

# AI-Based Deepfake Voice Detection System with Real-Time Alerts and Physical Notification

Minal Bisen

UG Student, Dept of Information Technology  
GH Raisoni College of Engineering  
Nagpur, India  
[minal.bisen.it@ghrce.raisoni.net](mailto:minal.bisen.it@ghrce.raisoni.net)

Shreyasi Jaiswal

UG Student, Dept of Information Technology  
GH Raisoni College of Engineering  
Nagpur, India  
[shreyasi.jaiswal.it@ghrce.raisoni.net](mailto:shreyasi.jaiswal.it@ghrce.raisoni.net)

Yashika Dubey

UG Student, Dept of Information Technology  
GH Raisoni College of Engineering  
Nagpur, India  
[yashika.dubey.it@ghrce.raisoni.net](mailto:yashika.dubey.it@ghrce.raisoni.net)

Gauri Ghangal

UG Student, Dept of Information Technology  
GH Raisoni College of Engineering  
Nagpur, India  
[gauri.ghangal.it@ghrce.raisoni.net](mailto:gauri.ghangal.it@ghrce.raisoni.net)

Dr. Mahendra Gaikwad

Head of Dept, Dept of Information Technology  
GH Raisoni College of Engineering  
Nagpur, India  
[mahendra.gaikward@raisoni.net](mailto:mahendra.gaikward@raisoni.net)

**Abstract** — AI-generated voices, often referred to as deepfake audio, represent an increasing risk by enabling fraud, identity theft, and the dissemination of false information. This paper introduces VoiceShield, a lightweight, real-time detection system designed to recognize AI-generated voices during live conversations. In contrast to traditional methods that depend on post-processing and cloud-based analysis, VoiceShield conducts on-device inference with minimal latency, providing users with immediate feedback. The system combines audio preprocessing, feature extraction, and machine learning models with backend communication protocols to send alerts via both a React-based user interface and an Arduino-driven physical alert mechanism. By merging software capabilities with hardware notifications, the system is crafted for straightforward usability by average users and is optimized for implementation across mobile, web, and embedded platforms. Our solution aims to provide a practical, privacy-centric cybersecurity tool that enables users to detect and address deepfake voice threats in real time.

**Keywords** – Deepfake Detection, Voice Security, AI-generated Audio, Real-time Processing, On-device Inference, Mobile Security, WebRTC, Embedded Systems, Arduino-based Alerts, Machine Learning.

## I. INTRODUCTION

The emergence of generative AI has resulted in synthetic voices becoming increasingly realistic and accessible, yet also more susceptible to misuse in areas such as fraud, identity theft, and misinformation. Current detection systems tend to rely on cloud resources, require significant processing power, or only provide analysis after calls, making them impractical for everyday users. To fill this void, we introduce VoiceShield, an efficient, real-time detection system for voice deepfakes that operates on the device itself. It integrates machine learning-based audio analysis with a user-friendly interface built on React, along with

an Arduino-based physical alert mechanism, giving users immediate, privacy-conscious feedback during live conversations. Voice Shield delivers a viable cybersecurity solution that is straightforward to use and can be easily implemented across mobile, web, and embedded platforms.

## II. METHODOLOGY

The proposed system for identifying AI-generated voices in real-time utilizes a structured method that aims to balance scientific precision with practical functionality. This method comprises six crucial stages: data collection, data preparation, feature extraction, model development, user interface integration, and output decision-making. Each step is vital for building a robust and scalable detection system designed for non-technical users. The effectiveness and reliability of the VoiceShield system depend significantly on the variety and quality of the dataset employed for training and assessment. Given that deepfake voices can differ greatly based on synthesis methods, recording settings, speaker traits, and linguistic subtleties, it is crucial to assemble a dataset that captures both academic standards and practical situations.

### Public Datasets :

To guarantee a comprehensive representation of both genuine and artificial voices, the system utilizes well-established datasets, which includes ASVspoof 2019, A widely acknowledged dataset featuring both authentic and spoofed audio samples produced through voice conversion (VC) and text-to-speech (TTS) techniques. FakeAVCeleb, A dataset that includes high-quality synthetic voices of celebrities, presenting intricate adversarial challenges for spoof detection. These datasets serve as a foundational basis for training and evaluating the deepfake detection model. **Creating a Custom Dataset:** Alongside public datasets, a custom dataset is developed to mirror real-world audio conditions, ensuring the system

continues to perform effectively in varying environments: Authentic recordings are gathered from different settings, including quiet rooms, bustling cafés, and phone conversations. Synthetic samples are created by transforming actual recordings using open-source TTS frameworks such as Tacotron2 and FastSpeech, along with proprietary APIs like ElevenLabs and Respeecher.

The dataset encompasses variations in speaker emotions, intonation, and background noise to represent the challenges users may face. This mixed dataset ensures that the training of the system incorporates both controlled benchmarks and real-world audio variations.

#### **Data Augmentation :**

To further improve generalization and robustness, various augmentation methods are utilized:

Noise addition: Gaussian noise and background sounds are combined to mimic real-world environments. • Pitch shifting: Adjustments of  $\pm 2$  semitones are made to replicate different vocal qualities. Time-stretching: The speed of audio is modified between  $0.9\times$  and  $1.1\times$  to simulate variations in speaking rates. Compression artifacts: Audio files are processed using MP3 and AMR codecs to imitate low-quality phone call sound. Reverberation effects: Impulse responses from both small and large environments are applied to recreate different acoustic settings. Standardization: All audio files are converted to a sampling rate of 16 kHz, mono channel, and 16-bit PCM WAV format to maintain consistency during preprocessing and feature extraction. By combining diverse sources, augmentation methods, and realistic adversarial scenarios, this data collection framework is fundamental to the VoiceShield system, allowing it to perform reliably across various platforms and user environments.

#### *A. Preprocessing :*

Transforming raw audio into a structured format suitable for machine learning analysis heavily relies on preprocessing. If not properly managed, issues such as inconsistent audio quality, background noise, volume variations, and silent segments can negatively impact detection accuracy. The VoiceShield system's preprocessing pipeline guarantees that audio inputs are unified and optimized for real-time inference across a variety of platforms.

Resampling: Audio recordings sourced from disparate datasets and origins frequently possess different sampling rates. To ensure consistency and maximize performance while minimizing computational burden, all audio samples are resampled to 16 kHz. This sampling frequency strikes a balance between temporal resolution and processing efficiency, making it ideal for real-time applications on both mobile and embedded platforms.

Voice Activity Detection (VAD): Non-speech areas, like silence or background noise, can unnecessarily complicate analysis and diminish the signal-to-noise ratio. Voice Activity Detection (VAD) algorithms are implemented to segment the audio and keep only the parts with active speech. By concentrating on speech segments, the system improves feature extraction and lowers computational demands, which is especially

advantageous in noisy settings or during extended conversations.

Noise Reduction: Audio inputs obtained from USB microphones or telephone conversations are often influenced by environmental sounds. The system utilizes noise reduction methods such as spectral subtraction and Wiener filtering to diminish background noise and improve the clarity of the speech signal. These methods ensure that the extracted features remain resilient even in tough acoustic environments.

Framing and Windowing: To facilitate efficient feature extraction, the continuous speech signal is partitioned into overlapping frames, each lasting 25 ms with a 10 ms overlap. A Hamming window is applied to each frame to reduce spectral leakage and maintain essential speech characteristics. These settings adhere to industry-standard practices in speech processing and provide a comprehensive temporal and spectral representation of the audio signal.

#### *B. Feature Extraction :*

Deepfake detection in our system involves identifying subtle acoustic differences between human voices and AI-generated speech in real-time. These differences are captured through carefully designed features that are continuously extracted from streaming audio.

#### **MFCCs (Mel-Frequency Cepstral Coefficients):**

MFCCs represent how the human ear interprets sound. Voices produced by AI frequently miss subtle natural changes, which can be seen in the trajectories of MFCCs. We compute 13 MFCCs together with their delta and delta-delta coefficients, yielding 39-dimensional feature vectors for each frame, continuously extracted from live audio.

**Spectral Features:** Assesses the "brightness" of the sound; synthetic voices may exhibit unnaturally uniform patterns. Monitors the reduction of high-frequency details, which is frequently restricted in speech generated by AI. Tracks the frequency of amplitude crossings in the signal; variations from natural patterns could suggest artificial speech. We assess pitch (F0), energy, rhythm, and pauses in real-time. While human voices display natural variations, voices generated by AI may reveal excessive regularity in prosody, which can be identified through these features. Live audio streams are converted into mel-spectrograms that serve as input for convolutional neural networks (CNNs). This two-dimensional depiction of frequency over time allows the model to capture unique local patterns. The spectrograms are also presented on the React interface for visual oversight. Feature vectors are normalized to have a mean of zero and a variance of one. Principal Component Analysis (PCA) minimizes redundancy while preserving more than 95% of the variance, facilitating efficient processing in real-time. The results of the detection are sent to an Arduino microcontroller, which activates physical alerts such as LEDs or buzzers, providing instant feedback in response to the detection of deepfake speech.

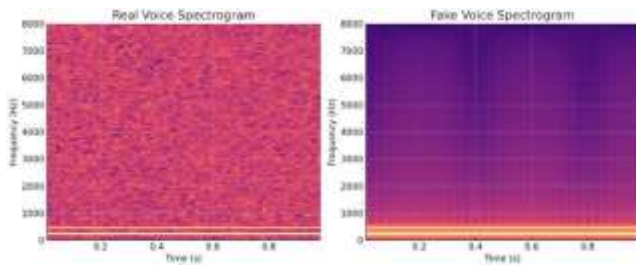


Fig. 1. Spectrograms of real and fake voices display clear acoustic differences.

### C. Model Development :

In order to successfully detect AI-generated deepfake voices, classical machine learning (ML) models and deep learning (DL) models are utilized and assessed according to their performance on extracted audio features like MFCC, pitch, and energy.

Classical Machine Learning Models are Random Forest (RF): Establishes a solid baseline with a high level of feature interpretability, Support Vector Machine (SVM): Works well in feature spaces with high dimensions, enabling precise discrimination between authentic and deepfake voices, Logistic Regression: Acts as a reference model for comparative analysis.

Deep Learning Models are Convolutional Neural Networks (CNNs): Utilized on spectrograms to identify spatial frequency-time patterns, Long Short-Term Memory (LSTM) Networks: Capture temporal relationships over frames, aiding in the detection of unnatural prosody in deepfake voices, Hybrid CNN-LSTM: Merges CNNs for extracting spatial features from spectrograms with LSTMs for temporal analysis, capitalizing on the benefits of both models.

### Training Process are

Data Split: The dataset is partitioned into 70% for training, 20% for validation, and 10% for testing, ensuring there is no overlap of speakers among the sets.

Optimizer: The Adam optimizer is employed with a learning rate of 0.001 to ensure steady convergence.

Regularization: Techniques such as dropout (0.3–0.5) and early stopping are applied to mitigate overfitting.

Batch Size: Configured to 32 and modified according to GPU memory limitations.

Evaluation Metrics: The performance of the model is evaluated by means of accuracy, precision, recall, F1-score, Equal Error Rate (EER), and ROC-AUC.

### D. User Interface Integration :

An easily navigable system is crucial for effective adoption. The initiative integrates backend AI models with a user-friendly web interface.

### Frontend (React.js):

Records live audio from a USB microphone through WebRTC, or permits the upload of audio files. Presents classification

results instantly: Genuine Voice → Green alert, Synthetic Voice → Red alert. Displays a confidence score next to color-coded indicators for enhanced interpretability. React.js, REST API (to retrieve results from Flask), Bootstrap or Tailwind.

### CSS.Backend (Flask):

Accepts audio streams or files from the frontend. Conducts real-time preprocessing, including noise elimination, trimming of silence, and extraction of features (MFCC, pitch, energy). Transmits the extracted features to the ML/DL model (RF, SVM, Logistic Regression, CNN, LSTM, CNN-LSTM) for classification. Relays the binary detection result to: React UI for software notifications, Arduino via USB Serial for hardware notifications. Utilized Tools: Flask (Python), PySerial, Librosa / PyDub, NumPy, SciPy, TensorFlow / PyTorch

### Workflow:

The user either speaks into a USB microphone or uploads an audio file. The Flask backend performs preprocessing and extracts features from the audio. The machine learning/deep learning model determines whether the voice is Real (1) or Fake (0). The React user interface updates in real-time, displaying a color-coded alert and a confidence score.

Arduino triggers **physical alert**: Real → Green LED illuminates, plays confirmation audio, and emits an optional single beep, Fake → Red LED illuminates, plays warning audio, and produces repeated or prolonged buzzer beeps.



Fig. 2. System methodology pipeline from data collection to decision output.

### E. Output Decision Layer :

The final phase of decision-making ensures outcomes that are both reliable and transparent.

Threshold-Based Classification: The model produces a probability score for its predictions. A conventional threshold of 0.5 is employed for classification. Adjusting the threshold based on the validation set aids in minimizing errors. Temporal Smoothing (Real-Time Mode): For streaming audio scenarios, predictions are made for overlapping 3-second intervals. Utilizing majority voting or averaging probabilities across these intervals helps to reduce false positives and guarantees

consistent results. User Alerts: Outcomes are communicated through simple binary labels and visual indicators (green = genuine, red = fake). To boost reliability, the system also displays a confidence score..

#### *F. Evaluation Strategy :*

The proposed AI-driven system for detecting deepfake voices is assessed through various dimensions to verify both its accuracy and real-time functionality.

Evaluation Metrics: The system's performance is measured using established classification metrics:

Accuracy – the overall correctness of the system's predictions

Precision – the ratio of voices identified as real that are genuinely real, Recall – the ratio of actual real voices that are successfully detected, F1-Score – the harmonic mean of precision and recall, ROC-AUC – the area under the receiver operating characteristic curve, Equal Error Rate (EER) – the threshold at which the false acceptance rate equals the false rejection rate.

Robustness Evaluation: The dependability of the system is assessed under difficult circumstances, encompassing. Ambient noise in live microphone input. Distortions from audio compression. Novel deepfake attacks from testing datasets that have not been encountered before

Latency Measurement: The total response time is assessed, starting from audio capture and continuing through the React UI and Arduino alerts, to ensure that real-time operation is achievable within one second.

Cross-Model Comparison: A comparative evaluation is conducted between traditional machine learning models (Random Forest, SVM, Logistic Regression) and deep learning approaches (CNN, LSTM, CNN-LSTM) to assess:

The accuracy of detection with various feature sets. Appropriateness for implementation in real time. Computational efficiency for operations requiring low latency

#### *G. Security and Ethical Considerations :*

The identification of AI-generated deepfake voices relies on sensitive audio information and carries consequences for privacy, fairness, and responsible application. The proposed system implements the following measures:

Data Privacy: Live audio recorded through the USB microphone or uploaded files is processed exclusively in memory. No audio recordings are stored on the server or transmitted to third-party services, safeguarding user privacy. Communication between the React UI and Flask backend is encrypted to prevent audio stream interception.

Bias Mitigation: The training dataset incorporates a broad range of speakers, various accents, gender and languages. This diversity helps mitigate bias and ensures consistent detection accuracy across different demographic groups.

Ethical Use: The system serves as a tool for fraud prevention and awareness, providing real-time notifications for potential deepfake voices. It is not meant to serve as conclusive forensic evidence; results should be understood within context. Users

are educated about the tool's limitations and appropriate usage through the user interface.

Responsible Real-Time Alerts (New Addition):

Physical alerts (Arduino LEDs, DFPlayer Mini audio, buzzer) are included solely to notify users, minimizing unnecessary disruption. Confidence scores and color-coded indicators on the UI offer clear feedback, enabling users to make well-informed choices.

#### **LIMITATIONS & FUTURE WORK**

The suggested AI-driven system for detecting deepfake voices in real-time demonstrates solid detection capabilities and user interaction through a combination of software (React UI) and hardware (Arduino alerts), though it does have some drawbacks. This system depends on benchmark datasets like ASVspoof and FakeAVCeleb, which may not capture the entire spectrum of new deepfake creation methods, resulting in less effective performance when faced with unfamiliar attacks or rare accents, languages, or vocal traits. While achieving sub-second real-time responsiveness is attainable on desktops or laptops, it could prove difficult on mobile or resource-constrained devices due to the high computational requirements of machine learning and deep learning models. Moreover, sporadic false positives or negatives might erode user trust, especially in critical areas such as fraud detection or security surveillance. Future developments will center on broadening the datasets to incorporate multilingual and multi-accent audio as well as novel deepfake generation techniques to enhance robustness. The research will also consider streamlined models that are fit for use on mobile and edge devices to ensure real-time performance is retained. The system may additionally include mechanisms for continuous or incremental learning to stay updated with changing deepfake strategies. Furthermore, research will delve into adversarial defense strategies and explainable AI approaches to improve both resilience and clarity. These enhancements aim to support practical uses in fields such as telecommunications and financial services, utilizing both the software interface and hardware alert systems to deliver prompt and actionable feedback.

#### **CONCLUSION :**

This research introduced a real-time solution for identifying AI-generated deepfake voices, tackling an escalating issue in cybersecurity, communication, and fraud mitigation. By integrating MFCC, spectral, and prosodic features with both traditional machine learning techniques (Random Forest, SVM, Logistic Regression) and deep learning approaches (CNN, LSTM, CNN-LSTM), the system demonstrated strong performance on recognized benchmark datasets. The combination of a React-based web interface for software notifications and an Arduino-based physical alert mechanism made the tool user-friendly and offered immediate feedback to users. Although there are limitations, such as vulnerability to new synthesis methods, real-time performance challenges on

edge devices, and the possibility of false positives or negatives, the system lays a solid groundwork for effective deepfake voice detection. Future enhancements in multilingual capabilities, lightweight deployment on edge devices, continuous learning, and resilience against adversarial attacks will improve its robustness and usability, making such systems critical for safeguarding individuals, organizations, and society against the increasing dangers of audio-related fraud and misinformation.

#### REFERENCES :

- [1] J. Yamagishi, M. Todisco, and N. Evans, "ASVspoof 2019: A large-scale public database for synthetic speech spoofing," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 3, no. 2, pp. 216–227, 2021.
- [2] A. Nautsch, X. Wang, N. Evans, T. Kinnunen, et al., "ASVspoof 2021: Automatic speaker verification spoofing and countermeasures challenge evaluation plan," *arXiv preprint arXiv:2109.00537*, 2021.
- [3] J. Patino, M. Todisco, and N. Evans, "Speaker verification spoofing countermeasures: A survey," *Speech Communication*, vol. 141, pp. 1–28, 2022.
- [4] A. K. Tiwari and A. Singh, "Deep learning approaches for detecting voice spoofing attacks: A review," *Journal of Information Security and Applications*, vol. 70, pp. 103–126, 2023.
- [5] R. K. Das, J. Yang, T. Kinnunen, Z. Wu, et al., "Long-term spoofing countermeasures: A new challenge for automatic speaker verification," *Computer Speech & Language*, vol. 77, p. 101418, 2023.
- [6] H. Wu, Y. Jia, and J. Tao, "WaveFake: A benchmark for audio deepfake detection," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.
- [7] S. Khalid, S. Javed, and M. Hussain, "A comprehensive survey on deepfake detection techniques in audio, image, and video domains," *ACM Computing Surveys*, vol. 56, no. 4, pp. 1–38, 2024.
- [8] M. Alzantot, Z. Wang, and M. Srivastava, "Deep residual neural networks for speech spoofing detection," *Proc. Interspeech*, pp. 1020–1024, 2019.
- [9] S. R. J. Ramakrishnan and R. V. Babu, "Speech deepfake detection using spectrogram and convolutional recurrent neural networks," *IEEE International Conference on Pattern Recognition (ICPR)*, 2022.
- [10] P. Korshunov and S. Marcel, "Vulnerability assessment and detection of audio deepfake attacks," *IEEE International Conference on Biometrics (ICB)*, pp. 1–8, 2018.
- [11] J. Lin, X. Zhang, and Y. Wang, "A CNN-RNN hybrid network for detecting synthesized speech," *Proceedings of the International Conference on Neural Information Processing (ICONIP)*, pp. 345–356, 2020.
- [12] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 1097–1105, 2012. (Referenced for CNN basics applied to spectrograms.)
- [13] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014. (Referenced for GRU/LSTM-based RNN design.)
- [14] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6645–6649, 2013.
- [15] H. Zhang, F. Wang, and W. Wu, "Real-time detection of deepfake voices for combating fraud calls," *IEEE Access*, vol. 11, pp. 45678–45690, 2023.
- [16] Y. Dong, R. Huang, and S. Li, "Audio forensics: Techniques for detecting synthetic speech in security-critical systems," *Journal of Cybersecurity and Privacy*, vol. 2, no. 2, pp. 112–132, 2022.
- [17] N. Kumar, P. Gupta, and V. Jain, "AI in cybersecurity: Applications, challenges and future directions," *Future Internet*, vol. 14, no. 5, pp. 120–138, 2022.
- [18] M. Westerlund, "The emergence of deepfake technology: A review," *Technology Innovation Management Review*, vol. 9, no. 11, pp. 39–52, 2019.
- [19] S. Verdoliva, "Media forensics and deepfakes: An overview," *IEEE Journal of Selected Topics in Signal Processing*, vol. 14, no. 5, pp. 910–932, 2020.
- [20] R. Subramanian and V. Raj, "Detecting deepfake voices: Challenges and countermeasures," *International Journal of Digital Crime and Forensics*, vol. 15, no. 1, pp. 24–38, 2023.
- [21] P. W. Jaronde, A. Vyas and M. Gaikwad, "A Survey on Energy Aware Cognitive Radio Network," 2022 2nd International Conference on Power Electronics & IoT Applications in Renewable Energy and its Control (PARC), Mathura, India, 2022, pp. 1–6, doi: 10.1109/PARC52418.2022.9726654.
- [22] D. Manikkule and P. Jaronde, "Encapsulation of Full Adder Using 180nm CNTFET," 2019 9th International Conference on Emerging Trends in Engineering and

Technology - Signal and Information Processing (ICETET-SIP-19), Nagpur, India, 2019, pp. 1-6, doi: 10.1109/ICETET-SIP-1946815.2019.9092017.

[23] D. Manikkule and P. Jaronde, "Design of Decoder Using Ternary Inverter," 2019 IEEE 5th International Conference for Convergence in Technology (I2CT), Bombay, India, 2019, pp. 1-3, doi: 10.1109/I2CT45611.2019.9033760.

[24] V. Patle, S. Raut, S. Tighare, N. Thakur, P. Jaronde and R. Nakhate, "Face Mask Detection Using Machine Learning," 2023 11th International Conference on Emerging Trends in Engineering & Technology - Signal and Information Processing (ICETET - SIP), Nagpur, India, 2023, pp. 1-4, doi: 10.1109/ICETET-SIP58143.2023.10151622.

[25] A. Bavistale, A. Dhokne, A. Kukade, A. Kumbhare, A. Talokar and P. Jaronde, "Energy and Spectrum Efficient Cognitive Radio Sensor Networks," 2023 11th International Conference on Emerging Trends in Engineering & Technology - Signal and Information Processing (ICETET - SIP), Nagpur, India, 2023, pp. 1-4, doi: 10.1109/ICETET-SIP58143.2023.10151496.

[26] Jaronde, P. ., Vyas, A. ., & Gaikwad, M. . (2023). Spectrum Efficient Cognitive Radio Sensor Network for IoT with Low Energy Consumption. International Journal on Recent and Innovation Trends in Computing and Communication, 11(11s), 469-479. <https://doi.org/10.17762/ijritcc.v11i11s.8176>

[27] Hirudkar, Tejshri, et al. "Emotion Detection System using Deep Learning." IRFSR, IC on Big Data, Machine Learning and IoT (2022).

[28] Dahiwal, Neha, et al. "Artificial Conversational Entity (AI Chatbot)." IJCRT 8.5 (2020).

[29] Pravin Jaronde, Suraj Rebbawar, "Artificial Intelligence Based Home Automation Using Alexa", IJACOI, Vol1, Issue2, Mar2020.

[30] Jaronde, P. W., Mangesh D. Ramteke, and Nilesh P. Bobade. "Design of Wireless Medical Monitoring System." International Journal of Engineering Research and General Science 3.2 (2015).

[31] Jaronde, P. W., Muratkar, N. A., Bhoyar, P. P., Gaikwad, S. J., & Nagrale, R. B. (2018). Review on biometric security

system for newborn baby. International Journal of Scientific Research in Science and Technology, 4(2), 907-909.

[32] Khapekar, Shreya Hansraj, et al. "AI-Driven Data Analytics Within Digital Twins: Transformative Potential and Ethical Consideration." Digital Twin Technology and AI Implementations in Future-Focused Businesses. IGI Global, 2024. 61-69.

[33] Chakraborty, Anirban, et al. "A Synergistic Integration of IoT, Machine Learning, and Flutter Technology for Precise Crop Management." International Conference on Smart Computing and Communication. Singapore: Springer Nature Singapore, 2024.

[34] Ambalkar, Prasad S., et al. "Digital Twins and AI, the Cosmic Revolution in Space Technology: Revolutionizing Space Operations, Digital Twins, and AI in Action." Harnessing AI and Digital Twin Technologies in Businesses. IGI Global, 2024. 132-142.

[35] Ghotekar, Pranoti Madhukar, et al. "Elevating Precision Medicine: Uniting Digital Twins and AI in Healthcare Web-Service Platform Design." Harnessing AI and Digital Twin Technologies in Businesses. IGI Global, 2024. 175-186.

[36] Chakraborty, Anirban Debabrata, and Pravin Jaronde. "A Case Study of Deep Learning-Based Dog Breed Classification: Paws and Pixels." Harnessing AI and Digital Twin Technologies in Businesses. IGI Global, 2024. 1-13.

[37] Jaronde, Pravin, Archana Vyas, and Mahendra Gaikwad. "Encapsulation of Energy Efficient, Clustering Algorithm and Spectrum Sensing for Cognitive Radio Based Internet of Things Networks." Journal of Electrical Systems 20.5s (2024): 2570-2578.

[38] Jaronde, Pravin, et al. "Graphic Designing, Animation and Digital Marketing Approach to Brand Development." (2023).

[39] Khapekar, S., Wankhade, S., Sawai, S., Agrawal, S., Jaronde, P. (2025). Comprehensive Vehicle Care System (CVCS). In: Kin Shun Wong, T., Kappala, V.K. (eds) 6G Communications Networking and Signal Processing. SGCNSP 2023. Lecture Notes in Electrical Engineering, vol 1270. Springer, Singapore. [https://doi.org/10.1007/978-981-97-7876-8\\_23](https://doi.org/10.1007/978-981-97-7876-8_23)