Archives available at journals.mriindia.com

International Journal of Recent Advances in Engineering and Technology

ISSN: 2347-2812
Volume 14 Issue 01, 2025

TweetScan: An Intelligent Framework for Deepfake Tweet Detection Using CNN and FastText

Dr. Balaji Adusumalli[1] , Lingam Nithish Kumar[2] , Nalluri Kavya[3] , Kumbha Paul Deepak[4] , Panchala Indu[5]

Professor & HOD,Department of Computer Science & Engineering ,Chalapathi Institute of Engineering and Technology, LAM, Guntur, AP, India[1]
Department of Computer Science and Engineering,Chalapathi Institute of Engineering and Technology, LAM, Guntur, AP, India [2]
Department of Computer Science and Engineering,Chalapathi Institute of Engineering and Technology, LAM, Guntur, AP, India [3]
Department of Computer Science and Engineering,Chalapathi Institute of Engineering and Technology, LAM, Guntur, AP, India [4]
Department of Computer Science and Engineering,Chalapathi Institute of Engineering and Technology, LAM, Guntur, AP, India [5]

Peer Review Information

Submission: 12 Jan 2025
Revision: 15 Feb 2025
Acceptance: 08 March 2025

Keywords

Deepfake Detection
Social Media
FastText
BOT Detection
Fake Tweet Identification

Abstract

With the widespread influence of social media platforms, the rapid dissemination of information—both authentic and deceptive—has become a major concern. One of the growing threats is the use of automated bots to spread deepfake or machine-generated tweets that mimic human-written content. These bots can manipulate public opinion, spread misinformation, and affect real-world events. To address this issue, this paper proposes a deep learning-based system for detecting deepfake tweets using FastText embeddings and various classification algorithms. A detailed performance comparison is conducted between traditional machine learning classifiers (Naïve Bayes, Logistic Regression, Decision Tree, Random Forest) and deep learning models such as CNN and LSTM. Among all models, CNN outperformed others in terms of accuracy and robustness. Furthermore, a hybrid model combining CNN feature extraction with Random Forest classification was introduced, achieving even higher performance. The system uses the publicly available "TweepFake" dataset and includes modules for dataset preprocessing, embedding, training, and real-time tweet prediction. The proposed solution effectively distinguishes between tweets authored by humans and bots, offering a valuable tool to combat social media-based misinformation.

INTRODUCTION

Social media has become a powerful medium for real-time communication, information sharing, and public discourse. Platforms such as Twitter, Facebook, and Instagram have revolutionized how people access news and express opinions. However, this rapid and largely unregulated exchange of information has also opened the door to new types of cyber threats. Among these, the proliferation of deepfake content—particularly in the form of machine-generated tweets and messages—is a rising concern. Deepfake tweets are generated by automated scripts or bots programmed to mimic human writing patterns, enabling the spread of false narratives, propaganda, or spam with deceptive authenticity.

These bots can influence public opinion, incite social unrest, or disrupt elections by making inauthentic tweets appear legitimate. Because of their increasingly sophisticated linguistic structure, it is becoming difficult for users and even standard filters to differentiate between real and fake tweets. Manual detection is both time-consuming and infeasible at scale. Hence, there is a critical need for automated systems that can intelligently analyze and classify textual content as human-generated or bot-generated.

In this paper, we propose a comprehensive deep learning-based solution for detecting deepfake tweets using FastText embeddings combined with a range of machine learning algorithms, including CNN, LSTM, Naïve Bayes, Logistic Regression, Random Forest, and Decision Tree. The system is trained and tested on the publicly available TweepFake dataset from Kaggle, which includes labeled examples of both human and bot-generated tweets. Our experiments show that Convolutional Neural Networks (CNNs) deliver the highest classification accuracy, thanks to their ability to capture n-gram level semantic features in text data.

To further enhance the model's performance, we introduce a hybrid architecture where optimized features extracted by CNN are used to train a Random Forest classifier. This hybrid model achieves superior results compared to standalone classifiers. Additionally, the system features a user-friendly interface where users can load datasets, apply FastText embedding, train multiple models, and make real-time predictions on any input tweet. The final outcome categorizes tweets as either "Human" or "Deep Bot," providing transparency in the detection process.

This work contributes a scalable, accurate, and interpretable approach to fake tweet detection, leveraging both deep learning and ensemble methods. It addresses a critical gap in social media security and opens the path for future integrations into moderation tools and anti-misinformation platforms.

RELATED WORKS

The detection of deepfake content on social media platforms has become a crucial area of research in the domains of natural language processing (NLP), cybersecurity, and digital media forensics. Various studies have been conducted to identify machine-generated text using traditional and deep learning models. This section reviews the relevant literature on fake tweet detection, bot behavior analysis, and text classification techniques.

1. Deepfake Text and Bot Detection

Early work in this field relied on rule-based systems and lexical analysis to detect anomalies in tweet structure, such as irregular timing, unnatural vocabulary, or repetitive patterns. However, with the evolution of natural language generation models, these methods became less effective. Studies have shown that bots can now emulate human-like sentence structure, grammar, and even emotional tone, making traditional detection methods obsolete.

To counter these advancements, machine learning approaches such as Support Vector Machines (SVM) and Random Forest have been employed to classify user-generated content. These models typically rely on engineered features such as word count, frequency of hashtags, and tweet intervals. While these models provide reasonable performance, their ability to scale and generalize across datasets is limited.

2. Embedding Techniques for Text Representation

Feature representation is critical for accurate text classification. Classical methods like TF (Term Frequency) and TF-IDF (Term Frequency–Inverse Document Frequency) convert raw text into numerical vectors based on word occurrence statistics. However, these methods often lose the semantic meaning and word order in the sentence.

To overcome these limitations, **FastText** and **Word2Vec** embeddings have been widely adopted. FastText, developed by Facebook AI Research, considers sub-word information and can generate embeddings for out-of-vocabulary words. This makes it particularly effective in social media contexts where slang, abbreviations, and informal grammar are common.

3. Deep Learning Models for Fake Text Classification

Recent research has demonstrated the power of deep learning models, especially Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks, for classifying fake and real tweets. CNNs are known for their capability to capture local n-gram patterns, while LSTMs are effective in modeling long-term dependencies in sequential data. Studies have shown that CNNs outperform traditional models in short-text classification due to their efficiency and ability to extract deep features from the input embeddings.

Some researchers have also proposed hybrid models that combine the strengths of different architectures. For example, CNNs are used for feature extraction, followed by decision tree-based classifiers for final prediction. These ensemble approaches have shown improvements in classification accuracy and robustness, particularly on noisy social media data.

4. 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

4.1 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.

5. 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.

5.1 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.

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.

1. System Architecture

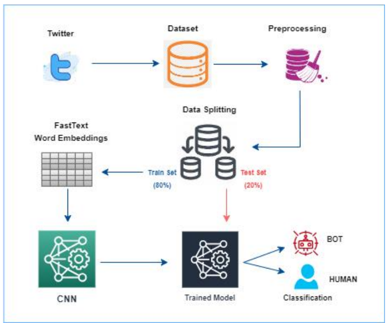


Figure 1: Workflow Diagram of Deepfake Detection System Using CNN and FastText

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.

2. 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. 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.

4. 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.

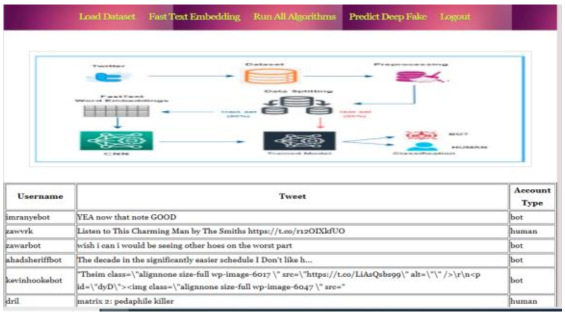
5. Real-Time Deepfake Prediction
The system includes a real-time prediction module where users can input a tweet through the GUI. The entered text undergoes the same preprocessing and embedding steps, after which the trained CNN or Hybrid CNN model classifies it as *Human* or *Bot*. The result is displayed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

6. 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.

RESULTS
This section presents the experimental results obtained by implementing and evaluating various machine learning and deep learning algorithms for deepfake detection on social media. The evaluation metrics include Accuracy, Precision, Recall, and F1-Score. The experiments were carried out using a preprocessed dataset of Twitter posts, which were split into training and testing sets in an 80:20 ratio. A range of algorithms—from traditional machine learning models to advanced deep learning architectures—were tested and compared.



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.

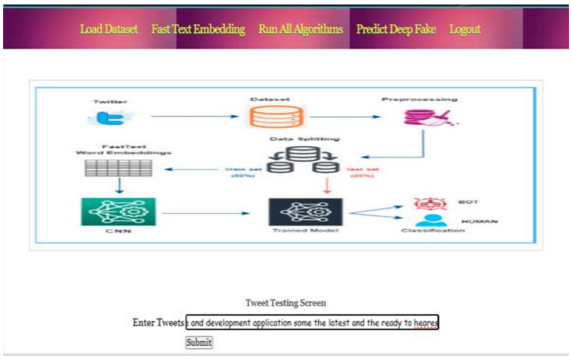


Figure 3: Tweet Testing and Prediction Interface

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.

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.

Table 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

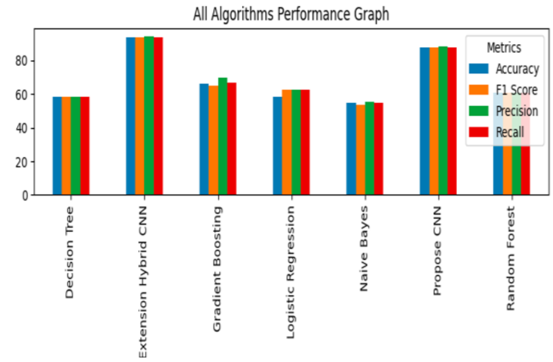


Figure 5: All Algorithms Performance Graph

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.

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

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.

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.

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.

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.

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.

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.

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

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.

M. Ciftci, J. R. Hernandez-Ortega, O. Deniz and A. M. Peinado, "FakeCatcher: Detection of Synthetic Portrait Videos Using Biological Signals," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 44, no. 11, pp. 8429–8443, Nov. 2022.

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.

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.

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

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

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.

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.

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.

Vellela, S. S., & Balamanigandan, R. (2024). An efficient attack detection and prevention approach for secure WSN mobile cloud environment. Soft Computing, 28(19), 11279-11293.

Reddy, B. V., Sk, K. B., Polanki, K., Vellela, S. S., Dalavai, L., Vuyyuru, L. R., & Kumar, K. K. (2024, February). Smarter Way to Monitor and Detect Intrusions in Cloud Infrastructure using Sensor-Driven Edge Computing. In 2024 IEEE International Conference on Computing, Power and Communication Technologies (IC2PCT) (Vol. 5, pp. 918-922). IEEE.

Sk, K. B., & Thirupurasundari, D. R. (2025, January). Patient Monitoring based on ICU Records using Hybrid TCN-LSTM Model. In 2025 International Conference on Multi-Agent Systems for Collaborative Intelligence (ICMSCI) (pp. 1800-1805). IEEE.

Dalavai, L., Purimetla, N. M., Vellela, S. S., SyamsundaraRao, T., Vuyyuru, L. R., & Kumar, K. K. (2024, December). Improving Deep Learning-Based Image Classification Through Noise Reduction and Feature Enhancement. In 2024 International Conference on Artificial Intelligence and Quantum Computation-Based Sensor Application (ICAIQSA) (pp. 1-7). IEEE.

Vellela, S. S., & Balamanigandan, R. (2023). An intelligent sleep-awake energy management system for wireless sensor network. Peer-to-Peer Networking and Applications, 16(6), 2714-2731.

Haritha, K., Vellela, S. S., Vuyyuru, L. R., Malathi, N., & Dalavai, L. (2024, December). Distributed Blockchain-SDN Models for Robust Data Security in Cloud-Integrated IoT Networks. In 2024 3rd International Conference on Automation, Computing and Renewable Systems (ICACRS) (pp. 623-629). IEEE.

Vullam, N., Roja, D., Rao, N., Vellela, S. S., Vuyyuru, L. R., & Kumar, K. K. (2023, December). An Enhancing Network Security: A Stacked Ensemble Intrusion Detection System for Effective Threat Mitigation. In 2023 3rd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA) (pp. 1314-1321). IEEE.

Vellela, S. S., & Balamanigandan, R. (2022, December). Design of Hybrid Authentication Protocol for High Secure Applications in Cloud Environments. In 2022 International Conference on Automation, Computing and Renewable Systems (ICACRS) (pp. 408-414). IEEE.

Praveen, S. P., Nakka, R., Chokka, A., Thatha, V. N., Vellela, S. S., & Sirisha, U. (2023). A novel classification approach for grape leaf disease detection based on different attention deep learning techniques. International Journal of Advanced Computer Science and Applications (IJACSA), 14(6), 2023.

Vellela, S. S., & Krishna, A. M. (2020). On Board Artificial Intelligence With Service Aggregation for Edge Computing in Industrial Applications. Journal of Critical Reviews, 7(07).

Reddy, N. V. R. S., Chitteti, C., Yesupadam, S., Desanamukula, V. S., Vellela, S. S., & Bommagani,

N. J. (2023). Enhanced speckle noise reduction in breast cancer ultrasound imagery using a hybrid deep learning model. Ingénierie des Systèmes d'Information, 28(4), 1063-1071.

Vellela, S. S., Balamanigandan, R., & Praveen, S. P. (2022). Strategic Survey on Security and Privacy Methods of Cloud Computing Environment. Journal of Next Generation Technology, 2(1).

Polasi, P. K., Vellela, S. S., Narayana, J. L., Simon, J., Kapileswar, N., Prabu, R. T., & Rashed, A. N. Z. (2024). Data rates transmission, operation performance speed and figure of merit signature for various quadurature light sources under spectral and thermal effects. Journal of Optics, 1-11.

Vellela, S. S., Rao, M. V., Mantena, S. V., Reddy, M. J., Vatambeti, R., & Rahman, S. Z. (2024). Evaluation of Tennis Teaching Effect Using Optimized DL Model with Cloud Computing System. International Journal of Modern Education and Computer Science (IJMECS), 16(2), 16-28.

Vuyyuru, L. R., Purimetla, N. R., Reddy, K. Y., Vellela, S. S., Basha, S. K., & Vatambeti, R. (2025). Advancing automated street crime detection: a drone-based system integrating CNN models and enhanced feature selection techniques. International Journal of Machine Learning and Cybernetics, 16(2), 959-981.

Vellela, S. S., Roja, D., Sowjanya, C., SK, K. B., Dalavai, L., & Kumar, K. K. (2023, September). Multi-Class Skin Diseases Classification with Color and Texture Features Using Convolution Neural Network. In 2023 6th International Conference on Contemporary Computing and Informatics (IC3I) (Vol. 6, pp. 1682-1687). IEEE.

Praveen, S. P., Vellela, S. S., & Balamanigandan, R. (2024). SmartIris ML: harnessing machine learning for enhanced multi-biometric authentication. Journal of Next Generation Technology (ISSN: 2583-021X), 4(1).

Sai Srinivas Vellela & R. Balamanigandan (2025). Designing a Dynamic News App Using Python. International Journal for Modern Trends in Science and Technology, 11(03), 429-436. https://doi.org/10.5281/zenodo.15175402

Basha, S. K., Purimetla, N. R., Roja, D., Vullam, N., Dalavai, L., & Vellela, S. S. (2023, December). A Cloud-based Auto-Scaling System for Virtual Resources to Back Ubiquitous, Mobile, Real-Time Healthcare Applications. In 2023 3rd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA) (pp. 1223-1230). IEEE.

Vellela, S. S., & Balamanigandan, R. (2024). Optimized clustering routing framework to maintain the optimal energy status in the wsn mobile cloud environment. Multimedia Tools and Applications, 83(3), 7919-7938.