I.J. Intelligent Systems and Applications, **2024, 6, 40-72**
Published Online on December 8, 2024 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijisa.2024.06.03

Information Technology for Sound Analysis and Recognition in the Metropolis based on Machine Learning Methods

Lyubomyr Chyrun
Applied Mathematics Department, Faculty of Applied Mathematics and Informatics, Ivan Franko National University of Lviv, Lviv, 79000, Ukraine
E-mail: Lyubomyr.Chyrun@lnu.edu.ua
ORCID iD: https://orcid.org/0000-0002-9448-1751

Victoria Vysotska
Department of Information Systems and Networks, Institute of Computer Sciences and Information Technologies, Lviv Polytechnic National University, Lviv, 79013, Ukraine
Osnabrück University, Osnabrück, 49076, Germany
E-mail: Victoria.A.Vysotska@lpnu.ua, victoria.vysotska@uni-osnabrueck.de
ORCID iD: https://orcid.org/0000-0001-6417-3689

Stepan Tchynetskyi
Department of Information Systems and Networks, Institute of Computer Sciences and Information Technologies, Lviv Polytechnic National University, Lviv, 79013, Ukraine
E-mail: stepan.tchynetskyi.msaad.2022@lpnu.ua
ORCID iD: https://orcid.org/0009-0004-5201-8429

Yuriy Ushenko*
Department of Computer Science, Educational and Research Institute of Physical, Technical and Computer Sciences, Yuriy Fedkovych Chernivtsi National University, 58012, Ukraine
E-mail: y.ushenko@chnu.edu.ua
ORCID iD: https://orcid.org/0000-0003-1767-1882
*Corresponding author

Dmytro Uhryn
Department of Computer Science, Educational and Research Institute of Physical, Technical and Computer Sciences, Yuriy Fedkovych Chernivtsi National University, 58012, Ukraine
E-mail: d.ugryn@chnu.edu.ua
ORCID iD: https://orcid.org/0000-0003-4858-4511

Received: 22 October 2023; Revised: 15 March 2024; Accepted: 11 May 2024; Published: 08 December 2024

Abstract: The goal of designing and implementing an intelligent information system for the recognition and classification of sound signals is to create an effective solution at the software level, which would allow analysis, recognition, classification and forecasting of sound signals in megacities and smart cities using machine learning methods. This system can help people in various fields to simplify their lives, for example, it can help farmers protect their crops from animals, in the military it can help with the identification of weapons and the search for flying objects, such as drones or missiles, in the future there is a possibility for recognizing the distance to sound, also, in cities can help with security, so a preventive response system can be built, which can check if everything is in order based on sounds. Also, it can make life easier for people with impaired hearing to detect danger in everyday life. In the part of the comparison of analogues of the developed product, 4 analogues were found: Shazam, sound recognition from Apple, Vocapia, and SoundHound. A table of comparisons was made for these analogues and the product under development. Also, after comparing analogues, a table for evaluating the effects of the development was built. During the system analysis section, a variety of audio research materials were developed to indicate the characteristics that can be used for this design: period, amplitude, and frequency, and, as an example, an article on real-world audio applications is shown. A precedent scenario is described using the RUP methodology and UML diagrams are constructed: Diagram of use

cases; Class diagram; Activity chart; Sequence diagram; Diagram of components; and Deployment diagram. Also, sound data analysis was performed, sound data was visualized as spectrograms and sound waves, which clearly show that the data are different, so it is possible to classify them using machine learning methods. An experimental selection of the machine learning method as staandart clasificers for building a sound recognition model was made. The best method turned out to be SVC, the accuracy of which reflects more than 30 per cent. A neural network was also implemented to improve the obtained results. The result of training a model based on a neural network during 100 epochs achieved a result of 97.7% accuracy for training data and 47.8% accuracy when checking performance on test data. This result should be higher, so it is necessary to consider improving recognition algorithms, increasing the amount of data, and changing the recognition method. Testing of the project was carried out, showing its operation and pointing out shortcomings that need to be corrected in the future.

Index Terms: Data Augmentation, Intelligent System, Application, Sound Waves, Sound Spectrum, SkLearn, Feature Extraction, Sound Analysis, Machine Learning Methods.

1. Introduction

The powerful leap of information technologies in recent years stimulates the development of new areas of their application, designed to automate routine operations and increase the convenience and efficiency of production processes and service provision. Today, it is possible to observe the large-scale implementation of methods and means of artificial intelligence, as well as IoT technologies in the everyday lives of people. Now you won't surprise anyone with voice control of household appliances, cars and other convenient gadgets. All this unites a new technology called "smart home", or even more broadly, "smart technologies". Important achievements have also been achieved in the areas of pattern recognition, processing of natural language and several others, which allows to improve and increase the quality of life of the population. However, the quality and timeliness of the provision of services by emergency services in the event of emergencies, in particular, by ambulances, firefighters, police, etc., is a determining factor in human life.

The sounds reproduce the noise, bustle, and bustle of the metropolis, which in turn affects the psycho-emotional state and health of the residents. At the same time, the analysis, recognition and classification of sounds will improve the quality of service in megapolises and smart cities. In particular, taking into account the frantic pace of development of road traffic associated with the increase of motor vehicles and the slow steps towards the expansion of infrastructure in large cities, there are problems with car traffic jams. The presence of traffic jams in cities does not allow emergency services to work effectively, since the accumulation of cars at traffic lights and imperfect infrastructure do not allow them to quickly arrive at the scene of an emergency. Therefore, an urgent task is the development of a computer system for the automatic recognition and classification of sound signals generated by sirens of special-purpose vehicles for the formation of green corridors when passing through intersections with traffic light regulations.

The computer system for the recognition and classification of sound signals is designed to automate the process of determining and establishing the identity of the signals generated by the sound source, based on which certain decisions are made. The designed system should provide the ability to collect sound signals and convert them into a digital format. In addition, received signals with appropriate characteristics based on models and machine learning algorithms should be automatically classified into appropriate classes with appropriate labels.

Computer systems for the recognition and classification of sound signals have a wide range of applications, ranging from autonomous independent systems to subsystems of more complex IT solutions. Autonomously, such systems can be used in the formation of music albums by genres, artists or other categories. As subsystems of more complex solutions, a computer system for recognizing and classifying sound signals can be used in the organization of security and voice control of a "smart home". In addition, taking into account the number of cars and their constant growth with unchanged infrastructure, the creation of systems for analysing the sound signals of cars is urgent. This will allow emergency services to get to the places of dangerous situations more quickly and efficiently, by forming green corridors. The computer system must allow for the accumulation of data on car horns and the automatic identification of horn labels. At the hardware level, the system should be responsive to events, portable and able to coexist with other compatible systems.

The work will consider the main aspects of the sound environment in the metropolis, the methods and technologies of sound recognition, and their application in various areas, from measuring the noise level to detecting emergencies and monitoring sound pollution. An analysis of the possible effects of the sound environment on human health and life will also be conducted, as ways of optimization to create a more comfortable and safer urban environment will be considered. The work is aimed at highlighting the importance of the problem of sound recognition in the metropolis, as at the search for innovative solutions and the new approaches development to solving this urgent problem.

The goal of designing and implementing an intelligent information system for the recognition and classification of sound signals is to create an effective solution at the software level, which would allow analysis, recognition, classification and forecasting of sound signals in megacities and smart cities using machine learning methods. The task

of the work is to develop software for recognizing sounds using machine learning methods, which could identify sounds in real-time and from recordings. List of main tasks for solving the problem:

- Analyse the sound recognition market;
- Analyse possible solutions to this problem and choose the optimal one;
- Conduct a system analysis for the future recognition system;
- Develop MVP of this system.

The object of research is the process of sound recognition. The subject of the research is the system of sound analysis and recognition in the metropolis. A scientific novelty is the analysis of the possibilities of machine learning methods in working with sound with the help of various activation functions and at different. The practical value of this work is the developed system product that will be able to recognize sound using machine learning methods.

This application can be used in various areas:

- Simple user – can use to control buildings, in case of absence;
- As a radio nanny;
- As an aid to people with hearing impairments;
- To help automation, for example, farmers to scare birds;
- To determine the type of drone (it is possible to determine the location if the sound of the drone is set);
- To determine the type of ammunition used.

This topic is relevant because it can help ensure order, help people with various activities and capabilities, and help in military affairs. Information technology use helps detect animals in agriculture and automate animal deterrent measures, as 35% of annual crops are lost by farmers due to the birds' impact [1]. In military applications, this system can detect the flight of missiles at low altitudes or drones, such as the Shahed 136. The Shahed-136 is a modern combat drone developed by the Iranian industry, designed to neutralize ground targets at a distance. This aircraft bypasses anti-aircraft defences and attacks ground targets, launching from a launch pad. His work is characterized by a distinctive sound similar to a moped. This drone was discovered thanks to published footage in December 2021 [2]. It can also help in the detection and classification of ground targets, such as air defence systems or artillery installations.

In the current digital era, the importance of sound recognition systems has increased in many fields, including speech recognition, music classification, acoustic monitoring, etc. These systems attempt to automatically decode, interpret, and extract relevant data from audio signals. Neural networks have become effective tools for audio recognition tasks thanks to recent developments in artificial intelligence and machine learning.

Traditional audio recognition methods often rely on hand-crafted features and rule-based algorithms, which can be time-consuming and have limitations when processing complex audio data. Neural networks, in turn, have the ability to automatically detect and extract relevant features from raw audio signals, increasing the accuracy and reliability of sound recognition tasks.

The demand for efficient and accurate sound recognition systems in many fields has motivated the development of this research. For example, in speech recognition, accurate transcription and understanding of spoken language is essential for voice assistants, transcription services, and speech processing applications. Similar to visual surveillance, acoustic surveillance can enhance security systems and monitoring procedures by being able to automatically detect and classify certain sounds. Therefore, in our work, we strive to overcome the shortcomings of conventional approaches and develop the field of sound recognition, using the capabilities of machine learning technology.

The focus of this research is on audio recognition tasks such as speech recognition, environmental sound classification, and audio event detection. The research will build and deploy machine learning models, collect or use relevant datasets, pre-process audio data, train and evaluate the models, and analyze the results.

The importance of voice recognition systems lies in their potential to improve automation, user interaction, security, enable intelligent decision-making, and create new applications in various fields.

Thus, this research is relevant and aimed at creating systems that will simplify processes such as voice commands, speaker identification, emotion detection, music genre classification, environmental monitoring, and abnormal sound detection through accurate sound identification and classification.

2. Literature Review

2.1. Analytical Review of Developments and Research in the Field of Sound Recognition

The life that surrounds us is permeated with various sounds, which can range from pleasant melodies to loud noises. These audio elements, being an inseparable part of our existence, have a significant impact on our emotional state and shape our perception of the world around us [3-4]. Even when we unconsciously react to sounds, our brain constantly analyses them, creating a comprehensive picture of our environment, which can provide us with important information about the environment [5-7]. There are deep connections between the acoustic environment and our emotional state [9-10]. For example, pleasant music can lift the mood and create an atmosphere of joy, while unpleasant

noises can cause stress and irritation. Sound impressions can also affect our decisions and concentration. Thus, the analysis of sounds not only expands our understanding of the world around us but also opens up new opportunities for enriching our emotional experience and improving the quality of our lives. Thanks to modern technologies that allow deeper analysis of sound signals, we can get much more information and understand their impact on us [11-12]. Sound classification in audio deep learning is not only a powerful tool for distinguishing sound signals but also a key element in the development of modern technologies aimed at understanding the sound environment. This approach involves studying whole aspects of sounds to provide accurate classification and prediction of their category. One of the main advantages of sound classification in deep learning is its versatility. This method can be successfully applied to various scenarios, which extends its practicality. For example, in the classification of music videos, it can be used to automatically determine the genre of music, which facilitates the search and selection of music content. Also, the classification of short utterances based on a set of speakers allows one to identify a specific speaker by voice, which can be used in recognition and personal identification systems. This approach is becoming an important element in the development of modern audio technologies, which opens wide prospects for automating and improving various aspects of our interaction with sound signals. Accordingly, further improvement of sound classification methods in deep learning will open up new opportunities for expanding our abilities to understand and use the sound environment in various aspects of life [13-14].

Audio analysis is a complex and dynamic process that involves the transformation, examination and interpretation of audio signals captured by digital devices. This process is used to uncover the depths of sound data and identify important characteristics and regularities in them [15-21]. The most popular types of audio analysis are environmental sound recognition; Music recognition; Voice recognition; and Language recognition. Audio data is analogue sounds converted to digital format while preserving the key characteristics of the original. This technological achievement opens up opportunities for more convenient and efficient analysis and processing of sound signals. According to the basics of physics, sound is a wave of vibrations that propagates through the medium and reaches our ears. The basic characteristics of sound, such as period, amplitude and frequency, determine its basic properties and perception. The period indicates the interval between oscillations, and the amplitude determines the loudness. The frequency indicates the number of oscillations per unit of time and affects the pitch of the sound. These parameters not only help in recognizing the nature of the sound but also allow its reproduction in digital format. The importance of these characteristics becomes apparent when analysing audio data, as they determine many aspects of audio signals. A high frequency can indicate light tones or noise, a significant amplitude - a loud sound and a change in the period can indicate a variety of sound events [20].

To test the effectiveness of sound recognition in the real world, you can refer to the work of Avijeet Kumar and Roop Pahuja, who used sound recognition to identify bird species in their research [21]. They developed a special tool with an efficient graphical user interface that processes audio recordings and generates a statistically evaluated characteristic matrix based on a spectrogram obtained by Fourier transformation. This matrix is used to determine the vocalization patterns of different bird species. This technology has wide applications in ornithology for studying migratory routes of birds and can be useful in agriculture.

2.2. Features of Sound Analysis

In our daily life, sound is a crucial component. It allows us to communicate, enjoy music and perceive our environment. But what exactly is sound? Sound is a type of energy that travels through a medium in the form of waves, such as air, water, or solids. The particles of the medium contract and expand as these waves travel through it. Being mechanical waves, sound waves require a physical medium to travel through. As a result, sound cannot be heard in space because it cannot travel in a vacuum. Sound waves occur when an object vibrates because it causes a disturbance in the environment in which it is located. Frequency, amplitude, and phase are just some of the characteristics that define sound waves. Frequency, which is measured in hertz (Hz), is the number of complete oscillations or cycles per second. The range of frequencies that the human ear can hear is from 20 to 20,000 Hz. Conversely, amplitude, which is expressed as the loudness or intensity of a sound, is measured in decibels (dB). The position of a sound wave in a wave cycle is determined by its phase.

Sound waves are analog in nature and must be converted to a digital format in order to be stored, transmitted, or processed. Sampling and quantization are the two most important steps of the transformation. Sampling involves taking repeated measurements or snapshots of a sound wave at predetermined time intervals. Each instantaneous amplitude of the sound wave is recorded in samples. The rate at which these samples are taken is called the sampling rate, and is usually expressed in oscillations per second or hertz (Hz). 44.01 kHz (CD), 48 kHz (common for video and audio), and 96 kHz (for high-resolution audio) are typical sampling rates used in digital audio systems. Assigning numerical values to discretized amplitudes is known as quantization. In other words, it involves converting a continuous range of analog amplitudes into discrete values that can be represented digitally. For this, the amplitude range is divided into an arbitrary number of levels, each of which is assigned a numerical value. The number of levels determines the bit depth or resolution of the digital audio representation. 16-bit and 24-bit are the two most commonly used bit depths in digital audio. A 24-bit signal can represent 16,777,216 discrete levels compared to the 65,536 discrete levels of a 16-bit signal. More levels mean that the digital representation can more accurately capture the subtleties of the original analog sound wave.

Analog-to-digital conversion, or ADC for short, is the process of converting an analog sound wave into a digital representation. The steps that make up this conversion process:

- Sampling: A continuous analog sound wave is sampled regularly. How many samples are taken every second depends on the sample rate. For example, a sample rate of 44.1 kHz means that 44,100 samples are taken every second.
- Quantization: After sampling, the values are quantized, giving each sample a numerical value. The analog value is rounded to the nearest discrete level during the quantization process depending on the selected bit depth. Quantization error adds some distortion or noise to the digital representation.
- Encoding: Binary representation is used to encode quantized values. The binary code called pulse-code modulation (PCM) is most often used. Each sample is represented by a binary number, with the number of bits used varying depending on the bit depth selected.
- An analog sound wave can be accurately modeled in the digital domain by following these steps. Each discrete sample in the resulting digital audio file has a corresponding numerical value that represents the amplitude of the output sound wave at that precise moment in time.

Digital audio can be stored in a variety of formats, each with its own characteristics and compression algorithms. Among the common audio formats:

- WAV (Waveform Audio File Format): WAV is an uncompressed audio format that preserves the full fidelity of the original sound wave. It is widely supported, but can take up a lot of storage space.
- MP3 (MPEG Audio Layer-3): MP3 is a popular audio format that uses lossy compression to reduce file size. It achieves compression by removing irrelevant audio data. The trade-off is a slight loss of sound quality.
- AAC (Advanced Audio Coding): AAC is another popular audio format that offers better compression efficiency compared to MP3. It provides improved audio quality at lower bitrates, making it suitable for streaming and portable devices.
- FLAC (Free Lossless Audio Codec): FLAC is a lossless audio format that compresses audio data without compromising quality. It offers smaller file sizes compared to WAV while maintaining the original audio fidelity.

These are just a few examples of digital audio formats, and each has its own strengths and uses. The choice of format depends on factors such as memory capacity, desired audio quality, and intended use.

Once the audio is converted to a digital format, it becomes suitable for various digital signal processing (DSP) techniques. Digital audio processing enables a wide range of applications, including audio editing, effects processing, equalization and noise reduction. Here are some key concepts and techniques used in digital signal processing for audio:

- Fast Fourier Transform (FFT): FFT is a mathematical algorithm that transforms a time-domain signal, such as an audio signal, into its frequency-domain representation. It decomposes the signal into its frequency components, which allows analysis and manipulation in the frequency domain. FFT is widely used in sound spectrum analysis, filtering and effects processing.
- Filtering: Filtering techniques are used to change the frequency content of an audio signal. Low-pass filters pass frequencies below a certain cutoff limit while attenuating higher frequencies. High-pass filters do the opposite, passing higher frequencies and attenuating lower frequencies. Bandpass filters allow you to pass a certain range of frequencies while rejecting others. Filtering is often used to remove unwanted noise, shape the sound spectrum, or create certain effects.
- Echo and Reverb: Echo and reverb effects are commonly used in audio production and sound design. Echo creates repetitions of the original sound with reduced intensity, simulating reflections from distant surfaces. Reverberation simulates the complex acoustic environment of a room or space. Both effects are achieved using delay lines and feedback loops, where the delayed sound is combined with the original signal to create the desired effect.
- Equalization: Equalization is the process of adjusting the relative amplitudes of the various frequency components of an audio signal. This allows you to tonally shape and correct by emphasizing or reducing certain frequency ranges. Graphic EQs provide adjustable gain bands to control different frequency ranges, while parametric EQs offer more precise control with adjustable center frequencies, bandwidth and gain.

For storage and transmission purposes, audio compression is critical to reducing the size of digital audio files. While maintaining a decent level of sound quality, compression algorithms reduce the file size. Lossless compression and lossy compression are the two main types of compression techniques. Lossless compression: Lossless compression algorithms minimize file size without degrading audio quality. By taking advantage of statistical redundancy in audio data, they can compress the data. Lossless audio compression formats include Apple Lossless (ALAC) and FLAC. If it is important to maintain audio quality, for example during professional audio creation or archiving, lossless compression is suitable. Lossy Compression: Lossy compression algorithms reduce audio data that is considered less

important to perception by permanently removing it. Audio quality is slightly degraded as a result of the final deletion of data. Lossy audio compression formats such as MP3 and AAC are common in many applications. When it comes to music streaming, mobile devices, and web distribution, lossy compression is often used because smaller file sizes are critical.

By simulating sound localization and spatial cues, spatial audio technologies aim to create immersive and realistic sound. These developments give sound reproduction a new perspective, improving the perception of depth, directionality and movement. Applications such as virtual reality (VR), augmented reality (AR) and gaming benefit greatly from spatial audio. Multichannel audio systems, binaural recording and playback, ambisonics, and object-oriented audio are all used in spatial audio techniques. Multi-channel audio systems, such as 5.1 or 7.1 surround sound configurations, use multiple speakers to play sound from different angles. Using specialized microphone technology and headphones, binaural recording and playback aims to mimic the way human ears naturally perceive sound. Ambisonics is a technique that captures and reproduces the full spectrum of sound using a spherical array of microphones and speakers. Object-based audio enables dynamic audio reproduction where audio objects can be positioned and moved in 3D space. Entertainment, communications and other industries are being revolutionized by the development of spatial audio technologies that offer more realistic and immersive sound.

Sounds and audio signals have numerous applications in various fields. Here are a few key areas where sound plays a crucial role:

- Entertainment: Sound is an integral part of entertainment media, including music, movies, television and games. This heightens the emotional impact, creates an immersive experience, and adds depth to the narrative. Audio engineers and sound designers work tirelessly to create and manipulate soundscapes, musical compositions and special effects that capture and manipulate audiences.
- Communication: Sound plays a vital role in human communication. From everyday conversations to public address systems, sound waves carry spoken words and transmit emotions. Telecommunications systems, including telephones, voice over IP (VoIP), and video conferencing, rely on audio signals to facilitate remote communication.
- Broadcasting: Radio and television broadcasting rely heavily on audio signals to transmit news, music and other content. Sound engineers and broadcasters work together to capture and deliver high-quality audio that engages and informs audiences.
- Speech recognition and synthesis: Audio signals are used in speech recognition systems that convert spoken words into text. These systems have applications in voice assistants, transcription services, and accessibility tools. In contrast, speech synthesis technologies convert text into spoken words, enabling applications such as text-to-speech systems and voice assistants.
- Medical Applications: Sound waves are used in various medical imaging techniques such as ultrasound and sonography. Ultrasound imaging uses high-frequency sound waves to create real-time images of internal body structures, aiding in diagnosis and monitoring. In addition, auditory cues and sound therapy are used in audiology and rehabilitation facilities.
- • Automotive Systems: Audio and sound play an important role in automotive systems. Car audio systems provide entertainment and enhance the driving experience. In addition, warning signals such as horn and engine sounds help with communication and road safety.

These are just a few examples of the various applications of beeps and beeps. Sound technology continues to evolve, opening up new possibilities in areas such as virtual reality, artificial intelligence, machine/deep learning methods, and human-computer interaction.

2.3. Comparison of Analogues of the Product under Development

In this subsection, a comparison of analogues with the product under development is made, which will help identify the shortcomings and advantages of the development. As the analysis result, analogues of sound recognition programs with completely different fields of application were found (Fig. 1): Shazam; Sound recognition from Apple; Vocapia; and SoundHound.

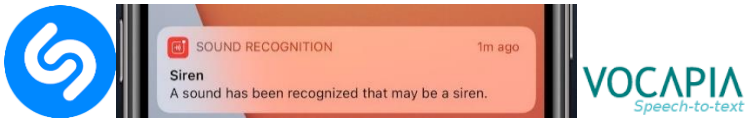


Fig.1. The shazam logo, using apple voice recognition and the vocapia logo

Shazam is a mobile app and service dedicated to music recognition. Users can use an app to capture a short audio clip from the surrounding sound, and Shazam identifies a famous piece of music, providing artist, song and album information. The service also allows users to purchase music or go to platforms to listen to a full track. Shazam also supports Mac OS and Windows operating systems [22]. This product impressed with intelligence and wide popularity. To use it, simply hold your Shazam-enabled phone up to an unknown music piece and it will automatically send you a

text message with the song title and the artist, as a link to purchase the track. The software effectively filters out unrelated background noise, allowing users to identify music tracks used in movies or TV shows. In addition, it can recognize music played in animated pubs or clubs [23]. Currently, in the 2020s, this service has information on more than 11,000,000 music tracks in its repertoire [22]. Shazam's annual revenue is $92.0 million. Shazam is one of the most popular and widely used music recognition apps available today. Shazam has a huge user base thanks to its user-friendly interface and powerful recognition capabilities. Shazam can quickly identify a track and provide users with details such as song title, artist, album and even lyrics, simply by playing a short audio clip of the song. Users can discover new music, create playlists and enjoy seamless listening by integrating Shazam's extensive music database with popular streaming services.

Google Voice Search gives users an easy way to recognize songs playing around them and is integrated into a suite of Google services, including Google Assistant and Google Now. Google Voice Search uses audio fingerprint technology to identify a song and provide relevant information when the app is activated and allowed to listen to the audio. Users can learn more about a specific song, listen to it on different platforms and find related music based on their preferences.

Sound recognition from Apple. Apple's new smartphones are equipped with sound recognition features that can detect a variety of common audio signals, such as smoke alarms, animal voices including cats and dogs, fire alarms, doorbells, running water, and babies crying. This feature is based on the fact that even if you can't hear a certain sound, your iPhone or iPad can recognize it. If a sound is detected (like a cat barking), your device vibrates, plays the sound, and sends you a message. During the tests, it can be noted that the sound recognition function works quite effectively. It should be taken into account that the device must be close enough to detect the sound. In addition, when the voice recognition function is activated, it is possible to call Siri. However, it's important to note that this approach isn't perfect, as the feature may not recognize some of the sounds it was originally configured to recognize. It should be remembered that sound recognition should be considered as an auxiliary tool in cases where you cannot listen to sounds personally. It is cautioned against relying on this feature in emergency or risky situations [24]. When the device detects a certain sound, such as the doorbell, you receive a notification on the screen. In addition, it is possible to delay the recognition of this sound for 5, 20 minutes or 2 hours. If the device is in sleep mode, there is also the possibility to receive messages on the lock screen [24].

Vocapia is a software suite that uses advanced speech analysis technologies, including speech recognition, speech detection, speaker recording, and speech-to-text alignment. This tool can convert audio and video documents, such as broadcasts and parliamentary hearings, into text format. It works with different languages and provides web services through a REST API to convert speech to text. It is also possible to use services for analysing telephone speech and creating subtitles for videos [25]. Vocapia Research specializes in the creation of advanced multilingual speech processing technologies that use artificial intelligence techniques, in particular, machine learning. These technologies provide continuous large-vocabulary speech recognition, automatic audio segmentation, speech recognition, speaker recording, and audio-to-text synchronization. The Vocapia VoxSigma™ speech-to-text software suite guarantees high performance for different languages when processing various types of audio data, such as broadcasts, parliamentary hearings, and conversational data conversion [26].

SoundHound is an innovative language platform that uses artificial intelligence and is developed based on the company's unique technology. It provides a unique voice interface with individual customization and full transparency in data usage. The popular sound recognition app SoundHound has a wide range of uses. With additional features such as lyrics display and voice search, it goes beyond music identification. SoundHound's music recognition capabilities cover a variety of input methods, such as humming, singing, or typing. The app gives users access to detailed song information, including artist bios, music videos, song previews, and the ability to share discoveries with friends and on social networking sites. The main advantages of SoundHound are:

- Performs both steps of the speech-to-text process and vice versa in one step, eliminating the need for a conventional two-step approach. This allows you to get results faster and more accurately;
- The voice assistant can effectively answer several questions at the same time and refine the results, taking into account the user's intentions to respond to complex questions;
- The multimodal interface provides customers with the opportunity to receive immediate interaction in real-time, thanks to instant audiovisual feedback, and also allows changing or updating requests using voice commands and a touch screen [27].

When it comes to voice recognition and intelligent assistance on iPhone, iPad and other Apple devices, Siri, Apple's virtual assistant, has become known as the industry standard. Users can interact with their devices using natural language commands and voice requests thanks to Siri's voice recognition capabilities. Siri uses speech recognition technology to understand and respond to user voice input, making everyday tasks more convenient and hands-free, whether users are setting reminders, sending messages, making calls, or getting information.

Google Assistant, which uses voice recognition technology, is a well-known virtual assistant that can be found on Android devices and other platforms. Google Assistant interprets voice commands, provides answers to requests, performs tasks and controls compatible smart devices thanks to its sophisticated speech recognition capabilities. Through integration with various Google services, users can access personalized information, receive personalized

recommendations and have a seamless voice experience across devices.

These modern apps and sound recognition apps have completely changed the way we interact with technology and discover music. They have become essential tools for music enthusiasts, offering personalized and immersive sound thanks to their precise recognition algorithms and user-friendly interfaces. The addition of voice recognition technology to virtual assistants like Siri, Google Assistant, and Cortana has also changed the way we interact with our devices, making tasks more convenient and accessible through voice commands. In summary, sound and audio signals play a fundamental role in our lives, providing communication, entertainment and perception of the world around us. Understanding the nature of sound waves and their representation in the digital realm is essential to working effectively with sound and audio signals. As technology advances, sound and audio signal processing will continue to evolve, providing new opportunities for creative expression, immersive experiences, and innovative applications in a variety of fields.

We will conduct a comparison of analogues with the product under development, for this we will develop a comparison table of development and analogues. However, first, you need to define a grading system. Scores will be integers from 1 to 10, where 1 is very bad (not developed), and 10 is perfect, if it is impossible to answer the question with a score, then Boolean values yes/no can be used (Table 1).

Table 1. Comparative table of the developed product and analogues

Characteristics	Products				
	Development	Shazam	Sound recognition from Apple	Vocapia	SoundHound
WEB	7	8	9	5	9
Mobile	7	6	7	6	7
Functionality	8	8	8	2	8
Reliability	8	9	10	7	8
Productivity	8	7	7	7	9
Maintainability	8	6	8	4	9
Programming language	10	7	8	7	7
Convenience	9	8	9	7	10
Intelligibility	8	8	9	5	8
Security	9	7	8	7	8
Suitability for use	9	6	7	7	9

We will define innovations and evaluate the effect they will have on development. In advance, for this, you need to develop a rating system: Numerical rating (from 0 to 10); and Boolean evaluation (yes or no). Based on a developed evaluation system and goals, we will build a table for evaluating the effects of the developed product (Table 2):

Table 2. Estimates of development effects

Target	Effect	Units of measurement	Value of assessment
Development of the project is financially beneficial in the field of technical application	financial	Numerical	7
Development of functional methods	economic	Numerical	8
Development of optimization methods	temporal	Numerical	7
Develop a guide to using the methods	educational	Numerical	3
Develop an advertising campaign	economic	Numerical	no
Develop a multi-cloud platform	technical	Numerical	4
Develop a 24/7 online customer support system	economic	Numerical	3
Placing social ads on the platform	social	Boolean	no

After conducting the work in this section, we can say with confidence that the chosen research topic has a great demand and a wide range of applications, which makes the developed product universal for future use.

3. Material and Methods

3.1. Methods of Sound Signal Processing

Audio signal processing techniques allow you to manipulate, enhance, and analyse audio signals. Here are some common audio signal processing techniques:

- Audio effects: Audio effects change the characteristics of audio signals to achieve certain artistic or technical goals. Examples of sound effects include reverb, delay, chorus, flanger, and distortion. These effects can be applied to individual tracks or mixed audio to create desired textures, atmospheres or stylistic elements.
- Dynamic Range Compression: Dynamic Range Compression aims to reduce the difference between the loudest and quietest parts of an audio signal. It is commonly used in audio mastering and broadcasting to ensure a consistent perceived volume level. Compression methods include ratio, threshold, attack time, release time, and compensation gain control.
- Pitch Shifting and Time Stretching: Pitch shifting changes the perceived pitch of an audio signal without affecting its duration, while time stretching changes the duration without changing the pitch. These techniques are used in music production, sound design, and audio post-production to correct pitch, create harmony, or manipulate the tempo of audio recordings.
- Noise Reduction: Noise reduction techniques aim to minimize unwanted background noise or artifacts in audio signals. A variety of algorithms and filters, such as spectral subtraction, Wiener filtering, and noise gating, are used to reduce noise while preserving the desired audio content.
- Audio Equalization: Audio Equalization adjusts the frequency response of an audio signal to boost or cut certain frequency ranges. It is used to correct tonal imbalance, shape sound or compensate room acoustics. Graphic equalizers, parametric equalizers, and shelving filters are commonly used in an audio equalizer.
- Spatial audio processing. Spatial audio processing techniques manipulate audio signals to create an immersive sound field with precise localization and spatial cues. These techniques include techniques such as panning, spatial filtering, and binaural reproduction to create a realistic auditory environment.

These techniques, among many others, form the basis of audio signal processing, allowing artists, engineers and researchers to shape and transform audio signals to suit their creative and technical requirements.

3.2. Problems and Considerations in Sound and audio Signal Processing

Although sound and audio signal processing offer many opportunities, there are also a number of challenges and factors that must be taken into account. Below are some of the main difficulties.

- The time that elapses between the input of an audio signal and its processed output is called delay. Low latency is essential to maintain synchronization and responsiveness in real-time applications such as live performances, games, and communication systems.
- Computational complexity involved: A number of complex audios processing techniques, including convolution, reverberation and spatial rendering of audio, can be computationally intensive.
- Such algorithms must be processed in real-time, subject to hardware or platform limitations. This requires effective implementation strategies.
- Audio signal processing constantly strives to provide high quality sound processing, reducing artifacts and distortion. It can be difficult to find the right balance between artistic or technical goals and maintaining sound fidelity.
- Human perception of audio is complex and individual. Factors such as psychoacoustics, individual hearing differences, and cultural preferences should be considered when designing sound processing algorithms and systems.
- Due to the wide range of audio formats, devices and platforms available, ensuring compatibility and interoperability between different systems and formats can be difficult. The Audio Engineering Society (AES) and Audio Video Interleave (AVI) are standards and protocols that help solve these problems.
- In addition, ethical issues such as privacy, security, and accessibility must be taken into account when designing and implementing audio signal processing systems.

3.3. Traditional Methods of Sound Recognition

Traditional sound recognition techniques encompass a number of approaches, including rule-based systems and statistical methods. These methods use signal processing techniques and machine learning algorithms to analyze audio signals and classify them into different sound categories. Here is a brief overview of these approaches.

Rule-based systems rely on predefined rules and heuristics to recognize sounds. These rules are usually developed by experts in the field based on their knowledge of the characteristics and features of specific classes of sound. The rules are designed to capture the discriminative properties of different sounds and use signal processing techniques to extract relevant features from the audio data. For example, in speech recognition, rule-based systems can use techniques such as phonetic analysis, language modeling, and grammar rules to identify words or phrases. Similarly, in environmental sound recognition, rule-based systems can use features such as spectral characteristics, temporal patterns, and amplitude variations to distinguish between different classes of sounds. Although rule-based systems provide interpretability and explicit control over the recognition process, they may be limited in their ability to handle complex or ambiguous audio scenarios. Creating rules manually can be time-consuming, and performance can depend heavily on the experience of the rule developer.

Statistical methods use mathematical models and algorithms to analyze sound signals and make decisions based on statistical properties. These methods often involve feature extraction, where relevant acoustic features are extracted from the audio data, followed by a classification step.

Signal processing techniques are used to extract discriminative features from audio signals, which are then used for classification. Commonly used features include:

- Mel-Frequency Cepstral Coefficients (MFCC): MFCCs capture the spectral characteristics of sound by analyzing frequency bands and their amplitudes.
- spectral centroid: This function represents the center of mass of the frequency distribution and provides information about the perceived brightness or darkness of the sound.
- zero crossing rate: This function counts the number of times the audio signal crosses the zero amplitude line and indicates the time characteristics of the sound.
- short-time Fourier transform (STFT): STFT decomposes an audio signal into frequency components over time, allowing analysis of spectral content and changes.

After feature extraction, statistical models such as Hidden Markov Models (HMMs), Gaussian Mixture Models (GMMs) or Support Vector Machines (SVMs) are trained to classify the extracted features into different sound classes.

Machine/deep learning algorithms are used for learning:

- patterns and relationships from audio data without explicit rules. These algorithms are trained on labeled audio samples to create models that can automatically classify new, unseen audio data.
- supervised learning: In supervised learning, a labeled data set is used to train a classification model of audio samples. Popular algorithms include decision trees, random forests, naive Bayes algorithms, and various neural network architectures such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs).
- unsupervised learning: unsupervised learning techniques aim to detect patterns or groupings in audio data without predefined labels. Clustering algorithms such as K-means or Gaussian Mixture Models (GMM) can be used to identify similar patterns or sound clusters.
- deep learning: Deep learning models, such as deep neural networks, have gained considerable popularity in audio recognition tasks. Deep learning architectures can automatically learn hierarchical representations of audio data, resulting in improved classification accuracy. Convolutional neural networks (CNNs) are often used to analyze spectrograms or other image-like representations of audio, while recurrent neural networks (RNNs) and variants such as long-short-term memory (LSTM) networks are suitable for sequential audio like speech or music.

These statistical and machine/deep learning methods have the advantage of being able to process complex audio data and adapt to different sound environments. They can study large volumes of labeled data and automatically detect discriminative features, providing more reliable and accurate sound recognition.

However, these methods also require large labeled datasets for training and can be sensitive to changes in acoustic conditions and recording parameters. In addition, the effectiveness of these methods largely depends on the quality and representativeness of the extracted features and the availability of a variety of training data.

3.4. System Functioning Purpose Analysis

Before starting work in this section, let's define what a system analysis is and what it is used for. System analysis is aimed at studying the system or its constituent parts to determine its goals. This method of problem-solving is aimed at improving the system and ensuring that all its components work effectively to achieve the objectives. System analysis determines what functions the system should perform [28]. The purpose of this work is software for the analysis and recognition of sounds in the metropolis using machine learning methods, which will help people in everyday life, in farming, in military affairs and public order. As a final result of the project, the user should be able to use a mobile phone or a website to recognize any sound with a microphone connected to the device, in real-time or using pre-recorded audio. To achieve the main goal, we will highlight goals that will help implement the project:

- Receive data with different sounds, process the data and analyse the received data. This goal must be fulfilled in the first version of the developed product;
- Determine the machine learning method that is most suitable for solving the given problem: sound recognition. This must be done in the first version of the product under development;
- Implement the best machine learning method obtained after completing the previous goal. It must be done in the first version of the product under development;
- Develop an API for user usability. Not critical for the first version of the developed product;
- Develop the user interface. Not critical for the first version of the product under development.

3.5. Modelling System Requirements and Risks

User requirements are documents that specify in detail how a system, tool, or process must function to meet user

requirements and expectations. The user can be both a person and a machine that actively interacts with the process or system [29]. To determine the requirements, we will build a table of the order of formulating the requirements for the system (Table 3):

Table 3. Procedure for formulating system requirements

Type of requirements	Business requirements	User requirements	Functional requirements	Non-functional requirements
Appointment	1. Full development of functional methods using machine learning 2. To optimize the operation of methods 3. Development of a guide for users 4. Make the platform multi-cloud 5. Develop social advertisements 6. Conducting an advertising campaign 7. Development of a 24/7 customer support system	1. Ability to download audio from users. 2. Displaying the results on the screen. 3. Authorization in the system. 4. Fair subscription system.	1. Implemented methods of machine learning data processing. 2. Presentation of data on the screen. 3. Reading audio in real-time. 4. Review history 5. Downloading audio. 6. Saving audio 7. Possibility of vibration mode to determine the danger.	1. Methods must process information within 2 seconds. 2. The results must be presented in an understandable form. 3. The system must work reliably 4. Audio must be stored by the subscription
Example of the content of the requirements	1. This product is intended for people who need help with sound recognition from various fields. 2. Works with data received from customers in real-time and with records. 3. Effects: financial, economic, social, educational, temporal	1. Users systems may be users who depend on or work with sound or need help with recognition. 2. Users receive recognized sound. 3. Effects: temporary, economic 4. You need to register, issue a subscription, and choose a method of work, and audio for them.	1. Performs user authorization in the product, and processes data using machine learning. 2. Data on various sounds 3. Choice of registration or logging in, choice of system operation method. 4. Data processing should be carried out in real-time or with recordings.	1. Users are divided by subscriptions to which types of activities are tied and by the history retention length. 2. Quality attributes: clarity, comprehensibility, reliability 3. Requirements: data must be processed quickly; results must be clear and accessible.

Description of the RUP standard case scenario deployment:

- *Stakeholders of the precedent and their requirements:*

  - User – wants to recognize the sound.
  - Data analyst – wants to find data and build machine learning models to recognize sounds.

- *The user of the product is the main actor of this precedent:*

It is generally the user who will choose the work methods he needs for sound recognition.

- *Preconditions of the precedent (preconditions):*

  - The product under development must be functional;
  - Developers need to find data to train machine learning methods;
  - Developers should find a way to receive data from users and microphones;
  - Developers should find an opportunity to process user data;
  - The data must be correct;
  - The payment system must work properly;

- *The main successful scenario:*

  - The user logs into the system, registers/authorized;
  - The user is authenticated;
  - The user pays for the subscription;
  - The user allows access to audio and microphone;
  - The user selects the operating mode;
  - The user provides audio;
  - The user receives recognition results;

- *Expansion of the main script or alternative streams:*

The user chooses the type of recognition:

- Danger recognition mode;
- "Protection" recognition mode;
- Audio recording recognition mode;
- Real-time recognition mode.

- *Post-conditions:*

- The user received the results;
- Audio saved;

- *Special JI:*

- Ensure the reliability of data transmission;
- Provide a convenient interface;
- Provide round-the-clock support;
- To ensure fast processing of the request.

- *List of necessary technologies and additional devices:*

- The product under development must be a web platform or a mobile application;
- Device for visual display of results.
- Microphone.

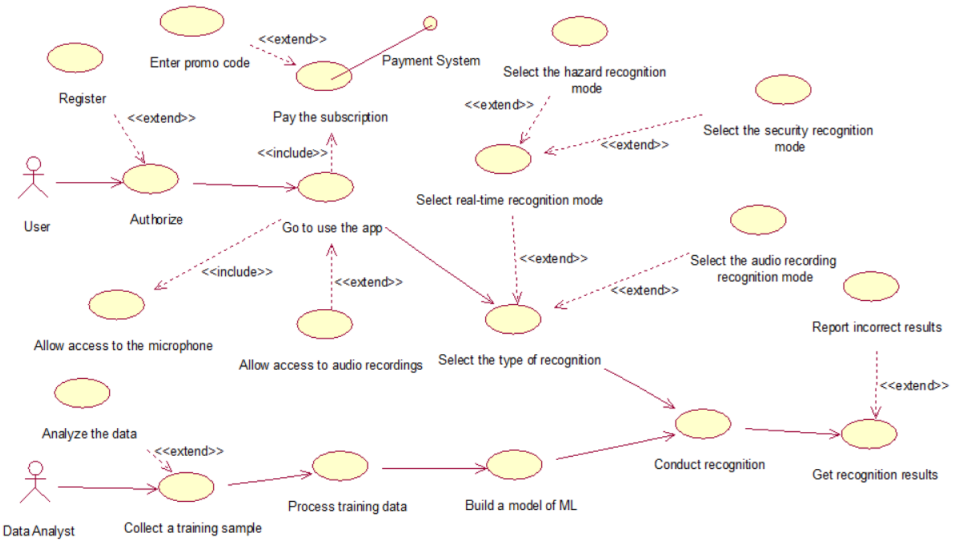


Fig.2. Use case diagram

Let's build a use case diagram to illustrate the functional content of our data analysis system (Fig. 2). A use case diagram (sometimes known as a use case diagram) is a general representation of the functional purpose of a system that answers the key modelling question: What does the system do in the outside world? Fig. 2 shows a diagram of use cases. The diagram shows two actors: The user and the Data Analyst. For successful work, a data analyst must:

- Collect a study sample;
- Analyse data;
- Process training data;
- Build a machine learning model.

To work in the system, the user needs:

- Log in to the system, if the user does not enter the system for the first time, then it is necessary to register, the user can also use a promotional code;

- The user must proceed to use the application, at the same time give the system access to audio recordings and the microphone and pay for the subscription, if this has not already been done;
- The user must select the recognition type by pointing to the microphone or audio recording. The user can choose 1 of the operating modes: danger recognition mode, "Protection" recognition mode; audio recording recognition mode, or real-time recognition mode;
- The user goes to the recognition step;
- The user receives a result and can leave feedback on incorrect recognition.

Table 4. Description of object classes of the system of sound analysis and recognition in the metropolis

Object classes		Class attributes		Class methods	
Class name	Class assignments	Attribute name	Attribute content	Method name	Method content
Management	A class that implements system management	Path to Audio	A record that contains the audio path	Download Model	Loads the ML model for recognition
		Path to ML	A record that contains the path to the MN model	Select Audio	A method that selects audio from the user's library
		Recognition type	The record that contains the selected recognition work type	Select Recognition Type	A method that sets the type of recognition selected by the user
System management	A class that follows Control, a user session in the system				
Result	Implements work with results	ResultRecognition	The sound class that was recognized	Save Recognition	Stores recognition in the system
				DeleteResult	Deletes the result from the system
Subscription	Implements the possibility of user subscription for access to the system	NameSubscriptions	Subscription display name	Pay	Subscription payment (wrapper for third-party payment system)
		Description Subscriptions	Contains a detailed description of subscription options		
		Cost of subscriptions	Subscription price	CheckPayment	Checks the validity of a user's subscription
		Eligibility term	User subscription expiration date		
Authentication	Implements user authorization and authentication	ID	User ID in the system	Register	Registers a user in the system
		Login	Unique username feed		
		Password	User's secret feed	Sign in	Performs user login
		Electronic Mail	User email		
		Name	Display Name	Delete Account	Deletes the user's account permanently
Audio	Implements audio display in the system	File name	File display and ID in the system	RecordAudio	Record audio from the user's microphone
		PathToFile	Audio placement path	DownloadAudio	Downloading user audio
				DeleteAudio	Removes the user's audio from the system
				ProcessAudio	Processing of user audio for further recognition
ML	Implements the ML model class	Study sample	Data for model training	Train ML	Trains a machine learning model
		Test data	Data for testing the trained model	Protest ML	Tests a machine learning model
				Save ML	Saves the model in the system
				RecognizeAudio	Recognizes the sound specified by the user

3.6. Modelling of Subject Area Objects

Let's start by defining the classes, their methods and attributes that implement the data analysis system. A class diagram is used to display the static structure of a system model, using the terminology of classes in object-oriented programming. This diagram is a form of a graph, where the vertices are elements of the "classifier" type, which are connected by various types of structural relations. It is important to note that a class diagram can also include relationships, packages, interfaces, and even individual instances such as relationships and objects. In general, this diagram represents a static structural model of the system, which is considered a graphical representation of such

structural relationships of the logical model of the system that remain constant concerning time. Let's build a table of system class descriptions (Table 4):

Let's build a table to determine the relationship between classes in Table 5 and draw a class diagram in Fig. 3.

Table 5. Description of relations between classes of the system of analysis and recognition of sounds in the metropolis

Relationship name	Classes between which a relationship is defined		Type of relation	Dimensionality
Relationship 1	System management	Audio	Aggregation	1-0..n
Relationship 2	System management	Subscription	Aggregation	1-1
Relationship 3	System management	ML	Aggregation	1-1..n
Relationship 4	System management	Authentication	Aggregation	1-1..n
Relationship 5	System management	Management	Follows	-
Relationship 6	System management	Result	Aggregation	1-0..n

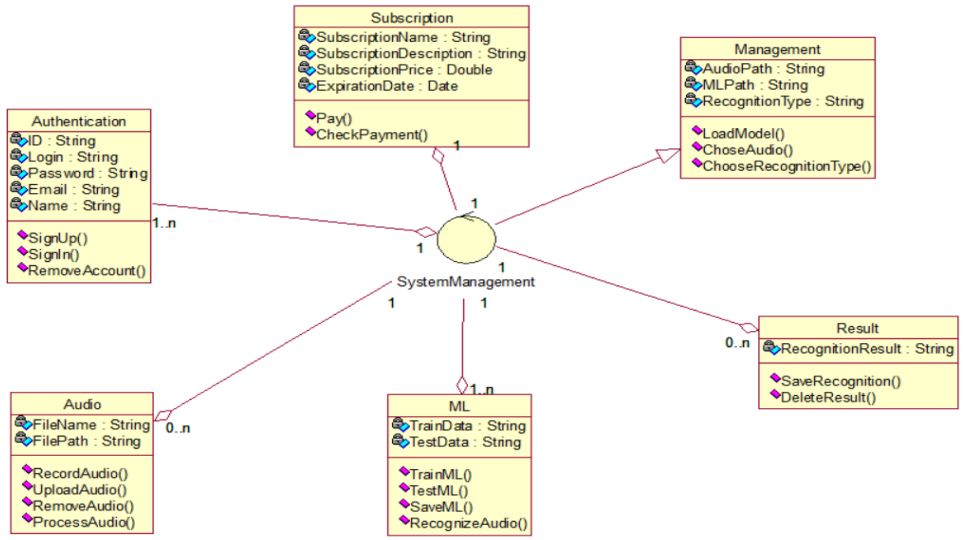


Fig.3. Class diagram

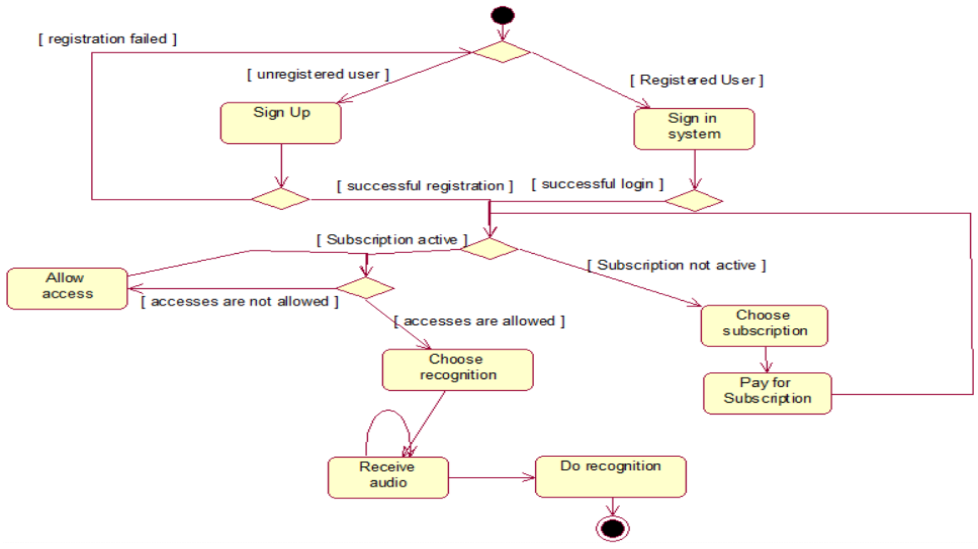


Fig.4. Activity diagram

3.7. Modelling of System Processes

An activity diagram (Fig. 4) is an alternative to a state diagram, and it is based on a difference in approach: activities are the main component in an activity diagram, while static state is important in a state diagram. In the context of an activity diagram, the key element is the "action state", which specifies the expression of a specific action that must be unique within the given diagram. An action state is a specific state initiated by certain input actions and has at least one output. The visualization of an activity diagram is similar to a graph of a finite automaton, where vertices correspond to specific actions and transitions occur after actions are completed. An action acts as the basic unit of

behaviour definition in a specification, taking a set of inputs and converting them into outputs. Let's draw an activity diagram for our system:

This diagram shows the activities that occur during program execution:

- If the user is not registered, he is registered in the system;
- If the user is already registered, he enters the system;
- If registration or login is unsuccessful, the user returns to the beginning;
- If the user does not have a paid subscription, then he must choose a subscription and pay for it;
- If the user has an active subscription and has allowed all settings, he enters the recognition stage, otherwise, he must allow access and return to the previous stage;
- Next, the User selects the recognition mode and provides audio;
- After which the recognition is done and the user receives the result.

Let's build a sequence diagram to determine the sequence of interaction of objects in chronological order (Fig. 5). The diagram provides information about the order of events and messages between objects during a certain period. This simplifies the perception of the sequence of various actions and their interaction. The diagram also indicates how objects interact in a particular scenario, emphasizes parallel execution of actions, and can indicate synchronization between objects. Chronological sequence of system actions:

- A data analyst trains a machine learning model;
- The data analyst tests the machine learning model;
- The user registers in the system;
- The user logs into the system;
- The user pays for the subscription;
- Management of the system checks the payment;
- The user records audio;
- The Audio class processes audio;
- The user selects audio;
- The user chooses the type of recognition;
- The system loads the portable model;
- System management starts recognition;
- The user can save recognition.

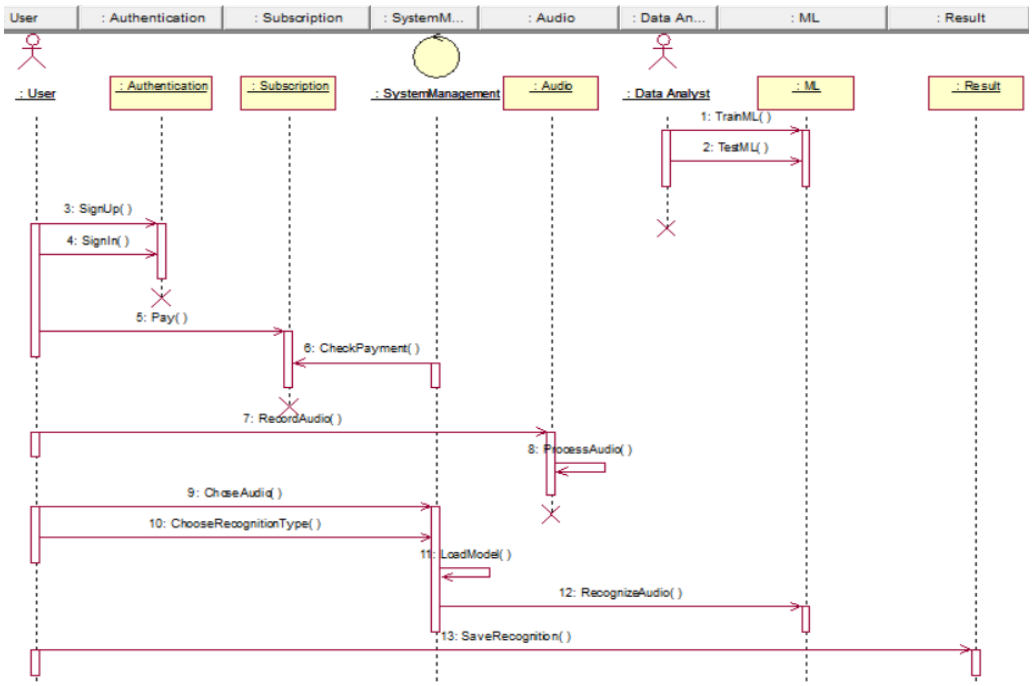


Fig.5. Sequence diagram

Let's build a diagram of components to visualize the architecture of the system structure in Fig. 6.

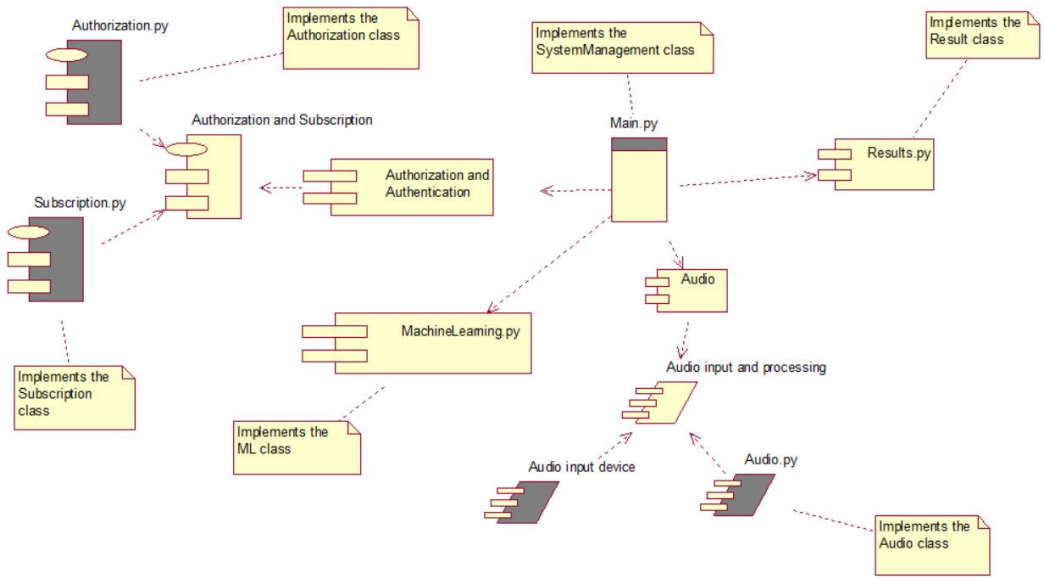


Fig.6. Component diagram

Fig. 6 is shown diagram of the components, which shows the structural interaction between the components:

- Main.py – implements the System Management class, this class operates the entire system;
- Results.py – implements the Result class;
- MachineLearning.py – implements the ML class;
- The authorization and authentication component contains the package specification, which is implemented using: Subscription.py – which implements the Subscription class and Authorization.py implements the Authorization class;
- The audio component contains the task specification - audio input and processing, which is implemented by the audio input device and Audio.py, which implements the Audio class.

Let's build a deployment diagram to see how users interact with the system in Fig. 7.

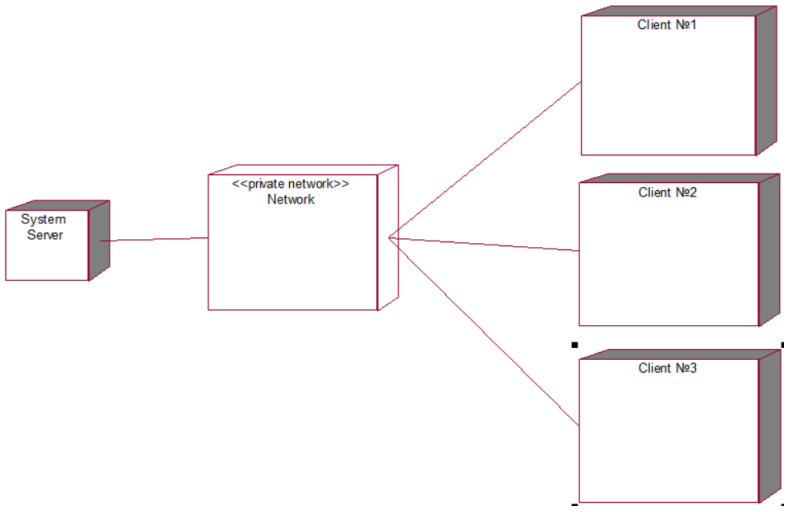


Fig.7. Deployment diagram

The deployment diagram shows 3 processors and 1 device. System processors:

- User #1 – system user, can be connected both from the phone and the web application. Has a network connection with the device.
- User #2 – system user, can be connected both from the phone and the web application. Has a network connection with the device.
- User #3 – system user, can be connected both from the phone and the web application. Has a network connection with the device.

- The system server is the place where all the activities of the program are performed, from receiving data from the user to returning the results to the user. Has a network connection with the device.

Devices:

- Network – a closed network that performs the function of communicating between the user and the system. It has a connection with the processors: User #1, User #2, User #3 and with the System Server.

The above experiments will significantly help and simplify the development of the project because the main concepts and principles of the system have already been described.

4. Experiments

4.1. Formulation and Justification of the Problem

The task of this work is to create software for the analysis and recognition of sounds in megacities using machine learning methods. A working machine learning model that can recognize certain types of sounds can be considered a good performance of this work.

This system can be used by users of various industries. Nowadays, loud sounds have become a constant stress, so even at home, this system will be able to help distinguish a real explosion from falling materials at a construction site. In farming, this drone can be used to automate the driving away of animals from crops. In military affairs, the data of this system can help with the detection of extraneous sounds, for example, drones, and in the future, the recognition of the distance of the sound echo, which can help in easier detection of enemy equipment and missiles. In cities, it can help with security, so a preventive response system can be built, which can check whether everything is in order based on sounds. Also, it can make life easier for people with impaired hearing to detect danger in everyday life.

The system is very easy to use, a user of any age and profession will be able to use it easily. All you need is a smartphone and a microphone. The effects obtained from the implementation of the project:

- Economical – investment in installing cheaper microphones instead of video surveillance cameras.
- Temporary - the period of people's reaction to a certain loud sound will decrease.
- Social - help to people with impaired hearing.
- Educational – can inspire people to study sound and machine learning techniques even more carefully.

4.2. Building a Model for Problem-solving

It is necessary to build a model of the machine learning method. For this, we need:

$$Get\ data \rightarrow Analyse\ data \rightarrow Visualize\ data \rightarrow Choose\ the\ best\ method \rightarrow Build\ a\ model$$

To begin with, consider the possible methods of machine learning [30-42]:

Support vector classification (SVC) in machine learning uses support vectors to efficiently classify objects. The support vectors are the points closest to the class separation boundary, and they define the hyperplane that separates the classes in the feature space. The method also uses functions to solve nonlinear classification problems. The regularization parameter controls the trade-off between the accuracy of the training data and the generality of the model. It is used in various fields such as pattern recognition, bioinformatics and financial analytics [30].

ExtraTreesClassifier is a classification method that uses a combination of decision trees and is known for its extreme randomness. When using it to build a model, features and thresholds are randomly selected for each tree node. All this is intended to reduce the risk of overtraining and ensure greater stability of the model. ExtraTreesClassifier allows solving classification tasks where it is necessary to determine whether objects belong to certain categories based on their features [31].

AdaBoost is a machine learning method that uses a sequential approach to training classifiers. At each step, the algorithm trains a weak classifier and assigns a weight to it based on how effectively it classified the data. By combining these classifiers, AdaBoost tries to focus on those areas of the data where previous classifiers made mistakes. The main idea is to train new classifiers in such a way that they focus on objects that were incorrectly classified by previous classifiers. The importance of each classifier is governed by its accuracy. Thanks to this approach, AdaBoost can create a strong classifier that can effectively solve classification problems, even if weak classifiers are included [32].

MLPClassifier is a classifier that uses an artificial neural network to solve the classification problem in machine learning. It belongs to the scikit-learn library and is used to model complex relationships in input data. A neural network has different layers, such as input, hidden, and output, and learns the relationships between them during training. MLPClassifier is used to solve classification problems, where the model tries to understand the relationship between the input data and the target class. Various parameters of this classifier, such as the number of layers, the number of neurons in each layer, and the learning rate, can be adjusted to achieve optimal results on specific data [33].

4.3. Selection and Justification of Problem-solving Methods

The best methodology in our case is the use of Agile methodologies. Agile software development has several advantages that justify its use:

- Flexibility and adaptability: Agile provide the ability to respond quickly to changing requirements, even at late stages of development. This is especially useful in a changing business environment and user requirements.
- Iterative approach: Agile involves developing a product in small iterations (sprints), which allows the user to obtain the functionality of the product at each stage of development and make adjustments.
- Involvement of the user and teamwork: The methodology promotes active interaction between the user and the development team. This helps to better understand the requirements and resolve possible misunderstandings.
- Better product quality: More frequent integration testing and small, frequent releases allow bugs to be discovered and fixed more quickly, resulting in a higher-quality software product.
- User satisfaction: Agile allows the user to see the results of the work at early stages and actively influence the development process, which contributes to the user's satisfaction with the product.
- Containment of costs and risks: Agile allows you to control costs and reduce risks, as it allows you to quickly respond to changes in requirements or market conditions.
- Promotes team self-organization: Agile puts the focus on developing team self-organization, which can positively impact team productivity and creativity.

Therefore, development according to the Agile methodology contributes to the improvement of product quality, provides flexibility and speed of response to changes, and also improves interaction between the user and the development team. Therefore, since our product does not yet have a single target audience, it is better to develop flexibly to be able to change the product according to needs.

4.4. Development of Problem-solving Algorithms

To solve the given task, you need:

$$\textit{Analyse data} \rightarrow \textit{Find the best method for machine learning} \rightarrow \textit{Build a model} \rightarrow \textit{Test the developed model}$$

To analyse the data, you need:

$$\textit{Download data} \rightarrow \textit{Analyse the downloaded data} \rightarrow \textit{Visualize sound waves} \rightarrow \textit{Visualize spectrograms}$$

To find the best method, it is necessary to conduct experiments with various models and compare them. To build a model, you need to find out how you can implement the model. Test the model - you need to load the data from the test dataset and compare it with the expected ones.

4.5. Selection and Justification of Development Tools

To begin with, we need a laptop on which this data analysis system will be created, we have a Lenovo Legion and a cloud provider for raising our data analysis system. AWS - cloud development from Amazon, Google Cloud Platform - development from Google and Microsoft Azure - development from Microsoft can be identified as leaders among cloud providers. In our case, we will choose AWS as the services in it are perfectly described, and their development is very easy. From software resources, we need an OS. Popular OSes include Linux, Mac, and Windows. In our case, we will choose Windows as it is very convenient. We need text editors, in the case of Python, it is best to use Visual Studio Code, as it contains very convenient plugins for work. Also, we need a message broker among the most popular Kafka and RabbitMQ, since we will perform asynchronous tasks, it is better to use Kafka.

To implement this project, the following tools are used:

- Lenovo Legion laptop with Intel i5 gen 7 processor, 16 GB RAM and 1T hard disk;
- IBM Rational Rose for building UML diagrams;
- Visual Studio Code text editor;
- Apache Kafka – for sending asynchronous messages;
- Windows 10;
- AWS EC2 for system deployment.

Apache Kafka is a powerful data streaming system that is often used as a message broker between microservices. She has a number of strengths that make her an attractive choice for the role. It is designed to process huge streams of data in real time. Its high throughput means it can process hundreds of thousands or even millions of messages per second. This makes Kafka ideal for large systems that generate large amounts of data. In addition, Kafka is extremely reliable. It has built-in failover mechanisms that provide fault tolerance and ensure that messages are not lost if one or more nodes fail. Apache Kafka is also known for its flexibility. It supports various models of data processing, including

the implementation of both queues and topics (publish-subscribe model). This means that Kafka can be used to support a wide range of usage scenarios, from simple message forwarding to complex data processing flows. Finally, Kafka scales well, allowing you to increase the amount of data processing as the system grows. It supports a distributed architecture that allows you to add nodes to a Kafka cluster to process even more data.

To support service independence and rapid development, the following 4 types of services were selected:

- Main (SSO) Service: performs tasks of user authorization, storage of their projects, general data, etc.
- Audio Meta Service: deals only with data processing. Since the processing of audio files is a long, blocking and synchronous process, we moved it to a separate service and added operation queues using BullMQ.
- Render Service: an orchestrator of render machines, which, in cooperation with Kafka, operates render machines, creates render tasks and controls the process of their passage. It also has an API for managing project templates.
- Render Worker: a render engine that in turn reads a render task from a Kafka topic and starts rendering, downloading all required assets and files from S3

The AWS EC2 service was used as an environment that provides the opportunity to conduct a dialogue between users executing the server part.

In order to implement one of the possibilities of the project, which requires a separate environment for hosting the server part and testing the data transfer capabilities, one of the Amazon services was chosen. Amazon Elastic Compute Cloud (EC2) is one of the key services of the Amazon Web Services (AWS) platform. It provides scalable computing resources in the cloud, allowing users to easily run and manage virtual servers. To characterize it, several theses can be distinguished, such as:

- Flexibility - AWS EC2 allows you to choose the necessary computing resources (processor, RAM, storage) according to the requirements of your application or project. You can choose different instance types and sizes to optimize performance and cost;
- Scalability - EC2 allows you to scale your resources up or down to meet the changing needs of your application. You can easily change the number of instances, their sizes, and their configuration to ensure optimal performance.
- Automation - AWS EC2 provides an extensive set of tools to automate the configuration, deployment, and management of your instances. You can use services such as AWS CloudFormation, AWS Elastic Beanstalk, or Amazon EC2 Auto Scaling to simplify the process of deploying and managing your servers.
- High availability - EC2 provides the ability to deploy your instances in different regions and availability zones to ensure high availability. You can use different backup and monitoring strategies to ensure your application is up and running.

The advantages of using the cloud service Amazon Web Services Elastic Compute Cloud include the following points:

- EC2 provides an intuitive interface and documentation that helps you quickly configure and manage virtual servers.
- You can scale EC2 resources up or down based on your growing business needs.
- EC2 provides high performance due to the use of powerful physical servers and the ability to choose the optimal configuration of resources.
- AWS EC2 offers a variety of security measures such as access control, resource isolation, and data encryption to ensure the reliability of your environment.
- AWS EC2 has a wide range of capabilities and integrates with other AWS services, which allows you to create complex and advanced architectures.
- AWS is a leading cloud service provider with extensive experience and a reliable infrastructure that ensures high availability and security of your data.
- AWS EC2 provides flexible pricing plans, including the ability to pay only for the resources used, which allows you to save money.

Overall, AWS EC2 is a powerful cloud computing service that provides flexibility, scalability, reliability, and security. It meets the needs of different businesses and applications, allowing them to easily manage computing resources in the cloud and focus on the development of their projects.

The following will be used for software development:

- Python is one of the most convenient programming languages for building machine-learning models;
- Sklearn – a library for building machine learning models;
- Librosa – a library for working with sound waves;
- Matplotlib and Seaborn – libraries for building visualizations;

- Numpy – a library for working with data arrays;
- Pandas – a library for working with datasets.

Python is a high-level programming language used to create a variety of programs. It is characterized by ease of study and readability of the code. Python is an interpreted language, which means that the execution of programs occurs when they are run, not at the pre-compilation stage. This language is popular among both beginners and experienced developers due to its versatility and various applications [34].

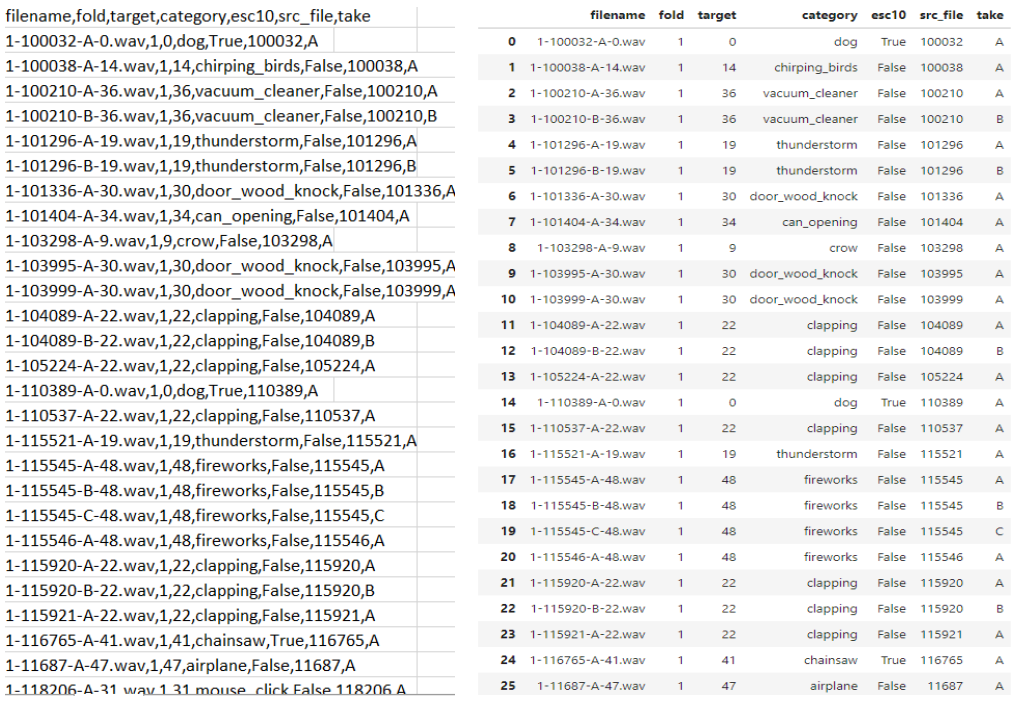
Sklearn is a machine-learning library for Python. It provides a variety of tools for performing tasks such as classification, regression, clustering, and measuring the quality of machine learning models. Scikit-learn is considered one of the key tools for developing and implementing ML algorithms in the Python environment [35].

Librosa is a software library designed to work with audio signals using the Python programming language. It is widely used in the field of audio processing, in particular in the analysis of musical signals. With Librosa, you can extract a variety of audio characteristics, such as spectrograms, mel spectrograms, tempo properties, and more. This allows research and development of algorithms for the analysis and processing of audio signals [36].

Matplotlib and *Seaborn* are libraries for data visualization in the Python programming language. Matplotlib is a core library for plotting and visualizing data in Python. It allows you to create various types of graphs, including line graphs, bar charts, pie charts, and many others [37]. Seaborn is a layer above Matplotlib that makes it easy to create attractive statistical plots. It provides a high-level interface for working with data and is used to create stylized graphs suitable for data analysis [38]. In general, Matplotlib is used for basic visualization and Seaborn adds styling and functions for statistical analysis of data.

NumPy is a library for the Python programming language aimed at processing mathematical and numerical operations. It allows the use of high-level data structures, in particular multidimensional arrays and matrices, and provides a powerful set of mathematical functions for efficient processing of these structures. Using NumPy is particularly useful for large amounts of data and solving scientific and engineering problems, as it allows operations on numerical data with high performance and efficiency. The multidimensional arrays provided by NumPy simplify a variety of calculations and data analysis, making the library popular in science and engineering fields [39].

Pandas is a library for the Python programming language aimed at processing and analysing data. It provides two basic data structures: a DataFrame, which represents a two-dimensional table, and a Series, which is used to represent one-dimensional data. A DataFrame is an efficient tool for organizing data in the form of a table with rows and columns. This is especially useful for data analysis and manipulation, similar to working with database tables. Series is used to represent a single row or column of data and can contain different types of information. Pandas make it easy to load, process and analyse data, making it indispensable in the field of data analytics and scientific research [40-42].



filename,fold,target,category,esc10,src_file,take
1-100032-A-0.wav,1,0,dog,True,100032,A
1-100038-A-14.wav,1,14,chirping_birds,False,100038,A
1-100210-A-36.wav,1,36,vacuum_cleaner,False,100210,A
1-100210-B-36.wav,1,36,vacuum_cleaner,False,100210,B
1-101296-A-19.wav,1,19,thunderstorm,False,101296,A
1-101296-B-19.wav,1,19,thunderstorm,False,101296,B
1-101336-A-30.wav,1,30,door_wood_knock,False,101336,A
1-101404-A-34.wav,1,34,can_opening,False,101404,A
1-103298-A-9.wav,1,9,crow,False,103298,A
1-103995-A-30.wav,1,30,door_wood_knock,False,103995,A
1-103999-A-30.wav,1,30,door_wood_knock,False,103999,A
1-104089-A-22.wav,1,22,clapping,False,104089,A
1-104089-B-22.wav,1,22,clapping,False,104089,B
1-105224-A-22.wav,1,22,clapping,False,105224,A
1-110389-A-0.wav,1,0,dog,True,110389,A
1-110537-A-22.wav,1,22,clapping,False,110537,A
1-115521-A-19.wav,1,19,thunderstorm,False,115521,A
1-115545-A-48.wav,1,48,fireworks,False,115545,A
1-115545-B-48.wav,1,48,fireworks,False,115545,B
1-115545-C-48.wav,1,48,fireworks,False,115545,C
1-115546-A-48.wav,1,48,fireworks,False,115546,A
1-115920-A-22.wav,1,22,clapping,False,115920,A
1-115920-B-22.wav,1,22,clapping,False,115920,B
1-115921-A-22.wav,1,22,clapping,False,115921,A
1-116765-A-41.wav,1,41,chainsaw,True,116765,A
1-11687-A-47.wav,1,47,airplane,False,11687,A
1-118206-A-31.wav,1,31,mouse_click,False,118206,A

	filename	fold	target	category	esc10	src_file	take
0	1-100032-A-0.wav	1	0	dog	True	100032	A
1	1-100038-A-14.wav	1	14	chirping_birds	False	100038	A
2	1-100210-A-36.wav	1	36	vacuum_cleaner	False	100210	A
3	1-100210-B-36.wav	1	36	vacuum_cleaner	False	100210	B
4	1-101296-A-19.wav	1	19	thunderstorm	False	101296	A
5	1-101296-B-19.wav	1	19	thunderstorm	False	101296	B
6	1-101336-A-30.wav	1	30	door_wood_knock	False	101336	A
7	1-101404-A-34.wav	1	34	can_opening	False	101404	A
8	1-103298-A-9.wav	1	9	crow	False	103298	A
9	1-103995-A-30.wav	1	30	door_wood_knock	False	103995	A
10	1-103999-A-30.wav	1	30	door_wood_knock	False	103999	A
11	1-104089-A-22.wav	1	22	clapping	False	104089	A
12	1-104089-B-22.wav	1	22	clapping	False	104089	B
13	1-105224-A-22.wav	1	22	clapping	False	105224	A
14	1-110389-A-0.wav	1	0	dog	True	110389	A
15	1-110537-A-22.wav	1	22	clapping	False	110537	A
16	1-115521-A-19.wav	1	19	thunderstorm	False	115521	A
17	1-115545-A-48.wav	1	48	fireworks	False	115545	A
18	1-115545-B-48.wav	1	48	fireworks	False	115545	B
19	1-115545-C-48.wav	1	48	fireworks	False	115545	C
20	1-115546-A-48.wav	1	48	fireworks	False	115546	A
21	1-115920-A-22.wav	1	22	clapping	False	115920	A
22	1-115920-B-22.wav	1	22	clapping	False	115920	B
23	1-115921-A-22.wav	1	22	clapping	False	115921	A
24	1-116765-A-41.wav	1	41	chainsaw	True	116765	A
25	1-11687-A-47.wav	1	47	airplane	False	11687	A

Fig.8. CSV file view

4.6. Description of the Developed Project Tools

First, let's analyze the dataset from the CSV file (Fig. 8). Sound data is in .wav format with 5 seconds each, total of 2,000 files. The data used is ESC-50, which consists of 50 classes of environmental audio dataset [43-45]. Fig. 8 shows

how the data is stored in CSV format, so the file contains 2000 records for 50 different sound classes. Dataset columns:

- Filename – audio file name;
- Fold – index of cross-validation;
- Target – a certain class of sound is within [0-49] (Table 1);
- Category – label of a certain sound;
- esc10 - indicates whether this file belongs to the ESC-10 subset;
- src_file – file source;
- take – a letter to disambiguate different fragments from the same Freesound clip.

The dataset consists of 5-second-long recordings organized into 50 semantical classes (with 40 examples per class) loosely arranged into 5 major categories (Table 6).

Table 6. A certain class of sound

Exterior/urban noises	Interior/domestic sounds	Human, non-speech sounds	Natural soundscapes & water sounds	Animals
Hand saw	Glass breaking	Drinking, sipping	Thunderstorm	Crow
Fireworks	Clock tick	Snoring	Toilet flush	Sheep
Airplane	Clock alarm	Brushing teeth	Pouring water	Insects (flying)
Church bells	Vacuum cleaner	Laughing	Wind	Hen
Train	Washing machine	Footsteps	Water drops	Cat
Engine	Can opening	Coughing	Chirping birds	Frog
Car horn	Door, wood creaks	Breathing	Crickets	Cow
Siren	Keyboard typing	Clapping	Crackling fire	Pig
Chainsaw	Mouse click	Sneezing	Sea waves	Rooster
Helicopter	Door knock	Crying baby	Rain	Dog

Let's build a distribution chart to see if the categories are evenly distributed (Fig. 9). From Fig. 9 we can see that the categories are evenly distributed, so this should not affect learning. Let's depict what sound waves of different categories look like, to understand how the categories, differ from each other, since in theory recognition should take place (Fig. 10-12):

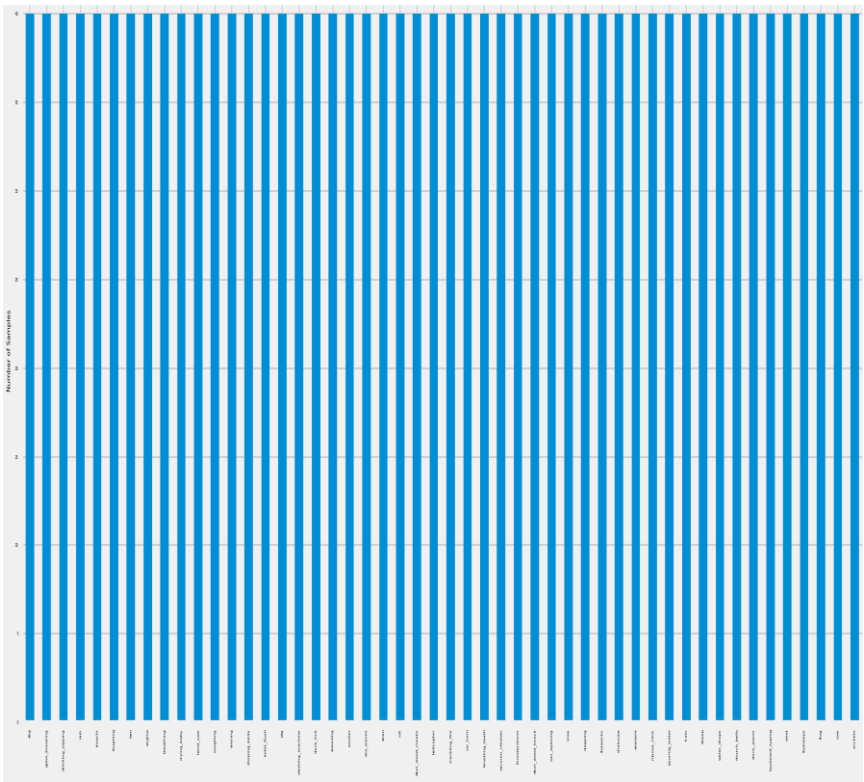


Fig.9. Distribution of categories (number of audio samples per category)

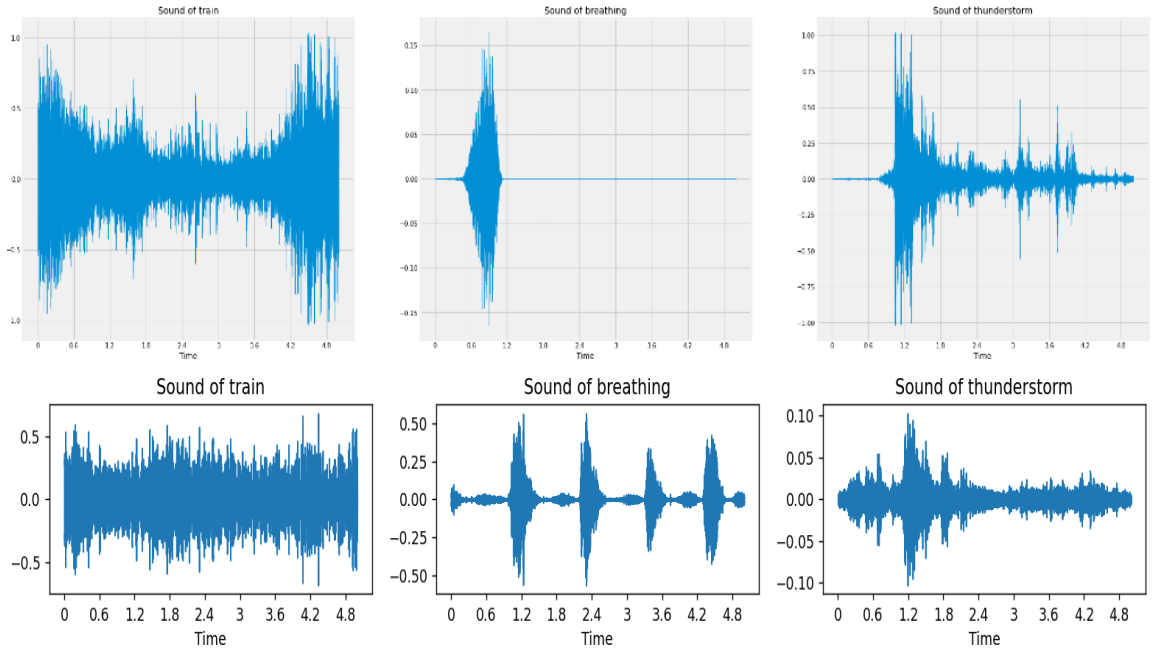


Fig.10. Shown are the sound waves of a train, breath, and thunder, respectively

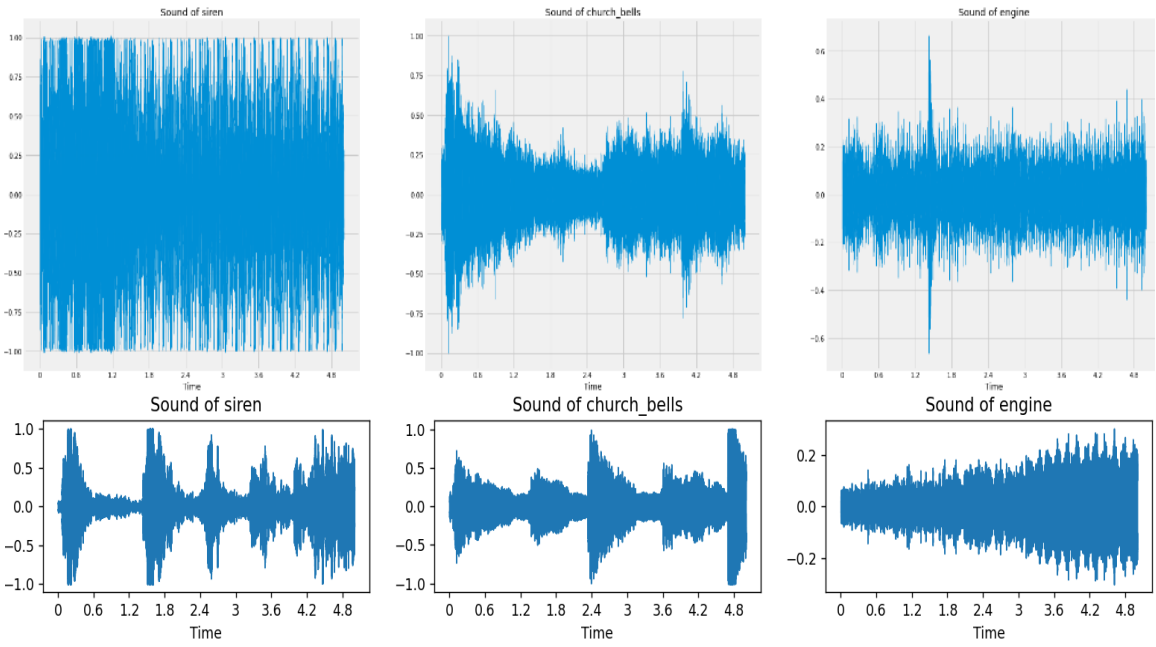
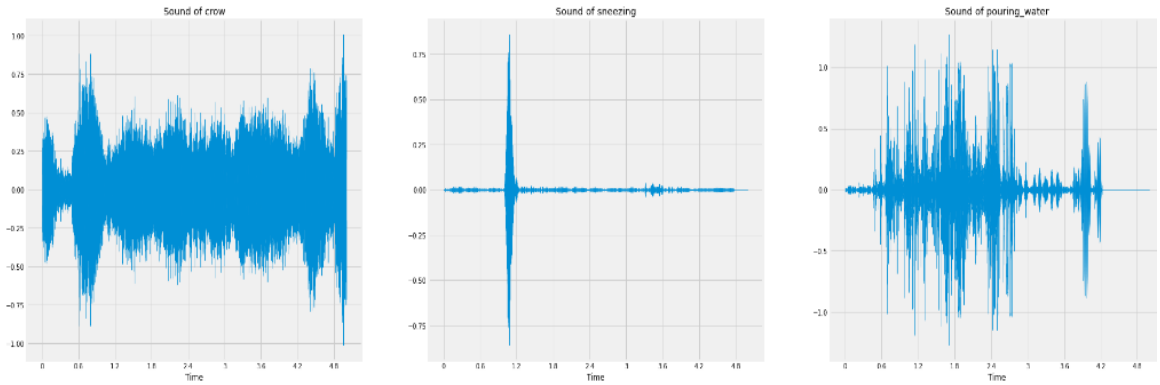


Fig.11. The sound waves of a siren, church bells and an engine are depicted, respectively



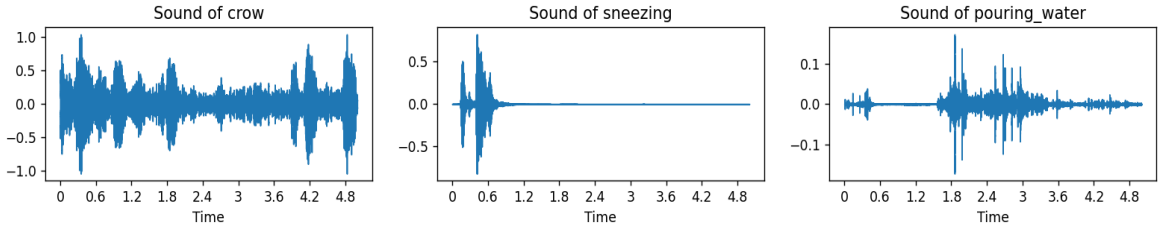


Fig.12. The sound waves of a crow's voice, a sneeze, and water pouring down are depicted

From the sound wave patterns (Fig. 10-12), we can see that most of the categories differ from each other significantly, which can help in the further creation of a machine-learning model that will determine the sound in the metropolis.

We will conduct a spectral analysis of the data (Fig. 13-15):

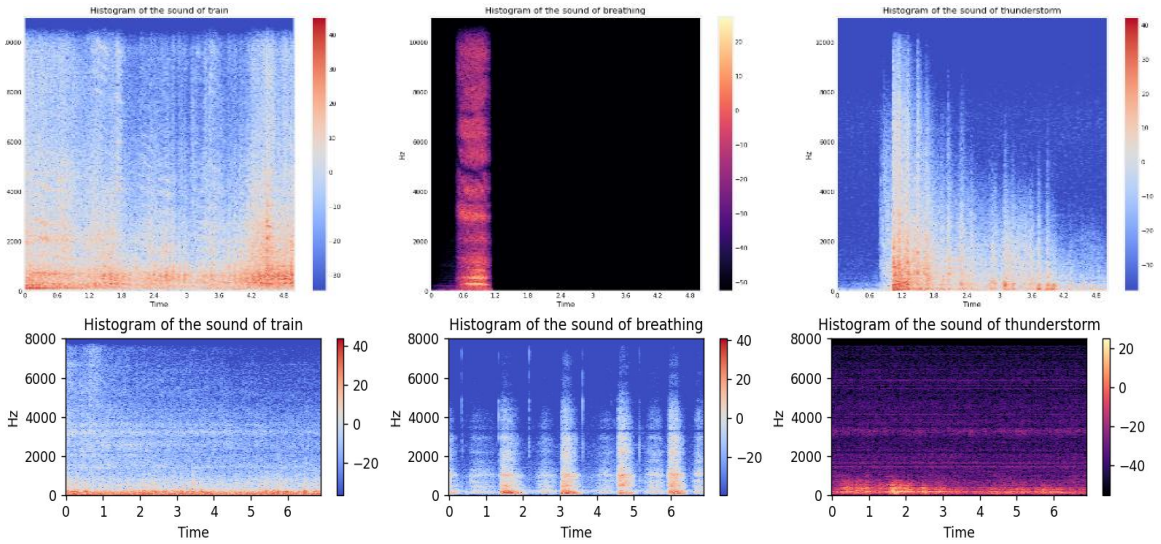


Fig.13. Spectrograms of train, breath and thunder are shown, respectively

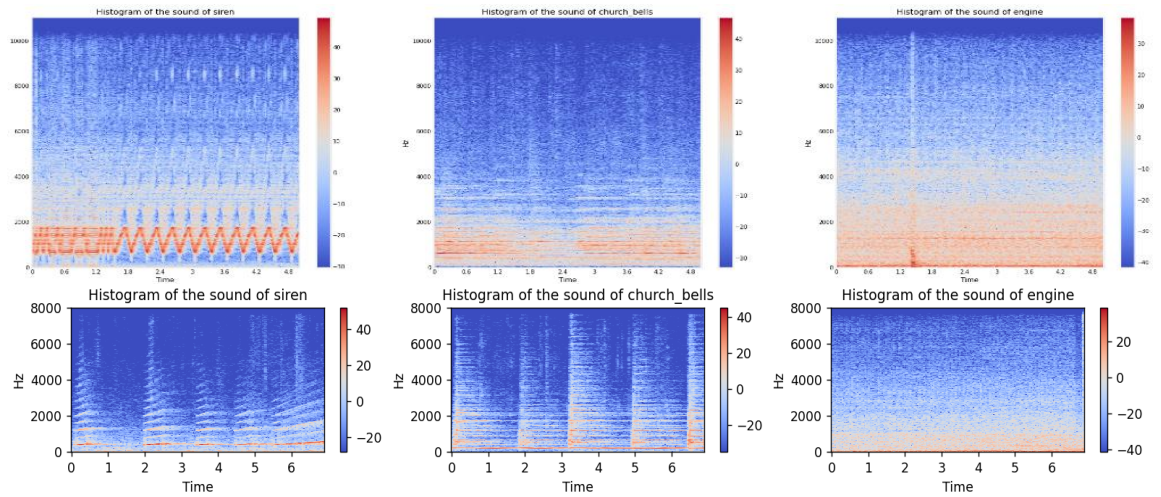


Fig.14. The spectrograms of the siren, church bells, and engine are shown, respectively

The image spectrograms in the figures above only prove that each class has its differences, which is great for sound recognition and machine learning training. Let's move on to the choice of machine learning methods. For this, it is necessary to artificially increase the data dataset by augmentation. Sound manipulation can include several techniques:

- Time Stretch: Changes the duration of the audio signal, making it either faster or slower. This can be used to emulate variations in audio tempo.
- Pitch Shift: Changes the pitch of a sound by raising or lowering it. This can be used to simulate variations in pitch.

- Volume scaling: changes the volume of the audio signal by increasing or decreasing its scale. It can emulate a variety of audio volumes.
- Add Noise: Adds noise to the audio signal to simulate various noise conditions.
- Time Shift: Changes the position of the audio signal in time, moving it forward or backwards. This can be useful for emulating variations in audio timing.
- Echo: Adds a delayed copy of the audio signal to create an echo effect.

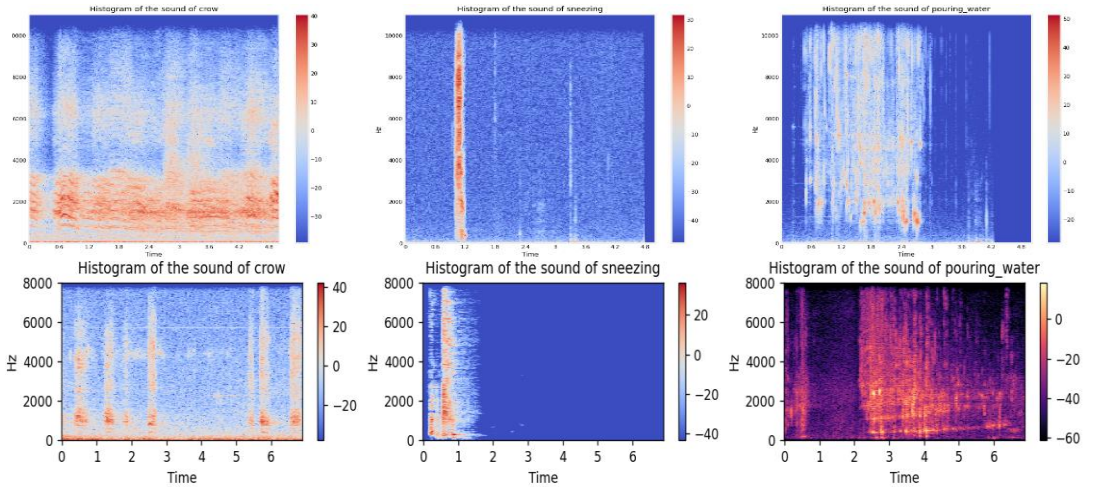


Fig.15. Spectrum scars of a crow's voice, a sneeze, and pouring water are depicted, respectively

Fig.16. View of the dataset after removing functions

After data augmentation, you need to extract functions, for audio, we will apply the following functions extraction:

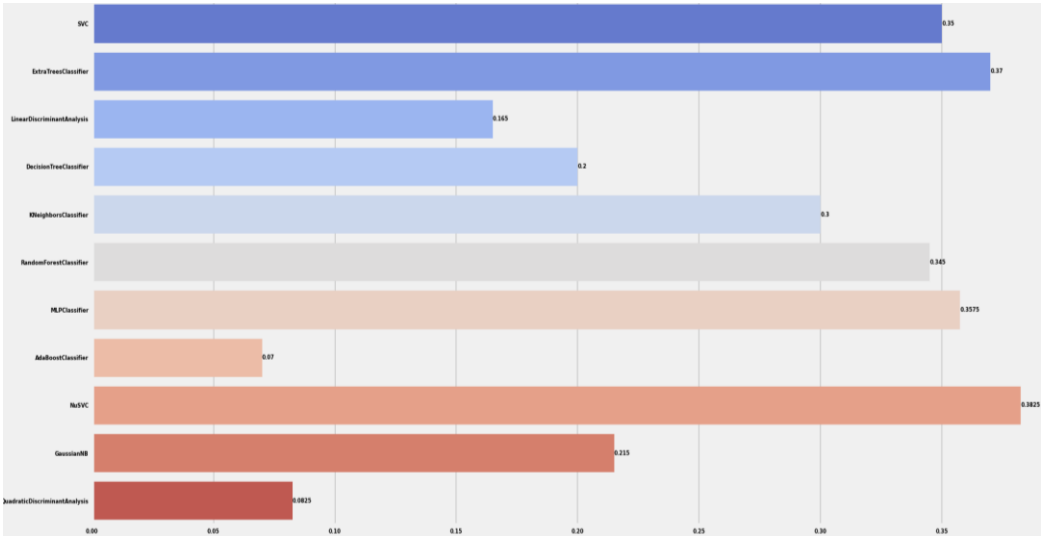
- Mel-Frequency Coefficients (MFCC): this is a set of characteristics that reflect the spectral features of an audio signal and are based on human perception of sound.
- Spectral decay: determines the shape of the spectrum of an audio signal and can provide information about its timbre.
- Colour function: displays the harmonic structure of an audio signal, and is used to classify music genres, recognize chords, and analyze tonality.
- Spectral centroid: determines the central weight of the audio spectrum, used to estimate sound brightness.
- Spectral Bandwidth: Indicates the width of the audio spectrum used to determine the sharpness of the sound.
- Zero-crossing rate: determines the number of crossings of the zero-level amplitude by the audio signal, used to assess the presence of noise or harmonic content [41-42].

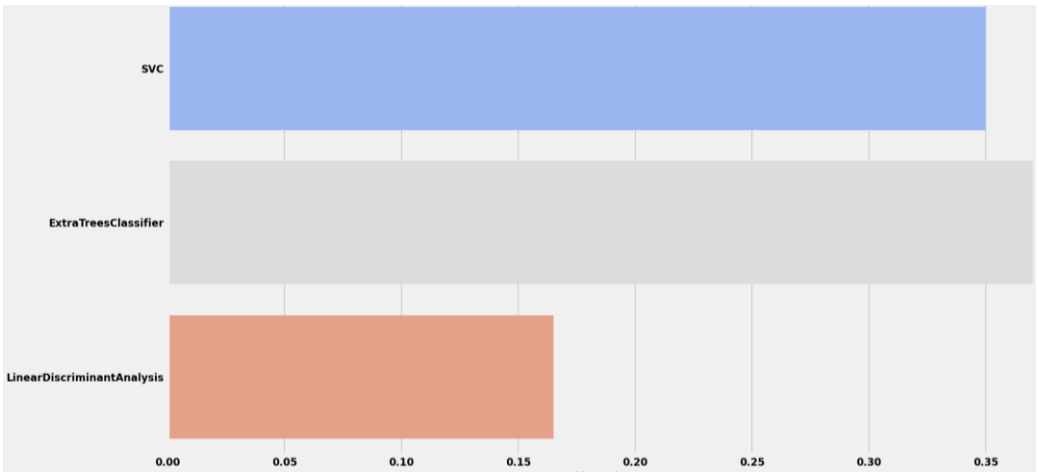
After that, our dataset will look like this:

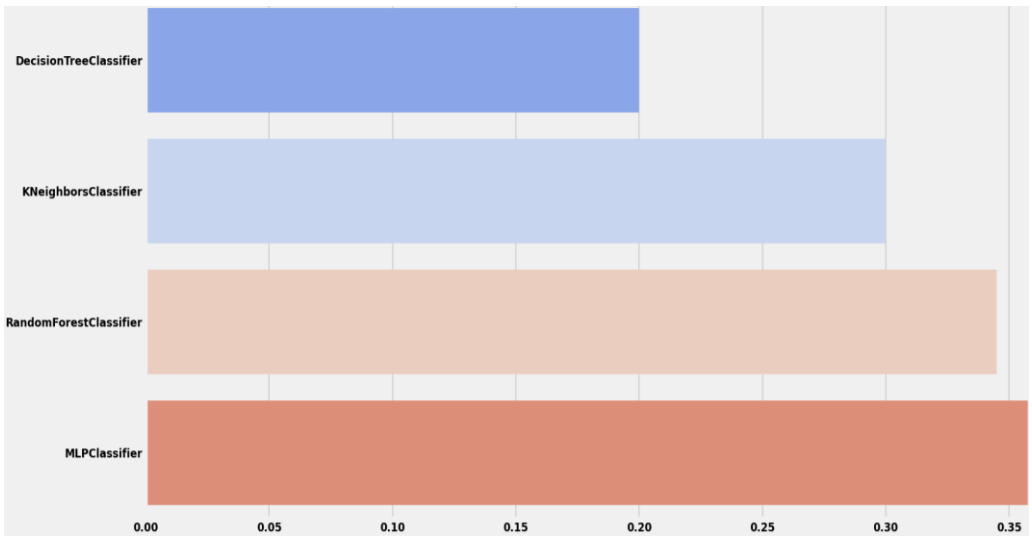
The dataset has grown significantly after removing the features that will be used to train the machine-learning model. To determine the method for the model, we will experiment using different methods and choose the best one.

From Fig. 17 we can see that the best methods for our system are:

- AdaBoost;
- ExtraTrees;
- RandomForest;
- GradientBoosting;
- SVC.







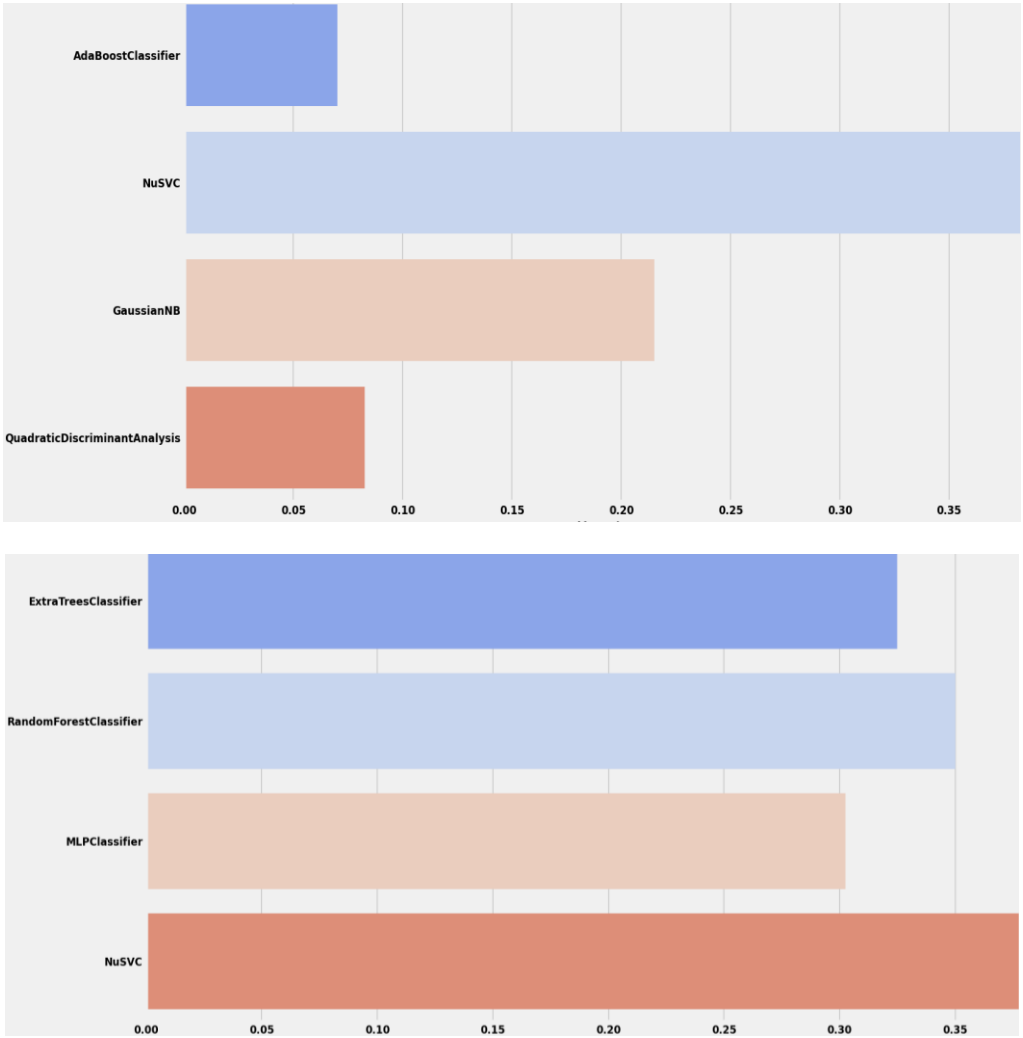


Fig.17. An experiment on choosing a machine learning method

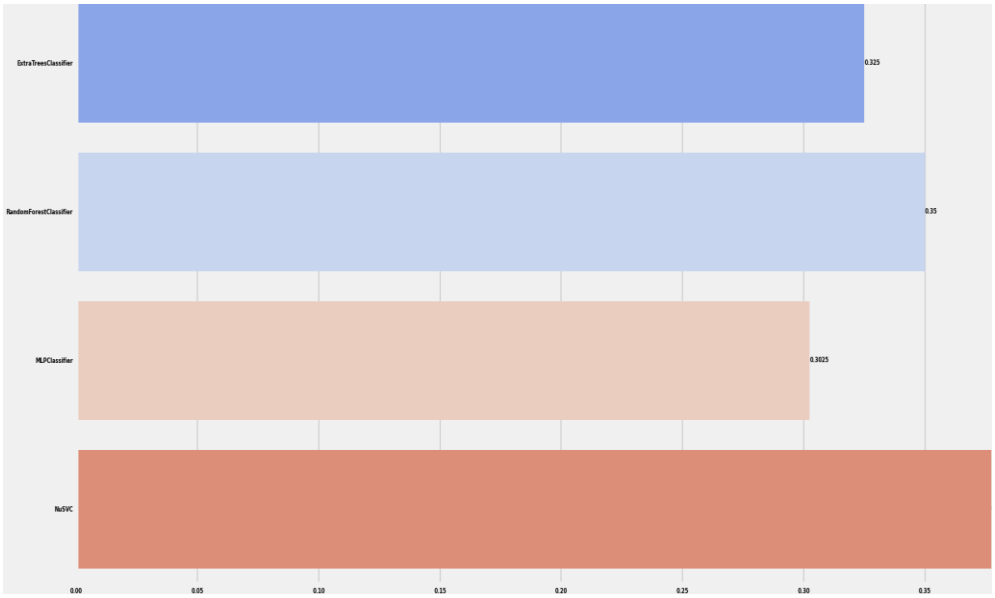


Fig.18. An experiment on choosing a machine learning method after optimization

To improve them, we will optimize the hyperparameters. From Fig. 18, we can see that the best method is SVC, and we will use it in the implementation.

5. Results and Discussion

5.1. Project Testing

To test the model being developed, you first need to save it, so we save the best model found in the previous section using the joblib library for further use.

```
top_models = sorted(models, key=lambda tup: tup[1])
from joblib import dump, load
dump(top_models[-1][2], 'svc.joblib')
```

Fig.19. Saving the SVC model

Next, you need to create a test data set, for this, we will select all data in which fold = 5, as a result, we will get:

	Unnamed: 0	esc10	zero_crossing_rate	chroma_stft	rmse	spectral_centroid	spectral_bandwidth	beat_per_minute	...	mfcc_12	mfcc_13	mfcc_14	mfcc_15	mfcc_16	mfcc_17	mfcc_18	mfcc_19
1600	1600	False	0.089446	0.407441	0.130152	1658.040494	1801.552744	161.499023	...	-0.121251	3.973692	-1.987514	2.229929	-5.938939	-1.003734	0.258683	1.242714
1601	1601	False	0.046093	0.085744	0.109192	889.356191	723.221983	161.499023	...	-0.606728	2.842365	4.207792	5.537512	0.245878	-1.652280	-5.290014	0.438210
1602	1602	False	0.033217	0.219135	0.064306	723.931148	890.696441	161.499023	...	-1.780481	3.785877	-3.674458	0.795371	-0.777418	1.631779	-1.808756	1.549646
1603	1603	False	0.040145	0.057050	0.042946	566.910480	372.962641	161.499023	...	2.345146	1.909373	0.088983	2.677883	-0.710170	-0.229108	-0.148270	2.937163
1604	1604	False	0.129729	0.335196	0.169369	2227.649616	1946.017036	161.499023	...	-2.879095	6.796273	3.612435	7.173410	2.206520	8.037888	-3.256182	0.609923
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1995	1995	False	0.103796	0.380825	0.088098	1732.357906	1593.858265	161.499023	...	-5.620143	-6.867295	-3.785263	4.204754	-3.889776	8.874669	-6.556036	-3.047258
1996	1996	False	0.308209	0.246187	0.113179	4012.107167	2883.123910	161.499023	...	-22.577835	-0.536868	-14.613399	1.727276	-10.408116	-12.115975	-19.507008	-3.788116
1997	1997	False	0.041217	0.414763	0.071444	1374.272249	1783.566072	161.499023	...	-0.716517	2.628664	-1.704746	2.333961	-1.132552	-0.208566	-2.815182	1.033301
1998	1998	False	0.111645	0.442994	0.060926	2283.525840	2222.922528	161.499023	...	1.772502	9.605663	-1.861323	3.854740	-2.246718	5.008191	-11.807304	1.522162
1999	1999	True	0.121324	0.312858	0.024640	2050.319371	1441.394928	161.499023	...	-3.255963	0.299455	-0.829646	3.300714	-0.207643	1.654797	-1.749118	0.618410

[400 rows x 29 columns]

Fig.20. Test dataset

Let's perform data normalization using MinMaxScaler:

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler(feature_range = (0, 1))

dataset_transformed = scaler.fit_transform(X_test)
X_test = pd.DataFrame(dataset_transformed, columns=X_test.columns)
```

Fig.21. Normalization of text data

As a result, we obtain normalized text data.

	Unnamed: 0	esc10	zero_crossing_rate	chroma_stft	rmse	spectral_centroid	spectral_bandwidth	beat_per_minute	...	mfcc_12	mfcc_13	mfcc_14	mfcc_15	mfcc_16	mfcc_17	mfcc_18	mfcc_19
0	0.000000	0.0	0.144098	0.484733	0.248517	0.249598	0.499041	0.0	...	0.622807	0.542725	0.574509	0.417564	0.464026	0.423928	0.561420	0.383662
1	0.002506	0.0	0.071768	0.055562	0.207924	0.121128	0.165578	0.0	...	0.616245	0.522304	0.693687	0.489642	0.613975	0.410012	0.442317	0.364562
2	0.005013	0.0	0.050286	0.233517	0.120993	0.093480	0.217368	0.0	...	0.600378	0.539335	0.542057	0.386303	0.589165	0.480475	0.517106	0.390949
3	0.007519	0.0	0.061845	0.017282	0.079626	0.067237	0.057264	0.0	...	0.656147	0.505463	0.614454	0.427326	0.590796	0.440548	0.552697	0.423891
4	0.010025	0.0	0.211306	0.388352	0.324469	0.344796	0.543715	0.0	...	0.585528	0.593674	0.682234	0.525291	0.661510	0.617925	0.486082	0.368638
..	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
395	0.989975	0.0	0.168039	0.449225	0.167071	0.262018	0.434814	0.0	...	0.548475	0.347040	0.539926	0.460599	0.513707	0.635879	0.415353	0.281811
396	0.992481	0.0	0.509082	0.269606	0.215646	0.643032	0.833506	0.0	...	0.319248	0.461307	0.331627	0.406611	0.355672	0.185504	0.137760	0.264222
397	0.994987	0.0	0.063633	0.494501	0.134818	0.202171	0.493479	0.0	...	0.614761	0.518446	0.579948	0.419831	0.580555	0.440989	0.495535	0.378690
398	0.997494	0.0	0.181134	0.532163	0.114448	0.354135	0.629346	0.0	...	0.648406	0.644385	0.576936	0.452972	0.553543	0.552919	0.302797	0.390296
399	1.000000	1.0	0.197284	0.358551	0.044173	0.315159	0.387666	0.0	...	0.580433	0.476403	0.596782	0.440899	0.602979	0.480969	0.518385	0.368840

[400 rows x 29 columns]

Fig.22. Normalization of text data

As a result, we will get normalized test data:

```
[ 3 43 26 43 26 43 42 37 42 43 16 26 42 42 45 13 42 20 26  9  9  9 13 43
 16 16 16 47  4  4 42  4  9  4  4  9  4  9  9  4  4  9 41 41 26 41 37 37
 15 26 26 32 12 40 40 40 40 13 46  4 13 26 26 26 23 26 43 43 42 13 26 10
  7 37 37 37 22 42 37 13 41 41 22 40  4  4 37 21 37 30 15 10 42 45 45 12
 12  4 40 48 19 12 12 13  1 10  8  3  3  8  0  1  3 37 13 16 10 35 32 48
  3 18  4 36 36 36 10 42 20 20 20 20 20 20 20 45 17 17  8  8 43 43  8  9
  9  1  1  1 41 10 42 46 20 21 18 18  1 33  3 40  1  1 10 17 13 13 26 46
 26 13 13 40 37 34 35 35 17 34 37  3  0 38 24 10 10 13 24 20 33  4 42 13
 38 37 15 37 13 42 37 21  0 17 37 17 41 37 12  9  9  9  9  0 17 37 37  9
 32 13 13 47 19 47 37 12 12 17 37 37 13  9 41 41  0 16 16 16 17 17  9 30
 30  9 37 37 34  0 11 10 11 21 21 15  0 21 34 15  9  9 21 34 23  9 41 44
 32 17 24 37 32  5  7  0 43 34  9 28  1 43 48 15 39 37 13 34 13 30 23 23
  1  1 32 17 37 24 38 15 31 47 31 34 34  4 17 13 34 23 17 37 13 23  9  9
  9 26 26 32 23 17 13 13 14 37 30  4 43 22 17 43 26 17 23 37 15 26 31 32
  7  5  5 43 15 22 30 43 17 37 37 30 30 26 47 22  2 43 34 43 43 22 43 44
 22 35 44 44 26  9 15 31  5 30 37 37 14 26 15 17 15 34 43 39 34 34 42  9
 30 25 23 23  4 43 23 23 17  4  8 26 37 43 26 21]
```

Fig.23. Test results

Let's check for recognition accuracy:

```
accuracy_test = accuracy_score(y_test, y_test_predicited)
```

Fig.24. Checking the results for accuracy

After performing this function, we got a result of 0.3775, which is a small recognition level and most likely cannot be used by the user yet, but it is a good start. To improve the recognition quality, there are two solutions: try another method, for example, a Convolutional Neural Network, or try to find a larger amount of data, which will increase the recognition percentage.

In this work, the Tensorflow library and its Keras division were also used to improve the process of voice recognition and create our own neural network architecture consisting of several different layers. Convolutional layers are designed to detect features (patterns). This layer creates a convolution kernel that convolves the layer's inputs over one spatial (or temporal) dimension to produce a tensor of outputs. The filter parameter specifies the number of nodes in each layer. Each layer will be increased in size from 32 to 128, while the parameter kernel_size = 2. Each convolutional layer has an associated pooling layer of type MaxPooling1D. Max pooling is a type of operation that is usually added to CNNs after individual convolutional layers. When added to a model, max pooling reduces the dimensionality of images, reducing the number of pixels in the results of the previous convolutional layer. The output layer has 50 nodes (classes) and returns the probability distribution by class. The ESC-50 dataset is a labeled collection of 2000 environmental audio recordings suitable for benchmarking environmental sound classification methods. The dataset consists of 5-second recordings organized into 50 semantic classes (with 40 examples per class), divided into 5 main categories. Audio recordings are saved in WAV format, which significantly facilitates their analysis and preparation.

Layer (type)	Output Shape	Param
conv1d_15 (Conv1D)	(None, 25, 32)	96
dropout_12 (Dropout)	(None, 25, 32)	0
conv1d_16 (Conv1D)	(None, 24, 32)	2080
max_pooling1d_9 (MaxPooling 1D)	(None, 12, 32)	0
conv1d_17 (Conv1D)	(None, 11, 64)	4160
dropout_13 (Dropout)	(None, 11, 64)	0
conv1d_18 (Conv1D)	(None, 10, 64)	8256
max_pooling1d_10 (MaxPoolin g1D)	(None, 5, 64)	0
conv1d_19 (Conv1D)	(None, 4, 128)	16512
max_pooling1d_11 (MaxPoolin g1D)	(None, 2, 128)	0
flatten_3 (Flatten)	(None, 256)	0
dropout_14 (Dropout)	(None, 256)	0
dense_6 (Dense)	(None, 512)	131584
dropout_15 (Dropout)	(None, 512)	0
dense_7 (Dense)	(None, 50)	25650

Total params: 188,338
Trainable params: 188,338
Non-trainable params: 0

Animals	Natural soundscapes & water sounds	Human, non-speech sounds	Interior/domestic sounds	Exterior/urban noises
Dog	Rain	Crying baby	Door knock	Helicopter
Rooster	Sea waves	Sneezing	Mouse click	Chainsaw
Pig	Crackling fire	Clapping	Keyboard typing	Siren
Cow	Crickets	Breathing	Door, wood creaks	Car horn
Frog	Chirping birds	Coughing	Can opening	Engine
Cat	Water drops	Footsteps	Washing machine	Train
Hen	Wind	Laughing	Vacuum cleaner	Church bells
Insects (flying)	Pouring water	Brushing teeth	Clock alarm	Airplane
Sheep	Toilet flush	Snoring	Clock tick	Fireworks
Crow	Thunderstorm	Drinking, sipping	Glass breaking	Hand saw

Fig.25. Architecture of a neural network and all classes of the studied data set

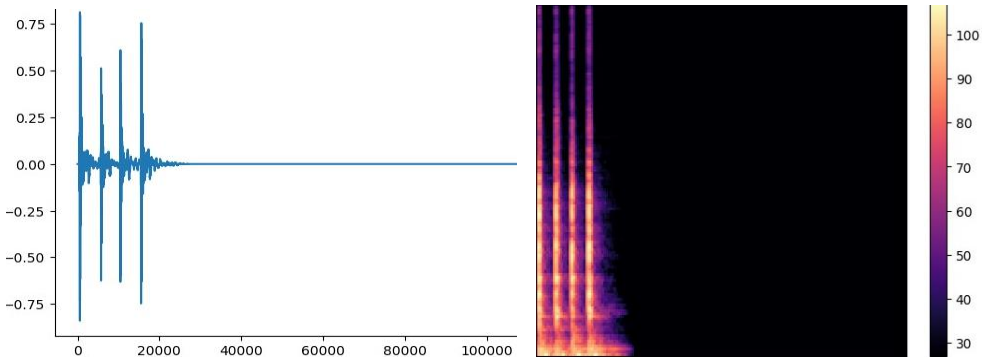
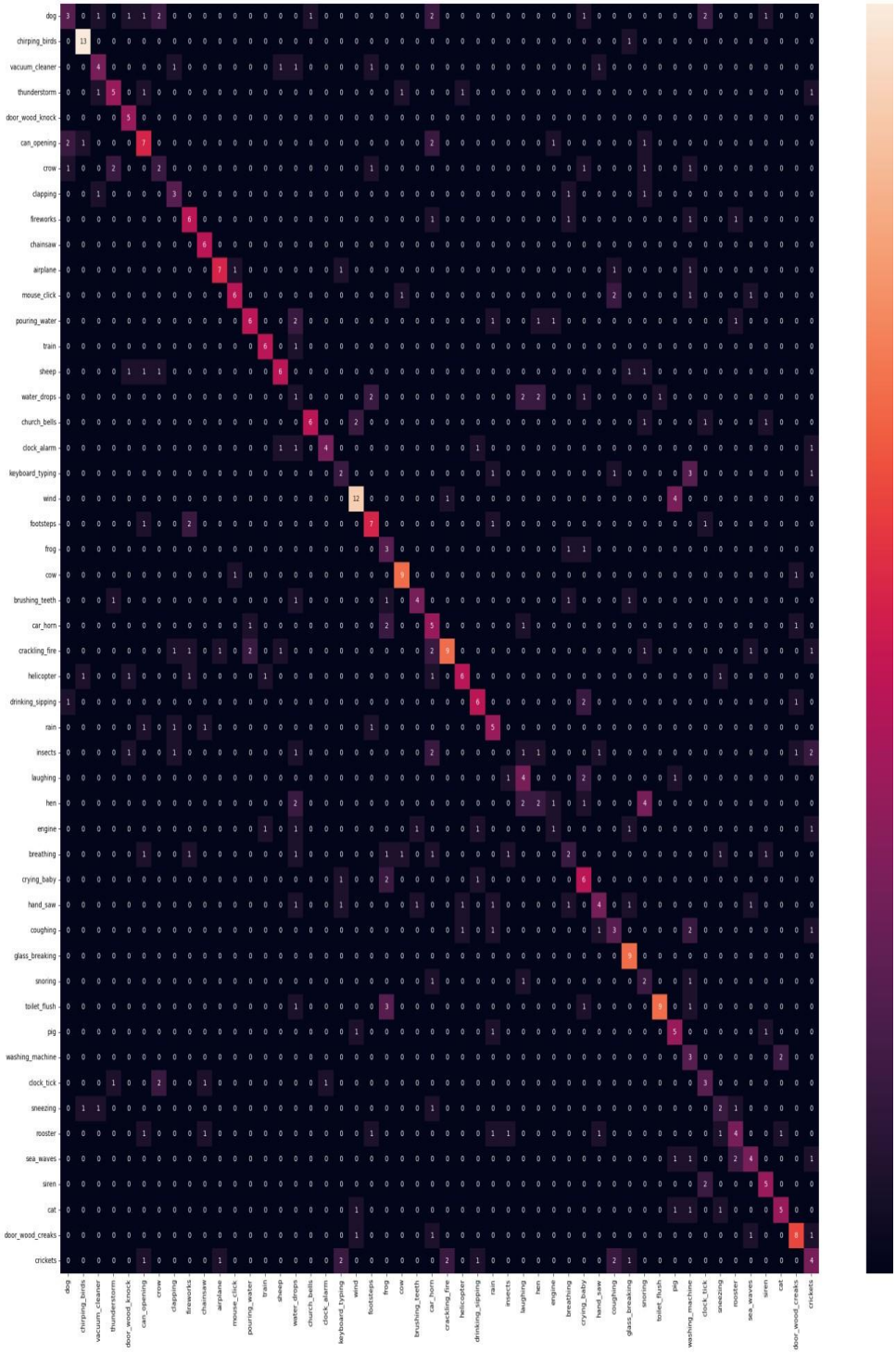


Fig.26. Representation of an audio recording as a wave (door wood knock category) and chalk-spectogram of an audio recording



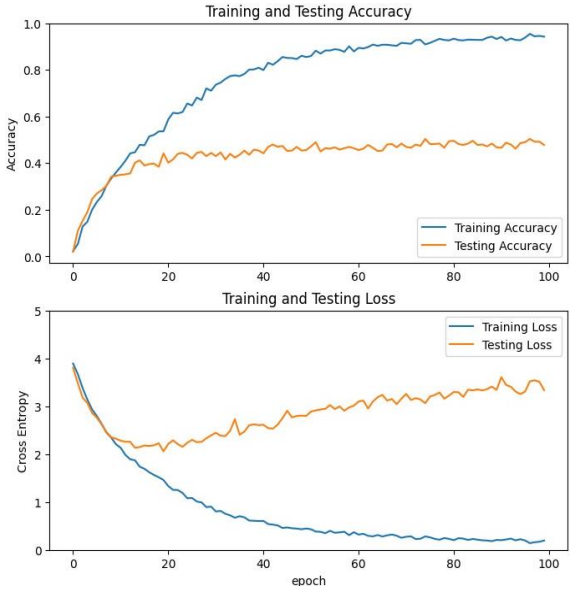


Fig.27. Matrix of model verification results

```
1 Example
1/1 [==============================] - 0s 182ms/step
predict num class: 13
predict class: crickets

filename: 5-210540-A-13.wav
class: crickets
num class: 13
```

Fig.28. Model training schedule and "Cricket" class

Next, all audio recordings from the data set are processed, a wave graph is output, and their chalk spectrograms are created. Mel - a spectrogram is a visual representation of the frequency content of an audio signal over time. It is widely used in speech and audio processing applications, including speech recognition, music analysis, and sound classification. The Mel spectrogram is obtained from the Fourier transform of the audio signal. The Fourier transform transforms a time-domain signal into its frequency-domain representation, providing information about the different frequencies present in the signal. However, the human auditory system does not perceive frequencies linearly. Our pitch perception is more logarithmic, with smaller frequency differences more noticeable at lower frequencies than at higher frequencies. Examples of audio file analysis are shown in Fig. 26.

Next, we extract the features from these audio files and the melspectogram and store the features for each file as a string. These features include the root mean square for the sample, the frequency interval, the center of mass of the sound mfcc. In the future, the model will be trained on these features. As a result of training the model for 100 epochs, a result of 97.7% accuracy was achieved for training data and 47.8% accuracy when checking performance on test data. The graph of the dependence of the accuracy of the model on the training epochs is shown in Fig. 27-28.

```
2 Example
1/1 [==============================] - 0s 19ms/step
predict num class: 34
predict class: can_opening

filename: 1-41615-A-34.wav
class: can_opening
num class: 34
```

```
3 Example
1/1 [==============================] - 0s 19ms/step
predict num class: 16
predict class: wind

filename: 3-117504-A-16.wav
class: wind
num class: 16
```

Fig.29. "Opening a bank" class and "Wind" class

```
4 Example
1/1 [==============================] - 0s 19ms/step
predict num class: 31
predict class: mouse_click

filename: 1-19118-A-24.wav
class: coughing
num class: 24
```

```
5 Example
1/1 [==============================] - 0s 19ms/step
predict num class: 28
predict class: snoring

filename: 3-124795-A-28.wav
class: snoring
num class: 28
```

Fig.30. "Mouse Click" and "class Snoring" class

Let's choose 5 random files to test the program. With the help of a random number generator, we obtained a sample of 5 files of different classes, shown in Fig. 28-30. You can also play each of these audio recordings to see for yourself.

5.2. Deployment of the Project

To develop this project, it is necessary to carry out a huge amount of work on the development of the backend, frontend and part of machine learning. This work gave a good start to the development of a machine learning model for sound recognition and laid the foundation for thinking about what needs to be changed or added that will solve the recognition problem better, maybe it is to increase the amount of data, remove unnecessary functions, and maybe try to analyse a convolutional neural network. To expose this product to the world, you also need to spin up an AWS EC2 instance and deploy the code on it, as well as give access to the user instance, as shown in the deployment diagram. For the backend part, you need to develop an API that can perform basic operations for communicating with the user and configure reading data from the microphone. For the front-end part, you need to make a very simple interface in which the user can download his audio recording, read the data from the microphone and see the recognition result. As an example, you can take the Shazam frontend. The implemented model is only the MVP of the project, so mistakes are always allowed here, after refinement, the project will recognize sounds with a much higher percentage of accuracy.

6. Conclusions

During the execution of the work, the MVP of the project on the analysis and recognition of sounds in the metropolis was developed. This system can help people in various fields to simplify their lives, for example, it can help farmers protect their crops from animals, in the military it can help with the identification of weapons and the search for flying objects, such as drones or missiles, in the future there is a possibility for recognizing the distance to sound, also, in cities can help with security, so a preventive response system can be built, which can check if everything is in order based on sounds. Also, it can make life easier for people with impaired hearing to detect danger in everyday life. In the part of the comparison of analogues of the developed product, 4 analogues were found: Shazam, sound recognition from Apple, Vocapia, and SoundHound. A table of comparisons was made for these analogues and the product under development. Also, after comparing analogues, a table for evaluating the effects of the development was built. During the system analysis section, a variety of audio research materials were developed to indicate the characteristics that can be used for this design: period, amplitude, and frequency, and, as an example, an article on real-world audio applications is shown. A precedent scenario is described using the RUP methodology and UML diagrams are constructed: Diagram of use cases; Class diagram; Activity chart; Sequence diagram; Diagram of components; and Deployment diagram. Also, sound data analysis was performed, sound data was visualized as spectrograms and sound waves, which clearly show that the data are different, so it is possible to classify them using machine learning methods. An experimental selection of the machine learning method for building a sound recognition model was made. The best method turned out to be SVC, the accuracy of which reflects more than 30 per cent. A neural network was also implemented to improve the obtained results. The result of training the model based on the neural network during 100 epochs was a result of 97.7% accuracy for training data and 47.8% accuracy when checking the performance on test data.This result should be higher, so it is necessary to consider improving recognition algorithms, increasing the amount of data, and changing the recognition method. Testing of the project was carried out, showing its operation and pointing out shortcomings that need to be corrected in the future. Also, the next steps of the research will be to apply the model in different real-world settings with testing and analysis of its performance in these settings. For example, a study of real-time recognition results in a noisy urban environment.

Acknowledgement

The research was carried out with the grant support of the National Research Fund of Ukraine "Information system development for automatic detection of misinformation sources and inauthentic behaviour of chat users", project registration number 187/0012 from 1/08/2024 (2023.04/0012). Also, we would like to thank the reviewers for their precise and concise recommendations that improved the presentation of the results obtained.

References

[1] Destroyer birds: how to protect your farm from birds? 2015. Kurkul. URL: https://kurkul.com/blog/80-ptahi-nischivniki-yak-zahistiti-svoye-gospodarstvo-vid-pernatih
[2] HESA Shahed-136. 2022. Militaryfactory. URL: https://www.militaryfactory.com/aircraft/detail.php?aircraft_id=2520
[3] T. Basyuk, A. Vasyliuk, Peculiarities of matching the text and sound components in the Ukrainian language system development, CEUR Workshop Proceedings 3723 (2024) 466-483.
[4] T. Kovaliuk, I. Yurchuk, O. Gurnik, Topological structure of Ukrainian tongue twisters based on speech sound analysis, CEUR Workshop Proceedings 3723 (2024) 328-339.
[5] Peleshchak, R.M., Kuzyk, O.V., Dan'kiv, O.O.: The influence of ultrasound on formation of self-organized uniform nanoclusters. In: Journal of Nano- and Electronic Physics 8(2),02014. (2016)

[6] Peleshchak, R., Kuzyk, O., Dan'Kiv, O.: The criteria of formation of InAs quantum dots in the presence of ultrasound. In: International Conference on Nanomaterials: Applications and Properties, NAP, 01NNPT06. (2017)
[7] Peleshchak, R.M., Kuzyk, O.V., Dan'kiv, O.O.: The influence of ultrasound on the energy spectrum of electron and hole in InAs/GaAs heterosystem with InAs quantum dots. In: Journal of Nano- and Electronic Physics 8(4),04064. (2016)
[8] Altexsoft. 2022. Audio Analysis with Machine Learning: Building AI-Fueled Sound Detection App. URL: https://www.altexsoft.com/blog/audio-analysis/
[9] V. Motyka, Y. Stepaniak, M. Nasalska, V. Vysotska, People's Emotions Analysis while Watching YouTube Videos, CEUR Workshop Proceedings, 3403, 2023, pp. 500–525.
[10] O. Turuta, I. Afanasieva, N. Golian, V. Golian, K. Onyshchenko, Daniil Suvorov, Audio processing methods for speech emotion recognition using machine learning, CEUR Workshop Proceedings 3711 (2024) 75-108.
[11] Audio Deep Learning Made Simple: Sound Classification, Step-by-Step. 2021. Towardsdatascience. URL: https://towardsdatascience.com/audio-deep-learning-made-simple-sound-classification-step-by-step-cebc936bbe5
[12] Sartiukova, O. Markiv, V. Vysotska, I. Shakleina, N. Sokulska, I. Romanets, Remote Voice Control of Computer Based on Convolutional Neural Network, in: Proceedings of the IEEE 12th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Dortmund, Germany, 07-09 September 2023, pp. 1058 – 1064.
[13] Koshovyy, V., Ivantyshyn, O., Mezentsev, V., Rusyn, B., Kalinichenko, M., Influence of active cosmic factors on the dynamics of natural infrasound in the earth's atmosphere. In: Romanian Journal of Physics, 2020, 65(9-10), pp. 1–10, 813
[14] Soundproofcow. What Are the Characteristics of a Sound Wave?.URL: https://www.soundproofcow.com/characteristics-of-sound-wave/
[15] Vladyslav Tsap, Nataliya Shakhovska, Ivan Sokolovskyi, The Developing of the System for Automatic Audio to Text Conversion, in: CEUR Workshop Proceedings, Vol-2917, 2021, pp. 75-84.
[16] Basystiuk, O., Shakhovska, N., Bilynska, V., ...Shamuratov, O., Kuchkovskiy, V., The developing of the system for automatic audio to text conversion. In: CEUR Workshop Proceedings, 2021, 2824, pp. 1–8
[17] L. Kobylyukh, Z. Rybchak, O. Basystiuk, Analyzing the Accuracy of Speech-to-Text APIs in Transcribing the Ukrainian Language, CEUR Workshop Proceedings, Vol-3396, 2023, 217-227.
[18] K. Tymoshenko, V. Vysotska, O. Kovtun, R. Holoshchuk, S. Holoshchuk, Real-time Ukrainian text recognition and voicing, CEUR Workshop Proceedings, Vol-2870, 2021, pp. 357-387.
[19] Trysnyuk, V., Nagornyi, Y., Smetanin, K., Humeniuk, I., & Uvarova, T. (2020). A method for user authenticating to critical infrastructure objects based on voice message identification. Advanced Information Systems, 4(3), 11–16. https://doi.org/10.20998/2522-9052.2020.3.02
[20] Bisikalo, O., Boivan, O., Khairova, N., Kovtun, O., Kovtun, V., Precision automated phonetic analysis of speech signals for information technology of text-dependent authentication of a person by voice. In: CEUR Workshop Proceedings, 2021, 2853, pp. 276–288
[21] ScienceDirect. 2021. Sound-spectrogram based automatic bird species recognition using MLP classifier. URL: https://www.sciencedirect.com/science/article/abs/pii/S0003682X21001705
[22] Apple. 2022. Shazam turns 20. URL: https://www.apple.com/sa/newsroom/2022/08/shazam-turns-20/
[23] Wayback Machine. 2009. Shazam names that tune. URL: https://web.archive.org/web/20120807220614/http://www.director.co.uk/magazine/2009/11%20December/shazam_63_04.html
[24] Indianexpress. 2021. What is Sound Recognition in iOS 14 and how does it work? URL: https://indianexpress.com/article/technology/mobile-tabs/what-is-sound-recognition-in-ios-14-and-how-does-to-work-7311903/
[25] TopAI.tools. Vocapia. URL: https://topai.tools/t/vocapia
[26] Vocapia. 2023. Speech to Text Software. URL: https://www.vocapia.com/
[27] SoundHoundAI. Voice AI platform. URL: https://www.soundhound.com/voice-ai-products/platform/
[28] Tutorialspoint. (2018). System Analysis and Design – Overview. URL: https://www.tutorialspoint.com/system_analysis_and_design/system_analysis_and_design_overview.htm
[29] Business Analysts Handbook. User Requirements. URL: https://businessanalyst.fandom.com/wiki/User_Requirements
[30] MEDIUM. 2023. Understand Linear Support Vector Classifier (SVC) In Machine Learning a Classification Algorithm. URL: https://blog.tdg.international/understand-linear-support-vector-classifier-svc-in-maschine-learning-a-classification-algorithm-3deb385f6e7d
[31] MEDIUM. 2022. What? When? How?: ExtraTrees Classifier. URL: https://towardsdatascience.com/what-when-how-extratrees-classifier-c939f905851c
[32] AlmaBetter. AdaBoost Algorithm in Machine Learning. URL: https://www.almabetter.com/bytes/tutorials/data-science/adaboost-algorithm
[33] MEDIUM. 2021. A Multi-layer Perceptron Classifier in Python; Predict Digits from Gray-Scale Images of Hand-Drawn Digits from 0 Through 9. URL: https://medium.com/@polanitzer/a-multi-layer-perceptron-classifier-in-python-predict-digits-from-gray-scale-images-of-hand-drawn-44936176be33
[34] OpenSource. What is Python?. URL: https://opensource.com/resources/python
[35] scikit-learn. Machine Learning in Python. URL: https://scikit-learn.org/stable/
[36] Librosa. Librosa. URL: https://librosa.org/doc/latest/index.html
[37] Matplotlib. Matplotlib: Visualization with Python. URL: https://matplotlib.org/
[38] Seaborn. seaborn: statistical data visualization. URL: https://seaborn.pydata.org/
[39] NumPy. NumPy documentation. URL: https://numpy.org/
[40] Pandas. pandas documentation . URL: https://pandas.pydata.org/
[41] Developedia. 2021. Audio Feature Extraction. URL: https://devopedia.org/audio-feature-extraction
[42] Krishna Kumar, "Audio classification using ML methods" y M.Tech Artificial Intelligence REVA Academy for Corporate Excellence - RACE, REVA University Bengaluru, India
[43] Find any sound you like. https://freesound.org/

[44] ESC-50: Dataset for Environmental Sound Classification. https://github.com/karolpiczak/ESC-50?tab=readme-ov-file
[45] ESC-50 audio classification. https://github.com/shibuiwilliam/audio_classification_keras/blob/master/esc50_classification.ipynb

Authors' Profiles

Lyubomyr Chyrun is an Associate Professor at the Ivan Franko National University of Lviv. Since 2001, he worked at the Lviv Polytechnic University at the Institute of Computer Sciences and Information Technologies. in the position of associate professor of the Department of Information Systems and Networks. In 2007 defended his PhD thesis on May 1, 2002 - "Mathematical modelling and computational methods". He is the co-author of the monograph "Continuous Fractions and Complex Numbers". Author of more than 120 scientific publications. His areas of scientific interest are pattern recognition, application of numerical methods in information technologies, object-oriented programming, web technologies, fake identification, natural language processing, computer linguistics, data science, system analysis, information technologies, and machine learning.

Victoria Vysotska is a Professor at the Information Systems and Networks Department of Lviv Polytechnic National University. She defended her Doctoral degree in Technical Science, speciality in «structural, applied and mathematical linguistics» in 2023 on the topic "Analysis and synthesis of computational linguistic systems for processing Ukrainian textual content" Also, she received a PhD degree in Information Technologies from Lviv Polytechnic National University in 2014. She has currently published more than 500 publications. Her main research interests are focused on the identification of disinformation/fakes/propaganda, detection of sources of disinformation and inauthentic behaviour (bots) of coordinated groups based on artificial intelligence, NLP, computer linguistics, data science, system analysis, information technologies, machine learning, cyber security.

Stepan Tchynetskyi is a PhD student at the Lviv Polytechnic National University. He has currently published 3 publications. His research interests are Computer Science and Technology Applications, Machine Learning, and Big Data.

Yuriy Ushenko is a Professor at the Computer Science Department, Chernivtsi National University, Chernivtsi, Ukraine. Research Interests: Data Mining and Analysis, Computer Vision and Pattern Recognition, Optics & Photonics, Biophysics. His research interests include information systems design, data mining, pattern recognition and digital image processing, artificial neural networks, laser polarimetry and interferometry.

Dmytro Uhryn graduated from Yuriy Fedkovych Chernivtsi National University, Chernivtsi. Currently, he is a Doctor of Technical Sciences and associate professor at Yuriy Fedkovych Chernivtsi National University. His research interests are data mining, information technologies for decision support, swarm intelligence systems, and industry-specific geographic information systems.

How to cite this paper: Lyubomyr Chyrun, Victoria Vysotska, Stepan Tchynetskyi, Yuriy Ushenko, Dmytro Uhryn, "Information Technology for Sound Analysis and Recognition in the Metropolis based on Machine Learning Methods", International Journal of Intelligent Systems and Applications(IJISA), Vol.16, No.6, pp.40-72, 2024. DOI:10.5815/ijisa.2024.06.03