DATA COMPRESSION FOR AUDIO-BASED SMART BEEKEEPING

Guangyu Shi

Submitted in fulfillment of the requirements for the degree of
Master of Applied Technology

Institute of Unitec
[2024]

Primary Supervisor: Dr. Lei Song

Secondary Supervisor: Dr. Iman Ardekani

Abstract

The research aimed to address the challenges associated with audio data compression in beehive monitoring by exploring the feasibility and effectiveness of using the Free Lossless Audio Codec compression format. The contribution is demonstrated by the efficacy of FLAC compression in reducing resource consumption without feature loss, thereby not compromising AI performance. The methodology involved using FLAC techniques for audio compression, extracting relevant acoustic features using Mel-Frequency Cepstral Coefficients, and implementing Support Vector Machine models to classify and analyse hive conditions. The results demonstrated that Free Lossless Audio Codec outperformed MPEG-1 Audio Layer 3 and uncompressed Waveform Audio File formats in maintaining the efficiency of audio signals and the integrity of critical acoustic features. These metrics include waveform characteristics, classifier accuracy, compression degree and speed, and transmission speed through the inclusion of multiple data sources. The findings highlight Free Lossless Audio Codec as an advantageous option for beehive monitoring systems. Despite the positive findings, the research has some limitations. The datasets used in the experiments may not encompass all possible beehive conditions and environmental variations. Additionally, the SVM models were implemented with specific parameters that may not generalize to all contexts. Further research is needed to assess the robustness of the findings across a broader range of conditions and to explore different machine learning approaches.

Acknowledgments

I would like to extend my deepest gratitude to all those who have supported and inspired me throughout my doctoral journey.

First and foremost, I am immensely grateful to my primary supervisor, Dr. Lei Song, and my secondary supervisor, Dr. Iman Ardekani. Their invaluable guidance, insightful feedback, and unwavering support have been instrumental in the successful completion of my research. Dr. Lei's dedication to his students and his availability for discussion and advice have made the research process both smooth and rewarding. Dr. Iman’s expertise and constructive critiques have significantly enriched my work.

I would also like to thank my classmates in the lab for their friendship and support. Special thanks to Kartikeye Dangwal for his encouragement and assistance during the course of my thesis. The interactions and collaborations with An Le have been both intellectually stimulating and enjoyable. The supportive atmosphere in our lab has greatly contributed to my academic and personal growth.

I am profoundly thankful to my family for their unwavering love and support. My wife, Xiaoyuan Zhang, has been a pillar of strength, working tirelessly to provide the best possible environment for me to pursue my studies. Her sacrifices and dedication are deeply appreciated. My mother, Liuyang Chun, has been a constant source of encouragement and love, always there to support me through challenges and to celebrate my successes. Her resilience and care have been vital to my journey.

Lastly, I would like to express my gratitude to the Institute of Unitec for providing an excellent environment and the necessary resources to facilitate my research. The support and infrastructure offered by Unitec have been crucial to the successful completion of my studies.

Table of Contents

Abstract ii

Acknowledgements iii

List of Figures vi

List of Tables vii

List of Abbreviations viii

Chapter 1. Introduction 9

1.1. Background 9

1.2. Context 10

1.3. Purposes 11

1.4. Significance, SCOPE, and Definitions 12

1.5. Thesis Outline 15

Chapter 2. Literature Review 17

2.1. Audio Compression Techniques 17

2.2. Audio Compression in Beehive Monitoring 31

2.3. Practical Applications of Beehive Monitor 33

2.4. Challenges in Beehive Audio Data 36

2.5. Research Gap: Data Compression Without Feature Loss 38

2.6. Hypotheses and Research Questions 39

2.7. Summary and Implications 40

Chapter 3. Research Design 41

3.1. Independent Variable 41

3.2. Dependent Variables 41

3.3. Hypotheses 42

3.4. Instruments 43

3.5. Participants 45

3.6. Procedure and Timeline 46

3.7. Ethics and Limitations 47

Chapter 4. Methodology 48

4.1. Free Lossless Audio Codec 48

4.2. Mel Frequency Cepstral Coefficients 52

4.3. Support Vector Machine 55

Chapter 5. Experiment 58

5.1. Audio Data 59

5.2. Compression 60

5.3. MQTT Transmission 61

5.4. Feature extraction 62

5.5. Support Vector Machine Classifier 66

Chapter 6. Results 69

6.1. NU Dataset Result 69

6.2. OSBH Dataset 73

6.3. Stanford BeeAudio Dataset 76

6.4. Compression performance 79

6.5. Transmission Performance 80

Chapter 7. Discussion 83

7.1. Impact of Waveform on Audio Compression 83

7.2. Compression Efficiency: Speed and Degree 84

7.3. Distortion Characteristics 88

7.4. Robustness 90

7.5. Findings 91

Chapter 8. Conclusions and Future Works 93

8.1. Conclusions 93

8.2. Future works 98

References 100

List of Figures

FIGURE 1. STRUCTURE OF FLAC ENCODER. 50
FIGURE 2. WAV WAVEFORM IS DRAWN. 51
FIGURE 3. MP3 WAVEFORM IS DRAWN 51
FIGURE 4. FLAC WAVEFORM IS DRAWN 51
FIGURE 5. STUDY OF MFCC FEATURE EXTRACTION METHODS 52
FIGURE 6. SUPPORT VECTOR MACHINE THEORY 56
FIGURE 7. EXPERIMENT STEPS INCLUDE THREE PHASES 58
FIGURE 8. THE WAV SPECTROGRAM 63
FIGURE 9. THE MP3 SPECTROGRAM 63
FIGURE 10. THE MP3 SPECTROGRAM 64
FIGURE 11. THE FLAC SPECTROGRAM. 64
FIGURE 12. WAV ROC CURVE FOR NU 71
FIGURE 13. MP3 64K ROC CURVE FOR NU 72
FIGURE 14. MP3 16K ROC CURVE FOR NU 72
FIGURE 15. FLAC ROC CURVE FOR NU 72
FIGURE 16. WAV ROC CURVE FOR OSBH 75
FIGURE 17. MP3 64K ROC CURVE FOR OSBH 75
FIGURE 18. MP3 16K ROC CURVE FOR OSBH 75
FIGURE 19. FLAC ROC CURVE FOR OSBH. 76
FIGURE 20. WAV ROC CURVE FOR BEEAUDIO 78
FIGURE 21. MP3 64K ROC CURVE FOR BEEAUDIO 78
FIGURE 22. MP3 16K ROC CURVE FOR BEEAUDIO 79
FIGURE 23. FLAC ROC CURVE FOR BEEAUDIO 79
FIGURE 24. MP3 COMPRESSION TIME. 80
FIGURE 25. FLAC COMPRESSION TIME. 80
FIGURE 26. MQTT TRANSMISSION TIME FOR WAV 80
FIGURE 27. MQTT TRANSMISSION TIME FOR MP3 81
FIGURE 28. MQTT TRANSMISSION TIME FOR FLAC 81

List of Tables

TABLE 1. TRAINING CATEGORY.. 59
TABLE 2. NU DATA SET SEGMENTS. 69
TABLE 3. NU CLASSIFICATION ACCURACY AND FILE SIZE. 70
TABLE 4. UN CONFUSION MATRIX. ANALYSE 70
TABLE 5. CLASSIFICATION REPORT OF NO QUEEN 70
TABLE 6. QUEEN CLASSIFICATION REPORT 71
TABLE 7. OSBH DATA SET SEGMENTS 73
TABLE 8. OSBH CLASSIFICATION ACCURACY AND FILE SIZE 74
TABLE 9. OSBH CONFUSION MATRIX. 74
TABLE 10. CLASSIFICATION REPORT FOR NOQUEEN 74
TABLE 11. CLASSIFICATION REPORT FOR QUEEN 74
TABLE 12. DISTRIBUTION OF DATA SEGMENTS 76
TABLE 13. COMPARISON OF CLASSIFICATION. 77
TABLE 14. CONFUSION MATRIX DETAILS 77
TABLE 15. CLASSIFICATION REPORT IN THE NO-QUEEN 77
TABLE 16. CLASSIFICATION REPORT IN THE QUEEN 77
TABLE 17. COMPRESSION TIME AND TRANSMISSION TIME. 81

List of Abbreviations

1. WAV - Waveform Audio File Format
2. MP3 - MPEG Audio Layer III
3. FLAC - Free Lossless Audio Codec
4. MFCC - Mel-Frequency Cepstral Coefficients
5. FFT - Fast Fourier Transform
6. GMM - Gaussian Mixture Model
7. HMM - Hidden Markov Model
8. SVM - Support Vector Machine
9. ROC - Receiver Operating Characteristic
10. OSBH - Open Source Beehive
11. MQTT - Message Queuing Telemetry Transport

Chapter 1. Introduction

This chapter outlines the background (section 1.1), stating the overall status of beekeeping and New Zealand beekeeping, and the context of the research (section 1.2), highlighting the focus and problem situation. It then describes the purposes of the study (section 1.3), emphasizing the specific aims and objectives. Section 1.4 discusses the significance and scope of the research and provides definitions of terms used. Finally, section 1.5 includes an outline of the remaining chapters of the thesis, setting the stage for the detailed exploration of the research.

1.1. BACKGROUND

1.1.1 Overview of Beekeeping Practices Worldwide

Beekeeping, an ancient practice dating back millennia, plays an essential role in the sustainability of agriculture and the preservation of ecological balance. As pollinators, bees are pivotal in the propagation of numerous plant species, many of which constitute the primary food sources for humans and other animals (V. Patel, Apr. 2020). The global food supply is deeply intertwined with the health of bee colonies, as these diligent insects are responsible for pollinating about one-third of the food crops we consume. The significance of beekeeping extends beyond agricultural yield to encompass biodiversity conservation. Bees contribute to the reproduction of wild plants, fostering diverse habitats and ecosystems. This biodiversity in turn supports a wide range of fauna, underpinning the resilience of natural systems against environmental changes and disturbances (Unep, May 20, 2019).

In recent years, the beekeeping industry has faced challenges due to environmental threats such as habitat loss, climate change, pesticides, and diseases. These challenges have led to a decline in bee populations worldwide, a phenomenon that has dire implications for food security and ecosystems (Zacepins, 2015). Consequently, there is a heightened need for innovative approaches to beekeeping that can support colony health and ensure the continuation of their pollination services.

Given the critical role bees play in our ecosystems and food supply, advancements in beekeeping practices are not only beneficial but necessary. The integration of technology, such as audio monitoring and data analysis, offers promising avenues to enhance the effectiveness of beekeeping. By developing and implementing smart beekeeping tools, we can better understand and protect these vital pollinators, securing the future of our agricultural practices and natural biodiversity (Boys, R., 2021).

1.1.2 Beekeeping in New Zealand

In New Zealand, beekeeping is a significant industry with 3,508 registered beekeepers managing 12,770 apiaries, comprising a total of 194,213 hives. The distribution of these hives is uneven across the country, with 57% located in the North Island and 43% in the South Island (C. Schmidt, Jan. 2021). The South Island's hives are primarily concentrated to the east of the Southern Alps, on the flat coastal plains and downlands where sheep farming is prevalent. The North Island, on the other hand, has a more uniform hive distribution, with the highest concentration around Hamilton, a major dairy farming region (Cook, 2015).

New Zealand's honey production, which includes various types like clover, mānuka, and honeydew, is significant on a global scale. In 2018, the country produced 20,000 tons of honey, with a substantial portion being exported. Mānuka honey, in particular, is renowned for its exceptional antibacterial properties, making it highly valuable and sought after internationally (Donovan, Nov. 08, 2023).

Traditional beekeeping in New Zealand, like elsewhere, relies on manual inspections and empirical knowledge passed down through generations. However, the limitations of this approach in terms of precision and data collection have become apparent, especially in the face of modern challenges. Smart beekeeping, which integrates modern technology such as sensors and data analytics, offers a way to overcome these limitations. Technologies like temperature, humidity, weight, and sound sensors provide real-time data on hive conditions, enabling more informed and timely decision-making (Blanc, Simone, Mar. 2018).

1.2. CONTEXT

The beekeeping industry is increasingly integrating advanced technologies such as the Internet of Things (IoT) and edge computing to enhance the management and monitoring of bee colonies (Jukan, Masip-Bruin, & Amla, 2017). Traditional beekeeping methods, which rely heavily on manual inspections and experience, are being supplemented and, in some cases, replaced by sophisticated digital systems (Blanc, Mar. 2018). IoT devices, including sensors and cameras, are now commonly used to collect detailed data on various parameters like hive temperature, humidity, weight, and bee activity. This data is crucial for maintaining the health and productivity of bee colonies.

Edge computing plays a pivotal role in this technological shift by processing data locally at or near the source, which reduces latency and bandwidth usage (Wolfert, Ge, Verdouw, & Bogaardt, 2017). By analyzing data on-site, beekeepers can receive immediate insights and alerts, enabling them to respond swiftly to potential issues such as disease outbreaks, pest infestations, or environmental changes. This rapid, local processing is particularly important in remote apiaries where internet connectivity may be limited or unreliable (Berckmans, October 2004).

To facilitate the effective transmission of data from the field to centralized data centers for further analysis, robust communication protocols are essential. These protocols ensure that large volumes of data can be transmitted quickly and reliably, providing a continuous flow of information that supports both immediate and long-term decision-making processes (Gil-Lebrero, 2016).

One of the most significant factors affecting transmission speed is the size of the data files. Larger files take longer to transmit, especially over networks with limited bandwidth. Therefore, optimizing file size is key to enhancing transmission speed (Schertz Willett, 1995). This can be achieved through data compression techniques, efficient data encoding, and selective transmission of only the most relevant data. By focusing on reducing file size, the beekeeping industry can improve the efficiency of data transmission, ensuring that critical information reaches data centers quickly. This, in turn, supports faster data analysis and more timely responses to emerging issues within bee colonies, ultimately leading to better hive management and increased productivity.

1.3. PURPOSES

The primary purpose of this study is to explore the feasibility and effectiveness of using the Free Lossless Audio Codec (FLAC) in beehive monitoring systems. The specific aims are to evaluate the impact of FLAC compression on the accuracy of machine learning models used in the analysis of beehive sounds, compare the performance of FLAC with both uncompressed formats (e.g., WAV) and lossy formats (e.g., MP3) in terms of file size, audio quality, and computational efficiency, and develop recommendations for the implementation of FLAC in practical beehive monitoring systems. The study seeks to answer the following research questions: How does FLAC compression affect the fidelity of audio data critical for beehive monitoring? What are the trade-offs between file size reduction and audio quality when using FLAC compared to other audio formats? Can FLAC provide a practical balance of data quality and resource consumption for real-time beehive monitoring applications? By addressing these questions, the research aims to contribute to the development of more efficient and effective beehive monitoring technologies, thereby supporting better management practices and enhancing the sustainability of beekeeping operations.

1.4. SIGNIFICANCE, SCOPE, AND DEFINITIONS

1.4.1 Significance

1.4.1.1 Significance of Research

The significance of this research lies in its potential to resolve two critical issues in beehive monitoring: transmission efficiency and early detection accuracy. By exploring the feasibility and effectiveness of using the Free Lossless Audio Codec (FLAC) in beehive monitoring systems, this study aims to enhance the transmission of audio data and improve the accuracy of detecting early signs of hive distress. The research involves evaluating FLAC's impact on machine learning model accuracy, comparing it with uncompressed and lossy formats, and developing practical recommendations for its implementation (I. Z. Ochoa, Oct. 01, 2019). Addressing these challenges can lead to more efficient data handling and more precise monitoring of hive health, ultimately supporting better beekeeping management practices and contributing to the sustainability of the industry, thereby benefiting the beekeeping industry.

1.4.1.2 Significance of Literature Review

The literature review encompasses smart beekeeping practices, the current use of audio monitoring in beekeeping, and the various data compression techniques utilized in similar fields. The review reveals that traditional beekeeping relies heavily on manual inspections and empirical knowledge. However, these methods are often insufficient for early detection of issues within the hive due to the lack of detailed data. Smart beekeeping practices, which incorporate technologies such as sensors and data analytics, provide a modern approach to managing hives. These technologies can monitor various parameters such as temperature, humidity, weight, and sound within the hive, offering real-time data that supports informed decision-making (A. L. Imoize, 3, Aug. 2020).

Current audio monitoring techniques in beekeeping utilize both uncompressed and lossy compressed audio formats. Uncompressed formats like WAV provide high-quality audio but result in large file sizes, which are impractical for long-term monitoring and storage. Lossy formats like MP3, while reducing file sizes significantly, can introduce distortions that impact the accuracy of audio analysis and machine learning models. The gap in the literature lies in the application of lossless compression formats, such as FLAC, in beekeeping. FLAC offers a promising balance between file size reduction and audio quality preservation, which is crucial for accurate analysis (Israelsson, Jan. 01, 1996). No substantial research currently exists on the impact of FLAC compression on the performance of machine learning models in beehive audio monitoring. This study aims to fill this gap by evaluating FLAC's feasibility and effectiveness in maintaining audio fidelity while providing manageable file sizes.

1.4.1.3 Significance of Methodology

The methodology used in this study highlights several key features of FLAC that make it particularly valuable for beehive monitoring. Firstly, FLAC's compression speed is very fast, even faster than MP3, allowing for quick data processing and transmission. Additionally, while FLAC files are smaller than uncompressed WAV files, they do not suffer from any distortion, unlike lossy formats such as MP3. This ensures that the audio quality remains intact, which is crucial for accurately detecting subtle changes in beehive health. Furthermore, FLAC can coordinate effectively with Mel-Frequency Cepstral Coefficients (MFCC) extraction and Support Vector Machines (SVM) for detailed analysis. Importantly, FLAC ensures that MFCC features are never lost, thereby maintaining the accuracy of the SVM models used for detecting beehive health (D. T. Várkonyi, Mar. 2023). By leveraging these advantages, the methodology demonstrates that FLAC is a practical and effective tool for monitoring bee colonies, enhancing both the efficiency and accuracy of health assessments in beekeeping.

1.4.1.4 Significance of Experiment

The experiment conducted in this study holds significant importance in fulfilling the research aim of evaluating the feasibility and effectiveness of employing the Free Lossless Audio Codec (FLAC) in beehive monitoring systems. By systematically comparing WAV, MP3, and FLAC compression formats, the experiment addresses critical aspects related to transmission efficiency and accuracy in detecting beehive health. The collection and processing of diverse datasets ensure the inclusivity of various scenarios, enhancing the experiment's robustness and applicability to real-world settings. Analysing the compression speed of WAV to MP3 and FLAC sheds light on the computational efficiency of each format, crucial for assessing their practical implementation in real-time monitoring systems. Simulating transmission via MQTT enables the evaluation of compression formats' performance in communication channels, providing insights into data transfer efficiency—a vital consideration for timely monitoring in beekeeping operations. Furthermore, the extraction of Mel-Frequency

Cepstral Coefficients (MFCC) ensures consistent feature representation across all compressed formats, facilitating accurate analysis and comparison. By applying Support Vector Machines (SVM) for classification tasks, the experiment demonstrates the practical implementation of machine learning algorithms in beehive monitoring, directly addressing the research aim of maintaining detection accuracy. Replicating the experiment across multiple datasets validates the robustness of the findings and enhances confidence in the conclusions drawn, highlighting the broader applicability of FLAC in diverse beehive monitoring scenarios. Overall, the experiment provides crucial evidence for assessing FLAC's suitability in enhancing transmission efficiency and maintaining detection accuracy, contributing to advancements in beehive monitoring technology.

1.4.2 Scope

- Evaluate the Free Lossless Audio Codec for use in beehive monitoring systems.
- Collect and analyze beehive audio data from different sources.
- Compare FLAC compression with other formats like WAV and MP3.
- Apply machine learning models to assess the impact of compression on data accuracy.
- Develop recommendations for implementing FLAC in practical monitoring systems.
- Focus solely on audio data compression and analysis, excluding other aspects of beekeeping technology such as visual monitoring or chemical analysis.

1.4.3 Definitions

Beehive Monitoring: The process of observing and analysing the conditions and activities within a beehive to ensure the health and productivity of the bee colony.

FLAC (Free Lossless Audio Codec): An audio compression format that reduces file size without any loss of audio quality, allowing the original data to be perfectly reconstructed from the compressed data.

Lossy Compression: A data compression method that reduces file size by permanently eliminating certain information, resulting in a loss of quality that cannot be recovered (Kim B. &., 2018).

Lossless Compression: A data compression method that reduces file size without any loss of information, allowing the original data to be fully reconstructed from the compressed data (Hans, 2001).

Machine Learning: A subset of artificial intelligence that involves the development of algorithms and statistical models that enable computers to perform tasks without explicit instructions, relying instead on patterns and inference.

1.5. THESIS OUTLINE

- **Chapter 2: Literature Review**

The literature review covers several key areas relevant to the study. It begins with an exploration of the historical context of beekeeping and the current challenges facing the industry. It then delves into traditional practices and recent technological advancements in beehive monitoring. This is followed by an examination of audio data compression techniques, comparing different formats such as WAV, MP3, and FLAC. The review also investigates the application of machine learning, particularly focusing on MFCC and SVM, in analyzing beehive sounds. Finally, it identifies gaps in existing literature, highlighting the underexplored use of FLAC in beehive monitoring systems.

- **Chapter 3: Research Design**

This chapter outlines the research design, focusing on the variables, hypotheses, instruments, procedures, and ethical considerations. The independent variable is the audio compression format, while the dependent variables are classification accuracy and file size. The hypotheses examine the impact of audio compression formats on these dependent variables, aiming to validate the effectiveness of FLAC compression in preserving classification accuracy while reducing file size. The study uses relevant hardware and software tools for audio compression and analysis. The procedure and timeline section details the research steps and schedule, from data collection to reporting. Finally, the ethics and limitations section addresses ethical considerations and acknowledges study limitations.

- **Chapter 4: Methodology**

This chapter details the methodology of our experiment. The independent variable is the audio compression format, and the dependent variables are classification accuracy and file size. Hypotheses aim to validate FLAC compression's effectiveness in maintaining accuracy while reducing size. We discuss FLAC theory for lossless compression, MFCC theory for audio feature extraction, and SVM theory for audio classification. The study uses relevant hardware and software tools. The procedure and timeline outline the research steps, and the ethics and limitations section address ethical considerations and acknowledges study limitations.

- **Chapter 5: Experiment**

This chapter details the experimental process, including data collection, compression, transmission, feature extraction, and classification. We collected audio data and compared compression time among three formats using ffmpeg for compression. Transmission times were assessed using MQTT protocol. For feature extraction, we utilized MFCC, which is crucial for capturing the audio signal's characteristics. The classification accuracy was evaluated using an SVM classifier. To ensure robustness, we validated our method with two additional datasets, comparing the results across different sources to demonstrate the reliability and effectiveness of our approach.

- **Chapter 6: Results**

This chapter presents the results of our experiment. We compare the classification accuracy of the three audio formats, providing a detailed classification report, confusion matrix, file size and ROC curve for each format. Additionally, we analyze the compression time cost and transmission time cost for each format. These results offer a comprehensive evaluation of the performance and efficiency of the audio compression and classification methods used in our study.

- **Chapter 7: Discussion**

This chapter discusses the implications of the results presented in Chapter 5. We analyze the waveform characteristics of the audio data, examining how different compression formats affect the audio signal. We also discuss the compression speed and degree, evaluating how efficiently each format reduces file size while maintaining audio quality. Additionally, we consider transmission speed, exploring the impact of compression on the time required to transmit audio data using the MQTT protocol. This discussion provides insights into the trade-offs between compression efficiency, transmission speed, and classification accuracy.

- **Chapter 8: Conclusion**

Conclusion and Recommendations summarizes the findings of the research, evaluates the implications for the beekeeping industry, and discusses the limitations of the study. It also provides practical recommendations for implementing FLAC in beehive monitoring systems and suggests areas for future research to further enhance beekeeping technologies and practices.

Chapter 2. Literature Review

In this literature review, we embark on a journey through the intricate world of beehive monitoring. Our exploration begins with an examination of audio compression techniques, seeking efficient methods to handle the rich tapestry of bee-related sounds. As we delve deeper, we focus on the unique context of beehive monitoring, where audio compression plays a pivotal role. Real-world applications come to life, demonstrating how these compressed audio data streams contribute to disease detection, hive health assessment, and environmental monitoring. Amidst existing studies, we identify gaps ripe for investigation, setting the stage for our hypotheses.

2.1. AUDIO COMPRESSION TECHNIQUES

2.1.1 Uncompressed Audio

Uncompressed audio formats are characterized by their ability to store audio signals without any form of compression, preserving the original audio quality exactly as it was recorded (Djebbar, F., & Ayad, B., 2017). These formats are used in various professional and high-fidelity applications where audio quality is of utmost importance (Polyak, 2021 June).

2.1.1.1 WAV

The Waveform Audio File Format (WAV), a standard for audio file storage on computers, is integral to various fields such as digital audio processing, multimedia applications, and broadcasting. Established by Microsoft and IBM, WAV files are known for their quality and versatility, though they come with trade-offs in terms of file size. The WAV format was developed in 1991 as part of the Resource Interchange File Format (RIFF), designed to store data in tagged chunks (Microsoft, 1991). This format became the cornerstone for audio file storage, particularly on Windows platforms, due to its compatibility and simplicity.

Technical Specifications

WAV files store audio data in an uncompressed format, typically using the Linear Pulse Code Modulation (LPCM) method. This ensures high fidelity by preserving the audio's original waveform (Pohlmann, 2011).The primary advantage of this uncompressed format is the retention of audio quality, making WAV files suitable for professional audio applications.

However, the trade-off is a larger file size compared to compressed formats like MP3 or AAC (Spanias, A., & Atti, V. , 2010). WAV files are extensively used in professional audio recording and editing, sound design, and broadcasting. Their uncompressed nature allows for easy editing and high-quality audio reproduction, which is essential for professional work (Watkinson, 2013). Furthermore, many software applications and digital audio workstations (DAWs) support WAV files natively, enhancing their utility in professional settings (Mulyadi, Y., & Daryana, H. A., 2021).

Applications and Usage

i. Professional Audio Production

In professional audio production, WAV files are indispensable due to their high quality and versatility (Djebbar, F., & Ayad, B., 2017). Audio engineers and producers rely on WAV files throughout the entire production process, from initial recording to final mastering.

During recording sessions in studios, WAV files are used to capture high-fidelity audio. The format's support for various sample rates and bit depths allows for the detailed capture of sound, accommodating everything from standard CD-quality (16-bit/44.1 kHz) to high-resolution audio (24-bit/96 kHz and beyond) (Frantiska Jr, 2008). This high-quality capture is essential for ensuring that the recording accurately reflects the performance and nuances of the audio source (KARAGIANNIS, 2000).

ii. Sound Design

WAV files are essential in sound design for movies, television, video games, and other multimedia applications. Sound designers use WAV files to create, manipulate, and integrate sound effects with high precision and fidelity. In film and television production, sound designers use WAV files to produce and integrate sound effects, dialogue, and background scores (Griffin, 2016). The high resolution and uncompressed quality of WAV files allow for precise synchronization with visual elements, ensuring that every sound detail enhances the viewer's experience (Sharma, 2022). For instance, sound effects like explosions, footsteps, and ambient noises can be meticulously crafted and placed in the audio track to match the on-screen action perfectly.

iii. Scientific Research

Uncompressed audio formats like WAV are extensively used in AI and machine learning for training and testing purposes. The high-quality, unaltered audio data ensures that algorithms receive the most accurate input possible, which is crucial for developing sophisticated models for speech recognition, audio analysis, and other AI applications (Jaganmohan, 2023).

In the realm of speech recognition, WAV files play a critical role. These files provide the purest form of audio, free from any compression artifacts that might distort the sound. When developing speech recognition systems, it is essential to have clear and accurate audio data so that the models can learn the nuances of human speech, including different accents, intonations, and speaking speeds (Herrmann, 2023). By using WAV files, researchers and developers can ensure that their training data reflects the true characteristics of spoken language, leading to more robust and accurate speech recognition systems (Zheng G. X., 2021).

Audio analysis, another key area in AI, also benefits from the use of WAV files. Tasks such as sound classification, event detection, and audio segmentation require high-quality audio inputs to accurately identify and classify sounds (Gupta, 2017). For instance, in environmental sound classification, models are trained to distinguish between different types of sounds like traffic noise, bird songs, or machinery sounds (Cristia, 2021). The uncompressed nature of WAV files ensures that the subtle details and characteristics of these sounds are preserved, enabling the models to learn and perform more effectively.

Advantages and Limitations

The primary advantage of WAV files is their superior audio quality. This is particularly important in professional settings where audio fidelity cannot be compromised. Additionally, the straightforward structure of WAV files makes them easy to process and manipulate (Bosi, M., & Goldberg, R. E. , 2002). However, the large file size can be a significant limitation, particularly for storage and transmission over networks with limited bandwidth.

Recent Developments and Future Directions

Recent advancements in storage technology and bandwidth have somewhat mitigated the drawbacks of WAV files' large size (Rashidinejad, 2013). Additionally, new developments in audio compression algorithms continue to improve the balance between quality and file size. The ongoing evolution of digital audio technologies ensures that WAV files remain relevant, particularly in professional and archival contexts (Brandenburg, 2010).

The Waveform Audio File Format remains a critical component of digital audio technology, balancing quality and ease of use against file size. Its enduring relevance in professional audio underscores the importance of maintaining high-fidelity audio storage formats in an increasingly compressed digital world (Liu, 2011).

2.1.2 Lossy Compression

Lossy compression techniques are used to reduce the file size of audio data by removing parts of the sound that are less audible to human ears (Kim, B., & Rafii, Z, 2018, September). This process results in a significant reduction in file size with some loss of audio quality. One of the most common lossy compression formats is MP3.

2.1.2.1 MP3

The MP3 format, or MPEG-1 Audio Layer 3, is one of the most popular lossy audio compression formats. Developed by the Moving Picture Experts Group (MPEG), MP3 revolutionized the way audio data is stored and transmitted, making digital audio more accessible and portable (Brandenburg, 2010). MP3 compression works by using a perceptual audio coding method that reduces the accuracy of certain sound components that are less perceptible to human hearing. This method exploits the limitations of human auditory perception to achieve high compression rates while maintaining acceptable audio quality (Spanias, A., & Atti, V. , 2010).

Technical Specifications

The MP3 format, or MPEG-1 Audio Layer 3, revolutionized digital audio storage and transmission with its efficient lossy compression method developed by the Moving Picture Experts Group (MPEG) (Brandenburg, 2010). MP3 compression, based on psychoacoustic principles, selectively removes inaudible frequencies and masked sounds, achieving compression ratios of about 10:1 compared to uncompressed WAV files while maintaining acceptable audio quality. It supports a wide range of bit rates (typically from 32 kbps to 320 kbps), sampling rates (including 32 kHz, 44.1 kHz, and 48 kHz), and channels (mono, stereo, joint stereo, and dual channel modes), providing flexibility for various audio applications (McCandless, 1999). MP3 files use a frame-based structure with headers containing encoding parameters, allowing for variable bit rate (VBR) encoding to optimize quality and file size balance. Additionally, MP3 files often include metadata in ID3 tags for organizing and managing digital music libraries (Egidi, 2005). While MP3 provides efficient compression, it sacrifices some error resilience compared to lossless formats, making it susceptible to quality loss from corruption or transmission errors.

Applications and Usage

MP3 became widely popular due to its balance between sound quality and file size, which facilitated the sharing and distribution of music over the internet. It played a crucial role in the rise of digital music distribution and streaming services (Kozamernik, 2002). MP3 files are supported by virtually all digital audio players and software, making them extremely versatile and user-friendly.

i. Music Distribution

The MP3 format significantly impacted the music industry by enabling the easy and efficient distribution of music online. Services like Napster, iTunes, and later streaming platforms such as Spotify and Apple Music, owe their success in part to the MP3 format. These platforms allowed users to download or stream high-quality music files without requiring extensive storage space, leading to the widespread adoption of digital music (Kozamernik, 2002). MP3 files made it possible for users to create extensive personal music collections on their computers and portable devices (Beckman, 2006). This shift was a departure from physical media such as CDs and vinyl records, allowing users to store thousands of songs in a fraction of the space. The convenience of carrying a vast music library on devices like iPods, smartphones, and MP3 players contributed to the format's popularity (Manning, 2013).

ii. Professional Recording and Editing

In the professional realm, MP3 files are commonly used for distributing preliminary mixes and demos due to their manageable size and decent audio quality. Although uncompressed formats like WAV are preferred for recording and mastering, MP3 files provide a practical solution for sharing audio files quickly and efficiently among collaborators and clients (Pohlmann, 2011). The MP3 format has also become the standard for podcasts and audiobooks. Its compression capabilities make it ideal for long-form audio content, which can be downloaded or streamed with ease. The format's ubiquity ensures compatibility with a wide range of playback devices, from computers to dedicated audiobook readers and smart speakers (Berry, 2006).

iii. Data Storage and Transmission

MP3's efficient compression allows for significant savings in storage space and bandwidth. This is particularly beneficial for online content delivery, where storage costs and data transfer limits are critical considerations (Li, 2010, July). Websites and online platforms can host large libraries of audio content without incurring excessive costs, making the MP3 format an economic choice for many content providers (Spanias, A., & Atti, V. , 2010). While not typically used for archival purposes due to its lossy nature, MP3 files still find a place in various research and educational contexts. Their small size makes them suitable for use in databases and digital archives where space constraints exist, although higher quality formats are preferred for preserving the integrity of original recordings (Burkart, 2008).

Advantages and Limitations

The main advantage of MP3 is its ability to compress audio files to a fraction of their original size while retaining a level of quality that is generally satisfactory for most listeners. This makes it ideal for use in portable devices and streaming over the internet where bandwidth and storage space are limited (Brandenburg, 2010). However, the lossy nature of MP3 means that some audio quality is inevitably sacrificed, which may not be acceptable for high-fidelity audio applications (D'Alessandro, 2009, September).

Recent Developments and Future Directions

Although newer audio codecs such as AAC (Advanced Audio Coding) and OGG Vorbis have emerged, offering better sound quality at similar or lower bitrates, MP3 remains widely used due to its established infrastructure and compatibility (Bosi, M., & Goldberg, R. E. , 2002). The ongoing challenge is to continue improving audio compression techniques to balance quality, file size, and compatibility.

2.1.2.2 AAC

AAC uses a more advanced compression algorithm compared to MP3, which includes techniques such as temporal noise shaping, backward adaptive linear prediction, and increased flexibility in joint stereo coding. These techniques allow AAC to achieve higher efficiency and better sound quality at lower bitrates (Bosi, M., & Goldberg, R. E. , 2002). AAC supports sample rates from 8 kHz to 96 kHz, up to 48 channels, and provides more flexibility for future developments.

Technical Specifications

AAC uses a more advanced compression algorithm compared to MP3, resulting in better audio quality for a given bit rate. It utilizes techniques such as spectral band replication (SBR) and perceptual noise shaping (PNS) to achieve higher compression efficiency while preserving audio fidelity (Brandenburg, 2010). Unlike MP3, AAC supports a wider range of bit rates, sampling rates, and channel configurations, providing greater flexibility for different audio applications. Common bit rates for AAC encoding range from 64 kbps to 320 kbps, with higher rates offering better sound quality but larger file sizes. AAC also supports various sampling rates, including 8 kHz, 16 kHz, 22.05 kHz, 44.1 kHz, and 48 kHz, catering to different audio recording and playback requirements (Wolters, 2003, October). Additionally, AAC supports up to 48 audio channels, allowing for multichannel audio encoding for immersive surround sound experiences (Gunawan, 2017).

Applications and Usage

AAC is widely used in various applications due to its superior audio quality and efficient compression. It complements MP3 by providing enhanced audio experiences in many modern applications.

i. Broadcasting

AAC is extensively used in digital radio broadcasting. For instance, it is part of the Digital Radio Mondiale (DRM) and HD Radio standards, which require high-quality audio at low bitrates for efficient broadcasting. Its adoption in broadcasting ensures that listeners experience high-fidelity audio even at low signal strengths (Ibraheem, 2017).

ii. Video Streaming

AAC is also a standard audio format for video streaming platforms such as Netflix, Hulu, and YouTube. Its ability to deliver high-quality audio with efficient compression makes it ideal for video content, where it is often paired with video codecs like H.264 or H.265 (Kalampogia, 2017). This combination ensures that users experience high-quality audio and video without excessive buffering or data consumption.

iii. Mobile Devices

AAC is the preferred audio format for many mobile devices, including smartphones, tablets, and portable media players. Its efficient compression ensures that users can store more audio content on their devices without sacrificing quality (Fiannaca, 2017). Additionally, AAC is widely supported across various operating systems and playback software, enhancing its versatility.

iv. Gaming

The gaming industry also benefits from AAC’s capabilities. Modern video game consoles and PC games use AAC for in-game audio and streaming services like Twitch and YouTube Gaming, ensuring that gamers experience high-quality sound with minimal impact on performance and bandwidth (Boyd, 2017).

Advantages and Limitations

The primary advantage of AAC is its improved sound quality at lower bitrates compared to MP3. This makes it ideal for applications where bandwidth and storage space are at a premium (Takag, K., & Takishima, Y., 2007 July). However, despite its advantages, AAC has not completely replaced MP3, partly due to MP3's entrenched position and widespread compatibility. Additionally, some older devices and software may not support AAC, limiting its usability in certain contexts (Higginbotham, D. Jeffery, et al., 2007).

Recent Developments and Future Directions

Recent developments in AAC technology have focused on further improving audio quality, compression efficiency, and compatibility with emerging audio technologies. One significant advancement is the introduction of High-Efficiency AAC (HE-AAC), also known as AAC+, which combines AAC with spectral band replication (SBR) to achieve even higher compression ratios without sacrificing audio quality (Deshmukh, Soham, et al., 2024). HE-AAC is particularly well-suited for streaming applications, where bandwidth efficiency is crucial for delivering high-quality audio over limited network connections.

Looking ahead, future directions for AAC technology include exploring new compression techniques, such as machine learning-based approaches, to further enhance audio quality and compression efficiency. Additionally, ongoing standardization efforts aim to extend AAC's capabilities for emerging audio formats and delivery platforms, ensuring its relevance and compatibility in the evolving landscape of digital audio (ISO/IEC, 2020). With continued innovation and collaboration across the audio industry, AAC is poised to remain a cornerstone of digital audio compression for years to come (Vogel, 2024).

2.1.2.3 Opus

Opus is an advanced lossy audio compression format that has gained significant popularity for its versatility and high-quality audio output. Developed by the Internet Engineering Task Force (IETF), Opus is designed to handle a wide range of audio applications, from low-latency voice communication to high-fidelity music streaming.

Technical Specifications

Opus combines the strengths of two existing codecs: SILK, used primarily for voice encoding, and CELT, designed for high-quality audio. By integrating these codecs, Opus achieves exceptional efficiency and adaptability, capable of providing high audio quality across various bitrates and conditions (Han, 2014). Opus supports bitrates from 6 kbps to 510 kbps, sample rates from 8 kHz to 48 kHz, and frame sizes from 2.5 ms to 60 ms, making it highly versatile for different audio needs.

One of the defining features of Opus is its ability to seamlessly switch between SILK and CELT modes or combine them, depending on the requirements of the audio being processed (DeCambra, Weston. , 2024). This flexibility allows Opus to optimize for different types of audio content and network conditions, ensuring optimal performance. Opus is particularly well-suited for real-time applications due to its low-latency capabilities. With a minimum algorithmic delay of just 5 ms, Opus can deliver near-instantaneous audio transmission, making it ideal for voice and video calls, live streaming, and interactive applications (Skoglund, Jan, and Jean-Marc Valin, 2019).

Applications and Usage

Opus's versatility makes it a preferred choice for a broad range of audio applications, from voice over IP (VoIP) and video conferencing to music streaming and gaming.

i. Voice Communication

Opus is widely adopted in voice communication platforms, including VoIP services like Skype, Discord, and WhatsApp. Its ability to provide high-quality audio at low bitrates, combined with its low-latency performance, ensures clear and reliable voice communication even under varying network conditions (Vos, 2012).

ii. Video Conferencing

In video conferencing, Opus enhances the overall experience by delivering clear and synchronized audio. Platforms such as Zoom and Google Meet use Opus to ensure that participants can communicate effectively, regardless of bandwidth fluctuations (Maruschke, Michael, et al., 2015 September). Opus is a standard codec in Web Real-Time Communication (WebRTC) applications, which are used for peer-to-peer communication over the web. WebRTC enables developers to build applications such as video chat, file sharing, and live streaming directly into web browsers, with Opus ensuring high-quality audio transmission (Suciu, 2020).

iii. Music Streaming

Opus is also used in music streaming services like Spotify, where its efficient compression and high audio quality improve the listening experience while minimizing data usage. The codec's flexibility allows it to adapt to different streaming qualities, providing a consistent experience across various devices and network conditions (Valin, 2016).

Advantages and Limitations

Opus offers several advantages, including superior audio quality at various bitrates, low-latency performance, and flexibility across different audio applications. Its open-source nature and royalty-free licensing make it an attractive choice for developers and companies looking to integrate high-quality audio into their products (Skoglund, Jan, and Jean-Marc Valin, 2019).

However, despite its many advantages, Opus is not as widely supported as more established formats like MP3 and AAC. Some older devices and software may not be compatible with Opus, limiting its usability in certain contexts. Additionally, while Opus is highly efficient, it may not always match the compression ratios of specialized codecs for very low-bitrate applications (Kaul, 2019).

Recent Developments and Future Directions

Opus continues to evolve, with ongoing improvements aimed at enhancing its performance and extending its capabilities. Recent updates have focused on optimizing the codec for better speech and music quality, as well as improving its efficiency for streaming and real-time communication applications. Future developments may include further refinements to the codec's adaptability and support for emerging audio technologies (Lin, 2024).

2.1.3 Lossless Compression

Lossless compression techniques are used to reduce the file size of audio data without any loss of information, ensuring that the original audio can be perfectly reconstructed from the compressed data (Deepu, 2017). This makes lossless formats ideal for applications where audio fidelity is paramount. Two widely used lossless compression formats are ALAC and FLAC.

2.1.3.1 ALAC

Apple Lossless Audio Codec (ALAC) is a lossless audio compression format developed by Apple Inc. It is designed to reduce the file size of audio tracks without compromising quality, making it particularly suitable for use within the Apple ecosystem.

Technical Specifications

ALAC works by using linear prediction, a method that models audio signals as a linear combination of their past samples. This allows ALAC to achieve compression ratios typically between 40% to 60% of the original file size, depending on the complexity of the audio content

(Sushkov, 2023). The codec supports various sample rates and bit depths, providing flexibility for different audio qualities and file sizes (Apple Inc., 2011).

Applications and Usage

ALAC is primarily used within Apple's ecosystem, including iTunes, iOS devices, and macOS. It is the preferred format for users who want to maintain high audio quality without the large file sizes associated with uncompressed formats like WAV or AIFF.

i. Music Libraries

ALAC is commonly used by audiophiles and music enthusiasts who manage their music libraries through iTunes or Apple Music. The format ensures that users can enjoy high-fidelity audio while saving storage space on their devices (Nanabeka, 2017).

ii. Professional Audio Work

In professional audio environments, ALAC is used for archiving and distributing high-quality audio files. Its lossless nature ensures that audio engineers and producers can work with exact replicas of original recordings, which is crucial for tasks such as mixing and mastering (Dittmar, 2017.).

iii. Streaming Services

Apple Music supports ALAC for streaming lossless audio to users. This provides an enhanced listening experience compared to lossy formats, catering to users who demand the highest audio quality (Williams, 2022).

Advantages and Limitations

The main advantage of ALAC is its ability to provide high-quality audio without the large file sizes of uncompressed formats. It also benefits from seamless integration within Apple's ecosystem, ensuring compatibility across various Apple devices and software (Plummer, (2014)). However, ALAC is not as widely supported outside of Apple's ecosystem, which can limit its usability with non-Apple devices and applications.

Recent Developments and Future Directions

Recent developments in Apple Lossless Audio Codec (ALAC) technology have primarily focused on enhancing compatibility, improving efficiency, and expanding its adoption across various platforms and devices (Porter, Alastair, et al., 2015 Oct). One notable development is the increased integration of ALAC support into third-party software and hardware products, enabling users to enjoy high-quality lossless audio playback across a broader range of ecosystems (Apple Inc., 2011).

Apple has also continued to refine the ALAC codec to optimize its performance and efficiency, particularly in the context of streaming and online distribution. With the growing popularity of high-resolution audio streaming services, there has been a concerted effort to ensure that ALAC remains a viable option for delivering pristine audio quality over the internet (Apple Inc., 2011).

2.1.3.2 FLAC

Free Lossless Audio Codec (FLAC) is a widely used open-source format known for its ability to compress audio files without any loss of quality. Developed by the Xiph.Org Foundation (Železnik, 2020), FLAC is popular among audiophiles, professionals, and music enthusiasts who prioritize high audio fidelity and versatile compatibility.

Technical Specifications

FLAC employs linear prediction and residual coding to achieve its lossless compression. This method allows the codec to reduce file sizes by about 50% to 70%, depending on the complexity of the audio (Fang, 2009). The format supports a range of sample rates from 1 Hz to 655,350 Hz and bit depths from 4 to 32 bits per sample, ensuring flexibility and high-quality reproduction for various audio types (Firmansah, 2016). FLAC works by analyzing the audio data and identifying patterns that can be efficiently encoded. It uses a combination of predictive coding and entropy coding to compress the audio data without losing any information. The process involves breaking down the audio signal into blocks, applying linear prediction to model the signal, and then encoding the residual difference between the predicted and actual signal using entropy coding (Ye J. K., 2010, October).

Applications and Usage

FLAC's lossless compression and open-source nature make it ideal for a wide range of applications, from personal music libraries to professional audio production.

i. Personal Music Collections

Many audiophiles prefer FLAC for managing their personal music collections due to its superior sound quality and open-source license. FLAC files are compatible with numerous media players, operating systems, and devices, making it easy for users to organize and enjoy their music without compromising on quality (Sάenz Lόpez, 2022). The format's support for metadata tagging allows users to store rich information about their music, such as artist, album, and track details, enhancing the overall listening experience. In professional audio environments, FLAC is used for recording, editing, and archiving high-quality audio. Its lossless nature ensures that audio professionals can work with exact replicas of original recordings, which is crucial for tasks such as mixing, mastering, and restoration (Kumar, 2014). The format's ability to handle high-resolution audio makes it a preferred choice for preserving the integrity of original recordings.

ii. Recording and Editing

FLAC is often used during the recording and editing phases in professional studios. It allows sound engineers to capture and manipulate audio without worrying about data loss, ensuring that the final product retains the highest possible quality (Jackson, 2015). For archiving purposes, FLAC's lossless compression is invaluable. It provides a space-efficient way to store audio recordings without compromising their fidelity, making it ideal for preserving historical audio recordings and valuable sound archives (Kromer, 2017).

iii. Music Distribution

Artists and record labels frequently use FLAC to distribute music to ensure listeners receive the highest quality audio. Online music stores and distribution platforms such as Bandcamp, HDtracks, and Tidal offer FLAC downloads and streams for users who demand lossless audio (Cho, 2007, July). The format's open-source nature and royalty-free licensing encourage widespread adoption among distributors and consumers alike. Many online music stores offer FLAC as an option for downloading high-quality audio. This ensures that consumers have access to the best possible sound quality for their purchases, catering to the demands of audiophiles and music enthusiasts (Rivero, 2008). Streaming services that focus on high-fidelity audio, such as Tidal and Qobuz, use FLAC to deliver lossless streams. This allows subscribers to enjoy studio-quality sound, making FLAC a key component in the competitive landscape of premium audio streaming (Sisario, 2019).

iv. Data Analysis Application

Jin and Kim (Jin, R., & Kim, J., 2014) proposed a method for recovering FLAC music files downloaded via BitTorrent by decoding split FLAC files, as detailed in their paper "Analysis of FLAC Music Pieces Recovery." The recovery process involves obtaining complete frames from partial FLAC files and adding temporary headers to facilitate decoding, achieving a success rate of over 90%. The study addresses the challenges posed by BitTorrent, a popular

P2P file-sharing protocol that often results in the distribution of FLAC files in fragmented pieces, complicating the extraction of essential metadata for copyright protection (Hawa, 2012). FLAC, or Free Lossless Audio Codec, compresses audio without any loss of information, typically reducing file sizes by 50-60%. The basic structure of a FLAC file includes a mandatory STREAMINFO block and audio frames (Ye H. a., 2018). The authors' proposed method involves adding a temporary header to facilitate the decoding of these fragmented FLAC files. In their experiments with 360 pop songs, they demonstrated a decoding success rate exceeding 90% for piece sizes ranging from 128KB to 16MB (Jin, 2014). Additionally, the study explores the extraction of musical features such as zero-crossing rate, signal energy, and pitch using methods like FFT-based spectrum analysis (Ye J. K., 2010, October). Tonality analysis is performed using chromogram, which maps frequency data to pitch classes, aiding in the identification of audio. The results indicate that the decoding and identification success rates improve with larger piece sizes, with piece sizes above 1MB showing stable and high success rates. This method effectively recovers and identifies FLAC audio files distributed via BitTorrent, aiding in tracking illegal content without needing the complete file, thus having significant implications for digital copyright protection (Kim H. , 2017).

Advantages and Limitations

FLAC offers numerous advantages that make it ideal for high-fidelity audio applications. Its lossless compression ensures the original audio quality is preserved, providing an excellent solution for audiophiles and professionals who require the highest sound fidelity. As an open-source and royalty-free format, FLAC encourages widespread use and support across various platforms, with no licensing fees (Koller, 1999, September). This broad compatibility is a significant benefit, as FLAC is supported by many media players, devices, and operating systems, ensuring its usability across different contexts. Additionally, FLAC provides efficient compression, significantly reducing file sizes without any loss of quality, thereby balancing storage needs and maintaining audio fidelity (Gunawan, 2017).

However, FLAC files are generally larger than lossy formats like MP3 or AAC, which can be a drawback for users with limited storage space or bandwidth. Additionally, while FLAC is widely supported, there are still some older devices and software that may not be compatible with the format (Fang, 2009).

Recent Developments and Future Directions

FLAC continues to evolve, with ongoing improvements aimed at enhancing its performance and expanding its capabilities. Recent updates have focused on optimizing the codec for better compression efficiency and improving support for high-resolution audio (Debnath, 2024). Future developments may include further enhancements to the codec's adaptability and integration with emerging audio technologies. FLAC is also used in various research and development contexts where high-quality audio is essential. Researchers working on audio compression algorithms, acoustic analysis, and audio processing often use FLAC to ensure that their work is based on accurate and unaltered audio data (Vernyi, 2024).

2.2. AUDIO COMPRESSION IN BEEHIVE MONITORING

2.2.1 Importance of Audio Data in Beehive Monitoring

Beehive monitoring has become an essential aspect of modern apiculture, leveraging technological advancements to ensure the health and productivity of bee colonies (Qandour, Amro, et al., 2014). One of the most valuable data types collected in this context is audio data. The significance of audio data in beehive monitoring is multifaceted, providing insights into the colony's health, behaviour, and environmental conditions.

2.2.1.1. Health Monitoring

Audio data allows beekeepers to monitor the health of the hive by analyzing the sounds produced by bees. Healthy bees produce a distinct buzzing sound that can change when they are stressed, sick, or experiencing environmental issue. Specific audio signatures can indicate the presence of diseases, pests, or other health problems within the hive. For instance, the sound patterns can help detect the presence of the Varroa destructor mite, which is known to alter the acoustic environment of a hive (Ferrari, S., Silva, M., Guarino, M., & Berckmans, D, 2008).

A critical aspect of health monitoring through audio data is assessing the status of the queen bee. The queen's presence and health are vital for colony stability, as she is responsible for laying eggs and producing pheromones that regulate the hive's activities. Changes in the audio patterns can indicate the queen's health status or her absence, which requires immediate attention (Ali et al., 2021, November). Piping and tooting are specific sounds associated with queen bees. Piping is a high-pitched sound made by virgin queens, often signalling their presence to worker bees and other queens. Tooting is a response from other queens or worker bees. These sounds are crucial for maintaining the social hierarchy within the hive and preventing conflicts. The presence of piping and tooting can indicate the process of queen replacement or the presence of multiple queens (Yang, M. D., & Su, T. C., 2008).

Besides queen bee sound, normal bee sounds also provide a wealth of information about the health, behaviour, and environment of a bee colony. By continuously monitoring these sounds, beekeepers can maintain the well-being of their hives and ensure the productivity and sustainability of their colonies. The integration of audio monitoring with advanced analytical tools enhances the ability to detect and respond to issues promptly, making it a valuable component of modern beekeeping practices (Blut, 2017).

2.2.1.2. Behavioural Insights

The behaviour of bees, including activities such as foraging, swarming, and communication, can be effectively monitored through audio recordings. Bees communicate through a variety of sounds, including vibrations and buzzing. By analysing these sounds, researchers and beekeepers can gain insights into the hive's internal activities and predict behaviours such as swarming, which is critical for colony management and prevention of colony loss (Rustam, 2024).

Bee sounds are integral to understanding various behaviours within a hive, making them valuable for effective beehive monitoring. Different sounds correspond to specific activities such as foraging, swarming, queen events, colony defence, temperature regulation, and brood care (Zaman, A., & Dorin, A., 2023). For example, the intensity and frequency of buzzing can indicate foraging levels or the presence of swarming, while specific queen sounds signal queen replacement or swarming events (Banharnsakun, 2019). Defensive sounds highlight potential threats, and temperature regulation buzzes reflect hive conditions. By analysing these acoustic signals, beekeepers gain real-time insights into the hive’s health and activities, enabling proactive management and timely interventions to ensure colony stability and productivity. This non-invasive monitoring method enhances beekeeping practices and contributes to the overall health of bee colonies (Hall, 2023).

2.2.1.3. Environmental Monitoring

Audio data provides crucial insights into the environmental conditions within and around the hive, significantly impacting bee activity and health. Changes in the hive’s acoustic environment can indicate alterations in temperature, humidity, and other environmental factors (Cecchi, 2020). For instance, the buzz frequency of bees can change with temperature fluctuations, as bees fan their wings to cool the hive during hot weather or generate heat during colder periods. These temperature-related sounds allow beekeepers to infer whether the hive's internal environment remains within optimal ranges, ensuring that the bees are neither overheating nor freezing (Kridi, 2016).

Humidity levels also influence bee behaviour and hive acoustics. High humidity can affect the bees' ability to maintain hive conditions, leading to increased buzzing as they work harder to regulate the internal environment (Gil-Lebrero e. a., 2016). Conversely, low humidity might alter their feeding and brood-rearing activities, impacting colony health and productivity. By monitoring these acoustic signals, beekeepers can take necessary actions to maintain appropriate humidity levels, such as providing additional water sources or improving hive ventilation. (Tashakkori, 2021)

Additionally, audio data can reveal the presence of intruders or disturbances. A sudden spike in noise levels might indicate an animal or human intruder, prompting immediate action from the beekeeper to protect the hive. Persistent, high-pitched buzzing could also signal stress or discomfort among the bees, possibly due to environmental stressors like pesticide exposure or nearby construction activities (Murphy, 2015. June).

2.3. PRACTICAL APPLICATIONS OF BEEHIVE MONITOR

Beehive monitoring has evolved significantly with the advent of modern technologies, transforming traditional beekeeping practices. The integration of IoT (Internet of Things) and edge computing has facilitated real-time monitoring and management of beehives, allowing beekeepers to remotely oversee hive conditions and bee behaviour with unprecedented accuracy and convenience.

2.3.1 Basic Infrastructure

The basic infrastructure for modern beehive monitoring systems often includes IoT devices equipped with various sensors and audio recording capabilities. These devices collect data on hive temperature, humidity, and audio signals. Edge computing plays a crucial role in processing this data locally at the hive site, reducing the latency and bandwidth requirements associated with transmitting large volumes of raw data to a central server. This setup ensures that critical data is analyzed promptly, enabling swift responses to any issues that may arise within the hive (Tashakkori, 2021).

Efficient audio data transmission is essential for continuous monitoring. IoT-enabled beehive monitors typically employ wireless communication protocols such as Zigbee, Wi-Fi, or LoRaWAN to send data from the hive to a central hub (Kontogiannis, 2019). From there, data can be transmitted to cloud-based servers for long-term storage and more in-depth analysis. The challenge here lies in maintaining a balance between data fidelity and transmission efficiency, particularly in remote locations where network connectivity might be limited (Tashakkori, 2021).

2.3.2 AI-Driven Analysis

The analysis of beehive audio data has greatly benefited from the integration of machine learning (ML) and deep learning (DL) techniques. These advanced methods enable the automatic detection and classification of various bee sounds, which correspond to different hive activities and health conditions (Ali et al., 2021, November). For instance, ML algorithms can identify the unique buzzing patterns associated with foraging, swarming, or the presence of a queen. Deep learning models, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), have proven particularly effective for complex pattern recognition tasks, providing high accuracy in distinguishing between normal and abnormal bee sounds (Nassif, Shahin, Attili, Azzeh, & Shaalan, 2019).

Various feature extraction techniques, such as Mel-frequency cepstral coefficients (MFCC) and linear predictive coefficients (LPC), have been employed in beehive monitoring systems to extract meaningful data from audio signals (Zheng, Zhang, & Song, 2001). These features are then used in conjunction with classifiers like Gaussian mixture models (GMM) and hidden Markov models (HMM) to categorize bee activities. The design of an IoT system for acoustic swarm monitoring using MFCC and LPC features, compared with classifiers like GMM and HMM, has shown that HMMs can provide effective solutions for the classification of bee activities. Many established approaches from automatic speech recognition (ASR) have been adapted for this purpose (Zgank, 2021).

Despite the success of HMMs and GMMs in some tasks, our research leverages deep neural networks (DNNs) due to their superior performance in complex pattern recognition tasks and their proven effectiveness in related fields such as ASR (Zacepins, 2015). DNNs are capable of handling large datasets and learning intricate features that simpler models might miss, making them ideal for the acoustic analysis of beehive sounds (Andrijević, 2022). CNNs and RNNs are adept at capturing the temporal and spatial characteristics of audio data, enhancing the accuracy of acoustic classifications. This approach represents an efficient solution, facilitating rapid development and deployment in practical applications (Yesodha, 2024).

The main contribution of this work is to apply deep learning models for bee activity acoustic classification, building upon the foundation of feature extraction techniques like MFCC (Abdollahi, 2022). By utilizing DNNs, we aim to achieve high accuracy in classifying bee sounds, ensuring reliable monitoring of hive health and activities. Additionally, we explore the impact of audio compression, such as the lossy MP3 codec, on the classification performance, striving to develop methods that minimize feature loss while maintaining data integrity (Zaman, A., & Dorin, A., 2023).

The study by Niella et al. (Niell, 2018) explores the use of Support Vector Machines (SVM) to evaluate beehives as biomonitors for pesticide presence in agroecosystems. The research aimed to classify environments based on the impact of pesticides using both biological indicators (such as bee population and brood area estimations) and chemical indicators (including the number of pesticides detected and related toxic units). The SVM models demonstrated that while biological indicators alone provided a modest classification accuracy of 57%, incorporating chemical analysis significantly improved accuracy to 100% (Qandour, Amro, et al., 2014). This methodology highlights the potential of using beehive data to monitor environmental health and identify at-risk ecosystems, offering a cost-effective approach for large-scale agricultural biomonitoring.

2.3.3 Compression Formats in Beekeeping

The paper by Andrej Zgank (Zgank, 2021) introduces the application of MP3 and ACC (accelerometer) technologies as innovative tools for beehive monitoring, enhancing traditional methods of environmental assessment. The MP3 devices are employed to record the acoustic environment of beehives. By capturing audio data, researchers can analyze the soundscapes within the hive to infer the activity levels and overall health of the bee colony. This audio data serves as a non-invasive method to monitor bees, providing continuous and real-time insights into the hive dynamics (Cunha, 2020).

In parallel, accelerometers (ACC) are utilized to measure the vibrations and movements within the beehive. These devices capture detailed information about the physical activity and behavior of bees. The data collected from accelerometers can reveal patterns related to bee movements, hive temperature regulation, and responses to external stressors such as pesticide exposure (Cecchi, 2020). By detecting subtle changes in the hive's vibrational profile, researchers can gain a deeper understanding of how environmental factors and contaminants affect bee colonies.

These technological advancements, MP3 and AAC, complement the biological and chemical indicators traditionally used in beehive monitoring (Zgank, 2021). The integration of sound and vibration data with chemical analyses of pesticide residues and biological assessments of bee populations and brood areas allows for a more comprehensive evaluation of hive health and environmental status. This multifaceted approach facilitates the identification of at-risk ecosystems and the seasons when bees are most vulnerable to pesticide exposure (Szczurek, 2023).

The use of MP3 and AAC technologies represents a significant step forward in agricultural biomonitoring. It provides researchers with robust, detailed, and real-time data that enhances the ability to detect and understand the impacts of agrochemicals on bee populations. This integrated monitoring framework not only improves the accuracy of environmental assessments but also supports the development of more effective strategies for mitigating the adverse effects of pesticides on vital pollinator species (Crawford, M., 2017).

Overall, the incorporation of MP3 and AAC technologies into beehive monitoring exemplifies the innovative use of modern tools to address complex ecological challenges. By leveraging these technologies, the study underscores the potential to enhance traditional monitoring methods, offering a more dynamic and holistic view of agroecosystem health and sustainability.

2.4. CHALLENGES IN BEEHIVE AUDIO DATA

In the context of beehive monitoring, managing audio data efficiently while maintaining data integrity presents several challenges:

2.4.1 Resource Consumption of Uncompressed Audio

Using uncompressed audio formats, such as WAV, results in exceptionally large file sizes. For instance, continuous 24-hour monitoring of a beehive generates vast amounts of data, which can quickly become unwieldy (Melchior, 2019). Large file sizes not only strain storage resources but also complicate data transmission. High volumes of data require substantial bandwidth for real-time monitoring and increase the energy consumption of IoT devices deployed in the field. This inefficiency can lead to frequent maintenance, higher operational costs, and potential data loss due to storage overflow (Zacepins, 2015).

2.4.2 Distortion from Compressed Audio

On the other hand, using compressed audio formats like MP3 introduces another set of challenges. While MP3 compression significantly reduces file size, it is a lossy compression format, meaning that some audio data is irretrievably lost during the compression process. This loss can lead to distortions that may obscure critical acoustic features necessary for accurate analysis (Yang, Yunzhao, et al., 2019). For example, the nuances in the buzzing patterns of bees, which are essential for detecting specific behaviors or health conditions, may be degraded. This degradation can adversely affect the performance of AI algorithms used for classifying and analyzing bee sounds, leading to reduced accuracy and reliability of the monitoring system (Li, 2010, July).

In the paper by Zgank (Zgank, 2021), the authors address the issue of distortion among different MP3 bitrates when using audio recordings for beehive monitoring. The quality of audio data is crucial for accurately interpreting the acoustic environment within beehives. The bitrate of an MP3 file, which determines the amount of data processed per second, directly impacts the fidelity and clarity of the recorded sounds.

The study highlights that lower bitrate MP3 recordings tend to introduce more compression artifacts and distortions, which can obscure important acoustic signals necessary for assessing hive activity and health. These distortions can make it challenging to distinguish between natural bee sounds and other noises, potentially leading to misinterpretations or missed detections of critical events within the hive (Khalil, 2021).

The authors suggest that careful consideration must be given to the choice of MP3 bitrate when deploying audio recording devices in beehives. Ensuring high-quality audio recordings is essential for the reliable use of acoustic data in environmental monitoring and assessment. By selecting appropriate bitrates, researchers can maximize the utility of MP3 recordings, providing clearer and more precise data for evaluating the health and activity of bee colonies (Crawford, M., 2017).

This discussion underscores the importance of technical specifications in the effective application of MP3 technology for ecological research. It highlights that while modern tools offer significant advantages, the quality of data collection must be maintained to achieve accurate and meaningful results in environmental monitoring.

2.4.3 Implications for Beehive Monitoring

The challenges associated with both uncompressed and compressed audio data necessitate a careful consideration of the trade-offs between file size and audio quality. Efficient beehive monitoring systems must balance these factors to ensure that the data remains manageable while preserving the essential acoustic features required for accurate analysis.

Our research aims to address these challenges by exploring audio compression techniques that minimize feature loss. This approach seeks to develop solutions that provide a practical balance, enabling the effective transmission and storage of audio data without compromising the integrity needed for reliable AI-driven analysis (Guruprasad, 2024). By optimizing audio compression strategies, beehive monitoring systems can become more scalable and effective, ultimately supporting better decision-making in beekeeping practices.

2.5. RESEARCH GAP: AUDIO COMPRESSION WITHOUT FEATURE LOSS

Given the challenges outlined above, there is a clear need for innovative solutions that address the limitations of both uncompressed and lossy compressed audio formats in beehive monitoring. Existing research and applications have primarily focused on either maintaining high data fidelity with uncompressed formats or reducing file size at the cost of some level of data loss with formats like MP3.

However, a significant research gap exists in the development of audio compression techniques that effectively balance these two aspects—retaining critical audio features essential for AI-based analysis while significantly reducing file size. This gap highlights the necessity for advanced compression algorithms capable of preserving the nuanced acoustic features of bee sounds, which are crucial for accurate monitoring and analysis.

Our research is poised to fill this gap by investigating and developing new compression methodologies that ensure minimal feature loss. These methodologies aim to maintain the integrity of audio data to support precise classification and analysis of bee behaviors and hive health conditions. By achieving this balance, our work will contribute to more efficient, reliable, and scalable beehive monitoring systems, facilitating better management and decision-making for beekeepers.

2.6. HYPOTHESES AND RESEARCH QUESTIONS

2.6.1 Feasibility of FLAC

FLAC presents a feasible solution for beehive monitoring due to its ability to preserve audio quality, efficient compression ratio, and open-source nature. Unlike lossy formats like MP3, FLAC compresses audio without any loss of fidelity, which is crucial for detecting subtle sound variations that indicate hive health (Firmansah, 2016). It achieves a significant size reduction, typically 30-50% smaller than uncompressed WAV files, making it practical for storing and transmitting large audio datasets efficiently while maintaining quality. Additionally, FLAC's fast and efficient encoding and decoding processes are beneficial for real-time monitoring applications, allowing timely analysis and alerts for activities such as swarming or predator presence (Zhao, 2007). As an open-source format, FLAC is free to use and supported across various platforms, making it accessible to beekeepers and researchers working with budget constraints (van Beurden M. Q., 2022 Aug. 21).

2.6.2 Hypotheses

FLAC compression can significantly reduce audio file size without compromising the accuracy of AI-based beehive monitoring systems.

The hypothesis suggests that employing FLAC compression in audio files for beehive monitoring systems will yield substantial reductions in file size while ensuring that the accuracy of AI-based monitoring remains unaffected. This hypothesis anticipates that the compression algorithm will efficiently reduce the data footprint without compromising the fidelity of the audio recordings, thus enhancing the overall efficiency and practicality of beehive monitoring systems without sacrificing accuracy in activity classification.

2.6.3 Research Questions

- **Question 1**

How does FLAC compression impact the classification accuracy of different bee activities compared to uncompressed and MP3 compressed formats?

- **Question 2**

What are the storage and transmission efficiency gains achieved by using FLAC in beehive monitoring?

These two research questions logically address key aspects of the study's objectives. The first examines how FLAC compression affects the classification accuracy of bee activities compared to uncompressed and MP3 formats, crucial for effective monitoring. The second question evaluates the storage and transmission efficiency gains of using FLAC in beehive monitoring, essential for optimizing resource usage in real-world applications. Together, they form a concise yet comprehensive framework for exploring FLAC's impact on both accuracy and efficiency in bee monitoring.

2.7. SUMMARY AND IMPLICATIONS

In the exploration of audio compression techniques (2.1), their application in beehive monitoring (2.2), and the practical implications of such monitoring (2.3), challenges emerge, notably in managing beehive audio data (2.4). These observations lead to the identification of a research gap: the need for data compression techniques without feature loss (2.5). Hypotheses and research questions (2.6) are then formulated to address this gap, aiming to assess the impact of compression methods on accuracy and efficiency in beehive monitoring systems.

Chapter 3. Research Design

In this pivotal chapter, we meticulously craft the blueprint for our research endeavor, focusing on the intricate design elements shaping our exploration. At the heart of our study lies the independent variable of compression format, directly influencing our dependent variables—classification accuracy and file size. Through carefully crafted hypotheses, consideration of hardware and software instruments, delineation of our timeline, and conscientious reflection on ethical implications and limitations, we establish a robust framework for inquiry. This chapter serves as the cornerstone of our methodological approach, ensuring rigor, transparency, and ethical integrity throughout our investigation.

3.1. INDEPENDENT VARIABLE

The choice of audio compression format serves as the independent variable because it is under the control of the researcher and can be manipulated independently of other factors. In experimental design, the independent variable is the factor that is deliberately changed or manipulated to observe its effect on the dependent variable(s). In this study, researchers can intentionally select and apply different compression formats (WAV, FLAC, MP3) to the audio data without any influence from external factors. By varying the compression format, researchers can examine how changes in compression levels affect dependent variables such as classification accuracy and file size. Therefore, the audio compression format meets the criteria for being an independent variable in this study as it can be systematically altered, and its effects observed and measured.

3.2. DEPENDENT VARIABLES

The classification accuracy and file size serve as dependent variables in this study because they are outcomes or measurements that are influenced by changes in the independent variable, which is the choice of audio compression format.

3.2.1 Classification Accuracy

This variable represents the accuracy of classification models trained on audio data that has been compressed using different formats (e.g., WAV, FLAC, MP3). The accuracy of the classification models is dependent on the quality of the compressed audio data, which in turn is influenced by the compression format applied. Higher classification accuracy indicates better performance in accurately classifying audio data into their respective categories (e.g., queen bee sound vs. no-queen bee sound).

3.2.2 File Size

This variable measures the size of audio files after compression, expressed in bytes or kilobytes. The file size is dependent on the level of compression applied by the chosen audio compression format. Different compression formats (e.g., lossless FLAC vs. lossy MP3) result in varying degrees of file size reduction while attempting to preserve audio quality. Therefore, the file size of compressed audio data is influenced by the compression format selected as the independent variable.

3.3. HYPOTHESES

These hypotheses serve as testable propositions that guide the experimental investigation into the effects of audio compression formats on classification accuracy and file size. By testing these hypotheses, researchers aim to validate the effectiveness of FLAC compression in preserving classification accuracy while achieving file size reduction, contributing to a better understanding of the trade-offs between audio quality and storage efficiency in audio compression.

3.3.1 Hypothesis 1

This hypothesis posits that FLAC, as a lossless compression format, will maintain the same classification accuracy as the original uncompressed audio format. It implies that compressing audio using FLAC should not result in any loss of quality, meaning the accuracy of classification models should remain unchanged compared to when using the original uncompressed audio data.

3.3.2 Hypothesis 2

This hypothesis suggests that FLAC compression will effectively reduce file size compared to the uncompressed audio format while maintaining the same classification accuracy. Additionally, it posits that FLAC can achieve this file size reduction without necessitating a significant increase in hardware costs, making it a practical and efficient storage solution. The hypothesis also considers that the reduced file size will save transmission time, further enhancing the overall efficiency of data handling.

3.4. INSTRUMENTS

This section outlines the hardware and software used in the study, detailing the specific operating environment and the reasons for selecting these tools. The rationale behind the choice of each software is also provided to ensure clarity. This information is intended to ensure transparency and to assist other researchers who may wish to replicate the experiment.

3.4.1 Hardware

- Processor: Intel(R) Core (TM) m3-6Y30 CPU @ 0.90GHz (4 CPUs), ~1.5GHz
- Memory: 4096MB RAM
- Storage: NTFS, 120GB
- DirectX Version: DirectX 12
- Bandwidth: Optical fiber, download speed**:** 309Mbps
- Operating System: Windows 10 64-bit (10.0, Build 19045)

3.4.2 Software

3.4.2.1. Open Research Data Platform: Zenodo

Zenodo is an open research data platform that facilitates the sharing, preservation, and citation of research outputs across all fields of science. Developed by CERN, the European Organization for Nuclear Research, Zenodo offers researchers a free and user-friendly platform to deposit and publish a wide range of research outputs, including datasets, software, images, videos, and more. It provides persistent identifiers (DOIs) for deposited content, ensuring its long-term accessibility. Researchers can also assign licenses to their data, enabling others to reuse and build upon their work while respecting intellectual property rights. Zenodo promotes open science principles by fostering collaboration, transparency, and reproducibility in research.

Open Source Beehives (OSBH) project, which aims to develop open-source beehive designs and monitoring tools to support beekeepers and research efforts to protect bee populations, has chosen Zenodo as a repository for sharing its research outputs, data, and related materials. By utilizing Zenodo, the OSBH project ensures that its work is openly accessible to the broader scientific community, contributing to collaboration and knowledge-sharing in the field of beekeeping and pollinator conservation.

3.4.2.2. Compress Tool: FFmpeg

FFmpeg is a powerful and versatile open-source software tool primarily used for handling multimedia data. It includes a wide range of functionalities related to audio, video, and other multimedia formats, making it a popular choice for professionals and enthusiasts alike. One of its key features is its ability to compress audio and video files efficiently while maintaining high quality. FFmpeg supports various codecs and formats, allowing users to transcode, encode, decode, and manipulate multimedia files with ease. Whether you need to resize, convert, or compress multimedia content, FFmpeg offers a comprehensive set of tools to meet your needs.

3.4.2.3. Runtime Platform: Python

Python is a versatile and user-friendly programming language renowned for its simplicity, readability, and extensive capabilities. With its clear and concise syntax, Python facilitates rapid development and is suitable for a wide range of applications, from web development and automation to scientific computing, artificial intelligence, and data analysis. Its extensive standard library and vast ecosystem of third-party packages provide developers with powerful tools to tackle complex problems efficiently. Python's cross-platform compatibility and strong community support make it a popular choice for beginners and experienced developers alike. Furthermore, its popularity continues to rise steadily, solidifying its position as one of the most widely used programming languages in the world.

3.4.2.4. Development Environment: Visual Studio

Visual Studio is a comprehensive integrated development environment (IDE) developed by Microsoft. It provides a powerful and feature-rich platform for software development across various programming languages and platforms, including but not limited to C#, C++, Visual Basic, Python, and JavaScript. Visual Studio offers a wide range of tools and features to streamline the development process, including code editing, debugging, testing, version control, and collaboration tools.

Visual Studio supports Jupyter notebooks, a popular tool for data exploration and analysis in Python. Developers can create, edit, and run Jupyter notebooks directly within Visual Studio, seamlessly integrating data analysis workflows into their Python development environment. These features enhance the versatility of Visual Studio for Python developers, supporting a wide range of use cases from software development to data science and analysis.

3.5. PARTICIPANTS

3.5.1 NU-Hive Dataset

The NU-Hive dataset is a comprehensive collection of audio recordings from the NU-Hive project (S. Cecchi, 2018), aimed at the automatic recognition of beehive sounds. These recordings originate from controlled, homogeneous environments primarily associated with two hives. Each audio segment is labelled according to the presence or absence of the queen bee. The dataset's recordings include external sounds such as traffic, car honks, and birds, providing a contrast to the bee-specific sounds. Approximately 60% of the annotated recordings are sourced from the NU-Hive dataset, representing data from the two hives.

3.5.2 Open Source Beehive (OSBH) Dataset

The OSBH dataset consists of audio recordings gathered from the Open Source Beehive project, which aims to develop a beehive monitoring system (Project., 15 December 2020). These recordings come from a citizen science initiative, with contributions from the public capturing sounds from their beehives. The dataset is highly diverse, reflecting different recording conditions, devices, hive environments, and microphone positions. The remaining recordings in the annotated dataset are from the OSBH dataset, featuring contributions from six different hives located in North America, Australia, and Europe.

3.5.3 BeeAudio Dataset

The BeeAudio Dataset is a significant initiative aimed at leveraging computational methods to address bee population decline by remotely and instantly detecting the health status of beehives through sound data analysis (Yang, 2022). This dataset comprises the largest single collection of bee audio recordings, collected using a custom IoT device that combines an ESP32 Wi-Fi module, an INMP441 microphone module, and a BME280 temperature/humidity sensor. All data is original, sourced from European Honeybee hives in California, with recordings divided into 60-second chunks. This provides a comprehensive resource for studying bee communication and behaviours. The BeeAudio Dataset consists of 7100 samples, contributing significantly to the diversity and richness of the annotated dataset for research purposes.

3.5.4 Sample Type and Size

Each dataset includes varying numbers of recordings of different lengths, contributing to a total duration of approximately 2 hours. The annotated dataset comprises time-labeled segments categorized as "Queen" or "noQueen," representing pure beehive sounds and periods with external sounds, respectively.

3.5.5 Reasons for the Number Selected

The selection of segment numbers for both the training and testing datasets was guided by several key considerations aimed at ensuring robust model training and evaluation. Maintaining a balanced ratio between the training and testing datasets is crucial for preventing bias and overfitting in the model. The chosen ratio of 3:1 for training to testing data ensures that the model is exposed to a diverse range of instances during training while still retaining a substantial portion for independent evaluation.

Each duration of the training dataset spans 90 minutes, while the testing dataset covers a duration of 30 minutes. By segmenting the audio recordings into 3-second segments, each dataset achieves a balanced distribution of segment types within their respective durations. The selected numbers of segments are statistically significant for conducting meaningful model training and evaluation, providing a sufficiently large sample size to capture the variability present in the audio recordings from both hive states.

Overall, the chosen numbers of segments for the training and testing datasets were determined to strike a balance between representation, balance, duration considerations, and statistical significance, thereby facilitating robust model development and assessment in the context of audio-based smart beekeeping.

3.5.6 Basis for Selection

Recordings within each dataset are selected based on their relevance to the research objectives, providing a representative sample of bee-related audio phenomena encountered in real-world beekeeping environments.

3.6. PROCEDURE AND TIMELINE

This section presents a comprehensive overview of the procedure and timeline followed in the study. The research was conducted in a series of well-defined stages, each building upon the previous one to ensure a structured and systematic approach.

1. Define research objectives. (1 month)
2. Literature & Algorithm Review (1 month)
3. Data Acquisition & Initial Analysis (1 month)
4. Feasibility Analysis (1 month)
5. Proposal Report and Defence (1 month)

6. Experiments (2 month)
7. Evaluation and Fine-tuning (1 month)
8. Results Analysis and Interpretation (1 month)
9. Report Writing and Finalization (2 month)

3.7. ETHICS AND LIMITATIONS

3.7.1 Ethical Considerations

This research adheres to ethical guidelines concerning the use of datasets and experimental procedures. All datasets used in this study were obtained from publicly available sources or with appropriate permissions and consent. The privacy and confidentiality of individuals associated with the datasets have been preserved, and no personally identifiable information is disclosed. Additionally, any potential biases in dataset collection, annotation, or usage have been carefully considered and addressed to ensure the integrity and fairness of the research outcomes.

3.7.2 Potential Problems and Limitations

Despite rigorous methodological approaches, several limitations and potential problems exist within this research. One limitation concerns the generalizability of the findings, as the datasets used may not fully represent the diversity of real-world beekeeping environments. Additionally, variations in recording conditions, equipment quality, and environmental factors may introduce noise and variability into the data, affecting the performance of the machine learning algorithms. Furthermore, the reliance on audio-based data for bee sound recognition may overlook other important indicators of hive health and behavior, necessitating complementary approaches for comprehensive hive monitoring.

3.7.3 Threats to Validity

Several factors pose potential threats to the validity of the results obtained in this study. These include selection bias in dataset compilation, model overfitting due to limited dataset size or complexity, and algorithmic biases inherent in machine learning methodologies. Additionally, the lack of standardized evaluation metrics for bee sound recognition tasks may impact the comparability of results across studies. To mitigate these threats, robust validation procedures, including cross-validation and independent testing, have been employed to ensure the reliability and validity of the research findings.

Chapter 4. Methodology

The methodology section of this research paper outlines the approach taken to bridge the identified research gap of "Audio Compression without Feature Loss." Firstly, the methodology involves an in-depth exploration and understanding of the FLAC compression technique, focusing on its ability to minimize feature loss in audio data. FLAC is selected as the target compression format due to its lossless nature, which aims to preserve all audio features while reducing file size. Secondly, the evaluation of compression techniques provides empirical evidence regarding the effectiveness of these methods in preserving audio features during compression. Mel-Frequency Cepstral Coefficients combined with Support Vector Machines are selected as the approach to demonstrate the suitability of FLAC as a compression format. The rationale behind this approach lies in the effectiveness of MFCC in capturing essential audio features, making it suitable for tasks such as audio classification. By extracting MFCC features from audio data compressed in different formats, this study aims to evaluate the impact of compression techniques on the preservation of crucial audio characteristics. Support Vector Machine is a robust supervised machine learning algorithm widely used for both classification and regression tasks.

4.1. FREE LOSSLESS AUDIO CODEC

FLAC, which stands for Free Lossless Audio Codec, is an audio compression codec that is known for its ability to compress audio files without any loss of quality. Here's a detailed overview of FLAC: It employs a process of linear prediction, where it analyzes the audio signal and predicts subsequent samples based on previous ones. The differences between these predictions and the actual samples, known as residuals, are then calculated. These residuals typically have smaller values and less complexity than the original audio data, making them easier to compress. FLAC encodes these residuals using a method such as Rice coding, which is particularly effective for sequences with smaller numerical values. Along with the encoded residuals, FLAC also stores essential metadata about the audio, like the sample rate and number of channels. During decompression, FLAC reverses this process, using the stored prediction model and residuals to reconstruct the audio signal precisely, ensuring that the output is a perfect match to the original uncompressed audio. As a result, FLAC provides an ideal solution for high-quality audio storage and playback where maintaining the original sound's integrity is crucial. FLAC provides lossless compression, meaning that the original audio data can be perfectly reconstructed from the compressed file. This is crucial for applications where audio quality is paramount.

While not as compact as lossy formats like MP3 or AAC, FLAC typically reduces file sizes to about 50-70% of their original size, depending on the source material. This is quite efficient for a lossless format. Since FLAC is lossless, it preserves the full fidelity of the original audio signal. This makes it ideal for high-quality audio storage, archiving, and playback. FLAC is designed to be computationally efficient for both encoding (compression) and decoding (decompression). This makes it suitable for real-time applications.

How FLAC Works

4.1.1 Encoding Process

FLAC (Free Lossless Audio Codec) stands out for its ability to compress audio data without compromising its original fidelity. This process is meticulously orchestrated through a series of steps. First, the input audio signal is segmented into fixed-size blocks or frames, typically around 4096 samples each, through a process called blocking. Subsequently, linear predictive coding (LPC) is applied to each block, where past audio samples are utilized to forecast future samples. This prediction process employs various LPC models, ranging from simple fixed-order models to more complex ones, often tailored to the characteristics of the audio signal. Once the predicted samples are determined, the residual—essentially the difference between the actual and predicted samples—is calculated. This residual typically exhibits a smaller dynamic range than the original audio signal, rendering it more amenable to compression. The next step, quantization, involves representing the residual signal with a finite number of bits, a process facilitated by Rice coding, which is particularly effective for signals with a limited dynamic range. Following quantization, entropy coding, specifically Rice coding, is employed to further compress the quantized residuals by exploiting their statistical properties. This brief process is shown as Figure 1. This compression method takes advantage of the fact that the residuals tend to follow certain statistical distributions, thereby reducing redundancy and optimizing storage efficiency. Finally, metadata such as tags, cover art, and additional information can be embedded into the encoded audio blocks to enrich the final FLAC file. This meticulous encoding process ensures that the resulting audio file maintains its original quality while significantly reducing its size, making it an ideal choice for applications where preserving audio fidelity is paramount, such as in professional audio production and archival storage (van Beurden M. Q., Aug. 21, 2022).

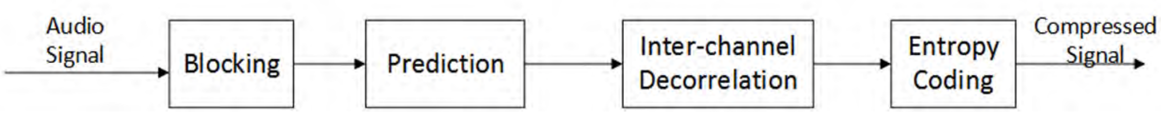

Figure 1. Structure of FLAC encoder; the paper "A review of lossless audio compression standards and algorithms" (Muin, 2017) highlights that many lossless audio compression methods share a similar theoretical basis, often employing Linear Prediction Coding (LPC) as a core technique. FLAC (Free Lossless Audio Codec) is a notable example that utilizes LPC.

4.1.2 Decoding Process

The decoding process for FLAC mirrors the encoding process in its meticulousness, ensuring that the audio can be faithfully reconstructed without any loss of information. This process involves several key steps, beginning with reading the FLAC file, which contains both the encoded audio blocks and associated metadata. Entropy decoding follows, wherein the residuals, originally compressed using Rice coding during encoding, are decoded back to their quantized form. Subsequently, inverse quantization reverses the quantization process, restoring the quantized residuals to their original form. Prediction inversion then comes into play, utilizing the LPC coefficients and residuals to reconstruct the original audio samples precisely. This step involves combining the predicted values with the residuals to recreate the exact original samples. The reconstructed audio blocks are then recombined to form the continuous audio signal through re-blocking. Ultimately, the decoding process yields a final output—an audio stream that is bit-for-bit identical to the original input—ensuring that no information is lost throughout the compression and decompression procedures (van Beurden M. Q., 2022 Aug. 21).

4.1.3 Validation of FLAC Performance

The section objective is to visually compare the waveforms of audio files in different compression formats. This comparison aims to reveal the effects of various compression techniques on the audio signal's amplitude over time. The formats being compared are WAV (an uncompressed format), MP3 (a lossy compression format), and FLAC (a lossless compression format).

The waveform comparison provides insights into the effects of bee sound compression. The waveform curves in Figure 2, Figure 3 and Figure 4 demonstrated that while FLAC and

WAV formats maintain high audio fidelity, the MP3 format, due to its lossy compression, results in a loss of audio detail. This is particularly evident in the finer aspects of the waveform, where MP3 fails to replicate the exact amplitude variations present in the original uncompressed audio. For AI analysis where audio quality is paramount, FLAC emerges as a suitable alternative to WAV, offering both compression and preservation of audio quality.

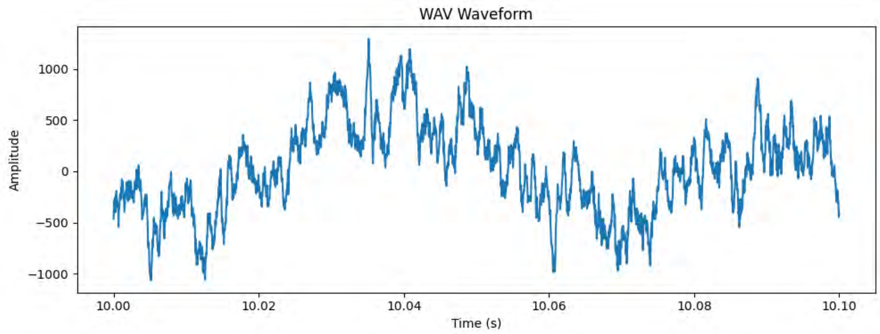

Figure 2. WAV Waveform is drawn by python library scipy.io. This results in higher quality and more detailed waveforms.

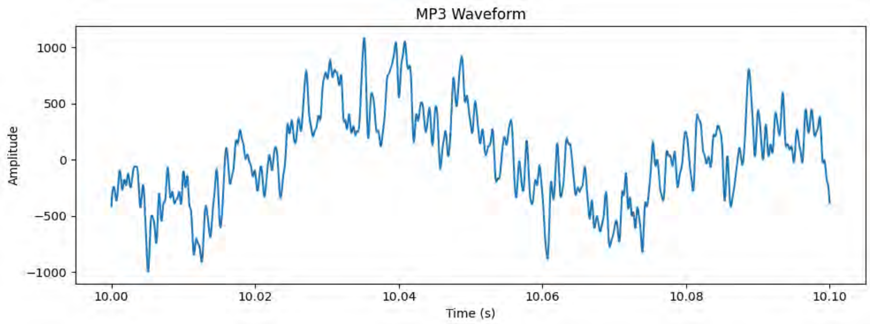

Figure 3. MP3 Waveform is drawn by python library scipy.io. The MP3 waveform may appear less detailed due to the lossy compression.

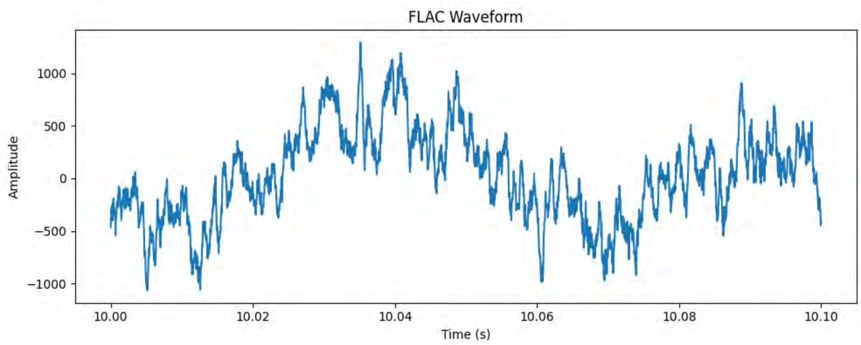

Figure 4. FLAC Waveform is drawn by python library scipy.io. Because FLAC preserves all the original audio data, its waveform is identical to that of a WAV file.

In finalizing our decision for a compression method in beekeeping research, FLAC stood out as the ideal choice. Its attributes uniquely align with our requirements. As an open-source codec, FLAC offers cost-effective accessibility. Its lossless nature is crucial, ensuring that intricate bee sounds, particularly those from the queen, are preserved without any quality degradation. This aspect is vital for accurate data analysis. FLAC's specialization in audio compression means it's adept at handling the specific nuances of bee audio data efficiently. Importantly, FLAC's compression is specific and not bundled with archiving functionalities, meaning it focuses solely on reducing audio file sizes effectively while maintaining the original data integrity. Thus, FLAC stands out as the most suitable audio compression method for our research, balancing cost-effectiveness, fidelity, and efficiency in data handling.

4.2. MEL FREQUENCY CEPSTRAL COEFFICIENTS

Mel Frequency Cepstral Coefficients are a fundamental feature extraction technique in the field of audio signal processing, particularly in tasks related to speech recognition, sound classification, and acoustic analysis. MFCCs are inspired by the human auditory system's perception of sound and are designed to capture the essential spectral characteristics of an audio signal. The process of extracting MFCCs involves several key steps as shown Figure 5.

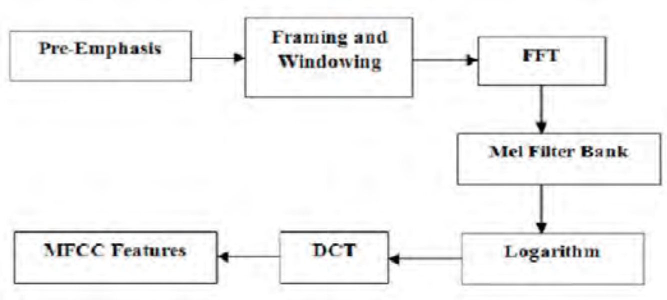

Figure 5. Study of MFCC Feature Extraction Methods with Probabilistic Acoustic Models for Speaker Biometric Applications - Scientific Figure on ResearchGate (A, S., Thomas, A., & Mathew, D., 2018). Available from: https://www.researchgate.net/figure/Block-diagram-of-MFCC-Feature-extraction_fig1_329046751.

Pre-Emphasis

The pre-emphasis step is the initial stage in the calculation of Mel Frequency Cepstral Coefficients (MFCCs). Its primary purpose is to amplify the high-frequency components of an audio signal and mitigate the effects of high-frequency noise. This step is crucial for improving the signal's spectral characteristics and enhancing its overall signal-to-noise ratio, making it more suitable for subsequent analysis, such as feature extraction.

The relationship between Pre-Emphasis and Analog-to-Digital Conversion (ADC) is integral to understanding this process. ADC is the process by which the continuous analog audio signal, like human speech or any other sound, is converted into a digital representation consisting of discrete numerical values. This digital representation, often referred to as the digital audio signal, is obtained through two key steps: sampling and quantization.

The fundamental idea behind pre-emphasis is to emphasize the importance of high-frequency information in the audio signal, as this information often contains valuable spectral details related to speech or other audio events. By boosting the high-frequency components, the pre-emphasis filter makes these details more prominent and less susceptible to being masked by noise or other unwanted artifacts.

Framing

Framing is a critical step in the feature extraction process for audio signals, including the calculation of Mel Frequency Cepstral Coefficients (MFCCs). It involves dividing the continuous audio signal into smaller, overlapping frames or segments. This segmentation allows us to analyze the audio signal in a localized manner, capturing its spectral characteristics over short time intervals. The framing process involves partitioning the continuous audio signal into overlapping or non-overlapping frames of fixed duration. Mathematically, this is achieved using the following equation:

$$N = \text{Frame Duration} \times \text{Sampling Rate}$$

During framing, you can choose to overlap consecutive frames by a certain percentage (usually 50% overlap is common). The overlap facilitates smoother transitions between frames and provides additional temporal context for feature extraction.

Windowing

Windowing is a critical step in audio signal feature extraction, particularly in the calculation of Mel Frequency Cepstral Coefficients (MFCCs). Following the segmentation of the audio signal into frames, each frame undergoes element-wise multiplication with a chosen window function. This process serves several vital purposes. Firstly, it gently tapers the edges of each frame to zero, mitigating abrupt discontinuities at frame boundaries that might introduce unwanted spectral artifacts during subsequent analysis, such as Fourier Transforms. Secondly, it facilitates smoothing by reducing the impact of sudden signal changes near frame boundaries, ensuring continuity, and minimizing spectral leakage. Additionally, windowing promotes the assumption of stationarity within each frame, a crucial requirement for many spectral analysis techniques. Lastly, windowing allows customization of the frame's spectral content, enabling a balance between frequency and temporal resolution, with different window functions offering various trade-offs tailored to specific analysis requirements.

Fourier Transform

The Fast Fourier Transform (FFT) is a powerful mathematical algorithm that bridges the time and frequency domains in signal processing. It enables the efficient conversion of a time-domain signal into its frequency-domain representation. Let's explain the FFT in the context of both domains: In the time domain, a signal is represented as a sequence of amplitude values over discrete time intervals. Each data point corresponds to the signal's amplitude at a specific point in time. Consider an example of an audio waveform, wherein the time domain, you have a series of amplitude values sampled at regular intervals, representing the sound wave's behavior over time. The frequency domain represents the same signal in terms of its frequency components. It provides information about which frequencies are present in the signal and the magnitude (amplitude) and phase of those frequencies. When you apply the FFT to the time-domain signal, it transforms the signal into the frequency domain. The output of the FFT is a spectrum that shows the amplitude of each frequency component present in the signal. In the frequency domain, you can identify dominant frequencies, harmonics, and noise components. It provides a more detailed view of the signal's spectral characteristics. Time-Domain Input starts with a time-domain signal, such as an audio waveform or any time-varying signal. When you apply the FFT algorithm to the time-domain signal, it efficiently calculates the Discrete Fourier Transform (DFT) values. The result is a set of complex numbers representing the signal's frequency components. The output of the FFT is a spectrum that shows the amplitude and phase of each frequency component. This spectrum provides a detailed view of the signal's frequency composition.

Mel Filterbank

The Mel Filterbank is an essential component in audio signal processing, particularly in the calculation of Mel Frequency Cepstral Coefficients (MFCCs). It serves to transform the linear frequency representation of an audio signal into the perceptually motivated Mel scale. Achieved through a series of triangular filters, the Mel Filterbank emphasizes frequencies relevant to human auditory perception while discarding less critical information. These filters, varying in width along the Mel scale, discretize the entire frequency spectrum, and their outputs provide a weighted representation of the signal's spectral content. By logarithmically scaling these filterbank outputs and applying the Discrete Cosine Transform (DCT), MFCCs are derived, capturing the essential spectral features of audio signals. This technique finds widespread application in speech recognition, audio classification, and speaker identification, enabling efficient and informative analysis of audio data.

Logarithm

After obtaining the filterbank outputs, the next step involves applying the logarithm (typically the natural logarithm, ln) to these values. This logarithmic scaling serves several essential purposes in MFCC computation. Human perception of loudness and pitch is roughly logarithmic. Logarithmic scaling aligns the representation with human auditory perception, making the MFCCs more perceptually relevant. Logarithmic scaling compresses the dynamic range of the filterbank outputs, reducing the influence of large energy variations and making the MFCCs more robust to noise and intensity differences. Logarithmic scaling reduces the dimensionality of the feature vector. This compact representation is computationally efficient and often leads to improved model performance.

Discrete Cosine Transform (DCT)

Finally, the Discrete Cosine Transform (DCT) is applied to the log-filterbank energies. The resulting coefficients, known as MFCCs, capture the spectral characteristics of the audio signal while reducing redundancy. The first few coefficients often contain the most relevant information and are used as features in subsequent analyses. MFCCs offer several advantages for audio analysis. They compactly represent the spectral content of audio signals, reducing dimensionality while retaining essential information. Additionally, they are robust to variations in speech or sound duration and have been widely used in speech recognition and audio classification tasks due to their effectiveness in capturing acoustic features. In the following sections, we will explore the application of MFCCs in the context of our research on bee-related audio data, highlighting their role in characterizing queen bee and worker bee sounds.

4.3. SUPPORT VECTOR MACHINE

Support Vector Machine (SVM) is a robust supervised machine learning algorithm widely used for both classification and regression tasks. It is known for its effectiveness in high-dimensional spaces and its ability to handle complex relationships between input features.

Objective

The primary objective of SVM is to identify the optimal hyperplane that best separates different classes in the feature space. This hyperplane serves as the decision boundary that discriminates between data points of varying classes. The key is to select a hyperplane that not only separates the classes but does so with the maximum possible margin.

Margin

The margin in SVM is defined as the distance between the hyperplane and the nearest data points from each class, which are known as support vectors. SVM aims to maximize this margin, as a larger margin is associated with better generalization capabilities and improved performance on unseen data. The optimization process focuses on finding the hyperplane that provides the maximum margin, thereby enhancing the classifier's robustness.

Support Vectors

Support vectors are the critical data points that are closest to the hyperplane. These points lie on the edge of the margin and play a pivotal role in determining the position and orientation of the hyperplane. Interestingly, only a subset of the training data, specifically the support vectors, influences the construction of the decision boundary in SVM. This characteristic makes SVM particularly efficient, as it relies on these few critical points rather than the entire dataset (shown as Figure 6).

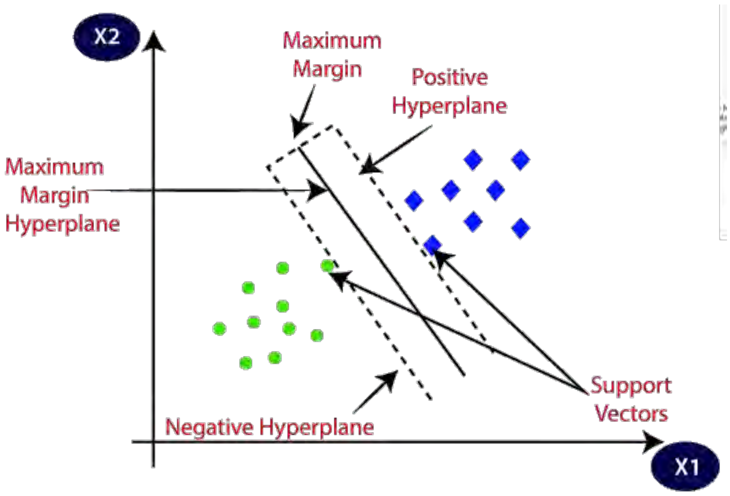

Figure 6. Support Vector Machine Theory (JavaTPoint., 2021). suitable as a supervised learning algorithm.

Kernel Trick

SVMs are capable of handling non-linearly separable data using the kernel trick. Instead of explicitly mapping the input data into a higher-dimensional space, the kernel function implicitly performs this transformation by computing the dot product of data points in the feature space. This approach allows SVM to capture complex patterns without the computational burden of explicit transformation. Common kernel functions include linear, polynomial, radial basis function (RBF), and sigmoid kernels.

Regularization Parameter (C)

The regularization parameter (C) in SVM controls the trade-off between maximizing the margin and minimizing classification errors. A smaller value of C allows for a larger margin but tolerates more misclassified points, promoting a simpler model. Conversely, a larger value of C aims to minimize misclassification, resulting in a narrower margin and a more complex model. Tuning this parameter is crucial for achieving the desired balance between bias and variance.

SVM Kernels

Linear Kernel is suitable for linearly separable data, where the decision boundary is a straight line. This kernel is computationally efficient and works well when the data can be separated with a linear hyperplane. Radial Basis Function (RBF) Kernel is suitable for non-linearly separable data, where the decision boundary is more flexible and can adapt to complex patterns. The RBF kernel can handle situations where the relationship between class labels and attributes is non-linear, making it a powerful tool for capturing intricate data structures.

Training

Training an SVM involves optimizing the hyperplane parameters, namely the weights and bias, to minimize the classification error while maximizing the margin. This optimization is framed as a convex optimization problem, typically solved using techniques such as Sequential Minimal Optimization (SMO) or gradient descent methods. The training process entails selecting a kernel function, setting the regularization parameter, and solving the optimization problem.

Chapter 5. Experiment

In this experiment, we conduct a thorough investigation into audio classification methodologies, covering key aspects from data acquisition to classifier implementation. Our process begins with the collection of audio data from open platforms, ensuring a balanced representation across categories and employing compression techniques for efficient storage. Subsequently, we extract essential features from the audio signals, focusing on Mel-frequency cepstral coefficients (MFCCs) known for their effectiveness in capturing spectral characteristics. Finally, we implement a classifier using Support Vector Machine (SVM) algorithms, leveraging the extracted MFCC features to train and evaluate the model's performance. This experiment (steps shown as Figure 7) aims to provide insights into the efficacy of audio classification techniques and their applicability in real-world scenarios.

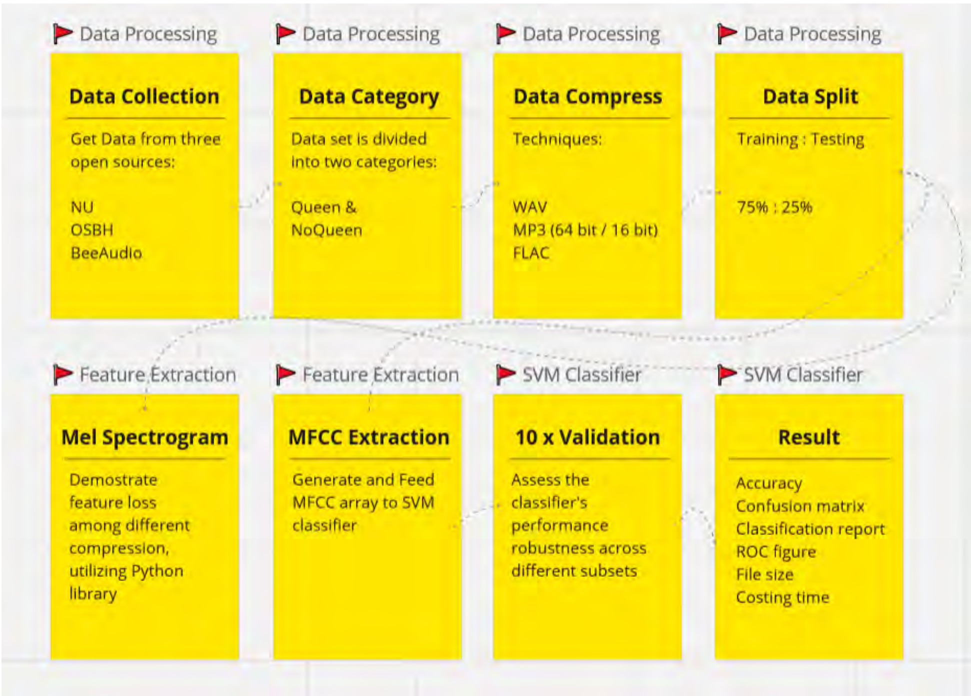

Figure 7. Experiment Steps include three phases as data processing, feature extraction and SVM classifier.

5.1. AUDIO DATA

This research focuses on utilizing audio data for acoustic research purposes. Bee sound serves as the primary subject due to its relevance to study. However, it's important to note that bee sound recordings typically contain a small amount of data, posing a limitation to the depth of analysis. One valuable resource for acquiring bee sound data is the Open Source Beehives (OSBH) project (Project., 15 December 2020). This initiative gathers bee sound recordings, including queen sound, from various beehives belonging to beekeepers worldwide. These recordings offer a diverse range of data, reflecting variations among beehives, locations, and equipment setups.

The majority of the bee sound recordings adhere to standard specifications, typically captured at a 44.1 kHz sampling rate, with 16-bit resolution, and either in stereo or mono format. Furthermore, to preserve the integrity of the data, recordings are stored in a lossless format, such as Microsoft WAV, chosen for their lossless compression. This preservation maintains the original audio quality, serving as reliable benchmarks for comparison.

Researchers interested in accessing bee sound data for their studies can explore the OSBH project's repository, which is readily available for research purposes on the Zenodo open data sharing platform. This comprehensive dataset provides a valuable resource for acoustic research in the field of bee behavior and ecology.

The audio data undergoes segmentation into two categories: queen and non-queen bee sound, which is shown as Table 1. This segmentation ensures a balanced representation of both categories in the dataset.

Segment Type	Categories	Segment Length
Training and Testing	Queen, No Queen	3 seconds

Table 1. Training Category. Training and testing data maintaining a ratio of 3:1.

For the training dataset, which spans 90 minutes in total, each segment type (queen and no queen) comprises 1835 segments. Similarly, for the testing dataset, with a duration of 30 minutes, each segment type contains 616 segments. This preprocessing strategy ensures that both categories are adequately represented in the training and testing datasets, facilitating robust model training and evaluation.

5.2. COMPRESSION

5.2.1 MP3 Compression

To demonstrate the performance of FLAC (Free Lossless Audio Codec) compression, we conduct a comparative analysis with MP3 (MPEG Audio Layer III), a widely used lossy audio compression format. The goal is to evaluate the impact of compression on audio quality and file size. Using FFmpeg, we compress the original WAV files to MP3 at two different bitrates: 64 kbps and 16 kbps.

Pseudocode

```
Python\\libs\\ffmpeg\\bin\\ffmpeg.exe -file Stanford_Queen_Training_10mins.wav -bitrate 64k Stanford_Queen_Training_10mins_64K.mp3
```

5.2.2 FLAC Compression

In our experiments on audio file conversion for beehive monitoring data, we developed a Python script utilizing the pydub library. The script's primary function, convert_to_flac, efficiently converts WAV audio file formats to the FLAC format, ensuring lossless compression and high-quality audio preservation. The script begins by importing the necessary AudioSegment module from the pydub library and the time module to potentially track the conversion process duration. The core function, convert_to_flac, takes two arguments: source_path and target_path.

Pseudocode

```
FUNCTION convert_to_flac(source_path, target_path):
audio = LOAD_AUDIO_FROM_FILE(source_path)
    // This function loads an audio file from the specified source path using the pydub library.
    // Parameters:
    // source_path: A string representing the file path of the audio file to be loaded.
    // Returns:
    // An AudioSegment object representing the loaded audio.
    audio.export(target_path, format="flac")
```

5.3. MQTT TRANSMISSION

Using Java code to set up an MQTT client, connects to a secure MQTT broker, and publishes parts of an audio file to a specific topic. Each part of the audio file is sent as a separate MQTT message, allowing for efficient and manageable transmission of large audio data. This approach is useful for applications such as beehive monitoring, where continuous and reliable data transmission is essential.

Pseudocode

```
FUNCTION main():
    SET brokerUrl TO "ssl://broker_address"
    SET username TO "your_username"
    SET password TO "your_password"
    SET topic TO "your_topic"
    SET filePath TO "path_to_audio_file"

    SET client TO NEW MqttClient(brokerUrl, GENERATE_CLIENT_ID())
```

```
    // Set up MQTT connection options
    SET options TO NEW MqttConnectOptions()
    options.setUserName(username)
    options.setPassword(password)

    // Connect to MQTT broker
    client.connect(options)
  // Publish part
    client.publish(topic, NEW MqttMessage(partContent))
   PRINT "Part published successfully in", seconds"
   client.disconnect()
```

By systematically measuring and comparing the compression time and transmission time of WAV, MP3, and FLAC formats, we can determine the most appropriate audio format for beehive monitoring. This comprehensive evaluation not only optimizes the technical performance of the monitoring system but also provides deeper insights into the practical aspects of audio data management. Such an approach enriches the research by addressing the critical factors influencing the efficiency and effectiveness of beehive monitoring solutions.

5.4. FEATURE EXTRACTION

MFCC captures this characteristic by dividing the audio spectrum into mel-frequency bands, which are spaced according to the human perception of sound. For each mel-frequency band, MFCC calculates the cepstral coefficients, representing the magnitude of each frequency component. To extract MFCC features from audio segments, we utilize the librosa library in Python, specifically the librosa.feature.mfcc function. The number of MFCC coefficients is set to 13. By invoking this function with the appropriate parameters, we obtain a matrix of MFCC coefficients representing the spectral characteristics of the audio segment. These coefficients serve as feature vectors for further analysis and classification in our research.

Pseudocode

```
function visualize_mel_spectrogram(audio_file, title):
    # Load audio file
    audio_data, sampling_rate = load_audio(audio_file)
    # Calculate Mel spectrogram
    mel_spectrogram = calculate_mel_spectrogram(audio_data, sampling_rate)
    log_mel_spectrogram = convert_to_log_scale(mel_spectrogram)


    # Plot Mel spectrogram
            plot_mel_spectrogram(log_mel_spectrogram, sampling_rate, title)
```

In the feature extraction phase, four audio files were meticulously chosen, each representing distinct formats: WAV, MP3 at 64Kbps, MP3 at 16Kbps, and FLAC. Leveraging the powerful capabilities of Librosa's librosa.feature.melspectrogram function, Mel spectrograms shown as Figure 8, Figure 9, Figure 10 and Figure 11 were computed for each audio file, encapsulating their frequency distributions over time. This comprehensive approach enabled a nuanced comparison of spectrogram characteristics among different formats, shedding light on potential distortions induced by compression algorithms and format-specific encoding schemes. Through visual inspection and quantitative analysis, discernible discrepancies in frequency content and temporal patterns were meticulously scrutinized, offering valuable insights into the fidelity and robustness of Mel spectrograms across diverse audio formats.

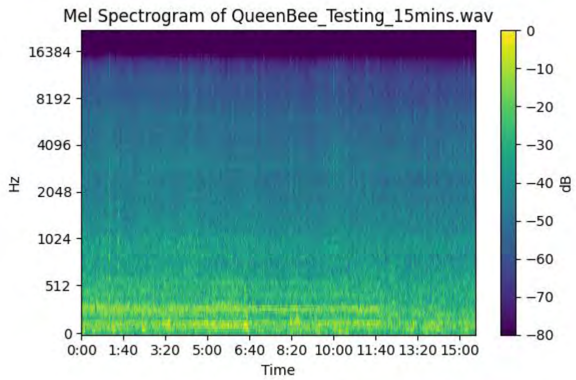

Figure 8. The WAV spectrogram shows a consistent presence of lower frequencies (below 512 Hz) with higher amplitudes (shown in yellow and green), and higher frequencies (above 4096 Hz) with lower amplitudes (shown in blue and purple).

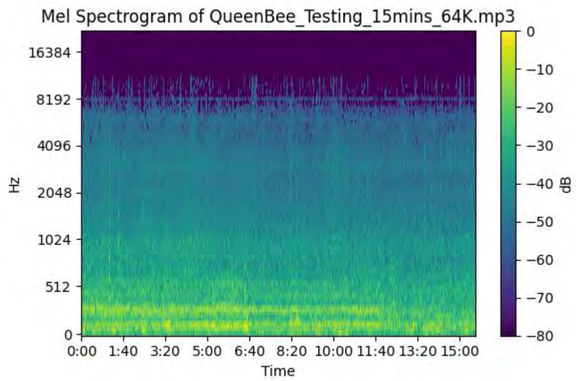

Figure 9. The mp3 spectrogram at 64 kbps shows a similar pattern to the wav file, but with more noticeable artifacts (visible as irregular patterns) in the higher frequencies. This is common for lower bitrate mp3 files, which use more aggressive compression.

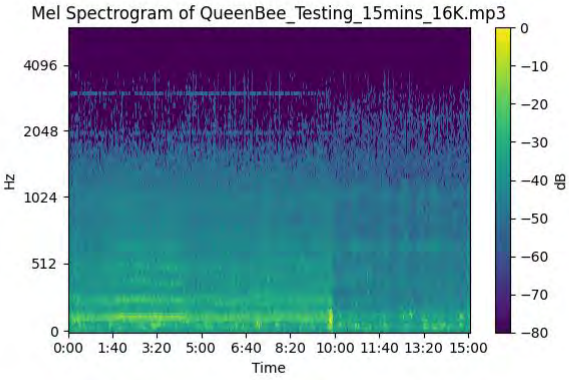

Figure 10. The MP3 spectrogram at 16 kbps shows significant loss of information in the higher frequencies (above 4096 Hz), and the lower frequencies are also less defined. This indicates heavy compression and loss of audio quality.

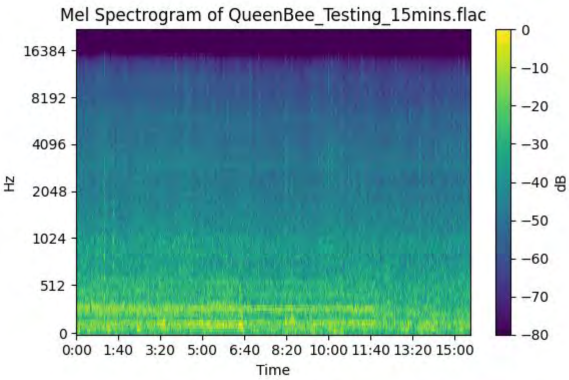

Figure 11. The FLAC spectrogram shows a similar pattern to the WAV file, with well-defined lower frequencies and consistent higher frequencies. FLAC is a lossless format, which means it retains all the audio information without compression artifacts.

After generated spectrogram, we describe the process of generating Mel-Frequency Cepstral Coefficients from audio data segments, which are subsequently used as features for training and testing an SVM classifier in the smart beekeeping system. Initially, the audio data is divided into smaller segments of a fixed length of 3 seconds, making it easier to process and analyse. This is done separately for the "queen" and "no queen" data. To ensure uniform segment length, shorter segments are padded with zeros. This standardization is necessary for consistent feature extraction. MFCCs are then extracted from each segment; these coefficients provide a compact and effective representation of the audio signal's spectral properties. The MFCCs are resampled to a uniform number of columns to match the shortest segment, ensuring that all feature vectors have the same dimensionality, which is crucial for the SVM classifier.

Each segment is labelled appropriately: segments from the "queen" data are labelled as 1, and those from the "no queen" data are labelled as 0. Below is the pseudocode code used to perform the above steps.

Pseudocode

```
# Segment the data into fixed-length frames
queen_testing_data_segments = segment(queen_testing_data, segment_length_frames)
no_queen_testing_data_segments = segment(no_queen_testing_data, segment_length_frames)
# Padding segments to ensure uniform length
max_length_queen = find_max_length(queen_testing_data_segments)
queen_testing_data_segments_padded = pad_segments(queen_testing_data_segments, max_length_queen)

max_length_no_queen = find_max_length(no_queen_testing_data_segments)
no_queen_testing_data_segments_padded = pad_segments(no_queen_testing_data_segments, max_length_no_queen)

# Initialize lists for MFCCs and labels
initialize empty lists: queen_testing_mfccs, queen_testing_labels
initialize empty lists: no_queen_testing_mfccs, no_queen_testing_labels

# Define the target number of columns (minimum segment length)
target_number_of_columns = min_length

# Extract MFCCs and resample them to uniform length
for each segment in queen_testing_data_segments_padded:
    queen_mfcc = compute_mfcc(segment, queen_sampling_rate, 13)
```

```
    queen_testing_mfcc_resampled = resample_mfcc(queen_mfcc, target_number_of_columns)
    append(queen_testing_mfccs, queen_testing_mfcc_resampled)
    append(queen_testing_labels, 1)  # Label for queen

  for each segment in no_queen_testing_data_segments_padded:
    no_queen_mfcc = compute_mfcc(segment, queen_sampling_rate, 13)
    no_queen_testing_mfcc_resampled = resample_mfcc(no_queen_mfcc, target_number_of_columns)
    append(no_queen_testing_mfccs, no_queen_testing_mfcc_resampled)
    append(no_queen_testing_labels, 0)  # Label for no queen
```

5.5. SUPPORT VECTOR MACHINE CLASSIFIER

Support Vector Machine (SVM) was chosen as the classifier for this classification task due to its effectiveness in handling high-dimensional data and its ability to find optimal decision boundaries. The Mel-frequency cepstral coefficients (MFCC) extracted from the audio data, along with their corresponding labels, were prepared for model training. MFCCs provide a compact representation of the spectral features, which are essential for classification tasks. The Python library scikit-learn was utilized for implementing SVM. The classifier model was initialized using svm.SVC, allowing for customization of kernel functions and other hyperparameters.

Pseudocode

```
  function process_audio_files()
    # Load and process training data
    # Initialize 10-fold cross-validation
    kfold = initialize_kfold(n_splits=10, shuffle=True, random_state=42)

    # Train the classifier
    clf = initialize_svm_classifier(kernel='linear')
```

```
# Lists to store performance metrics
accuracy_scores = []

# Loop over the folds
for train_index, val_index in kfold.split(X_train, y_train):
    X_fold_train, X_fold_val = split_training_data(X_train, train_index, val_index)
    y_fold_train, y_fold_val = split_label_data(y_train, train_index, val_index)

    # Train the classifier on the training fold
    clf.fit(X_fold_train, y_fold_train)
    # Make predictions on the validation fold
    y_pred_fold_val = clf.predict(X_fold_val)

    # Calculate accuracy and store it
    accuracy_fold_val = calculate_accuracy(y_fold_val, y_pred_fold_val)
    accuracy_scores.append(accuracy_fold_val)
```

A 10-fold cross-validation strategy was employed to assess the classifier's performance robustness across different subsets of the training data. The training data were divided into 10 folds, ensuring each fold contains a balanced representation of both classes for training and validation. The classifier was trained on each training fold and validated on the corresponding validation fold to evaluate its generalization performance. After iterating over all folds, the average accuracy across all folds is calculated by summing up the accuracy scores and dividing by the total number of folds. The average accuracy is printed to the console, providing an overall assessment of the classifier's performance across the 10 folds.

Following the training phase, the trained Support Vector Machine (SVM) classifier was evaluated on the testing dataset to assess its performance on unseen data. The trained SVM classifier was utilized to make predictions on the testing dataset, consisting of previously unseen audio samples. The classifier leveraged the extracted Mel-frequency cepstral coefficients (MFCC) features to predict the class labels of the testing samples. This evaluation focuses solely on the practical aspects of the experiment, detailing the steps involved in testing the SVM classifier and evaluating its performance using various classification metrics. A confusion matrix was generated to visualize the distribution of true positive, true negative, false positive, and false negative predictions. This matrix provided insights into the classifier's ability to correctly classify samples belonging to different classes and identify any potential misclassifications.

Chapter 6. Results

In this section, we present the outcomes of our classification experiments aimed at distinguishing between queen and non-queen bee sounds using Support Vector Machine (SVM) models with 10-fold cross-validation. To ensure robustness and generalizability, we leveraged three distinct data sources: the NU project, the OSBH project, and data processed at Stanford University. The audio data was carefully divided and categorized into queen and non-queen bee sounds, ensuring balanced representation in both the training and testing datasets. This preprocessing strategy aimed to facilitate robust model training and evaluation by providing a comprehensive and representative sample of bee sounds. Building upon the prepared dataset, we now present the outcomes of our classification experiments using a Support Vector Machine (SVM) with 10-fold cross-validation.

6.1. NU DATASET RESULT

The NU-Hive project was developed by researchers Inês Nolasco and Emmanouil Benetos (Nolasco & Benetos, 2018.). Their work focuses on the automatic recognition of beehive sounds. Specifically, the training dataset comprised 929 queen training segments and 924 non-queen training segments, while the testing dataset included 314 queen testing segments and 302 non-queen testing segments (shown as Table 2).

Data Type	Number of Segments
Queen Training	929
No Queen Training	924
Queen Testing	314
No Queen Testing	302

Table 2. NU Data Set Segments. The dataset is divided into four distinct categories based on the presence of the queen bee and the purpose of the data (training or testing).

The following table 3 provides detailed accuracy percentages and corresponding file sizes for different audio formats represented by various bitrates.

Bitrate	10-Fold Validation Accuracy	Testing Accuracy	File Size (Bytes)
Uncompressed (WAV)	98.97%	90.91%	325,105,800
MP3 64 kpbs	98.22%	85.88%	22,167,345
MP3 16 kpbs	98.60%	82.63%	11,083,799
FLAC	98.97%	90.91%	125,039,638

Table 3. NU Classification Accuracy and File Size. This result is generated by 10-Fold validation to increase robustness.

A confusion matrix in Table 4 is a table that is often used to evaluate the performance of a classification algorithm. It presents a detailed breakdown of correct and incorrect classifications made by the model, providing insights into its effectiveness.

Audio Format	True Positives (TP)	True Negatives (TN)	False Positives (FP)	False Negatives (FN)
WAV	313	247	55	1
MP3 - 64K	313	216	86	1
MP3 - 16K	313	196	106	1
FLAC	313	247	55	1

Table 4. NU Confusion Matrix. Analyse the four formats performance of the audio formats based on the confusion matrices provided.

The classification report serves as a comprehensive evaluation of the performance of our classification model in distinguishing between different classes within our dataset. The classification report in Table 5 and 6 outlines the precision, recall, and F1-score for each class, shedding light on the model's efficacy in correctly predicting instances of each category while minimizing false classifications.

Audio Format	Precision	Recall	F1-Score	Support
WAV	1.00	0.79	0.88	302
MP3 - 64K	1.00	0.72	0.83	302
MP3 - 16K	1.00	0.65	0.79	302
FLAC	1.00	0.79	0.88	302

Table 5. Classification report of No Queen (Precision, Recall, F1-Score, and Support) for different audio formats (WAV, MP3 - 64K, MP3 - 16K, and FLAC) in the NU dataset.

Audio Format	Precision	Recall	F1-Score	Support
WAV	0.83	1.00	0.91	314
MP3 - 64K	0.78	1.00	0.88	314
MP3 - 16K	0.75	1.00	0.86	314
FLAC	0.83	1.00	0.91	314

Table 6. Queen classification report showing precision, recall, and F1-score for different audio formats (WAV, MP3 - 64K, MP3 - 16K, and FLAC).

The Receiver Operating Characteristic (ROC) curve is a fundamental tool in assessing the performance of classification models, particularly in scenarios involving binary classification tasks. In our research, the ROC curve serves as a critical component in evaluating the effectiveness of our classification model in distinguishing between different classes of audio data (shown as Figure 12, 13, 14 and 15), specifically targeting the discrimination between "Queen" and "No Queen" instances. As shown in Figure 12, the ROC curve for the WAV format rises steeply and then levels off near the top. This indicates that the classifier can achieve a high true positive rate (sensitivity) with a low false positive rate early in the process. In Figure 13, the ROC curve for the MP3 64K format initially rises almost vertically before flattening out. The small horizontal portion at the start of the curve indicates that achieving additional true positives becomes increasingly difficult without also increasing the number of false positives. Figure 14 shows the ROC curve for the MP3 16K format. The curve has a less steep rise and does not plateau as quickly as in higher bitrate formats, indicating that the classifier struggles more to distinguish between classes at this lower bitrate. Figure 15 presents the ROC curve for the FLAC format, which demonstrates strong performance, comparable to that of the WAV format. The similarity in their curves suggests that FLAC, like WAV, maintains high accuracy in distinguishing between classes.

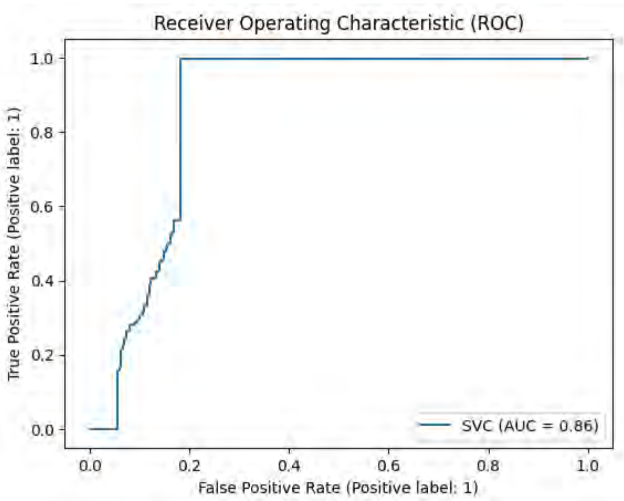

Figure 12. WAV ROC Curve for NU.

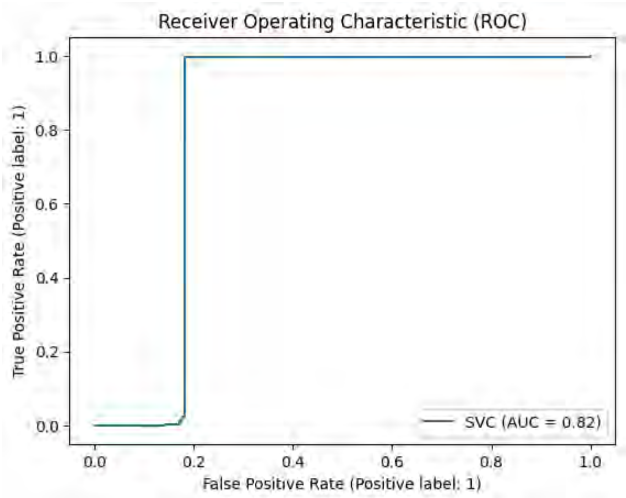

Figure 13. MP3 64K ROC Curve for NU.

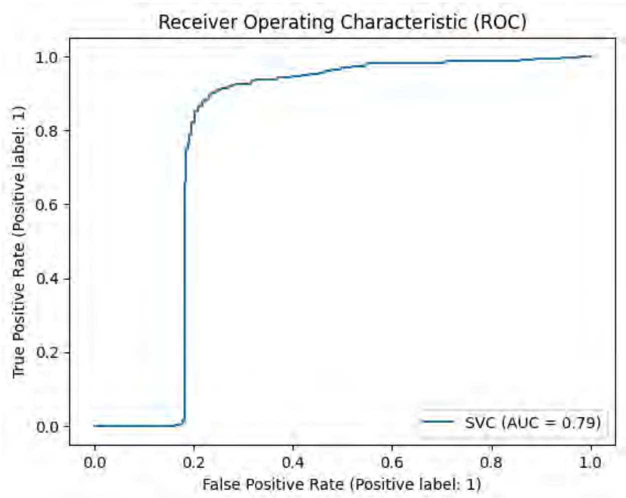

Figure 14. MP3 16K ROC Curve for NU.

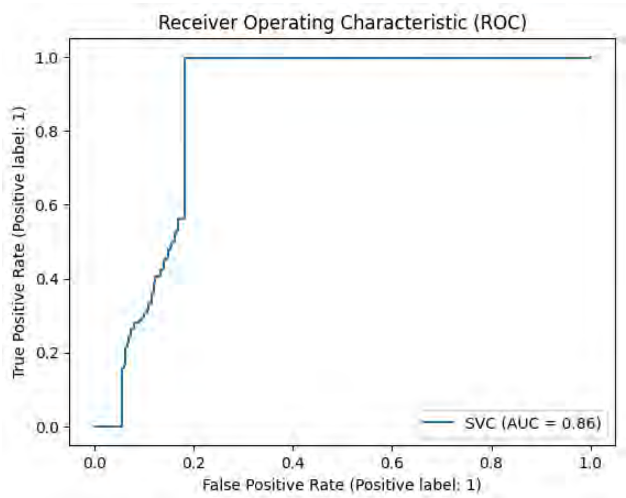

Figure 15. FLAC ROC Curve for NU, whose performance is strong, much like WAV.

6.2. OSBH DATASET

The OSBH (Open Source Beehives) project is a collaborative initiative aimed at advancing research and innovation in the field of bee monitoring and conservation. Led by a diverse community of researchers, beekeepers, and technology enthusiasts, the OSBH project seeks to develop accessible and open-source tools for monitoring bee populations and environmental conditions impacting bee health. For the OSBH project, the dataset was divided into training and testing segments for both queen and non-queen bee sounds (shown as Table 7). The following table provides a breakdown of the number of segments in each category:

Data Type	Number of Segments
Queen Training	600
No Queen Training	600
Queen Testing	200
No Queen Testing	144

Table 7. OSBH Data Set Segments. The dataset is divided into four distinct categories based on the presence of the queen bee and the purpose of the data (training or testing).

The following tables 8, 9, 10 and 11 provide a comprehensive analysis of classification accuracy, file sizes, confusion matrices, and classification reports for different audio formats—Uncompressed WAV, MP3 at 64 kbps and 16 kbps, and FLAC—across various aspects of the OSBH dataset. This dataset aims to evaluate the classification performance and robustness of each audio format in the context of beekeeping monitoring systems. These tables offer valuable insights into the accuracy and robustness of each audio format, considering both validation and testing scenarios. Additionally, the confusion matrices highlight the distribution of true positives, true negatives, false positives, and false negatives, providing further context on classification errors. Furthermore, the classification reports offer detailed metrics such as precision, recall, and F1-score, enabling a comprehensive assessment of classification performance across different audio formats. By examining these metrics collectively, we gain a deeper understanding of the strengths and limitations of each audio format in accurately classifying audio data in beekeeping monitoring applications. These insights serve as essential reference points for optimizing audio format selection and ensuring reliable classification performance in diverse beekeeping environments.

Bitrate	10-Fold Validation Accuracy	Testing Accuracy	File Size (Bytes)
Uncompressed (WAV)	100.00%	99.71%	201,107,113
MP3 64 kpbs	99.92%	99.42%	13,854,590
MP3 16 kpbs	99.83%	98.55%	6,927,374
FLAC	100.00%	99.71%	78,149,773

Table 8. OSBH Classification Accuracy and File Size. This result is generated by 10-Fold validation to increase robustness.

Audio Format	True Positives (TP)	True Negatives (TN)	False Positives (FP)	False Negatives (FN)
WAV	200	143	1	0
MP3 - 64K	199	143	1	1
MP3 - 16K	199	140	4	1
FLAC	200	143	1	0

Table 9. OSBH Confusion Matrix. WAV and FLAC have the least number of errors (1 FP each, 0 FN), whereas MP3 - 16K has the highest number of false positives (4) and a single false negative.

Audio Format	Precision	Recall	F1-Score	Support
WAV	1.00	0.99	1.00	144
MP3 - 64K	0.99	0.99	0.99	144
MP3 - 16K	0.99	0.97	0.98	144
FLAC	1.00	0.99	1.00	144

Table 10. Classification report (Precision, Recall, F1-Score, and Support) for different audio formats (WAV, MP3 - 64K, MP3 - 16K, and FLAC) in the No-Queen dataset.

Audio Format	Precision	Recall	F1-Score	Support
WAV	1.00	1.00	1.00	200
MP3 - 64K	1.00	1.00	1.00	200
MP3 - 16K	0.98	1.00	0.99	200
FLAC	1.00	1.00	1.00	200

Table 11. Classification report (Precision, Recall, F1-Score, and Support) for different audio formats (WAV, MP3 - 64K, MP3 - 16K, and FLAC) in the Queen dataset.

Analysing the ROC curves based on their shapes and the performance indicated by these shapes from the new dataset for OSBH in Figure 16, 17, 18 and 19.

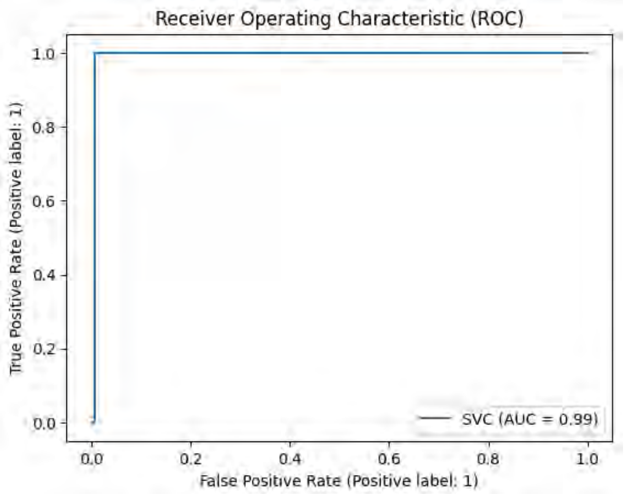

Figure 16. WAV ROC curve for OSBH, not fluctuating as NU dataset, assume data quality is high.

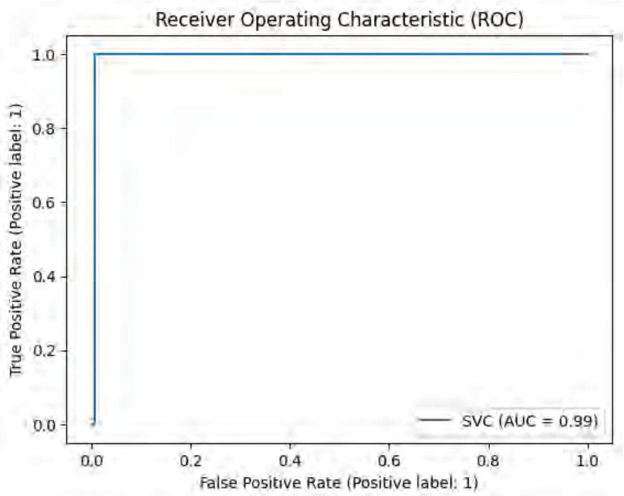

Figure 17. MP3 64K ROC Curve for OSBH, not fluctuating as NU dataset, assume data quality is high.

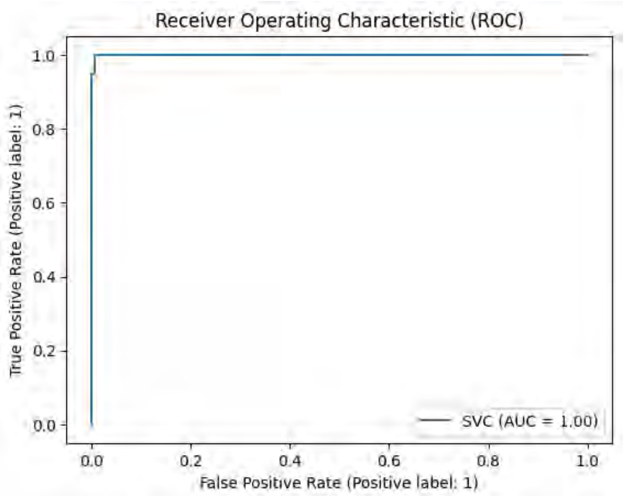

Figure 18. MP3 16K ROC Curve for OSBH, not fluctuating as NU dataset, assume data quality is high.

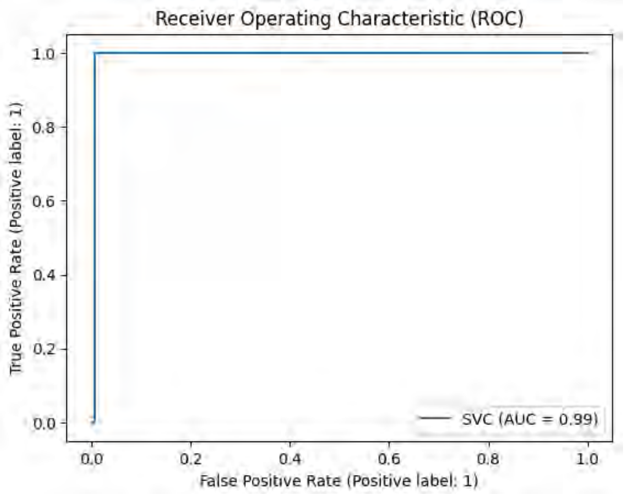

Figure 19 FLAC ROC Curve for OSBH, whose performance for FLAC is strong, much like WAV.

6.3. STANFORD BEEAUDIO DATASET

The dataset for the Stanford project, curated by Anna Yang, a student at Stanford University, was made publicly available on the Kaggle platform. The dataset comprises audio segments of bee sounds meticulously labelled and categorized into queen and non-queen bee sounds. The following table 12 provides a breakdown of the number of segments in each category:

Data Type	Number of Segments
Queen Training	620
No Queen Training	600
Queen Testing	201
No Queen Testing	240

Table 12. Distribution of data segments for BeeAudio dataset, categorized by type (Queen and No Queen) and purpose (Training and Testing).

The following tables 13, 14, 15 and16 provide a comprehensive analysis of classification accuracy, file sizes, and performance metrics for different audio formats—Uncompressed WAV, MP3 at 64 kbps and 16 kbps, and FLAC—across various aspects of the BeeAudio dataset. This additional dataset aims to evaluate the robustness of each audio format in accurately classifying audio data under varying conditions. These tables offer valuable insights into the effectiveness of each audio format, considering both classification accuracy and robustness metrics. By examining these metrics collectively, we gain a deeper understanding of the strengths and limitations of each audio format in the context of beekeeping monitoring systems, enhancing our ability to make informed decisions in real-world applications. These insights serve as crucial reference points for optimizing audio format selection and ensuring robust performance in diverse beekeeping environments.

Bitrate	10-Fold Validation Accuracy	Testing Accuracy	File Size (Bytes)
Uncompressed (WAV)	99.75%	99.55%	218,191,812
MP3 64 kpbs	98.70%	98.05%	14,877,413
MP3 16 kpbs	97.38%	99.32%	7,438,791
FLAC	99.75%	99.55%	83,919,220

Table 13. Comparison of classification accuracy and file sizes for different audio formats (Uncompressed WAV, MP3 64 kbps, MP3 16 kbps, and FLAC) in the BeeAudio dataset.

Audio Format	True Positives (TP)	True Negatives (TN)	False Positives (FP)	False Negatives (FN)
WAV	199	240	0	2
MP3 - 64K	199	238	2	2
MP3 - 16K	198	236	4	3
FLAC	199	240	0	2

Table 14. Confusion matrix details for different audio formats (WAV, MP3 - 64K, MP3 - 16K, and FLAC) in the BeeAudio dataset.

Audio Format	Precision	Recall	F1-Score	Support
WAV	0.99	1.00	0.99	240
MP3 - 64K	0.99	0.99	0.99	240
MP3 - 16K	0.99	0.98	0.98	240
FLAC	0.99	1.00	0.99	240

Table 15. Classification report (Precision, Recall, F1-Score, and Support) for different audio formats (WAV, MP3 - 64K, MP3 - 16K, and FLAC) in the No-Queen classification report of the BeeAudio dataset.

Audio Format	Precision	Recall	F1-Score	Support
WAV	1.00	0.99	0.99	201
MP3 - 64K	0.99	0.99	0.99	201
MP3 - 16K	0.98	0.99	0.98	201
FLAC	1.00	0.99	0.99	201

Table 16. Classification report (Precision, Recall, F1-Score, and Support) for different audio formats (WAV, MP3 - 64K, MP3 - 16K, and FLAC) in the Queen classification report of the BeeAudio dataset.

Analysing the ROC curves based on their shapes and the performance indicated by these shapes from the new dataset for BeeAudio in Figure 20, 21, 22 and 23. We will examine the ROC curves, discussing their shapes and the corresponding AUC values to evaluate the classification accuracy of the BeeAudio system under various conditions.

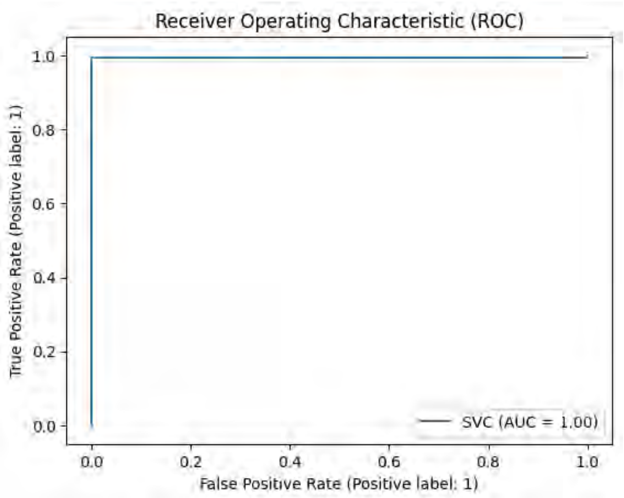

Figure 20. WAV ROC Curve for BeeAudio. This dataset has high quality causing the curve rises almost vertically.

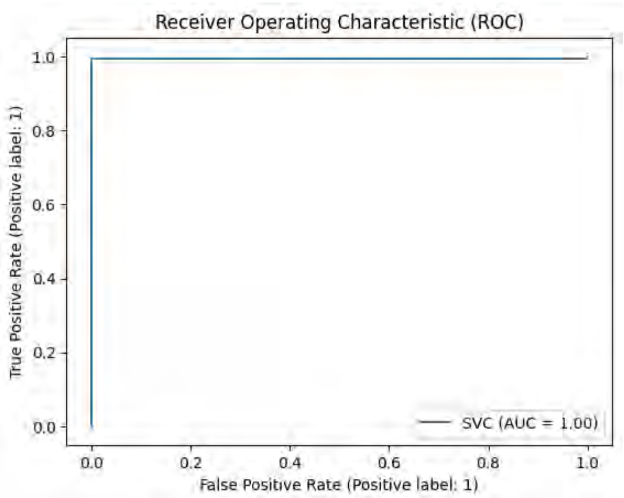

Figure 21. MP3 64K ROC Curve for BeeAudio. This dataset has high quality causing the curve rises almost vertically without obvious differences with WAV.

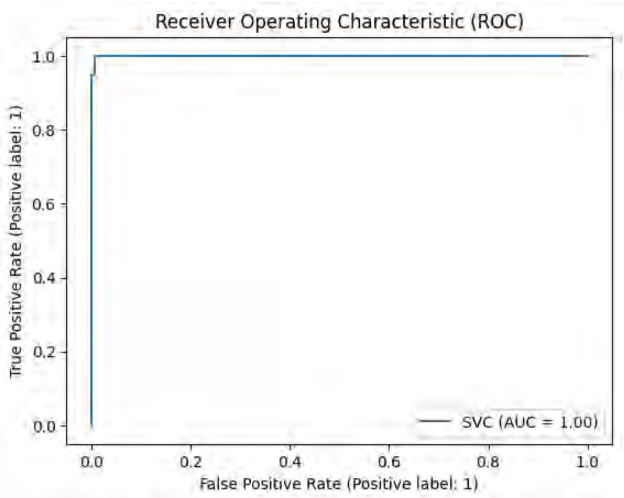

Figure 22. MP3 16K ROC Curve for BeeAudio. Despite the slight deviation at the start, the ROC curve shape and the AUC of 1.00 indicate that the classifier for MP3 16K achieves near-perfect sensitivity (true positive rate) with almost no loss in specificity.

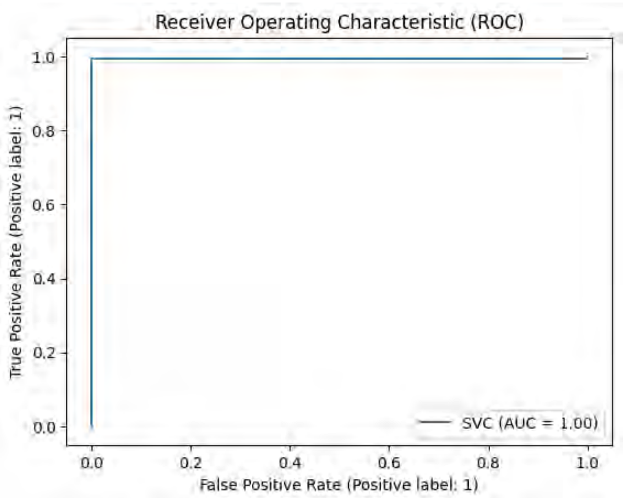

Figure 23. FLAC ROC Curve for BeeAudio. The performance for FLAC is strong, much like WAV.

6.4. COMPRESSION PERFORMANCE

During the experiments, we found notable differences in the compression times for the two formats (shown as Figure 24 and 25). For instance, compressing a 10-minute audio file into MP3 format took approximately 20 seconds, whereas the same file could be compressed into FLAC format in just 2 seconds. This recorded experimental data serves as a basis for evaluating the performance of MP3 and FLAC compression in practical applications. The findings will be further analysed and discussed in detail in the discussion section of this thesis. The analysis will focus on understanding the trade-offs between compression speed, audio quality, and the suitability of each format for real-time audio monitoring in beehive environments.

```
WAV -> MP3 Execution time: 19.11723756790161 seconds
ffmpeg version 2023-10-23-git-ff5a3575fe-essentials_build-www.gyan.dev Copyright (c) 2000-2023 the FFmpeg developers
  built with gcc 12.2.0 (Rev10, Built by MSYS2 project)
  configuration: --enable-gpl --enable-version3 --enable-static --pkg-config=pkgconf --disable-w32threads --disable-autodetect
  libavutil      58. 27.100 / 58. 27.100
  libavcodec     60. 30.102 / 60. 30.102
  libavformat    60. 15.100 / 60. 15.100
  libavdevice    60.  2.101 / 60.  2.101
  libavfilter     9. 11.100 /  9. 11.100
  libswscale      7.  4.100 /  7.  4.100
  libswresample   4. 11.100 /  4. 11.100
  libpostproc    57.  2.100 / 57.  2.100
```

Figure 24. MP3 Compression Time, which was implemented by FFmpeg tool.

```
[2024-05-13 20:57:38] Loading audio file from: Stanford_Queen_Testing_10mins.wav
[2024-05-13 20:57:38] Converting to FLAC: Stanford_Queen_Testing_10mins.flac
[2024-05-13 20:57:39] Conversion completed. Duration: 1.50 seconds.
```

Figure 25. FLAC Compression Time which was implemented by Python audio library.

6.5. TRANSMISSION PERFORMANCE

In this experiment, the efficiency of MQTT transmission times for an audio file named "QueenBee_Testing_15mins" in three different formats—WAV, MP3, and FLAC—was analyzed. The results, as depicted in Figures 26, 27 and 28, revealed distinct transmission times for each format. Specifically, the WAV format exhibited a transmission time of 302 seconds, contrasting with the MP3 format's swift transmission of 5 seconds. Meanwhile, the FLAC format occupied an intermediate position with a transmission time of 104 seconds. These findings provide valuable insights into the practical implications of audio format selection in real-time applications such as beekeeping monitoring systems.

```
Console ×  Problems  Declaration  Call Hierarchy
Java Stack Trace Console
Part 44 (Size: 1732381 bytes) published successfully in 3 seconds
Part 45 (Size: 1463662 bytes) published successfully in 3 seconds
Part 46 (Size: 1417099 bytes) published successfully in 3 seconds
Part 47 (Size: 9189471 bytes) published successfully in 8 seconds
Part 48 (Size: 3444528 bytes) published successfully in 3 seconds
Part 49 (Size: 4479763 bytes) published successfully in 4 seconds
Part 50 (Size: 4311153 bytes) published successfully in 4 seconds
WAV Compression MQTT Transmission Total time: 302 seconds
Audio file sent successfully.
```

Figure 26. MQTT transmission time for WAV, costing 302 seconds as evidence to evaluate the transmission performance.

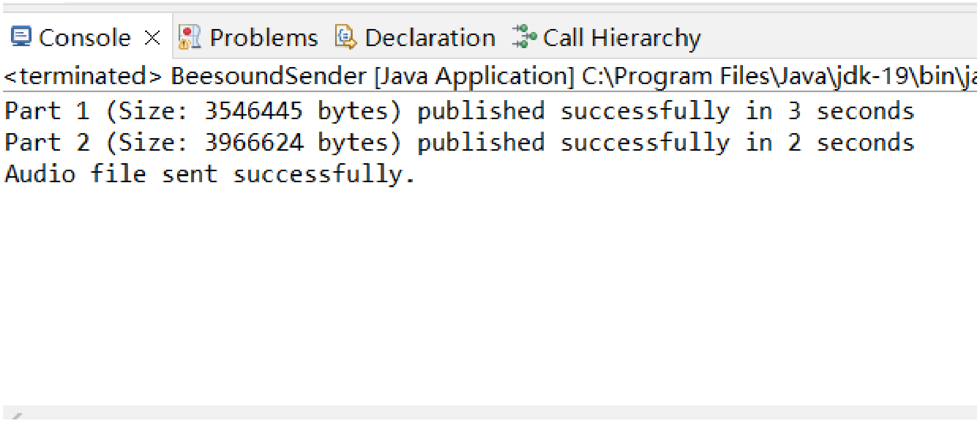

Figure 27. MQTT transmission time for MP3, which is much faster than WAV.

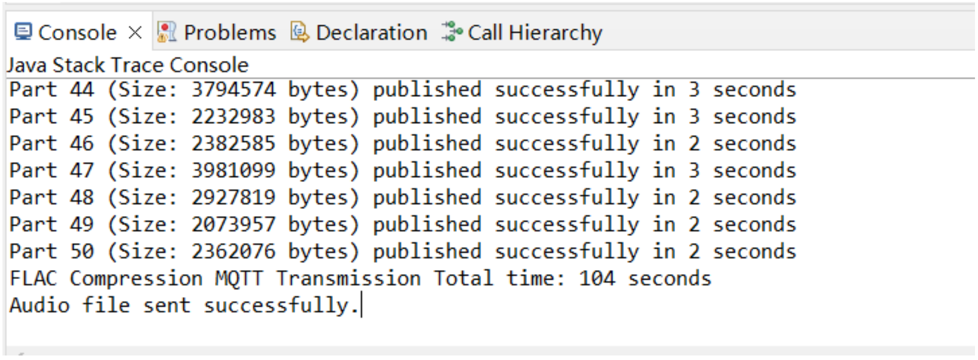

Figure 28. MQTT transmission time for FLAC shows that FLAC files, while larger than MP3 files, are significantly smaller than WAV files.

The experimental data have been summarized into the Table 17, facilitating direct comparative analysis. WAV files, being uncompressed and containing raw audio data, result in significantly larger file sizes compared to compressed formats. This large file size is reflected in the long transmission time of 302 seconds, indicating that uncompressed audio data requires more bandwidth and time to transmit over MQTT, which can lead to increased latency.

Audio Format	**Compression Time (seconds)**	**Transmission Time (seconds)**
WAV	N/A	302
MP3	20	5
FLAC	2	104

Table 17. Compression time and transmission time for different audio formats (WAV, MP3, and FLAC).

On the other hand, MP3 files are highly compressed and designed to reduce file size while maintaining reasonable audio quality. The short transmission time of 5 seconds highlights the efficiency of the MP3 compression algorithm. The smaller file size allows for faster data transfer over MQTT, making MP3 an ideal format for scenarios where bandwidth is limited, or quick transmission is required.

FLAC files, being lossless compressed audio files, retain the original audio quality while reducing the file size compared to uncompressed formats like WAV. The transmission time of 104 seconds shows that FLAC files, while larger than MP3 files, are significantly smaller than WAV files. This balance between maintaining high audio quality and achieving moderate file compression results in a reasonable transmission time for high-fidelity audio needs.

Chapter 7. Discussion

In this chapter, we delve into a comprehensive discussion, interpretation, and evaluation of our results, anchored in the relevant literature. Structured around the objectives of our study and theoretical framework, each section scrutinizes our findings in relation to existing research, highlighting both similarities and differences. Through this comparative analysis, we aim to develop theoretical insights and models, elucidating the impact of waveform on audio compression efficiency, distortion characteristics, robustness, and overall findings.

7.1. IMPACT OF WAVEFORM ON AUDIO COMPRESSION

The waveform is a graphical representation of an audio signal, showing how the amplitude of the sound varies over time. It captures the raw, natural features of the audio, such as its intensity, frequency, and temporal characteristics. Different audio formats represent this waveform with varying degrees of fidelity, depending on whether they are lossless or lossy (Zhang, 2001).

The waveforms of WAV and FLAC formats are nearly identical, as shown in the figures. Both of these formats are lossless, meaning they store audio data without any compression. This results in waveforms that preserve all the original details of the audio signal. The peaks and troughs in the waveform represent the variations in amplitude, which are faithfully captured in both WAV and FLAC formats. The detailed structure of the waveform reflects the full frequency content of the audio, including both low and high-frequency components. The precise timing of sound events is accurately represented, preserving the natural rhythm and patterns of the audio.

The MP3 format, on the other hand, is a lossy compression format. Its waveform, especially at lower bitrates, shows some differences compared to the lossless formats. The MP3 waveform may appear smoothed out or less detailed, as the compression process removes some of the audio information to reduce file size. Certain subtle features of the original audio might be lost or altered, leading to a less accurate representation of the natural sound. While the overall shape of the waveform is maintained, some of the finer amplitude variations and high-frequency details may be diminished.

The fidelity of the waveform in representing audio features has several implications. Higher fidelity waveforms (WAV and FLAC) maintain the full quality of the original audio, capturing all nuances and details. This is crucial for applications where audio quality is paramount, such as music production and high-fidelity audio playback. Lossless formats provide a more accurate and immersive listening experience, as they preserve the natural characteristics and subtleties of the sound. For any form of detailed audio analysis, whether for scientific research, audio engineering, or machine learning, maintaining the full detail of the waveform is essential to ensure accurate and reliable results.

In summary, the waveform is a natural and fundamental feature of audio, providing a detailed representation of how the sound varies over time. Lossless formats like WAV and FLAC offer the highest fidelity, preserving all the intricacies and details of the original audio signal. In contrast, lossy formats like MP3, while useful for reducing file size, compromise some of this detail, which can impact both the perceptual quality and the accuracy of any subsequent audio analysis. Understanding these differences is crucial for choosing the appropriate audio format based on the specific needs of the application.

7.2. COMPRESSION EFFICIENCY: SPEED AND DEGREE

In our research, we evaluated the performance of different audio compression formats, focusing specifically on MP3 and FLAC, with WAV as a baseline for uncompressed audio. The results of our experiments provided insights into the compression time, transmission time, and the degree of compression achieved by each format.

7.2.1 Degree of Compression

The degree of compression is a vital factor in determining the efficiency of data transmission in the beekeeping industry, particularly when dealing with audio data for hive monitoring. Compression reduces the file size, which in turn decreases transmission times, enabling more rapid data analysis and response. The following discussion focuses on the degree of compression for various audio formats: WAV, MP3 (at different bitrates), and FLAC, as evidenced by their file sizes.

7.2.1.1 MP3 Format

MP3 employs lossy compression, which significantly reduces file sizes by discarding some audio information. Our study evaluated MP3 at two different bitrates: 64 kbps and 16 kbps.

- **MP3 64 kbps**

This bitrate results in a file size of 22,167,345 bytes. The compression process for MP3 at this bitrate takes 20 seconds, with a transmission time of only 5 seconds, totaling 25 seconds. This high degree of compression greatly reduces the file size to approximately 6.8% of the original WAV file, making it highly efficient for rapid data transmission without severely compromising audio quality.

- **MP3 16 kbps**

At this lower bitrate, the file size is further reduced to 11,083,799 bytes. This represents about 3.4% of the original WAV file size. The decreased file size translates into even faster transmission, with the same total time of 25 seconds for compression and transmission as observed at 64 kbps. However, it is essential to consider that lower bitrates may affect the audio quality, which could impact the effectiveness of audio-based monitoring systems.

7.2.1.2 FLAC Format

FLAC, a lossless compression format, reduces the file size without any loss of audio quality. The file size for FLAC is 125,039,638 bytes, which is about 38.5% of the original WAV file size. The compression time for FLAC is 2 seconds, with a transmission time of 104 seconds, totaling 106 seconds. While not as compact as MP3, FLAC strikes a balance by maintaining high audio fidelity with a significantly reduced file size compared to WAV. This makes it suitable for applications where audio quality is critical, though with a trade-off in transmission speed.

7.2.1.3 Comparative Analysis

The degree of compression directly affects file size and transmission time, crucial factors for efficient data handling in beekeeping. MP3 formats, particularly at 64 kbps and 16 kbps, achieve significant file size reductions, facilitating rapid transmission times. FLAC, while offering lossless compression, presents a moderate reduction in file size with longer transmission times, balancing quality and efficiency. WAV, with its lack of compression, results in large file sizes and slow transmission, making it impractical for real-time applications.

In conclusion, the degree of compression is a critical factor in optimizing data transmission for beekeeping applications. MP3 formats provide significant file size reduction, making them suitable for scenarios requiring quick data transfer. FLAC offers a balance between file size and audio quality, suitable for applications where maintaining high audio fidelity is essential. The uncompressed WAV format, due to its large file size and lengthy transmission time, is less practical for real-time monitoring needs. By carefully considering the degree of compression, beekeepers can enhance the efficiency of their data transmission processes, ensuring effective and timely management of their bee colonies.

7.2.2 Speed of Compression and Transmission

7.2.2.1 Compression Speed

Compression speed is a critical factor influencing the efficiency of data handling in beekeeping applications. The WAV format, being uncompressed, requires no time for compression. While this eliminates any delay due to compression, it results in large file sizes that subsequently impact transmission times. The MP3 format, which uses lossy compression, has a compression time of 20 seconds. This relatively short compression period is balanced by the significant reduction in file size, making MP3 an efficient choice for quick data turnaround. On the other hand, FLAC, known for its lossless compression, requires only 2 seconds for compression. This minimal compression time, combined with the advantage of preserving high audio quality, makes FLAC a strong candidate for applications needing detailed audio fidelity. When comparing these formats, FLAC demonstrates the fastest compression speed, followed by MP3, with WAV requiring no compression time but suffering from large file sizes. The brief compression times of MP3 and FLAC enhance their practicality for real-time monitoring, where swift data processing is essential. Among the three formats, FLAC exhibits the fastest compression speed at 2 seconds, followed by MP3 at 20 seconds. WAV, being uncompressed, requires no compression time but results in large file sizes. MP3 and FLAC both offer quick compression processes, with FLAC being slightly faster and providing lossless audio quality. The brief compression times for MP3 and FLAC enhance their practicality for real-time beekeeping applications, where swift data processing is essential.

7.2.2.2 Transmission Speed

Transmission speed is pivotal in determining the overall efficiency of data transfer in beekeeping. The WAV format, with a transmission time of 302 seconds, is the longest among the formats due to its large file size. This extended transmission duration can delay real-time data analysis and hinder timely interventions, making it less suitable for immediate response scenarios. In contrast, the MP3 format significantly reduces transmission time to just 5 seconds, thanks to its lossy compression that produces much smaller file sizes. This rapid transmission capability is advantageous for real-time monitoring, allowing beekeepers to receive and analyze data promptly, facilitating quick responses to hive issues. FLAC, offering a moderate transmission time of 104 seconds, strikes a balance between speed and audio quality. While not as fast as MP3, FLAC's reduced file size compared to WAV enables more efficient transmission while maintaining high audio fidelity. Thus, MP3 stands out with the fastest transmission time, making it the most efficient format for rapid data transfer. FLAC provides a good balance of speed and quality, suitable for applications where audio fidelity is crucial. The WAV format, with its lengthy transmission time, is the least efficient in terms of speed, limiting its practicality for real-time monitoring. Considering transmission speed when selecting an audio format can enhance the effectiveness of beekeeping monitoring systems, ensuring timely and accurate data management. MP3 stands out with the fastest transmission time of 5 seconds, making it the most efficient format for rapid data transfer. FLAC, with a transmission time of 104 seconds, offers a compromise between speed and quality, suitable for applications requiring high audio fidelity. WAV, with its lengthy transmission time of 302 seconds, is the least efficient in terms of speed, limiting its practicality for real-time beekeeping monitoring.

7.2.2.3 Comparative Analysis

Comparing the three formats, MP3 demonstrates the highest overall efficiency with the shortest total process time of 25 seconds. This includes both compression and transmission, making MP3 ideal for real-time monitoring where rapid data transfer and analysis are crucial. MP3 is already widely used in many existing beehive monitor systems due to its efficiency in data handling and rapid transmission capabilities.

FLAC, with a total process time of 106 seconds, provides a balanced option for applications requiring high audio fidelity. Despite its longer transmission time compared to MP3, FLAC offers a reasonable compromise between speed and quality. FLAC is a format that we want to explore further because of its feature of maintaining high audio quality without loss, which can be beneficial for detailed audio analysis in beekeeping applications.

WAV, with its lengthy transmission time of 302 seconds and no compression, is the least efficient, limiting its practicality for real-time applications due to the significant delay in data availability. While WAV preserves the highest audio quality, its impractical transmission speed makes it less suitable for dynamic and responsive hive monitoring.

In scenarios where accuracy is a priority over speed, such as research studies or applications requiring detailed audio analysis, FLAC stands out as the optimal choice. Its ability to maintain the integrity of audio data ensures that beekeepers can make informed decisions based on the most accurate and reliable information available. Therefore, when precision and fidelity are paramount, FLAC should be the preferred option for audio data transmission in beekeeping monitoring systems.

7.3. DISTORTION CHARACTERISTICS

Distortion introduced by audio compression formats can significantly impact the effectiveness of machine learning models used in beehive monitoring. Analyzing the distortion characteristics of WAV, MP3, and FLAC formats reveals important insights into their relative performance and suitability for accurately classifying beehive audio data.

7.3.1 Impact on Classification Accuracy

The accuracy of classification models is directly influenced by the quality of the input audio data. In our experiments, both WAV and FLAC formats maintained high validation and testing accuracy, indicating that these formats preserve the essential characteristics of the audio signals needed for accurate classification. This preservation is crucial in beehive monitoring, where distinguishing between subtle audio cues of "Queen" and "No Queen" states is essential.

In contrast, MP3 format showed a decline in testing accuracy, especially at lower bitrates. The lossy nature of MP3 compression introduces distortions that degrade the audio quality, leading to a reduced ability of the classification model to accurately identify the different states. This decline in accuracy highlights the risk of using highly compressed lossy formats in critical monitoring applications where precision is paramount.

For beehive monitoring, maintaining high classification accuracy is vital to correctly identify the health and activity of the hive. Formats like WAV and FLAC, which preserve audio fidelity, ensure that critical audio details are not lost, thereby supporting reliable model performance. The drop in accuracy observed with MP3 suggests that using lossy compression can lead to misidentifications, potentially compromising the ability to monitor beehive conditions accurately.

7.3.2 Classification Report

The precision, recall, and F1-score metrics further illuminate the impact of distortion on classification performance. For the "No-Queen" state, both WAV and FLAC achieved the highest recall (0.79) and F1-score (0.88), demonstrating their reliability in correctly identifying the absence of a queen. MP3, particularly at lower bitrates, exhibited lower recall and F1-scores, indicating an increased rate of false negatives and overall reduced performance.

For the "Queen" state, WAV and FLAC again outperformed MP3, with precision and recall values of 0.83 and 1.00, respectively, resulting in an F1-score of 0.91. MP3's precision dropped to 0.78 at 64 kbps and 0.75 at 16 kbps, with corresponding F1-scores of 0.88 and 0.86. These results suggest that the distortions introduced by MP3 compression lead to misclassifications, potentially impacting the accuracy of beehive health assessments.

High precision and recall are essential in beehive monitoring to ensure accurate detection of queen presence or absence. The superior performance of WAV and FLAC in these metrics indicates their suitability for such applications, as they reduce the likelihood of false positives and negatives. In contrast, the poorer performance of MP3, particularly at lower bitrates, underscores the challenges posed by lossy compression in maintaining reliable monitoring systems.

7.3.3 Confusion Matrix

WAV and FLAC formats demonstrate a high number of true positives and true negatives with minimal false positives, indicating their robustness in maintaining audio integrity. MP3 formats, especially at lower bitrates, show increased false positives, reflecting the negative impact of lossy compression on model performance. The higher rate of false positives with MP3 suggests that the format introduces artifacts that are misinterpreted by the classification model, potentially leading to incorrect assessments in a beehive monitoring system.

For beehive monitoring, minimizing false positives and false negatives is critical to ensure accurate and reliable hive health assessments. The results for WAV and FLAC, showing lower false positives and negatives, highlight their effectiveness in preserving audio quality, which is crucial for accurate model performance. The increased false positives in MP3 formats indicate potential issues with misinterpretation of audio data, suggesting that lossy compression can compromise the reliability of beehive monitoring systems.

7.3.4 Impact on ROC Curve

In our research, we have employed ROC (Receiver Operating Characteristic) curves as a key metric to evaluate the performance of our classification model in distinguishing between different classes of audio data within beehive environments. Specifically, our model aims to discriminate between audio recordings that indicate the presence of a "Queen" bee and those that indicate "No Queen". The effectiveness of the model is quantified using the Area Under the Curve (AUC) values derived from the ROC curves for various audio formats including WAV, MP3 (at 64K and 16K bitrates), and FLAC.

Both WAV and FLAC formats achieved the highest AUC values of 0.86. The ROC curves for these formats rise steeply towards the top-left corner, indicating high sensitivity and low false positive rates. This performance can be attributed to the lossless nature of these formats, which preserve the full range of audio frequencies and nuances essential for accurately detecting the presence of the queen bee. The ability to maintain high audio quality without compression artifacts ensures that the subtle audio cues associated with the queen bee are not lost, thereby enhancing the model's accuracy. The MP3 format at 64K bitrate showed a slightly lower AUC of 0.82, while the 16K bitrate had the lowest AUC of 0.79. The ROC curves for these formats indicate a more gradual rise, reflecting a moderate trade-off between true positive and false positive rates. The decrease in AUC with higher compression rates suggests that significant audio details are lost due to compression, which adversely affects the model's ability to distinguish between "Queen" and "No Queen" instances. The compression artifacts and reduced audio fidelity in MP3 formats, especially at lower bitrates, result in the loss of crucial audio signals necessary for precise classification.

The ROC curve analysis underscores the importance of audio format selection in beehive audio monitoring. The high AUC values for lossless formats (WAV and FLAC) indicate that these formats are more suitable for accurately detecting queen bee presence due to their ability to preserve the integrity of the audio signals. This is particularly critical in beehive monitoring, where the detection of subtle audio cues such as the buzzing frequency and patterns associated with the queen bee is crucial. Conversely, while compressed formats like MP3 offer advantages in terms of reduced file sizes and storage efficiency, they come at the cost of decreased classification accuracy. This trade-off needs to be carefully considered depending on the specific requirements of the monitoring system. For applications where high accuracy is paramount, such as in scientific research or critical beekeeping operations, the use of lossless formats like WAV or FLAC is recommended. However, in scenarios where storage constraints are significant, and a slight reduction in accuracy is acceptable, higher bitrate MP3 formats might still be viable.

7.4. ROBUSTNESS

Robustness in machine learning models refers to their ability to maintain performance across different datasets and varying conditions. In the context of beehive monitoring, this robustness ensures that the classification models can reliably detect queen presence or absence across diverse audio recordings from different environments and equipment. To evaluate the robustness of our models, we analysed their performance on the OSBH (Open Source Beehive) and Stanford BeeAudio datasets, comparing the results with our primary dataset.

7.4.1 Performance on OSBH Dataset

The OSBH dataset results underscore the reliability of the models across various audio formats. Uncompressed WAV and FLAC formats maintained high accuracy, demonstrating their robustness in preserving audio fidelity essential for accurate classification. Despite a slight reduction in performance, MP3 formats still exhibited strong accuracy, highlighting their resilience in maintaining high classification accuracy even with lossy compression.

The strong performance of WAV and FLAC formats on the OSBH dataset reaffirms their suitability for critical monitoring applications. MP3, despite its lossy nature, provides robust enough performance for less critical monitoring tasks or when storage efficiency is paramount. This robustness ensures that the monitoring system can adapt to different data sources and still deliver reliable results.

7.4.2 Performance on Stanford BeeAudio Dataset

The Stanford BeeAudio dataset results similarly reinforce the robustness of our models. Uncompressed WAV and FLAC formats again achieved high accuracy, illustrating their effectiveness in maintaining audio quality and classification accuracy across datasets. MP3 formats showed slightly more variation but still delivered reliable results, suggesting that lossy compression effects can vary based on specific dataset characteristics.

The consistently high performance of WAV and FLAC formats across different datasets ensures that beehive monitoring systems can be deployed in various environments without compromising accuracy. The unexpected robustness of MP3 at lower bitrates under certain conditions offers flexibility in storage and transmission strategies, ensuring reliable monitoring results even with compressed audio formats.

7.5. FINDINGS

MP3 offers the fastest transmission time, but it does so at the expense of audio quality due to lossy compression. WAV, despite offering the best audio quality, is impractical for rapid transmission due to its large file size. FLAC provides a middle ground, offering significant reductions in transmission time compared to WAV, while preserving audio quality, making it an optimal choice for applications where maintaining audio fidelity is important. This aligns with the research gap focusing on compression without feature loss, highlighting FLAC as a suitable format for preserving audio features while enabling more efficient data transmission.

Chapter 8. Conclusions and Future Works

The research aimed to address the challenges associated with audio data compression in beehive monitoring by exploring the feasibility and effectiveness of using the FLAC compression format. This was accomplished through a systematic approach encompassing the evaluation of compression techniques, MFCC feature extraction, and machine learning models. Our experiments compared the performance of FLAC, MP3, and uncompressed WAV formats across three datasets, each representing different beehive conditions and audio environments. The goal was to determine how well these formats preserved the essential acoustic features necessary for accurate hive condition analysis.

8.1. CONCLUSIONS

The research aimed to address the challenges associated with audio data compression in beehive monitoring by exploring the feasibility and effectiveness of using the FLAC compression format. This was accomplished through a systematic approach encompassing the evaluation of compression techniques, MFCC feature extraction, and machine learning models. Our experiments compared the performance of FLAC, MP3, and uncompressed WAV formats across three datasets, each representing different beehive conditions and audio environments. The goal was to determine how well these formats preserved the essential acoustic features necessary for accurate hive condition analysis.

8.1.1 Contribution

This thesis makes significant contributions to the field of smart beekeeping by demonstrating the efficacy of FLAC compression in reducing the resource consumption without feature loss, thereby without compromising AI performance. By validating that FLAC maintains the same classification accuracy as uncompressed formats, this research offers a practical solution for efficient data storage and transmission. Additionally, by comparing FLAC with other codecs, the thesis provides empirical evidence on highlighting FLAC as a superior option for resource-constrained beehive monitoring systems. These findings advance technical knowledge in audio-based beehive monitoring and propose practical solutions to enhance the efficiency and effectiveness of beekeeping monitoring systems, laying a foundation for future research and innovation in smart agriculture.

8.1.2 Methodology

In the methodology phase, the Free Lossless Audio Codec (FLAC) was implemented to balance storage efficiency with audio fidelity. The encoding process of FLAC involves segmenting the audio signal into blocks, predicting the signal using a model based on past samples, and encoding the prediction error using lossless methods such as Rice coding. This step-by-step encoding preserves the original audio data while achieving compression. During decoding, the process reverses: the encoded data is decoded to reconstruct the prediction error, which, when combined with the prediction model, recreates the original audio signal without any loss of information. This ensures that FLAC maintains the integrity of the audio waveform, as demonstrated by waveform figures that illustrate the indistinguishable similarity between FLAC and the original WAV files. These comparisons highlight FLAC's ability to reduce file size significantly without compromising audio quality, making it ideal for applications where maintaining high fidelity is paramount.

Secondly, Mel-Frequency Cepstral Coefficients (MFCCs) are widely adopted for their efficacy in capturing crucial spectral features from audio signals. The MFCC computation begins with pre-emphasis to enhance high-frequency components, followed by segmentation into short frames and application of windowing functions to mitigate spectral leakage. Each frame undergoes a Fourier Transform to convert it into the frequency domain, where it is then passed through a Mel Filterbank that mimics the human auditory system's frequency sensitivity. The resulting filterbank energies are logarithmically compressed to approximate human perception of sound intensity. Finally, a Discrete Cosine Transform (DCT) is applied to decorrelate the coefficients, yielding a compact representation of the audio's spectral characteristics. This process allows MFCCs to efficiently capture and encode essential aspects of the audio signal. Importantly, the Discrete Cosine Transform (DCT) is particularly suitable for machine learning analysis due to its ability to decorrelate the coefficients, providing a concise yet informative representation that enhances the efficiency and effectiveness of subsequent classification tasks. This makes MFCCs indispensable for tasks requiring robust feature extraction and analysis, such as speech recognition, music genre classification, and in our case, distinguishing queen and no queen states in smart beehive monitoring.

Finally, Support Vector Machine theory revolves around finding the optimal hyperplane that separates different classes in a high-dimensional feature space. It achieves this by maximizing the margin between the classes, which enhances generalization and reduces overfitting. SVM is particularly suitable as a supervised learning algorithm due to its ability to handle complex, non-linear relationships in data through kernel functions, such as the radial basis function (RBF) kernel. SVM aims to minimize classification errors while maintaining a maximum margin between support vectors, which are the data points closest to the decision boundary. This approach ensures robust classification performance and is well-suited for tasks like distinguishing queen and no queen states in beehive monitoring, where the data can be high-dimensional and non-linear.

These methodologies form the backbone of our research, providing a cohesive framework for effective beehive monitoring. By integrating these advanced techniques, we have addressed key challenges and demonstrated significant improvements in the detection and analysis of bee colony health.

8.1.3 Experiment

The experimental phase of this study provided significant insights into the process of classifying bee sound recordings, including the effectiveness of data segmentation, the impact of audio compression, the utility of feature extraction methodologies, and the operationalization of the classification system using MQTT.

The segmentation of bee sound recordings into 3-second clips proved to be an effective approach for creating manageable and analyzable audio samples. This segmentation allowed for a detailed examination of the audio signals and facilitated the classification task by providing a consistent unit of analysis. Labeling these segments as "queen" and "no queen" categories was crucial in addressing the research question, enabling a focused and relevant classification task.

Next, the segmented audio clips underwent compression using FFmpeg. Both MP3 and FLAC formats were employed to evaluate the impact of different compression methods. MP3 compression was tested at bitrates of 64 kbps and 16 kbps to assess the effect of lossy compression, while FLAC compression was used as a lossless alternative to preserve audio quality.

A critical component of this study was the integration of the classification system with MQTT (Message Queuing Telemetry Transport) for real-time data transmission. The implementation of MQTT facilitated the efficient and reliable communication of classification results. This integration enabled the system to send alerts and updates promptly, ensuring that users could respond to changes in the beehive environment in a timely manner. The MQTT protocol proved to be a robust solution for the transmission of classification results, demonstrating its suitability for real-time monitoring applications.

In the feature extraction phase, Mel-Frequency Cepstral Coefficients (MFCCs) were utilized to capture the essential spectral properties of audio signals. MFCCs are effective because they divide the audio spectrum into mel-frequency bands spaced according to the human perception of sound. To extract MFCC features from audio segments, the Librosa library in Python was used, specifically the librosa.feature.mfcc function. By invoking this function with the appropriate parameters, a matrix of MFCC coefficients representing the spectral characteristics of the audio segment was obtained. In this phase, four audio files were meticulously chosen, each representing distinct formats: WAV, MP3 at 64Kbps, MP3 at 16Kbps, and FLAC. Leveraging the powerful capabilities of librosa feature melspectrogram function, mel spectrograms were computed for each audio file, encapsulating their frequency distributions over time. This comprehensive approach enabled a nuanced comparison of spectrogram characteristics among different formats.

The Support Vector Machine (SVM) classifier trained on MFCC features demonstrated strong performance in distinguishing between "queen" and "no queen" bee sounds. The cross-validation results indicated that the classifier generalized well to unseen data, achieving high accuracy and low error rates. This highlighted the effectiveness of the SVM classifier in this specific audio classification task. The classifier's performance was evaluated using various metrics: accuracy of the SVM classifier, confusion matrix provided a detailed breakdown of true positives, true negatives, false positives, and false negatives and the classification report included precision, recall, and F1-score for each class ("queen" and "no queen"). These metrics provided insights into the classifier's effectiveness in correctly identifying each class.

8.1.4 Findings

The findings of our experiments are discussed by integrating the evaluation metrics across different audio formats: WAV, MP3, and FLAC. These metrics include waveform characteristics, classifier accuracy, compression degree and speed, and transmission speed.

Waveform Characteristics and Classifier Accuracy

The WAV format provided the most accurate and detailed waveform representation, preserving all audio details due to its uncompressed nature. This high fidelity was evident in the spectrograms, which showed consistent frequency and amplitude patterns. Consequently, the Support Vector Machine (SVM) classifier trained on Mel-Frequency Cepstral Coefficients (MFCC) features extracted from WAV files achieved the highest accuracy. The rich and unaltered audio information in WAV files contributed significantly to the classifier's strong performance in distinguishing between "queen" and "no queen" bee sounds.

In contrast, MP3 files, especially at lower bitrates like 16 kbps, exhibited noticeable artifacts and loss of high-frequency details. These artifacts were visible in the spectrograms as irregular patterns and reduced frequency information. The SVM classifier trained on MP3 files showed lower accuracy compared to WAV, as the lossy compression process led to the loss of critical audio features. However, at a higher bitrate of 64 kbps, the quality of MP3 files improved, resulting in better classifier performance, though still not matching the accuracy achieved with WAV or FLAC files.

FLAC, being a lossless format, maintained high-quality waveforms similar to WAV. The spectrograms for FLAC files showed well-defined frequency patterns, indicating no loss of audio information. Consequently, the SVM classifier trained on FLAC files performed comparably well to WAV, achieving high accuracy due to the preservation of all audio features. This demonstrates FLAC's suitability for tasks requiring high-fidelity audio analysis without the drawbacks of lossy compression.

Compression Degree, Speed, and Transmission Speed

When considering compression degree and speed, MP3 files offered the highest compression, significantly reducing file size, enhancing efficiency in storage and processing. However, this compression came at the cost of audio quality, especially at lower bitrates. The reduced file size of MP3 files also resulted in the fastest transmission speeds, making them ideal for applications where storage and bandwidth are limited, such as streaming and real-time audio transmission.

WAV files, being the original and uncompressed format, do not require compression. This results in slower transmission speeds and larger storage requirements due to their large size. While WAV files offer the best audio quality, their lack of compression can be a drawback for practical applications that need efficient data management and transmission.

FLAC files, although larger than MP3 files, provide lossless compression, meaning no audio information is lost. FLAC has the fastest compression speed among the formats we evaluated, making it efficient for scenarios where maintaining audio quality is critical. The moderate file size of FLAC compared to WAV results in moderate transmission speeds, balancing the need for high-quality audio with practical considerations for storage and bandwidth.

Limitation

Despite the positive findings, the research has some limitations. The datasets used in the experiments, while diverse, may not cover all possible beehive conditions and environmental variations. Additionally, the implementation of the SVM models was based on specific parameters that may not generalize to all contexts. Further research is needed to test the robustness of the findings across a wider range of conditions and to explore the use of different machine learning approaches.

8.2. FUTURE WORKS

The research opens several avenues for future work, each promising to enhance and expand the utility of audio classification and real-time monitoring in various domains.

One potential avenue for future research is to investigate the application of Free Lossless Audio Codec in other agricultural monitoring systems. Given FLAC's ability to compress audio without any loss of quality, it could be highly beneficial for other areas of precision agriculture. For instance, monitoring animal sounds, soil health, or crop conditions using acoustic signals can benefit from FLAC’s lossless compression, ensuring data integrity. By applying FLAC in diverse agricultural contexts, researchers can assess its broader utility and potentially develop comprehensive monitoring solutions that leverage high-quality audio data for improved decision-making and management.

Another promising direction is to explore the combination of FLAC with advanced AI techniques, such as deep learning. Deep learning models, particularly those involving convolutional neural networks (CNNs) and recurrent neural networks (RNNs), have shown remarkable success in various audio processing tasks. Integrating FLAC-compressed audio data with these sophisticated AI models could further enhance the capabilities of smart beekeeping systems. This combination could enable more accurate and nuanced analysis of bee sounds, leading to better detection of hive conditions, health issues, and behavioural patterns. Additionally, leveraging deep learning could open possibilities for real-time, automated monitoring systems that provide actionable insights with minimal human intervention.

Future work could also focus on exploring alternative codecs that might achieve a better balance between audio distortion and resource consumption. While FLAC offers lossless compression, other newer advancements in audio compression might provide efficient lossy compression with acceptable levels of distortion. These codecs could potentially reduce the computational and storage requirements while maintaining sufficient audio quality for classification tasks. By systematically evaluating various codecs under different conditions, researchers can identify the most suitable options for specific use cases, thereby optimizing the trade-off between audio quality and system performance.

References

A, S., Thomas, A., & Mathew, D. (2018). Study of MFCC and IHC Feature Extraction Methods With Probabilistic Acoustic Models for Speaker Biometric Applications. *Procedia Computer Science*, 143, 267–276.

A. L. Imoize, S. D. (3, Aug. 2020). Development of a Low-Cost Wireless Bee-Hive Temperature and Sound Monitoring System. *Indonesian Journal of Electrical Engineering and Informatics (IJEEI)*.

Abdollahi, M. H. (2022). The importance of context awareness in acoustics-based automated beehive monitoring. *Applied Sciences*, 13(1).

Ali et al. (2021, November). A review on the stingless beehive conditions and parameters monitoring using IoT and machine learning. *Conference Series*.

Andrijević, N. U. (2022). IoT monitoring and prediction modeling of honeybee activity with alarm. . *Electronics*, 11(5).

Apple Inc. (2011). Apple Lossless Audio Codec. *Document*.

Banharnsakun, A. (2019). Towards improving the convolutional neural networks for deep learning using the distributed artificial bee colony method. *International journal of machine learning and cybernetics*, 10(6), 1301-1311.

Beckman, S. A. (2006). From CD to MP3: . *Compression in the New Age of Technology Overlooked Infringement or Fair Use. Gonz. L. Rev.*, 42, 469.

Berckmans, D. (October 2004). Automatic on-line monitoring of animals by precision livestock farming. In Proceedings of the International Congress on Animal Production in Europe. *Saint-Malo, France*, pp. 27–30.

Berry, R. (2006). Will the iPod kill the radio star. *Profiling podcasting as radio*, 2(2).

Blanc, S. (Mar. 2018). TRADITIONAL BEEKEEPING IN RURAL AREAS: - ProQuest. *www.proquest.com*.

Blanc, Simone. (Mar. 2018). TRADITIONAL BEEKEEPING IN RURAL AREAS. *www.proquest.com*.

Blut, C. C. (2017). Automated computer-based detection of encounter behaviours in groups of honeybees. *Scientific Reports*, 7(1), 17663.

Bosi, M., & Goldberg, R. E. . (2002). Introduction to Digital Audio Coding and Standards. Springer.

Boyd, L. E. (2017). Evaluating an iPad game to address overselectivity in preliterate AAC users with minimal verbal behavior. *In Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility*, pp. 240-249.

Boys, R. (2021). Beekeeping in a Nutshell—Listen to the Bees. *Self Published Work.*

Brandenburg, K. (2010). MP3 and AAC explained. *In AES 119th Convention*.

Burkart, P. (. (2008). Trends in digital music archiving. *The Information Society*, 24(4), 246-250.

C. Schmidt, K. E. (Jan. 2021). New Zealand mānuka honey - A review on specific properties and possibilities to distinguish mānuka from kānuka honey. *LWT*, vol. 136, p. 110311, .

Cecchi, S. S. (2020). A smart sensor-based measurement system for advanced bee hive monitoring. 20(9).

Cho, C. S. (2007, July). A Fixed-point Implementation of the MPEG-4 ER-BSAC Decoder. *International Technical Conference on Circuits*.

Cook, V. A. (2015). Facts about Beekeeping in New Zealand. *Bee World*, pp. 88–100.

Crawford, M. (2017). Automated collection of honey bee hive data using the Raspberry Pi. *Doctoral dissertation, Appalachian State University*.
Cristia, A. L. (2021). A thorough evaluation of the Language Environment Analysis (LENA) system. *Behavior research methods*, 53, 467-486.
Cunha, A. S. (2020). A novel non-invasive radar to monitor honey bee colony health. *Computers and Electronics in Agriculture*.
D. T. Várkonyi, J. L. (Mar. 2023). Dynamic noise filtering for multi-class classification of beehive audio data. *Expert Systems with Applications*, vol. 213, p. 118850, .
D'Alessandro, B. &. (2009, September). MP3 bit rate quality detection through frequency spectrum analysis. *In Proceedings of the 11th ACM Workshop on Multimedia and Security*, (pp. 57-62).
Debnath, A. &. (2024). Lossless audio CODEC using non-repeated dynamic block encoding. *International Journal of Information Technology*, 1-8.
DeCambra, Weston. . (2024). The Morphology and Biochemical Structure of Spider Egg Cases. *Potential Arachnid-based Silk Nanosealants in Medicine*.
Deepu, C. J. (2017). A Hybrid Data Compression Scheme for Power Reduction in Wireless Sensors for IoT. *ieeexplore.ieee.org,* .
Deshmukh, Soham, et al. (2024). Training audio captioning models without audio. *Speech and Signal Processing (ICASSP)*.
Dietlein, D. (n.d.). A Method for Remote Monitoring of Activity of Honeybee Colonies by Sound Analysis. *Apic. Res.*, 24, 176–183.
Dittmar, T. (2017.). A Beginner's Guide to Music Production. Routledge, . *Audio Engineering*.
Djebbar, F., & Ayad, B. (2017). Energy and Entropy Based Features for WAV Audio Steganalysis. *Hiding Multim. Signal Process*, 8(1), 168-181.
Donovan, B. J. (Nov. 08, 2023). Interactions Between Native and Introduced Bees in New Zealand. *New Zealand Journal of Ecology*, pp. 104–116.
Egidi, L. &. (2005). Bringing multimedia contents into MP3 files. *IEEE Communications Magazine*, 43(5), 90-97.
Fang, Z. W. (2009). December). Design and Research on Free Lossless Audio Decoding Systems. *Second International Symposium on Information Science and Engineering*, pp. 223-226.
Ferrari, S., Silva, M., Guarino, M., & Berckmans, D. (2008). Monitoring of swarming sounds in bee hives for early detection of the swarming period. *Computers and Electronics in Agriculture*, 64(2), 72-77.
Fiannaca, A. e. (2017). AACrobat: Using mobile devices to lower communication barriers and provide autonomy with gaze-based AAC. *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*.
Firmansah, L. a. (2016). Data audio compression lossless FLAC format to lossy audio MP3 format with Huffman shift coding algorithm. *International Conference on Information and Communication Technology (ICoICT)*.
Frantiska Jr, J. J. (2008). One sound is worth a thousand words: Using and understanding audio files. *Technology and Engineering Teacher*, 67(5), 23.
Gil-Lebrero, e. a. (2016). Honey bee colonies remote monitoring system. *Sensors*, 17(1), 55.
Griffin, A. (2016). Finding Hi-Res music. *Sound & Vision*, 81(4), 40-42.
Gunawan, T. S. (2017). Investigation of various algorithms on multichannel audio compression. *IEEE 4th International Conference on Smart Instrumentation Measurement and Application (ICSIMA). .*
Gupta, A. R. (2017). Speech Recognition Using Correlation Technique. *International Journal of Current Trends in Engineering & Research*, 2455-1392.

Guruprasad, S. M. (2024). BeeOpen—An Open Data Sharing Ecosystem for Apiculture. *Agriculture* , 14.3.

Hall, H. B. (2023). Automated, non-invasive Varroa mite detection by vibrational measurements of gait combined with machine learning. *Scientific Reports*, 13(1), 10202.

Han, Y. e. (2014). Determination of bit-rate adaptation thresholds for the opus codec for VoIP services. *IEEE Symposium on Computers and Communications (ISCC)*.

Hans, M. &. (2001). Lossless compression of digital audio. *IEEE Signal processing magazine*, 18(4), 21-32.

Hawa, M. R.-A.-N. (2012). File size models for shared content over the BitTorrent Peer-to-Peer network. *Peer-to-peer Networking and Applications*, 5, 279-291.

Herrmann, B. (2023). The perception of artificial-intelligence (AI) based synthesized speech in younger and older adults. *International Journal of Speech Technology*, 26(2), 395-415.

Higginbotham, D. Jeffery, et al. (2007). Access to AAC: Present, past, and future. *Augmentative and alternative communication*, 243-257.

I. Z. Ochoa, S. G. (Oct. 01, 2019). Internet of Things: Low Cost Monitoring BeeHive System using Wireless Sensor Network. *IEEE Xplore*.

Ibraheem, M. S. (2017). Fast and parallel AAC decoder architecture for a digital radio mondiale 30 receiver. *IEEE Access,* , 5, 14638-14646.

Israelsson, J. I. (Jan. 01, 1996). Short description of FLAC. *ScienceDirect*.

Jackson, W. (2015). Digital audio editing fundamentals. *Apress*.

Jaganmohan, P. B. (2023). WAV-RNN MUSIC GENERATOR.

JavaTPoint. (2021). Support Vector Machine (SVM). *Algorithm - Javatpoint. Www.javatpoint.com*.

Jin, R., & Kim, J. (2014). Analysis of FLAC Music Pieces Recovery. *Journal of Advances in Computer Networks*, 2(2), 134-137.

Jin, X. a. (2014). Audio fragment identification system. *International Journal of Multimedia and Ubiquitous Engineering*, 307-320.

Jukan, A., Masip-Bruin, X., & Amla, N. (2017). Smart computing and sensing technologies for animal welfare. *A systematic review. ACM Comput. Surv*, 50, 10.

Kalampogia, A. &. (2017). video bandwidth prediction. *IEEE Transactions on Multimedia*, 20(1), 171-182.

KARAGIANNIS, K. (2000). Making Music on a PC. Poptronics. 1(4), 37-37.

Kaul, S. a. (2019). Opus and session initiation protocol security in voice over IP. *European Journal of Engineering and Technology Research*, 4.12 (2019).

Khalil, H. (2021). Identifcation of Insect Pollinator Species using Bioacoustics and Artifcial Intelligence. *Doctoral dissertation University of York*.

Kim, B., & Rafii, Z. (2018, September). Lossy audio compression identification. *In 2018 26th European Signal Processing Conference (EUSIPCO)*, (pp. 2459-2463). IEEE.

Kim, H. (2017). Embedded Metadata in Digital Audio For Copyright Management. *International Information Institute (Tokyo)*, 20(2B), 1083.

Koller, J. S. (1999, September). Improving lossless audio coding. *17th International Conference: High-Quality Audio Coding.* .

Kontogiannis, S. (2019). An internet of things-based low-power integrated beekeeping safety and conditions monitoring system. *Inventions*, 4(3), 52.

Kozamernik, F. (2002). Digital audio broadcasting. *EBU Technical Review*, 293, 3-25.

Kridi, D. S. (2016). Application of wireless sensor networks for beehive monitoring and in-hive thermal patterns detection. *Computers and Electronics in Agriculture*, 127, 221-235.

Kromer, R. (2017). Matroska and FFV1: One file format for film and video archiving. *Journal of Film Preservation*, (96), 41.
Kumar, P. &. (2014). Performance analysis for audio streaming in Cloud. *IOSR-JCE*, 16, 98-104.
Li, W. L. (2010, July). Robust audio identification for MP3 popular music. *In Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pp. 627-634.
Lin, H.-C. A. (2024). Hybrid spherical array for clinical optoacoustic ultrasound (OPUS) imaging. *Photons Plus Ultrasound: Imaging and Sensing*.
Liu, B. X. (2011). Thwarting audio steganography attacks in cloud storage systems. *International Conference on Cloud and Service Computing*, (pp. 259-265). IEEE.
Manning, P. (2013). Electronic and Computer Music. *Oxford University Press*.
Maruschke, Michael, et al. (2015 September). Review of the Opus codec in a WebRTC scenario for audio and speech communication. *Speech and Computer: 17th International Conference*, 20-24,.
McCandless, M. (1999). The MP3 revolution. *IEEE Intelligent Systems and their Applications*, 14(3), 8-9.
Melchior, V. R. (2019). High-resolution audio: a history and perspective. *Journal of the Audio Engineering Society*, 67(5), 246-257.
Microsoft. (1991). Waveform audio file format.
Muin, F. A. (2017). A review of lossless audio compression standards and algorithms. . *AIP Conf*, https://doi.org/10.1063/1.5002024.
Mulyadi, Y., & Daryana, H. A. (2021). DAW (Digital Audio Workstation) Technology In The Music Of West Java Traditional Theatre. *Prosiding ISBI Bandung,* , 1(1).
Murphy, F. E. (2015. June). An automatic, wireless audio recording node for analysis of beehives. *Irish Signals and Systems Conference* , (pp. 1-6). IEEE.
Nanabeka, A. J. (2017). Updating and Modernizing WICN's Music Library.
Nassif, A., Shahin, I., Attili, I., Azzeh, M., & Shaalan, K. (2019). Speech recognition using deep neural networks. *A systematic review*.
Niell, S. J. (2018). Beehives biomonitor pesticides in agroecosystems:. *Ecological indicators*.
Nolasco, I., & Benetos, E. (2018.). To bee or not to bee: An annotated dataset for beehive sound recognition. *Zenodo* .
Open Source Beehives. (n.d.). Open Source Beehives Project. *https://www.osbeehives.com/.*
Plummer, M. ((2014)). Apple Pro Training Series. *GarageBand. Peachpit Press.*
Pohlmann, K. C. (2011). Principles of Digital Audio. *McGraw-Hill Education*.
Polyak, A. W. (2021 June). High fidelity speech regeneration with application to speech enhancement. *Speech and Signal Processing* , pp. 714.
Porter, Alastair, et al. (2015 Oct). Acousticbrainz: a community platform for gathering music information obtained from audio. *16th International Society for Music Information Retrieval Conference*, 26-30.
Povey, D.; Ghoshal, A.; Boulianne, G.; Burget, L.; Glembek, O.; Goel, N.; Hannemann, M. (2011). Workshop on Automatic Speech Recognition and Understanding;. *IEEE Signal Processing Society: Piscataway*.
Project., O. S. (15 December 2020). Open Source Beehives Project. *online: https://www.osbeehives.com/pages/about-us*.
Qandour, Amro, et al. (2014). Remote beehive monitoring using acoustic signals.
Rashidinejad, A. &. (2013). Achieving the upper bound time-bandwidth product for radio-frequency arbitrary waveform generation. *IEEE International Topical Meeting on Microwave Photonics (MWP)* , (pp. 186-189). IEEE.

Rivero, C. &. (2008). Lossles audio compression: A case study. *Department of computer and information Science and Engineering, University of Florida*, 08-415.
Rustam, F. S. (2024). Bee detection in bee hives using selective features from acoustic data. *Multimedia Tools and Applications*, 83(8), 23269-23296.
S. Cecchi, A. T. (2018). A preliminary study of sounds emitted by honey bees in a beehive. *Audio Engineering Society Convention* , 144.
Sáenz López, A. (2022). A methodology for the sonic analysis of modern pop music! *Critical listening and its apllication in four top mixes from 2020.*
Schertz Willett, L. C. (1995). A Computer Model for Economic Analysis of Beekeeping Operations. *Bee Economics*.
Sharma, M. J. (2022). A comprehensive empirical review of modern voice activity detection approaches for movies and TV shows. *Neurocomputing,*, 494, 116-131.
Sisario, B. (2019). Amazon Offers Millions of Songs in High-Quality Audio. *Even Neil Young Approves. International New York Times*.
Skoglund, Jan, and Jean-Marc Valin. (2019). Improving Opus low bit rate quality with neural speech synthesis.
Spanias, A., & Atti, V. . (2010). Audio Signal Processing and Coding. . *Wiley*.
Suciu, G. e. (2020). WebRTC role in real-time communication and video conferencing. *Global Internet of Things Summit (GIoTS)*, pp. 1-6.
Sushkov, N. a. (2023). Evaluation of the Quality of Compression of Acoustic Signals in the Low-Frequency and Mid-Frequency Sound Range Using a Neural Network Codec. *International Russian Automation Conference (RusAutoCon)*.
Szczurek, A. M. (2023). Monitoring system enhancing the potential of urban beekeeping. *Applied Sciences*, 13(1).
Takag, K., & Takishima, Y. (2007 July). High speed AAC bit rate reduction on the compressed domain. *IEEE International Conference on Multimedia and Expo* , (pp. 1287-1290). IEEE.
Tashakkori, R. H. (2021). Beemon: An IoT-based beehive monitoring system. *Computers and Electronics in Agriculture*, 190, 106427.
Unep. (May 20, 2019). Why bees are essential to people and planet. *UNEP*.
V. Patel, N. P. (Apr. 2020). Why bees are critical for achieving sustainable development. *Ambio*, vol. 50, 1,.
Valin, J. M. (2016). High-quality, low-delay music coding in the opus codec. *arXiv preprint arXiv*.
van Beurden, M. Q. (Aug. 21, 2022). Free Lossless Audio Codec. *IETF Datatracker*, Available: https://datatracker. ietf. org/doc/draft-ietf-cellar-flac/(visited on 09/18/2022.
Vernyi, K. J. (2024). High Definition Video and Audio Streaming\Across High-rate Delay Tolerant Space Networks. *In AIAA SCITECH 2024 Forum*, p. 1738.
Vogel, A. P. (2024). Optimizing Communication in Ataxia: A Multifaceted Approach to Alternative and Augmentative Communication (AAC). *The Cerebellum*, 1-10.
Vos, K. (2012). Real-Time Communication with Opus Codec. *IEEE Communications Magazine*.
Watkinson, J. (2013). The Art of Digital Audio. *Focal Press*.
Williams, A. (2022). Digital Streaming Services. . *Geo. L. Tech. Rev.,*, 6, 322.
Wolfert, S., Ge, L., Verdouw, C., & Bogaardt, M. (2017). Big data in smart farming. *A review. Agric. Syst*, 153, 69–80.
Wolters, M. K. (2003, October). A closer look into MPEG-4 , High Efficiency AAC. . *Audio Engineering Society.In Audio Engineering Society Convention 115*.
Yang, A. (2022). Smart Bee Colony Monitor. *Clips of Beehive Sounds*.

Yang, M. D., & Su, T. C. (2008). Automated diagnosis of sewer pipe defects based on machine learning approaches. . *Expert Systems with Applications*, 35(3), 1327-1337.

Yang, Yunzhao, et al. (2019). Defining joint embedding distortion for adaptive MP3 steganography. *Proceedings of the ACM Workshop on Information Hiding and Multimedia Security*.

Ye, H. a. (2018). FLAC decoding using GPU acceleration. *IEEE Intl Conf on Parallel & Distributed Processing with Applications*.

Ye, J. K. (2010, October). Vowel recognition based on FLAC acoutic features and subspace classifier. *In IEEE 10th INTERNATIONAL CONFERENCE ON SIGNAL PROCESSING PROCEEDINGS* , (pp. 530-533). IEEE.

Yesodha, K. K. (2024). Intrusion detection system extended CNN and artificial bee colony optimization in wireless sensor networks. *Peer-to-Peer Networking and Applications*, 1-26.

Zacepins, A. B. (2015). Challenges in the development of Precision Beekeeping. *Biosystems Engineering*, 130, 60-71.

Zaman, A., & Dorin, A. (2023). A framework for better sensor-based beehive health monitoring. *Computers and Electronics in Agriculture*.

Železnik, L. S. (2020). Overview of lossless audio codecs.

Zgank, A. (2021). IoT-based bee swarm activity acoustic classification using deep neural networks. *Sensors*, 21(3), 676.

Zhang, T. &. (2001). Audio content analysis for online audiovisual data segmentation and classification. *IEEE Transactions on speech and audio processing*, 9(4), 441-457.

Zhao, K. (2007). *Lossless audio data compression. Acadia University.*

Zheng, F., Zhang, G., & Song, Z. (2001). Comparison of Different Implementations of MFCC. *Comput. Sci. Technol*, 16, 582–589.

Zheng, G. X. (2021). Wav-bert: Cooperative acoustic and linguistic representation learning for low-resource speech recognition. *arXiv preprint arXiv*, 2109.09161.