AUDIO FEATURES BASED SCREAM RECOGNITION AND CLASSIFICATION SYSTEM

A MINOR PROJECT REPORT

SUBMITTED FOR THE PARTIAL FULFILLMENT OF AWARD OF THE DEGREE OF

Bachelor of Technology in Computer Science and Engineering

By

KEDAR KUMAR DORA 21BTCSE31

Under the Guidance

of

DR. KALYAN DAS

Department of Computer Science Engineering and Application

SAMBALPUR UNIVERSITY INSTITUTE OF INFORMATION TECHNOLOGY

Jyoti Vihar, Burla, Odisha- 768019

December, 2024

CERTIFICATE

This is to certify that the minor project entitled **“Audio Features Based Scream Recognition and Classification System”** has been submitted by **Kedar Kumar Dora** bearing **21BTCSE31,** in partial fulfillment of the requirements for the award of the Degree of **Bachelor of Technology** in Computer Science and Engineering. This record of bonafide work carried out by him under my guidance and supervision. The result embodied in this project report has not been submitted to any other university or institute for the award of any degree.

Dr. Kalyan Das	Dr. Kalyan Das	Dr. Kalyan Das
Asst. Professor & Guide	Faculty in-charge, B.Tech	Assistant Professor & HOD
Dept. of CSE	Dept. of CSE	Dept. of CSE

DECLARATION

I do hereby declare that the work embodied in this minor project report entitled "Audio Classification Model for Scream Detection" is the outcome of genuine work carried out by me under the direct supervision of Dr. Kalyan Das, Assistant Professor, Department of Computer Science Engineering and Application is submitted by me to Sambalpur University Institute of Information Technology, Burla for the award of the degree of Bachelor of Technology. The work is original and has not been previously formed the basis for the award of any other degree or diploma.

Kedar Kumar Dora

21BTCSE31

ACKNOWLEDGEMENTS

I feel honored to avail myself of this opportunity to express my deep sense of gratitude to my guide Dr. Kalyan Das, Assistant Professor, Department of Computer Science Engineering, Sambalpur University Institute of Information Technology, Burla, Odisha, India for his valuable inspiration, guidance, and warm encouragement throughout my research work. His profound knowledge and timely advice were very much helpful in my research work without which my thesis could never be able to see this day. Proud to be working under him and he has given me every freedom for working in this Project.

Kedar Kumar Dora

ABSTRACT

Audio classification has become a pivotal aspect of machine learning, enabling systems to identify and categorize sounds efficiently. This project focuses on classifying audio into two distinct categories: screaming and non-screaming sounds. Utilizing state-of-the-art deep learning architectures, including GRU, LSTM, RNN, and neural networks, the model achieves remarkable accuracy of up to 95.58%, with comprehensive evaluations across various metrics. Extensive preprocessing steps, feature extraction techniques like MFCC, and model optimization strategies were employed to ensure robustness and scalability. While the hybrid model aimed to enhance accuracy further, individual models performed slightly better. This project lays the foundation for real-world applications, particularly in safety systems, where identifying distress sounds is critical. The findings indicate the potential for future advancements, including real-time implementation in mobile applications, enabling autonomous monitoring and alert systems.

Keywords: Audio classification, deep learning, hybrid models, scream detection, Audio Features, machine-learning

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

MFCC	Mel-Frequency Ceptral Coefficients
RNN	Recurrent Neural Network
GRU	Gated Recurrent Unit
LSTM	Long Short-Term Memory
SMOTE	Synthetic Minority Oversampling Technique
ROC-AUC	Receiver Operating Characteristics
ZCR	Zero Crossing Rate
ML	Machine Learning
DL	Deep Learning
NN	Neural Network
AI	Artificial Intelligence
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
API	Application Programming Interface
CSV	Comma Separated Value
MSE	Mean Squared Error
MCC	Matthew’s correlation coefficient
MAE	Mean Absolute Error
LLV	Log Loss Value
CNN	Convolutional Neural Network
LR	Logistic Regression
RF	Random Forest
DT	Decision Tree

TABLE OF CONTENTS

1. CHAPTER 1

INTRODUCTION

Sound is a vital component of the environment, carrying significant information that aids in communication, alertness, and understanding contextual surroundings. Audio classification, a branch of machine learning and signal processing, has gained substantial attention in recent years for its potential in various applications, such as speech recognition, emotion detection, and environmental sound analysis. This project focuses on leveraging audio classification techniques to distinguish between human screams and non-screaming sounds, with an emphasis on creating a reliable framework that can be utilized for safety and security applications.

The core of this study lies in extracting meaningful audio features, such as Mel-Frequency Cepstral Coefficients (MFCCs), spectral properties, and zero-crossing rates, to represent sound patterns accurately. These features form the basis for training multiple machine learning (ML) and deep learning (DL) models, including Neural Networks (NN), Long Short-Term Memory Networks (LSTM), Gated Recurrent Units (GRU), and Simple Recurrent Neural Networks (RNN). Furthermore, the project explores a hybrid model approach to combine the strengths of individual models and enhance classification performance.

Class imbalance, a prevalent challenge in real-world datasets, was addressed using Synthetic Minority Oversampling Technique (SMOTE), ensuring a balanced representation of all sound categories. Extensive experimentation with these models provided insights into their comparative performance, with metrics such as accuracy, precision, recall, and F1 score evaluated for each.

Although this project currently focuses on the audio classification component, it sets a robust foundation for integrating the trained model into an application. Such an application could act as a real-time personal safety assistant capable of detecting danger signals in its environment. This report presents the methodologies, experimentation results, and a detailed analysis of model performance, providing a comprehensive understanding of the implemented audio classification system.

2. CHAPTER 2

LITERATURE REVIEW

Audio classification, particularly in detecting human screams, has garnered significant attention due to its applications in surveillance, security, and emergency response systems. Various machine learning techniques have been employed to enhance the accuracy and efficiency of such systems.

Nazir et al. (2018) provided a comprehensive review of scream detection and classification techniques, highlighting the importance of selecting appropriate acoustic parameters and classification methods for effective situation understanding. Their work emphasizes that screams can be categorized into various emotional classes, such as happiness, sadness, fear, and danger, underscoring the complexity of accurate scream classification -The Sai Organization.

In the realm of hybrid models, the integration of Convolutional Neural Networks (CNNs) with Recurrent Neural Network (RNN) variants has shown promise in audio classification tasks. A study by Khan et al. (2023) proposed a hybrid architecture combining CNN with RNN variants like Long Short-Term Memory (LSTM), Bidirectional LSTM (Bi-LSTM), Gated Recurrent Unit (GRU), and Bidirectional GRU (Bi-GRU). Their experiments on music classification revealed that the CNN+Bi-GRU combination achieved the highest accuracy of 89.30% when using Mel-spectrogram features, indicating the effectiveness of such hybrid models in capturing both spatial and temporal features of audio signals -MDPI.

Furthermore, the combination of CNNs with RNNs has been explored in various contexts. For instance, a study on classifying local language slangs using a CNN-RNN hybrid model demonstrated that the blended combination of CNN with GRU achieved 86% accuracy, while CNN with LSTM produced 91% accuracy, showcasing the potential of such architectures in handling complex audio classification tasks -IEEE Xplore.

In the specific context of scream detection, several studies have explored different machine learning approaches. For example, a project demonstrated the development of a scream detection system using Python and machine learning techniques to automatically identify scream sounds from other non-alert sounds, highlighting the practical applications of such systems in real-time scenarios -Medium.

Additionally, a study proposed a machine learning approach to identify screams in voice recordings, comparing the performance of Support Vector Machines (SVM) and LSTM networks. The analyses showed that both methods provided superior scream detection performance, with SVM slightly outperforming LSTM. Due to its lower complexity, SVM was considered a better candidate for real-time implementation in autonomous embedded systems -MDPI.

These studies underscore the potential of hybrid models in audio classification tasks, particularly in detecting human screams. By leveraging the strengths of different neural network architectures, such models can effectively capture the complex patterns inherent in audio signals, leading to improved classification performance.

3. CHAPTER 3

OBJECTIVE

The primary objective of this minor project is to develop and evaluate a robust audio classification system capable of accurately categorizing audio signals into predefined classes. This involves leveraging various machine learning models, including individual architectures as well as a hybrid ensemble model that combines these architectures with specific weights.

The primary objectives of this project are:

i. **Effective Scream Detection:** Develop a robust system capable of accurately distinguishing human screams from non-screaming sounds, ensuring reliable classification even in noisy environments.

ii. **Feature Utilization:** Identify and utilize relevant audio features that significantly contribute to improved classification accuracy, such as MFCCs, spectral properties, and temporal patterns.

iii. **Enhancement of Safety Applications:** Provide a foundation for building real-time safety applications that respond proactively to distress signals, enhancing individual security and emergency response.

iv. **Adaptability to Class Imbalance:** Address the challenges posed by imbalanced datasets to ensure fair and balanced model performance across all classes.

v. **High Model Performance:** Aim for optimal precision, recall, and overall classification accuracy, ensuring the system is reliable for practical use.

vi. **Scalability and Future Integration:** Design the classification system in a manner that supports future enhancements, such as integration into mobile applications or real-time audio monitoring devices.

vii. **Support for Real-World Scenarios:** Ensure the model’s adaptability to diverse audio inputs, including suppressed screams, environmental noises, and overlapping sounds.

These objectives collectively aim to develop a reliable and effective solution for scream recognition, addressing its significance in modern safety applications.

4. CHAPTER 4

METHODOLOGY

This project involves the design, development, and evaluation of an audio classification system capable of distinguishing between two classes of audio: screaming sounds and non-screaming sounds. The methodology is systematically structured to ensure robust data handling, model development, and evaluation processes, while the implementation turns this design into a working system, utilizing modern tools and computational resources. Below, we provide a comprehensive integration of both the methodology and implementation steps to give a clear view of how the project was executed.

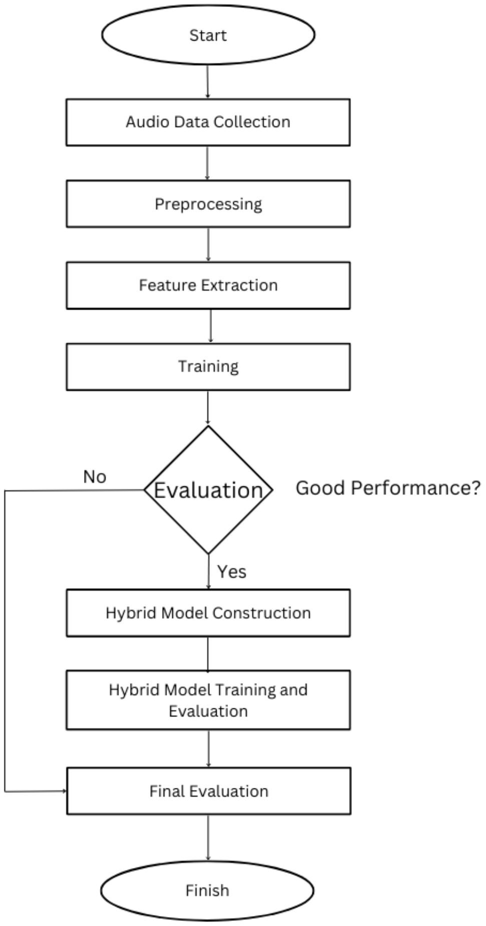

Figure 1: Flowchart of Methodology

4.1. Data Collection and Preparation

The success of any machine learning model depends heavily on the quality and diversity of the data used. For this project, audio samples were sourced from publicly available datasets, supplemented by additional recordings to ensure realistic, diverse scenarios. The dataset was split into two broad categories:

a) ***Screaming Sounds***: Human screams captured in various contexts that represented distress or danger.
b) ***Non-Screaming Sounds***: This category includes daily life sounds such as background noises, animal sounds, vehicle horns, and ambient music.

The dataset contained a total of 3,177 audio files, split equally between the two categories to maintain a balanced distribution, which is essential for training a classifier without bias. To ensure uniformity across the dataset, raw audio data was preprocessed as follows:

a) ***Loading and Resampling***: All audio files were resampled to a consistent 22,050 Hz and converted to mono. This ensured uniformity in audio properties, which is crucial for feature extraction.
b) ***Trimming and Padding***: Audio files longer than 5 seconds were trimmed, while files shorter than this were padded with silence to a fixed length of 5 seconds. This step was important for ensuring that all input data had the same duration.

4.2. Feature Extraction

Key features from the audio signals were extracted using both time-domain and frequency-domain techniques. The focus was on:

a) ***Mel Frequency Cepstral Coefficients (MFCCs)***: These were calculated for the first 13 coefficients, which capture the spectral properties of the audio.
b) ***Spectral Features***: This included spectral centroid, bandwidth, and roll-off, which help describe the frequency content of the audio.
c) ***Temporal Features***: Key temporal features like zero-crossing rate and root mean square energy were extracted to capture time-domain characteristics.
d) ***Spectrogram Features***: The mean, median, and variance of the spectrogram intensities were computed to capture the energy distribution across time.

All features were then scaled between 0 and 1 using MinMaxScaler to normalize the feature space, ensuring that the features were on a comparable scale for model input.

4.3. Data Balancing and Augmentation

Though initial efforts ensured a balanced distribution of the dataset, minor class imbalance still persisted. To further address this issue, the Synthetic Minority Oversampling Technique (SMOTE) was applied. SMOTE works by generating synthetic samples for the minority class to balance out the data, ensuring that the classifier learns the characteristics of both classes equally. This was a critical step in preventing the model from being biased towards the majority class.

4.4. Model Development and Training

With the preprocessed data, several machine learning and deep learning models were developed and trained to classify the audio samples. Three types of models were designed and implemented:

a) ***GRU Model***: The Gated Recurrent Unit (GRU) model was used to capture temporal patterns in the audio data. It was designed with one GRU layer having 128 units, followed by dense layers. ReLU activation functions were used for hidden layers, and a sigmoid activation function was applied at the output layer. The Adam optimizer was used with a learning rate of 0.0005.

b) ***Neural Network (NN)***: A fully connected feedforward neural network was implemented, featuring two dense layers with 128 and 64 units, respectively. Similar to the GRU model, ReLU activation functions were applied for hidden layers, and the output layer used a sigmoid activation. L2 regularization and dropout were used to prevent overfitting.

c) ***RNN Model***: A simple Recurrent Neural Network (RNN) was developed to capture sequential dependencies in the audio data. It included one RNN layer with 50 units and used the Adam optimizer with the same learning rate of 0.0005.

These models along with 8 other models were trained using a training set (80%) and validated using a test set (20%), with class weights applied to address any lingering class imbalance. Each model was trained for up to 100 epochs, with early stopping and learning rate scheduling applied to prevent overfitting and improve training efficiency.

During the training process, batch size was set to 32. Training was carried out on a high-performance machine with an Intel Core i7 7820HQ processor (Quad-Core, up to 2.90 GHz), 16 GB of RAM, 8 GB Nvidia 930MX GPU, and 20 GB of free disk space. This hardware setup was essential for optimizing the model training process, particularly for deep learning models.

4.5. Model Evaluation and Metrics

Model evaluation was performed using key performance metrics, including:

a) Accuracy:

The percentage of correctly classified samples.

$$\frac{TP + TN}{TP + TN + FP + FN}$$

Eq. 1

b) Precision:

The proportion of true positive predictions among all positive predictions.

$$\frac{TP}{TP + FP}$$

Eq. 2

c) Recall:

The proportion of actual positives correctly identified by the model.

$$\frac{TP}{TP + FN}$$

Eq. 3

d) F1-Score:

The harmonic mean of precision and recall, providing a balance between the two.

$$2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Eq. 4

e) Error Rate:

It is the inverse of accuracy, shows the proportion of error from the predictions.

$$1 - Accuracy$$

Eq. 5

f) Mean Squared Error:

Mean Squared Error is used to measure the average squared difference between the actual and predicted values. It is commonly used for regression tasks.

$$\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

Eq. 6

Where:

- y_i = Actual value
- $\hat{y}_i$= Predicted value
- N = Number of samples

g) Mean Absolute Error:

Mean Absolute Error calculates the average of the absolute differences between the actual and predicted values. It is also used for regression tasks and provides a linear score to penalize errors.

$$\frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$$

Eq. 7

h) Matthews Correlation Coefficient:

Matthews Correlation Coefficient is a metric that takes into account all four components (TP, TN, FP, FN) and is particularly useful for binary classification tasks, especially when classes are imbalanced.

$$\frac{TP.TN - FP.FN}{(TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN)}$$

Eq. 8

The MCC produces a value between -1 and 1, where:

- **1** indicates perfect classification,
- **0** indicates no better than random prediction,
- **-1** indicates total disagreement.

i) Log Loss Value:

Log Loss evaluates the performance of a classification model where the prediction input is a probability value between 0 and 1. It measures the distance between the predicted probability and the actual class labels.

$$-\frac{1}{N}\sum_{I=1}^{N}[y_i \log(p_i) + (1 - y_i)\log(1 - p_i)]$$

Eq. 9

Where:

- y_i = Actual label (0 or 1)
- p_i = Predicted probability for class 1
- N = Number of samples

Lower values of log loss indicate better model performance.

j) ROC-AUC:

The area under the Receiver Operating Characteristic curve, indicating the model's discriminative power between the two classes.

$$AUC = \int_0^1 True\ Positive\ Rate(x)\, dx$$

Eq. 10

The results were compared for all models GRU, NN, RNN and the hybrid model using these metrics. Visualization tools like precision-recall curves, accuracy plots, and a comparative table were generated to give a clear understanding of how each model performed. Interestingly, while the hybrid model showed a slightly lower accuracy of 94.91%, the GRU model emerged as the best individual model with an impressive 95.58% accuracy.

4.6. Hybrid Model Development

Given the complementary strengths of the individual models, a hybrid model was created by combining the predictions of the GRU, NN, RNN and LSTM models using a weighted averaging method. The weights for the models were assigned as follows based on their individual performance during evaluation:

Table 1: Models and Weights assignment table

Model Name	Model Weightage
Gated Recurrent Units	0.5
Simple Neural Network	0.4
Recurrent Neural Network	0.05
Long Short Term Memory	0.05

This ensemble approach aimed to leverage the strengths of each model, attempting to improve the overall performance and generalizability of the classifier. However, it was later observed that the hybrid model did not outperform the individual GRU model as expected. This could be attributed to the conflicting decision boundaries of the individual models, or suboptimal weighting of the models in the ensemble.

4.7. Observations and Analysis

Several key observations were made during the evaluation process:

Observations on the basis of the factors and metrics mentioned above were made. We shall see the observations in the next chapter (Results and Analysis).

These observations were crucial in fine-tuning the models and guiding future improvements. In particular, it was clear that further investigation into ensemble techniques and model optimization could potentially lead to better hybrid performance.

4.8. Code Structure and Modularity

The codebase was structured in a modular fashion to ensure scalability and maintainability. It was divided into separate scripts and functions that handled different tasks:

a) ***Preprocessing***: Scripts for handling audio normalization, trimming, and feature extraction.

b) ***Model Definition***: Functions to build and define the GRU, NN, RNN & Other models along with hybrid models.

c) ***Training***: Training routines with early stopping and model saving.

d) ***Evaluation***: Evaluation scripts that computed metrics, generated visualizations, and compared models.

This modular structure ensured that the code could be easily expanded or adapted for future work, such as integrating the models into a larger application framework or experimenting with more advanced machine learning techniques.

4.9. Summary

The implementation successfully established a functional audio classification system capable of distinguishing between screaming and non-screaming sounds using a combination of GRU, NN, LSTM, RNN and 8 other models.

By the observation of the performance of these models and different hyperparameters, we tried making a hybrid model with ensembled learning capabilities and weighted average prediction and this enabled the hybrid model to learn from multiple models at a time.

This structural way of implementation with a modular approach has not only helped a lot in the successful implementation of the whole project but also has made the whole process easier to understand and less time consuming.

5. CHAPTER 5

RESULTS AND ANALYSIS

The Results and Analysis section outlines the performance of the implemented models in classifying audio data into screaming and non-screaming categories. The evaluation metrics used include accuracy, precision, recall, and F1-score, among others. A detailed comparison of the models and insights derived from their performance are discussed below.

5.1. Performance of Individual Models

The models implemented GRU, NN, LSTM, and RNN demonstrated high accuracy and efficient classification capabilities. Key results for these models are summarized as follows:

a) ***GRU***: Achieved the highest accuracy of 95.58%, showcasing superior performance in distinguishing between the two audio classes.
b) ***NN***: Performed marginally lower than GRU with an accuracy of 95.49%. The simpler architecture contributed to its efficiency.
c) ***LSTM***: Secured an accuracy of 95.09%, benefiting from its ability to capture temporal dependencies in the audio features.
d) ***RNN***: Delivered an accuracy of 95.08%, slightly lagging behind LSTM due to potential issues like vanishing gradients in handling long sequences.

Despite their differences, all models provided high precision and recall values, reflecting their ability to minimize false positives and false negatives.

5.2. Hybrid Model Performance

The hybrid model, designed as an ensemble of GRU, NN, RNN, and LSTM with weights assigned as 0.5, 0.4, 0.05, and 0.05, respectively, achieved an accuracy of 94.91%. Contrary to expectations, the hybrid model's performance was slightly lower than the top-performing individual models.

Analysis:

The weighted averaging approach might have introduced conflicting decision boundaries among the models, reducing the overall performance.

The weights assigned to the models could have been further optimized through grid search or other hyperparameter tuning techniques.

5.3. Evaluation Metrics Comparison

The detailed evaluation metrics for all models, including precision, recall, F1-score, and accuracy, were compiled into a comprehensive table.

Table 2: Results table including metrics like accuracy, precision, f1-score, etc.

Model	Accuracy	F1	Precision	Recall	ROC	LLV	MAE	MSE	MCC
SVM	0.9238	0.9264	0.9378	0.9153	0.9242	2.7806	0.0762	0.0762	0.8476
RNN	0.9508	0.9534	0.9485	0.9584	0.9505	1.7720	0.0492	0.0492	0.9016
RF	0.9288	0.9341	0.9074	0.9624	0.9271	2.5667	0.0712	0.0712	0.8586
NN	0.9550	0.9575	0.9470	0.9683	0.9543	1.6231	0.0450	0.0450	0.9100
LSTM	0.9510	0.9537	0.9443	0.9634	0.9503	1.7669	0.0490	0.0490	0.9019
NB	0.8212	0.8156	0.8878	0.7543	0.8246	6.4442	0.1788	0.1788	0.6524
LR	0.8636	0.8649	0.9000	0.8324	0.8652	4.9150	0.1364	0.1364	0.7299
KNN	0.9273	0.9308	0.9282	0.9335	0.9270	2.6214	0.0727	0.0727	0.8542
GRU	0.9559	0.9583	0.9492	0.9678	0.9553	1.5896	0.0441	0.0441	0.9118
DT	0.8773	0.8807	0.8979	0.8642	0.8779	4.4235	0.1227	0.1227	0.7550
GBM	0.9182	0.9237	0.9033	0.9451	0.9168	2.9490	0.0818	0.0818	0.8366
CNN	0.9343	0.9375	0.9342	0.9410	0.9339	2.3698	0.0657	0.0657	0.8684
Hybrid	0.9491	0.9517	0.9475	0.9562	NAN	NAN	NAN	NAN	NAN

Insights from Evaluation Metrics:

a) Precision for all models remained consistently above 94%, indicating effective identification of true positives while minimizing false positives.

b) Recall values exceeded 95% for GRU and NN, highlighting their ability to correctly identify most screaming sounds.

c) F1-scores demonstrated a balanced trade-off between precision and recall, with GRU leading at 95.3%.

5.4. Visual Analysis

A collage of plots was created to provide a visual comparison of model performance. These plots include accuracy trends during training and validation, precision-recall curves, and F1-score comparisons.

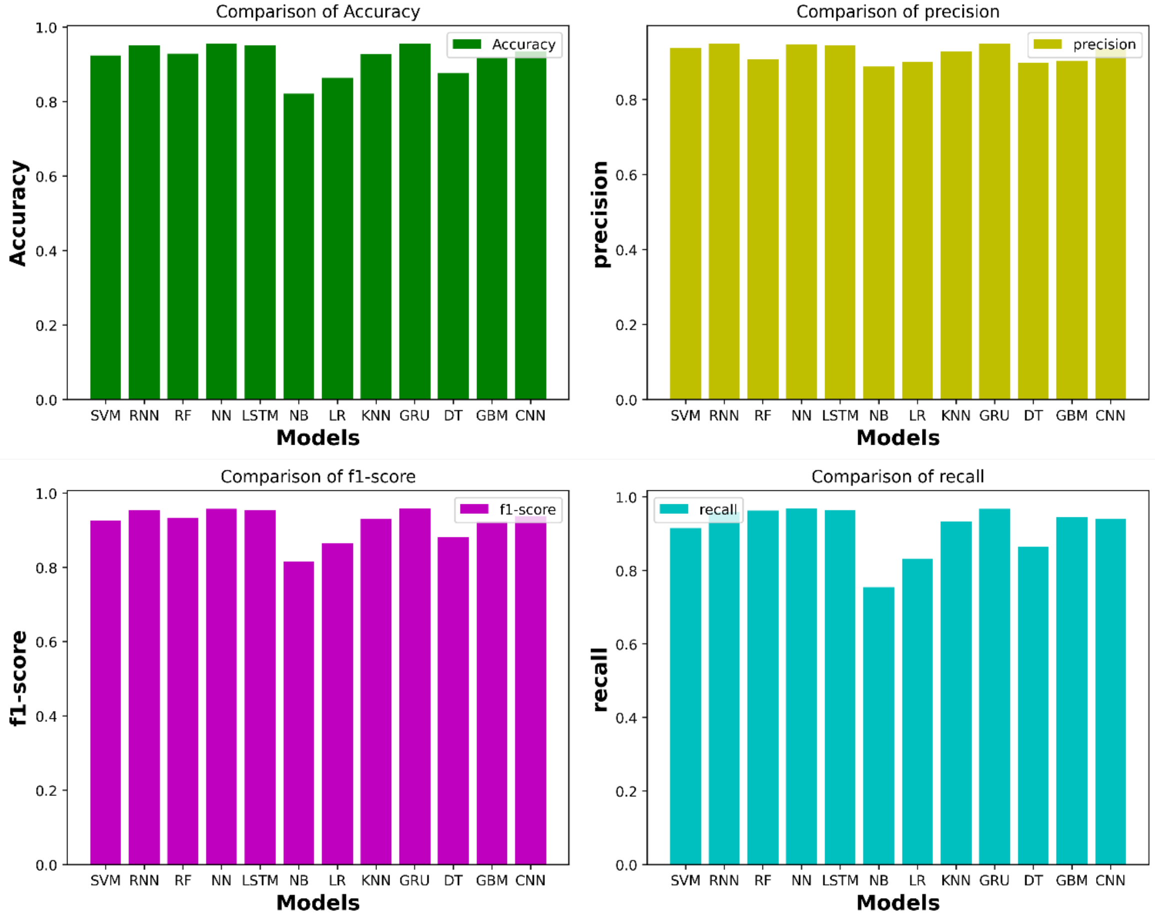

Figure 2: Accuracy, f1 score, precision and recall comparison of different models

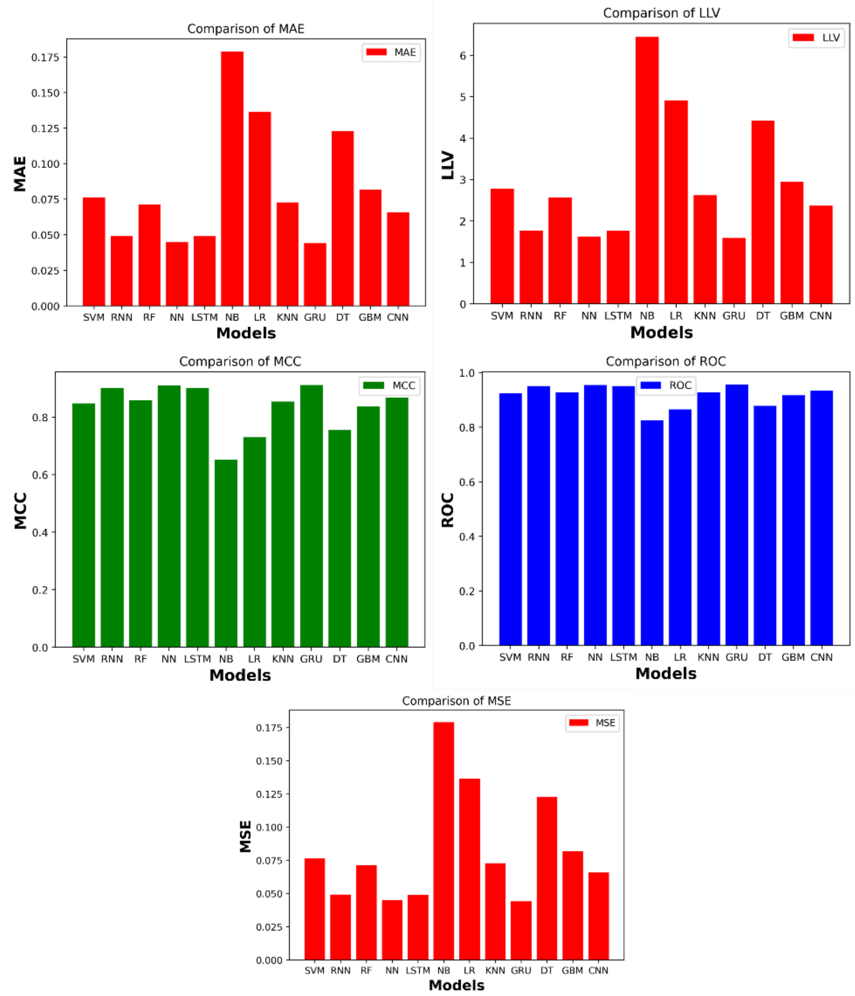

Figure 3: Comparison of MAE, MCC,LLV, ROC & MSE for different models

Observations:

a) Training accuracy curves for all models showed smooth convergence, indicating stable learning.

b) Validation accuracy remained close to training accuracy, suggesting minimal overfitting.

c) Precision-recall curves demonstrated the robustness of GRU and NN in distinguishing between screaming and non-screaming sounds.

5.5. Comparative Analysis

The GRU model emerged as the most reliable classifier due to its ability to capture temporal features effectively.

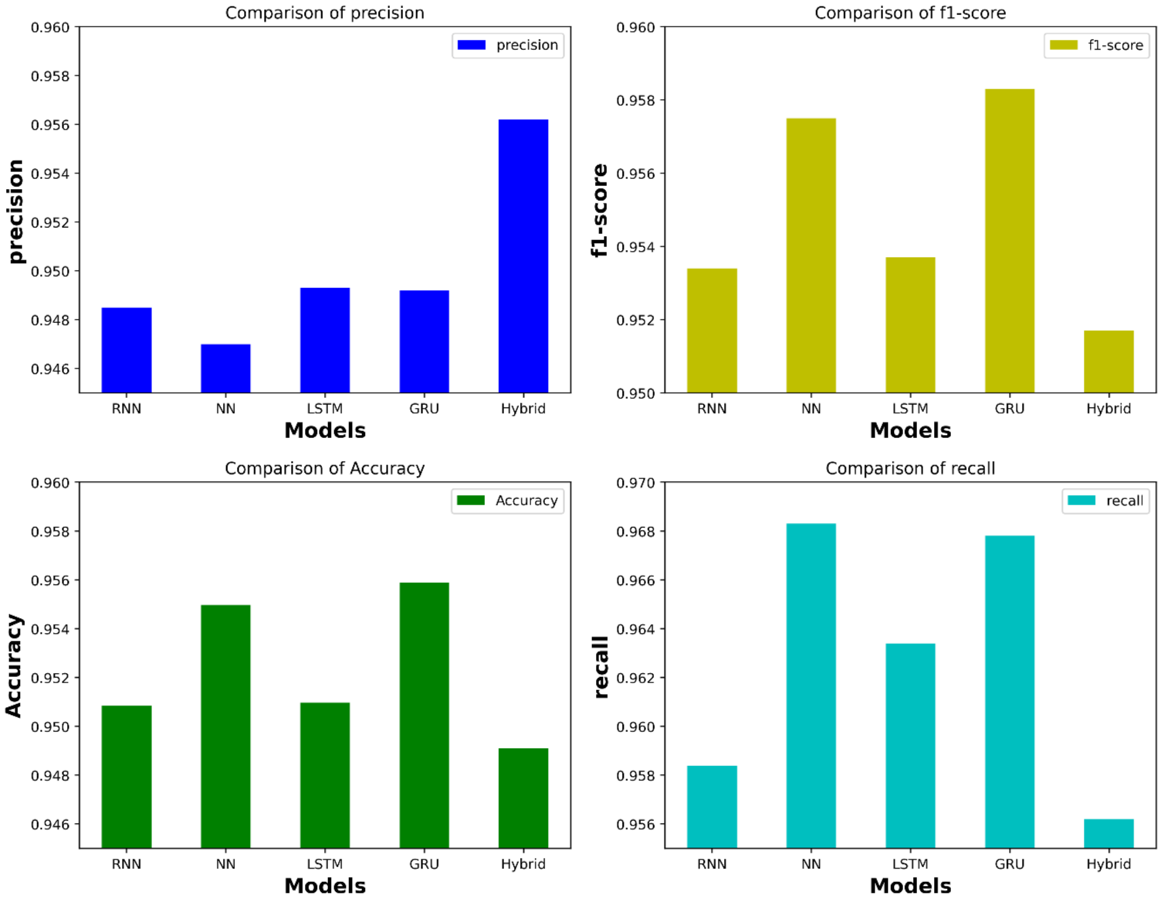

Figure 4: Comparison of hybrid model with other individual models

However, the slightly lower performance of the hybrid model raised questions about ensemble optimization. A deeper investigation revealed potential areas for improvement:

a) Recalibration of weights to emphasize the strengths of individual models.

b) Exploration of stacking or boosting techniques as alternative ensemble methods.

5.6. Error Analysis

Misclassifications were analyzed to identify patterns or common characteristics:

a) Some non-screaming sounds with high-pitched frequencies (e.g., animal cries or alarm tones) were misclassified as screams.

b) Suppressed or low-intensity screams occasionally fell into the non-screaming category, indicating a need for more sophisticated feature engineering or data augmentation.

Both these issues have been minimized and they exist in very small proportion.

5.7. Summary

The Results and Analysis highlight the effectiveness of the implemented models, with the GRU model achieving the best performance. While the hybrid model did not outperform individual models, it provided valuable insights into ensemble learning. Future enhancements will focus on refining ensemble techniques and addressing misclassification patterns to achieve even greater accuracy and reliability.

6. CONCLUSION

This study successfully demonstrated the implementation of a deep learning-based audio classification system aimed at distinguishing between screaming and non-screaming audio signals. The project employed multiple models, including GRU, NN, LSTM, and RNN, which were rigorously evaluated based on metrics such as accuracy, precision, recall, and F1-score. The GRU model emerged as the most effective, achieving an accuracy of 95.58%, followed closely by NN at 95.49%, LSTM at 95.09%, and RNN at 95.08%.

While the hybrid model, which combined these individual models with weighted contributions, was expected to outperform the standalone models, it achieved a slightly lower accuracy of 94.91%. This outcome highlighted the complexity of ensemble learning and the need for further optimization of model weights and decision boundaries.

The project's success underscores the importance of selecting appropriate machine learning architectures for audio classification tasks. Temporal models like GRU and LSTM excelled in capturing sequential patterns, while NN provided competitive performance with its simpler architecture. Additionally, the use of SMOTE for addressing class imbalance ensured balanced learning and improved model generalization.

The analysis revealed that while the models performed well overall, certain challenges remain, such as misclassification of suppressed screams or high-pitched non-screaming sounds. These issues point to potential areas for future research, including advanced feature engineering, data augmentation, and the exploration of more sophisticated ensemble techniques like stacking or boosting.

In conclusion, this project lays a strong foundation for scream detection systems by leveraging deep learning techniques. The insights gained will guide future enhancements, including integrating the classification model into a real-world application for robust audio-based safety systems.

7. FUTURE SCOPE

7.1. Applications in Personal Safety

a) Wearable Devices: Integration into wearable devices, mobile apps, or smart home systems for real-time distress sound detection, such as screams, enabling immediate alerts to emergency contacts or authorities.

b) Smart Home Security: Enhances traditional security systems by detecting distress sounds (e.g., screams), providing an additional layer of security.

7.2. Public Safety and Surveillance

a) Law Enforcement and Public Spaces: Potential for use in surveillance cameras across parks, streets, and transportation hubs to identify distress sounds and threats, facilitating crime prevention and faster rescue operations.

b) Smart City Integration: Assists in proactive monitoring of public spaces for potential emergencies.

7.3. Healthcare and Elderly Care

a) Monitoring Facilities: Can be used in healthcare or eldercare settings to monitor patients or elderly individuals who may need immediate assistance but are unable to make a call for help.

7.4. Expected Improvements and Integrations

a) Optimization of Real-Time Detection: Focus on enhancing the system's speed while maintaining high accuracy for real-time application.

b) Multilingual and Cultural Expansion: Extending the model to detect screams across various languages and cultural variations, making it applicable globally.

c) IoT Integration: Incorporating the technology into IoT devices, creating a connected network of safety devices for more comprehensive monitoring.

d) Distress Sound Classification Expansion: Future versions could classify additional distress sounds such as gunshots, explosions, or cries for help, transforming the system into a broader sound-based safety monitoring solution.

7.5. Technological Advancements

a) Deep Learning Integration: Continual improvements in deep learning models and hardware could lead to more robust, scalable, and efficient systems for both individual and societal safety.

b) Enhanced Scalability: With the expansion of use cases, the system can be adapted to more environments, offering larger-scale applications in real-world settings.

This project opens several opportunities for future research and application, particularly in the areas of public safety, personal security, and healthcare monitoring, with a clear path for evolution into more sophisticated and interconnected systems.

8. REFERENCES

1. MathWorks. (n.d.). Feature selection for audio classification. MathWorks. https://www.mathworks.com/help/matlab/examples/feature-selection-for-audio-classification.html
2. DigitalOcean. (n.d.). Audio classification with deep learning. DigitalOcean Community Tutorials. https://www.digitalocean.com/community/tutorials/audio-classification-with-deep-learning
3. Devopedia. (n.d.). Audio feature extraction. Devopedia. https://devopedia.org/audio-feature-extraction
4. Analytics Vidhya. (2022). Guide to audio classification using deep learning. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2022/04/guide-to-audio-classification-using-deep-learning/
5. MathWorks. (n.d.). Audio feature extraction. MathWorks Documentation. https://www.mathworks.com/help/audio/ug/audio-feature-extraction.html
6. GitHub. (n.d.). Audio classification deep learning. GitHub Repository. https://github.com/abishek-as/Audio-Classification-Deep-Learning
7. Room, C. (2021). Audio feature extraction. Machine Learning, 16, 51. https://www.scirp.org/reference/referencespapers?referenceid=3572780
8. IEEE Xplore. (2022). A survey of audio classification using deep learning. IEEE Journals & Magazine. https://ieeexplore.ieee.org/document/10258355
9. Reddit. (2020). Surfboard: Audio feature extraction for modern machine learning. Reddit r/MachineLearning.https://www.reddit.com/r/MachineLearning/comments/gqvnpv/p_surfboard_audio_feature_extraction_for_modern/
10. ArXiv. (2020). Surfboard: Audio feature extraction for modern machine learning. arXiv preprint arXiv:2005.08848. https://arxiv.org/abs/2005.08848
11. ACM Digital Library. (2022). Research on audio scene classification method based on deep learning. ACM Transactions on Multimedia Computing, Communications, and Applications. https://dl.acm.org/doi/fullHtml/10.1145/3675417.3675527
12. Stack Exchange. (n.d.). Feature extraction for sound classification. DSP Stack Exchange. https://dsp.stackexchange.com/questions/16994/feature-extraction-for-sound-classification

13. ArXiv. (2020). Audio classification using deep learning. arXiv preprint arXiv:2007.11154. https://arxiv.org/abs/2007.11154
14. Medium. (2020). Deep learning audio classification. Medium. https://medium.com/analytics-vidhya/deep-learning-audio-classification-fcbed546a2dd
15. Mohaimenuzzaman, M., Bergmeir, C., & Meyer, B. (2023). Deep active audio feature learning in resource-constrained environments. arXiv preprint arXiv:2308.13201. https://arxiv.org/abs/2308.13201
16. Khandelwal, R. (2020). Deep learning audio classification. Medium. https://medium.com/analytics-vidhya/deep-learning-audio-classification-fcbed546a2dd
17. Palanisamy, K., Singhania, D., & Yao, A. (2020). Rethinking CNN models for audio classification. arXiv preprint arXiv:2007.11154. https://arxiv.org/abs/2007.11154