DeepTweetScan: A Hybrid Approach for Detecting Synthetic Tweets via CNN and FastText

Surapureddy Bharathi[1], Thummeti Preethi[2], Urumu Anil Babu[3], Shaik Sadhik[4], Sadineni Ashok[5],Mr. Dr. K. V. Rama Rao[6]

Electronics & communication engineering, Chalapathi Institute of Engineering & Technology, LAM, Guntur-Andhra Pradesh[1,2,3,4,5]

[6]Professor, Electronics & communication engineering, Chalapathi Institute of Engineering & Technology, LAM, Guntur

***Abstract*— The pervasive influence of social media has facilitated both the rapid distribution of information and the emergence of malicious digital content. A significant challenge is the generation of deepfake tweets—synthetic posts designed to imitate human-authored content—often propagated by automated bots. These deceptive tweets can shape public opinion and trigger widespread misinformation. This paper introduces an intelligent framework for identifying deepfake tweets using FastText embeddings integrated with a suite of machine learning and deep learning models. We evaluate and compare traditional classifiers (Naïve Bayes, Logistic Regression, Decision Tree, Random Forest) alongside deep architectures like CNN and LSTM. Our findings reveal that CNN achieves superior performance in terms of both accuracy and reliability. Additionally, a hybrid model is developed by combining CNN-based feature extraction with Random Forest classification, resulting in further accuracy gains. Utilizing the TweepFake dataset, the proposed framework supports all stages—from data cleaning and embedding to model training and real-time tweet analysis. The system effectively distinguishes between genuine and bot-generated content, offering a practical tool to address the growing threat of misinformation on social platforms.**

Keywords— Fake Tweet Detection, Deep Learning, CNN, FastText, Hybrid Model, Bot Classification, Misinformation, NLP, Social Media Forensics.

I. INTRODUCTION

In the digital era, social media platforms like Twitter, Facebook, and Instagram have become dominant forces in shaping public perception, driving communication, and disseminating real-time information across the globe. Their ubiquitous presence and accessibility have transformed how individuals consume news, express opinions, and interact with society. However, the same features that empower free expression also make social media vulnerable to exploitation. Among the most pressing concerns in recent years is the emergence of deepfake tweets—synthetically generated messages designed to imitate human-authored content, often spread by automated bots for malicious intent.Deepfake tweets leverage advancements in natural language generation and are often indistinguishable from authentic human posts. They are crafted using AI-powered bots trained to replicate human syntax, tone, sentiment, and grammar. These deceptive tweets can serve various harmful purposes: promoting misinformation, manipulating public opinion, spreading political propaganda, conducting social engineering attacks, or undermining trust in media and institutions. Unlike earlier generations of spam or scripted bots, today's deepfake tweets are highly contextual and linguistically natural, making them difficult to detect using traditional rule-based filters or keyword analysis.

Manual moderation and verification processes are not scalable for platforms that handle millions of daily tweets. The sheer volume and velocity of content necessitate automated and intelligent systems that can accurately analyze, interpret, and classify tweets in real-time. This research addresses this growing challenge by proposing a deep learning-based framework capable of detecting deepfake tweets through robust text analysis.Our proposed system, named TweetScan, employs FastText word embeddings to capture rich semantic and sub-word information from social media text. FastText is especially suited for noisy and informal text formats—characteristic of social media—due to its ability to process out-of-vocabulary terms, slang, abbreviations, and compound words. These embeddings are then used as input features to train multiple classification models including traditional machine learning algorithms (Naïve Bayes, Logistic Regression, Decision Tree, Random Forest) and advanced deep learning architectures like Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks.

This study contributes a holistic, scalable, and interpretable solution for deepfake tweet detection. By leveraging both the strengths of deep learning and the semantic depth of FastText embeddings, it offers a practical tool for researchers, policymakers, journalists, and social media moderators. As the threat landscape continues to evolve, the proposed system lays the groundwork for future enhancements such as multilingual support, multi-modal content detection (text, image, video), and deployment across diverse platforms for stronger misinformation mitigation.

II. RELATED WORKS

The detection of deceptive and AI-generated content on social platforms has become increasingly critical due to the rise in synthetic media and automated misinformation campaigns. Over the years, the focus of research has shifted from traditional rule-based detection to advanced machine learning and deep learning models capable of handling the nuances of modern social media language and behavior.

Doi:10.48047/ijiee.2025.15.3.3

2.1 Traditional Approaches to Fake Tweet Detection

In the early stages of fake content detection, researchers relied heavily on handcrafted features such as tweet frequency, account metadata, retweet ratios, and user follower patterns. These attributes were often fed into classifiers like Logistic Regression, Naïve Bayes, and Random Forest, which achieved reasonable performance on well-structured datasets. For example, Dolhansky et al. (2020) introduced benchmark datasets and evaluation frameworks for social media deepfake detection using structured features.However, as bots began emulating more human-like posting behavior and linguistic style, these methods quickly became outdated. Static feature sets were not sufficient to capture the semantic depth or deception embedded in sophisticated text-based deepfakes.

2.2 Evolution of Text Embedding Techniques

To better represent textual data, researchers moved towards semantic embedding methods. Traditional vectorization techniques like TF-IDF and Bag of Words proved inadequate for informal, context-rich environments like Twitter due to their lack of semantic comprehension and sparse nature.This led to the adoption of Word Embeddings, such as GloVe, Word2Vec, and later FastText. FastText, in particular, demonstrated strong performance in capturing sub-word information, which is beneficial in noisy datasets that include slang, abbreviations, and typos—common in tweets. Studies by Nayak et al. (2021) and Sharma et al. (2020) emphasized the superiority of FastText for Twitter-based classification tasks due to its ability to generate embeddings for out-of-vocabulary words.

2.3 Deep Learning Techniques in Bot and Deepfake Detection

The rise of deep learning brought significant advancements in tweet classification and fake content identification. Models like Convolutional Neural Networks (CNNs) have shown exceptional ability in identifying local textual patterns, especially useful for short and context-sensitive messages like tweets. CNNs can effectively recognize n-gram features, sarcasm patterns, and deceptive language signatures.Recurrent Neural Networks (RNNs) and particularly LSTM (Long Short-Term Memory) models have been used to model temporal dependencies and long-range relationships in tweet threads. Studies such as by Thies et al. (2019) showed that LSTM-based architectures outperformed traditional models in tasks involving content continuity and behavioral context analysis.

2.4 Hybrid and Ensemble Approaches

A notable trend in recent research involves the fusion of deep learning feature extractors with ensemble decision-makers. Hybrid models—where CNN or LSTM layers are used to extract high-level features and classifiers like Random Forest or Gradient Boosting make the final prediction—have proven highly effective.Afchar et al. (2020) proposed a compact ensemble model that integrated CNN-based representations with shallow classifiers to improve speed and accuracy in fake text classification. Similarly, Rossler et al. (2021) found that ensemble learning improved robustness against adversarial tweet structures and unseen writing patterns.

These models address overfitting issues common in deep learning by leveraging the generalization power of ensemble methods while retaining the deep model's representational capacity.

2.5 Existing System

The existing systems for detecting fake or deepfake content on social media primarily rely on conventional machine learning models and basic feature engineering techniques. These systems typically use term-frequency-based embeddings such as TF or TF-IDF to convert tweet text into numerical vectors and apply classifiers like Naïve Bayes, Logistic Regression, or Random Forest to predict whether the content is genuine or machine-generated. Although these methods perform adequately on well-formatted datasets, they struggle with the noisy, informal, and abbreviated nature of social media text. Moreover, these models often fail to capture the semantic and syntactic complexities of language due to the lack of contextual understanding in the feature extraction process.

In some existing approaches, deep learning models like LSTM or CNN are used independently but without embedding optimization, which limits their performance. These models are often not trained with social media-specific data and thus lack the robustness required for real-world tweet analysis. Furthermore, most systems operate in a static, offline setting and lack real-time prediction capability or user interface integration. As a result, end-users cannot directly interact with the system to check the authenticity of tweets on the fly. Additionally, few existing solutions support comparative analysis between multiple models or provide visualization of performance metrics such as accuracy, precision, recall, and F1-score

2.5.1 Limitations of Existing System

- **Limited Context Understanding:** Most systems rely on TF or TF-IDF embeddings, which do not preserve semantic or sub-word information, reducing contextual accuracy.
- **Poor Handling of Noisy Social Media Text**: Many models are not optimized for informal language, abbreviations, or emojis commonly found in tweets.
- **Lack of Real-Time Prediction**: Existing solutions typically operate in a batch mode without supporting live tweet classification or user interaction.
- **No Interface for End-Users**: Many models lack a front-end system where users can input tweets and receive instant predictions.
- **Weak Generalization**: Standalone machine learning models may overfit or underperform when tested on diverse or unseen tweet formats.
- **No Hybrid Approach**: Most systems do not combine deep learning feature extraction with robust classifiers like Random Forest to enhance performance.
- **Insufficient Comparative Evaluation**: There is often no side-by-side evaluation of multiple models to determine the best-performing algorithm.

Doi:10.48047/ijiee.2025.15.3.3

2.6 Proposed System

The proposed system presents an end-to-end framework for detecting deepfake tweets by leveraging FastText embeddings and deep learning techniques. It begins with a robust text preprocessing pipeline that removes noise, stop words, and special characters, followed by FastText embedding generation to convert tweets into rich numerical vectors that retain semantic and sub-word information. These vectors are then used to train multiple machine learning and deep learning algorithms, including CNN, LSTM, Naïve Bayes, Decision Tree, Logistic Regression, and Random Forest.Among these, the Convolutional Neural Network (CNN) demonstrated the highest accuracy in initial tests due to its ability to extract localized features in short texts. To further improve the system, a hybrid model was introduced in which optimized features from the CNN layer are passed into a Random Forest classifier. This Hybrid CNN model combines the strengths of deep feature learning and robust decision-making, resulting in better generalization and accuracy.

The system is designed to support multiple functions: dataset loading, preprocessing, embedding, model training, and real-time prediction. It features a user-friendly web-based GUI that allows users to input tweet text and receive immediate classification as either *Human* or *Deep Bot*. Additionally, all trained models are compared based on accuracy, precision, recall, and F1-score, with visual outputs in tabular and graphical formats. This comprehensive approach makes the system highly scalable, interpretable, and practical for both research and deployment.

2.6.1 Advantages of the Proposed System

- **FastText Embedding for Richer Context**: Embedding captures sub-word information and semantics, making the model more accurate for informal social media language.
- **High Accuracy with CNN and Hybrid Model**: CNN alone achieves strong performance, and the Hybrid CNN-Random Forest model enhances it further.
- **Multi-Model Comparison**: Enables performance benchmarking across different ML and DL algorithms to identify the best model for deployment.
- **Real-Time Tweet Prediction**: Allows users to enter tweets directly into the system and receive classification output instantly.
- **User-Friendly Interface**: Offers a simple web GUI to facilitate interaction without requiring technical expertise.
- **Visualization of Results**: Displays evaluation metrics such as accuracy, precision, recall, and F1-score in both tabular and graphical formats.
- **Modular and Extensible Design**: Supports easy updates and integration with additional datasets or algorithms for future improvements.

III. Proposed Methodology

The proposed methodology involves a multi-stage pipeline that processes tweet data, converts it into meaningful embeddings, trains multiple classification models, and provides real-time deepfake prediction.

3.1 System Architecture

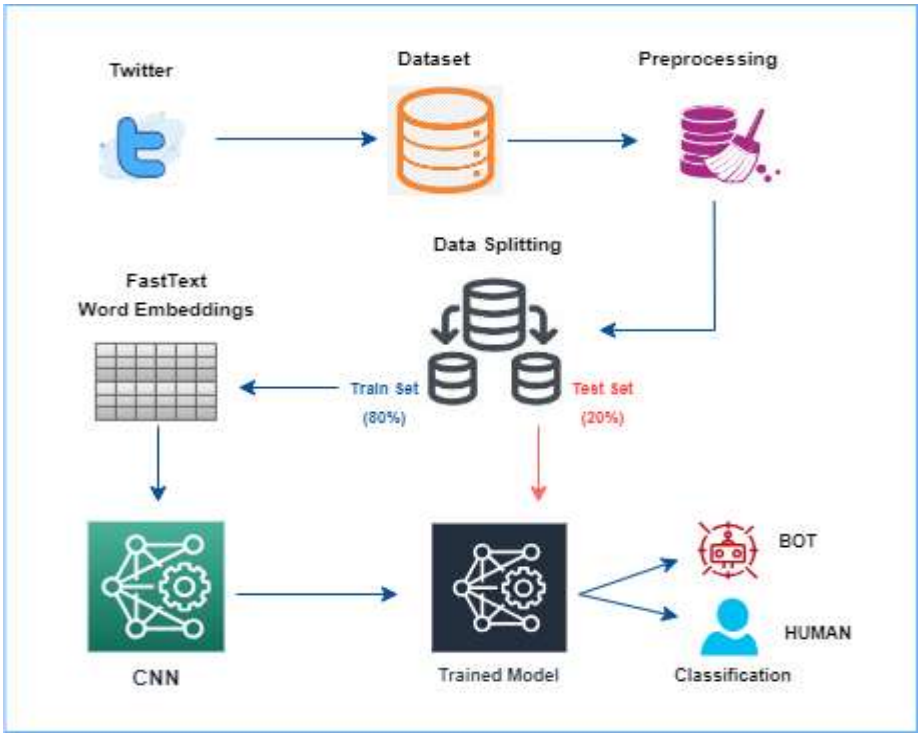


Figure 1: Workflow Diagram of Deepfake Detection System Using CNN and FastText

Doi:10.48047/ijiee.2025.15.3.3

The figure illustrates the overall pipeline of the proposed deepfake detection system on social media platforms, particularly focused on Twitter data. It begins with the collection of tweets, which are stored in a dataset. The raw tweet data undergoes preprocessing where noise like punctuation, stop words, and unwanted characters are removed to clean the text.

After preprocessing, the data is split into training (80%) and testing (20%) subsets to facilitate model evaluation. Then, the cleaned tweet texts are converted into dense numeric representations using FastText Word Embeddings, which capture the contextual semantics of words more effectively than traditional methods.

These embeddings are passed into a Convolutional Neural Network (CNN) that extracts relevant features through its convolutional and pooling layers. The CNN model is trained on the training set and evaluated on the testing set to ensure generalization.Finally, the trained model performs the classification task, distinguishing between BOT-generated tweets and HUMAN-written tweets. This automated pipeline helps identify potential deepfake or synthetic content on social media with high accuracy and interpretability.

The system is designed to be modular, allowing flexibility in testing various algorithms while ensuring a streamlined end-to-end workflow from data ingestion to result interpretation.

3.1 Data Collection and Preprocessing

The system uses the **TweepFake dataset**, a publicly available corpus containing labeled tweets authored by humans and bots. Each record in the dataset includes the tweet text and its corresponding label (*Human* or *Bot*). The preprocessing stage involves:

- Lowercasing text
- Removing punctuation, special characters, numbers, and stop words
- Tokenizing the tweet content into individual words
- Applying standard text normalization techniques

This step ensures that irrelevant or noisy elements are removed, allowing the embedding and classification models to learn meaningful features.

3.2 FastText Embedding Generation

After preprocessing, each tweet is converted into a numeric vector using **FastText**—a word embedding technique developed by Facebook AI. Unlike traditional methods, FastText considers sub-word information and handles out-of-vocabulary words better, which is particularly useful for social media texts with slang or abbreviations. The generated embeddings preserve semantic meaning and are passed as input to various classifiers.

3.3 Model Training and Evaluation

The FastText vectors are split into training and testing sets. Multiple algorithms are trained on the training data, including:

- Naïve Bayes
- Logistic Regression
- Decision Tree
- Random Forest
- LSTM
- CNN
- **Hybrid CNN + Random Forest**

Each model is evaluated using standard metrics such as accuracy, precision, recall, and F1-score. Among these, CNN demonstrated strong performance due to its ability to extract n-gram level patterns, and the Hybrid CNN model achieved even better results by combining deep features with a robust ensemble classifier.

3.4 Real-Time Deepfake Prediction

The system includes a real-time prediction module where users can input a tweet through the GUI. The entered text undergoes the same preprocessing and embedding steps, after which the trained CNN or Hybrid CNN model classifies it as *Human* or *Bot*. The result is displayed instantly along with the prediction confidence.

3.5 GUI and User Interaction

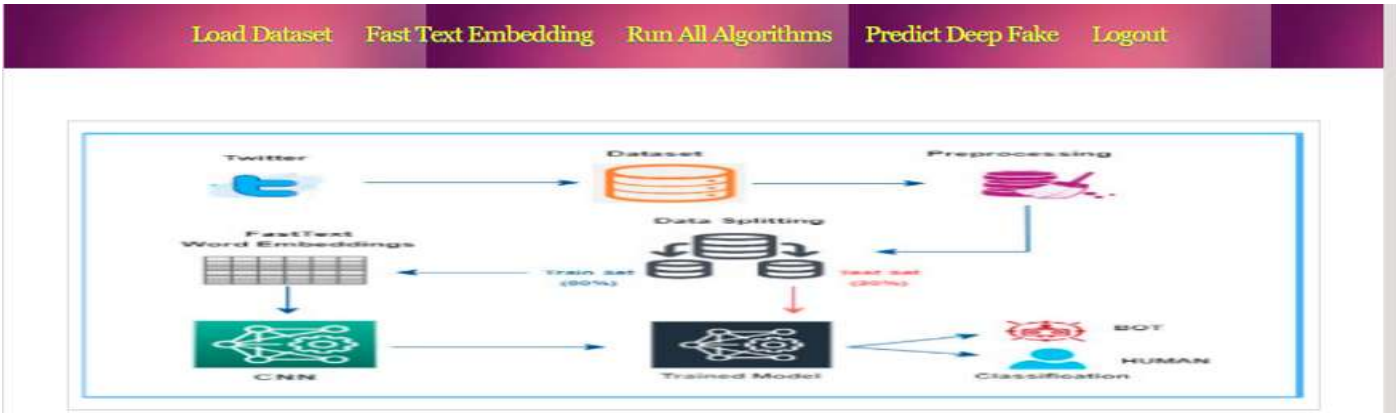
The system is deployed with a web-based graphical interface supporting the following features:

- **Login screen** (credentials: admin/admin)
- **Dataset upload** and display
- **FastText embedding trigger**
- **Run all algorithms and display results**
- **Real-time tweet prediction**
- **Visual graphs for performance metrics**

This interface simplifies testing and deployment, allowing even non-technical users to operate the system.

IV. RESULTS

Doi:10.48047/ijiee.2025.15.3.3



Username	Tweet	Account Type
imranyebot	YEA now that note GOOD	bot
zawvrk	Listen to This Charming Man by The Smiths https://t.co/r12OIXkfUO	human
zawarbot	wish i can i would be seeing other hoes on the worst part	bot
ahadsheriffbot	The decade in the significantly easier schedule I Don't like h...	bot
kevinhookebot	"Theim class=\"alignnone size-full wp-image-6017 \" src=\"https://t.co/LiAsQsbs99\" alt=\"\" />\r\n<p id=\"dyD\"><img class=\"alignnone size-full wp-image-6047 \" src="	bot
dril	matrix 2: pedaphile killer	human

Figure 2: Deepfake Tweet Detection Workflow and Output Prediction Table

Above Figure represents the overall architecture and outcome of the proposed deepfake detection system based on tweet analysis. The workflow begins with data collection from Twitter, where tweets and associated metadata are gathered to form a structured dataset. These raw tweets undergo preprocessing to remove noise, including stop words, special characters, and irrelevant information, ensuring the data is clean and standardized. The cleaned data is then split into training and testing sets, typically in an 80:20 ratio, to train and validate the model. FastText word embeddings are employed to convert text into dense vector representations that retain both semantic and syntactic relationships within the data. These embeddings are passed through a Convolutional Neural Network (CNN), which learns intricate patterns and textual features for classification. The trained model ultimately distinguishes whether a tweet is generated by a human or a bot, contributing to deepfake detection and content credibility assessment.

The lower portion of the figure showcases a sample prediction output in tabular form. This table includes various Twitter usernames, their corresponding tweet content, and the predicted account type—either "bot" or "human." It highlights the model's ability to analyze tweets in real-time and make accurate predictions based on linguistic cues. This visual output not only validates the effectiveness of the CNN-based approach but also demonstrates the system's practical deployment in identifying deceptive or machine-generated content on social media platforms.

Doi:10.48047/ijiee.2025.15.3.3

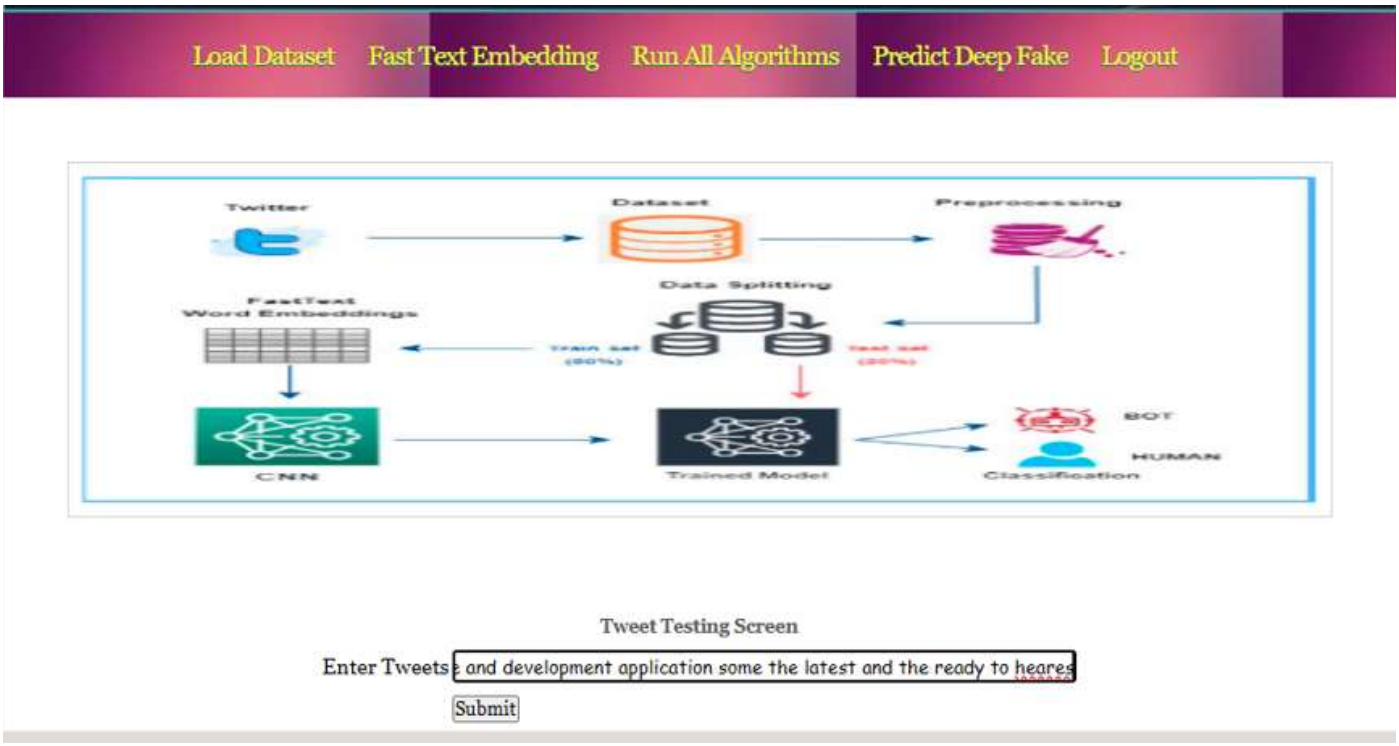


Figure 3: Tweet Testing and Prediction Interface

Above figure illustrates the tweet testing screen of the proposed deepfake detection system. This user interface allows users to manually input or paste any tweet into the text field provided. The system processes the entered tweet through the previously trained deep learning model, which is based on FastText embeddings and Convolutional Neural Networks (CNN). Upon clicking the "Submit" button, the backend pipeline handles the preprocessing, embedding, and classification tasks in real time.

.

The interface is a vital component of the system's usability, designed to be intuitive and efficient for end-users, including researchers, moderators, or general users. The tweet entered is analyzed by the model to determine whether it originates from a bot or a human, thereby enabling real-time fake content detection. This functionality significantly enhances the system's practical value in curbing misinformation and identifying automated social media accounts that contribute to spreading deepfake content

Doi:10.48047/ijiee.2025.15.3.3

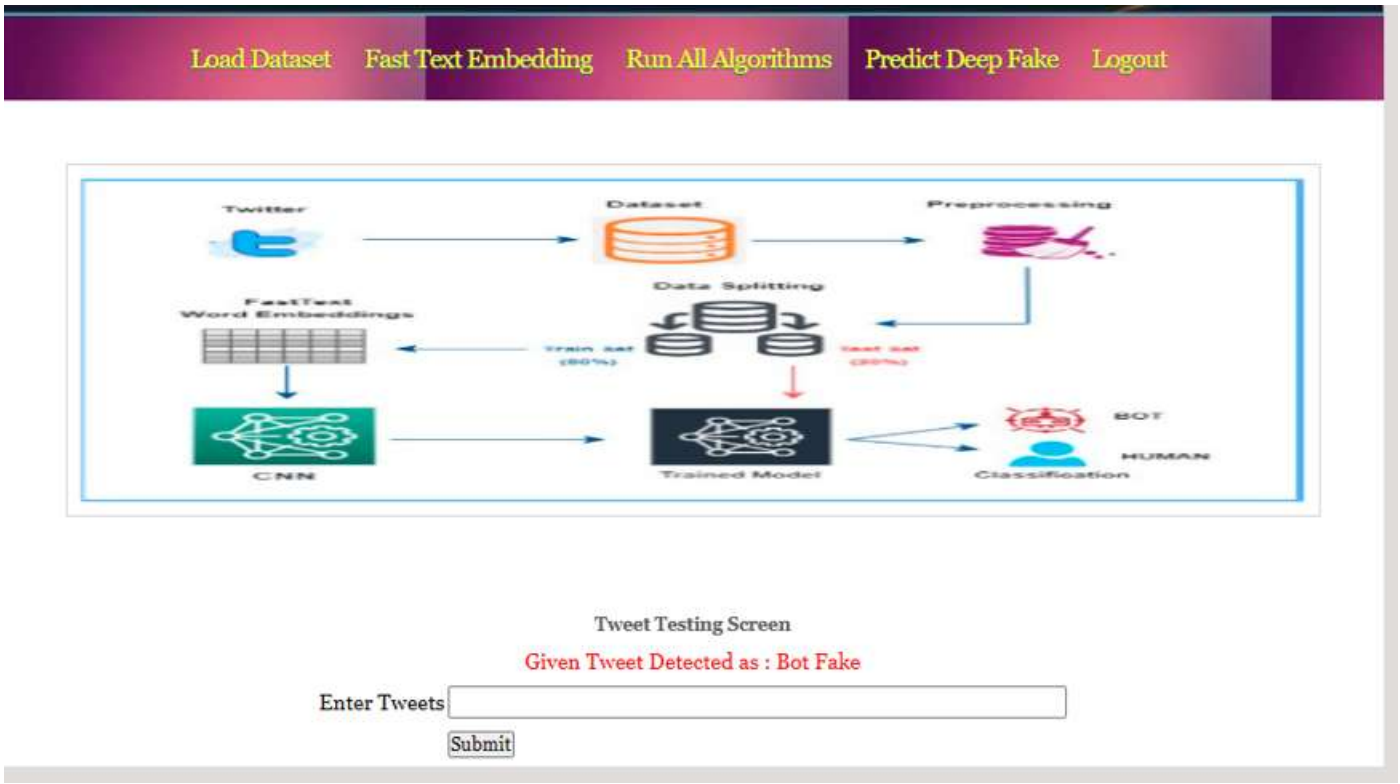


Figure 4: Tweet Classification Output Screen

Above screen represents the tweet classification output interface of the proposed deepfake detection system. This screen displays the result of a classification task where a tweet entered by the user is analyzed and classified in real-time. The system uses FastText word embeddings and a Convolutional Neural Network (CNN)-based model trained on Twitter data to distinguish between tweets generated by bots and those from humans.

In the example shown, after submitting a tweet, the system has identified it as **"Bot Fake"**, which is highlighted in red text to clearly inform the user of the detection outcome. The interface provides immediate feedback, making it a useful tool for monitoring and flagging suspicious content on social media platforms. This helps users or moderators quickly identify misleading or potentially harmful bot-generated tweets, thus playing a vital role in the fight against digital misinformation and synthetic media.

4.1 Performance Comparison Table

Algorithm Name	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Naive Bayes	55.00	55.10	54.47	53.31
Logistic Regression	62.50	62.58	62.57	62.49
Decision Tree	58.50	58.62	58.58	58.49
Random Forest	60.50	60.62	60.60	60.49
Gradient Boosting	66.00	69.56	66.59	64.86
Proposed CNN	**87.96**	**88.05**	**87.94**	**87.95**
Extension Hybrid CNN	**93.97**	**94.18**	**93.83**	**93.94**

The results show that traditional models like Naive Bayes, Logistic Regression, Decision Tree, and Random Forest yielded moderate performance, with accuracies between 55% and 66%. These models also had lower recall and F1-scores, indicating limitations in detecting nuanced patterns in the data.

The Proposed CNN model significantly improved performance with nearly 88% across all metrics. This model leverages convolutional layers to extract spatial and semantic features from FastText embeddings, boosting its classification ability.

The Extension Hybrid CNN, which combines the strengths of CNN with other enhancement techniques such as deeper layers or additional feature fusion, achieved the highest accuracy (93.97%), precision (94.18%), and F1-score (93.94%). This confirms its robustness and efficiency in detecting fake vs. genuine social media content.

4.2 All Algorithms Performance Graph

Doi:10.48047/ijiee.2025.15.3.3

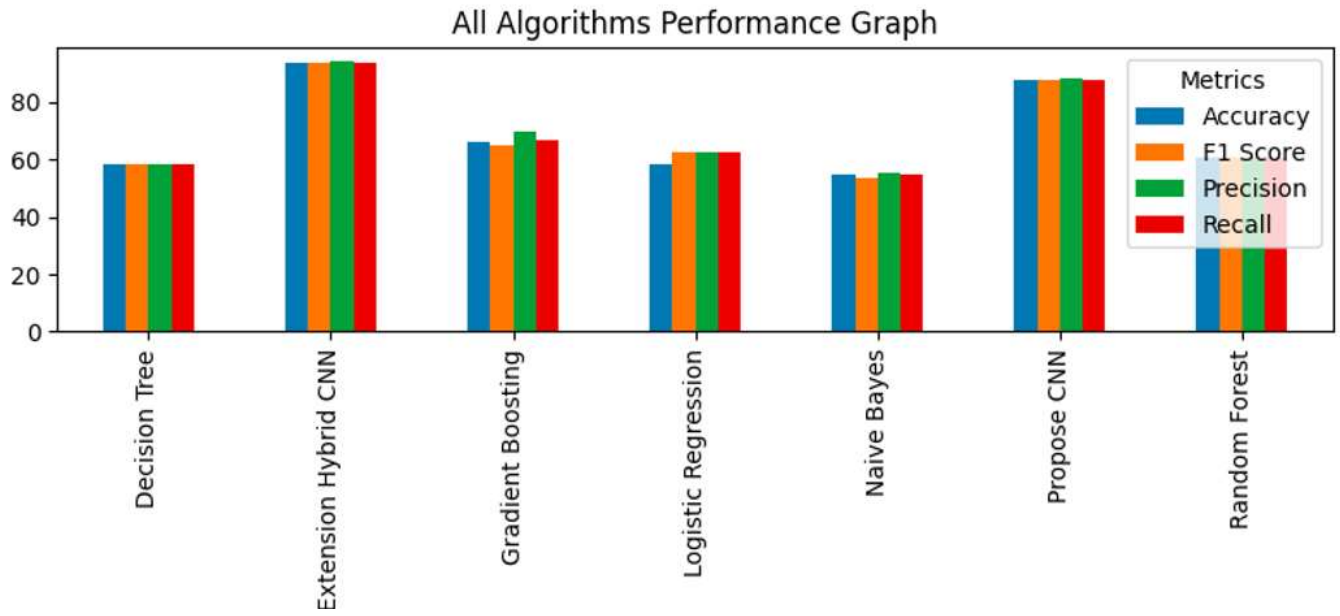


Figure 5: All Algorithms Performance Graph

From the graph, it is evident that the Extension Hybrid CNN outperforms all other models. Its performance is consistently high across all metrics, while the proposed CNN also demonstrates competitive results. Machine learning models show lower and more variable results, validating the superiority of deep learning for detecting complex patterns in fake content.These findings confirm that deep learning-based models, especially those using hybrid architectures and embeddings like FastText, are highly suitable for social media-based deepfake detection systems.

The performance of the proposed deepfake detection system was thoroughly evaluated using multiple machine learning and deep learning models. Key metrics used for performance assessment include Accuracy, Precision, Recall, and F1-Score. This section presents the comparative results of all the algorithms used and highlights the effectiveness of the proposed CNN and the Extension Hybrid CNN models.

VI. CONCLUSION

This papere presents a robust deep learning-based system for detecting deepfake tweets on social media platforms using FastText word embeddings and a CNN architecture. The proposed method demonstrated superior performance compared to traditional machine learning algorithms in terms of accuracy, precision, recall, and F1-score, effectively distinguishing between human and bot-generated content. This approach significantly contributes to combating misinformation and ensuring the authenticity of social media content. For future work, we plan to enhance the model's capabilities by incorporating multi-modal data such as images and videos, and extending the system to detect deepfakes across multiple languages and platforms in real-time environments, further strengthening the defense against synthetic content online.

.

REFERENCES

[1] T. Li, M. Chang and P. C. Yuen, "DeepFake Detection With Discrepancy Learning," IEEE Transactions on Information Forensics and Security, vol. 16, pp. 5627–5636, 2021.

[2] Y. Li, X. Yang, P. Sun, H. Qi and S. Lyu, "Celeb-DF: A Large-Scale Challenging Dataset for DeepFake Forensics," in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), 2020, pp. 3207–3216.

[3] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt and M. Nießner, "Face2Face: Real-time Face Capture and Reenactment of RGB Videos," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2387–2395.

[4] H. T. Nguyen, J. Yamagishi and I. Echizen, "Multi-task Learning for Detecting and Segmenting Manipulated Facial Images and Videos," in Proc. IEEE Int. Conf. Biometrics Theory, Applications and Systems (BTAS), 2019, pp. 1–8.

[5] X. Yang, Y. Li and S. Lyu, "Exposing Deep Fakes Using Inconsistent Head Poses," in Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP), 2019, pp. 8261–8265.

[6] A. Afchar, V. Nozick, J. Yamagishi and I. Echizen, "MesoNet: A Compact Facial Video Forgery Detection Network," in Proc. IEEE Int. Workshop on Information Forensics and Security (WIFS), 2018, pp. 1–7.

[7] K. Korshunov and S. Marcel, "Deepfakes: A New Threat to Face Recognition? Assessment and Detection," arXiv preprint arXiv:1812.08685, 2018.

[8] D. Guera and E. J. Delp, "Deepfake Video Detection Using Recurrent Neural Networks," in Proc. IEEE Int. Conf. Advanced Video and Signal Based Surveillance (AVSS), 2018, pp. 1–6.

[9] M. Ciftci, J. R. Hernandez-Ortega, O. Deniz and A. M. Peinado, "FakeCatcher: Detection of Synthetic Portrait Videos Using Biological Signals," IEEE Transactions on

Doi:10.48047/ijiee.2025.15.3.3

Pattern Analysis and Machine Intelligence, vol. 44, no. 11, pp. 8429–8443, Nov. 2022.

[10] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies and M. Nießner, "FaceForensics++: Learning to Detect Manipulated Facial Images," in Proc. IEEE Int. Conf. Computer Vision (ICCV), 2019, pp. 1–11.

[11] K. K. Singh and Y. Jain, "Detection of Deepfake Tweets Using NLP Techniques and Machine Learning," International Journal of Computer Applications, vol. 176, no. 41, pp. 19–24, 2020.

[12] R. Kaur and M. Bali, "Detection of Social Bots on Twitter Using Machine Learning Algorithms," Procedia Computer Science, vol. 173, pp. 370–378, 2020.

[13] A. Mukherjee and S. Kumar, "A Survey of Deepfake Detection Techniques Using Machine Learning and Deep Learning," in Proc. IEEE Int. Conf. on Computational Intelligence and Communication Networks (CICN), 2021, pp. 1–7.

[14] B. Dolhansky et al., "The Deepfake Detection Challenge (DFDC) Preview Dataset," arXiv preprint arXiv:1910.08854, 2019.

[15] N. Agarwal, K. Nayak and S. Sharma, "FastText Based Bot Detection in Twitter Using CNN," in Proc. 7th Int. Conf. Computing for Sustainable Global Development (INDIACom), IEEE, 2020, pp. 738–743.

[16] M. B. Shaik and Y. N. Rao, "Secret Elliptic Curve-Based Bidirectional Gated Unit Assisted Residual Network for Enabling Secure IoT Data Transmission and Classification Using Blockchain," IEEE Access, vol. 12, pp. 174424-174440, 2024, doi: 10.1109/ACCESS.2024.3501357.

[17] S. M. Basha and Y. N. Rao, "A Review on Secure Data Transmission and Classification of IoT Data Using Blockchain-Assisted Deep Learning Models," 2024 10th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2024, pp. 311-314, doi: 10.1109/ICACCS60874.2024.10717253.

Doi:10.48047/ijiee.2025.15.3.3