Question 1 :Case Study - Identifying anger in audio voice recordings

Ms. Sneha Ramesh Dharne
Msc in Data Analytics
National College of Ireland
Dublin, Ireland
x23195703@student.ncirl.ie

***Abstract*— This report provides the present state of the art in anger detection from audio samples and put forward a strategy that adopts both classical and deep learning approaches. The technique copes with such problems as data skewness and noise fluctuations and can be considered as efficient and universal for the purpose of anger categorization in audio streams.**

Keywords*— *Anger detection, audio analysis, machine learning, deep learning, feature extraction, emotion recognition, data imbalance, noise variation.

I. INTRODUCTION

The identification of emotions especially anger from audio voice samples is a research topic is of immense importance with applications in call centre, health and well-being of patients, and machine learning. Because the identification of nonverbal emotional parameters concerning speech, in particular, forms a significant and crucial part of the contemporary systems, neuromorphic growth in terms of machine learning has become highly beneficial in probing the potential and providing a solid foundation for the development of such apparatus. In this report, I have incorporated a literature review of recent work done in the domain to propose a detailed approach of anger detection in audio samples. [1]

Interpreting the language and the signs that indicate the presence of anger involves rather complex analysis. Most of the previous approaches include the use of hand-crafted features and classical machine learning techniques The more recent approaches have included deep learning techniques that do not need feature extraction.[2] However, problems like data imbalance, variation in noise levels or recording environments and the requirement for online processing still remains an issue hence a careful selection of models has to be made with caution.

Below is a proposed methodology, which will incorporate elements of both classical and deep learning approaches for the purpose of recording and categorising anger into audio clips. The developed methodology is aimed at reflecting the specific peculiarities of the domain, providing a versatile and, at the same time, most effective approach that takes advantage of the variety of the machine learning models while avoiding their weaknesses. By the selection of the features, models, and training techniques, the proposed approach offers an effective and general solution for the anger detection in the audio signals.

II. PROPOSED APPROACH

A. Exploratory Data Analysis (EDA) / Data Preparation

Assumptions: Anger is one of the many emotional states that are identified on the audio recordings in the collection.
Noise and unpredictability may be introduced by the audio data, which may originate from a variety of environments with differing quality.

Suggested-Method:

Data Inspection: To gain an understanding of the distribution of emotional categories, we will start with an exploratory examination of the dataset, paying special attention to how anger is represented. We will Examine the data distribution visually to spot any anomalies or imbalances in the classes.

Techniques for Signal Processing: To normalize the audio files, apply signal processing techniques (such as converting stereo to mono or resampling to a constant frequency). This stage guarantees consistency throughout the dataset, which is essential for training the model.

Noise Reduction and Silence Removal: To improve the signal-to-noise ratio,[3] will use noise reduction techniques and eliminate silence portions from recordings.

Rationale: Exploratory Data Analysis (EDA) is critical for understanding the structure and characteristics of the dataset, identifying potential issues such as imbalanced classes, and ensuring that the data is in a suitable format for modelling. Literature emphasizes the importance of thorough EDA in audio-based emotion detection to mitigate issues related to noise and variability. By normalizing and cleaning the data, the model is less likely to learn spurious patterns, leading to better generalization.

B. Dimensionality Reduction / Feature Selection

Assumptions: The extracted features including, MFCCs, Spectrograms, and prosodic features may be of very high dimension this may cause over fitting.

Proposed Approach:

Dimensionality Reduction: PCA or LDA to reduce the dimensionality of the feature space that keeps important features only to make the data set tidy. This step is especially critical if a large number of features was adopted in the first place.

Feature Selection: Preferring the methods like Recursive Feature Elimination (RFE) [4] or use mutual information when deciding on the most important features on which to base emotions, to minimize the issues with overfitting models and to increase the interpretability of the model.

Rationale:

Large-dimensional features are sometimes noisy and create overfitting, especially when dealing with small sample size datasets. Those shortcomings are resolved by dimensionality reduction and feature selection where the dimensionality of the model can be reduced without compromising accuracy. This approach has often been applied in speech emotion recognition to find out the lower dimensional representation of the data reverting the most variance in fewer components than the original data set.

C. Feature Engineering /Feature Extraction

Assumptions : Prosodic features, spectrograms, and MFCCs are examples of extracted features that can be used to accurately depict the emotional content of audio data.

Suggested Method: Feature extraction involves taking the audio recordings and extracting spectrograms, MFCCs, and prosodic features (duration, pitch, and intensity). It has been demonstrated that these characteristics are useful in capturing the emotional content of speech.[5]

Feature engineering: If more features are needed, create them. For example, you could create spectral contrast or delta-MFCCs, which are temporal derivatives of MFCCs that could provide more information about the audio data.

Rationale: The research has demonstrated the efficacy of prosodic characteristics and MFCCs in emotion identification tasks. With deep learning models, spectrograms are especially helpful since they enable CNNs to identify local patterns in the frequency domain.

D. Choice of modelling techniques

Assumptions: - Recognizing fury in speech recordings requires analysing intricate patterns in the audio data, encompassing both spectral and temporal aspects.

The dataset requires a strong and adaptable model because it contains a variety of recordings with possible heterogeneity in speaker qualities, recording settings, and emotional intensity.

Proposed Approach:

1. **Baseline Model** - Support Vector Machines (SVM)
   SVMs work effectively for classification jobs where the data is low dimensional, and the feature space is well-defined. Because they may deliver good performance on tiny datasets and are very easy to apply, they are especially helpful for setting a baseline.

   Rationale: Support vector machines (SVMs) are effective when combined with prosodic features and MFCCs [5] because they can identify the best hyperplane to maximize the margin between various emotion classes, including rage. This is especially important when the dataset is unbalanced because SVMs can be adjusted to manage these kinds of situations by varying the regularization parameters and kernel functions.

2. **Model Hybrid:** CNN-LSTM: Using convolutional layers to first extract spatial features from spectrograms and LSTM layers to describe the temporal dynamics of these features, the hybrid CNN-LSTM model incorporates the best aspects of both CNNs and LSTMs.[6]
   Rationale: The CNN part of the model is in charge of spotting significant elements in the frequency spectrum of the audio, like spectral peaks or pitch shifts that signify fury. The LSTM component receives these extracted features and uses them to represent the sequential nature of the data and capture the evolution of these features over time. Because it enables the model to take into account both the instantaneous auditory properties and their temporal evolution, this combination is very successful for rage detection.

Challenges and Considerations:

- The model may be biased toward non-anger classes if anger is underrepresented in the dataset. During model training, take into account methods like class-weighted loss functions or the Synthetic Minority Over-sampling Technique (SMOTE) [7]to address this.
- computing Resources: Large computing resources are needed for deep learning models, particularly for hybrid CNN-LSTM architectures. These needs can be managed with the help of distributed computing and GPU efficiency, particularly when training huge datasets or fine-tuning models.

In conclusion, a thorough approach to rage detection in audio recordings is made possible by the modelling techniques chosen, which are based on both sophisticated deep learning architectures and traditional machine learning techniques. The selection of CNNs and LSTMs, especially in a hybrid model,[6] is justified by their demonstrated capacity to capture both spatial

E. Hyperparameter Optimisation

Assumptions: To prevent overfitting and maximize model performance, optimal hyperparameters are essential.

Proposed Approach:

Grid Search and Random Search: To fine-tune hyperparameters, use grid search. Begin with a coarse grid and focus your search on the most promising values of your hyperparameters. A more thorough investigation of the hyperparameter space can be accomplished by random search.

Using Bayesian optimization, you can effectively explore the hyperparameter space and discover the global optimum with fewer evaluations than you would with standard methods.

Rationale: Performance of the model is greatly impacted by hyperparameter tweaking. Grid search is comprehensive, but it can be computationally costly. By randomly sampling hyperparameters, random search provides a more effective substitute. Bayesian optimization,[8] in which the performance is represented as a effective in finding optimal configurations with fewer iterations.

F. Model Evaluation :

Assumptions: A strong assessment approach is required to guarantee that the model applies effectively to untested data.

Suggested Method:
Cross-validation: To evaluate model performance across various data subsets and provide a reliable estimate of generalization error, apply k-fold cross-validation.[9]

Metrics for Evaluation: Utilize measures like memory, accuracy, precision, and F1-score; for the anger class, pay special attention to recall and F1-score because of their significance in the issue domain.

Rationale: For a more accurate estimation of model performance, cross-validation is crucial, particularly when there is a lack or imbalance in the data.
The evaluation metrics used are in line with the objective of precisely identifying anger, which might not be well-represented in the dataset. Stressing F1-score and recall guarantees.

G. Scalability Issues:

Assumptions:
If the model is used in a production context, it must be scalable to manage big datasets or real-time processing.

Suggested Method:

Model Optimization: To increase scalability and decrease inference time, apply strategies such model quantization, trimming, and hardware acceleration (such as using a GPU or TPU).

Distributed Computing: To expedite the training process on huge datasets, use distributed training strategies using frameworks that support multi-GPU setups, such as TensorFlow [10]or PyTorch.

Rationale: scalability is crucial for deploying models in real-world applications, especially when dealing with large-scale or real-time audio processing.
Techniques like model pruning and quantization help reduce model size and improve inference speed without significant loss in accuracy. Distributed computing further ensures that the model can handle large datasets efficiently.

H. ***Ethical Implications***

Assumptions: There may be serious ethical ramifications from the use of rage detection in audio recordings, especially in regard to bias and privacy.

Suggested Method:

Bias Mitigation: To prevent bias in the model's predictions, make sure the training dataset is varied and represents a range of demographics. Check the model frequently for biases and make any required adjustments to the training set or model architecture.

Privacy-Related Issues: Establish stringent data privacy procedures, such as anonymizing audio recordings, storing data securely, and abiding by all applicable laws (such as the GDPR).

Justification: Ethical factors are crucial while developing emotion identification systems. A biased model may treat some groups unfairly, and privacy infractions may have detrimental effects. Proactively addressing these problems guarantees that the model is equitable, responsible, and effective (Crawford et al.,

III. REFERENCES

[1] A. Surana *et al.*, "An audio-based anger detection algorithm using a hybrid artificial neural network and fuzzy logic model," *Multimed Tools Appl*, vol. 83, no. 13, pp. 38909–38929, Apr. 2024, doi: 10.1007/s11042-023-16815-7.

[2] Sumera, K. Vaidehi, and Q. Nisha, "A Machine Learning and Deep Learning based Approach to Generate a Speech Emotion Recognition System," in *2024 11th International Conference on Computing for Sustainable Global Development (INDIACom)*, Feb. 2024, pp. 573–577. doi: 10.23919/INDIACom61295.2024.10498783.

[3] S. Vos, O. Collignon, and B. Boets, "The Sound of Emotion: Pinpointing Emotional Voice Processing Via Frequency Tagging EEG," *Brain Sciences*, vol. 13, no. 2, Art. no. 2, Feb. 2023, doi: 10.3390/brainsci13020162.

[4] A. Priyatno and T. Widiyaningtyas, "A SYSTEMATIC LITERATURE REVIEW: RECURSIVE FEATURE ELIMINATION ALGORITHMS," *JITK (Jurnal Ilmu Pengetahuan dan Teknologi Komputer)*, vol. 9, pp. 196–207, Feb. 2024, doi: 10.33480/jitk.v9i2.5015.

[5] T. Mary Little Flower, T. Jaya, and S. Christopher Ezhil Singh, "Data augmentation using a 1D-CNN model with MFCC/MFMC features for speech emotion recognition," *Automatika*, vol. 65, no. 4, pp. 1325–1338, Oct. 2024, doi: 10.1080/00051144.2024.2371249.

[6] "Emotion Detection Model for a Bot using CNN and LSTM - ProQuest." Accessed: Aug. 22, 2024. [Online]. Available: https://www.proquest.com/openview/ce4f34a79e9ebcd56a0c016efc42806b/1?pq-origsite=gscholar&cbl=18750&diss=y

[7] V. V. N. Raju *et al.*, "Enhancing emotion prediction using deep learning and distributed federated systems with SMOTE oversampling technique," *Alexandria Engineering Journal*, vol. 108, pp. 498–508, Dec. 2024, doi: 10.1016/j.aej.2024.07.081.

[8] S. Michael and A. Zahra, "Multimodal speech emotion recognition optimization using genetic algorithm," *Bulletin of Electrical Engineering and Informatics*, vol. 13, no. 5, Art. no. 5, Oct. 2024, doi: 10.11591/eei.v13i5.7409.

[9] S. Bhattacharya, S. Borah, and B. K. Mishra, "Deep Multimodal K-Fold Model for Emotion and Sentiment Analysis in Figurative Language," Feb. 07, 2024, *Rochester, NY*: 4719406. doi: 10.2139/ssrn.4719406.

[10] A. A. S and J. D, “Voice Assisted Facial Emotion Recognition System For Blind Peoples With Tensorflow Model,” in *2024 IEEE International Students’ Conference on Electrical, Electronics and Computer Science (SCEECS)*, Feb. 2024, pp. 1–4. doi: 10.1109/SCEECS61402.2024.10481892.

Question 2: Paper Review : A Novel Deep Multi-head Attentive Vulnerable Line Detector

IV. Structure and Title

The concept of the study can be grasped from the paper title, 'A Novel Deep Multi-head Attentive Vulnerable Line Detector,' which also spells out the approach to detecting vulnerable software code lines. The paper's structure makes sense: An important thing to know is that a logical flow is observed, and while an introduction, a detailed methodology, the results, and the discussion are stated, it can complete the composition. This type of research paper should have this format which assists the reader to jump between a number of various sections. Nonetheless, it would have been even better if the authors expanded the methods and results sections to include more analysis.

V. Abstract

In essence, the abstract provides the simplest identification of the study: problem being solved, method used and research findings. As it emerges, it underlines most explicitly the fact that the approach is innovative and that it is superior to other approaches with respect to the goals that are relevant according to the problem in question. Still, one could see that the abstract does not contain enough information concerning the results, which could be specified further, for instance, the scale of improvement concerning baseline models.

VI. Introduction

The introduction stated a brief background of the problem in identifying vulnerable codes in software and etching out the significance of line-level as I compared it to function-level detection. All in all, it provides a good background for the authors to launch their discussion because it presents an overview of the current literature and points to the existing research limitations, including the scarcity of line-level detection. The purpose, significance and novelty of the study are also well defined and elaborated in the introduction part. Still, it might give a more detailed vision of the target audience apart from the sphere of software engineering and machine learning.

VII. Graphical Abstracts

The graphical abstract explains how deep learning model can be used to analyse line of code of a software to determine its vulnerability. It forms a sequence from data preparation to the encoding of source code lines followed by the development of memory and multi-head attention network and at last providing the prediction of code vulnerability.

VIII. Methodology

Another strength of the paper is the portrayal of the methodology section whereby the authors expounded the deep learning model for vulnerability detection– which is the study's central focus. The authors also patiently elaborate how exactly it is more enhanced than previous models such as the multi-head attention and memory networks. The tasks in data pre-processing, the structure of the investigated model and training are detailed, thus enhancing the papers' reproducibility. The only aspect which could have been discussed in higher detail regarding their applicability is the head-to-head comparison with the other approaches. The approach presented in the paper as the admissible method of practice embraces a new deep learning approach that is aimed at discovering those peculiarities of the source code which are associated with potential threats as well as aims at operating at the LoC level. The first process which is performed here is the preprocessing, in which the source code and each line, the tokens and their types

are tokenized. This step involves the use of Clang tokenizer in such a way that the model is in a position to relate the content of the token with the others within a code. After that, the line encoding process takes each line of a code and converts it into the vector taking into account the tokens along with the type of the tokens. The following vectors are through positional encoding, to retain the position of the tokens which is very crucial for reading code. The key and unique feature of the methodology is as simple as it is called – the hybrid memory and multi-head attention network. Rather, this network uses these encoded vectors to quantify the relations exquisite between different lines of code. This is made possible by the multi-head attention mechanism, which means that the model can learn distinct related representations and, therefore, discover various aspects of these relations at the same time that would allow the identification of weakness.

The memory network framework in turn improves this process by making use of attention mechanism in multiple layers which in turn helps in having a better context recognition.

Finally, the processed information is feed to a fully connected network by which vulnerability status for each of the lines of code is calculated. It complements the outcomes and sharpness to a greater level than the existing processes of differentiating vulnerabilities in a massive manner.

IX. RESULTS

Regarding the results of the analysis it is possible to conclude that the efficiency of the proposed deep learning model for the indication of the presence of Line of Code vulnerabilities in the software is proved. The model was benchmarked against other baseline models for example: memory networks, LSTM, GRU and Translator networks as well as the built-in static analysers such as Clang, Cppcheck and Frama-c.

In sub-cases of more elaborated classifications, whereby it was possible to determine how accurately the model pinpointed the exact type of vulnerability, the improvement that was suggested achieved 98% of accuracy. performed experiments in which the accuracy rises to 8% relative to the baseline models. For instance, the so called Transformer network, which is classified as one of the better basic models aimed at hitting the 96. And with memory network it got 93% and only 8% of the people reminiscence the ad when they were watching the first episode of the show. The two ads were created for the company Lux. 3%. This proves that having given the multi-head attention into the memory networks assist the model in understanding the relationships that exist between different lines of code.

The proposed model was able to maintain a low loss and a high levels F1 score of 0.99 in the binary model of safe and unsafe that was developed in the coarser deterministic analysis. It can claim that, on average, it achieves 0% accuracy and yet this is so much higher than the standard static analysis tools out there. Once more, the given choices were justified by the outcomes of the ablation study, according to which certain peculiarities, including multi-head attention, were really beneficial.

These facts are significantly suggestive of the fact that the proposed model is not only superior in the sense that it is more accurate in the identification of granule level significant vulnerabilities but is also more consistent in the performance of the task.

X. CONCLUSION/DISCUSSION

At the end, the authors emphasize that the developed hybrid memory and multi-head attention network can increase the effectiveness of software vulnerability detection. The crisis the research responds to is the absence of methods that detect security problems at the precise line-of-code level, which offers more

valuable information to programmers than the function level. The authors underscore the fact that their model is more accurate and yields higher accuracy and F1 scores under different forms of evaluation than the other model tested; they therefore claim that their model may be more useful than the other in flagging insecurity-containing sources of vulnerability in software code.

Furthermore, the conclusion focuses on epilogue of this research. The main advantage of the proposed method is an improvement of the granularity of the vulnerability detection; which will in turn reduce amount of time that developers spend in hunting for vulnerabilities that need to be fixed in order to produce more secure software systems. The authors also rightly addressed the limitations of their work mainly using synthetic datasets to train and test the model and have recommended for future research to investigate the effectiveness of the proposed model if used with real world data.

The conclusion in general reactivates that the proposed model was new and has brought a positive change into the improvement of software security. It also leaves the door open for further elaborations and utilisation of the defined approach, and the approach could be complemented with other method and tools that work within the SDL.

XI. PREVIOUS RESEARCH

The paper references a wide range of the existing literature that is closely related to the topic: the more consistent reference list has been provided, the higher research outcome and credibility of the paper is. It can be seen that the citations are current, relevant and they are properly used in the argument. Although the aspects of ethical thinking are raised in the paper, the discussion on ethical issues is somewhat restrained, which is quite questionable given the possible use of automated tools for searching vulnerabilities in the software product. Certainly, the study could have incorporated considerably more profound analysis of these aspects.