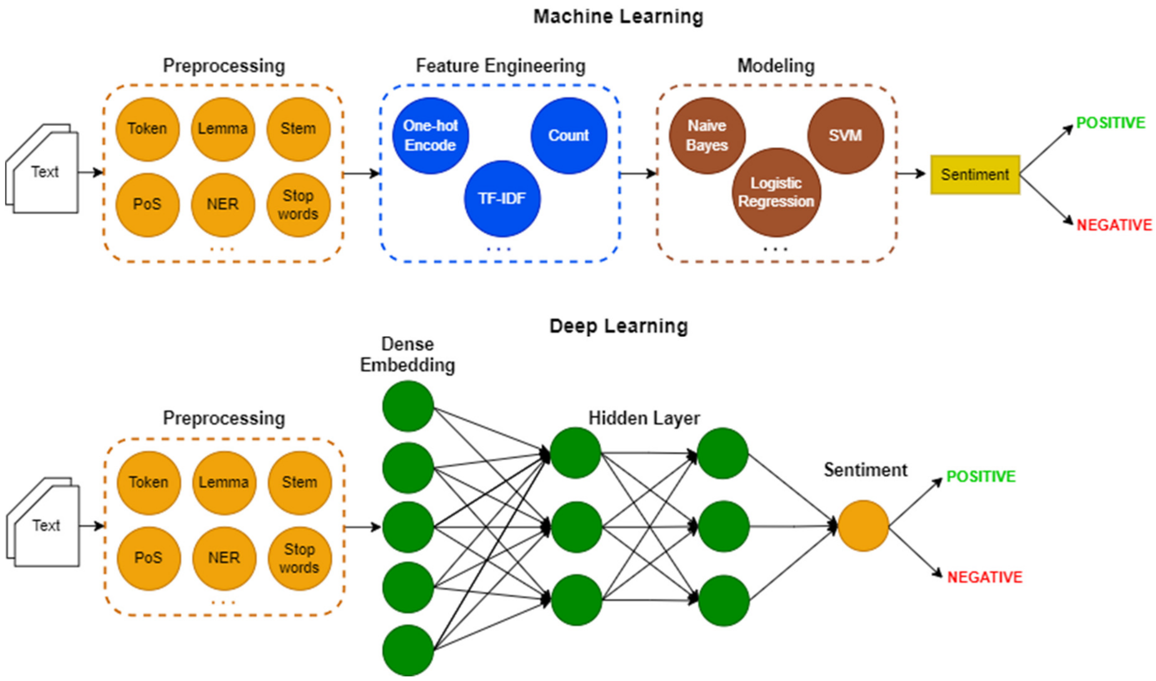

Fig. 1. Machine learning and deep models in sentiment analysis.

An early unified neural architecture for a wide range of NLP tasks was proposed in [15]. The architecture, referred to as CNN, takes the combination of word vectors as its input data. It also includes convolutional and max-pooling layers, preceding the general neural network structure.

Convolution Layer: This layer is a type of generalization of the window technique. In this layer, a window of a defined dimension is shifted over the sentence, and the weight matrix is the same for each sequence [16]. Convoluting over each sequence results in the generation of a single feature vector. This layer's job is to parse out local characteristics based on the order of the syllables in the statement. The network is capable of having a variety of window widths and a variety of weight matrices, with each of these combinations constituting a single channel. The duration of the output of the convolution layer is determined by the length of the input statement as well as the number of channels that were utilized. The Max-Pooling Layer: The max-pooling layer is used to determine the highest value for each feature across all windows. This is done to ensure that the height of the sentence vector is consistent throughout. This method is preferred over simply aggregating the results because not all words contribute equally to categorization. The max-pooling layer captures the proportional significance of each word's contribution. Figure 2 depicts a mockup of the CNN network's internal infrastructure. RNNs have feedback loops because the connections across the neurons in an RNN [17, 18] create a unidirectional circle, which results in the development of feedback loops within the RNN itself. The primary function of an RNN is to process consecutive inputs based on the internal memory accumulated by directed cycles, utilizing the collected information from these cycles. RNNs, as opposed to typical neural networks, can retain the results of prior computations of information and repurpose these results by applying them to the following item in the input sequence.

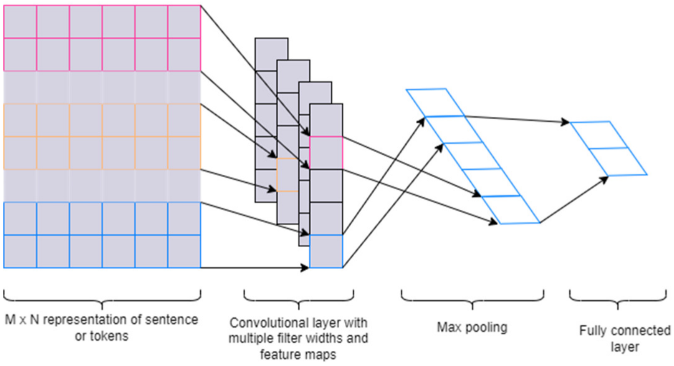

Fig. 2. CNN model architecture.

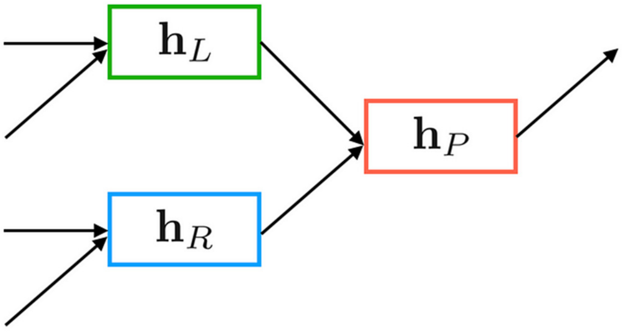

Fig. 3. Tree configuration of a RecNN.

The architecture of a RecNN is typified by a tree configuration characterized by a specified number of branches. Using the binary tree as an exemplar, one can deduce the hidden state vector of a prevailing node from the vectors associated with its adjacent left and right child nodes, a phenomenon graphically represented in Figure 3. For precision in scholarly communication, differentiating between these two distinct networks is essential.

$$h_P = a\left(W\begin{bmatrix} h_L \\ h_R \end{bmatrix} + b\right) \quad (5)$$

Equation (5) delineates the procedure computed progressively from the leaf nodes to the root node. Given its architecture, the RecNN is postulated to articulate the correlations between the distant elements more effectively than the RNN. This is attributed to its depth, approximated by log2(T) when there are T elements.

LSTM is a specialized kind of RNN that is capable of using long memory as the input of activation functions in the hidden layer [19]. An illustration of the LSTM architecture is provided in Figure 4. Each cell contains its own distinct collection of data, and the lines with arrows indicate the movement of the latter from one collection of data to another. The data that are entered are preprocessed so that they may be reshaped for the embedding matrix. The LSTM layer directly follows the embedding stage and comprises 200 hidden units with tanh activations and a recurrent dropout of 0.2 to mitigate overfitting. Its output is fed into a fully connected (dense) layer of 128 units, each employing a ReLU activation; this layer also incorporates L2 regularization ($\lambda = 0.01$) and a dropout rate of 0.5. The final classification layer applies a sigmoid activation to produce binary probabilities for positive versus negative sentiment. The models were trained using the Adam optimizer (learning rate = 1×10^{-3}, $\beta_1 = 0.9$, $\beta_2 = 0.999$), a batch size of 64, and up to 25 epochs with early stopping (patience = 3) based on the validation F1-score. The word inputs were mapped to 300-dimensional pre-trained GloVe embeddings, which remained fixed during training.

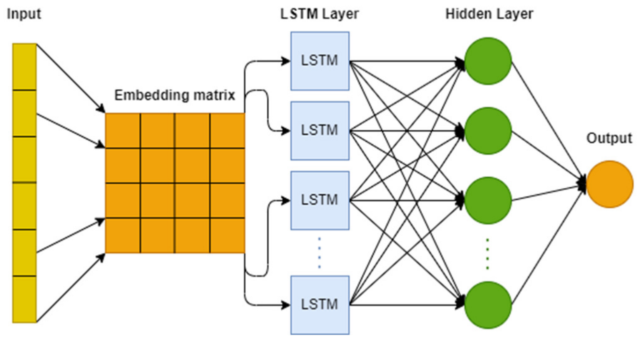

Fig. 4. LSTM network for sentiment analysis.

The input vector at time step t is denoted by xt, and the hidden state at time step t is denoted by ht. The hidden state at time step t is computed by taking into account both the input at the most recent time step and the preceding hidden state.

$$h_t = f\left(w^{hh} h_{t-1} + w^{hx} x_t\right) \quad (6)$$

In most cases, the tanh function or the ReLU function will be used in place of the activation function f in Equation (5). The input x_t is conditioned using a weight matrix, which is denoted by w^{hx}. The weight matrix denoted by w^{hh} was applied to condition the preceding concealed state, h_{t-1}.

The variable y_t denotes the outcome probability distribution in the vocabulary at stage *t*. To forecast the next word in a statement, for instance, a vector of possibilities that covers the entire word repertoire needs to be created.

$$y_t = soft\ \max\left(w^{yh} h_t\right) \quad (7)$$

The hidden state, also known as h_t, is considered to be the network's memory. It records information about everything that took place in all of the preceding time steps. y_t is computed entirely based on the memory that was present at time *t* and the weight matrix that corresponds to it, W^{yh}.

In contrast to a feedforward neural network, which employs unique parameters at each layer, an RNN makes use of the same parameters (W^{hx}, W^{hh}, W^{yh}) throughout all of its stages [20]. This means that it continues to carry out the same work at each step, albeit with different inputs. Because of this, the overall number of characteristics that need to be learned is drastically reduced.

BiLSTM, LSTM encodes a sequence left-to-right. BiLSTM runs two LSTM chains in parallel—one left-to-right and one right-to-left—with independent parameters; at each step, their hidden states are concatenated into a single vector that represents the token with context from both directions [20]. The gate set (input, forget, output) is unchanged; parameters and compute are approximately twice those of a single LSTM. This bidirectional context typically improves sentence-level sentiment when cues depend on both left and right neighbors, at the cost of higher training and inference time. Figure 5 illustrates the BiLSTM architecture.

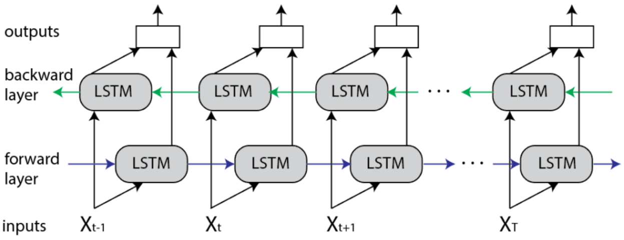

Fig. 5. Flowchart of the BiLSTM.

To address the short-term memory issues similar to those of LSTM, GRU, the latest variation of RNN, was implemented. GRU does not have a cell state and relies on a hidden state, instead, to convey information [21]. It is made up of two gates, a reset gate (r^t) and an update gate (z^t), both of which are illustrated by:

$$z^t = \sigma\left(w_{z^t} x^t + U_{z^t} h_{t-1} + b_{z^t}\right) \qquad (8)$$

$$r^t = \sigma\left(w_{r^t} x^t + U_{r^t} h_{t-1} + b_{r^t}\right) \qquad (9)$$

where z^t is the update gate and rt the reset gate. $\sigma(\cdot)$ The logistic sigmoid w and U are learnable weight matrices (input to hidden and hidden to hidden), and b is a bias vector; the input at time t is $x^{t,}$ the candidate hidden state is h^t, the secret/output state is h^t, and element-wise multiplication is denoted. Any constant is included in the bias, so no separate c is used.

The update gate controls the interpolation between the previous hidden state and the candidate state, effectively combining the roles of the LSTM's input and forget gates.[22]. With fewer gates than LSTM, GRU training can be completed faster because the reset gate controls the amount of previous data that is forgotten [23]. Figure 6 provides an architecture of a gated recurrent network.

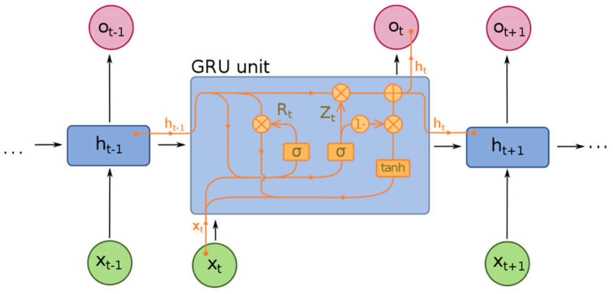

Fig. 6. Flowchart of the gated recurrent network.

III. RESULTS

This section demonstrates evaluation parameters to assess the proposed methods and compare them. The experimental results are visualized using confusion matrices, ROC curves (with AUC-ROC) to assess threshold-free ranking behavior. Aggregate results are summarized in TABLES I–III, enabling like-for-like comparison under a standard evaluation protocol.

A. Evaluation Metrics

This study evaluates the sentiment classification problem using various evaluation parameters, including Accuracy, Precision, Recall, F, and AUC-ROC Curve. The formulas for each evaluation parameter are [24-26]:

$$accuracy = \frac{TP + TN}{P + N} \qquad (10)$$

$$preision = \frac{TP}{TP + FP} \qquad (11)$$

$$recall = \frac{TP}{TP + FN} \qquad (12)$$

$$F1 = \frac{2 \times precision \times recall}{precision + recall} \qquad (13)$$

B. Experiment Results

Figure 7 provides an analytical comparison between six conventional DL models and three CNN-integrated hybrid models, all tailored for sentiment classification. Delving into the resultant data, there is a discernible edge with which the LSTM and GRU models operate, marking a distinguishable superiority in their performance in comparison to other traditional DL strategies.

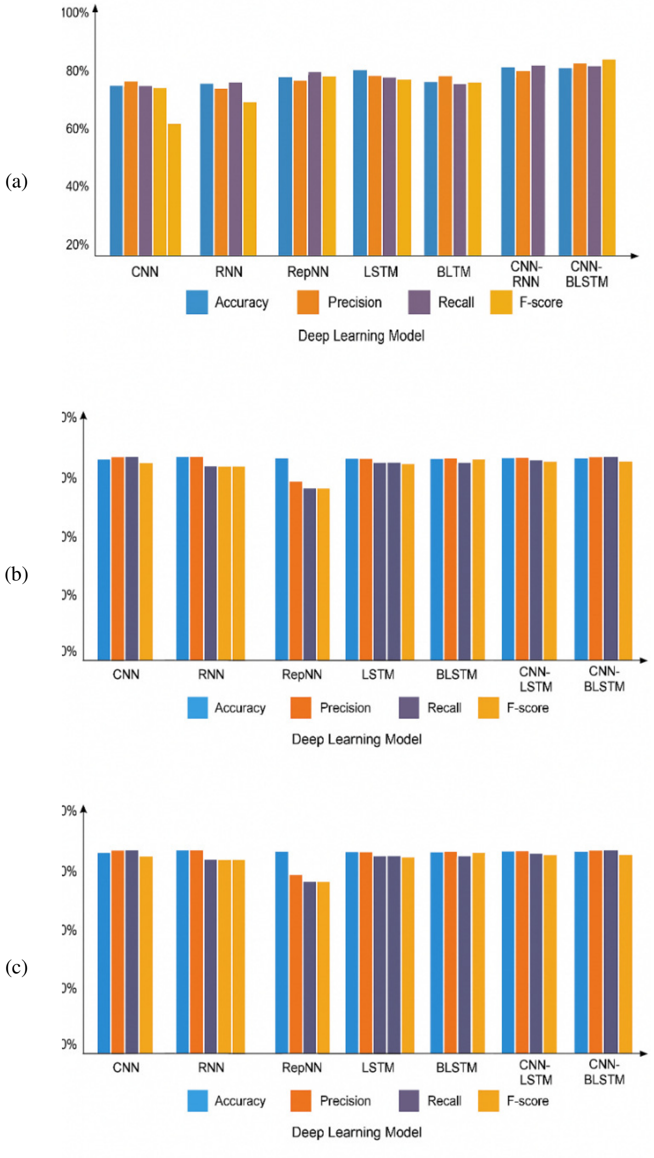

Fig. 7. Results of DL methods in sentiment analysis. (a) Results on STS-Test, (b) results on sentiment strength Twitter dataset (SS-Tweet), and (c) results on SemEval-2013 dataset.

However, among the evaluated models, CNN–BiLSTM consistently attains the best performance across STS-Test, SS-Tweet, and SemEval-2013, yielding the highest macro-F1 and AUC-ROC. The hybrid leverages complementary strengths, localized n-gram extraction (CNN) and bidirectional context modeling (BiLSTM) and outperforms both standalone and alternative hybrid baselines under the standard evaluation protocol.

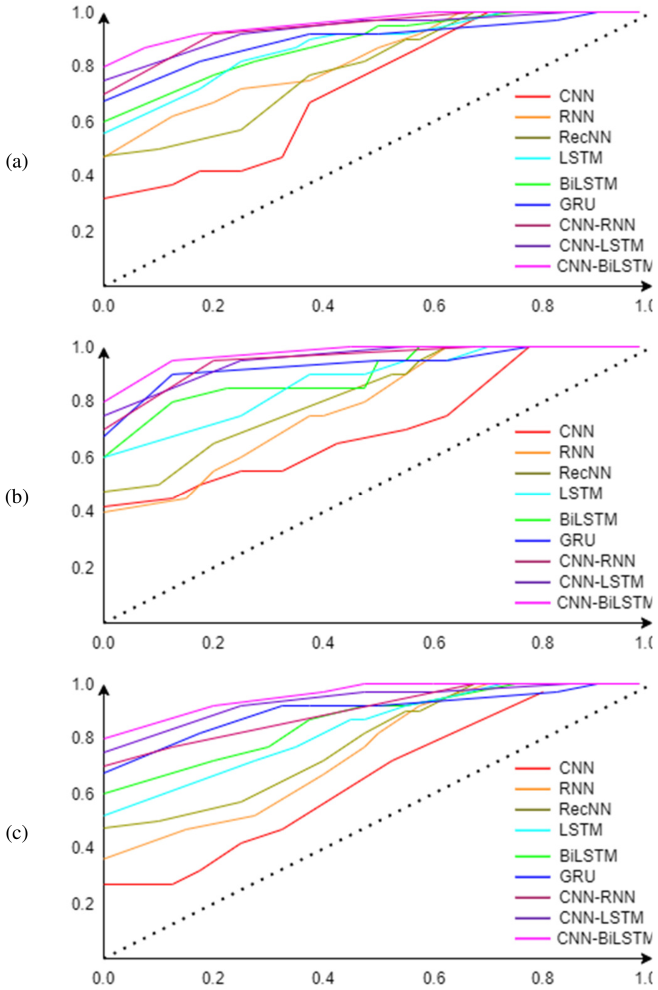

Fig. 8. The ROC curve results for the DL models in sentiment analysis. (a) The ROC curve results for STS-Test, (b) the ROC curve results for SS-Tweet, and (c) the ROC curve results for SemEval-2013 dataset.

Figure 8 elucidates the Receiver Operating Characteristic (ROC) Curves corresponding to various DL and hybrid methodologies employed in sentiment analysis, spanning three distinct datasets. Upon the examination of the results, the CNN-LSTM and CNN-BiLSTM networks manifest superior classification outcomes in addressing the sentiment analysis conundrum. In contrast, the CNN and RNN techniques register comparatively suboptimal results. Given these observations, it can be inferred that the CNN-BiLSTM network, strengthened by its evaluation metrics, including the ROC curve, stands as a viable tool for deployment in sentiment analysis.

TABLE I. SENTIMENT CLASSIFICATION RESULTS ON STS-TEST DATASET

Model type	Method	Accuracy	Precision	Recall	F1-score	AUC-ROC
Deep models	CNN	71.41	70.67	61.46	40.63	58.92
	RNN	71.87	68.37	68.51	50.17	62.19
	RecNN	71.34	68.87	71.14	60.54	65.11
	LSTM	73.31	71.57	73.18	71.12	76.27
	BiLSTM	75.28	73.56	73.45	73.27	79.74
	GRU	73.18	70.86	71.12	70.67	77.38
Hybrid models	CNN-RNN	79.61	77.82	77.64	77.38	79.63
	CNN-LSTM	79.92	77.93	77.68	77.18	80.68
	CNN-BiLSTM	88.68	85.37	88.46	82.27	87.37

Table I demonstrates the sentiment classification results for the STS-Test Dataset, where six classical DL models and three hybrid models were applied for the sentiment classification problem. GRU and BiLSTM are the two methods that show higher accuracy than the other algorithms. The results show that hybrid models perform better than the classical deep models, and the CNN-BiLSTM network gives the highest performance.

TABLE II. SENTIMENT CLASSIFICATION RESULTS ON SS-TWEET DATASET

Model type	Method	Accuracy	Precision	Recall	F1-score	AUC-ROC
Deep models	CNN	89.12	83.23	72.82	68.58	72.81
	RNN	88.24	89.63	90.96	88.07	82.63
	RecNN	91.24	89.38	83.63	83.54	86.85
	LSTM	90.36	80.48	79.67	79.65	87.38
	BiLSTM	91.48	90.19	90.13	90.07	90.34
	GRU	90.72	90.67	90.69	89.82	91.61
Hybrid models	CNN-RNN	90.64	89.71	89.28	89.12	91.45
	CNN-LSTM	91.17	90.78	90.49	90.13	92.08
	CNN-BiLSTM	98.76	98.71	98.52	96.47	97.71

Table II presents the sentiment classification results for the SS-Tweet. Due to the larger number of training samples in this dataset, the models were trained longer and achieved higher efficiency compared to the STS-Test Dataset. On average, the evaluation parameter performance on STS-T was higher than that of the six models trained on STS-Test.

TABLE III. SENTIMENT CLASSIFICATION RESULTS ON SEMEVAL-2013 DATASET

Model type	Method	Accuracy	Precision	Recall	F1-score	AUC-ROC
Deep models	CNN	79.34	77.79	75.51	73.58	75.81
	RNN	80.96	78.45	79.42	77.24	78.28
	RecNN	80.28	77.93	77.50	77.60	77.02
	LSTM	82.28	81.64	80.71	80.93	82.54
	BiLSTM	80.73	78.24	78.18	78.12	81.46
	GRU	80.39	82.75	82.78	79.05	82.46
Hybrid models	CNN-RNN	80.87	78.24	77.83	76.68	83.48
	CNN-LSTM	81.92	80.25	78.87	77.87	83.46
	CNN-BiLSTM	96.17	95.07	94.84	92.08	96.59

Table III demonstrates the sentiment classification results for the SemEval-2013 Dataset, which contains the highest number of training samples among the other applied datasets. The training times of the models on this dataset were the longest. Nevertheless, the performances of the models were the highest when they were trained on the SemEval-2013 dataset. As observed, the performance of the CNN-BiLSTM network on the sentiment classification problem was the highest.

Considering the performance rates it achieved, the proposed strategy may be accepted as a possible method for identifying instances of sentiment analysis within social networking sites. Furthermore, considering all assessment criteria, the deep neural network demonstrated the highest performance in sentiment analysis on social networks. The effectiveness of the proposed method can be attributed to the utilization of the deep neural network presented for weight and bias adjusting, as well as a reduction in the required training time. The findings indicate that the proposed strategy using deep neural networks can be easily adapted to accommodate both brief and lengthy texts.

IV. DISCUSSION

The findings of this study indicate that the proposed hybrid LSTM–CNN architecture enables cyberbullying detection by combining the temporal modeling capacity of the recurrent networks with the localized feature extraction of convolutional layers. Empirically, the hybrid model consistently outperforms traditional classifiers, support vector machine, random forest, and logistic regression, as well as standalone DL approaches across key metrics, including recall, F1-score, and AUC-ROC. These gains underscore the primary contribution of this work: a unified framework that learns both sequential dependencies and n-gram patterns directly from raw text, reducing the reliance on manual feature engineering and improving the sensitivity to subtle, context-dependent abusive language.

Beyond establishing superior classification performance, the study provides a comprehensive comparative analysis that benchmarks five traditional algorithms alongside three deep architectures using a balanced, expert-annotated Reddit dataset. This empirical foundation addresses a gap in the literature, where prior research has often emphasized single-model evaluations or has relied on small-scale corpora. By assessing all methods under identical preprocessing and evaluation protocols, the analysis offers practical guidance on the trade-offs among the model complexity, computational cost, and detection accuracy, which is essential for practitioners aiming to deploy real-time moderation systems.

Notwithstanding these strengths, there are several limitations. First, the experiments were conducted on a single English-language Reddit corpus, which may not capture the full range of linguistic nuances, platform conventions, and cultural contexts present across social networks. Second, the hybrid model's performance improvements are accompanied by increased training time and resource consumption, which can constrain its applicability in settings with limited computational capacity. Third, the present approach treats each post in isolation, omitting thread-level context and user-history signals that could reduce the false positives and improve disambiguation.

Future work should address these limitations by validating the hybrid framework on multilingual and cross-platform datasets to assess the generalizability in more heterogeneous environments. To improve efficiency, model-compression strategies (e.g., quantization, pruning, knowledge distillation) and lightweight architectures could be explored to reduce the inference latency and hardware requirements. Incorporating attention mechanisms or transformer-based embeddings may further enhance the model's ability to focus on the most salient spans of a message. Finally, extending the input scope to include thread-level context and user-interaction graphs could better capture coordinated harassment patterns. Pursuing these directions would build upon the present contributions toward more robust, efficient, and context-aware systems for automated cyberbullying moderation.

V. CONCLUSION

The term "social media" refers to a group of online platforms that have only become widely used in the past several years. Machine learning is used in a wide range of contexts, including the analysis of social networks. The purpose of this evaluation is to provide a comprehensive overview of the various applications that use DL models and their hybrid solutions to evaluate social networking sites for sentiment analysis problems. This article investigates each stage of the process for sentiment analysis, including gathering data, data cleaning, feature engineering, the application of DL models, and text categorization. Various solutions have been proposed to address issues, such as generic metadata architecture, benchmark settings, and fragmentation in the process of sentiment analysis on social network data streams. The analysis also proposes a general metadata infrastructure for sentiment analysis on social media to address the challenges associated with categorizing positive and negative texts in the data collected from social networks. Compared to proportional approaches, the proposed architecture's performance was superior across all assessment parameters for the given problem.

In addition, a more robust automatic sentiment classification system can be created by taking into account issues, such as class disparity data, binary and multi-classification, scalability, multilingualism, benchmark setting, and disintegration. In this way, the system will be able to function for an extended period of time.

ACKNOWLEDGEMENT

This work was supported by the research project "Automatic detection of cyberbullying among young people in social networks using artificial intelligence" funded by the Ministry of Science and Higher Education of the Republic of Kazakhstan. Grant No. IRN AP23488900.

REFERENCES

[1] J. Tariq, H. Rahman, M. A. Masood, A. F. Subahi, O. I. Khalaf, and Y. Alotaibi, "Multi-Tier Sentiment Analysis of Social Media Text Using Supervised Machine Learning, "*Computers, Materials and Continua*,

vol. 74, no. 3, pp. 5527–5543, Dec. 2022 https://doi.org/10.32604/cmc.2023.033190.

[2] S. Peng et al., "A survey on deep learning for textual emotion analysis in social networks," *Digital Communications and Networks*, vol. 8, no. 5, pp. 745–762, Oct. 2022, https://doi.org/10.1016/j.dcan.2021.10.003.

[3] P. K. Jain, W. Quamer, V. Saravanan, and R. Pamula, "Employing BERT-DCNN with sentic knowledge base for social media sentiment analysis," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 8, pp. 10417–10429, Aug. 2023, https://doi.org/10.1007/s12652-022-03698-z.

[4] T. Kanan, A. Mughaid, R. Al-Shalabi, M. Al-Ayyoub, M. Elbes, and O. Sadaqa, "Business intelligence using deep learning techniques for social media contents, "*Cluster Computing*, vol. 26, no. 2, pp. 1285–1296, Apr. 2023, https://doi.org/10.1007/s10586-022-03626-y.

[5] D. Ariyus, D. Manongga, and I. Sembiring, "Enhancing Sentiment Analysis of Indonesian Tourism Video Content Commentary on TikTok: A FastText and Bi-LSTM Approach," *Engineering, Technology & Applied Science Research*, vol. 14, no. 6, pp. 18020–18028, Dec. 2024, https://doi.org/10.48084/etasr.8859.

[6] D. Sultan et al., "A Review of Machine Learning Techniques in Cyberbullying Detection, "*Computers, Materials & Continua*, vol. 74, no. 3, pp. 5625–5640, 2023, https://doi.org/10.32604/cmc.2023.033682.

[7] S. Bengesi, T. Oladunni, R. Olusegun, and H. Audu, "A Machine Learning-Sentiment Analysis on Monkeypox Outbreak: An Extensive Dataset to Show the Polarity of Public Opinion From Twitter Tweets," *IEEE* Access, vol. 11, pp. 11811–11826, 2023, https://doi.org/10.1109/ACCESS.2023.3242290.

[8] V. Yadav, P. Verma, and V. Katiyar, "Long short term memory (LSTM) model for sentiment analysis in social data for e-commerce products reviews in Hindi languages," *International Journal of Information Technology*, vol. 15, no. 2, pp. 759–772, Feb. 2023, https://doi.org/10.1007/s41870-022-01010-y.

[9] B. Al sari et al., "Sentiment analysis for cruises in Saudi Arabia on social media platforms using machine learning algorithms, "*Journal of Big Data*, vol. 9, no. 1, Feb. 2022, Art. no. 21, https://doi.org/10.1186/s40537-022-00568-5.

[10] A. Go, R. Bhayani, and L. Huang, "Twitter Sentiment Classification using Distant Supervision," *Stanford University*, CS224N project report, 2000. https://www-cs.stanford.edu/people/alecmgo/papers/TwitterDistantSupervision09.pdf.

[11] M. Thelwall, K. Buckley, G. Paltoglou, D. Cai, and A. Kappas, "Sentiment strength detection in short informal text, "*Journal of the American Society for Information Science and Technology*, vol. 61, no. 12, pp. 2544–2558, 2010, https://doi.org/10.1002/asi.21416.

[12] S. Baker, N. Bloom, S. Davis, and T. Renault, "Twitter-Derived Measures of Economic Uncertainty," 2021, [Online]. Available: https://www.semanticscholar.org/paper/Twitter-Derived-Measures-of-Economic-Uncertainty-Baker-Bloom/3ade96292da964a2dc9c59402c07b728f05699d3.

[13] M. Garg, "Mental Health Analysis in Social Media Posts: A Survey, "*Archives of Computational Methods in Engineering*, vol. 30, no. 3, pp. 1819–1842, 2023, https://doi.org/10.1007/s11831-022-09863-z.

[14] Ş. Genç and E. Surer, "ClickbaitTR: Dataset for clickbait detection from Turkish news sites and social media with a comparative analysis via machine learning algorithms," *Journal of Information Science*, vol. 49, no. 2, pp. 480–499, 2023, https://doi.org/10.1177/0165551521100774.

[15] R. Collobert, "Deep Learning for Efficient Discriminative Parsing," *in Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics,* June 2011, vol. 15, pp. 224–232, Available: https://proceedings.mlr.press/v15/collobert11a.html.

[16] B. Omarov, A. Suliman, and A. Tsoy, "Parallel Backpropagation Neural Network Training for Face Recognition" *Far East Journal of Electronics and Communications*, vol. 16, no. 4, pp. 801–808, Dec. 2016, https://doi.org/10.17654/EC016040801.

[17] E. Shadadi, S. Kouser, L. Alamer, and P. Whig, "Novel approach of Predicting Human Sentiment using Deep Learning" *Journal of Computer Science and Engineering (JCSE)*, vol. 3, no. 2, pp. 107–119, Sept. 2022, https://doi.org/10.36596/jcse.v3i2.533.

[18] P. Chauhan, N. Sharma, and G. Sikka, "The emergence of social media data and sentiment analysis in election prediction," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 2, pp. 2601–2627, Feb. 2021, https://doi.org/10.1007/s12652-020-02423-y.

[19] H. M. Alghamdi, S. H.A. Hamza, A. M. Mashraqi, and S. Abdel-Khalek, "Seeker Optimization with Deep Learning Enabled Sentiment Analysis on Social Media," *Computers, Materials & Continua,* vol. 73, no. 3, pp. 5985–5999, 2022, https://doi.org/10.32604/cmc.2022.031732.

[20] J. Praful Bharadiya, "A Comprehensive Survey of Deep Learning Techniques Natural Language Processing," *European Journal of Technology*, vol. 7, no. 1, pp. 58–66, May 2023, https://doi.org/10.47672/ejt.1473.

[21] M. Chen and L. Zhang, "Application of edge computing combined with deep learning model in the dynamic evolution of network public opinion in emergencies," *The Journal of Supercomputing*, vol. 79, no. 2, pp. 1526–1543, Feb. 2023, https://doi.org/10.1007/s11227-022-04733-8.

[22] M. S. Viñán-Ludeña and L. M. de Campos, "Discovering a tourism destination with social media data: BERT-based sentiment analysis," *Journal of Hospitality and Tourism Technology*, vol. 13, no. 5, pp. 907–921, Aug. 2022, https://doi.org/10.1108/JHTT-09-2021-0259.

[23] A. Nguyen, R. Pellerin, S. Lamouri, and B. Lekens, "Managing demand volatility of pharmaceutical products in times of disruption through news sentiment analysis," *International Journal of Production Research*, vol. 61, no. 9, pp. 2829–2840, May 2023, https://doi.org/10.1080/00207543.2022.2070044.

[24] B. Omarov et al., "Electronic Stethoscope for Heartbeat Abnormality Detection," *in Smart Computing and Communication*, Cham, 2021, pp. 248–258, https://doi.org/10.1007/978-3-030-74717-6_26.

[25] Md. A. A. Noman et al., "A computer vision-based lane detection technique using gradient threshold and hue-lightness-saturation value for an autonomous vehicle," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 13, no. 1, Feb. 2023, Art. no. 347, https://doi.org/10.11591/ijece.v13i1.pp347-357.

[26] R. Pandey and J. P. Singh, "BERT-LSTM model for sarcasm detection in code-mixed social media post," *Journal of Intelligent Information Systems,* vol. 60, no. 1, pp. 235–254, Feb. 2023, https://doi.org/10.1007/s10844-022-00755-z.