

An Effective Sentiment Analysis System using Hybrid Approach of CNN and Machine Learning

Dr. Geentanjali Sharma¹, Kavya Veer², Tejas Weldode³, Vedant Pawar⁴, Imadoddin Shaikh⁵ and Asmita Manna⁶

¹⁻⁶Pimpri Chinchwad College of Engineering, Department of Computer Engineering, Pune, India

Email: geetu2k3@gmail.com, veerkavya221@gmail.com, tejasrahulweldode@gmail.com, vedantpawar333999@gmail.com, shaikhimaddodin75@gmail.com, asmita.manna@pccoepune.org

Abstract—This research introduces a hybrid approach to sentiment analysis with the integration of Convolutional Neural Networks (CNNs) with traditional machine learning techniques. We aim to take advantage of CNNs' deep learning capabilities for extracting features and use traditional algorithms for classification. Our integrated model identifies sentiment polarity across different types of textual data. The model demonstrates increased performance by incorporating pre-trained word embeddings and adopting a multi-channel architecture. We aim to emphasize the efficiency of CNNs in text classification and the crucial role of pre-trained word vectors in deep learning methodologies for Natural Language Processing (NLP). Our results revealed that SVC was the highest-performing classifier compared to logistic regression and Random Forest, achieving an accuracy of 75.45% when used with the Multichannel CNN, indicating opportunities for further refinement and exploration of ensemble methods to boost classification accuracy.

Index Terms— CNN, SVM, Random Forest, Logistic Regression, Word2vec, Sentence classification, Sentimental analysis.

I. INTRODUCTION

Sentiment analysis plays a crucial role in Natural Language Processing (NLP) by extracting sentiments, emotions, and opinions from text. With the increasing volume of unstructured data from sources such as online reviews, social media, and feedback systems, developing effective and scalable sentiment analysis models has become more essential. Although traditional machine learning techniques have shown success in classification tasks, they often fail to capture the intricate semantic and contextual nuances of text. On the other hand, deep learning approaches, especially Convolutional Neural Networks (CNNs), have gained prominence for their ability to automatically learn hierarchical features from raw data.

This paper introduces a hybrid approach to sentiment analysis, combining CNNs' strength in feature extraction with the efficiency of traditional machine learning classifiers for better classification performance. The proposed model integrates pre-trained word embeddings to improve its semantic understanding and utilizes a multi-channel CNN architecture to capture various features from the text. By blending the representational power of CNNs with the interpretability and stability of conventional machine learning methods, this model aims to tackle key challenges in sentiment analysis.

To assess the performance of the proposed hybrid approach, we conducted experiments using several classifiers, including Logistic Regression, Random Forest, and Support Vector Classifier (SVC). The results revealed that

SVC outperformed the other classifiers, achieving an accuracy of 75.45% when combined with the multi-channel CNN architecture. These findings underscore the effectiveness of the hybrid model in detecting sentiment polarity and highlight the significance of pre-trained word embeddings in enhancing the overall performance.

This research contributes to the expanding field of sentiment analysis by showcasing the potential of integrating deep learning techniques with traditional machine learning approaches. Furthermore, it paves the way for future studies, such as exploring ensemble methods and advanced optimization strategies to further improve classification accuracy. The proposed framework not only highlights the advantages of hybrid approaches in NLP but also lays the groundwork for developing more sophisticated sentiment analysis systems.

II. RELATED WORK

In recent years, researchers have increasingly turned to deep learning models to improve sentiment analysis techniques, especially for analyzing product reviews and user-generated content on e-commerce platforms. Yi Liu and collaborators developed a sentiment categorization method for e-commerce reviews that integrates a BERT-BiGRU-Softmax model with hybrid masking, review segmentation, and attention mechanisms. By leveraging BERT for feature extraction, a Bidirectional GRU to capture long-term dependencies, and a Softmax layer for sentiment classification, their model achieved impressive results, surpassing RNN and BiLSTM models with an accuracy above 95.5% [1].

Piyush Jaiswal and his team explored the best method for ranking video games based on brief reviews. Their analysis employed Doc2Vec, TextCNN, and BERT, combined with extensive pre-processing to maximize accuracy. Among the classifiers tested, Doc2Vec was the most effective, achieving an accuracy of 79%. The researchers suggested that further model improvements could expand its applicability to a wider range of games and audiences [2].

Junjie Tang and colleagues applied the NLP-AHP technique to Amazon review data of microwave ovens, focusing on discourse analysis of e-commerce comments. Their findings revealed relatively low prediction accuracy, with an R2 score of around 0.523, indicating challenges in generating accurate predictions [3].

Jingnan Liu and his team tackled data cleaning, feature selection, and model fusion by developing a multi-modal sentiment classification model using BERT, BiGRU, and Softmax. This model incorporated various multi-modal features, significantly improving sentiment classification performance and achieving an impressive accuracy of 90.9% [4].

Rashid Rana and Muhammad Rizwan designed a sentiment analysis model using BiLSTM-CF and BiGRU, specifically aimed at analyzing customer feedback while considering multiple product features and perspectives. Their model demonstrated an accuracy of 92.56%, showcasing its potential to enhance recommendation systems through comprehensive sentiment analysis [5].

Rahul Pradhan and his team applied NLP techniques to analyze sentiment in YouTube comments, utilizing the VADER lexicon for sentiment categorization and exploring how audiences interact with video content through sentiment analysis [6].

Jiajia Duan and colleagues developed a neural network model combining CNN, ML, BiGRU, and attention mechanisms for news text classification. Their model achieved an outstanding accuracy of 96.38%, underscoring the effectiveness of combining convolutional and recurrent layers for feature extraction from text data [7].

The Detox Detection paper proposed a classifier to detect toxic language and replace euphemisms, using machine learning and NLP methods to identify toxic content in specific contexts, providing valuable insights into this approach [8].

Qiannan Zhu and his team introduced the BiGRU-Att-HCNN model, which integrates BiGRU, CNN, and attention mechanisms to extract global features from the IMDB movie review dataset. Their model achieved impressive results with a precision of 0.9034 and a recall of 0.9029 [9].

Md. Shymon Islam and collaborators used a dataset of 44,491 reviews from restaurants and social media platforms to investigate Bangla sentiment analysis in meal evaluations. Their research found that Random Forest (RF) and CNN-BiGRU models outperformed other algorithms, achieving accuracies of 88.73% and 90.96%, respectively [10].

Nandakumar R applied sentiment analysis to categorize student feedback into positive and negative comments, analyzing 2200 student reviews. He found that 60% of the comments were positive and 40% negative, utilizing machine learning and NLP techniques for analysis [11].

Tian Lan and his team used Baidu's ERNIE pretraining model to capture long semantic dependencies and knowledge. By feeding ERNIE's output into BiGRU, followed by an attention mechanism, they achieved an

accuracy of 89.40%, a recall of 94.37%, and an F1 score of 0.8837 [12].

Kunli Zhang employed NLP techniques like word cloud analysis, TF-IDF, and regular matching to extract text features from computer dissertations and identify issues raised in review comments [13].

Yoon Kim and his team applied CNNs with Word2Vec embeddings for sentence classification, demonstrating the effectiveness of CNN-static models and showing improvements with fine-tuning embeddings (CNN-non-static). Their multi-channel model, combining both static and non-static embeddings, showed promising results [14].

JianghaiLv and colleagues proposed a BERT-BiGRU-CRF model for entity relationship extraction, achieving strong results with a precision of 84.3 and a recall of 85.2, highlighting the effectiveness of combining pre-trained word vectors with downstream tasks [15].

Kavya N.C. and her team used unsupervised N-programs and POS tagging to classify positive and negative customer comments on e-commerce websites. They employed a negative lexicon to identify negative terms, flagging them as spam [16].

III. DATASET

The dataset employed in this study consists of a collection of general sentences, each paired with a sentiment label of either positive or negative. This dataset serves as the foundation for training and evaluating the sentiment analysis models proposed in this research. Each sentence is annotated with sentiment labels, facilitating the supervised learning process essential for sentiment classification tasks. The dataset encompasses a broad array of textual sources, ensuring that the models are assessed across a variety of contexts. By utilizing this dataset, the study aims to tackle the challenges and complexities of sentiment analysis in real-world scenarios, contributing to the advancement of research in natural language processing.

TABLE I: THE NUMBER OF TRAINING AND TESTING SENTENCES TAKEN FROM BOTH DATASETS FOR THE PROPOSED EMPIRICAL PERFORMANCE ASSESSMENT OF CNN-ML MODELS

Class	Train	Test
Positive	4314	518
Negative	4321	549

IV. METHODOLOGY

We propose a composite framework for sentiment analysis, by combining the representational capacity of Convolutional Neural Networks (CNNs) with the discriminative power of traditional machine learning classifiers. For our framework, we use shallow classifiers for robust sentiment classification and pre-trained word embeddings to leverage semantic knowledge for feature representation.

A. Word2Vec

Word2Vec is an unsupervised learning framework that generates dense vector representations of words. It encapsulates both semantic and syntactic relationships within a continuous vector space. The embeddings generated by Word2Vec are useful in downstream natural language processing tasks. The methodology encompasses two primary architectures:

1. Continuous Bag-of-Words (CBOW): The CBOW model predicts the target word w_t with the help of its contextual neighbors within a specified window size c . This is done by maximizing the conditional probability,

$$P(\omega_t | \omega_{t-c}, \dots, \omega_{t-1}, \omega_{t+1}, \dots, \omega_{t+c}) (1)$$

where the word order within the context window is disregarded, emphasizing the "bag-of-words" assumption.

2. Continuous Skip-Gram: In contrast, the continuous skip-gram model predicts surrounding words within a context window of size c based on the target word w_t , optimizing,

$$P(\omega_{t-e}, \dots, \omega_{t-1}, \omega_{t+1}, \dots, \omega_{t+e} | \omega_t) (2)$$

Both these architectures utilize hierarchical softmax or negative sampling to be computationally efficient. This allows them to be applicable for large-scale datasets.

B. Feature Extraction Techniques

Feature extraction techniques are used for converting raw textual data into machine learning classifiers. Prominent methodologies utilized include:

1. Bag-of-Words (BoW): This model represents textual data in the form of a sparse matrix. The feature space is

defined with the frequency of each word across documents. This approach automatically discards the order of the words but retains the information about its frequency.

2. Term Frequency-Inverse Document Frequency (TF-IDF): This model refines the BoW model by weighting word importance relative to a document and the corpus. It is mathematically expressed as,

$$TF - IDF(t, d) = TF(t, d) \cdot IDF(t) \quad (3)$$

where,

$$TF(t, d) = \frac{\text{Frequency of } t \text{ in } d}{\text{Total terms in } d}, IDF(t) = \log \frac{N}{n_t} \quad (4)$$

with n_t representing the number of documents containing the term t and N representing the total number of documents.

C. Convolutional Neural Networks (CNNs)

CNN is a class of deep learning models known for their hierarchical feature extraction capabilities. They are suitable for particularly high-dimensional structured data. The architectural hierarchy includes:

1. Convolutional Layers: These layers apply convolution operations to extract localized patterns to produce feature maps. These feature maps encode spatial and semantic hierarchies.
2. Pooling Layers: These layers help in reducing the spatial dimensions by aggregating feature map activations. They help in mitigating overfitting and computational complexity.
3. Fully Connected (FC) Layers: These layers function as a classifier, combining the learned features in previous steps to produce the final result.

The CNN architecture continuously learns increasingly complex patterns, evolving from fundamental features to higher-level abstractions.

D. Classifiers

We have used three different machine learning classifiers to evaluate the hybrid framework:

1. Logistic Regression (LR): Logistic Regression helps to model the posterior probability of class membership using the logistic function,

$$P(y = 1 | X) = \frac{1}{1 + e^{-z}} \quad (5)$$

where $Z = X\beta + \epsilon$, X represents the feature matrix, β denotes the coefficient vector, and ϵ is the error term.

2. Random Forest (RF): Random Forest generates multiple decision trees using bootstrapped data subsets and aggregates their predictions. Thus it is an ensemble learning method. Majority voting then determines the final output or averaging, thereby reducing variance and enhancing robustness.

3. Support Vector Classifier (SVC): SVC identifies the optimal hyperplane, $\omega^T X + b = 0$ that helps in maximizing the margin among classes in the feature space. The objective function for margin maximization can be formulated as,

$$\min_{w, b} \frac{1}{2} \|\omega\|^2, \text{subject to } y_i(\omega^T X_i + b) \geq 1, \forall_i \quad (6)$$

where $y_i \in \{-1, 1\}$ represents the class labels, and X_i is the feature vector.

V. MODEL 1: SINGLE-CHANNEL STATIC CNN WITH FEATURE EXTRACTION

Our first model employs a single-channel TextCNN architecture. Which is enhanced by feature extraction and subsequent classification using logistic regression, support vector classifier (SVC) and random forest. This hybrid model combines deep learning with traditional machine learning to perform sentiment classification. The architecture is a combination of the following components.[Figure-1]

A. Preprocessing and Embedding Layer

During preprocessing, the input textual data is converted into a sequence of word indices. These indices are then mapped to a pre-trained Word2Vec embedding space. The embedding layer is configured with the parameter “trainable = True”, enabling the embeddings to be fine-tuned during training. This layer functions as the initial feature extractor, providing dense, semantically rich representations of the input text to the layers that follow.

Mathematically, the embedding process can be represented as:

$$E(X) = \{v_1, v_2, \dots, v_n\}, v_i \in R^d \quad (7)$$

where $E(X)$ represents the embedded sequence, n is the sequence length, and d is the dimensionality of the word embeddings.

B. Convolutional and Pooling Layers

The model uses a single-channel TextCNN architecture. The 1D convolutional layers (Conv1D) with 128 filters

and a kernel size of 3 are applied to the embedded sequence. These layers help to capture local patterns in the data, such as n-grams, within the text.

The convolution operation is expressed as:

$$f_{i,j} = \sigma\left(\sum_{k=1}^d W_{j,k} \cdot x_{i+k-1,k} + b_j\right) \quad (8)$$

Where $f_{i,j}$ represents the activation of the j -th filter at position i , $W_{j,k}$ and b_j denote the filter weights and biases, and σ is the activation function (e.g ReLU).

After convolution, max pooling is applied to down-sample the feature maps while retaining the salient features. Max pooling is defined as:

$$P_j = \max_{i \in R} f_{i,j} \quad (9)$$

Where p_j represents the pooled output for filter j , and R is the pooling window.

C. Feature Extraction

Traditional machine learning classifiers use the flattened output of the CNN architecture as their feature set. These features are used to train:

1. Logistic Regression (LR): The linear classifier uses a sigmoid activation function to calculate the posterior probability of the class label γ .
2. Random Forest (RF): An ensemble-based classifier that aggregates predictions from multiple decision trees.
3. Support Vector Classifier (SVC): A margin-based classifier that identifies the optimal hyperplane for separating classes.

The extracted features F from the CNN are mathematically represented as:

$$F = Flatten(CNN(E(X)) \quad (10)$$

where $F \in R^m$ with m being the total number of features.

D. Training and Evaluation

We use a backpropagation algorithm for training the model on the training dataset. This approach minimizes the loss function for the CNN layers. In traditional classifiers, the features extracted from the CNN are split into training and testing datasets. We use evaluation metrics such as accuracy, precision, recall, and F1-score, to assess the performance of the model.

To enhance performance we use hyperparameter optimization. Key hyperparameters include the filter size, kernel size, learning rate, and batch size for the CNN, as well as the number of estimators and depth for the random forest classifier.

This hybrid approach effectively combines the representation learning capabilities of CNNs with the robustness of traditional classifiers. Hence we achieve competitive performance in sentiment classification tasks.

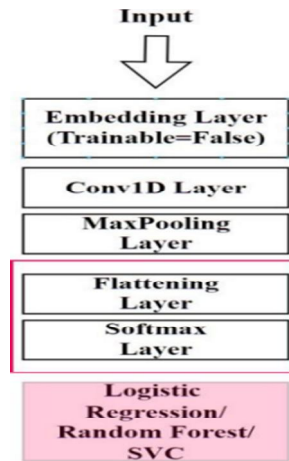


Figure 1. Single channel static CNN-ML mode.

VI. MODEL 2: SINGLE-CHANNEL TEXTCNNWITH FEATURE EXTRACTION

The second model improves upon the single-channel TextCNN architecture by integrating feature extraction With classification through logistic regression, support vector classifier (SVC), and random forest. The architecture comprises the following components.[Figure-3]

A. Preprocessing and Embedding Layer

In this layer the input text data is preprocessed to convert it into sequences of word indices. These word indices are then passed through an embedding layer initialized with pre-trained Word2Vec embeddings. This layer is configured with the parameter trainable = False to maintain static embeddings. This layer serves as the foundational feature representation for the other following layers. Thus, ensuring robust and semantically rich initialization.

Mathematically, the embedding transformation is expressed as:

$$E_{static}(X) = \{v_1, v_2, \dots, v_n\}, v_i \in R^d(11)$$

where n represents the sequence length, and d is the dimensionality of the word embeddings.

B. Convolutional and Pooling Layers

This model uses a single-channel TextCNN architecture. The 1D convolutional layers (Conv1D) with 128 filters and a kernel size of 3 are applied to extract local patterns from the word embeddings. These patterns represent contextual n-grams within the text.

The convolution operation is defined as,

$$f_{i,j} = \sigma(\sum_{k=1}^d W_{j,k} \cdot x_{i+k-1,k} + b_j)(12)$$

Where $f_{i,j}$ denotes the activation for the j -th filter at position i , $W_{j,k}$ and b_j are filter weights and biases and σ represents the activation function.

Max pooling is subsequently applied to extract salient features:

$$p_j = \max_{i \in R} f_{i,j} (13)$$

where p_j is the pooled output for filter j , and R represents the pooling window.

C. Feature Extraction

The flattened output of the CNN is used in the form of input features for downstream machine learning classifiers. These features include logistic regression, random forest, and SVC. The conversion of high-dimensional outputs of the convolutional layers into a compact, informative representation is ensured by the feature extraction process.

The extracted features can be mathematically expressed as:

$$F = Flatten(CNN(E_{static}(X))(14)$$

Where $F \in R^m$ and m is the dimensionality of the flattened feature vector.

D. Training and Evaluation

Training and testing of the model is done using the specified training and testing datasets. Model optimization is done by fine-tuning the hyper parameters such as learning rate, kernel size, and classifier-specific parameters (e.g., the number of estimators in random forest or the regularization parameter in SVC). For evaluation we employ some standard performance metrics, such as, recall, precision, accuracy, and F1-score.

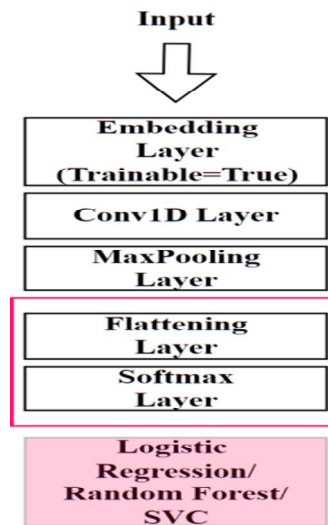


Figure 2. Single channel non-static CNN-ML model.

VII. MODEL 3: DUAL-CHANNEL TEXT CNN WITH FEATURE EXTRACTION

By modifying the single-channel architecture, this model introduces a dual-channel TextCNN framework to utilize both the static and non-static word embeddings. This allows for enhanced feature representation. This dual-channel approach allows the model to capture both the fixed semantic relationships as well as the adaptable contextual patterns.[Figure-4]

A. Dual-Channel Architecture

The model employs two distinct embedding layers:

1. Static Channel: Pre-trained Word2Vec embeddings with fixed weights (trainable = False) are used to retain the inherent semantic relationships.
2. Non-Static Channel: Trainable embeddings make the model to adaptively learn the task-specific representations during trainings.

The embeddings for the static and non-static channels can be represented as:

$$E_{static}(X) = \{v_1^{static}, \dots, v_n^{static}\}, E_{non-static}(X) = \{v_1^{non-static}, \dots, v_n^{non-static}\} \quad (15)$$

B. Convolutional and Pooling Layers

Both these channels undergo identical convolutional operations using 1D convolutional layers (Conv1D) with 128 filters and a kernel size of 3. To extract the most salient features, max pooling is applied independently to each channel.

The convolution and pooling operations for each channel are performed as described in Model 2.

C. Feature Extraction and Fusion

Unified representation is formed by concatenating the features extracted from both the static and non-static channels:

$$F = \text{Concat}(F_{static}, F_{non-static}) \quad (16)$$

where F_{static} and $F_{non-static}$ denote the flattened feature vectors from the static and non-static channels, respectively.

Dropout regularization is applied to the concatenated feature vector to prevent over fitting.

D. Training and Evaluation

The dual-channel model is trained using dual-channel input data. It allows the static and non-static embeddings to complement each other. The extracted features are classified using SVC, logistic regression, and random forest. To optimize the model for metrics such as recall, precision, and accuracy, hyper parameter tuning is performed.

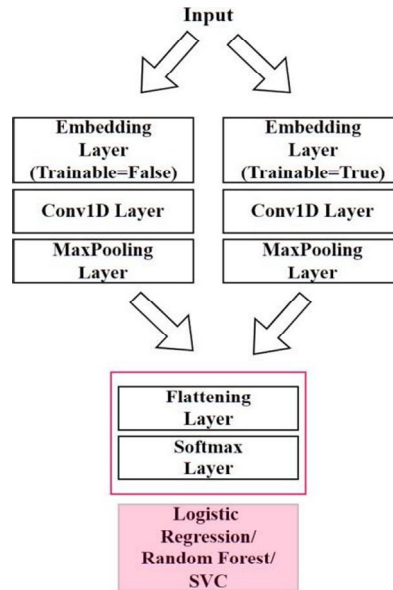


Figure 3. Multi-channel static, non-static CNN-ML model.

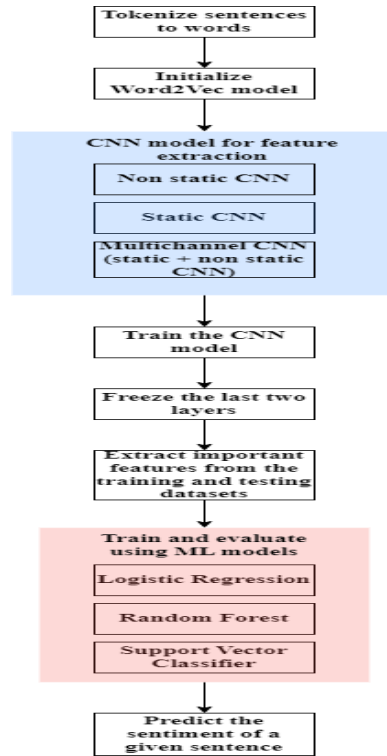


Figure 4. Flowchart for hybrid CNN and ML model for binary classification

TABLE II: THE PARAMETERS SET DURING THE PERFORMANCE ASSESSMENT OF CNN-ML MODELS

Parameter	Value
Optimizer	Adam
Learning rate	8×10^{-8}
Batch- size	32
Filter	128
Kernel size	3
Padding	Valid
Activation	ReLu
n_estimator	85
min_samples_spin	5
min_samples_leaf	2
max_features	sqrt
SVC_kernel	Rbf

VIII. RESULTS AND ANALYSIS

This section shows the estimation results of the unique CNN models—Static CNN, Non -Static CNN, and Multi-Channel CNN—integrated with different machine learning classifiers, involving logistic regression, Support Vector Classifier (SVC), and Random Forest. The results secured are examined, featuring observations, insights and key outcomes are analyzed in detail.

A. Static CNN with Logistic Regression

Results: Reached an accuracy of 71.6% during testing and 74.51% during training.

Analysis: In training and testing the small gap between accuracy shows over-fitting. A simple model known as Logistic regression delivers competitive performance.

Outcomes: Logistic Regression indicates that it can serve as a feasible classifier for CNN-based models, specially for uncomplicated classification tasks where interoperability is essential

B. Static CNN with Support Vector Classification (SVC)

Results: Reached an accuracy of 70.95% during testing and 75.45% during training.

Analysis: In training and testing the large gap between accuracy shows over-fitting. The optimization of hyper-parameters such as (CCC, γ gamma, and kernel type) for SVC are required to improve generalization.

Outcomes: SVC indicates that it is effective for CNN models but requires careful tuning of hyper-parameters to reduce over-fitting.

C. Static CNN with Random Forest

Results: Reached an accuracy of 70.29% during testing and 99.8% during training.

Analysis: Random Forest opposes generalization when used together with static CNN significantly this happens due to over-fitting. Although it captures training data patterns effectively, but it fails to capture on unseen data

Outcomes: Random forest indicates need for hyper-parameter tuning (e.g., tree depth, number of trees) and regularization to reduce overfitting and improve robustness.

D. Non-Static CNN with Logistic Regression

Results: Reached an accuracy of 71.6% during testing and 99.7% during training.

Analysis: In the Non-Static CNN the trainable embedding shows particular over-fitting. But Static CNN with logistic regression matches the testing accuracy which shows the productiveness of trainable embedding for depiction learning.

Outcomes: In Non-Static CNN with logistic regression indicates that trainable embedding requires regularization techniques to control over-fitting.

E. Non-Static CNN with Support Vector Classification (SVC)

Results: Reached an accuracy of 70.95% during testing and 99.99% during training.

Analysis: In training and testing the gap between accuracy shows over-fitting similar to other SVC-based models. But Non-static CNN is slightly more adaptable because it benefits from task-specific embedding optimization.

Outcome: Non-Static CNN with Support Vector Classification(SVC) indicates the need for regularization and hyper-parameter tuning in SVC to grasp the benefits for sentiment classification.

F. Non-Static CNN with Random Forest

Results: Reached an accuracy of 70.29% during testing and 99.7% during training.

Analysis: Random Forest captures patterns in training data but it fails to vulgarize effectively. While Over-fitting exists, as seen in the Static CNN-Random Forest combination.

Outcomes: Indicates the need of taking a look at other ensemble techniques or applying regularization to upgrade Random Forest's performance.

G. Multi-Channel CNN with Logistic Regression

Results: Reached an accuracy of 74.51% during testing and 96.78% during training.

Analysis: The static and non-static embedding are effectively utilized by Multi-Channel CNN, Due to this it results in improved performance compared to single-channel architectures. In this hybrid environment as a classifier Logistic regression shows its strength.

Outcomes: Multi-Channel CNN's indicates that it enhances feature extraction, making logistic regression a authentic choice for sentiment analysis tasks.

H. Multi-Channel CNN with Support Vector Classification (SVC)

Results: Reached an accuracy of 75.45% during testing and 97.01% during training.

Analysis: In Multi-Channel CNN with Support Vector Classification profits from the enriched feature representation which leads it to superior performance. The effectiveness of the architecture highlights the reduced over-fitting compared to other models.

Outcomes: Multi-Channel CNN-SVC indicates that it is a top-performing model, exposing its strength for accurate and robust sentiment classification.

I. Multi-Channel CNN with Random Forest

Results: Reached an accuracy of 75.35% during testing and 97.26% during training.

Analysis: In the Multi-Channel CNN setup Random Forest performs aggressively with shrink over-fitting compared to single-channel architectures. The ensemble method takes advantage of the comprehensive feature extraction.

Outcomes: In Multi-Channel CNN with Random Forest indicates that it can be used as a strengthened classifier

for text classification tasks.

J. Summary of Observations and Insights

Performance Superiority: Across all classifiers, Multi-Channel CNN consistently scored better than Static and Non-Static CNNs, highlighting the benefits of using both static and trainable embeddings.

Classifier Effectiveness: SVC was the best-performing classifier, greatly aided by Multi-Channel CNN's enriched feature extraction.

Over-fitting Challenges: In single-channel architectures, Random Forest showed the most overfitting, highlighting the need of tuning and regularisation.

Hybrid Model Potential: By merging CNN-based architectures with traditional classifiers offers a ideal method for sentiment analysis, balancing feature representation and classification robustness.

IX. OUTCOMES

Hybrid Model Performance: Multi-Channel CNN-SVC proposed model reached the topmost overall testing accuracy of 75.45%, displaying the robustness of merging deep learning architecture with traditional classifiers.

2. Classifier Ranking: The most successful classifier that was found is SVC, followed by Random Forest and logistic regression that are dependent on testing accuracy across models.

3. Architectural Insights: Multi-Channel CNN's demonstrated superior performance compared to Static and Non-

3. Architectural Insights: The static and non-static CNN were having inferior performance compared to Multi-channel CNN which emphasized the importance of using various embedding channels for robust feature extraction.

4. Future Directions: To improve model performance and reduce overfitting the study highlights the possibilities of adjusting hyper-parameters, by using ensemble techniques and investigating advanced regularization methods.

TABLE III: TESTING ACCURACIES OBTAINED FOR ML MODEL OVER CONSIDERED HYBRID CNN MODEL WITH EARLY STOPPING

CNN Model	Classification Technique	Testing Accuracy (%)	Training Accuracy (%)
Static CNN	Logistic Regression	71.6	74.51
	svc	70.95	75.45
	Random Forest	70.29	99.8
Non-Static CNN	Logistic Regression	71.6	99.7
	svc	70.95	99.99
	Random Forest	70.29	99.7
Multi-Channel CNN	Logistic Regression	74.51	96.78
	svc	75.45	97.01
	Random Forest	75.35	97.26

X. CONCLUSION

In hybrid environment experimental results focus on the dominance of Multi-Channel CNN architectures. These models offer particular improvements in classification accuracy by merging static and non-static word embedding. In addition to this to reduce over fitting and maximize performance the analysis shows the necessity of classifier selection and optimization. In future ensemble techniques and advanced optimization strategies will be explored to later improve the strength of hybrid CNN models.

REFERENCES

- [1] Detox: NLP Based Classification And Euphemistic Text Substitution For Toxic Comments. (2021). 2021 12th International Conference on Computing Communication and Networking Technologies, ICCCNT 2021. <https://doi.org/10.1109/ICCCNT51525.2021.9579846>
- [2] Duan, J., Zhao, H., Qin, W., Qiu, M., & Liu, M. (2020). News Text Classification Based on MLCNN and BiGRU Hybrid Neural Network. Proceedings - 2020 3rd International Conference on Smart BlockChain, SmartBlock 2020, 137–142. <https://doi.org/10.1109/SmartBlock52591.2020.00032>
- [3] Jaiswal, P., Setia, H., Raghuvanshi, P., & Randhawa, P. (2023). A Natural Language Processing Model for Predicting Five-Star Ratings of Video Games on Short-Text Reviews. Engineering Proceedings, 59(1). <https://doi.org/10.3390/engproc2023059058>
- [4] Liu, J., Sun, Y., Zhang, Y., & Lu, C. (2024). Research on Online Review Information Classification Based on Multimodal Deep Learning. Applied Sciences (Switzerland), 14(9). <https://doi.org/10.3390/app14093801>

- [5] Liu, Y., Lu, J., Yang, J., & Mao, F. (2020). Sentiment analysis for e-commerce product reviews by deep learning model of Bert-BiGRU-Softmax. *Mathematical Biosciences and Engineering*, 17(6), 7819–7837. <https://doi.org/10.3934/MBE.2020398>
- [6] Pradhan, R. (2021). Extracting Sentiments from YouTube Comments. *Proceedings of the IEEE International Conference Image Information Processing*, 2021-November, 142–145. <https://doi.org/10.1109/ICIIP53038.2021.9702561>
- [7] Rana, M. R. R., Nawaz, A., Ali, T., El-Sherbeeney, A. M., & Ali, W. (2023). A BiLSTM-CF and BiGRU-based Deep Sentiment Analysis Model to Explore Customer Reviews for Effective Recommendations. In *Engineering, Technology and Applied Science Research* (Vol. 13, Issue 5, pp. 11739–11746). Dr D. Pylarinos. <https://doi.org/10.48084/etasr.6278>
- [8] Tang, J., & Guo, J. (2022). An Analysis Method for Online Shopping Platform Comments Based on NLP-AHP: Taking Amazon as An Example. *Proceedings - 2022 3rd International Conference on Education, Knowledge and Information Management, ICEKIM 2022*, 1080–1085. <https://doi.org/10.1109/ICEKIM55072.2022.00234>
- [9] Zhu, Q., Jiang, X., & Ye, R. (2021). Sentiment Analysis of Review Text Based on BiGRU-Attention and Hybrid CNN. In *IEEE Access* (Vol. 9, pp. 149077–149088). Institute of Electrical and Electronics Engineers Inc. <https://doi.org/10.1109/ACCESS.2021.3118537>
- [10] Islam, M. S., & Masudul Alam, K. (2023). Sentiment Analysis on Bangla Food Reviews Using Machine Learning and Explainable NLP. *2023 26th International Conference on Computer and Information Technology, ICCIT 2023*. <https://doi.org/10.1109/ICCIT60459.2023.10441309>.
- [11] Nandakumar, R., Pallavi, M. S., Pramath, P. H., & Hegde, V. (2022). Sentimental Analysis on Student Feedback using NLP & POS Tagging. *International Conference on Edge Computing and Applications, ICECAA 2022 - Proceedings*, 309313. <https://doi.org/10.1109/ICECAA55415.2022.9936569>
- [12] Lan, T., Li, X., Gao, Y., & Yuan, J. (2022). An Effective Approach for Rumor Detection Based on ERNIE-BiGRU-Attention. *2022 2nd International Conference on Consumer Electronics and Computer Engineering, ICCECE 2022*, 859–862. <https://doi.org/10.1109/ICCECE54139.2022.9712700>
- [13] Zhang, K., Ma, Y., & Zan, H. (2022). Analysis and suggestion of double-blind review comments based on NLP technology. *Proceedings - 2022 Global Conference on Robotics, Artificial Intelligence and Information Technology, GCRAIT 2022*, 564568. <https://doi.org/10.1109/GCRAIT55928.2022.00123>
- [14] Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification. <http://arxiv.org/abs/1408.5882>
- [15] Lv, J., Du, J., Zhou, N., & Xue, Z. (2020). BERT-BiGRU-CRF: A novel entity relationship extraction model. *Proceedings - 11th IEEE International Conference on Knowledge Graph, ICKG 2020*, 157–164. <https://doi.org/10.1109/ICKG50248.2020.00032>
- [16] Dinesh, L., Kaviya, G., Kavya, N. C., Kousalya, A., & Lavanya, N. (2022). Product Reviews on Opinion Mining using NLP Techniques. *7th International Conference on Communication and Electronics Systems, ICCES 2022 - Proceedings*, 952–957. <https://doi.org/10.1109/ICCES54183.2022.9836019>