IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 28 November 2023, accepted 25 December 2023, date of publication 1 January 2024,
date of current version 16 January 2024.

Digital Object Identifier 10.1109/ACCESS.2023.3349097

RESEARCH ARTICLE

Live Event Detection for People's Safety Using NLP and Deep Learning

AMRIT SEN[1], GAYATHRI RAJAKUMARAN[1], MIROSLAV MAHDAL[2], SHOLA USHARANI[1], VEZHAVENDHAN RAJASEKHARAN[3], RAJIV VINCENT[1], AND KARTHIKEYAN SUGAVANAN[1]

[1]School of Computer Science and Engineering, Vellore Institute of Technology, Chennai 600127, India
[2]Department of Control Systems and Instrumentation, Faculty of Mechanical Engineering, VSB-Technical University of Ostrava, 708 00 Ostrava, Czech Republic
[3]School of Mechanical and Building Sciences, Vellore Institute of Technology, Vellore 632014, India

Corresponding author: Miroslav Mahdal (miroslav.mahdal@vsb.cz)

This work was supported by the European Union under the REFRESH—Research Excellence For REgion Sustainability and High-Tech Industries via the Operational Program Just Transition under Project CZ.10.03.01/00/22_003/0000048.

ABSTRACT Today, humans pose the greatest threat to society by getting involved in robbery, assault, or homicide activities. Such circumstances threaten the people working alone at night in remote areas especially women. Any such kind of threat in real time is always associated with a sound/noise which may be used for an early detection. Numerous existing measures are available but none of them sounds efficient due to lack of accuracy, delays in exact prediction of threat. Hence a novel software-based prototype is developed to detect threats from a person's surrounding sound/noise and automatically alert the registered contacts of victims by sending email, SMS, WhatsApp messages through their smartphones without any other hardware components. Audio signals from Kaggle dataset are visualized, analyzed using Exploratory Data Analytics (EDA) techniques. By feeding EDA outcomes into various Deep Learning models: Long short-term memory (LSTM), Convolutional Neural Networks (CNN) yields accuracy of 96.6% in classifying the audio-events.

INDEX TERMS Natural language processing (NLP), deep learning, audio, recording, CNN, LSTM, classification, prediction.

I. INTRODUCTION

In the physical world, the occurrence of any physical event would bear with it a sound, particular to that event only. Be it the sound of a stone falling to the ground, a river flowing, a bird chirping, a person walking, a road being constructed etc. It is thus fair to take into account, just like an event with which positivity can be associated (crowds cheering, friends greeting each other, celebratory fireworks etc.) has a sound paired with it, negative events (a road accident, a landslide, a gunshot etc.) also have specific sounds associated with them as well. It is worth pondering how great would be a system which would be able to detect ambient noise and judge whether it is related to something positive or something negative. Even if direct applications as described here are not available to mankind's everyday use, the technology which is required to make it a reality already exists in one form or the other. Before directly discussing the prospects of natural language processing in the domain of security, one needs to understand the current scenario, and where and how sound/text detection/analysis is done by NLP and used by mankind in modern times.

Modern smartphones equipped with Artificial Intelligence based voice recognition systems are classic examples of the application of speech-based natural language processing in our day-to-day lives. With the likes of Google's Voice Assistant [23], [24] available on every android powered smartphone which searches the web and brings the necessary information in front of us without us typing a single search keyword, or the evolution of Apple's Siri [24] through the various versions of the iconic iPhone series which primarily works as a personal assistant for its users, or the availability of Amazon's Alexa [26] as a smart home voice assistant which can be used to control virtually any smart electrical device in our home, or the development of Samsung's own artificial intelligence-based personal assistant named Bixby [25],

The associate editor coordinating the review of this manuscript and approving it for publication was Felix Albu.

natural language processing has already come a long way in the domain of voice/text-based language processing for the betterment of human lives, all while keeping at par with the compatibility with the latest technologies and hardware.

Taking a few instances of how natural language processing with artificial intelligence has impacted our lives in recent times, one can take the example of how Google's Voice Assistant works. If one needs to just set an alarm, he/she can literally speak so, and the alarm would be set by the AI-based assistant. If one is driving a car and wants to look up the direction to the destination, one can just ask for it by just doing so after saying "Hey, Google"! The examples of such applications are numerous. It is virtually not necessary to touch the phone even for dialing another person's phone number so as to call that person. One can just ask Google to do so, and it would be done in no time.

Another example of the application of natural language processing is in the field of textual data analysis and classification. NLP can be used to determine the voice, tense, and type of a sentence, which can in turn be used to determine either the next part of the entire text or the emotion [28], [29] that the text is trying to convey. Google, for in-stance, uses a very similar application in their search engine where they start suggesting search queries to users as soon as they start typing something into the browser's search bar. The relevancy of the search result is also determined by NLP itself. Taking another instance, social media platforms like Twitter and Instagram use NLP for sentiment analysis and emotion detection and use the acquired insight to show their users similar posts on their timelines.

Now, observing the scenario presented above, NLP can be used to enhance the safety and security of individuals or a population as a whole, in many different ways. Both sound and text-based NLP can be used in multiple scenarios to provide different kinds of security solutions.

Taking the scenario of NLP on social media platforms [27], they can be of great help to security and emergency workers in times of crisis like a natural calamity (like a flood, landslide, tsunami, cyclone etc.) or man-made disasters (like terrorist attacks, hostage situations, aviation emergencies, civil accidents etc.). Since events of such kinds would stir heavy discussions on social media, NLP techniques can be put to work, which can keep track of posts of similar kinds, and if any post related to an emergency is detected on the social media platforms, then it would be re-directed to the appropriate emergency services with the help of customized algorithms. This way, emergency responders would have entire situational awareness, and would be able to act relatively more quickly than possible before, and would be able to help the victims of the situation in a more informed and better way, possibly minimizing further damage as much as possible.

A similar solution can be attained with the help of sound-based NLP techniques, where the sound detected from the surrounding would be able to draw insights into the actual situation of the victim. For instance, if a fire breaks out in a building, and residents get trapped inside, then with the help of sound detectors in appropriate locations inside the building, an automated application based on NLP could be used to detect if a person is in immediate danger of getting burned, from the ambient noise of the fire and the screams of the victim, and emergency services could be sent in his/her direction with definitive motive, without wasting time looking everywhere for the victim.

Natural disasters aside, individual human beings also face many dangers and perish too, which can be a direct cause of either an unfortunate accident or a peril caused by one human being to another (like a homicide). In most situations alike, the victim dies as the situation is not conveyed to the emergency services on time. Also, people working/walking late at night, especially women, in remote areas also face the danger of being robbed, assaulted, or being murder. Situations like these, call for a system which would be able to access the ambient noise of a person and detect whether he/she is in danger or not. A system similar to the one described here has been worked upon by the researchers in [1], but there, the system was primarily based on a hard-ware model and had to be worn at all times in order for it to work properly.

In this research, a software-based system is built as depicted in Figure 1 would be able to detect whether a person is in any dangerous situation or not by analyzing his/her surrounding noise. If dangerous situation is detected, then an automatic and immediate alert would be issued to the registered contact (or the emergency services

The research starts with understanding related studies in the process of event classification and adoption of various machine learning, deep learning models in the Technical Background Section II concludes by mentioning research gaps in the current literature. Proposed live event detection methodology of Section III depicts the overall procedure in identifying and classifying the real time events based on surrounding noise/sound and the understanding, explanation of the dataset is depicted in Section IV, which consists of over 9000 different audio clips, spread across 13 different classes of audio. The dataset is explored Section IV, by looking into its time-domain form, before transforming it to the frequency domain using Fast Fourier Transform and sampling it to 44.1 kHz. The Decibel Spectrogram and the Mel-Spectrograms are used to visualize the data. The Mel-Spectrogram form of the data is used by the deep learning models.

The data is then cleansed in Section IV-A, and for each audio signal, an audio envelope is created for better analysis by the deep learning models. The next step is to train the three deep learning models (1D-CNN, 2D-CNN, and LSTM) in section IV-B, IV-C, IV-D on the cleansed dataset, and to analyze the output.

Lastly, the live audio recording module is integrated with the prediction module Section IV-E, which listens for any sound from the person's surroundings, and sends the recorded

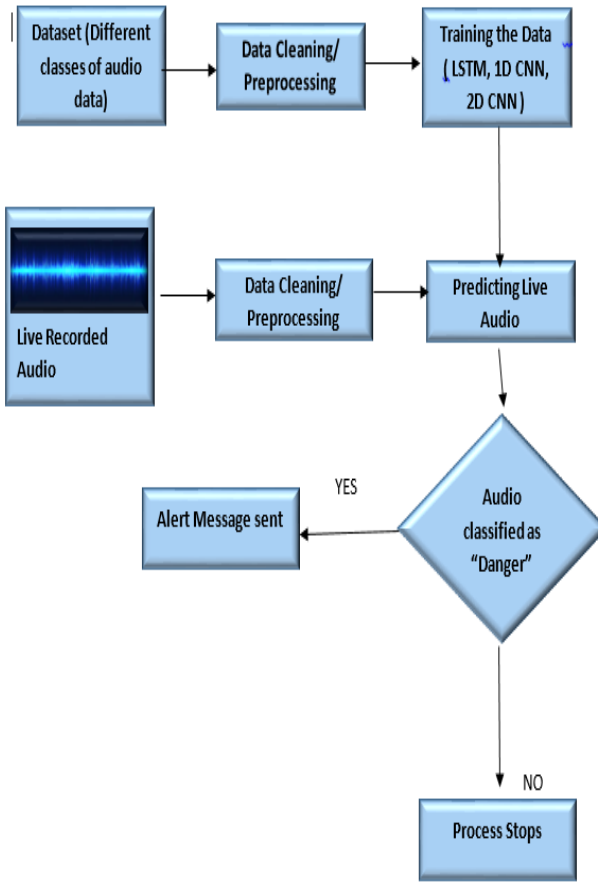

FIGURE 1. Flow of live event detection.

sound to the prediction module. The prediction Module then classifies the sound to be related to danger or not. If the sound is classified as related to a dangerous environment, then an automatic alert message is sent to the registered contact or emergency services via email (with an attachment containing the recorded audio), SMS, and WhatsApp. Lastly, the results and conclusion are discussed in Section-V, and Section-VI respectively.

II. TECHNICAL BACKGROUND

The topic of audio analysis and sound classification is at the heart of speech recognition technology in the modern times. However, this technology could also be used for increasing the safety of individuals or a population from dangers around them. All the current literature analysis related to the techniques adopted for the prediction, classification and detection of live input audio is summarized.

For instance, in [1], Raspberry-Pi based IoT device that includes a camera, sound sensor, GPS (Global Positioning System), and GSM (Global System for Mobile Communications) module. When a scream is recorded by the sound sensor, the SVM algorithm is able to recognize it. The camera is then turned on, which records a 30-second video clip and sends it to the closest police station or emergency services. The GPS module continuously tracks the victim's location, which is then transmitted to the emergency services together with the camera footage using the GSM module.

While the above method focus more on the application of IoT (Internet of Things) along with sound detection, the researchers in [2] have based their experiments on the similarity of the input to a group of learned prototypes in a latent space and utilize a frequency-dependent similarity measure that is built by taking into account different time-frequency resolutions in the feature space. Voice, music, and background noise are three different sound categorization tasks that the proposed model is capable of handling. Here, a Deep Learning Model containing a Prototype Layer, a Similarity Measure and Weighted Sum Layer, and a Fully-Connected Layer is utilized to extract insightful information from the input sound. Although this research is not furthered towards the application of the designed system in the field of individual security, it holds a great potential for the same.

Utilizing an ARM (Advanced RISC Machine) controller and an android application, the system proposed in [3] synchronizes the device and the smartphone using Bluetooth so that each may be turned on independently. Every two minutes, the device may send alert calls and messages to the pre-set contacts along with its current location, record audio for further analysis, and be followed in real time through a mobile application. An additional distinctive element of the system that one might employ to protect their privacy is a hid-den camera detector. Its major advantage is that it can be used to protect women against crimes including stalking, domestic violence, physical assault, and intrusive hidden cameras.

Focusing more on the domain of security in the urban scenario, one can study the research done in [4], where the researchers have gone by the domain of security in a slightly unorthodox manner. By identifying Unmanned Aerial Vehicles (UAVs) in loud outdoor and interior contexts, which have recently been utilized to carry out or support terrorist operations, the proposed effort aims to increase urban safety. Deep neural network-based techniques that can identify a UAV's spectral signature are used to detect UAVs in addition to sensors that might measure the sound they made.

With the approach provided by [5], a method for categorizing sounds, a thorough classification of the various noises present in an urban area is achievable. The result can be used to generate insights into the different kinds of activities going around in an area, and further be used to detect whether any individual/group of people is in any dangerous situation or not. The log-Mel spectrogram's FBank feature is first developed for auditory representation. A series of FBank feature vectors created from distinct acoustic signal frames are then used as input to a Convolutional Neural Network (CNN) for urban noise identification. Here, the traditional LPCC (Linear Prediction Cepstral Coefficients) and MFCC (Mel-Frequency Cepstral Coefficients) acoustic feature, the FBank image feature, the hierarchical extreme learning machine (H-ELM), and the multilayer extreme learning machine are integrated

with the support vector machine (SVM) and the extreme learning machine (ML-ELM).

For emotion recognition from speech, [6] looked at how noise affected two popular SER (Speech Emotion Recognition) architectures, Acoustic Features and End-to-end, as well as the potential benefits of implementing speech enhancement in SER applications, particularly in low SNRs. This system's ability to recognize speech (a sort of noise) even at very low Signal-to-Noise Ratios (SNRs), or for poor input sound quality, is a significant benefit. In this research, a number of SER techniques based on SVMs and openSMILE features are employed. The approach is based on stacked residual blocks of 2D convolution layers, which have been shown to efficiently learn rich representations of input signals in the past.

A major usage of natural language processing is in the field of sentiment analysis and emotion detection too. In [7], a method for sentiment (feeling) analysis that is non-predictive a priori and can handle audio recordings of arbitrary length is proposed. Mel spectrogram and Mel Frequency Cepstral Coefficients are used as audio description tools, and a Fully Convolutional Neural Network architecture is recommended as a classifier. An FCN architecture (Fully Convolutional Neural Network) is suggested by this study in order to classify audio files of any length and recognizing emotions in close to real-time. An FCN is primarily a CNN without fully connected layers that employs just convolutional layers and up- or down-scales input data to enable the system to accept variable input data.

A study similar to the aforementioned one is conducted by [8]. Built on significant elements gathered from several case studies, a Generative Model for NLP Applications is pro-vided in this paper. The generative model serves as a unified framework for several NLP disciplines and may address specific difficulties reading text, hearing speech, comprehending it, gauging mood, and determining the essential elements. The study proposes a model for a smart virtual assistant that might include the best traits from each case study looked at for an improvement over the present NLP models in addition to having the ability to understand more challenging languages like the Chinese language. The system's capacity to recognize speech and emotion in multiple languages can be a huge benefit in the security sector since it allows for the detection of threatening or abusive speech delivered to a person in different languages.

A study on emotion detection from text/speech is also conducted by [9], where it uses neural networks and automatically determine the speaker's emotions by analyzing vocal cues. In order to analyze texts/speech with multilingual forms utilizing cross-language functions and the lexical level function, a hybrid neural network made up of CNN and Bi-LSTM subnets is utilized. This network also identifies emotions in cross-language vocals/writings. The system may be utilized as a software-only pro-gram to automatically identify threat calls and so improve a person's security.

Event detection by analyzing posts on social media is also one of the major applications of NLP, and can be used to enhance security measures/operations by providing early information to emergency workers, who in turn can reach out to the victims be-fore the severity of the situation becomes graver. A similar attempt has been made by the research done in [10]. In order to select the language processing models striking the best balance between accuracy and processing speed for text-based natural language processing in the urban context, the researchers in [10] conducted a preemptive evaluation by contrasting several baseline language models previously used by researchers for event classification. To achieve the desired results, a number of algorithms are applied along with the pre-defined NLP models, including MNB (Multinomial Naive Bayes Classifier), CNB (Complement Naive Bayes Classifier), RF (Random Forest Classifier), Multiple Regression Analysis, General Regression Statistics, and ANOVA.

Similar to aforementioned method of event detection, [11] presents a simple yet effective method for social event recognition that mostly utilizes natural language processing. The researchers look at the distinctive characteristics of social media's natural language in order to select the most suitable characteristics. Second, they mix fundamental machine learning techniques with NLP methods to do classification and extract features. The bag-of-words (BoW) model, one of the methods employed in this paper, may be used to describe a text using the frequency of terms found in a dictionary. BoW completely disregards word order or structure, which is a highly powerful approach to communicate messages. The Support Vector Machine (SVM) algorithm is another one that is applied in this work. This system has the capacity to identify social events from short, hazy, and nonstandard English-written social media messages.

In [12] too, the researchers present an analytical framework for the analysis of tweets in order to identify and categorize specific information about a disaster, such as affected people, damaged infrastructure, and disrupted services, and to distinguish impact areas and time periods, as well as the relative prominence of each category of disaster-related information across space and time. Here, Latent Dirichlet Allocation (LDA) is employed in an unsupervised multi-label categorization of tweets utilizing LSTM (Long Short-Term Memory) networks.

The scope of NLP in the domain of security can be further extended to its usage inside modern AI-powered self-driving vehicles also. In [13], an image and audio-based solution is provided as a service to increase the security and trust within an autonomous shuttle. It is backed by special Artificial Intelligence (AI) algorithms. The two modalities allow for the real-time identification of small criminal scenarios, such as screaming, bag stealing, altercations, and vandalism. They also provide notifications to authorized personnel for necessary action. For audio classification, a two-dimensional Convolutional Neural Network (CNN) is employed, and for

visual analysis, an LSTM classifier that can perform binary or multi-class SoftMax classification is used. A rider's safety is maintained as an advantage of this system.

Short sounds/noises, which are needed to be detected very quickly, are often linked with dangerous situations. Thus, it calls for a system which would have the capability to do so. Reference [14] proposes pre-trained audio neural networks (PANNs) that were trained on the substantial Audio Set dataset. These PANNs take on additional audio-related duties - modeling the computational complexity of PANNs and investigating their performance using a variety of convolutional neural networks. Convolutional Neural Networks (CNN) and other methods for data balance and augmentation are the major techniques utilized for building PANNs. This study specifically uses the augmentation methods Mix-up and Spec Augment. PANNs can identify sounds with enormous accuracy, which significantly decreases the work required from humans to accomplish the same. The speed at which PANNs operate also makes them more suited for swiftly and precisely classifying common noises.

Returning to the classification of sound/noise, which can be classically used as the base for the detection danger from the noise around an individual/population, [15] demonstrates that sound categorization performance can still be improved by swapping out the recurrent architecture for a parallel processing structure during feature extraction. The research processes the huge data and uses it to develop the model using Deep Learning Algorithms, namely CNN (Convolutional Neural Networks) and LSTM (Long Short-Term Memory). A stack of L identical blocks with their own set of training parameters makes up the feature-extracted model is used. This study compares SVM to LR (Logistic Regression) and KNN (K-Nearest Neighbor), two other classifiers, and discusses the advantages of SVM as a classifier. The studies' findings demonstrated that the suggested technique may greatly improve sound classification accuracy, further enhancing the cause of improving individual security.

Exploring some unconventional usage of NLP, [30] provides the perfect example. After perimeter defenses (such as a firewall and network-based intrusion detection system) have failed or been circumvented, a host-based intrusion detection system (HIDS) is a useful final line of defense against cyber security threats. Since Security Operation Centers (SOC) of enterprises rank HIDS as one of the top two security tools, HIDS is widely employed in the business. For industrial companies, having a highly effective and efficient HIDS is ideal, however, when sophisticated attack patterns evolve, HIDS performance deteriorates due to various issues (e.g., a high false alarm rate that wears out SOC employees). An increasing number of HIDS are utilizing the advancements in Natural Language Processing (NLP) techniques, which have demonstrated effective and efficient performance in accurately detecting low-footprint, zero-day attacks and predicting an attacker's next steps. This is because NLP methods are better suited for identifying complex attack patterns. An integrated and thorough body of information about NLP-based HIDS is required given the current research trend of using NLP in HIDS. Notwithstanding the rapidly increasing usage of NLP in HIDS development, not much effort has been made to systematically examine and compile the peer-reviewed literature that is currently accessible in order to comprehend the role that NLP plays in HIDS development. Reference cite30 conducted a Systematic Literature Review (SLR) of the works on the end-to-end pipeline of the application of NLP in HIDS development since there was a dearth of a synthesis and a complete body of information on this crucial issue. Reference [30] identifies, taxonomically classifies, and systematically compares the state-of-the-art NLP techniques used in HIDS, attacks identified by these NLP methods, datasets, and evaluation metrics that are used to assess the NLP-based HIDS for the end-to-end NLP-based HIDS development pipeline. To assist the HIDS developers, [30] emphasizes the pertinent best practices, issues, benefits, and drawbacks, and also provides the planned future research paths for the development of NLP-based HIDS.

In [31], the researchers provide another instance where deep learning is used for sound classification in the urban landscape. They claim that building habitable and sustainable cities is severely challenged by the world's rapid urbanization and population expansion. Urban noises are increasing and becoming more diverse as a result of this increase. Since noise is central to the idea of smart cities, [31] turned these noises into information rather than merely being heard. Two fundamental techniques are utilized to categorize urban noises for this purpose. In the first of these, the sounds are subjected to signal processing techniques in order to extract handcrafted qualities. The alternative approach uses deep learning models to classify sounds based on their visual representation. This study looked at how different variables utilized in both approaches—individual and hybrid—affect how urban sounds are classified. Furthermore, a CNN model for hybrid feature classification was developed. The outcomes demonstrated that both strategies were successful in classifying data. Mel-spectrogram, scalogram, and spectrogram pictures yielded the best categorization success rate among the visual representation techniques. Accuracy was positively impacted by using the SVM classifier, mel-spectrogram, and auditory features. Datasets from UrbanSound8k and ESC-10 were used for the experiments. When utilizing the AVCNN model with the scalogram and acoustic characteristics, the ESC-10 achieved the greatest accuracy of 98.33%. By utilizing the SVM classifier to categorize the mel-spectrogram and acoustic characteristics derived from the AVCNN model, the maximum accuracy of 97.70% was achieved for UrbanSound8k.

The researchers in [32] focus their research on the Synthetic Polyphonic Ambient Sound Source (SPASS) dataset, a freely accessible source of synthetic polyphonic audio. SPASS was created to efficiently train deep neural networks for the purpose of detecting polyphonic sound events (PSED)

in urban sound environments. The five virtual areas that makeup SPASS are park, square, street, market, and waterfront. Following a hierarchical class taxonomy, a variety of monophonic sound sources were curated, virtual environments were set up using the RAVEN software library, all stimuli were created, and the data was processed to produce synthetic recordings of polyphonic sound events along with their corresponding metadata. The collection has 25,000 stimuli of 10 seconds each, or 5,000 audio clips per environment, virtually recorded at a 44.1 kHz sampling rate.

In this research, the audio analysis technique adopted is the Fourier Transform and Mel-Spectrogram (similar to [31]), and the audio was sampled at 44.1kHz (just like in [32]) for further processing. Post-cleaning, the sound data is subjected to three different deep learning models (1D-CNN, 2D-CNN, and LSTM) for the classification of sound from a person's surroundings (the likes of which have been used in various pieces of research cited above, for example: [15]), and to detect a threat from it. If the threat is detected, then an automatic alert message is sent to the registered help or the emergency services. Moreover, in general, research works like [34], [35], [36], [37], and [38] etc. have been referred to during the course of this work in order to generate more insights into how audio analysis and classification is done using different analysis techniques and deep learning models respectively.

The research gaps identified from the existing literature are depicted as below:

- Starting with [1], it provides the necessary solution for the problem of detection of threat around an individual, but it comes with a bulky hardware, which poses a difficulty in carrying it around for regular use.
- On the other hand, in [2], [15], and [32], the researchers use several techniques to analyze audio signals, but they don't further their work to provide a practical solution to the problem of danger detection around an individual.
- The research done in [3] is somewhat close to what has been achieved in this research, where the researchers have built a system to provide real time feedback from a person's surroundings, however, this also comes with an additional hardware component in addition to a smartphone.
- In [4], the researchers use noise detection to detect Unmanned Aerial Vehicles (UAVs) which might be used for criminal activities. However, this approach is not favorable to be applied at an individual level, and would not be suitable for detection of threat around an individual human being.
- Reference [5] also used similar techniques to detect threat for an individual/group in an urban context, but is unable to provide a solution to make the user friendly at an individual level with the use of no hardware.
- The research done in [6] and [7] focus on detection on the detection of emotion from speech, which can be helpful in determining whether a person is in agony or not, or if a person is being verbally threatened by another fellow human being or not. Systems like these, although beneficial, are unable to address the problem of physical safety of an individual.
- A very similar system is proposed by [8], where the speech detection is done for multiple languages to detect verbal threats, but not physical ones.
- Some other systems, like those proposed in [9], [10], [11], and [12], detect emotions and events from texts/speech, social media posts and tweets respectively. Although these are unique approaches to determine an individual's/group's live situation, they again fail to address the challenge of physical individual safety.
- Moving further, [13] provides a unique sound-detection-based approach towards safety of travelers inside a vehicle, but do not address the safety concerns for those individuals who are alone and not inside any vehicle.
- Reference [14] provides yet another approach towards the detection and classification of short sounds/noises, which can actually become very useful for approaches like the ones that this research has proposed, but the researchers in [14] do not further their research towards any practical solution to the problem of physical individual safety.
- References [30] and [31] use noise detection for perimeter defense techniques (like intrusion detection), and conversion of urban sounds into information respectively, but do not address how an individual can be helped with respect to physical threats.

III. PROPOSED LIVE EVENT DETECTION METHODOLOGY FOR INPUT AUDIO CLASSIFICATION

The main objective of the proposed system is to detect and classify the victim's live audio signals for immediate rescue. The system is intended to deliver its excellence as an application in any smartphone and it uses the default microphone configuration of it. On detection of suspicious audio patterns from the live input audio from microphone, the geographical location of the victim is shared to the emergency contacts in the phone as well as to the police patrol. The drawbacks inferred from the current violence detection scenarios related to audio event detection and classification accuracy are addressed for effective functioning which plays a vital role in avoiding false event classifications meanwhile ensuring the victim's safety through high classification accuracy. To carry out accurate prediction, training and testing of Kaggle dataset is carried out in 3 machine learning models namely LSTM, 1D CNN and 2D CNN which is illustrated in Figure 2.

The audio dataset used in this work consists of 13 classes (types) of audio signals, namely air conditioner, car horn, children playing, dog bark, drilling, engine idling, fire crackling, glass breaking, gunshot, jackhammer, scream, siren, and street music. Of these 13 classes, fire crackling, glass breaking, gunshot, and screaming are identified as audio types related to a potentially dangerous environment.

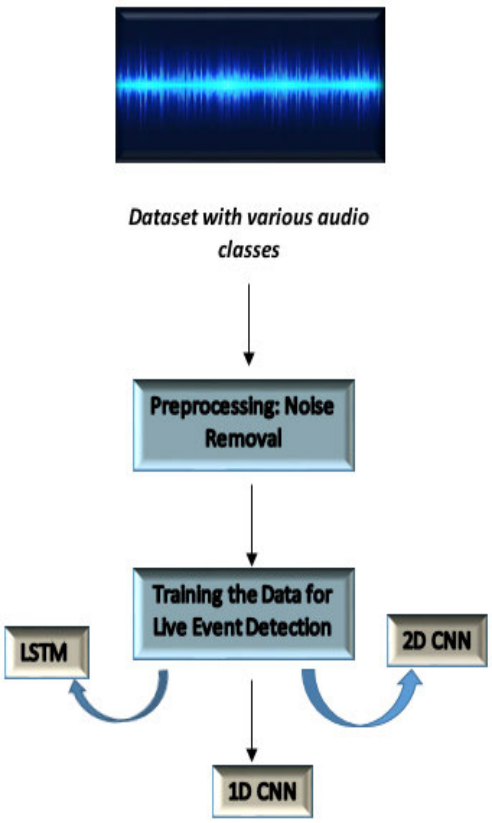

FIGURE 2. Training the dataset using ML models for live event detection.

The dataset described here is a customized dataset and is built by taking into account and combining parts of data from three different audio datasets in order to meet the requirements of the problem statement under research. The three datasets mentioned here are the [16] UrbanSound8K dataset, [17] – which is a dataset with audio data for 50 different environmental sounds, and [18] – which is a dataset for the 'screaming' noise as this sound type is a crucial one for the requirement of the project, which focuses on the detection of threat from the ambient noise.

Although the three original datasets from which the dataset for this project has been built, have over 50 different classes of audio signals, only 13 are kept for the final analysis as it would take less computational time for only 13 classes to be trained on the three deep learning models built during the course of this project, as compared to the larger size of a dataset with over 50 classes, taking into account the fact that this research focuses more on the successful development of a working prototype than a production-level software.

IV. RESEARCH METHODOLOGY FOR VISUALZING AND CLASSIFYING AUDIO SIGNALS

After training the model with Kaggle dataset, exploratory data analytics need to be carried out on the audio signals for easy visualization and classification of general audio categories. A change in a certain quantity over time is referred to as a signal. A few of the considered categories of audio signals are depicted in Figure 3. Air pressure is the variable quantity for audio. The air pressure may be measured over time with samples. In general, audio data is sampled at various rates, but most frequently at a rate of 44100 Hz, or 44100 samples per second, and becomes very difficult to visualize and work with in the time domain.

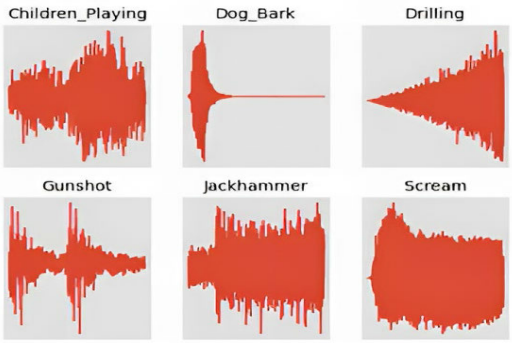

FIGURE 3. The audio signals in the time domain for one sample audio each from 6 of the 13 classes considered in the research.

Thus, audio data is preferred to be transformed to the frequency domain using a mathematical method called the Fast Fourier Transform (FFT), and the plot obtained from it is called the periodogram. This is so because numerous sound waves with a single frequency make up an audio signal. The resulting amplitudes are simply measured while periodically sampling the signal. The Fourier Transform makes it possible to separate a signal into its individual frequencies and amplitudes. To put it another way, it changes the signal from the time domain to the frequency domain. The result is referred to as a spectrum. This is achievable because any signal can be decomposed into a set of sine and cosine waves that sum to the original signal. The Fourier's Theorem makes this precise claim. The Fast Fourier Transform (FFT) method can be used to quickly compute the Fourier Transform. In signal processing, it is frequently employed. The general formula for a Fourier Transform is as given below, and the same can be used to compute an FFT on machines as well:

$$F(x) = \int_{-\infty}^{\infty} f(x)e^{-x}dt \tag{1}$$

Another concept which comes into play while sampling audio signals is the Nyquist Frequency, which considers exactly the half of the maximum sampling rate. In this research, the initial audio data is sampled down to 16000 Hz from 44100 Hz for easier analysis, and thus the Nyquist Frequency achieved in this case would be 8000 Hz (8000 samples per second). In other words, when the live audio recording module is used, the highest frequency captured by the microphone would be 8000 Hz, and all other frequencies above this threshold would be discarded. The periodogram of this signal would reach a maximum of 8000 Hz on the Frequency axis (X-axis). Figure 4 and 5 shows the audio signal for a gunshot in the time domain, and in the frequency domain after the application of the Fast Fourier Transform.

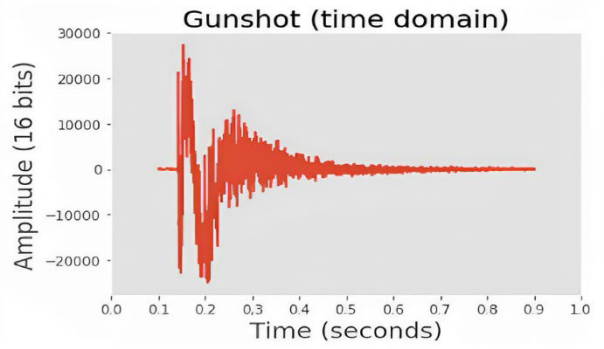

FIGURE 4. The audio signal for a gunshot in the time domain with a sampling period of 1 second on applying FFT.

It can be verified that the Nyquist Frequency achieved from the above gunshot representations are 8000 Hz, which is exactly the half of the sampling rate considered here i.e., 16000 Hz. The Fast Fourier Transform is a useful tool for examining a signal's frequency content, but what makes this approach stand out is when the signal's frequency content shifts over time. This is how the bulk of audio transmissions work, including speech and music. The science community call these signals non-periodic signals. To show how the spectrum of these signals evolves over time, a technique is needed, called the short-time Fourier trans-form, which is a method for computing many spectra by applying FFT to a number of windowed signal segments. To put it another way, Short-Time Fourier Transform (STFT) is a method for taking into account a lot of FFTs and stacking all the periodograms to produce a new visual metric known as the decibel spectrogram, which is created when the FFT is calculated on overlapping windowed portions of the signal (audio signal in this research). The general formula for an STFT is as given below:

$$X(\tau, \omega) = \int_{-}^{\infty} x(t)w(t-\tau)e^{-i\omega t}dt \qquad (2)$$

A spectrogram, sometimes known as a decibel spectrogram, is a visual depiction of the ''loudness'' or signal intensity (or amplitude) over time at various frequencies included in a particular waveform. A spectrogram is effectively a collection of FFTs stacked on top of one another. Some more information is being processed in the back-ground while the spectrogram is being calculated. This may be regarded as the amplitude's log scale. The color dimension is converted to decibels, and the y-axis is changed to a log scale. This is because humans can only sense a very limited and restricted range of frequencies and amplitudes. The quantity of energy at various frequencies, such as 2 Hz vs. 10 Hz, as well as how it varies over time, may be seen. In several scientific disciplines, spectrograms are widely used to display the frequencies of sound waves produced by humans, machinery, animals, whales, airplanes, etc. and recorded by microphones. In order to distinguish and categorize distinct earthquake types or other ground vibrations, the seismic community is increasingly using spectrograms to analyze the frequency content of continuous signals acquired by a single seismometer or a group of them.

The decibel spectrogram helps in visualizing how the signal changes over time (on the x-axis) along with its intensity, and the frequency (on the y axis). It is worth noting that the number of samples considered while creating the decibel spectrogram using the STFT is exactly half the number of samples considered for the FFTs (in accordance with the Nyquist Theorem) which is depicted in Figure 6.

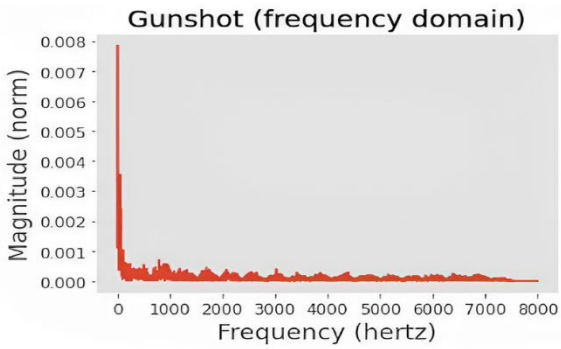

FIGURE 5. The audio signal for a gunshot in the frequency domain with a sampling period of 1 second on applying FFT.

From the above Figure 6, it can be seen from here that as the time changes along the x-axis, the intensity (loudness) changes along the y-axis (Example: the four vertical lines in the plot for the class ''scream'' signifies that at that part of the audio signal the person screams for four distinct instances).

It's crucial to take the Mel FilterBank into account to further the conversation on audio data analysis. Studies have shown that humans do not perceive frequencies on a linear scale. Humans can discriminate between lower frequencies more easily than higher frequencies. Humans can readily differentiate between 600 and 1200 Hz but will find it challenging to discern between 15,000 and 15,600 Hz, even though the difference between the two pairs is the same. However, for deep learning models to operate accurately, it is essential for them to be able to distinguish even between signals with the smallest of differences in frequencies at the higher end of the audible spectrum too. For this purpose, the Mel FilterBank on the Mel scale can be used to rephrase and rescale the audio signal for aiding in the accurate training of deep learning models. Reference [19] Stevens, Volkmann, and Newmann developed a unit of pitch in 1937 so that the listener would perceive equivalent distances in pitch as equal lengths. It is known as the Mel scale. In order to translate frequencies to the Mel scale, mathematics is needed.

The working of Mel spectrogram for a sample audio is depicted in Figure 7. To understand better how the Mel scale is related to the frequency, for lower frequencies as the frequency changes, the difference reflected on the Mel scale is quite considerable. However, as the frequency increases, the large differences in frequencies tend to yield smaller changes on the Mel scale i.e., the lower frequencies are given more importance than the higher frequencies. Thus,

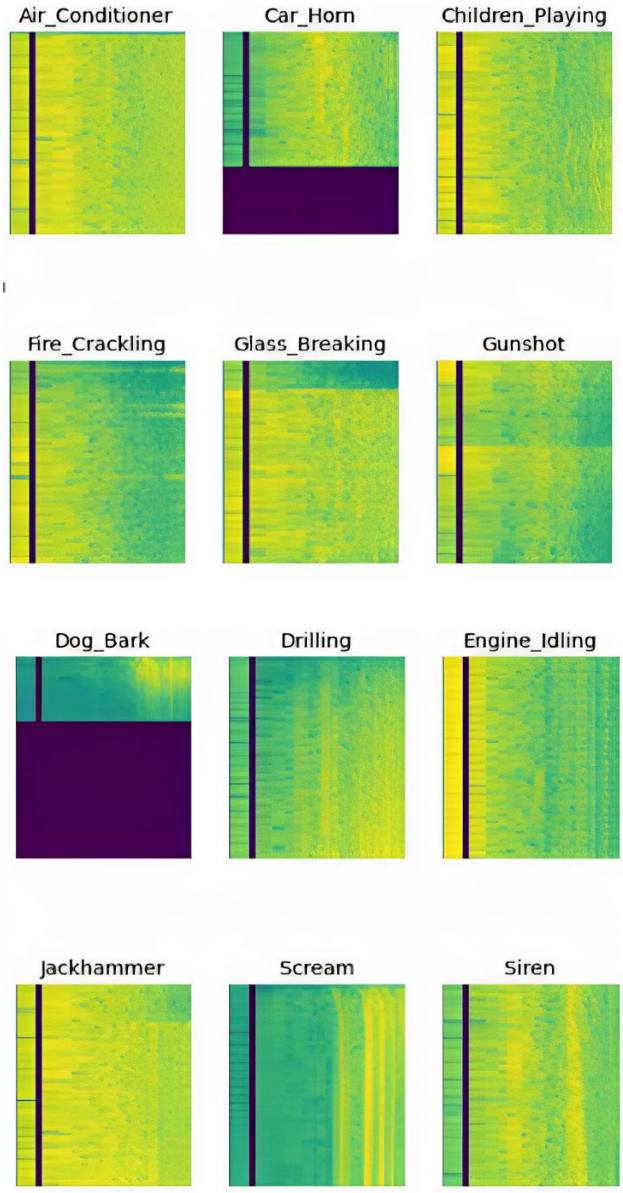

FIGURE 6. The decibel spectrogram for one sample audio each from 12 of the 13 classes considered in the research (the same as considered in the time domain plots above).

the Mel FilterBank is a sequence of several triangular filters, whose number can be varied as per requirement (128 bands used in this research – change in the number of bands has an effect on the memory requirement for the analysis), which are passed over the audio signal in the frequency domain (periodogram). Each of the triangular filters in the sequence corresponds to a specific frequency band, and the FilterBank as a whole decomposes the entire audio signal into separate frequency bands in the Mel Frequency Scale.

To summarize, when the frequencies are translated to the Mel scale, the spectrogram is called a Mel spectrogram. When audio signals are windowed in time, Mel-Spectrogram adds a bank of frequency-domain filters to the signals.

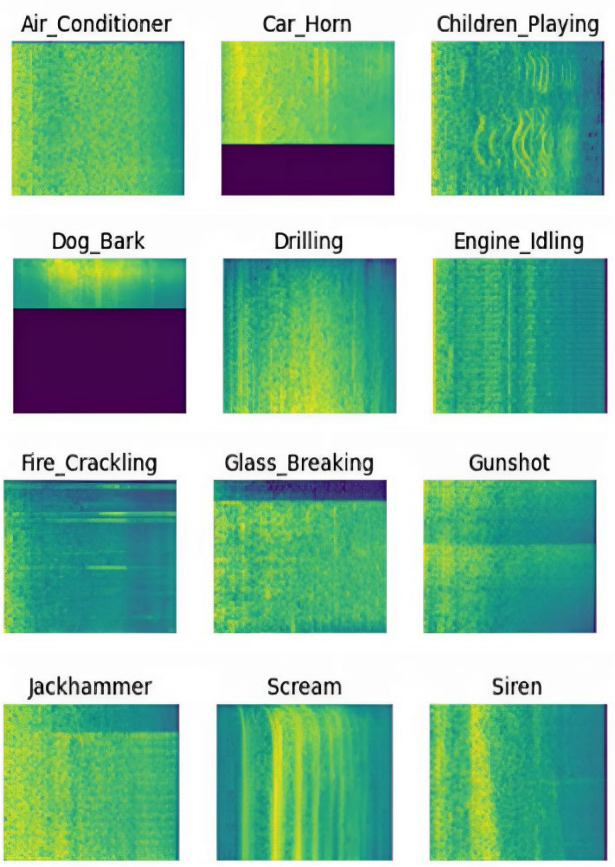

FIGURE 7. The Mel spectrogram for one sample audio each from 12 of the 13 classes considered in the research.

Furthermore, a Mel-Spectrogram has two main differences as compared to the regular Decibel Spectrogram – (i) The Mel Scale is used instead of frequency on the Y-axis, (ii) The Decibel Scale is used instead of amplitude to define the colors in the plot. Research has shown that Mel-Spectrogram performs better as inputs for deep learning models instead of Decibel Spectrogram when considering audio data, and the same has been implemented in this research as well.

A. CLEANING THE SOUND DATA

For the ease of specifying the classes of each of the audio data files, the audio files from the dataset are kept in respective folders, where the name of the folder specifies the class of the files inside. For this research, 13 folders are created (as 13 audio classes are being considered for training), and these class-wise folders are kept inside a parent folder, from which the audio files would later be fetched for cleaning, pre-processing and training purposes.

Also, since audio files in the ''.wav'' format are very quick to load into memory, all the audio files are converted to this format before-hand. However, on the downside, since audio files in the ''.wav'' format are uncompressed files, they tend to take up a lot of space in memory. Moreover, as the sound

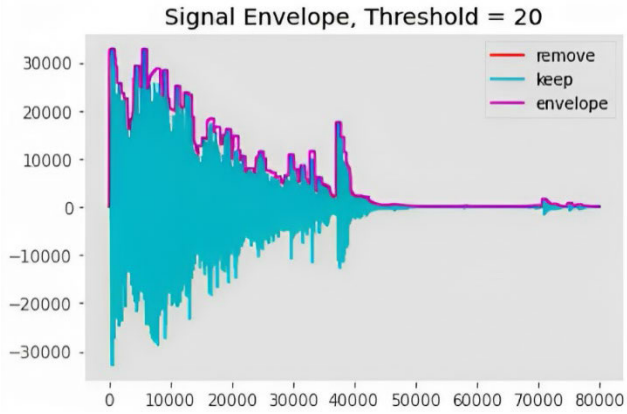

FIGURE 8. The signal envelope for one of the audio signals corresponding to "glass breaking". Here, since the threshold magnitude is considered as 20, any part of the signal below that threshold is not considered, and this can be verified with the help of the non-silent signal envelope created around the signal (visible in purple).

files contain 16-bit audio, the data type considered is NumPy 16-bit integer.

One of the problems that is faced with audio data is that most of the audio is concentrated in one general area in the audio stream, and as the audio progresses, the magnitude becomes so low that a significant part of the signal looks the same as there is a lot of silent area in the audio. The silent zones or dead spaces in the audio can be removed by creating a customized signal envelope, which is essentially what it sounds like – this signal envelope tracks the signal to study how it changes, and considers only the magnitudes above a particular threshold (20 is considered as threshold for this re-search). This is implemented by first converting the signal into a sequence, and obtaining the absolute (positive) value of the signal at any given point of time, and then applying a rolling window over the signal with a specific window length (set as 20 in this case), set for considering the maximum magnitude at any time instance, for creating the signal envelope with the specified threshold is depicted in Figure 8.

The dataset used in this project, as already described, is a collection of audio files across 13 classes. Since all these audio files are a recording of real-world environmental sounds, the format of the audio across each audio file is inconsistent. For most of the instances of audio data, the data available has more than one channel of audio signal with a sampling rate between 44100 Hz and 48000 Hz.

Since this variance across the data is a challenge towards the uniform analysis of the audio data, which in turn might lead to erroneous classification results after train-ng, the data is thus passed through a cleaning process. In this step, the audio data in each of the audio files are converted to a signal with a mono channel, and the sampling rate is down-sampled to 16000 Hz for the ease of analysis during the training phase of the data, including a reduced computation time.

The focus of down-sampling the data is to feed the deep learning models with a clean data to train on so as to have as accurate a learning as possible. However, when the prediction module is run, it is fed with the raw data, on which it then performs a similar data-cleaning step, as would be the case for the practical application of the model on a real-world audio data.

Once the down-sampling and the removal of dead space from the audio signals are done (with the help of the custom signal envelope), each audio signal is split into separate instances with a fixed specified time interval. This time interval has been referred to as the "delta time" (set as 1 second in this work) throughout this paper, and the models train over each of these intervals for learning purposes.

Once the data is cleaned, it is saved in the new form inside a separate directory with the same folder hierarchy (with 13 different directories, each denoting a separate class) as the parent directory used for the original audio signals. These new cleaned audio files from this new "clean data" directory are used later for training the deep learning models.

B. APPLIED DEEP LEARNING MODELS

In this project, three different deep learning techniques have been explored and implemented, namely LSTM (Long Short-Term Memory), 1D-CNN (1-Dimensional Convolutional Neural Network), and 2D-CNN (2-Dimensional Convolutional Neural Network). The primary reason behind the implementation of three different deep learning models is that each of these models has its own set of advantages and disadvantages, which would cater to different requirements with respect to the training, analysis and classification of the data based on the different properties and attributes associated with the dataset being used. The prediction module would have an option to choose between these three deep learning models for running the prediction on the real-world audio signal, based on the requirement.

Once the data is cleaned, the training is the next phase, for which the data has to be loaded into the models with appropriate parameters. For this, a custom data generator is built, both for training and validation, which would prepare (generate) the data to be loaded into the deep learning models for training, after it has been spilt into training and testing parts (a 90-10 split has been done in this work). The input to this data generator is the paths to the clean audio files (created earlier), split into intervals of 1 second, and the corresponding classes for the signals.

Beginning with the best practice of loading data into the system before analyzing with a deep learning technique, it is computationally more efficient to load the data in batches, till all the epochs are completed, instead of loading the entire data to the memory at once. So, the way this is recommended to be done in TensorFlow from Keras' perspective is to inherit the requirements from a class call "Sequence". This class provides the functionality to load the data in batches, and use multiple GPUs (Graphic Processing Units) to process the data through multiprocessing.

Three essential functions are implemented here for generating the final processed data on which the analysis would

be run later – namely ''__len__'', ''__getitem__'', and ''on_epoch_end'' [20]. The ''__len__'' function is used to specify the number of batches per epoch (total number of samples divided by the batch size, which can vary from 16, 32, 64, and so on). The ''__getitem__'' method outputs an ''X'' matrix (a time-series format of the audio data) whose 1st dimension is the batch size, 2nd dimension is the number of channels, and the 3rd dimension would be the total number of data points considered over the specified time interval, and a ''Y'' matrix which in turn would be the output in the form of a SoftMax layer (the probabilities of occurrence of the different classes) – the 1st dimension of this matrix is the batch size and the 2nd dimension is the number of classes considered in the research (13 in this work). The ''Y'' matrix would be later used to build a hot-encoded matrix using the ''to_categorical'' method of TensorFlow. The output from the ''__getitem__'' method is used as input to the Mel-Spectrogram layer in the deep learning models, where the data is fed as audio signals with one channel and 16000 data points for one second of time-series data (sampling rate). Lastly, the ''on_epoch_end'' method is typically used for data augmentation across deep learning projects, however, in this work, it has been mainly used for shuffling the data in between epochs, so that there is different distribution of data for different batches, and the models have a more holistic learning in the end.

Just as previously mentioned that a deep learning model tends to perform better with Mel-Spectrogram audio inputs, the concept of Mel-Spectrogram is applied to the input audio data files before the training of the models begin. For all three models, after feeding the input audio signals, with one channel and a sampling rate of 16000 Hz, a Mel-Spectrogram layer is included, just before the output from this layer is normalized with a 2D normalization layer which rescales the data to 0-mean, for further processing by the models. This layer is a custom layer which is added to the Keras mod-el(s) (the deep learning model(s)), with the help of another Python library called Kapre [20], which is an audio pre-processing library in Python, which allows the implementation of various custom signal processing techniques like STFT (Short-Time Fourier transform), Inverse STFT, Mel-Spectrogram etc. If instead of using Kapre, the Mel-Spectrogram is computed separately, then it has to be performed offline, and stored separately in memory and then the analysis would have to be run – in case some parameters are to be changed, then the entire process has to be re-iterated, and would take up a lot of computational time.

The details of the three models used in this work are discussed in the upcoming subsections.

1) THE 1D-CNN (1-DIMENSIONAL CONVOLUTIONAL NEURAL NETWORK) MODEL

A 1D-CNN performs fairly well when shorter (fixed-length) segments of the full dataset are anticipated to provide interesting features and the feature's location within the segment is not particularly important. This is pertinent to the analysis of time sequences of sensor data, such as that from an accelerometer or gyroscope. It also applies to the analysis of audio signals as well as any other signal data that has been gathered over a predefined period of time. Another use is in natural language processing (NLP), albeit LSTM networks have more promise in this field because word proximity isn't always a trustworthy indicator of a trainable pattern.

1D-CNN uses time distributed layers to wrap the 1D Convolutions over time. In-put to the model is time series format of the data, where the channels are going to be the first dimension. For consistency, a permute layer has been implemented on the dimensions of the input data, which is important to be done as a time distributed layer is being used here, as any time-based layer expects the dimensions of the data to be batched by time, features and channels - so the permute layer just swaps the features and time, and time becomes the 1st dimension.

Next, a time distributed 1D convolution is implemented with a small kernel size and a hyperbolic tangent as the first activation function. The activation used in the following time distributed layer is the ReLU (Rectified Linear Unit), which allows a deep learning model to be non-linear, addresses the vanishing gradients problem, helps prevent the exponential increase in computing required to run the neural network, and only takes into account half of the input data.

Before going any further, it is important to note that the goal here is to develop a functionality of classification, which can be done using a few deep learning layers to build out features from the data that is available (sound data in this case), and reduce them down to the point where a classifier of some sort can be built in the last few layers of the Convolutional Neutral Network, which are called as the head of the neural network.

The same is the case for the 1D time distributed CNN model used here - the number of dimensions is gradually reduced down as the model progresses through the layers, and the number of tuned parameters increase as the number of features are increased in each layer. In better words, the training initially starts with a limited number of features (a general start), and as the training progresses through the layers, more features are added to the network to specify with more granularity of what exactly the network should learn – this is what the last layer with 128 features (the maximum number) does. Wrapping this procedure with a time distributed layer with respect to sound data means that the network is instructed to go along the time dimension, and as it progresses, all the different frequencies from the input audio data is fed to the network, and the neural network can identify useful features from specific frequencies, and use all these features together to learn from it, which is practically not possible for human beings to do. So, the purpose of wrapping a 1D-CNN in time distributed layers is that the model looks at only the frequency spectrum of the audio signal over time

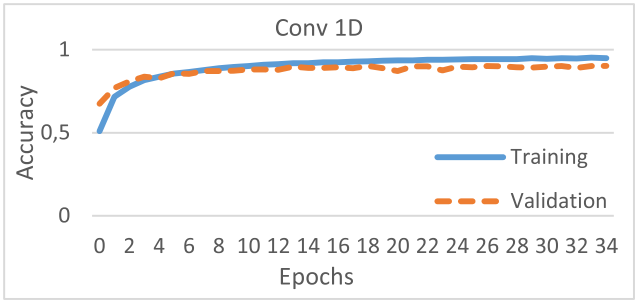

FIGURE 9. On running for 35 epochs, the maximum accuracy reached for the 1D-CNN model is found to be 95.2%, with a maximum validation accuracy of 90.2%. The metric used here for considering the best results is the validation loss. The Training (train) vs Validation (test) can be seen in this figure.

for training. The last 4 epochs of 1D-CNN is depicted in Figure 9.

Furthermore, a 2D GlobalMaxPooling layer is used, which takes only the latest feature size (the last dimension) for training and learning purposes, instead of flattening it out by considering a product of all the features (dimensions) available for the data – it is not concerned with the part of the neural network these features are obtained from, but just uses these features to implement the classification.

Towards the end of the network, a dropout and a regularization layer are used to prevent overfitting, and the flattened data from the GlobalMaxPooling layer is then be passed through a dense layer (with 64 activation units considered in the first dense layer, and the sound classes used in this research as the features in the 2nd dense layer) to build a classifier, and the output is presented using a SoftMax activation function (which converts the initial outputs of the neural network into a probability vector), which helps in the prediction of the class by calculating the probability of each possible outcome. This layer also creates a hot encoded matrix for the implementation of categorical classification.

2) THE 2D-CNN (2-DIMENSIONAL CONVOLUTIONAL NEURAL NETWORK) MODEL

The initial implementation of the standard Convolution Neural Network was made possible by the Lenet-5 design [22]. Conv2D is frequently applied to picture data. It is referred to as a two-dimensional CNN since the kernel moves along two dimensions on the data. The key advantage of using a 2D-CNN is that it can recover spatial information from the input using its kernel, unlike other networks.

The 2D-CNN is quite similar to the 1D-CNN, however the approach can be associated with a Computer Vision based approach, as a similar architecture is used. So, in this case, no time distributed implementation is done, but the entire frequency spectrogram is looked at as a whole, and the neural network learns from features that are next to each other, their interaction with one another, how they generate meaning through a pattern in their occurrence etc. – looking at the frequency spectrogram as a whole, and building features from it for learning purposes.

With steadily increasing numbers of activation units (8, 16, 32…), a sequence of Conv2D (2D Convolution) layers are utilized here, similar to 1D-CNN, with the goal of incorporating more particular characteristics as the model advances through the levels. All of these layers use ReLU as the activation function, with the exception of the first layer, which uses a hyperbolic tangent with the same S-shape as the sigmoid activation function. This function accepts any real value as an input and returns values between −1 and 1, or values centered around 0. Recurrent neural networks perform best when combined with the hyperbolic tangent activation function for tasks requiring speech recognition and natural language processing. All the convolution layers have a MaxPooling layer in between them, which creates a downscaled (pooled) feature map, by determining the maximum value for patches of a feature map.

At the end of the last convolution layer, a flattening layer is used, which considers the product of the number of remaining useful and tuned features from the previous layer for calculating the final number of actual features to be considered for learning purposes – this approach is somewhat different than the GlobalMaxPooling layer used in the 1D-CNN approach, which considers only the last specified number of features, and discards the rest.

It is interesting to note that if the delta time (time intervals) at which the audio files were segregated at the time of data cleaning and pre-processing (considered as 1 second in this research) is changed, then it would cause the number of Mel-bands (1st dimension of the input data) to reduce or the time dimension (2nd dimension of the input data) to increase in value – this is important to consider because this would determine the final dimension achieved at the end of the flattening layer, which is essential to be of a small size, otherwise would create a huge number of parameters to train on, and take up a lot of dynamic memory for processing, which in turn would require more computational power and time, and would eventually make the network slow. The last 4 epochs of 2D-CNN are depicted in Figure 10.

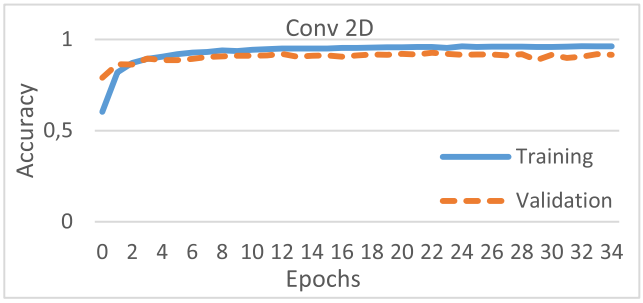

FIGURE 10. On running for 35 epochs, the maximum accuracy reached for the 2D-CNN model is found to be 96.3%, with a maximum validation accuracy of 92.7%. The metric used here for considering the best results is the validation loss. The Training (train) vs Validation (test) can be seen in this figure.

The last three layers are essentially the same as that used in the 1D-CNN model – with a dropout and a regularization layer to minimize overfitting, with two dense layers (with 64 activation units considered in the first dense layer, and the sound classes used in this research as the features in the 2nd dense layer) for building the classifier (using a hot encoded matrix) and obtaining the output (prediction probabilities) using a SoftMax activation layer.

3) THE LSTM (LONG SHORT-TERM MEMORY) MODEL

Unlike conventional feedforward neural networks, LSTM has feedback connections. Such a Recurrent Neural Network (RNN) is capable of analyzing both single da-ta points, such as photos, as well as whole data sequences, like audio or video. This characteristic makes LSTM networks ideal for managing and anticipating data. For example, voice recognition, machine translation, speech activity detection, robot control, video gaming, and healthcare are some applications of LSTM. Applications like connected, unsegmented handwriting identification and others are also possible to utilize it for. Numerous RNNs may pick up long-term dependencies, which is very useful for challenges involving sequence prediction. In addition to processing single data points like pictures, LSTM also features feedback links that enable it to process the full data stream.

A memory cell in an LSTM model is referred to as a ''cell state'' and performs a crucial role in the model by maintaining its state over time. In LSTMs, gates regulate the insertion and deletion of data from the cell state. Information may be able to enter and leave the cell through these gates. The method is aided by a layer of sigmoid neural networks and a pointwise multiplication function. An LSTM's sigmoid layer outputs integers in the range of 0 and 1, where 0 means that nothing should pass through and 1 means that everything should.

The LSTM neural network is specifically designed to study all the features and how they change over time. As done in the 1D-CNN model, the input is again batched by time, features and channels, and a permute layer is used to switch between the time and the feature dimensions. However, since a channel cannot be fed as an input to the LSTM network (mono channel sound being used in this research), it is combined with the feature dimension using a reshape layer.

Before entering the LSTM layers, a time distributed dense layer is used for some initial feature learning (which is not done in standard LSTM networks), using an appropriate number of activation units (64 in this case), and the activation function as a hyperbolic tangent. For this, it uses the feature dimension (128 features considered in this case), and reduces it by half so that the some more relevant features could be learnt about the data even before progressing into the LSTM layers.

The next layer used is a Bidirectional LSTM layer, which computes the gradient descent of learning for the data by going through the time dimension in both the for-ward and the backward direction, which basically means that the model not only studies the sound data only in a forward direction, but also in a reverse manner – this helps in obtaining better gradient descent updates. This layer has lesser number of features (32 in this case) as compared to that in the previous layer. This is so because LSTM networks do not need a large number of features (nodes) to learn relevant in-formation as opposed to the 1D/2D-CNN networks previously discussed. The output of this layer is returned as a sequence, and the feature size returned in this case is double of what was provided in the input (64 in this scenario).

The next procedure done in the LSTM implementation, which is quite common across networks like LSTMs, is called Skip-Connection, where the output features from the time-distributed dense layer are concatenated with the output of the Bidirectional LSTM layer – so in this case, 64 features from the time-distributed dense layer are concatenated with 64 features from the output of the Bidirectional LSTM layer to produce a total feature set of 128. This step lets the neural network take decisions based on both sets of features learnt before and after entering the LSTM. The last 4 epochs for the LSTM model are depicted in Figure 11.

As the network progresses, feature engineering is implemented with the help of two more dense layers with 64 and 32 activation units respectively, with a MaxPooling, and a Flattening layer in between. It is worth noting that the MaxPooling Layer used in this case is a 1D-MaxPooling (instead of 2D-MaxPooling) as the channel in-formation was deliberately lost beforehand when the input was reshaped.

After the flattening layer, a similar approach to building a classifier is implemented as was done in case of 1D-CNN and 2D-CNN, with a dropout and a regularization layer to minimize overfitting, with two dense layers (with 32 activation units considered in the first dense layer, and the sound classes used in this research as the features in the 2nd dense layer) for building the classifier (using a hot encoded matrix) and obtaining the output (prediction) using a SoftMax activation layer.

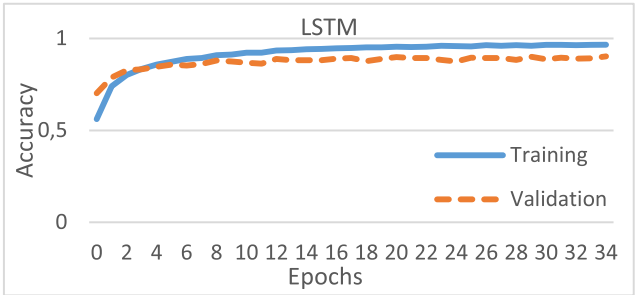

FIGURE 11. On running for 35 epochs, the maximum accuracy reached for the LSTM model is found to be 96.6%, with a maximum validation accuracy of 90.3%. The metric used here for considering the best results is the validation loss. The Training (train) vs Validation (test) can be seen in this figure.

C. TRAINING THE MODELS ON THE DATASET

The previously described dataset that is being used in this project is subjected to a training phase with respect to all

the three deep learning models discussed above. The training on each of the three models takes approximately 3 hours (approximately 9 hours for all the three models) for the dataset being considered in this project.

While training the three models, a call-back is implemented to log the results (validation accuracy/loss and training accuracy/loss) from each epoch (35 epochs considered for each model) into a specific CSV file for each of the models (used later for visualization of the validation and training accuracies). Along with this, a check-pointer is implemented which would monitor the specified parameter (set as validation loss during this research – can be set as validation accuracy as well) from the values logged, and the model would be saved accordingly – since the monitoring parameter was set as validation loss, the lowest validation loss would be saved on the models being trained (1D-CNN, 2D-CNN, LSTM), and the models would work in the best possible way while performing the actual predictions.

On training, the 1D-CNN model turned out to have the least overfitting. This can be attributed to the fact that the features considered in the 1D-CNN are a lot weaker as compared to the 2D¬CNN or the LSTM model. However, looking from another perspective, the LSTM network can also be attributed to being the best network of all the three networks built here, even though the deviation between the training and testing accuracies is higher as compared to that in the 1D-CNN. This gap discrepancy would get reduced further in the LSTM as the size of the data increases, whereas the 1D¬CNN would perform poorly in that case. The following plots show how the three models perform on training, with respect to their training and testing (validation) accuracies.

From the above Figures 9, 10, and 11, although, the 1D-CNN model seems to over fit the least for the given dataset and the number of classes considered as compared to the other two models (the gap between training and testing is the least), the LSTM model, which achieved the highest training accuracy among the three (96.6%), would perform the best when the size of the dataset and the number of classes considered would increase. Once the three models are trained on the given dataset, the models are saved with their training information in the ''.h5d'' format. The user would be able to choose the model to be used for audio class prediction for the real-world audio signal, by specifying the path to the necessary ''.h5d'' file (for 1D-CNN, 2D-CNN, and LSTM) as saved during the training phase.

D. THE PREDICTION MODEL

Since the prediction module takes in real-world audio data for identifying the class of the sound, it needs to pre-process the data before running the analysis on it. For this purpose, the sound data received, after the recording module records and saves the audio data, is down-Sampled to a mono channel with a sampling rate of 16000 Hz. This is similar to the down-sampling step performed during the data cleaning procedure. Moreover, just like the non-silent customized signal envelope was created during data pre-processing, the same step is performed on the live-recorded audio data, so that while performing prediction on the signal, the maximum of the non-silent part of the signal is considered.

For running the prediction on the live recorded audio, after all the above pre-processing, the audio data is batched up so that predictions can be made using the argmax function of NumPy on the output of the previously mentioned hot-encoded probabilities. The prediction is then achieved by taking 1 second intervals (specified as delta time previously) within the audio on which the prediction is run i.e., it considers every single second of the audio, and sums all the probabilities and takes the average of them.

Furthermore, it is configured in the prediction module itself that if the class of the recorded audio is identified as one related to a potentially hostile environment (like ''fire crackling'', ''glass breaking'', ''gunshot'', ''scream'' etc.), then an automatic alert message is immediately sent to the registered contact via e-mail, SMS, and WhatsApp. In the case of the e-mail message, it also contains the audio file saved from the live re-cording, which the audio recording module records, saves and prompts the prediction module to identify the class of.

E. THE LIVE AUDIO RECORDING AND PREDICTION MODULE

The audio recording module records live audio from the environment, and stops the recording when no more noise comes. This module also performs a set of other pre-processing steps on the recorded audio, apart from the prediction module, which is invoked later.

As part of pre-processing, the recorded audio is trimmed off of any silence at both ends of the recording, then it is normalized i.e., the volume of the audio signal is aver-aged over the entire length of the recording, and then an audio padding of 0.5 seconds is added to both ends of the recording so that different media players can play the audio without losing any crucial audio data which might be present at the very beginning or the ending of the audio signal.

After the pre-processing is done, the audio data is saved as a ''.wav'' file at a specified location, and the prediction module is automatically prompted to run the prediction to identify the class of the recorded audio, using the deep learning model as specified by the user (among the three models built during this research). After the prediction module is called by the audio recording module, it performs all the steps as mentioned automatically.

V. RESULTS

The problem statement of this research was to identify the type of real-world audio data using deep learning

based multi-class classification technique(s), and determine whether the identified sound type corresponds to a potentially dangerous/threatful environment. Furthermore, the focus had been to use the resultant prediction to develop a threat alert system, which would immediately and automatically issue an alert message to the registered help via e-mail (with an attachment containing the live audio recording of the victim's surrounding), SMS, and WhatsApp. This has been successfully achieved, with the help of the recording module for recording a clear audio signal, which in turn is pre-processed as per the requirements for running the prediction model, and on a detection of a potentially dangerous/threatful environment, the registered help is issued an immediate and automatic threat alert as well.

With respect to the deep learning models developed and used for classifying the audio data in this research, all the three models work approximately equally well. However, the resultant accuracy achieved by each model might significantly vary as the size of the dataset increases as well as the number of audio classes considered. This is one of the prime reasons as to why three different deep learning models have been used in this research.

While the accuracy achieved by the LSTM model is as high as 96.6\%, the 1D-CNN and the 2D-CNN are also not very far behind, with their accuracies at 95.2\% and 96.3\% respectively.

Another metric used for the evaluation of the three models is the Confusion Matrix. The Confusion Matrix is determined with the help of over 9000 records, across the 13 classes. The Confusion Matrix achieved in this research showcases how accurately the models are able to predict the classes of the audio signal files against their actual classes. It verifies the accuracy rate achieved by each of the three models, and the number of correct and incorrect classifications are also visually identifiable. For instance, it can be seen from the Confusion Matrix for the LSTM model depicted in Figure 12 is the model which is able to predict the classes for most of the audio files, while providing a maximum misprediction (31) for the class "gunshot" as a "dog bark" (which are actually quite similar in type, with an audio signal having very short duration, and a high frequency in both cases).

Although the working model built during the course of this research works accurately and as planned during the initial phases of the research, there are still a lot of scope for future research on the same given problem.

With respect to the dataset being used in the project currently, it consists of audio data corresponding to 13 classes only. However, this number can be increased or de-creased as per the requirement which would be needed to be satisfied by the product of this research as a whole. The user's choice of three different deep learning models for training and predicting the class of an audio signal would come to actual use once the size of the dataset increases significantly. Even though the overall computational time required by the models would go

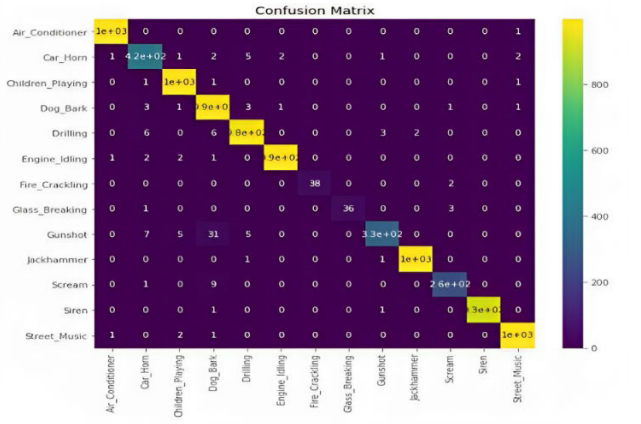

FIGURE 12. The confusion matrix attained in live event prediction.

up by a significant margin, the difference would be visible in the amount of time taken by each model specifically, and the effect of the dataset's size and complexity on its accuracy. The differences in accuracies achieved by the three models on changing the size of the dataset would become a domain of further research by itself. Furthermore, an increase in the number of classes considered for training would further improve the products ability to identify sound corresponding to a threatful environment form a longer list of such classes.

With respect to the aspect of analysis, more feature extraction can be done from the sound data, like amplitude analysis, phase analysis, harmonic distortion analysis etc., which in turn can provide more insights into the audio data, and facilitate the development of more complex and better models, suited to the requirements which this research tries to address.

On one side this research limits its study to the development of a 1D-CNN, 2D-CNN, and an LSTM model for the construction of a system which would detect threats from a person's surroundings, and alert the registered help and emergency services, more complex RNN (Recurrent Neural Networks) models like GRU (Gated Recurrent Unit), Bi-GRU, Bi-LSTM, CTRNN (Continuous Time RNN), HRNN (Hierarchical RNN) etc. can also be researched upon to meet the needs tried to be satisfied by this research. These models would provide even better results as compared to the current output, as these models use even more complex architectures.

Considering the functionalities provided by the product developed during the course of this research, the current model provides an immediate and automatic threat alert message to the registered help if the sound is predicted to be that of a potentially harmful environment. This message is sent via e-mail, SMS, and WhatsApp, and in-forms the registered contact of the type of sound identified from the victim's environment, with the e-mail consisting of the recorded audio file as an attachment as well. This aspect has a lot of scope for future work and further research, and a lot more integrations can be tried to be done to improve and

diversify the end-deliverables of the project. For instance, the location of the victim can also be shared along with the alert message, which can be done with further research on the intricacies of the technical implementation. Likewise, further functionalities can be added as well in order to improve the communication about the victim's distress, the assistance provided to him/her during the time of crisis, and the time taken from the first distress signal for the help to reach the victim.

To summarize, on one hand, the research leverages the power of the Fourier Transform and Mel-Spectrogram analysis to facilitate the development of three different models (1D- CNN, 2D-CNN, and LSTM) to detect threats from a person's surroundings and alert register/emergency services through email, SMS, and WhatsApp, it also has a few limitations. Other complex models, the likes of which (GRU, Bi-GRU, CTRNN, HRNN etc.) have already been mentioned before, can also be used to yield even better results and leaves a scope for further research on the topic.

Although email, SMS, and WhatsApp are great ways to communicate about a person's whereabouts, the research can be furthered to include more information about the victim, like their location, type of distress, intensity of the emergency, physical parameters (like heart rate, oxygen level etc.), environmental parameters (like altitude, temperature etc.), which in turn would allow the emergency services to prepare better for the rescue of the victim from their perilous situation.

A very distinct advantage of the proposed system is that it does not require any special external hardware/wearable devices (like apart watches), but can be implemented with the help of a mere smartphone configured accordingly. In most modern IoT (Internet of Things) systems, the system comes with associated hardware, like smartwatches [33], which constantly monitor a person's health parameters. A person would not have to carry bulky hardware around with them, but only remember to carry their phones with them. However, whenever, more and more modules would be tried to be integrated along with the existing system, the inclusion of a special hardware might become an issue.

The benefits of proposed solution from the fetched results are depicted as below:

As crime rates, and the frequency of natural and man-made disasters have increased significantly in modern days, it is of prime essence to have emergency help and services at the victim's assistance as quickly as possible, with immediate and accurate information about the victim's situation shared with them automatically. The yielded solution helps greatly to avoid any such type of hazard like a robbery, homicide or other threats to any individual as it helps greatly in the accurate prediction of surrounding sound/noise of victims in minimum time and alerts the emergency contact list of victims. Through this accurate prediction, the rate of crimes would go down for the fear of getting caught by the police. The recorded audio along with timestamp and geographical location could also serve as evidence against the person committing crime.

VI. CONCLUSION

A new software-based threat identification system has been developed to the dangerous situation of an individual from his/her ambient noise, and provide immediate assistance to the victim by automatically informing their emergency contacts about the situation. The functionality of recording the live audio on the victim's side, identifying the type of the sound and predicting threatful situation is carried out using the 1D-CNN, 2D-CNN, and LSTM models to achieve an accuracy of 95.2\%, 96.3\%, and 96.6\% respectively, with an average ac-curacy of 96.03\%. In addition to providing the user with three options to choose from for the prediction of the class of the live audio, the choice of more than one model becomes useful as and when the size and complexity of the dataset and the live recorded audio increases. In conclusion, the requirement of the research has been successfully achieved, fulfilling the target of providing an essential solution to one of the greatest practical problems humankind faces in today's world i.e., threat detection and alert system, all while exploring and delivering on the domain of deep learning and audio analysis for the detection of live events from ambient sounds.

REFERENCES

[1] T. P. Suma and G. Rekha, "Study on IoT based women safety devices with screaming detection and video capturing," *Int. J. Eng. Appl. Sci. Technol.*, vol. 6, no. 7, pp. 257–262, 2021.
[2] P. Zinemanas, M. Rocamora, M. Miron, F. Font, and X. Serra, "An interpretable deep learning model for automatic sound classification," *Electronics*, vol. 10, no. 7, p. 850, Apr. 2021.
[3] D. G. Monisha, M. Monisha, G. Pavithra, and R. Subhashini, "Women safety device and application-FEMME," *Indian J. Sci. Technol.*, vol. 9, no. 10, pp. 1–6, Mar. 2016.
[4] G. Ciaburro and G. Iannace, "Improving smart cities safety using sound events detection based on deep neural network algorithms," *Informatics*, vol. 7, no. 3, p. 23, Jul. 2020.
[5] J. Cao, M. Cao, J. Wang, C. Yin, D. Wang, and P.-P. Vidal, "Urban noise recognition with convolutional neural network," *Multimedia Tools Appl.*, vol. 78, no. 20, pp. 29021–29041, Oct. 2019.
[6] A. Triantafyllopoulos, G. Keren, J. Wagner, I. Steiner, and B. W. Schuller, "Towards robust speech emotion recognition using deep residual networks for speech enhancement," in *Proc. INTERSPEECH*, Graz, Austria, Sep. 2019.
[7] M. T. García-Ordás, H. Alaiz-Moretón, J. A. Benítez-Andrades, I. García-Rodríguez, O. García-Olalla, and C. Benavides, "Sentiment analysis in non-fixed length audios using a fully convolutional neural network," *Biomed. Signal Process. Control*, vol. 69, Aug. 2021, Art. no. 102946.
[8] A. Bhardwaj, P. Khanna, S. Kumar, and Pragya, "Generative model for NLP applications based on component extraction," *Proc. Comput. Sci.*, vol. 167, pp. 918–931, Jan. 2020.
[9] I. S. Malova and D. V. Tikhomirova, "Recognition of emotions in verbal messages based on neural networks," *Proc. Comput. Sci.*, vol. 190, pp. 560–563, Jan. 2021.
[10] A. Hodorog, I. Petri, and Y. Rezgui, "Machine learning and natural language processing of social media data for event detection in smart cities," *Sustain. Cities Soc.*, vol. 85, Oct. 2022, Art. no. 104026.
[11] D. D. Nguyen, M. S. Dao, and T. V. T. Nguyen, "Natural language processing for social event classification," in *Knowledge and Systems Engineering*. Cham, Switzerland: Springer, 2015, pp. 79–91.

[12] M. A. Sit, C. Koylu, and I. Demir, "Identifying disaster-related tweets and their semantic, spatial and temporal context using deep learning, natural language processing and spatial analysis: A case study of hurricane irma," *Int. J. Digit. Earth*, vol. 12, no. 11, pp. 1205–1229, Nov. 2019.

[13] D. Tsiktsiris, A. Vafeiadis, A. Lalas, M. Dasygenis, K. Votis, and D. Tzovaras, "A novel image and audio-based artificial intelligence service for security applications in autonomous vehicles," *Transp. Res. Proc.*, vol. 62, pp. 294–301, Jan. 2022.

[14] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, "PANNs: Large-scale pretrained audio neural networks for audio pattern recognition," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 28, pp. 2880–2894, 2020.

[15] L. Yang and H. Zhao, "Sound classification based on multihead attention and support vector machine," *Math. Problems Eng.*, vol. 2021, pp. 1–11, May 2021.

[16] *Urban Sound Datasets*. Accessed: Dec. 1, 2023. [Online]. Available: https://urbansounddataset.weebly.com/download-urbansound8k.html

[17] *Environmental Sound Classification 50*. Accessed: Dec. 1, 2023. [Online]. Available: https://www.kaggle.com/datasets/mmoreaux/environmental-sound-classification-50?select=audio

[18] *Audio Dataset of Scream and Non Scream*. Accessed: Dec. 1, 2023. [Online]. Available: https://www.kaggle.com/datasets/aananehsansiam/audio-dataset-of-scream-and-non-scream

[19] *Writing Your Own Callbacks*. Accessed: Dec. 1, 2023. [Online]. Available: https://www.tensorflow.org/guide/keras/custom callback

[20] *The Architecture of LeNet-5*. Accessed: Dec. 1, 2023. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/03/the-architecture-of-lenet-5/

[21] *Understanding the Mel Spectrogram*. Accessed: Dec. 1, 2023. [Online]. Available: https://medium.com/analytics-vidhya/understanding-the-mel-spectrogram-fca2afa2ce53

[22] *Kapre*. Accessed: Dec. 1, 2023. [Online]. Available: https://kapre.readthedocs.io/

[23] *Google Search by Voice: A Case Study*. Accessed: Dec. 1, 2023. [Online]. Available: https://research.google.com/pubs/archive/36340.pdf

[24] M. Assefi, G. Liu, M. P. Wittie, and C. Izurieta, "An experimental evaluation of apple Siri and Google speech recognition," in *Proc. ISCA SEDE*, 2015, p. 118.

[25] A. L. Nobles, E. C. Leas, T. L. Caputi, S.-H. Zhu, S. A. Strathdee, and J. W. Ayers, "Responses to addiction help-seeking from alexa, siri, Google assistant, cortana, and bixby intelligent virtual assistants," *npj Digit. Med.*, vol. 3, no. 1, p. 11, Jan. 2020.

[26] I. Lopatovska, K. Rink, I. Knight, K. Raines, K. Cosenza, H. Williams, P. Sorsche, D. Hirsch, Q. Li, and A. Martinez, "Talk to me: Exploring user interactions with the Amazon Alexa," *J. Librarianship Inf. Sci.*, vol. 51, no. 4, pp. 984–997, Dec. 2019.

[27] A. Farzindar, D. Inkpen, and G. Hirst, *Natural Language Processing for Social Media*. San Rafael, CA, USA: Morgan Claypool, 2015.

[28] S. Zad, M. Heidari, J. H. J. Jones, and O. Uzuner, "Emotion detection of textual data: An interdisciplinary survey," in *Proc. IEEE World AI IoT Congr. (AIIoT)*, May 2021, pp. 0255–0261.

[29] W. Graterol, J. Diaz-Amado, Y. Cardinale, I. Dongo, E. Lopes-Silva, and C. Santos-Libarino, "Emotion detection for social robots based on NLP transformers and an emotion ontology," *Sensors*, vol. 21, no. 4, p. 1322, Feb. 2021.

[30] Z. T. Sworna, Z. Mousavi, and M. A. Babar, "NLP methods in host-based intrusion detection systems: A systematic review and future directions," *J. Netw. Comput. Appl.*, vol. 220, Nov. 2023, Art. no. 103761.

[31] T. Özseven, "Investigation of the effectiveness of time-frequency domain images and acoustic features in urban sound classification," *Appl. Acoust.*, vol. 211, Aug. 2023, Art. no. 109564.

[32] R. Viveros-Muñoz, P. Huijse, V. Vargas, D. Espejo, V. Poblete, J. P. Arenas, M. Vernier, D. Vergara, and E. Suárez, "Dataset for polyphonic sound event detection tasks in urban soundscapes: The synthetic polyphonic ambient sound source (SPASS) dataset," *Data Brief*, vol. 50, Oct. 2023, Art. no. 109552.

[33] A. B. Shrestha, B. Khanal, N. Mainali, S. Shrestha, S. Chapagain, T. P. Umar, and V. Jaiswal, "Navigating the role of smartwatches in cardiac fitness monitoring: Insights from physicians and the evolving landscape," *Current Problems Cardiology*, vol. 49, no. 1, Jan. 2024, Art. no. 102073.

[34] P. Upretee and M. E. Yüksel, "Accurate classification of heart sounds for disease diagnosis by using spectral analysis and deep learning methods," in *Data Analytics in Biomedical Engineering and Healthcare*, New York, NY, USA: Academic, 2021, pp. 215–232.

[35] K. Presannakumar and A. Mohamed, "Deep learning based source identification of environmental audio signals using optimized convolutional neural networks," *Appl. Soft Comput.*, vol. 143, Aug. 2023, Art. no. 110423.

[36] A. M. Tripathi and A. Mishra, "Self-supervised learning for environmental sound classification," *Appl. Acoust.*, vol. 182, Nov. 2021, Art. no. 108183.

[37] M. Mohaimenuzzaman, C. Bergmeir, I. West, and B. Meyer, "Environmental sound classification on the edge: A pipeline for deep acoustic networks on extremely resource-constrained devices," *Pattern Recognit.*, vol. 133, Jan. 2023, Art. no. 109025.

[38] S. Dong, Z. Xia, X. Pan, and T. Yu, "Environmental sound classification based on improved compact bilinear attention network," *Digit. Signal Process.*, vol. 141, Sep. 2023, Art. no. 104170.

AMRIT SEN received the B.Tech. degree in computer science and engineering with a specialization in cyber-physical systems from the Vellore Institute of Technology, Chennai. His research interests include data science, machine learning, and deep learning. During the course, he worked on several in-house projects. He also worked on a deep learning project, named "A Brain Tumour Segmentation and Classification System using Mini Batch K-Means Clustering and CNN" and carried out Industrial Research Internship with Samsung and worked on a team project titled "Graph Embedding Generation for Link Prediction and User Classification."

GAYATHRI RAJAKUMARAN received the bachelor's degree from the Rajiv Gandhi College of Engineering and Technology, in 2009, the master's degree from the Pondicherry Engineering College, in 2011, and the Ph.D. degree from the Vellore Institute of Technology (VIT), Chennai, in 2020, under cloud security specialization. She is currently affiliated with VIT Chennai as an Assistant Professor (Senior) with the Department of Computer Science and Engineering. Her research interests include cloud security, information and cyber security, the IoT, and machine learning. She has published numerous journals in high indexed journals and holding two patents related to domains agriculture and the IoT. She is a Reviewer of *The Journal of Super Computing*. She played the role of an Editor and the author for publishing the books *Grid and Cloud Computing*, *Cloud Computing*, and *Cloud Security*. She was the Sponsorship Chair of International Conference on Big Data and Cloud Computing (ICBCC) 2018 and attracted fund from a government funding agency. She is the current Linux Club Co-Coordinator with VIT Chennai.

MIROSLAV MAHDAL is currently the Vice-Dean for Science, Research and Doctoral Studies with the Faculty of Mechanical Engineering, VSB-Technical University of Ostrava, and an Associate Professor with the Department of Control Systems and Instrumentation. His research interests include the control of mechatronic systems, control systems, automatic control theory, wireless technologies, artificial intelligence, cloud computing, optimization methods, and the programming of control systems. He has nearly more than 80 articles to his credit.

SHOLA USHARANI received the Ph.D. degree from the Vellore Institute of Technology (VIT), Chennai, in 2020. She is currently an Associate Professor with the School of Computing Science and Engineering, VIT Chennai. Her research interests include embedded systems, the IoT, machine learning, computer networks, cloud computing, and security. She has published articles in Scopus indexed journals, guided more number of UG and PG projects in the area of embedded systems, the IoT, machine learning, and security. She is an active ACM Member. She received Research Award, from 2014 to 2015. She is a Reviewer in journals, such as *IGI Global* journal systems and Book Systems. She is acting as an Android Club Coordinator with VIT University.

VEZHAVENDHAN RAJASEKHARAN received the Bachelor of Engineering degree in mechanical engineering from the University of Madras, the Master of Technology degree in manufacturing engineering and management, and the Ph.D. degree in enterprise transformation. He began his career as a hardcore manufacturing Engineer and ventured into entrepreneurship to manufacture mill boards from waste paper. Later, he joined the Vellore Institute of Technology, Vellore Campus, where he is currently a Senior Associate Professor. He is also a Distinguished Mechanical Engineer and a Researcher with a good background in the application of artificial intelligence (AI) and machine learning (ML) in various engineering domains. His current research interests include quality management, entrepreneurship, AI, and ML. He has collaborated actively with researchers in several disciplines of AI, ML, and the IoT. His expertise, coupled with his passion for advancing the boundaries of AI and ML, continues to inspire the next generation of researchers and engineers in their pursuit of innovation and excellence.

RAJIV VINCENT received the master's degree in computer science and engineering from the College of Engineering Guindy, Anna University, Chennai, India. He is currently an Assistant Professor Senior with the School of Computing Science and Engineering, Vellore Institute of Technology (VIT), Chennai. He has been an Academician for the past 11 years and a System Administrator for two years. He has published a book in machine learning titled as *Image Processing for Machine Learning* (ISBN: 978-93-5445-509-4). He has published many research articles in reputed Scopus indexed and Web of Science journals, also two Indian patents and got one grant in international patent. His academic and research expertise covers a wide range of subject area, including deep learning, image processing, and web technologies.

KARTHIKEYAN SUGAVANAN is currently pursuing the degree with the Vellore Institute of Technology (VIT), Chennai. With a keen interest in data science and its intersection with AI and IoT, his academic journey has been marked by outstanding achievements and a thirst for knowledge in cutting-edge technologies. His dedication to this field and desire to make a meaningful impact on people's lives led him to collaborate with esteemed mentors, Dr. Shola Usharani and Dr. Gayathri Rajakumaran. Inspired by the potential of NLP and deep learning, he has been actively involved in their applications for live event detection for people's safety. This innovative project aims to harness the power of natural language processing and advanced machine learning algorithms to ensure the security and well-being of individuals during live events.