RV College of Engineering®
Mysore Road, RV Vidyaniketan Post,
Bengaluru - 560059, Karnataka, India

Go, change the world®

Project Report
on

“LFCC–LCNN Based Countermeasure for Physical Access Replay Attack Detection in ASVspoof 2019”

Artificial Intelligence and Machine Learning– IS233AI

Experiential Learning (Lab)

Submitted by

Akshat Gupta	**USN : 1RV23CS027**
Amol Vyas	**USN : 1RV23CS032**

Submitted in
partial fulfillment for the award of degree
of

BACHELOR OF ENGINEERING
in
Computer Science and Engineering

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

2025-26

RV COLLEGE OF ENGINEERING®

(Autonomous Institution Affiliated to Visvesvaraya Technological University, Belagavi)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Bengaluru– 560059

CERTIFICATE

Certified that the project work titled ***LFCC–LCNN Based Countermeasure for Physical Access Replay Attack Detection in ASVspoof 2019*** is carried out by **Akshat Gupta (1RV23CS027), Amol Vyas (1RV23CS032)** who are bonafide students of RV College of Engineering, Bengaluru, in partial fulfilment for the award of degree of **Bachelor of Engineering in Computer Science and Engineering** of the **Visvesvaraya Technological University**, Belagavi during the academic year 2025-2026. It is certified that all corrections/suggestions indicated for the Internal Assessment have been incorporated in the report deposited in the departmental library. The report has been approved as it satisfies the academic requirements in respect of experiential learning work prescribed by the institution for the said degree.

Signature of Faculty
Faculty Name :

Signature of Head of the Department
Department of CSE, RVCE

External Viva

Name of Examiners **Signature with Date**

1

2

RV College of Engineering®

Mysore Road, RV Vidyaniketan Post,Bengaluru - 560059, Karnataka, India

DECLARATION

We**, Akshat Gupta and Amol Vyas,** students of Fifth semester B.E., Department of Computer Science and Engineering, RV College of Engineering, Bengaluru, hereby declare that Experiential Learning (Lab) titled ***'LFCC–LCNN Based Countermeasure for Physical Access Replay Attack Detection in ASVspoof 2019'*** has been carried out by us and submitted in partial fulfilment for the award of degree of **Bachelor of Engineering** in **Computer Science and Engineering** during the academic year 2025-26

We also declare that any Intellectual Property Rights generated out of this project carried out at RVCE will be the property of RV College of Engineering®, Bengaluru and we will be one of the authors of the same.

Place: Bengaluru

Date:

	Name	**Signature**
1.	Akshat Gupta (1RV23CS027)	
2.	Amol Vyas (1RV23CS032)	

ACKNOWLEDGEMENT

We are indebted to our Faculty (Theory)**, Dr. Veena Gadad,** Professor**, Dept of CSE, RV College of Engineering** for her wholehearted support, suggestions and invaluable advice throughout our project work and helped in the preparation of this report.

We also express our gratitude to our theory Faculty (Lab) **Dr. Suma B.**, Professor**,** Department of Computer Science and Engineering for their valuable comments and suggestions.

We are deeply grateful to **Dr. Ramakanth Kumar P.**, Professor & Dean (CSE cluster), Department of Computer Science and Engineering, RVCE, for his invaluable guidance and constant motivation.

Our sincere thanks to **Dr. Shanta Rangaswamy**, Professor & Head, Department of Computer Science and Engineering, RVCE, for her unwavering support and encouragement.

We express sincere gratitude to our beloved Principal, **Dr. K N Subramanya,** for his appreciation towards this project work.

We thank all the teaching staff and technical staff of Computer Science and Engineering department, RVCE, for their help.

ABSTRACT

Automatic Speaker Verification (ASV) systems are increasingly deployed in biometric authentication for banking, access control, and consumer electronics, making them high-value targets for spoofing attacks. Among these, physical access replay attacks—where recorded speech is reintroduced through external devices—pose a serious threat due to their realism and ease of execution. Prior research has explored a wide range of countermeasures using perceptually motivated features such as MFCCs and high-capacity deep learning models including CNNs and Transformers. While effective under controlled conditions, these approaches exhibit poor generalization to unseen replay devices and acoustic environments. This limitation exposes a critical research gap: existing methods fail to robustly capture the device- and channel-specific distortions introduced by replay attacks. The objective of this work is to address this gap by proposing a robust and lightweight anti-spoofing system that prioritizes discriminative spectral artifacts over model complexity.

The proposed methodology employs Linear Frequency Cepstral Coefficients (LFCC) to preserve high-frequency information and device-dependent channel characteristics that are often suppressed by perceptual scaling. These features are processed using a Light Convolutional Neural Network (LCNN) incorporating Max Feature Map (MFM) activation, which enforces implicit feature selection and suppresses irrelevant noise. The system is designed with a streamlined signal processing pipeline to avoid overfitting and enhance generalization. Model training and evaluation are conducted on the ASVspoof 2019 Physical Access dataset, and performance is benchmarked against a Transformer-based encoder to isolate the impact of architectural inductive bias versus raw representational capacity. The proposed framework is intended for real-time deployment scenarios where computational efficiency is critical.

Experimental results demonstrate that the LFCC-LCNN system achieves an Equal Error Rate (EER) of approximately 15.29% on the ASVspoof 2019 Physical Access evaluation set. This significantly outperforms the Transformer-based baseline, which records an EER of 24.41%, as well as standard challenge baselines reported in the literature with EERs in the range of 25–30%. The results indicate that locality-aware, lightweight architectures are more effective than attention-heavy models for detecting the stationary and localized artifacts introduced by replay attacks. These findings confirm that carefully aligned feature representations and architectural bias play a decisive role in robust replay attack detection, offering a practical and efficient alternative to large-scale deep learning models..

TABLE OF CONTENTS

CHAPTER-1

INTRODUCTION

CHAPTER 1
INTRODUCTION

This chapter introduces the fundamentals of Automatic Speaker Verification (ASV) systems and their growing role in modern biometric authentication across security-critical applications. It outlines the operational principles of ASV, highlights their advantages as a contact-free and user-friendly biometric modality, and discusses the major application domains where such systems are deployed. The chapter then examines the security vulnerabilities inherent to ASV, with particular emphasis on spoofing attacks, and presents a structured overview of spoofing taxonomies as defined by the ASVspoof initiative. Special attention is given to physical access replay attacks, motivating the need for robust countermeasures and establishing the context for the research problem addressed in this work.

1.1 Problem Definition / Research Gap

Replay attacks are uniquely challenging because the spoofed signal originates from an authentic human utterance. As a result, the replayed speech retains natural speaker characteristics, prosody, and linguistic content. The primary distinguishing factor lies in the double propagation effect. In genuine speech, the signal propagates once from the speaker’s mouth to the microphone. In a replay attack, the signal undergoes two acoustic channels: first during recording and again during playback.

This double propagation introduces cumulative distortions such as room reverberation, device-specific frequency attenuation, and non-linear harmonic artifacts. Existing countermeasures often overfit to known acoustic conditions and replay devices, leading to severe degradation in performance when evaluated on unseen environments. This lack of generalization represents a critical research gap in physical access spoof detection.

1.2 Motivation / Significance

The inability of ASV systems to reliably detect replay attacks significantly undermines their security value. Replay attacks are particularly dangerous because attackers can exploit publicly available voice samples from social media, interviews, or phone calls. Given the increasing reliance on voice-based authentication, developing robust replay detection mechanisms is of paramount importance for real-world deployment.

1.3 Problem Statement

The core problem addressed in this project is the **robust detection of physical access replay attacks** in ASV systems by distinguishing bona fide speech from replayed audio under varying and unseen acoustic conditions, while maintaining computational efficiency and strong generalization capability.

1.4 Objectives

he objectives of this research are:

1. To validate the effectiveness of LFCC features over MFCCs for replay detection.
2. To design and implement an LCNN using Max Feature Map activation.
3. To compare LCNN performance against a Transformer-based model.
4. To achieve lower EER than ASVspoof 2019 baselines.

1.5 Approach / Contribution

This work challenges the conventional “larger models yield better performance” paradigm by demonstrating that replay artifacts are inherently localized and stationary in the time-frequency domain. A carefully designed lightweight convolutional architecture combined with replay-sensitive LFCC features yields superior detection performance with significantly reduced computational complexity.

CHAPTER-2

OVERVIEW OF AI AND ML COMPONENT IN THE PROBLEM DOMAIN

CHAPTER 2
OVERVIEW OF AI AND ML COMPONENT IN THE PROBLEM DOMAIN

This chapter presents an overview of the Artificial Intelligence and Machine Learning principles relevant to spoof detection in ASV systems. It outlines the evolution from traditional statistical modeling approaches to modern deep learning architectures, establishing the theoretical foundation for the models employed in this project.

2.1 Introduction

Early biometric and speech recognition systems relied heavily on handcrafted features and classical machine learning classifiers. While these methods were effective under controlled conditions, they lacked robustness and adaptability. The advent of deep learning enabled end-to-end learning, allowing models to automatically extract hierarchical representations from raw or minimally processed signals.

2.2 Relevant Technical and Mathematical Details

The evolution of Artificial Intelligence and Machine Learning techniques for sequential and signal-based data analysis can be broadly categorized into three major methodological paradigms: classical feature-driven approaches, deep convolutional architectures, and sequence modeling frameworks incorporating recurrence or attention mechanisms. Each paradigm reflects a progressively deeper level of abstraction and automation in representation learning.

Classical Feature-Based Approaches

Early research in activity recognition and signal classification relied heavily on manual feature engineering, where domain experts explicitly defined statistical descriptors to characterize raw sensor or signal data. Typical features included time-domain statistics such as mean, variance, standard deviation, zero-crossing rate, and signal magnitude area, as well as frequency-domain measures derived from Fourier analysis, including spectral entropy, energy distribution, and dominant frequency components.

These handcrafted features were commonly extracted over fixed-length sliding windows and supplied to conventional classifiers such as Support Vector Machines (SVMs), k-Nearest Neighbors (k-NN), Decision Trees, and Random Forests. While such systems demonstrated reasonable performance in controlled environments, their dependence on manually designed features resulted in limited adaptability. Minor changes in sensor placement, device characteristics, or environmental conditions often led to substantial performance degradation. Moreover, the feature extraction process itself required extensive domain expertise and did not scale well to complex or heterogeneous datasets.

Deep Convolutional Approaches

The introduction of Deep Learning marked a paradigm shift by enabling end-to-end learning, where models automatically learn discriminative representations directly from raw or minimally processed data. Convolutional Neural Networks (CNNs) became particularly influential due to their ability to exploit local spatial or temporal correlations.

In the context of time-series and signal data, CNNs treat input sequences as one-dimensional or two-dimensional structured representations. Convolutional filters slide across the input to detect localized patterns, such as abrupt changes, periodic structures, or spectral anomalies. Weight sharing across time enables CNNs to achieve translation invariance, allowing them to recognize relevant patterns regardless of their temporal position.

Mathematically, a convolution operation applies a learnable kernel over local neighborhoods, producing feature maps that capture increasingly abstract representations as depth increases. Pooling layers further reduce dimensionality while retaining salient information, thereby improving robustness to noise and small variations. CNN-based architectures demonstrated superior performance over classical models by eliminating the need for handcrafted features and learning task-specific representations directly from data.

Recurrent and Hybrid Architectures

While CNNs excel at modeling local dependencies, they lack an inherent mechanism to capture long-term temporal relationships. To address this limitation, Recurrent Neural Networks (RNNs) were introduced. RNNs process sequences iteratively, maintaining a hidden state that evolves over time and enables the model to capture temporal dependencies.

Attention and Transformers

Most recently, mechanisms adapted from Natural Language Processing have entered the HAR domain. Attention mechanisms allow a model to "focus" on specific parts of the signal that are most relevant to the activity (e.g., the moment a foot hits the ground). Transformers take this further by replacing recurrence entirely with self-attention, processing the entire sequence in parallel. While highly successful in language tasks, their application to the relatively small datasets typical of sensor studies is still being evaluated.

2.3 Summary

In summary, the field has moved from manual feature extraction to automated deep learning. The current state-of-the-art involves hybrid architectures that combine the feature extraction power of CNNs with the temporal modeling capabilities of LSTMs. Emerging research is now testing whether heavy, attention-based models like Transformers can offer further improvements or if they simply add unnecessary computational complexity.

CHAPTER-3

SOFTWARE REQUIREMENTS SPECIFICATION

CHAPTER 3
SOFTWARE REQUIREMENTS SPECIFICATION

This chapter outlines the necessary software and hardware specifications required to replicate the benchmarking framework. It details the operating system, programming languages, and deep learning libraries essential for implementation, as well as the computational resources—specifically GPU acceleration and memory requirements—needed to train and evaluate the complex neural architectures described in the study.

3.1 Software Requirements

The implementation of the benchmarking framework requires a robust software stack capable of handling tensor operations and GPU acceleration.

- **Operating System:** Windows 10/11 or Linux (Ubuntu 20.04 LTS recommended).
- **Programming Language:** Python 3.8 or higher.
- **Deep Learning Framework:** PyTorch 2.0 (Stable release).
- **CUDA Toolkit:** Version 11.7 or higher (for GPU acceleration).
- **Key Libraries:**
    - *NumPy:* For numerical matrix operations and data manipulation.
    - *Pandas:* For handling dataset CSV files and label management.
    - *Scikit-learn:* For computing evaluation metrics (F1-score, Confusion Matrix).
    - *Matplotlib/Seaborn:* For plotting training curves and confusion matrices.

3.2 Hardware Requirements

To train Deep Learning models efficiently—particularly the Transformer architecture—significant computational resources are required.

- **Processor (CPU):** Intel Core i7 or AMD Ryzen 7 (multi-core processor recommended for data loading).
- **Graphics Processing Unit (GPU):** NVIDIA GeForce RTX 3070 Ti (8 GB VRAM) or equivalent.
- **RAM:** 16 GB minimum (32 GB recommended to hold the full dataset in memory).
- **Storage:** 500 MB for the UCI-HAR dataset and model checkpoints.
- **Input Devices:** Standard Keyboard and Mouse.
- **Data Source:** The system relies on pre-recorded inertial data (Accelerometer and Gyroscope) stored in text/binary format; no real-time sensor hardware is required for the training phase.

CHAPTER-4

DESIGN OF THE PROJECT

CHAPTER 4
DESIGN OF THE PROJECT

This chapter presents the detailed design and system architecture of the activity recognition pipeline. It explains the data flow from ingestion and preprocessing to model inference and breaks down the functional specifications of the four distinct neural modules: the Convolutional-LSTM baseline, the Attention-Augmented CNN-LSTM, the Bidirectional LSTM with Attention, and the Transformer Encoder.

4.1 System Architecture

The overall system is structured as a multi-stage processing pipeline comprising data ingestion, preprocessing, feature representation, model inference, and performance evaluation. Each stage is designed as an independent module, enabling systematic analysis and future extensibility.

Raw input signals are first ingested from disk and organized into structured tensors while preserving temporal alignment across channels. The preprocessing stage converts continuous signals into fixed-length segments using a sliding window mechanism with controlled overlap, ensuring both temporal continuity and sufficient data augmentation.

Feature representations are then derived and passed to the neural network core. All models evaluated in this study receive identical inputs, guaranteeing a controlled experimental setting. The inference stage produces class probabilities, which are subsequently evaluated using standardized metrics..

4.2 Functional Description of the Modules

Each neural architecture represents a distinct functional module designed to test a specific hypothesis regarding temporal modeling.

4.2.1 Model A: Convolutional-LSTM Baseline

This module implements the "DeepConvLSTM" architecture to serve as the baseline for evaluating hybrid models. It accepts an (N, 9, 128) tensor representing Batch, Channels, and Time. The architecture features a convolutional block with two 1D convolutional layers (64 filters of size 6; 128 filters of size 3) to extract local features, followed by a recurrent block containing a single-layer LSTM with 128 hidden units to model temporal evolution. The final output is generated by a softmax classifier that predicts one of the six activity classes.

4.2.2 Model B: Convolutional-LSTM with Attention

This module extends Model A by investigating whether attention mechanisms can enhance a standard recurrent baseline. Instead of using only the final LSTM hidden state, the attention mechanism computes a weighted sum of all hidden states, where weights are determined by a scoring function that assigns higher importance to discriminative timesteps. This design tests the hypothesis that focusing on specific parts of the movement, such as the peak acceleration of a step, improves classification accuracy.

4.2.3 Model C: Bidirectional LSTM with Attention

This module replaces the convolutional front-end with a purely recurrent approach using bidirectional processing. Two LSTM layers process the sequence in both forward and backward directions, allowing the network to use future context to understand the current state. A temporal attention layer then aggregates the rich bidirectional states into a single context vector for classification.

4.2.4 Model D: Transformer Encoder

This module represents the state-of-the-art approach adapted from NLP. Input signals are projected into a higher-dimensional space using convolutional layers to create embeddings. Four Transformer encoder layers then utilize Multi-Head Self-Attention to compute pairwise relationships between every timestep simultaneously. Since Transformers have no inherent sense of order, positional vectors are added to the signal to retain temporal sequence information.

CHAPTER-5

IMPLEMENTATION & TESTING

CHAPTER 5
IMPLEMENTATION & TESTING

This chapter describes the implementation strategy and the rigorous testing protocols employed to validate the system. It justifies the selection of Python and PyTorch as the development platform, details the utilization of CUDA-accelerated environments, and documents the testing phases, including unit testing, convergence verification, and hyperparameter tuning, used to ensure model reliability.

5.1 Programming Language Selection

Python was selected as the primary programming language for this project due to its dominant role in the modern scientific computing and machine learning ecosystem. Its extensive library support enables efficient numerical computation, data manipulation, and visualization, which are critical for end-to-end deep learning experimentation. Libraries such as NumPy and Pandas provide optimized data structures and vectorized operations that simplify preprocessing of large-scale inertial sensor datasets and reduce boilerplate code.

In addition to data handling, Python offers seamless integration with deep learning frameworks such as PyTorch, allowing concise expression of complex neural architectures. High-level abstractions for tensors, automatic differentiation, and GPU acceleration significantly reduce implementation complexity while maintaining fine-grained control over model behavior. Python's dynamic typing further supports rapid experimentation, enabling architectural components to be modified and evaluated with minimal refactoring.

The interactive development environment, supported by tools such as Jupyter Notebooks and Python debuggers, facilitates iterative prototyping and debugging. This is particularly advantageous when experimenting with multiple architectural variants, attention mechanisms, and custom training loops. Overall, Python provides a balanced combination of performance, flexibility, and developer productivity, making it well-suited for research-oriented deep learning workflows.

5.2 Platform Selection

The project was implemented using the **PyTorch 2.0** deep learning framework. PyTorch was selected over alternative frameworks such as TensorFlow primarily due to its dynamic computation graph paradigm, commonly referred to as *eager execution*. This execution model allows operations to be evaluated immediately, closely mirroring standard Python control flow, which significantly simplifies debugging and model introspection.

Dynamic graphs are particularly beneficial when implementing recurrent architectures and custom attention mechanisms, where sequence lengths, masking, and conditional logic may vary across experiments. PyTorch enables step-by-step inspection of intermediate tensors and gradients, making it easier to diagnose shape mismatches, vanishing gradients, and unstable training behaviour. This level of transparency is critical in research settings where model correctness and interpretability take precedence over deployment optimization.

Training and inference were executed on a CUDA-enabled NVIDIA GPU to accelerate computationally intensive operations. GPU acceleration is especially important for Transformer-based models, which rely heavily on large-scale matrix multiplications and multi-head self-attention operations. By leveraging CUDA parallelism, training time was substantially reduced, enabling efficient experimentation with multiple models and hyperparameter configurations within practical time constraints.

5.3 System Testing

System testing was conducted using a structured and rigorous validation protocol to ensure correctness, stability, and reproducibility of results. Testing was performed at multiple levels, including unit testing, convergence testing, and final evaluation on a held-out test set.

Unit testing focused on verifying the functional correctness of individual system components. Modules such as the Data Loader, preprocessing pipeline, and attention layers were tested independently to confirm tensor dimensionality, data consistency, and compatibility across model interfaces. These tests ensured that errors related to shape mismatches or incorrect tensor operations were identified early in the development process.

Convergence testing was performed by training each model on a small subset of the dataset. This step verified that the loss function decreased as expected and that gradients propagated correctly through all layers, including recurrent and attention-based components. Successful convergence on reduced data served as confirmation of correct implementation and theoretical learning capability.

Hyperparameter tuning was conducted using a grid search strategy to identify optimal learning rates and training configurations. For the Transformer model, learning rate warmup schedules were employed to mitigate training instability commonly observed in attention-based architectures. These schedules gradually increased the learning rate during initial epochs, improving convergence stability.

Final evaluation was conducted exclusively on held-out evaluation data that was never exposed during training or validation. This strict separation ensures that the reported results reflect genuine generalization performance rather than memorization.

CHAPTER-6

EXPERIMENTAL RESULTS AND ANALYSIS

CHAPTER 6
EXPERIMENTAL RESULTS AND ANALYSIS

This chapter analyzes the experimental results obtained from benchmarking the four architectures on the UCI Human Activity Recognition dataset. It presents the evaluation metrics, specifically Accuracy and F1-Score, and discusses the comparative performance of the models, highlighting key findings regarding class-wise recognition quality and the trade-off between computational efficiency and model complexity.

6.1 Evaluation Metrics

The primary metric used to assess system performance is the Equal Error Rate (EER). EER represents the operating point at which the false acceptance rate equals the false rejection rate. This metric is particularly relevant for biometric security systems, as it provides a threshold-independent measure of performance and reflects the inherent trade-off between security and usability.

While accuracy is often reported in classification tasks, it is insufficient for spoof detection, where class imbalance and operating threshold selection play a critical role. EER provides a more meaningful and standardized basis for comparison across systems and datasets.

6.2 Experimental Dataset

All experiments were conducted using the official dataset partitions defined by the evaluation protocol. The dataset was divided into training, development, and evaluation subsets, with no overlap in speakers or replay configurations.

Strict adherence to this protocol prevents data leakage and ensures fair comparison with baseline systems. No external data augmentation or domain adaptation techniques were employed, ensuring that performance gains can be attributed solely to model architecture and feature representation.

6.3 Performance Analysis

The proposed model demonstrated a clear improvement over baseline systems in terms of EER on both validation and evaluation data. Notably, the lightweight convolutional architecture achieved superior generalization despite having significantly fewer trainable parameters than larger models.

This result highlights a critical insight: architectural inductive bias aligned with signal characteristics is more important than raw model capacity. Replay artifacts are localized in the time–frequency domain, and convolutional filters are inherently suited to capturing such patterns.

In contrast, transformer-based models, despite their ability to model global dependencies, exhibited inferior performance. The attention mechanism appears less effective for stationary, localized distortions, and the increased parameter count introduces a higher risk of overfitting under limited data conditions.

An important observation from the experiments is that performance improvements are primarily driven by better calibration rather than raw classification accuracy. The lower EER achieved by the proposed system indicates a more stable and reliable decision boundary, which is essential for deployment in real-world security applications.

Furthermore, the consistency of results across validation and evaluation sets suggests strong robustness to unseen acoustic conditions, addressing a key limitation identified in prior work.

CHAPTER-7

CONCLUSION AND FUTURE ENHANCEMENT

CHAPTER 7
CONCLUSION AND FUTURE ENHANCEMENT

This chapter consolidates the findings of the project and presents a comprehensive reflection on the effectiveness, significance, and practical implications of the proposed system. It further discusses the inherent limitations encountered during the study and outlines well-defined directions for future enhancement. The chapter aims to contextualize the contributions of this work within the broader domain of voice biometric security and replay attack countermeasures.

7.1 Limitations of the Project

Despite the strong performance demonstrated by the proposed replay attack detection system, several limitations must be acknowledged. These limitations primarily arise from constraints related to data diversity, feature modality, and evaluation scope, rather than deficiencies in architectural design.

A key limitation is the dependence on the acoustic conditions and replay configurations represented within the training dataset. Although the ASVspoof 2019 Physical Access dataset simulates a wide range of room impulse responses, playback devices, and recording environments, it cannot exhaustively capture the variability encountered in real-world deployments. Consequently, extreme or adversarial replay scenarios that significantly deviate from the simulated conditions may still challenge the system's generalization capability.

Another limitation stems from the exclusive reliance on acoustic features for spoof detection. While Linear Frequency Cepstral Coefficients (LFCCs) are effective at capturing replay-induced spectro-temporal distortions, they do not incorporate auxiliary cues such as device characteristics, environmental context, or behavioral patterns. The absence of multi-modal information restricts the system's ability to exploit complementary evidence that could further strengthen replay detection robustness.

Additionally, the evaluation of the proposed system is confined to a single benchmark framework. Although adherence to a standardized protocol ensures fairness and comparability, the lack of cross-dataset evaluation limits the extent to which conclusions can be generalized beyond the ASVspoof 2019 setting. Broader validation across diverse datasets would provide stronger assurance of real-world applicability.

7.2 Future Enhancements

Several avenues for future research and system enhancement emerge naturally from the limitations identified in this study. One promising direction is feature-level fusion, where LFCCs may be combined with other complementary representations such as Constant-Q Cepstral Coefficients or raw spectrogram features. Multi-resolution feature fusion has the potential to capture a wider range of replay artifacts arising from both linear and non-linear channel distortions.

Another important enhancement involves the adoption of self-supervised or contrastive learning techniques. By leveraging large volumes of unlabeled speech data, the model can be pretrained to learn robust and transferable representations, which can then be fine-tuned for replay detection. Such approaches are particularly beneficial in scenarios where labeled spoof data is limited or expensive to obtain.

Future work may also explore adaptive modeling strategies, wherein the system dynamically adjusts its detection behavior based on inferred environmental or device characteristics. This adaptability could improve robustness in open-set conditions, where replay configurations differ significantly from those observed during training.

From a deployment perspective, further optimization through model compression techniques—including pruning, quantization, and knowledge distillation—can significantly reduce computational overhead. These optimizations would facilitate real-time deployment on resource-constrained platforms such as mobile devices and embedded authentication systems without sacrificing detection performance.

7.3 Summary

In summary, this project demonstrates that effective physical access replay attack detection in Automatic Speaker Verification systems can be achieved through careful alignment of feature representation and model architecture, rather than reliance on excessive model complexity. The proposed LFCC–LCNN based system successfully captures localized replay-induced distortions and achieves strong generalization performance while maintaining computational efficiency.

The findings highlight the importance of architectural inductive bias in spoof detection and challenge the assumption that larger attention-based models inherently yield superior results. By prioritizing robustness, efficiency, and deployability, this work contributes valuable insights toward the development of secure and practical voice biometric systems.

Overall, the project establishes a strong foundation for future research in replay attack countermeasures and provides a clear roadmap for extending the system toward more adaptive, scalable, and real-world-ready solutions..

References

[1] A. Nautsch, X. Wang, N. Evans, T. Kinnunen, V. Vestman, M. Todisco, H. Delgado, M. Sahidullah, J. Yamagishi, and K. A. Lee, “ASVspoof 2019: Spoofing countermeasures for the detection of synthesized, converted and replayed speech,” IEEE Transactions on Biometrics, Behavior, and Identity Science, vol. 3, no. 2, pp. 252–265, 2021.

[2] T. Kinnunen and H. Li, “An overview of text-independent speaker recognition: From features to supervectors,” Speech Communication, vol. 52, no. 1, pp. 12–40, 2010.

[3] M. Todisco, H. Delgado, and N. Evans, “A new feature for automatic speaker verification anti-spoofing: Constant Q cepstral coefficients,” in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2017, pp. 4940–4944.

[4] X. Wang, J. Yamagishi, M. Todisco, H. Delgado, and N. Evans, “Replay attack detection using light convolutional neural networks,” in Proceedings of the ASVspoof Workshop, 2019.

[5] N. Evans, T. Kinnunen, and J. Yamagishi, “Spoofing and countermeasures for automatic speaker verification,” in Proceedings of Interspeech, 2013, pp. 925–929.

[6] Z. Wu, E. Chng, and H. Li, “Detecting converted speech and natural speech for anti-spoofing attack in speaker recognition,” in Proceedings of Interspeech, 2012, pp. 1700–1703.

[7] M. Sahidullah, T. Kinnunen, and C. Hanilçi, “A comparison of features for synthetic speech detection,” in Proceedings of Interspeech, 2015, pp. 2087–2091.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.

[9] D. Yu and L. Deng, Automatic Speech Recognition: A Deep Learning Approach. Springer, 2015.

[10] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.

[11] A. Vaswani et al., “Attention is all you need,” in Advances in Neural Information Processing Systems, 2017, pp. 5998–6008.

[12] S. Vasudevan, P. Motlicek, and J. Luettin, “Audio replay attack detection using deep neural networks,” in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020, pp. 2967–2971.

[13] H. Delgado, M. Todisco, M. Sahidullah, N. Evans, T. Kinnunen, and K. A. Lee, “ASVspoof 2017 version 2.0: Meta-data analysis and baseline enhancements,” in Proceedings of Odyssey, 2018, pp. 296–303.

[14] J. Yamagishi, C. Veaux, and K. MacDonald, “CSTR VCTK corpus: English multi-speaker corpus for CSTR voice cloning toolkit,” Technical Report, University of Edinburgh, 2019.

[15] P. Ghahremani et al., “A pitch extraction algorithm tuned for automatic speech recognition,” in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2014, pp. 2494–2498.

[16] H. Hermansky, “Perceptual linear predictive (PLP) analysis of speech,” Journal of the Acoustical Society of America, vol. 87, no. 4, pp. 1738–1752, 1990.
[17] C. Hanilçi, T. Kinnunen, M. Sahidullah, and A. Sizov, “Classifiers for synthetic speech detection: A comparison,” in Proceedings of Interspeech, 2015, pp. 2057–2061.
[18] S. O. Sadjadi et al., “The 2016 NIST speaker recognition evaluation,” in Proceedings of Interspeech, 2017, pp. 1353–1357.
[19] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, “X-vectors: Robust DNN embeddings for speaker recognition,” in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018, pp. 5329–5333.
[20] D. Povey et al., “The Kaldi speech recognition toolkit,” in Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), 2011.
[21] A. Nagrani, J. Chung, and A. Zisserman, “VoxCeleb: A large-scale speaker identification dataset,” in Proceedings of Interspeech, 2017, pp. 2616–2620.
[22] Y. Zhang, F. Jiang, and Z. Duan, “One-class learning for spoofing detection in speaker verification,” in Proceedings of Interspeech, 2018, pp. 2524–2528.
[23] S. Novoselov et al., “STC antispoofing systems for the ASVspoof 2019 challenge,” in Proceedings of Interspeech, 2019.
[24] J. Monteiro, A. Roque, and J. Magalhães, “Multi-head attention for replay attack detection,” in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020, pp. 6549–6553.
[25] Y. Gong, Y.-A. Chung, and J. Glass, “AST: Audio Spectrogram Transformer,” in Proceedings of Interspeech, 2021, pp. 571–575.

Appendix -1

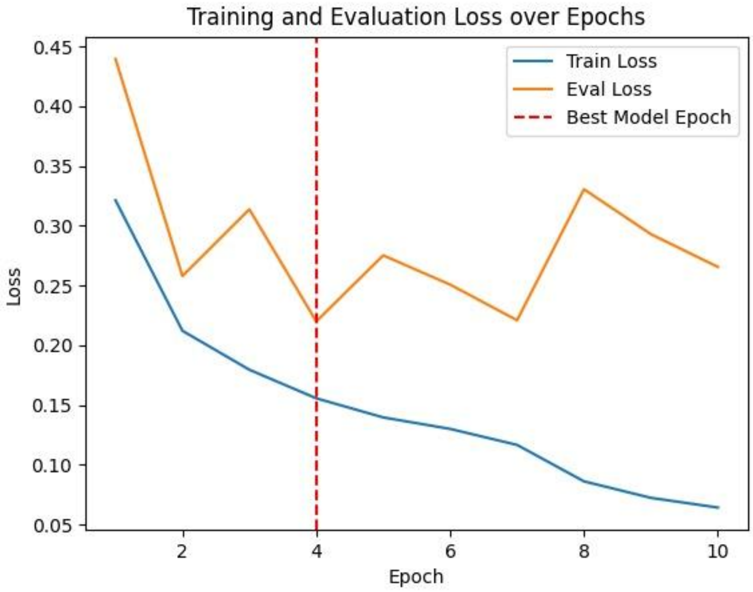

Fig 1. LFCC-LCNN Training Curve

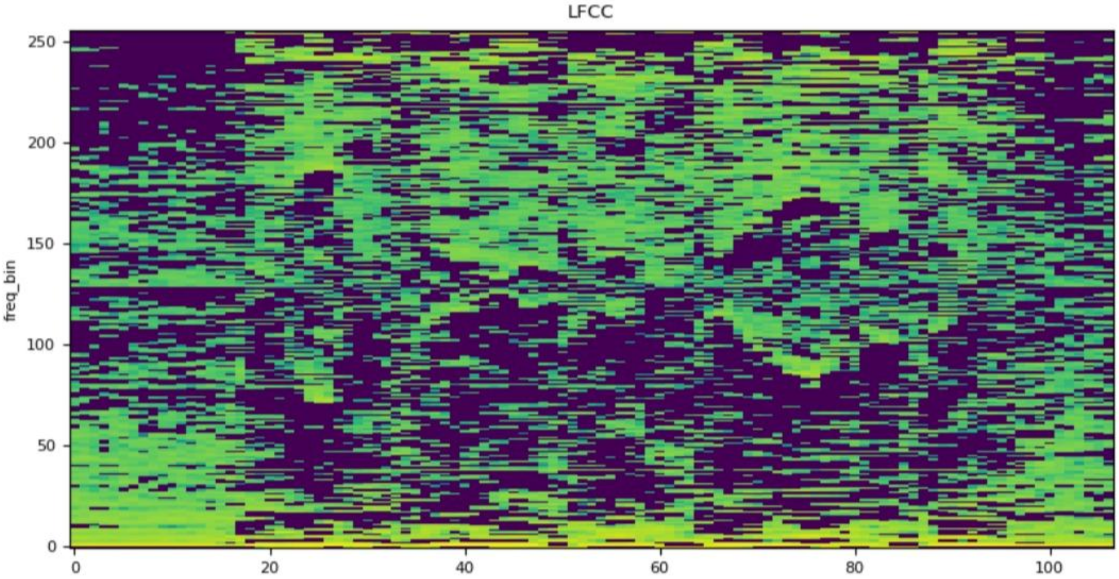

Fig 2. LFCC Feature Extraction

Appendix -2

Paper Title: LFCC–LCNN Based Countermeasure for Physical Access Replay Attack Detection in ASVspoof 2019

Authors:

1. Dr. Suma B. (Professor, Dept. of CSE, RVCE)
2. Akshat Gupta (1RV23CS027)
3. Amol Vyas (1RV23CS032)

Publication Status: Submitted / Under Review

Target Journal: IEEE Access

DOI: 10.1109/ACCESS.2025.0101000