

Article

Toward Inclusive Smart Cities: Sound-Based Vehicle Diagnostics, Emergency Signal Recognition, and Beyond

Amr Rashed ¹, Yousry Abdulazeem ², Tamer Ahmed Farrag ^{3,4}, Amna Bamaqa ^{5,4}, Malik Almaliki ^{6,4},
Mahmoud Badawy ^{5,4,7,*} and Mostafa A. Elhosseini ^{6,4,7}

¹ Department of Communications and Electronics Engineering, Faculty of Engineering, Mansoura University, Mansoura 35516, Egypt; amr_rashed2@hotmail.com

² School of Computational Sciences and Artificial Intelligence (CSAI), Zewail City of Science and Technology, Giza 12578, Egypt; yousry@mans.edu.eg

³ Department of Electrical Engineering, College of Engineering, Taif University, P.O. Box 11099, Taif 21944, Saudi Arabia; t.farrag@tu.edu.sa

⁴ King Salman Center for Disability Research, Riyadh 11614, Saudi Arabia; abamaqa@taibahu.edu.sa (A.B.); mrmalki@taibahu.edu.sa (M.A.); melhosseini@mans.edu.eg (M.A.E.)

⁵ Computer Science and Information Department, Applied College, Taibah University, Medinah 41461, Saudi Arabia

⁶ Department of Computer Science, College of Computer Science and Engineering, Taibah University, Yanbu 46421, Saudi Arabia

⁷ Computers and Control Systems Engineering Department, Faculty of Engineering, Mansoura University, Mansoura 35516, Egypt

* Correspondence: engbadawy@mans.edu.eg



Academic Editor: Farshid Naseri

Received: 17 February 2025

Revised: 15 March 2025

Accepted: 19 March 2025

Published: 21 March 2025

Citation: Rashed, A.; Abdulazeem, Y.; Farrag, T.A.; Bamaqa, A.; Almaliki, M.; Badawy, M.; Elhosseini, M.A. Toward Inclusive Smart Cities: Sound-Based Vehicle Diagnostics, Emergency Signal Recognition, and Beyond. *Machines* **2025**, *13*, 258. <https://doi.org/10.3390/machines13040258>

Correction Statement: This article has been republished with a minor change. The change does not affect the scientific content of the article and further details are available within the backmatter of the website version of this article.

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Abstract: Sound-based early fault detection for vehicles is a critical yet underexplored area, particularly within Intelligent Transportation Systems (ITSs) for smart cities. Despite the clear necessity for sound-based diagnostic systems, the scarcity of specialized publicly available datasets presents a major challenge. This study addresses this gap by contributing in multiple dimensions. Firstly, it emphasizes the significance of sound-based diagnostics for real-time detection of faults through analyzing sounds directly generated by vehicles, such as engine or brake noises, and the classification of external emergency sounds, like sirens, relevant to vehicle safety. Secondly, this paper introduces a novel dataset encompassing vehicle fault sounds, emergency sirens, and environmental noises specifically curated to address the absence of such specialized datasets. A comprehensive framework is proposed, combining audio preprocessing, feature extraction (via Mel Spectrograms, MFCCs, and Chromatograms), and classification using 11 models. Evaluations using both compact (52 features) and expanded (126 features) representations show that several classes (e.g., Engine Misfire, Fuel Pump Cartridge Fault, Radiator Fan Failure) achieve near-perfect accuracy, though acoustically similar classes like Universal Joint Failure, Knocking, and Pre-ignition Problem remain challenging. Logistic Regression yielded the highest accuracy of 86.5% for the vehicle fault dataset (DB1) using compact features, while neural networks performed best for datasets DB2 and DB3, achieving 88.4% and 85.5%, respectively. In the second scenario, a Bayesian-Optimized Weighted Soft Voting with Feature Selection (BOWSVFS) approach is proposed, significantly enhancing accuracy to 91.04% for DB1, 88.85% for DB2, and 86.85% for DB3. These results highlight the effectiveness of the proposed methods in addressing key ITS limitations and enhancing accessibility for individuals with disabilities through auditory-based vehicle diagnostics and emergency recognition systems.

Keywords: smart cities; intelligent transportation systems (ITSs); machine learning (ML); Bayesian optimization

1. Introduction

Intelligent Transportation Systems (ITSs) play a crucial role in developing smart cities through advanced technologies to enhance the efficiency and sustainability of transportation networks [1,2]. These systems integrate data from sensors, cameras, and GPS devices that provide real-time information on traffic flow, weather conditions, and other relevant factors. ITSs can also use this to adjust the traffic signals dynamically, manage toll road usage, and give drivers personalized route recommendations to reduce congestion and lower travel time. Additionally, ITSs can support autonomous vehicles and shared mobility services, further improving overall urban transportation system performance.

In smart cities, ITSs can also contribute to air quality and greenhouse gas emission reduction by making public transportation, cycling, and walking real options instead of private car travel [3]. This can be achieved through incentives for using sustainable modes of transportation by implementing smart parking systems and congestion pricing schemes, thus decreasing the overall demand for fossil fuel-powered vehicles. This would further allow easy intermodal connections, making urban mobility and the general urban environment even more accessible, sustainable, and inclusive.

The role of vision in systems has become an integral part of modern infrastructure, especially in ITSs, for furthering both safety and efficiency. However, these systems have a significant limitation: they cannot “hear” necessary auditory signals such as emergency sirens, mechanical faults, and environmental hazards [4]. There are various limitations of vision-based systems, such as the inability to detect auditory signals [5], environmental noise interference [6], mechanical fault detection [7]. Other than emergency signals and mechanical faults, environmental hazards such as construction noise, falling debris, or wildlife may create serious dangers. In this regard, the vision-based systems cannot detect such hazards until they become visually apparent, which may be too late to act effectively. Thus, sound-based diagnostics can provide early warnings against such dangers and enhance safety. Sound-based diagnostics can improve the reliability and efficiency of existing systems manifoldly. There are several reasons that sound-based diagnosis is highly respected, including enhanced situational awareness [8,9], real-time monitoring and alerts [10], cost-effectiveness [11], accessibility for hearing-impaired people [12], and improved emergency response [12].

ITSs are designed to improve the functionality and safety of transport systems using advanced technologies [13]. Whereas these systems are designed to streamline the process, people with disabilities face unique barriers to their movement and access to essential services in such contexts. These include, but are not limited to, information inaccessibility, lack of warnings in emergencies, navigation barriers, inadequate access to vehicles, and communication barriers. Sound-based systems are essential in offering complementarities and ensuring increased access within ITS environments based on alternative means of communication and information sharing. Some of the possible solutions include the following: audio cues for navigation, visual alerts for the hearing-impaired, sound detection for vehicle fault detection, emergency sirens and alerts, and improved communication systems [14]. ITS application covers a wide range of fields in implementing sound detection. For example, this will help in emergency response for quick and timely mitigation of hazards. In the field of public transportation, increased safety and efficiency for passengers may also be achieved. Smart cars use it to enhance driver awareness and the vehicle’s behavior.

Recent advances in artificial intelligence (AI) have paved the way for sophisticated data analysis techniques that drive innovative applications across various fields, including sound-based diagnostics. Within this AI framework, machine learning (ML)—defined as using algorithms and statistical models that enable computer systems to learn from data and improve their performance on specific tasks without explicit programming—has emerged

as a key enabler. By leveraging ML, our study can analyze complex auditory signals from vehicle faults and environmental sounds, transforming raw data into actionable insights for Intelligent Transportation Systems (ITSs).

We begin our approach with foundational ML models that provide robust baseline performance. Techniques such as Logistic Regression (LR) and k-nearest neighbors (kNN) are utilized to establish initial classification capabilities, forming the groundwork for further enhancements. These basic models are critical for understanding the underlying patterns in the audio data and serve as benchmarks against which more sophisticated methods can be compared.

The study introduces three novel datasets that capture various sounds relevant to Intelligent Transportation Systems. The first dataset (DB1) consists of 27 distinct vehicle fault classes featuring critical sounds such as Engine Misfire, Fuel Pump Cartridge Fault, Radiator Fan Failure, and Strut Mount Failure, all directly generated by the vehicle. The second dataset (DB2) comprises 22 environmental sound classes, including emergency signals like sirens and various transportation-related and ambient environmental noises. These datasets provide a rich collection of auditory signals that form the basis for robust sound-based diagnostic systems. The third dataset (DB3) also merges DB1 and DB2 to create a comprehensive collection of 49 classes. This unified dataset enables the framework to classify any sound from vehicle faults or external environmental events into the correct category. The study lays the groundwork for advancing machine learning research in sound-based diagnostics by addressing the scarcity of specialized, publicly available auditory datasets. It contributes to the development of more inclusive and responsive ITS applications.

Building on these fundamentals, our framework incorporates advanced variants of ML to address the challenges of differentiating acoustically similar classes. Ensemble methods such as AdaBoost, Random Forest (RF), and Gradient Boosting (GB) enhance accuracy by combining the strengths of multiple weak learners. Additionally, Support Vector Machines (SVM) and Stochastic Gradient Descent (SGD) optimize decision boundaries in complex feature spaces, while Decision Trees (DTs) provide interpretable classification logic. Complementing these are the CN2 algorithm and Naive Bayes (NB), which handle complex rule-based classification and probabilistic inference. Together, these diverse ML techniques form a comprehensive diagnostic system capable of robust performance in real-world ITS applications.

Integrating auditory intelligence into ITSs addresses several key research challenges, notably the difficulty of distinguishing acoustically similar classes—such as Universal Joint Failure versus Bad CV Joint and Knocking versus Pre-ignition Problem. These challenges demand advanced feature extraction techniques and robust machine learning models that capture subtle differences in sound signatures. Additionally, the scarcity of specialized, publicly available auditory datasets has historically hindered progress in this area; this research overcomes that barrier by introducing a comprehensive, curated dataset that serves as a benchmark for future work.

This study addresses a critical gap in Intelligent Transportation Systems (ITSs) by explicitly defining its aim to detect faults directly from sounds generated by vehicles, such as engine or brake noises, and to classify external alert sounds, including emergency sirens. The intended applications of these predictive outputs are articulated, emphasizing their role in real-time diagnostics for smart vehicle systems and providing auditory-to-visual alert conversions to assist sound-impaired drivers. Additionally, the study highlights the potential of auditory capabilities to enhance vehicle fault detection and accessibility for individuals with disabilities while addressing the scarcity of specialized datasets in this domain. This is achieved through the following key contributions:

- Introducing a novel dataset comprising vehicle fault sounds, emergency sirens, and environmental noises filling a critical gap in publicly available resources.

- Developing a comprehensive methodology for audio preprocessing, including normalization, resampling, and segmentation.
- Proposing robust feature extraction techniques, such as Mel Spectrograms, MFCCs, and Chromatograms, enabling compact and expanded feature representations.
- Evaluating multiple ML models in the first scenario, including neural networks, Logistic Regression, and Random Forests.
- Proposing a Bayesian-Optimized Weighted Soft Voting with Feature Selection (BOWSVFS) approach in the second scenario, achieving a classification accuracy of 91.04% on the car fault dataset (DB1) and outperforming the first scenario's results.
- Demonstrating the relevance of sound-based ITSs in promoting accessibility by offering real-time alerts and auditory-to-visual conversion solutions for individuals with disabilities.
- Aligning sound-based diagnostics with broader smart city goals, contributing to the development of safer and more inclusive transportation systems.

This research establishes a strong foundation for integrating auditory intelligence into ITSs, with significant implications for safety, accessibility, and inclusivity in smart cities. The framework demonstrates strong performance overall, with several fault classes being recognized with near-perfect accuracy. For example, classes such as Engine Misfire, Fuel Pump Cartridge Fault, Radiator Fan Failure, Strut Mount Failure, and Suspension Arm Fault consistently achieve 100% accuracy in many cases. This indicates that the framework effectively captures the distinct acoustic signatures associated with these faults. However, challenges remain for acoustically similar classes. Universal Joint Failure, for instance, is occasionally misclassified—often confused with Bad CV Joint—while Bad Wheel Bearing also shows minor misclassifications. More notably, the Knocking and Pre-ignition Problem classes face significant difficulties, with Pre-ignition Problem instances frequently being predicted as Engine Misfire. These misclassifications highlight the areas where further refinement in feature extraction or model tuning may be necessary to better differentiate between closely related acoustic patterns.

The structure of this paper is as follows: A summary of the current literature is given in Section 2, along with potential directions for further research. Section 3 introduces the datasets. Section 4 focuses on materials and explains the proposed methodology. Section 5 provides an overview of the experiments, including the experimental setup, methods, and findings collected, focusing on the performance metrics attained. The overall discussion in Section 6 wraps up with conclusions and future work, summarizing the paper's key contributions and suggesting directions for subsequent research in this domain.

2. Literature Review

This section discusses earlier attempts at sound recognition and sound-based defect detection in machinery, vehicles, trains, and aircraft systems. The majority of the methods assessed were developed using ML methods. Some, meanwhile, are more recent and rely on deep learning or vision transformers.

Nasim et al. introduced a sound-based early fault detection system for vehicles utilizing ML technology [15]. This system is specifically designed to target the faults in vehicles at their initial stages by analyzing the sound emitted by the vehicle. The system starts working by binary classification, which can decide whether the vehicle is faulty or healthy. They utilized time domain, frequency domain, and time–frequency domain features to detect normal and abnormal vehicle conditions effectively. Additionally, they employed abnormal vehicle data to classify them into fifteen other typical vehicle issues. Through experimentation, the random forest algorithm yielded the best accuracy of 97% for fault detection and 92% for problem classification when utilizing time–frequency features.

Hamad et al. proposed a rule-based ML technique that automatically detects engine problems [16]. The generalizability of the system is considered by time domain, frequency domain, and time–frequency domain features. The robustness of the developed system is evaluated using noisy sound data collected under various normal and abnormal conditions. The experimental results demonstrated that the approach outperformed other techniques by 2.6–6.0% and yielded the highest performance accuracy of 98.6%. Yildirim et al. proposed a testing and evaluation procedure on the sound quality of two types of cars [17]. The sound quality is analyzed through the car’s road running test on the provided ground with varying running speeds. They proposed a neural network predictor to model the system for possible experimental applications. In their experiments, only objective factors of loudness, sharpness, speech intelligibility, and sound pressure level are considered essential for sound quality. The computer simulations and experiments show evidence that the neural predictor algorithm provides reasonable accommodation in different cases and allows superior prediction in two-car sound analysis.

Mel-Frequency Cepstral Coefficients (MFCCs), DWT-based features, and the Extreme Learning Machine (ELM) classifier were employed in the vehicle problem diagnostic system that Akbalik et al. presented [18]. The proposed framework uses a big, diversified dataset that includes many vehicle models and real-world operating situations. The experiment results show that the MFCC-based features combined with the ELM classifier outperform the others in terms of accuracy, precision, recall, F1-score, macro F1-score, and weighted F1-score, which are 92.17%, 92.24%, 92.22%, 92.10%, and 92.06%, respectively. Murovec et al. created an acquisition system using the Zero-Crossing Signature (ZCS) technique [19]. To accomplish precise engine type classification, the study used a unique level-crossing (ZCS) feature that demonstrated excellent performance in differentiating engine sounds from surrounding noise. A dataset of 417 vehicle recordings was examined, and the classification performance of the ZCS was compared to the traditional Zero-Crossing (ZC) technique utilizing a Self-Organizing Map (SOM) with a 1D grid of nine neurons. Wang et al. proposed a method for diagnosing engine acoustic signal faults using multi-level supervised learning and time-frequency transformation [20]. First, it decomposes the fault diagnostic problem into feature augmentation, fault detection, and identification. Second, based on several time–frequency studies, it proposes an adaptive fault feature band extraction approach aimed at distinct features from different vehicle data. Finally, a frequency band attention module was designed to focus on the most meaningful frequency range to the characteristics of engine failure.

Boztas et al. proposed a learning model for improving machine fault classification using handcrafted attributes [21]. The approach utilized texture and statistical features in classifying faults with high performance. They developed a hybrid and multilevel feature extraction technique that maintains high efficiency while lowering the complexity associated with deep learning frameworks. Using a Chi2 feature selector to eliminate redundant features, the model focused on the most informative features throughout the classification step. In the MIMII (noisy) dataset, the proposed model effectively classified more than 90% of the five cases. A Variational Autoencoder/Convolutional Neural Network (VAE-CNN) was created by Wang et al. to diagnose rolling bearing faults [22]. The model was developed to extract complex vibration signal features to detect and categorize faults. While the CNN component increases the expressiveness of signal data and successfully handles issues like gradient vanishing and explosion, the VAE component improves noise robustness. The diagnostic accuracy of the VAE-CNN model for various fault types at varying rotational speeds typically exceeded 90%, yielding generally satisfactory diagnostic results. Xinwen Guo developed a defect diagnostic approach based on feature extraction and a word bag model using acoustics and vibration engineering science theories [23]. This approach mainly expands the three-layer structure of the word bag model and constructs

codebooks for each layer's feature vectors based on this model. Thereafter, it develops the failure detection system of a rolling bearing based on the adaptive extended word bag model. The findings revealed that the defect detection technique has excellent diagnostic accuracy and stability, offering dependable technical assistance for regular operation and safe mechanical equipment maintenance.

Li et al. developed a defect diagnosis system for railway turnout switch machines based on sound signals [24]. The method used Eigenmode Decomposition to improve the sound signal, reduce noise, and extract important statistical information from the time and frequency domains. The ReliefF algorithm is used for feature selection, dimension reduction, and fault classification with weighted parameters to address redundant information in high-dimensional features. The selected feature parameters are then utilized to train the Support Vector Machine. The results showed a defect diagnostic accuracy of 98% in the positioning work mode and 95.67% in the reversing work mode. Kreuzera et al. proved that diagnosing bearing defects in railway vehicles using aerial sound data is possible, even in complex real-world settings [25]. To that purpose, many characteristics are investigated, including Mel Frequency Cepstral Coefficients (MFCCs), which are best suited for diagnosing bearing problems by analyzing airborne sound. The MFCCs were utilized to train an MLP classifier. The suggested technique is assessed using real-world data from a cutting-edge commuter train car in a dedicated measurement effort. The classification results showed that the chosen MFCC features allowed for the reliable detection of bearing defects, including those not included in the training. Eunsun Yun and Minjoong Jeong proposed a feature extraction technique for fault sound identification in EPS motors [26]. This technique reduced the feature dimensionality while preserving the original raw waveform, which is crucial to maintaining the essential features in the waveform for anomaly detection. They combined DFMT with MFCC to optimize feature extraction. They applied LSTM-AE to classify data by segregating standard data from abnormal ones using reconstruction error metrics. The experimental results of the proposed method were proved efficient with an accuracy of 99.2%, recall of 94.0%, precision of 95.6%, and F1-score of 94.7%.

A sound-based engine classification was proposed by Shajie et al. to detect flaws in engine ball bearings [27]. They used sound-based component extraction techniques to find reoccurring patterns across time. They proposed modifications to the ResNet and hybrid CNN models based on the NASA-bearing dataset. To identify TIM-bearing faults, they employed time and frequency features that may be inferred from the signals and their spectra. The experiments considered realistic scenarios found in real-world industrial settings. They gained insights into the method's performance with reasonable accuracy rates. To improve industrial productivity and minimize machinery downtime, Khan et al. developed a technique for diagnosing robotic manipulator faults using motor sound analysis [28]. They investigate the efficiency of deep learning and conventional ML in detecting motor abnormalities using a dataset created with a specifically designed robotic manipulator. It obtained an F1-score of over 92%, outperforming the traditional methods significantly, hence proving the potential of sound analysis for automatic defect identification in robotic systems using the proposed custom CNN and 1D-CNN models. Kim et al. proposed a deep denoising autoencoder method to filter out various industrial noise levels from audio data [29]. They applied unsupervised learning models for rapid and accurate anomaly detection. They preprocessed audio data to adapt the denoising technique to the noise levels of different industrial contexts. Several experiments using different industrial equipment types demonstrated the proposed technique's effectiveness, efficiency, and processing speed. Senanayaka et al. diagnosed machinery defects by isolating audio sources from complex mixtures of sound waves [30]. First, they activated fault sound isolation and separated distinct fault noises from a complicated blend of

sound signals. Then, the isolated fault noises were passed through a 1D-CNN classifier to ensure correct classification. A machine fault simulator by Spectra Quest equipped with a condenser mic was employed to evaluate the proposed model. To improve early vehicle defect recognition, Hameed et al. investigated the application of ML for real-time engine knocking detection [31]. They analyzed several machine-learning techniques and retrieved frequency modulation amplitude demodulation features from engine sound data. With a classification accuracy of 66.01%, the coarse decision tree approach proved the most successful. The accuracy was then increased by employing deep learning models; a deep learning recurrent neural network (RNN) model in LSTM attained 90% accuracy.

Naryanto et al. developed a deep learning model to detect and classify damage or defects in diesel engines using artificial neural networks and convolutional neural networks [32]. They utilized the DEFault dataset, which has 3500 rows of data organized under four distinct labels. Results showed that ANN outperformed CNN for noisier datasets, but it outperformed for less noisy datasets. Yuan et al. proposed a defect detection approach for new energy vehicle engines using wavelet transforms and Support Vector Machines [33]. First, an abnormal noise signal identification model for vehicle engine faults is developed, and the time–frequency parameters of the basis function are adaptively changed. The engine surface radiation noise is then split into the inner mechanical and battery excitation components. The new energy vehicle engine failure signal was decomposed using feature decomposition and multiscale separation. Furthermore, fuzzy clustering and time–frequency analysis of fault signals in the fractional Fourier domain were used to detect faults in new energy vehicle engines. Chu et al. proposed an intelligent identification model for diesel engine faults based on mixed attention [34]. They proposed a multi-cylinder whole-machine fault diagnosis model that integrates 1D-CNN with self- and mutual attention mechanisms. Single-cylinder sensor data were integrated using self-attention in the model, and signal features of each cylinder were fused using the mutual attention mechanism. Simultaneously considering the mechanism knowledge of cylinder structural consistency and signal time delay similarity, this approach utilized single-cylinder fault data to develop a comprehensive fault recognition model for all cylinders. The average diagnosis accuracy reached 100% in known fault data and about 96.65% in unknown fault data.

Lee et al. proposed a bearing failure detection using an LSTM autoencoder with self-attention based on graph convolution networks [35]. Accordingly, they trained their model using data from the Fault Simulator Testbed and the Case Western Reserve University dataset. Results demonstrated that the proposed model attained an accuracy of 97.3% and 99.9%, respectively, in the CWRU dataset and Fault Simulator Dataset. Using a single microphone and a data-driven approach, Spadini et al. developed a model for intelligent fault diagnosis in rotating equipment that successfully identified 42 classes of defect kinds and severities [36]. They considered reliable data from the unbalanced MaFaulDa dataset to balance high performance and minimal resource consumption. The model achieved remarkable performance in terms of the analysis by time, frequency, mel-frequency, and statistical parameters with an accuracy of 99.54% and F-Beta of 99.52%. Using sound samples, Gantert et al. proposed a multiclass method for identifying anomalous samples in industrial machinery [37]. Integrating binary models commonly found in the literature aims to improve the model's generality while decreasing the number of classifiers. Using MIMII and ToyADMOS, two industrial sound datasets, they compared the proposed multiclass models with the binary alternative. Experiments revealed that 98% of the Toy-ADMOS dataset and 93% of the MIMII dataset were correctly classified. Table 1 summarizes the papers mentioned in this study highlighting the main characteristics and problems of each one.

Table 1. Comparison between literature review papers.

Study	Domains of Usage	Underlying Methodologies	Dataset	Size	Type	Classification	Accuracy	Problems
[15]	Vehicles	ML	Recordings	351	Car sounds	15 classes	92%	Performance depends on the quality and representativeness of the training data.
[16]	Vehicles	ML	Recordings	555	Car sounds	15 classes	98.6%	Difficult to adapt to new or unexpected problems; require extensive manual tuning.
[17]	Vehicles	NN	Recordings	2	Two Cars	Binary	$R^2 = 0.99$	Limited to the specific sound quality factors considered; may not capture subjective aspects of sound quality; requires labeled data for training.
[18]	Vehicles	ML	Recordings	280	Car sounds	6 classes	92.17%	Sensitive to the choice of hidden nodes; require careful tuning; may not capture all relevant information.
[19]	Vehicles	NN	Recordings	417	Car sounds	Binary	$F1 = 0.86$	Sensitive to noise and may not generalize well to unseen engine types; require careful parameter tuning.
[20]	Engines	DL	Recordings	100	Vehicles/ Induction motors	6 classes	$F1 = 0.95$	Time-frequency transformation can be computationally expensive; performance depends on the effectiveness of the adaptive fault feature band extraction.
[21]	Industrial	ML	MIMII	5101	Machine sounds	5 classes	95%	Handcrafted features may not capture all relevant information; performance may be limited by the quality of the feature selector.
[22]	Rolling bearing	VAE-CNN	CWRU	2048	Rolling bearing	10 classes	96.62%	VAE-CNN models can be complex and computationally expensive to train; require a large amount of data to achieve good performance.
[24]	Railway	ML	Recordings	1600	Turnout switch machine	10 classes	98%	Eigenmode Decomposition can be computationally expensive; SVMs can be sensitive to the choice of kernel and parameters.
[25]	Railway	ML	Recordings	25,000	Commuter train dataset	Binary	97.04%	MFCCs may not capture all relevant information in the sound signal; MLP classifiers can be prone to overfitting.
[26]	EPS motors	LSTM-AE	Recordings	29,759	Rolling bearing	Binary	99.2%	LSTM-AE models can be complex and computationally expensive to train; performance depends on the quality of the reconstruction error metric.
[27]	Vehicles	CNN	NASA Bearing Dataset	9463	Engine Ball Bearing	Binary	91%	CNN models can be data-hungry and computationally expensive to train; performance depends on the choice of network architecture and hyperparameters
[28]	Robotic manipulator	CNN	Recordings	181	Motors	Binary	92.34%	Custom CNN models may be difficult to generalize to other robotic manipulators; performance depends on the quality of the custom dataset.

Table 1. Cont.

Study	Domains of Usage	Underlying Methodologies	Dataset	Size	Type	Classification	Accuracy	Problems
[29]	Industrial	AE	MIMII	5101	Machine sounds	Binary	96.51%	Denoising autoencoders may require careful tuning of the noise level; performance depends on the characteristics of the industrial noise.
[30]	Industrial	CNN	Recordings	60 stem files	Engine Ball Bearing	4 classes	99.58%	Audio source isolation can be challenging in complex environments; 1D-CNN models may not capture all relevant spatial information in the sound field.
[31]	Vehicles	DL	Recordings	153	Engine Sounds Knocking	Binary	90%	LSTM models can be computationally expensive; performance depends on the quality of the frequency modulation amplitude demodulation features.
[32]	Diesel engines	CNN	DEFault	3500	Engine Sounds	4 classes	99.37%	ANN performance can be sensitive to parameter initialization and network architecture; CNN performance dependent on the noise level; dataset size.
[33]	New energy vehicles	ML	Recordings	N/A	Engine Sounds	Binary	90%	Wavelet transforms can be computationally expensive; SVMs can be sensitive to the choice of kernel and parameters.
[34]	Diesel engines	Mixed attention	N/A	N/A	Engine Sounds	Binary	98.17%	Complexity, computational cost, reliance on single-cylinder data for whole-machine diagnosis.
[35]	Rolling bearing	AE with Self-Attention	CWRU	N/A	Bearing fault	N/A	97.3%	Complexity, computational cost, potential overfitting, and reliance on specific datasets.
[36]	Industrial	ML	MaFaulDa	1951	Machine sounds	6 classes	99.54%	Complexity, computational cost, difficulty generalizing to new datasets.
[37]	Industrial	ML	MIMII	5101	Machine sounds	4 classes	93%	Possible overfitting for limited datasets, computational cost for multiclass integrations, and difficulty generalizing to new datasets.
			ToyADMOS	N/A		3 classes	98%	

Research gap: While Intelligent Transportation Systems have advanced significantly through vision-based technologies, a critical gap exists in integrating sound-based fault detection mechanisms. This gap is particularly evident in three areas: (1) the limited development of audio-based diagnostic systems that utilize real-time analysis of vehicle-generated sounds (e.g., engine or brake noises) and external emergency alert sounds (e.g., sirens), (2) the scarcity of comprehensive public datasets designed explicitly for vehicle sound analysis, and (3) insufficient attention to accessibility needs for individuals with disabilities within ITS frameworks. These limitations hinder the development of more inclusive and comprehensive transportation monitoring systems. To address these limitations, this study aims to develop a comprehensive dataset that serves as a “conscious ear” for intelligent systems in modern cities, vehicles, and transportation networks. This effort seeks to enhance the auditory capabilities of smart systems, enabling them to respond effectively to complex auditory scenarios, thereby enhancing safety and functionality in various applications.

3. Dataset Creation

The main problem with transportation sound-based fault diagnosis is the availability of datasets. Therefore, data from various sources are collected to create a tailored dataset for car sound analysis. Reliable audio samples are built by downloading videos from YouTube related to car faults, animal sounds, car crashes, siren sounds, etc. After this step, the model splits videos into segments and extracts those sections that may contain the target audio. Then, it converts the files into audio format. To expand the dataset, additional audio is supplemented from Kaggle datasets: FSC22 [38], Google AudioSet [39], Audio Classifier Dataset [40], Sound Classification of Animal Voice [41], and Vehicle Sounds Dataset [42]. Lastly, the model ensures that every sample is labeled and verified.

In this vein, the dataset was created and reviewed using a combination of publicly available datasets and real-world recordings, covering a wide range of vehicle faults, crashes, emergency sirens (police, ambulance, fire truck), wild animal sounds, car and truck horns, and other environmental road sounds. This approach ensures a diverse and realistic dataset that enhances model performance in detecting road-related events. The dataset creation procedure involves different stages as shown in Figure 1.

3.1. Data Collection and Annotation

We carefully selected publicly available datasets that include real traffic scenarios and vehicle fault cases. Additionally, we extracted relevant frames and sequences from YouTube videos, ensuring a diverse representation of traffic conditions and vehicle behaviors. Each data sample was manually labeled based on predefined criteria, focusing on vehicle states, traffic interactions, and specific fault conditions.

3.2. Expert Review and Validation

To enhance reliability, domain experts with extensive experience in automotive engineering and machine learning reviewed the dataset. The experts cross-checked and validated the labels to ensure accuracy and consistency with real-world vehicle behaviors.

3.3. Publicly Available Datasets Referenced

We utilized multiple datasets containing wild animal sounds, vehicle faults, and environmental noises to build a comprehensive dataset. The key datasets referenced include:

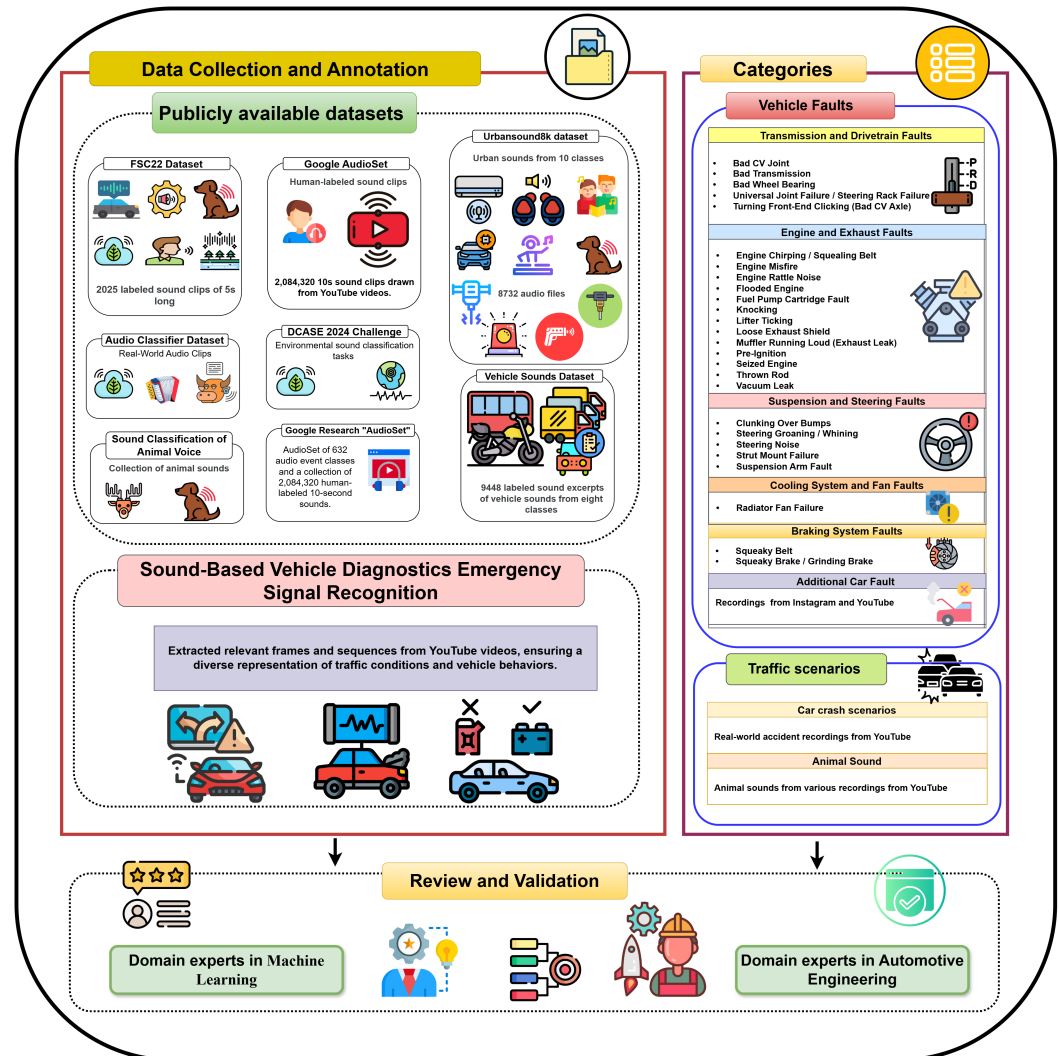


Figure 1. Dataset creation.

1. FSC22 Dataset: A collection focused on various sound categories, including vehicle sounds and environmental noises, useful for sound classification models [38].
2. Google AudioSet: A large-scale collection of audio data across thousands of categories, aimed at improving sound classification models [39].
3. Audio Data: A dataset containing diverse audio clips across various categories, useful for developing classification models [40].
4. Sound Classification of Animal Voice: A dataset containing sounds from different animals, useful for animal sound classification tasks [41].
5. DCASE 2024 Challenge: A dataset designed for the DCASE 2024 challenge, covering environmental sound classification tasks [43].
6. UrbanSound8K Dataset: Contains 8732 labeled sound excerpts from urban environments, categorized into 10 classes such as car horns and sirens [44].
7. AudioSet by Google Research: A vast dataset with over 2 million human-labeled 10 s sound clips spanning thousands of audio categories [45].
8. Vehicle Sounds Dataset: Contains various vehicle sounds useful for training models focused on transportation-related sound classification [42].

This dataset has been meticulously designed and validated to provide a diverse and realistic representation of road-related sounds, ensuring high-quality training data for machine learning models in the domain of automotive fault detection, traffic event classification, and environmental sound analysis. To standardize the process, files should

have the same duration; however, this is not the case. Preprocessing is performed as a solution to ensure samples of the same duration. Algorithm 1 contains the pseudo-code for this stage. The primary steps in the algorithm are:

1. Repeat the audio until it achieves the required duration.
2. Normalize the audio to standardize levels.
3. Resample the audio to a consistent sampling rate (e.g., 16 kHz or 44.1 kHz).
4. Segment lengthy recordings into shorter clips (e.g., 2–5 s each).

Algorithm 1: Pseudo-code for Audio Preprocessing Script

```

1 Input: Input audio file ( $F$ )
2 Output: Preprocessed Audio segments ( $F_p$ )
3  $rate \leftarrow$  Target sample rate = 16 kHz
4  $dur \leftarrow$  Audio segment duration = 2500 ms
5  $dur_{min} \leftarrow$  Audio file minimum duration = 10 s
6 if ( $dur(F) < dur_{min}$ ) then
7    $rep \leftarrow \lceil dur_{min} / dur(F) \rceil$  // No. of repetitions
   // Extend Audio to Minimum Duration
8   for  $i \leftarrow 0$  to  $rep$  do
9      $F_{Ext} \leftarrow F_{Ext} + F$ 
10   $F_{Ext} \leftarrow trim(F_{Ext}, dur)$ 
11 else
12   $F_{Ext} \leftarrow F$ 
13  $F_{norm} \leftarrow normalize(F_{Ext})$  // Normalize audio levels
14  $F_{norm} \leftarrow resample(F_{norm}, rate)$  // Resample audio to TARGET SAMPLE RATE
15 if  $F_{norm} > dur_{min}$  then
16   $F_p \leftarrow segment(F_{norm}, dur_{min})$ 
17 else
18   $F_p \leftarrow F_{norm}$ 
19 Return  $F_p$ 

```

4. Methodology

This study aims to develop an advanced sound-based early diagnosis system to support Intelligent Transportation Systems (ITS) by enabling real-time detection of vehicle faults and identification of emergency sounds. The main steps of this system are illustrated in Figure 2. To achieve this objective, addressing the primary challenges encountered in this field is essential, starting with the absence of comprehensive public datasets specifically designed for vehicle sound analysis. The initial phase of the proposed model involves the creation of a dataset that contains recordings of car fault sounds, emergency sirens, and ambient noises. This process includes audio data collection and preprocessing. Subsequently, the most significant features are extracted in two versions: a compact version with 52 features and an expanded one with 126 features.

In the final step, both sets of extracted features are classified using 11 distinct ML models. Another phase of optimization is provided by the system to enhance the accuracy of classification. It utilizes the best ML models with the highest-ranked features to build an ensemble optimization model. The following subsections provide further details on each stage of this model.

The key steps in our audio preprocessing pipeline to provide further insight into how the system operates are as follows:

1. Fixed Time Windows for Feature Extraction:

- We preprocess the raw audio files by extending them to a minimum duration of 10 s ($\text{MIN_DURATION_MS} = 10,000 \text{ ms}$), normalizing their levels, and resampling them to a target sample rate of 16 kHz ($\text{TARGET_SAMPLE_RATE} = 16,000 \text{ Hz}$).
 - The preprocessed audio is then segmented into fixed-length clips of 2.5 s ($\text{SEGMENT_DURATION_MS} = 2500 \text{ ms}$).
 - Only segments meeting the required length are retained for further processing.
2. Time Window Length Determination:
- The nature of the vehicle fault guided the choice of segment duration sounds, which are typically periodic and repetitive over short durations.
 - To ensure consistency across all samples, we set a minimum duration of 10 s for all audio files. If an audio file is shorter than this threshold, it is repeated and trimmed to match the minimum duration.
 - The fixed 2.5 s time window used for feature extraction ensures that features such as MFCCs, Mel Spectrogram, and Chroma Features capture sufficient temporal and spectral characteristics of the sounds.
3. Sliding Factor Consideration:
- A fixed windowing approach is used over overlapping sliding windows during segmentation. This ensures non-redundant segments while maintaining dataset balance.
 - However, future work could explore the impact of using overlapping windows to capture more temporal variations while controlling data redundancy.

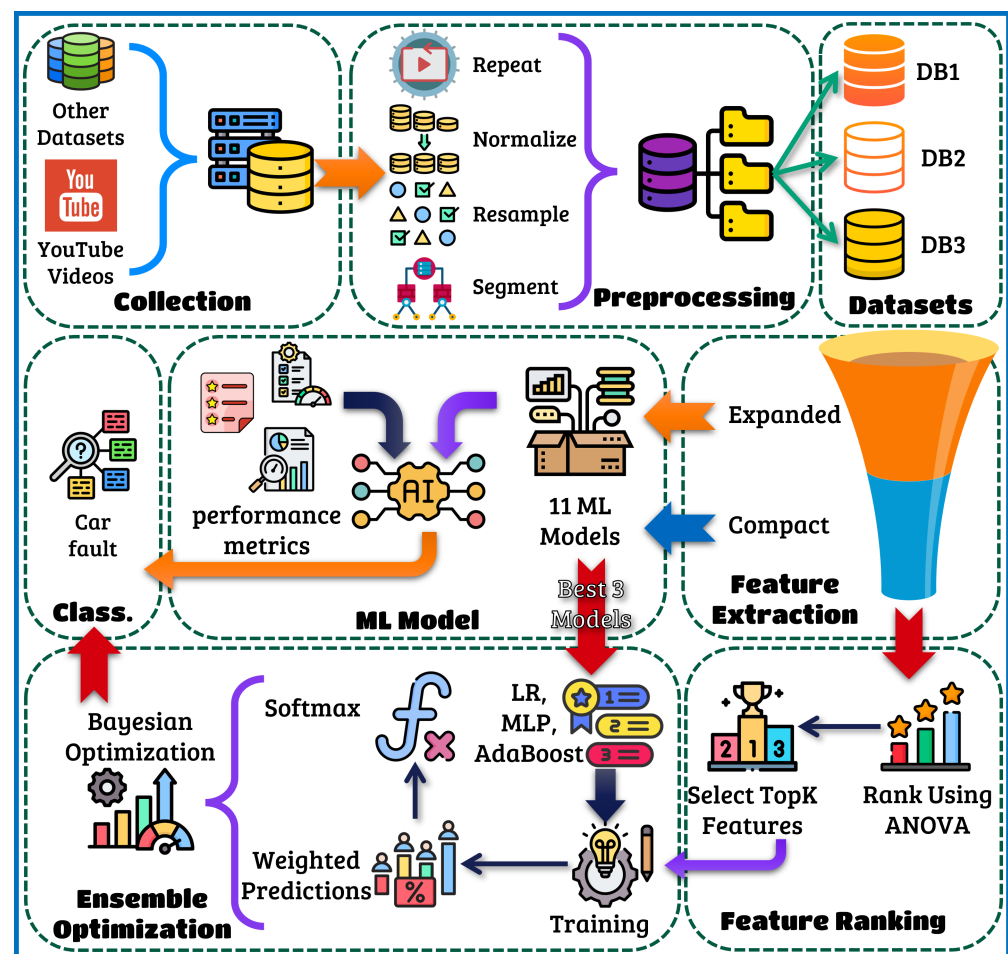


Figure 2. Main system phases and steps.

This structured approach ensures that the extracted sound features represent the fault categories well while maintaining computational efficiency.

By the end of this phase, audio files are sampled, labeled, and normalized to build the dataset. Three datasets are created: the first one contains car faults (DB1) with 133 audio files and 27 distinct classes, the second dataset contains other sounds (DB2) with 1031 audio files and 22 distinct classes, and the third dataset is a merged version between the latter two (DB3) with 1164 audio files and 49 different classes. Tables 2 and 3 show the labels and the corresponding file counts for DB1 and DB2, respectively.

Table 2. DB1 labels and file counts.

Labels	Count	Labels	Count
Bad Wheel Bearing	21	Squeaky Belt	4
Universal Joint Failure or Steering Rack Failure	10	Seized Engine	4
Knocking	5	Pre-Ignition	4
Wheel Bearing, Transmission Whining Noise and Catalytic Converter Issues	5	Bad Transmission	4
Bad CV Joint	4	Strut Mount Failure	4
Radiator Fan Failure	4	Lose Exhaust Shield	4
Turning Front End Clicking Bad CV Axle	4	Lifter Ticking	4
Steering Noise	4	Flooded Engin	4
Steering Groaning Whining Low Power Steering Fluid	4	Engine Rattle Noise	4
Squeaky Brake/Grinding Brake	4	Engine Misfire	4
Muffler Running Loud Exhaust Leak	4	Thrown Rod	4
Clunking Over Bumps Bad Stabilizer Link Noise	4	Suspension Arm Fault	4
Engine Chirping/Squealing Belt	4	Vacuum Leak	4
Fuel Pump Cartridge Fault	4	Total (27)	133

Table 3. DB2 labels and file counts.

Group	Labels	Count	Group	Labels	Count	Group	Labels	Count
Animals	Cats	200	Vehicles and Transportation	Car Crashes	103	Emergency Vehicles	Police Car Siren	41
	Sheep	80		Car Horn	24		Fire Truck Siren	37
	Bear	68		Motorcycle	20		Ambulance Siren	30
	Dog	68		Bus	20	Construction and Machinery	Drilling	24
	Monkey	60		Bike	20		Gunshot	14
	Lions	48		Train	20	Weapons and Explosions		
	Wolf	47		Truck	20			
	Horse	40		Truck Horn	19	Total (22)		1031
	Mouse	28						

4.1. Feature Extraction

After preparing the datasets, the next stage in the proposed system is feature extraction. Audio feature extraction is a significant task in processing an audio signal for the purpose of sound classification. From an audio signal, meaningful features can be extracted to analyze and understand the content of the audio. Figure 3 shows some key features commonly extracted from audio signals.

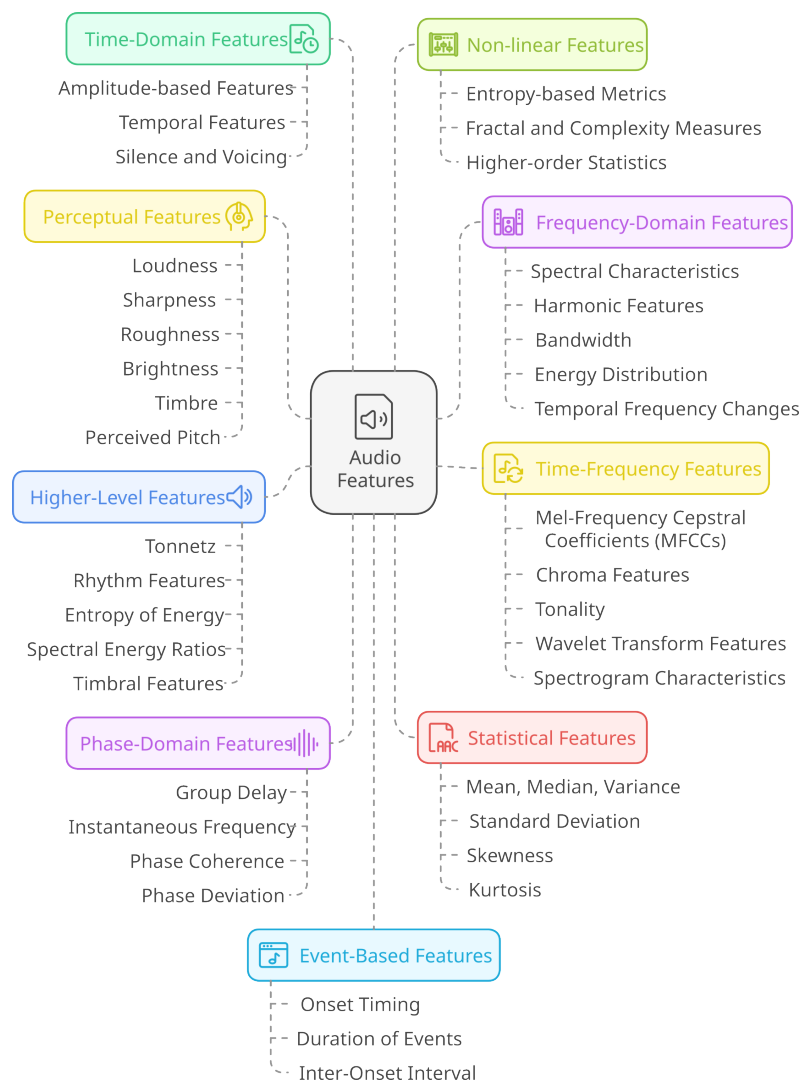


Figure 3. Common audio features.

The essential features in our study are extracted in two versions: a compact version with 52 features and an expanded one with 126 features. In the compact version, Mel Spectrogram [46], MFCCs [47], and Chroma Features [48] were used. Figure 4 shows an example of a Mel Spectrogram.

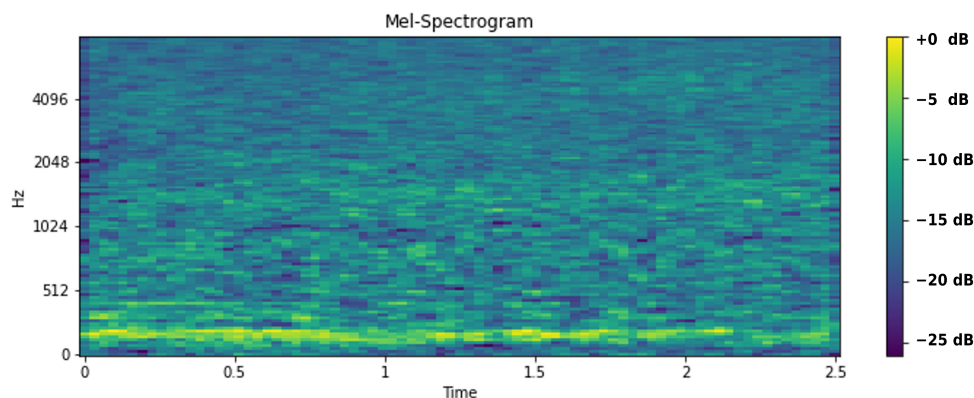


Figure 4. Example of a Mel Spectrogram of a bad CV joint.

For generating the expanded version, Spectral Features [49], Zero-Crossing Rate [19], Root Mean Square Energy (RMSE) [50], Chroma Features, MFCCs, and Extended MFCCs [51] were used. Table 4 defines these features, including their counts and the version(s), Compact (C) or Expanded (E), they appeared in. Figure 5 shows two-dimensional data projection DB1 compact features.

Table 4. Features extracted in the study.

Features Domain	Feature	Definition	Count	Usage
Time-Domain	Zero-Crossing Rate	How often the signal changes from positive to negative.	2	E
	RMSE	Measures the energy of the audio signal.	2	E
Frequency-Domain (Spectral Features)	Spectral Centroid	Indicates where the center of mass of the spectrum is located.	2	E
	Spectral Bandwidth	Measures the width of the spectrum around its centroid.	2	E
	Spectral Roll-off	The frequency below which 85% of the total spectral energy resides.	2	E
	Spectral Contrast	Refers to the difference in amplitude between peaks and valleys in the spectrum of an audio signal.	14	E
Time-Frequency	Mel Spectrogram	A visual representation of the frequency spectrum of an audio signal over time.	2	C
	Chroma Features	Represent the energy distribution across the 12 different pitch classes (notes) of the Western music scale.	24	C & E
	MFCCs	Capture spectral features related to the timbre of audio. They represent the short-term power spectrum of sound.	26	C & E
	MFCCs Delta		26	E
	MFCCs Delta2		26	E

The pseudo-code for extracting compact and expanded feature lists from audio files are listed in Algorithms 2 and 3, respectively.

Algorithm 2: Pseudo-code for extracting Compact feature list

```

1 Input: Input audio file ( $F$ )
2 Output: Compact features list ( $Feat_{Comp}$ )
3  $Feat_{Comp}, Features \leftarrow Emptylist$ 
4  $rate \leftarrow 16kHz$ 
5  $F \leftarrow resample(F, rate)$  // Resample audio to Target sample rate
6  $Mel \leftarrow melspectrogram(F)$ 
7  $Mel_{db} \leftarrow power\_to\_db(Mel)$ 
8  $MFCC \leftarrow mfcc(F, 13)$  // 13 features of MFCC
9  $Chroma \leftarrow chroma(F)$  // 12 classes of Chroma
10  $Features \leftarrow Mel_{db} + MFCC + Chroma$ 
11 foreach  $f$  in  $Features$  do
12    $f_{mean} \leftarrow mean(f)$ 
13    $Feat_{Comp} \leftarrow Feat_{Comp} + f_{mean}$ 
14    $f_{std} \leftarrow std(f)$ 
15    $Feat_{Comp} \leftarrow Feat_{Comp} + f_{std}$ 
16 Return  $Feat_{Comp}$ 

```

Algorithm 3: Pseudo-code for extracting Expanded feature list

```

1 Input: Input audio file ( $F$ )
2 Output: Expanded features list ( $Feat_{Exp}$ )
3  $Feat_{Exp}, Features \leftarrow Emptylist$ 
4  $rate \leftarrow 16kHz$ 
5  $F \leftarrow resample(F, rate)$  // Resample audio to Target sample rate
6  $ZCR \leftarrow zero\_crossing\_rate(F)$ 
7  $RMS \leftarrow root\_mean\_square(F)$ 
8  $S_{centroid} \leftarrow spectral\_centroid(F)$ 
9  $S_{bandwidth} \leftarrow spectral\_bandwidth(F)$ 
10  $S_{rolloff} \leftarrow spectral\_rolloff(F)$ 
11  $S_{contrast} \leftarrow spectral\_contrast(F)$  // 7 values of contrast
12  $Chroma \leftarrow chroma(F)$  // 12 classes of Chroma
13  $MFCC \leftarrow mfcc(F, 13)$  // 13 features of MFCC
14  $MFCC_{delta} \leftarrow mfcc\_delta(MFCC)$ 
15  $MFCC_{delta\_delta} \leftarrow mfcc\_delta(MFCC_{delta})$ 
16  $Features \leftarrow ZCR + RMS + S_{centroid} + S_{bandwidth} + S_{rolloff} + S_{contrast} + Chroma + MFCC +$   

 $MFCC_{delta} + MFCC_{delta\_delta}$ 
17 foreach  $f$  in  $Features$  do
18    $f_{mean} \leftarrow mean(f)$ 
19    $Feat_{Exp} \leftarrow Feat_{Exp} + f_{mean}$ 
20    $f_{std} \leftarrow std(f)$ 
21    $Feat_{Exp} \leftarrow Feat_{Exp} + f_{std}$ 
22 Return  $Feat_{Exp}$ 

```

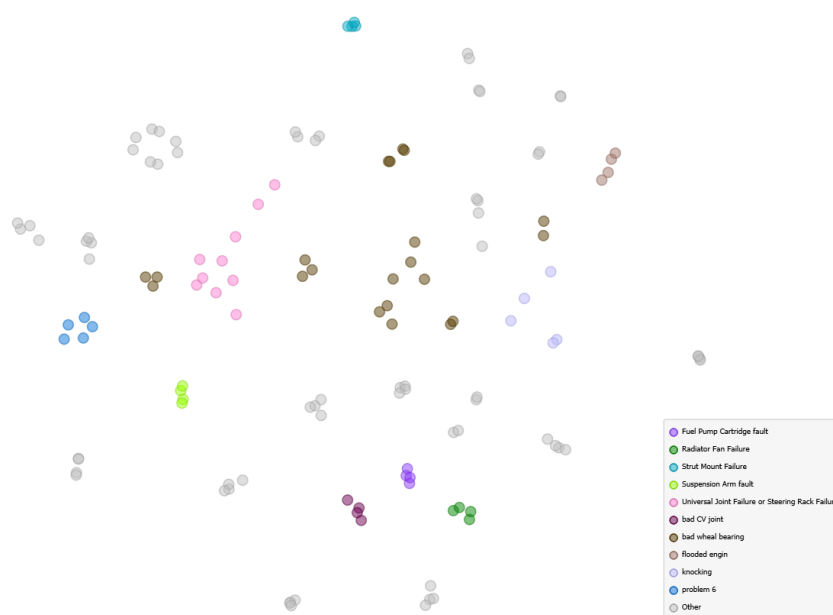


Figure 5. Two-dimensional data projection using t-SNE for compacted features extracted from DB1.

4.2. Classification

In this proposed system, the input audio is classified using ML techniques. The two versions of feature lists are used to test eleven different models on the three datasets created.

Neural Network (NN): A computational model consisting of interconnected neurons [52]. It is used for both regression and classification tasks. A neural network can be formulated by:

$$Y = f(WX + b)$$

where f is an activation function, W are weights, x is input, and b is bias.

Naive Bayes (NB): A probabilistic classifier based on Bayes' theorem, assuming independence among predictors [53]. The NB equation is given by:

$$P(C|X) = \frac{P(X|C)P(C)}{P(X)}$$

where C is the class and X is the feature vector.

Logistic Regression (LR): A statistical method for predicting binary classes [54]. The outcome is modeled using a logistic function, which outputs probabilities. Logistic Regression is formulated as follows:

$$P(Y = 1|X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n)}}$$

Stochastic Gradient Descent (SGD): An iterative method for optimizing an objective function, commonly used in training ML models, particularly neural networks [55].

$$\theta = \theta - \eta \nabla J(\theta)$$

where η is the learning rate and $\nabla J(\theta)$ is the gradient of the loss function.

k-Nearest Neighbors (kNN): A non-parametric method used for classification and regression by finding the k nearest data points in the feature space [56].

$$\hat{y} = \text{mode}(y_i) \text{ for } i \in k \text{ nearest neighbors}$$

Decision Tree (DT): A flowchart-like structure where each internal node represents a feature test, each branch represents an outcome, and each leaf node represents a class label [57].

$$\text{Class} = \text{leaf node based on features}$$

Random Forest (RF): An ensemble learning method that constructs multiple decision trees during training and outputs the mode of their predictions for classification tasks [58].

$$H(x) = \text{mode}(h_1(x), h_2(x), \dots, h_T(x))$$

Support Vector Machine (SVM): A supervised learning model that finds the optimal hyperplane that best separates different classes in the feature space [59].

$$f(x) = \text{sign}(W \cdot X + b)$$

CN2 Rule Induction: An algorithm for inducing classification rules from examples [60]. It generates rules based on the attributes of the training data.

$$\text{Class} = \text{if } (A_1 \wedge A_2 \wedge \dots \wedge A_n) \text{ then } C$$

where A_i are conditions based on attributes and C is the class label.

Adaptive Boosting (AdaBoost): An ensemble method that combines multiple weak classifiers to create a strong classifier by focusing on errors made by previous classifiers [61].

$$H(x) = \sum_{t=1}^T \alpha_t h_t(x)$$

where α_t is the weight of the classifier h_t .

Gradient Boosting (GB): An ensemble technique that builds models sequentially, with each new model correcting errors made by the previous ones [62].

$$F(x) = F_{m-1}(x) + \nu h_m(x)$$

where ν is the learning rate and h_m is the new model.

4.3. Feature Ranking

Feature ranking is considered a very crucial step in machine learning and data analysis and can select relevant features that contribute to the predictive models [63]. Four important feature ranking methods have been applied to estimate the importance of the different features used in previous classification experiments, including information gain, analysis of variance, ReliefF, and fast correlation-based filters. **Information Gain (IG)** measures the reduction in entropy or uncertainty after splitting a dataset based on a feature [64]. IG calculates the difference between the entropy of the target variable and the conditional entropy given the feature. Features with higher IG values are more informative.

$$IG = H(Y) - H(Y|X)$$

where $H(Y)$ is the entropy of the target variable, and $H(Y|X)$ is the conditional entropy.

Analysis of Variance (ANOVA) measures the ratio of between-class variance to within-class variance for a feature [65]. Features with higher ANOVA values are more discriminative.

$$ANOVA = \frac{Variance_{between-classes}}{Variance_{within-class}}$$

ReliefF is an extension of the Relief algorithm, estimating feature relevance by measuring the difference between the feature's values for nearest neighbours from different classes [66].

$$W(F) = W(F) + \frac{1}{k} \sum_{j=1}^k (\Delta(F, H_j) - \Delta(F, M_j))$$

where $W(F)$ is the current weight of feature F , k is the number of nearest neighbors considered, H_j are the nearest neighbors from the same class (hits), and M_j are the nearest neighbors from different classes (misses).

Fast Correlation-Based Filter (FCBF) evaluates feature relevance using correlation and redundancy [67]. It selects features with a high correlation to the target variable and low redundancy.

$$FCBF = Correlation(X, Y) - Redundancy(X)$$

4.4. Bayesian-Optimized Weighted Soft Voting with Feature Selection (BOWSVFS)

Ensemble learning is currently one of the most powerful methods in machine learning, combining many models to produce predicted performance better to that of standalone models [68]. Optimal combination in model weight determination remains an optimization challenge. To address this issue, the proposed approach employs Bayesian Optimization combined with Weighted Soft Voting.

In this model, WSV is the central process, with a few classifiers voting on a final prediction and assigning weights to each. Soft voting, unlike hard voting, does not involve direct class prediction; instead, it uses cross-class probability distributions [69]. Each prediction is weighted based on the perceived importance of each classifier in the entire ensemble, and the weighted probabilities are combined to obtain the final prediction.

The weights assigned to each classifier play a crucial role in the ensemble's performance. Traditional approaches often use equal weights or weights determined through grid search. However, these methods can be computationally expensive and may not find the optimal weight configuration, especially when dealing with multiple classifiers and features. Algorithm 4 depicts the procedures for calculating WSV. Figure 6 shows the steps of Bayesian-Optimized Weighted Soft Voting procedure.

Algorithm 4: Pseudo-code for calculating WSV

```

1 Input: Dataset ( $D$ ), weights ( $W$ ), No. of features ( $F$ )
2 Output: Mean Accuracy ( $ACC_{mean}$ )
3  $FOLDS_{num} \leftarrow 10$ 
4  $SEED_{rand} \leftarrow 42$ 
5  $X, l \leftarrow SeparateFeaturesAndLabels(D)$ 
6  $encoder \leftarrow LabelEncoder()$  // Label Encoding
7  $L_{encoded} \leftarrow encoder.Fit\_Transform(l)$ 
   // Feature Ranking
8  $selector \leftarrow SelectKBest(rankfunc, F)$ 
9  $F_{selected} \leftarrow selector.Fit\_Transform(X, L_{encoded})$ 
   // Feature Scaling
10  $scaler \leftarrow StandardScaler()$   $F_{scaled} \leftarrow scaler.Fit\_Transform(F_{selected})$ 
   // Initialize Classifiers
11  $classifiers \leftarrow [LR(SEED_{rand}), MLP(SEED_{rand}), AdaBoost(SEED_{rand})]$ 
   // Initialize Cross-Validation
12  $cv\_splits \leftarrow CreateKFold(FOLDS_{num}, shuffle, SEED_{rand})$ 
13  $ACC_{scores} \leftarrow []$ 
   // Perform Cross-Validation
14 foreach  $i_{train}, i_{test}$  in  $cv\_splits$  do
15    $X_{train} \leftarrow F_{scaled}[i_{train}]$ 
16    $l_{train} \leftarrow L_{encoded}[i_{train}]$ 
17    $X_{test} \leftarrow F_{scaled}[i_{test}]$ 
18    $l_{test} \leftarrow L_{encoded}[i_{test}]$ 
19    $Pred_{probability} \leftarrow []$ 
   // Train each classifier and get predictions
20   foreach  $classifier$  in  $classifiers$  do
21      $classifier.Fit(X_{train}, l_{train})$ 
22      $probs \leftarrow classifier.PredictProbabilities(X_{test})$ 
23      $Pred_{probability}.append(probs)$ 
   // Initialize weighted sum matrix with zeros
24    $Sum_{weighted} \leftarrow ZeroMatrix(LENGTH(Pred_{probability}[0]), CLASSES)$ 
   // Combine predictions with weights
25   for  $i \leftarrow 0$  to  $length(W)$  do
26      $Sum_{weighted} \leftarrow Sum_{weighted} + W[i] \times Pred_{probability}[i]$ 
27    $Prob \leftarrow Softmax(Sum_{weighted})$  // Apply softmax activation
28    $class_{predicted} \leftarrow ArgMax(Prob)$  // Get class with highest probability
29    $ACC \leftarrow CalculateAccuracy(l_{test}, class_{predicted})$  // Calculate accuracy
30    $ACC_{scores}.append(ACC)$ 
31  $ACC_{mean} \leftarrow mean(ACC_{scores})$  // Calculate mean accuracy
32 Return  $ACC_{mean}$ 

```

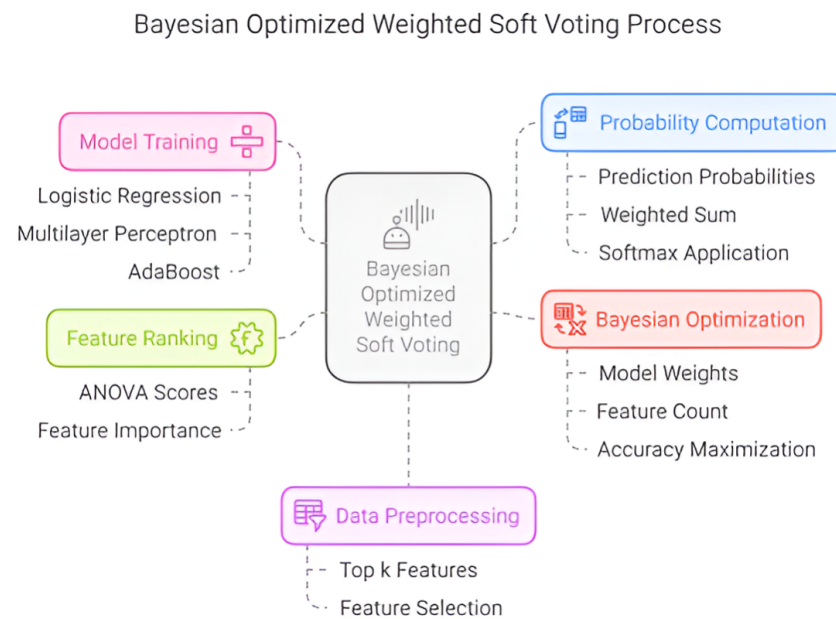


Figure 6. Steps for Bayesian-Optimized Weighted Soft Voting.

Bayesian optimization provides a more systematic and efficient technique to determining optimal weights [70]. Bayesian optimization uses a probabilistic model to predict the link between hyperparameters, weight, and feature counts, as well as model performance. This model is typically a Gaussian Process variation. This strategy is particularly useful since it swiftly explores the hyperparameter space by creating a proxy model of the objective function. It strikes a compromise between exploring unknown regions and exploiting known favorable locations, requiring fewer iterations than grid or random search algorithms. The steps for implementing the proposed Bayesian Optimization for Weighted Soft Voting are given in Algorithm 5, and these steps are:

1. **Feature Ranking Using ANOVA:** Rank features based on ANOVA (analysis of variance) scores.
2. **Data Preprocessing:** Select the top- k features based on ANOVA ranking using the following equation.

$$F_k = \text{SelectTopKFeatures}(F, \text{ANOVA}, k)$$

3. **Training Three Models:** Train Logistic Regression (LR), Multilayer Perceptron (MLP), and AdaBoost using the selected features with 10-fold cross-validation to ensure robustness using the following equation.

$$p^{(j)}(x) = f_j(x|F_k), \quad j \in \{LR, MLP, AB\}$$

4. **Computing Weighted Probabilities:**
 - Calculate the prediction probabilities for each model.
 - Compute the weighted sum of these probabilities using the following equation.

$$P_{sum}(c|x) = \sum_{j \in \{LR, MLP, AB\}} w_j p^{(j)}(x)_c$$

- Apply softmax to the final weighted sum of probabilities using the following equation.

$$P(c|x) = \frac{\exp(P_{sum}(c|x))}{\sum_{c'} \exp(P_{sum}(c'|x))}$$

5. **Bayesian Optimization:** Optimize the model weights (w_1, w_2, w_3) and the feature count (k) using Bayesian optimization techniques to maximize accuracy using the following equation.

$$\{w_1^*, w_2^*, w_3^*, k^*\} = \arg \max_{w_1, w_2, w_3, k} \text{Accuracy}(y, \hat{y}(x))$$

6. **Final Predictions Using Optimized Parameters:** Use the optimized weights and features for the final soft voting decision using the following equation.

$$\hat{y}(x) = \arg \max_c P(c|x)$$

Algorithm 5: Pseudo-code for Bayesian Optimization Weighted Soft Voting

```

1 Input: Dataset ( $D$ )
2 Output: Optimal Weights ( $W_{opt}$ ), Optimal Features ( $F_{opt}$ ), Best Accuracy ( $ACC_{best}$ )
3  $PARAM\_BOUNDS \leftarrow [w_1, w_2, w_3 \in [0, 1], F \in [20, 30]]$  // Parameter space bounds
4  $MAX\_ITERATIONS \leftarrow 20$ 
5  $INITIAL\_POINTS \leftarrow 10$ 
6  $SEED_{rand} \leftarrow 42$ 
   // Initialize Bayesian Optimization
7  $optimizer \leftarrow \text{BayesianOptimizer}(\text{ObjectiveFunction}, PARAM\_BOUNDS, SEED_{rand})$ 
   // Perform optimization
8 for  $i \leftarrow 1$  to  $INITIAL\_POINTS$  do
9    $optimizer.SampleRandomPoint()$ 
10 for  $iteration \leftarrow 1$  to  $MAX\_ITERATIONS$  do
   // Sample next point based on acquisition function
11    $point_{Next} \leftarrow optimizer.SuggestNextPoint()$ 
   // Normalize weights
12    $W_{sum} \leftarrow point_{Next}.w_1 + point_{Next}.w_2 + point_{Next}.w_3$ 
13    $W_{norm} \leftarrow [point_{Next}.w_1 / W_{sum}, point_{Next}.w_2 / W_{sum}, point_{Next}.w_3 / W_{sum}]$ 
   // Perform weighted soft voting with the given parameters
14    $ACC \leftarrow \text{CalculateWSVA}(D, W_{norm}, point_{Next}.F)$ 
   // Update optimizer with results
15    $optimizer.UpdateModel(point_{Next}, ACC)$ 
   // Extract best parameters
16  $P_{best} \leftarrow optimizer.GetBestParameters()$ 
17  $ACC_{best} \leftarrow optimizer.GetBestScore()$ 
   // Normalize final weights
18  $W_{sum} \leftarrow P_{best}.w_1 + P_{best}.w_2 + P_{best}.w_3$ 
19  $W_{opt} \leftarrow [P_{best}.w_1 / W_{sum}, P_{best}.w_2 / W_{sum}, P_{best}.w_3 / W_{sum}]$ 
20  $F_{opt} \leftarrow P_{best}.F$ 
21 Return  $W_{opt}, F_{opt}, ACC_{best}$ 

```

Implementing the proposed approach in practice can be performed by detecting the onset of an alert or emergency sound through a preprocessing step such as a sound event or voice activity detection (VAD). By distinguishing between background noise and relevant sound events, these techniques can help the system identify when a sound starts, even in noisy environments. Additionally, microphones or sensors can be strategically placed in or around the vehicle to capture sound more accurately, such as in isolated engine compartments with noise-canceling technology to improve sound capture quality. Also, the system can be integrated with existing vehicle monitoring systems to automatically trigger sound detection when certain conditions are met, such as abnormal engine behavior,

sudden changes in vehicle speed, or other sensor data that might indicate a fault or emergency event.

5. Experiments and Discussion

In this study, we evaluated the performance of eleven distinct machine learning models on three datasets, utilizing two versions of feature lists: a compact version comprising 52 features and an expanded version consisting of 126 features. The models were assessed based on several performance metrics, including Area Under the Curve (AUC), Classification Accuracy (CA), F1-score (F1), Precision (Prec), Recall, Matthews Correlation Coefficient (MCC), Specificity (Spec), and Logarithmic Loss (LogLoss).

Data acquisition involved online tools for downloading YouTube videos, while segmenting and audio extraction utilized FFmpeg (v6.0) and Veed.io. Audio conversion was performed using Online Audio Converter. Audio processing, feature extraction, and analysis were conducted using Python (v3.12) (Jupyter Notebook and Spyder IDE) on a computer equipped with an Intel Core i7 processor and 16 GB RAM. Key libraries employed include Librosa (v0.10.1) and Pydub (v0.25.1). All processes were completed using standard software tools. To ensure transparency and reproducibility, all datasets and codes are publicly available in our GitHub (v3.15) repository, along with comprehensive documentation and 72 references for dataset collection, including publicly available sources and YouTube audio samples.

5.1. Performance Metrics

This study used several performance indicators to analyze the efficiency of the models under evaluation. One significant metric utilized is the Area Under the Curve (AUC), which measures a model's ability to distinguish between positive and negative classes by computing the Area under the Receiver Operating Characteristic (ROC) curve. The AUC is determined using the following formula:

$$\text{AUC} = \int_0^1 \text{ROC}(x) dx$$

where ROC is the true positive rate plotted against the false positive rate at various threshold settings.

Another important metric is **Classification Accuracy (CA)**, calculated as the ratio of correctly predicted instances to the total number of instances in a dataset. The formula runs as follows:

$$\text{CA} = \frac{TP + TN}{TP + TN + FP + FN}$$

where: TP = True Positives, TN = True Negatives, FP = False Positives, FN = False Negatives. **Precision** is the ratio of true positive predictions the model provides to total positive predictions, indicating how precise the model is when producing positive predictions. Precision can be calculated as:

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall, also known as sensitivity, is the ratio of true positive predictions compared to all actual positive instances. It is one of the key metrics for evaluating the performance of a predictive model by its ability to identify positive instances correctly. It is calculated using the following formula:

$$\text{Recall} = \frac{TP}{TP + FN}$$

Specificity, on the other hand, measures the proportion of true negative predictions among all actual negative instances and is calculated as:

$$\text{Specificity} = \frac{TN}{TN + FP}$$

The **F1-score** comprehensively evaluates a model's performance by calculating the harmonic mean of Precision and recall. It's calculated with the following formula:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

The Matthews Correlation Coefficient (**MCC**) is a well-balanced measure that considers all four categories of the confusion matrix, providing a more comprehensive metric for binary classification. The MCC is calculated as follows:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

Logarithmic Loss (**LogLoss**) is a metric used to evaluate the performance of a classification model. It measures the accuracy of the probabilities assigned to each class. The LogLoss is calculated as:

$$\text{LogLoss} = -\frac{1}{N} \sum_{i=1}^N (y_i \log(p_i) + (1 - y_i) \log(1 - p_i))$$

where N = total number of instances, y_i = actual label (0 or 1), p_i = predicted probability of the positive class.

5.2. Hyperparameters

Hyperparameter tuning is one of the essential steps in any machine learning classification process [71,72]. It basically involves the selection of the best hyperparameters that control the learning process of a model. These hyperparameters are fixed before training since they are not learned from the data during training. Well-set hyperparameters can significantly improve the accuracy of a model and its generalization on unseen data. They form the basis for finding a good trade-off between bias and variance. Appropriate hyperparameters can speed up the training process, which means faster model development and deployment. Efficient hyperparameter settings can optimize resource utilization, thus reducing training time and costs.

Table 5 lists the configurations used for each model. The following are some common hyperparameters and their effect:

- **Learning Rate:** This hyperparameter controls the step size during gradient descent when moving toward the minimum. A very high learning rate results in instability, while a very low one slows down training.
- **Number of Trees/Estimators:** The number of trees in ensembling techniques like Random Forest and Gradient Boosting. More trees provide higher accuracy, but training a model takes longer.
- **Tree Depth:** The hyperparameter for tree-based models defines each tree's maximum depth. Deep trees can easily capture complex patterns but tend to overfit much more.
- **Regularization:** Methods such as L1 and L2 regularization prevent overfitting by penalizing large weights. The strength of regularization is a hyperparameter that needs tuning.
- **Number of Hidden Layers and Neurons:** This governs the model's architecture in neural networks.

Cross-validation is one of the best methods for hyperparameter tuning, and it was employed in this study. It evaluates model performance using techniques such as k-fold cross-validation to obtain a more accurate assessment of its performance. The sampling type used was a 10-fold cross-validation.

Table 5. Hyperparameters for each model and their values.

Model	Hyperparameter	Value	Model	Hyperparameter	Value
SGD	Algorithm		CN2 Rule Inducer	Rule ordering	Ordered
	Classification loss function	Hinge		Covering algorithm	Exclusive
	Regression loss function	Squared Loss		Rule search	
	Regularization method	Ridge (L2)		Evaluation measure	Entropy
	Regularization strength (α)	0.00001		Beam width	5
	Learning parameters			Rule filtering	
	Learning rate	Constant		Min. rule coverage	1
	Initial learning rate	0.01		Max. rule length	5
Neural Network	No. of hidden neurons	100	SVM	Cost	1
	Activation	ReLU		Regression loss (ϵ)	0.1
	Solver	SGD		Kernel	RBF
	Regularization (α)	0.0001		Gamma (γ)	Auto
	No. of iterations	1000		Numerical tolerance	0.001
	Training	Replicable		Iteration limit	100
Decision Tree	Type	Induce binary tree	AdaBoost	Base estimator	Tree
	Min. instances in leaves	2		No. of estimators	50
	Split subsets	>5		Learning rate	1
	Max. tree depth	100		Classification algorithm	SAMME
	Stop when majority	95%		Regression loss function	Exponential
Random Forest	No. of trees	10	Logistic Regression	Regularization type	Lasso (L1)
	Split subsets	>5		Strength	1
kNN	No. of neighbors	5	Gradient Boosting	No. of trees	100
	Metric	Euclidean		Learning rate	0.1
	Weight	Uniform		Tree depth	3
Naive Bayes	Default Settings			Training instances	1

5.3. Car Faults DB1 Evaluation

Various ML models were tested on the car faults dataset (DB1), analyzing two versions of feature lists: compact and expanded. Tables 6 and 7 display the measured performance metrics in both cases.

Table 6. Performance metrics of DB1 with compacted features.

Model	Train	Test	AUC	CA	F1	Prec	Recall	MCC	Spec	LogLoss
LR	13.724	0.06	0.965	0.865	0.863	0.877	0.865	0.859	0.995	0.844
NN	5.466	0.129	0.975	0.842	0.833	0.854	0.842	0.834	0.989	0.852
SGD	0.387	0.118	0.89	0.789	0.783	0.792	0.789	0.779	0.988	7.271
AdaBoost	4.102	0.164	0.912	0.722	0.706	0.711	0.722	0.707	0.984	3.834
kNN	0.097	0.243	0.964	0.714	0.694	0.753	0.714	0.704	0.988	1.715
RF	0.255	0.06	0.946	0.684	0.671	0.707	0.684	0.668	0.983	2.047
SVM	0.468	0.136	0.92	0.639	0.6	0.663	0.639	0.628	0.952	1.909
NB	0.168	0.104	0.975	0.639	0.571	0.576	0.639	0.64	0.989	5.678
DT	0.637	0	0.834	0.624	0.618	0.673	0.624	0.606	0.98	10.742
GB	36.681	0.081	0.834	0.504	0.509	0.612	0.504	0.472	0.951	5.033
CN2	71.023	0.064	0.68	0.406	0.402	0.435	0.406	0.376	0.968	2.848

Table 7. Performance metrics of DB1 with expanded features.

Model	Train	Test	AUC	CA	F1	Prec	Recall	MCC	Spec	LogLoss
LR	131.735	0.151	0.926	0.692	0.671	0.669	0.692	0.676	0.98	1.837
SGD	0.6	0.299	0.812	0.647	0.638	0.653	0.647	0.626	0.974	12.205
NN	5.768	0.316	0.932	0.624	0.614	0.625	0.624	0.603	0.977	1.519
AdaBoost	13.698	0.259	0.874	0.624	0.611	0.655	0.624	0.602	0.971	4.449
DT	1.311	0	0.827	0.609	0.606	0.659	0.609	0.592	0.983	11.262
NB	0.413	0.157	0.965	0.594	0.52	0.48	0.594	0.594	0.987	10.384
RF	0.415	0.14	0.915	0.586	0.568	0.606	0.586	0.561	0.967	2.75
kNN	0.217	0.32	0.9	0.459	0.421	0.411	0.459	0.431	0.965	4.502
GB	73.66	0.17	0.823	0.436	0.428	0.477	0.436	0.398	0.945	5.396
CN2	160.273	0.124	0.683	0.406	0.414	0.471	0.406	0.372	0.963	2.856
SVM	0.834	0.306	0.811	0.323	0.224	0.214	0.323	0.311	0.883	2.534

It was shown in the results of testing DB1 by the compact version of the extracted feature list that the Logistic Regression has the highest classification accuracy with the lowest Log Loss value among all evaluated models. Logistic Regression had the best classification accuracy but the second lowest Log Loss value when testing DB1 with the expanded version of the list of extracted features.

5.4. Other Sounds DB2 Evaluation

ML models were tested on other sound datasets (DB2), analyzing two versions of feature lists: compact and expanded. Tables 8 and 9 display the measured performance metrics in both cases.

Table 8. Performance metrics of DB2 with compact features.

Model	Train	Test	AUC	CA	F1	Prec	Recall	MCC	Spec	LogLoss
NN	25.526	0.143	0.99	0.884	0.884	0.886	0.884	0.874	0.992	0.462
LR	132.968	0.071	0.979	0.847	0.844	0.845	0.847	0.834	0.989	0.71
AdaBoost	27.621	0.236	0.966	0.845	0.843	0.844	0.845	0.831	0.989	2.206
SVM	3.519	0.336	0.987	0.831	0.827	0.844	0.831	0.817	0.985	0.636
RF	0.659	0.079	0.975	0.831	0.828	0.836	0.831	0.817	0.987	1.371
SGD	0.745	0.169	0.894	0.807	0.798	0.803	0.807	0.791	0.986	6.667
kNN	0.146	0.356	0.966	0.806	0.802	0.811	0.806	0.79	0.989	2.09
GB	232.345	0.224	0.976	0.805	0.805	0.814	0.805	0.788	0.985	0.933
DT	2.452	0.002	0.876	0.742	0.742	0.747	0.742	0.72	0.983	7.533
CN2	666.881	0.102	0.887	0.682	0.68	0.686	0.682	0.654	0.977	1.818
NB	0.348	0.076	0.933	0.577	0.583	0.691	0.577	0.556	0.982	5.333

Based on the DB2 testing results of the compact version of the list of features extracted, the Neural Networks model has the highest accuracy classification with the lowest Log Loss value among all models evaluated. Using the expanded version of the list of extracted features, Neural Networks obtained the second-highest accuracy after AdaBoost.

Table 9. Performance metrics of DB2 with expanded features.

Model	Train	Test	AUC	CA	F1	Prec	Recall	MCC	Spec	LogLoss
AdaBoost	64.35	0.488	0.969	0.844	0.841	0.844	0.844	0.83	0.989	2.032
NN	32.243	0.435	0.983	0.84	0.839	0.842	0.84	0.826	0.989	0.7
GB	539.137	0.336	0.979	0.835	0.834	0.842	0.835	0.821	0.987	0.844
SGD	1.418	0.304	0.903	0.822	0.819	0.819	0.822	0.806	0.988	6.164
LR	191.823	0.198	0.98	0.815	0.811	0.817	0.815	0.799	0.987	0.707
RF	1.004	0.161	0.972	0.812	0.805	0.812	0.812	0.796	0.986	1.485
SVM	6.311	0.588	0.982	0.79	0.78	0.792	0.79	0.772	0.983	0.732
DT	5.391	0	0.871	0.726	0.726	0.732	0.726	0.702	0.982	7.822
kNN	0.287	0.446	0.941	0.696	0.681	0.68	0.696	0.67	0.979	3.288
CN2	1256.612	0.18	0.893	0.687	0.688	0.694	0.687	0.659	0.979	1.8
NB	0.62	0.186	0.95	0.637	0.646	0.729	0.637	0.618	0.985	6.159

5.5. DB3 Evaluation

ML models were tested on the merged dataset (DB3), analyzing two versions of feature lists: compact and expanded. Tables 10 and 11 display the measured performance metrics in both cases.

Table 10. Performance metrics of DB3 with compact features.

Model	Train	Test	AUC	CA	F1	Prec	Recall	MCC	Spec	LogLoss
NN	33.206	0.156	0.988	0.855	0.851	0.857	0.855	0.845	0.993	0.6
LR	498.042	0.132	0.977	0.81	0.805	0.81	0.81	0.797	0.99	0.879
AdaBoost	52.826	0.326	0.955	0.786	0.779	0.782	0.786	0.772	0.99	2.775
kNN	0.118	0.368	0.966	0.766	0.758	0.768	0.766	0.751	0.99	2.205
RF	0.93	0.081	0.962	0.761	0.749	0.758	0.761	0.745	0.986	2.198
SVM	5.95	0.513	0.985	0.759	0.734	0.732	0.759	0.744	0.984	1.006
GB	1989.676	0.219	0.94	0.737	0.735	0.754	0.737	0.72	0.984	2.304
SGD	0.944	0.129	0.858	0.735	0.72	0.724	0.735	0.717	0.988	9.169
DT	3.696	0	0.858	0.676	0.67	0.673	0.676	0.654	0.984	8.976
CN2	993.324	0.085	0.853	0.605	0.606	0.613	0.605	0.578	0.981	2.666
NB	0.255	0.057	0.937	0.206	0.195	0.359	0.206	0.207	0.996	14.929

Table 11. Performance metrics of DB3 with expanded features.

Model	Train	Test	AUC	CA	F1	Prec	Recall	MCC	Spec	LogLoss
AdaBoost	126.229	0.446	0.953	0.799	0.792	0.804	0.799	0.786	0.989	2.525
NN	38.35	0.334	0.981	0.78	0.773	0.784	0.78	0.765	0.988	0.96
RF	1.37	0.252	0.962	0.772	0.755	0.753	0.772	0.757	0.987	2.217
SGD	2.012	0.311	0.874	0.764	0.755	0.753	0.764	0.748	0.991	8.16
LR	401.936	0.167	0.972	0.741	0.727	0.73	0.741	0.723	0.986	1.024
GB	3372.378	0.305	0.929	0.717	0.717	0.741	0.717	0.699	0.984	2.674
SVM	10.61	0.806	0.976	0.703	0.67	0.67	0.703	0.684	0.979	1.196
DT	7.743	0.002	0.843	0.655	0.653	0.66	0.655	0.632	0.985	9.877
kNN	0.261	0.464	0.927	0.631	0.612	0.609	0.631	0.606	0.981	4.366
CN2	1853.444	0.183	0.864	0.621	0.619	0.628	0.621	0.596	0.983	2.614
NB	0.451	0.276	0.932	0.068	0.033	0.065	0.068	0.069	0.997	29.18

The DB3 test results for the compact form of the list of features extracted showed that the Neural Networks model presents the minimum Log Loss value and the maximum accuracy of classification compared with the other models. Using the expanded version of the list of extracted features, Neural Networks reached the second-highest accuracy after AdaBoost.

5.6. Feature Ranking

Feature ranking was performed using the compact feature list on the DB1 dataset. Table 12 shows the rankings of the 52 features of the compact list. The table's rankings demonstrate that the top features across approaches were MFCC features (mean_10, mean_3, mean_2, mean_4) and Mel Spectrogram features (mean). The dominance of MFCC methods is evident. MFCC mean features are statistically significant across all measures, consistently outperforming standard deviation features. Chromagram characteristics, notably standard deviations, have a lower overall relevance. However, a few exceptions, such as chromagram_mean_7, have moderate rankings.

Table 12. Feature ranking for compact feature list applied on DB1.

Features		IG	ANOVA	Relieff	FCBF	Features		IG	ANOVA	Relieff	FCBF
Type	Name					Type	Name				
mfcc	mean_3	1.5308	10.3679	0.1138	0.8767	chromagram	mean_2	1.0131	5.0941	0.0402	0.4475
mfcc	mean_2	1.4491	10.6247	0.1171	0.7928	chromagram	mean_4	1.0087	12.9994	0.0702	0.4447
mel_spectrogram	std	1.3937	1.1654	0.0103	0.0001	chromagram	mean_10	0.9987	6.0528	0.049	0.4384
mfcc	mean_4	1.3734	22.9451	0.107	0.7215	chromagram	std_10	0.9859	6.1512	0.047	0
mfcc	mean_1	1.3498	16.7839	0.1006	0.7004	chromagram	std_8	0.9857	7.7565	0.0578	0
mfcc	mean_6	1.3418	12.0147	0.0832	0.6934	chromagram	std_11	0.9716	5.3361	0.0417	0.4214
mfcc	mean_10	1.2915	25.8226	0.0737	0.6505	chromagram	std_6	0.9697	5.4978	0.0306	0.4203
mfcc	mean_7	1.2911	8.428	0.0588	0.6501	chromagram	std_4	0.9529	7.7805	0.0421	0.41
mfcc	mean_5	1.2857	17.4085	0.085	0.6457	chromagram	std_3	0.9522	4.0328	0.0216	0.4096
mfcc	mean_11	1.2359	17.177	0.0705	0.6056	mfcc	std_4	0.9462	5.2936	0.055	0.406
mel_spectrogram	mean	1.2194	17.6741	0.075	0.5926	chromagram	std_5	0.9289	4.2578	0.0342	0
mfcc	mean_8	1.2028	14.3887	0.0551	0.5799	chromagram	std_2	0.9077	7.5226	0.0397	0
mfcc	mean_0	1.2	2.7237	0.019	0.5778	mfcc	std_1	0.874	3.2042	0.0382	0.3637
chromagram	mean_7	1.1776	8.5138	0.0678	0.561	mfcc	std_3	0.8657	2.9155	0.0373	0.359
mfcc	mean_9	1.1709	11.2358	0.0494	0.556	chromagram	std_1	0.8654	6.8358	0.0418	0.3588
mfcc	mean_12	1.1682	18.0902	0.0914	0.554	mfcc	std_2	0.8516	2.9743	0.0403	0.3511
chromagram	mean_8	1.1526	10.7271	0.0623	0.5426	mfcc	std_7	0.8077	8.099	0.0446	0.3271
chromagram	mean_0	1.1297	9.6163	0.0597	0.5262	mfcc	std_9	0.7995	7.6018	0.0328	0.3227
chromagram	mean_5	1.1043	7.3469	0.0648	0.5083	chromagram	std_0	0.7954	6.62	0.0522	0.3205
chromagram	mean_6	1.0997	6.3285	0.0555	0.5051	mfcc	std_6	0.7791	4.7404	0.0413	0.3119
chromagram	mean_9	1.0812	7.6679	0.0542	0	mfcc	std_8	0.7719	5.5775	0.0459	0.3082
chromagram	mean_1	1.0713	8.7643	0.0605	0.4857	mfcc	std_5	0.736	4.8213	0.0417	0.2897
chromagram	mean_3	1.0665	5.8873	0.0425	0.4825	mfcc	std_12	0.669	2.8505	0.0464	0.2565
chromagram	std_7	1.0527	6.2475	0.0376	0.4733	mfcc	std_11	0.6665	1.725	0.0312	0.2553
chromagram	std_9	1.0188	8.5498	0.056	0.4512	mfcc	std_0	0.6607	0.4204	−0.0035	0.2525
chromagram	mean_11	1.0162	5.4859	0.061	0	mfcc	std_10	0.6427	2.6443	0.0556	0.244

5.7. Evaluation of BOWSV

To incorporate Bayesian Optimization and Weighted Soft Voting into the proposed model, the previously selected features are ranked using ANOVA F-scores to select the most relevant features. Standardization is performed to scale the variables to the same scale. Multiple classifiers with diverse bases are included. Optimization starts with defining the bounds for classifier weights and feature counts. Then, cross-validation is applied to the results to obtain robust performance estimates, and acquisition function guides the search for optimal parameters. The optimization objective function evaluates the ensemble's performance using cross-validation to ensure that the estimates of the generalization performance are reliable. It converges to the best answer by iteratively proposing different weight combinations and assessing how well they work. Table 13 shows the metrics of the three datasets, DB1, DB2, and DB3, after applying ensemble optimization. For each iteration, the weights (w1, w2, w3), the number of features used, and the achieved accuracy are depicted.

Table 13. Metrics of datasets after ensemble optimization.

iter	DB1 Dataset					DB2 Dataset					DB3 Dataset				
	W1	W2	W3	Features	Accuracy	W1	W2	W3	Features	Accuracy	W1	W2	W3	Features	Accuracy
1	0.95	0.73	0.60	24	0.8879	0.95	0.73	0.60	23.75	0.8788	0.95	0.73	0.60	27.49	0.8557
2	0.16	0.06	0.87	22	0.8429	0.16	0.06	0.87	21.56	0.8332	0.16	0.06	0.87	23.12	0.8058
3	0.71	0.02	0.97	26	0.8797	0.71	0.02	0.97	26.01	0.7934	0.71	0.02	0.97	32.02	0.7861
4	0.21	0.18	0.18	28	0.8797	0.21	0.18	0.18	28.32	0.8739	0.21	0.18	0.18	36.65	0.8668
5	0.52	0.43	0.29	23	0.8951	0.52	0.43	0.29	23.04	0.8778	0.52	0.43	0.29	26.08	0.8488
6	0.14	0.29	0.37	26	0.8802	0.14	0.29	0.37	26.12	0.8797	0.14	0.29	0.37	32.24	0.8685
7	0.79	0.20	0.51	25	0.8648	0.79	0.20	0.51	24.56	0.839	0.79	0.20	0.51	29.12	0.817
8	0.05	0.61	0.17	26	0.8797	0.05	0.61	0.17	25.92	0.8885	0.05	0.61	0.17	31.85	0.8556
9	0.95	0.97	0.81	21	0.8813	0.95	0.97	0.81	20.65	0.87	0.95	0.97	0.81	21.3	0.8419
10	0.10	0.68	0.44	23	0.9104	0.10	0.68	0.44	23.05	0.8807	0.10	0.68	0.44	26.09	0.8556
11	0.01	0.96	0.14	23	0.9033	0.93	0.68	0.62	23.81	0.8798	0.17	0.05	0.19	36.68	0.8341
12	0.05	0.85	0.89	23	0.9033	0.01	0.91	0.18	26.57	0.8807	0.02	0.60	0.72	20.98	0.8453
13	0.02	0.99	0.72	23	0.8731	0.02	0.97	0.97	23.87	0.8768	0.04	0.47	0.35	20.09	0.8496
14	0.13	0.69	0.48	23	0.9104	0.15	0.04	0.02	26.95	0.8312	0.71	0.99	0.37	39.42	0.8668
15	0.02	0.29	0.33	23	0.9033	0.28	1.00	0.02	23.64	0.8817	0.16	0.80	0.10	29.43	0.8591
16	0.15	0.07	0.98	23	0.8275	0.01	0.01	0.13	23.54	0.8254	0.71	0.45	1.00	38.77	0.8496
17	0.25	0.88	0.41	24	0.9027	0.59	0.86	0.99	23.08	0.8846	0.88	0.23	0.37	35.5	0.8281
18	0.22	0.50	0.05	23	0.9027	0.87	0.99	0.27	22.6	0.8807	0.62	0.25	0.77	23.87	0.8109
19	0.14	0.99	1.00	24	0.8577	0.26	0.33	0.38	29.4	0.8778	0.46	0.98	0.53	20.13	0.8487
20	0.62	0.99	0.15	23	0.9027	0.21	0.99	0.89	28.78	0.8768	0.53	0.30	0.02	34.85	0.853
21	0.99	0.99	0.11	23	0.8648	0.98	0.93	0.15	28.95	0.8729	0.53	0.26	0.53	37.74	0.8513
22	0.97	0.01	0.07	22	0.85	0.97	0.01	0.07	22.06	0.7847	0.97	0.01	0.07	24.12	0.7294
23	0.09	0.05	0.84	24	0.8126	0.84	0.05	0.98	28.97	0.8138	0.09	0.05	0.84	28.97	0.8445
24	0.96	0.97	0.30	27	0.8731	0.02	0.89	0.01	29.14	0.8768	0.80	0.92	0.97	21.66	0.847
25	0.85	0.41	0.89	30	0.872	0.02	0.99	0.63	29.95	0.8768	0.29	0.69	0.67	27.29	0.8582
26	0.02	0.86	0.08	20	0.8885	0.70	0.76	0.01	29.94	0.8788	0.61	0.90	0.76	39.74	0.8685
27	0.85	0.03	0.24	20	0.8505	0.01	0.98	0.29	28.04	0.8768	0.75	0.04	0.40	34.28	0.8032
28	0.10	0.04	0.94	27	0.8044	0.23	0.95	0.26	20	0.8768	0.72	0.30	0.54	36.91	0.8419
29	0.12	0.96	0.08	29	0.8725	0.19	0.13	0.00	29.99	0.873	0.81	0.69	0.61	21.69	0.8402
30	0.02	0.94	0.94	20	0.8885	0.29	0.02	1.00	20.01	0.7682	0.97	0.73	0.04	35.85	0.859

Due to the large number of classes (27 for car faults, 22 for environmental sounds, and 49 for the merged dataset), a full confusion matrix would be impractical. Instead, key examples of classification performance have been summarized. Several classes—including Engine Misfire, Fuel Pump Cartridge Fault, Radiator Fan Failure, Strut Mount Failure, Suspension Arm Fault, and others—are classified with 100% accuracy, and the Bad CV Joint class achieves around 75% accuracy. Furthermore, the Bayesian-Optimized Weighted Soft Voting with Feature Selection (BOWSVFS) approach demonstrates the robustness of the model by achieving an overall accuracy of 91.04% on the car fault dataset (DB1).

However, some classes present challenges. For instance, the Universal Joint Failure or Steering class has an 80% correct classification rate, with misclassifications primarily as engine rattling noise. The Knocking class, in particular, exhibits significant difficulty, with only 40% of instances correctly classified and misclassifications distributed across categories such as bad wheel bearing, squeaky, and squeaky brake (or grinding brake). These examples highlight the strengths and areas for improvement within the proposed framework.

5.8. Outlook and Future Perspectives

The practical implications of this research are far-reaching. The framework enhances overall transportation safety through timely interventions and improved emergency response by enabling early fault detection and real-time classification of vehicle and environmental sounds. Furthermore, the ability to accurately interpret auditory cues supports the development of more accessible and inclusive ITSs. For instance, auditory alerts can be transformed into visual or haptic signals, thereby assisting individuals with disabilities and ensuring that critical safety information is disseminated effectively. These advances pave the way for smarter, more responsive urban transportation systems that improve efficiency and significantly elevate the safety and quality of life in smart cities.

Although the proposed classification methods can achieve a high degree of accuracy in sound-based early fault detection for vehicles in ITSs, there remains potential for further enhancement through the incorporation of explicit user feedback, such as ratings of the classification results. The efficacy of machine learning systems can be notably augmented by fostering a collaborative relationship with users, improving the system's accuracy and enhancing user understanding and trust in the system [73–75]. Users can contribute to the classification model by providing explicit collective feedback regarding its classification accuracy and the early detection of faults for vehicles in ITSs. This feedback can subsequently be utilized to refine the overall accuracy of the classification model. For instance, users might assign scores or ratings to the accuracy of detected faults. Nonetheless, sustaining user motivation for continuous feedback poses a challenge, as many users exhibit limited interest in participating in such evaluations [76].

The gamification concept is employed as a behavioral change strategy to enhance user motivation toward engaging in desired behaviours, such as providing feedback on the classification accuracy of detected faults for vehicles in ITS [77,78]. A prevalent application of gamification involves incorporating elements of video games, such as points and levels, into non-gaming contexts, such as educational settings [79]. Gamification has demonstrated successful implementation across various domains, including the promotion of healthy lifestyle choices [80], the enhancement of student engagement in academic courses [81], and the improvement of quality and productivity within business environments [82]. There are four primary elements of gamification commonly utilized in non-gaming contexts [83]:

1. **Points:** Many gamification strategies rely on point systems, which may include features such as levels and leaderboards. The classification accuracy of detected faults can be quantified through user ratings regarding the quality of fault detection for vehicles in ITS. Points accumulated or lost will subsequently inform the classification model's training to enhance its ability to detect sound-based faults for vehicles in ITSs early. Nevertheless, points should be integrated with other gamification elements to effectively motivate users [83].
2. **Digital Badges:** Users may receive digital badges as recognition for acquiring specific skills, knowledge, or achievements, thereby showcasing their accomplishments [84]. These badges are typically awarded based on predefined criteria [85–87]. For example, users might earn digital badges by reaching a specified number of points corresponding to their ratings on the classification accuracy of early detected sound-based faults for vehicles in ITSs.
3. **Levels:** Users must accumulate points to advance to higher levels. Upon reaching a predetermined point threshold, they can level up, thereby unlocking additional features within the system [88].
4. **Leaderboards:** Users can establish leaderboards to reflect their achievements or points earned or to track progress toward specific goals [86].

A recent study [89] identifies several factors that affect users' perceptions and responses to gamification elements utilized for feedback collection, revealing diverse preferences in this context. This underscores the necessity of systematically gathering users' explicit and collective feedback, which can be instrumental in optimizing our proposed classification model to align with user preferences. Neglecting this aspect could result in overlooking critical factors that enhance classification accuracy. To address this, one can utilize the application-independent conceptual framework proposed by [89], which can be adapted to gamify the feedback collection process regarding the accuracy of our sound-based early detection of faults for vehicles in the ITS system. This framework articulates the variations in user perceptions and needs concerning gamification elements, aiming to motivate users to provide high-quality feedback on the classification accuracy of our proposed system. It serves as a guiding resource for software engineers in encouraging users to offer their explicit and collective feedback, thereby facilitating further training of our classification model and potentially improving its early fault detection accuracy for vehicles in the ITS. Additionally, a category representing normal operational conditions or safe car sounds can be included to better differentiate faults from irrelevant auditory data.

6. Conclusions and Future Work

This study highlights the critical significance of sound-based diagnostics in improving Intelligent Transportation Systems (ITSs) in smart cities. The potential for real-time vehicle fault identification and enhanced accessibility for people with disabilities was demonstrated by creating a new dataset of automotive malfunction sounds and combining audio processing techniques with ML. The high accuracy rates attained by various ML models demonstrate the efficacy of sound-based techniques as a complement to classic vision-based systems. Finally, this study leads to a more inclusive and responsive transportation infrastructure, which aligns with the overall goals of smart city development.

Future research includes (i) expanding the dataset to cover diverse vehicles, faults, and real-world scenarios through collaborations with the automotive and public transport sectors; (ii) integrating sound data with visual and environmental sensors, which can enhance system robustness; (iii) developing real-time sound-based detection systems for urban applications and exploring advanced machine learning methods, such as deep learning and transfer learning, which will improve accuracy; (iv) incorporating the “no sound”

class in implementing the proposed approach in practice. (v) investigating generative model-based data augmentation strategies to boost dataset diversity and model resilience; and (vi) exploring domain adaptation techniques, few-shot learning, or data augmentation strategies to enhance generalization across a wider range of vehicles.

Author Contributions: A.R.: Conceptualization, Methodology, Software, Writing—Original Draft. Y.A.: Data Curation, Methodology, Investigation, Writing—Original Draft. T.A.F.: Visualization, Software. M.A.: Conceptualization, Methodology, Writing—Original Draft. A.B.: Data Curation, Investigation, Writing—Original Draft. M.B.: Methodology, Writing—Review and Editing, Supervision. M.A.E.: Supervision, Methodology, Writing—Review and Editing, Project Administration. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by funds from the King Salman Centre for Disability Research (Group no.: KSRG-2024-240).

Data Availability Statement: In this study, data from various sources are collected to create a tailored dataset for car sound analysis: Reliable audio samples are built by downloading videos from YouTube related to car faults, animal sounds, car crashes, siren sounds, etc. Additional audio is supplemented from Kaggle datasets: FSC22 [38], Google AudioSet [39], Audio Classifier Dataset [40], Sound classification of animal voice [41], and Vehicle Sounds dataset [42]. The data collected by the authors are available at: <https://github.com/amrrashed/Sound-Based-Vehicle-Diagnostics-Emergency-Signal-Recognition/tree/main> (accessed on 31 January 2025). Code availability: The code used is available at: <https://github.com/amrrashed/Sound-Based-Vehicle-Diagnostics-Emergency-Signal-Recognition/tree/main/codes> (accessed on 31 January 2025).

Acknowledgments: The authors extend their appreciation to the King Salman Centre for Disability Research for funding this work through Research Group No. KSRG-2024-240.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Gong, T.; Zhu, L.; Yu, F.R.; Tang, T. Edge Intelligence in Intelligent Transportation Systems: A Survey. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 8919–8944. [\[CrossRef\]](#)
2. Khalil, R.A.; Safelnasr, Z.; Yemane, N.; Kedir, M.; Shafiqurrahman, A.; Saeed, N. Advanced Learning Technologies for Intelligent Transportation Systems: Prospects and Challenges. *IEEE Open J. Veh. Technol.* **2024**, *5*, 397–427. [\[CrossRef\]](#)
3. Sarwatt, D.S.; Lin, Y.; Ding, J.; Sun, Y.; Ning, H. Metaverse for Intelligent Transportation Systems (ITS): A Comprehensive Review of Technologies, Applications, Implications, Challenges and Future Directions. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 6290–6308. [\[CrossRef\]](#)
4. Wang, B.; Li, Q.; Mao, Q.; Wang, J.; Chen, C.L.P.; Shangguan, A.; Zhang, H. A Survey on Vision-Based Anti Unmanned Aerial Vehicles Methods. *Drones* **2024**, *8*, 518. [\[CrossRef\]](#)
5. Masal, K.M.; Bhatlawande, S.; Shingade, S.D. Development of a visual to audio and tactile substitution system for mobility and orientation of visually impaired people: A review. *Multimed. Tools Appl.* **2024**, *83*, 20387–20427. [\[CrossRef\]](#)
6. Liu, F.; Lu, Z.; Lin, X. Vision-based environmental perception for autonomous driving. *Proc. Inst. Mech. Eng. Part D J. Automob. Eng.* **2023**, *239*, 39–69. [\[CrossRef\]](#)
7. Kiranyaz, S.; Can Devecioglu, O.; Alhams, A.; Sassi, S.; Ince, T.; Avci, O.; Gabbouj, M. Exploring Sound Versus Vibration for Robust Fault Detection on Rotating Machinery. *IEEE Sens. J.* **2024**, *24*, 23255–23264. [\[CrossRef\]](#)
8. Alqudaihi, K.S.; Aslam, N.; Khan, I.U.; Almuhaideb, A.M.; Alsunaidi, S.J.; Ibrahim, N.M.A.R.; Alhaidari, F.A.; Shaikh, F.S.; Alsenbel, Y.M.; Alalharith, D.M.; et al. Cough Sound Detection and Diagnosis Using Artificial Intelligence Techniques: Challenges and Opportunities. *IEEE Access* **2021**, *9*, 102327–102344. [\[CrossRef\]](#)
9. Vranken, E.; Mounir, M.; Norton, T., Sound-Based Monitoring of Livestock. In *Encyclopedia of Smart Agriculture Technologies*; Springer International Publishing: Berlin/Heidelberg, Germany, 2023; pp. 1–12. [\[CrossRef\]](#)
10. Pervez, F.; Shoukat, M.; Suresh, V.; Farooq, M.U.B.; Sandhu, M.; Qayyum, A.; Usama, M.; Girardi, A.; Latif, S.; Qadir, J. Medicine's New Rhythm: Harnessing Acoustic Sensing via the Internet of Audio Things for Healthcare. *IEEE Open J. Comput. Soc.* **2024**, *5*, 491–510. [\[CrossRef\]](#)
11. Kim, J.; Kim, J.; Kim, H. A Study on Gear Defect Detection via Frequency Analysis Based on DNN. *Machines* **2022**, *10*, 659. [\[CrossRef\]](#)

12. Koh, P.; Kim, S. Designing a Augmented Reality Auditory Training Game for in-situ training and diagnostic tool for the hearing impaired. In Proceedings of the Audio Engineering Society Conference: AES 2024 International Audio for Games Conference, Tokyo, Japan, 27–29 April 2024.
13. Oladimeji, D.; Gupta, K.; Kose, N.A.; Gundogan, K.; Ge, L.; Liang, F. Smart Transportation: An Overview of Technologies and Applications. *Sensors* **2023**, *23*, 3880. [CrossRef] [PubMed]
14. Mohammed, H.B.M.; Cavus, N. Utilization of Detection of Non-Speech Sound for Sustainable Quality of Life for Deaf and Hearing-Impaired People: A Systematic Literature Review. *Sustainability* **2024**, *16*, 8976. [CrossRef]
15. Nasim, F.; Masood, S.; Jaffar, A.; Ahmad, U.; Rashid, M. Intelligent Sound-Based Early Fault Detection System for Vehicles. *Comput. Syst. Sci. Eng.* **2023**, *46*, 3175–3190. [CrossRef]
16. Hamad, A.A.; Nasim, M.F.; Jaffar, A.; Khalaf, O.I.; Ouahada, K.; Hamam, H.; Akram, S.; Siddique, A. Cognitive Inspired Sound-Based Automobile Problem Detection: A Step Toward Xai. Available online: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4814232 (accessed on 25 January 2025) [CrossRef]
17. Yildirim, S.; Bingol, M.S. Design of a Proposed Neural Network for Sound Quality Analysis of Different Types for Car Systems. *Int. J. Mechatron. Appl. Mech.* **2024**, *16*, 76–81.
18. Akbalık, F.; Yıldız, A.; Ertuğrul, Ö.F.; Zan, H. Engine Fault Detection by Sound Analysis and Machine Learning. *Appl. Sci.* **2024**, *14*, 6532. [CrossRef]
19. Murovec, J.; Prezelj, J.; Ćirić, D.; Milivojević, M. Zero Crossing Signature: A Time-Domain Method Applied to Diesel and Gasoline Vehicle Classification. *IEEE Sens. J.* **2024**, *25*, 3. [CrossRef]
20. Wang, S.; Xu, Q.; Zhu, S.; Wang, B. Making transformer hear better: Adaptive feature enhancement based multi-level supervised acoustic signal fault diagnosis. *Expert Syst. Appl.* **2025**, *264*, 125736. [CrossRef]
21. Boztas, G.; Tuncer, T.; Aydogmus, O.; Yildirim, M. A DCSLBP based intelligent machine malfunction detection model using sound signals for industrial automation systems. *Comput. Electr. Eng.* **2024**, *119*, 109541. [CrossRef]
22. Wang, Y.; Li, D.; Li, L.; Sun, R.; Wang, S. A novel deep learning framework for rolling bearing fault diagnosis enhancement using VAE-augmented CNN model. *Heliyon* **2024**, *10*, e35407. [CrossRef]
23. Guo, X. Fault Diagnosis of Rolling Bearings Based on Acoustics and Vibration Engineering. *IEEE Access* **2024**, *12*, 139632–139648. [CrossRef]
24. Li, Y.; Tao, X.; Sun, Y. A Fault Diagnosis Method for Turnout Switch Machines Based on Sound Signals. *Electronics* **2024**, *13*, 4839. [CrossRef]
25. Kreuzer, M.; Schmidt, D.; Wokusch, S.; Kellermann, W. Real-World Airborne Sound Analysis for Health Monitoring of Bearings in Railway Vehicles. Available online: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4923626 (accessed on 19 January 2025). [CrossRef]
26. Yun, E.; Jeong, M. Acoustic Feature Extraction and Classification Techniques for Anomaly Sound Detection in the Electronic Motor of Automotive EPS. *IEEE Access* **2024**, *12*, 149288–149307. [CrossRef]
27. Shajie, D.; Juliet, S.; Ezra, K.; Annie Flora, J.B. Diagnostic Sonance: Sound-Based Approach to Assess Engine Ball Bearing Health in Automobiles. *Prz. Elektrotechniczny* **2024**, *1*, 74–78. [CrossRef]
28. Khan, F.A.; Jamil, A.; Khan, S.A.; Hameed, A.A. Enhancing robotic manipulator fault detection with advanced machine learning techniques. *Eng. Res. Express* **2024**, *6*, 025204. [CrossRef]
29. Kim, S.M.; Soo Kim, Y. Enhancing Sound-Based Anomaly Detection Using Deep Denoising Autoencoder. *IEEE Access* **2024**, *12*, 84323–84332. [CrossRef]
30. Senanayaka, A.; Lee, P.; Lee, N.; Dickerson, C.; Netchaev, A.; Mun, S. Enhancing the Accuracy of Machinery Fault Diagnosis through Fault Source Isolation of Complex Mixture of Industrial Sound Signals. *Int. J. Adv. Manuf. Technol.* **2024**, *133*, 5627–5642. [CrossRef]
31. Hameed, U.; Masood, S.; Nasim, F.; Jaffar, A.; Ahmed, Z.; Khan, R.; Hussain, A.; Ali, S.; Mehmood, A.; Shah, R. Exploring the Accuracy of Machine Learning and Deep Learning in Engine Knock Detection. *Bull. Bus. Econ.* **2024**, *13*, 203–210. [CrossRef]
32. Naryanto, R.F.; Delimayanti, M.K.; Naryaningsih, A.; Adi, R.; Setiawan, B.A. Fault Detection in Diesel Engines using Artificial Neural Networks and Convolutional Neural Networks. *J. Theor. Appl. Inf. Technol.* **2024**, *102*, 683–690.
33. Yuan, G.; Yang, Y. Fault detection method of new energy vehicle engine based on wavelet transform and support vector machine. *Int. J. Knowl. Based Intell. Eng. Syst.* **2024**, *28*, 718–731. [CrossRef]
34. Chu, S.; Zhang, J.; Liu, F.; Kong, X.; Jiang, Z.; Mao, Z. Fault identification model of diesel engine based on mixed attention: Single-cylinder fault data driven whole-cylinder diagnosis. *Expert Syst. Appl.* **2024**, *255*, 124769. [CrossRef]
35. Lee, D.; Choo, H.; Jeong, J. GCN-Based LSTM Autoencoder with Self-Attention for Bearing Fault Diagnosis. *Sensors* **2024**, *24*, 4855. [CrossRef]
36. Spadini, T.; Nose-Filho, K.; Suyama, R. Intelligent Fault Diagnosis of Type and Severity in Low-Frequency, Low Bit-Depth Signals. *arXiv* **2024**, arXiv:2411.06299. [CrossRef]

37. Gantert, L.; Zeffiro, T.; Sammarco, M.; Campista, M.E.M. Multiclass classification of faulty industrial machinery using sound samples. *Eng. Appl. Artif. Intell.* **2024**, *136*, 108943. [CrossRef]
38. Bandara, M.; Jayasundara, R.; Ariyaratne, I.; Meedeniya, D.; Perera, C. FSC22 Dataset. 2022. Available online: <https://www.kaggle.com/datasets/irmiot22/fsc22-dataset> (accessed on 19 January 2025).
39. Gemmeke, J.F.; Ellis, D.P.W.; Freedman, D.; Jansen, A.; Lawrence, W.; Moore, R.C.; Plakal, M.; Ritter, M. Audio Set: An ontology and human-labeled dataset for audio events. In Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LO, USA, 5–9 March 2017. [CrossRef]
40. Jacob, I. Audio Classifier Dataset. 2025. Available online: <https://www.kaggle.com/datasets/aklimarimi/audio-classifier-dataset> (accessed on 19 January 2025)
41. Putthewad, R.B. Sound Classification of Animal Voice. 2025. Available online: <https://www.kaggle.com/datasets/rushibalajiputthewad/sound-classification-of-animal-voice> (accessed on 19 January 2025).
42. Abderrahim, J. Vehicle Sounds Dataset. 2025. Available online: <https://www.kaggle.com/datasets/janboubiabderrahim/vehicle-sounds-dataset> (accessed on 19 January 2025).
43. Community, D. DCASE 2024 Challenge. 2024. Available online: <https://dcase.community/challenge2024/index> (accessed on 19 January 2025).
44. Community, D. UrbanSound8K Dataset. DCASE 2024 Challenge. 2024. Available online: <https://www.kaggle.com/code/prabhavsingh/urbansound8k-classification> (accessed on 19 January 2025).
45. Research, G. AudioSet. Available online: <https://www.kaggle.com/datasets/akela91/google-audioset> (accessed on 19 January 2025).
46. Li, H.; Wang, Z. Anomaly identification of wind turbine blades based on Mel-Spectrogram Difference feature of aerodynamic noise. *Measurement* **2025**, *240*, 115428. [CrossRef]
47. Lakdari, M.W.; Ahmad, A.H.; Sethi, S.; Bohn, G.A.; Clink, D.J. Mel-frequency cepstral coefficients outperform embeddings from pre-trained convolutional neural networks under noisy conditions for discrimination tasks of individual gibbons. *Ecol. Inform.* **2024**, *80*, 102457. [CrossRef]
48. Pandeya, Y.R.; Lee, J. GlocalEemoNet: An optimized neural network for music emotion classification and segmentation using timbre and chroma features. *Multimed. Tools Appl.* **2024**, *83*, 74141–74158. [CrossRef]
49. Constantinescu, C.; Brad, R. An Overview on Sound Features in Time and Frequency Domain. *Int. J. Adv. Stat. IT&C Econ. Life Sci.* **2023**, *13*, 45–58. [CrossRef]
50. Balingbing, C.; Kirchner, S.; Siebald, H.; Van Hung, N.; Hensel, O. Determining the sound signatures of insect pests in stored rice grain using an inexpensive acoustic system. *Food Secur.* **2024**, *16*, 1529–1538. [CrossRef]
51. Sanchez-Morillo, D.; Sales-Lerida, D.; Priego-Torres, B.; León-Jiménez, A. Cough Detection Using Acceleration Signals and Deep Learning Techniques. *Electronics* **2024**, *13*, 2410. [CrossRef]
52. Rizvi, S.; Pettee, M.; Nachman, B. Learning likelihood ratios with neural network classifiers. *J. High Energy Phys.* **2024**, *2024*, 1–41. [CrossRef]
53. Peretz, O.; Koren, M.; Koren, O. Naive Bayes classifier—An ensemble procedure for recall and precision enrichment. *Eng. Appl. Artif. Intell.* **2024**, *136*, 108972. [CrossRef]
54. Khashei, M.; Etemadi, S.; Bakhtiarvand, N. A New Discrete Learning-Based Logistic Regression Classifier for Bankruptcy Prediction. *Wirel. Pers. Commun.* **2024**, *134*, 1075–1092. [CrossRef]
55. Azimjonov, J.; Kim, T. Stochastic gradient descent classifier-based lightweight intrusion detection systems using the efficient feature subsets of datasets. *Expert Syst. Appl.* **2024**, *237*, 121493. [CrossRef]
56. Sun, Y.; Liu, Q. Collaborative filtering recommendation based on K-nearest neighbor and non-negative matrix factorization algorithm. *J. Supercomput.* **2025**, *81*, 79. [CrossRef]
57. Larisa, L. Optimized Composition of Business Process Web Services via QoS-Based Categorization Using Decision Tree Classifier and Knowledge-Based Decision Support. *Am. J. Bus. Oper. Res.* **2025**, *12*, 1–14. [CrossRef]
58. Bouke, M.A.; Alramli, O.I.; Abdullah, A. XAIRF-WFP: A novel XAI-based random forest classifier for advanced email spam detection. *Int. J. Inf. Secur.* **2025**, *24*, 5. [CrossRef]
59. Li, Y.; Xie, X. Two novel deep multi-view support vector machines for multiclass classification. *Appl. Intell.* **2025**, *55*, 1–17. [CrossRef]
60. Maszczyk, C.; Sikora, M.; Wróbel, Ł. Classification, Regression, and Survival Rule Induction with Complex and M-of-N Elementary Conditions. *Mach. Learn. Knowl. Extr.* **2024**, *6*, 554–579. [CrossRef]
61. Kumpf, K.; Protic, M.; Jovanovic, L.; Cajic, M.; Zivkovic, M.; Bacanin, N. Insider Threat Detection Using Bidirectional Encoder Representations From Transformers and Optimized AdaBoost Classifier. In Proceedings of the 2024 International Conference on Circuit, Systems and Communication (ICCSC), Fez, Morocco, 28–29 June 2024; pp. 1–6. [CrossRef]
62. Theerthagiri, P. Liver disease classification using histogram-based gradient boosting classification tree with feature selection algorithm. *Biomed. Signal Process. Control* **2025**, *100*, 107102. [CrossRef]

63. Aljohani, M.; AbdulAzeem, Y.; Balaha, H.M.; Badawy, M.; Elhosseini, M.A. Advancing feature ranking with hybrid feature ranking weighted majority model: A weighted majority voting strategy enhanced by the Harris hawks optimizer. *J. Comput. Des. Eng.* **2024**, *11*, 308–325. [\[CrossRef\]](#)
64. Gao, J.; Wang, Z.; Jin, T.; Cheng, J.; Lei, Z.; Gao, S. Information gain ratio-based subfeature grouping empowers particle swarm optimization for feature selection. *Knowl. Based Syst.* **2024**, *286*, 111380. [\[CrossRef\]](#)
65. Jamil, M.A.; Khanam, S. Influence of One-Way ANOVA and Kruskal–Wallis Based Feature Ranking on the Performance of ML Classifiers for Bearing Fault Diagnosis. *J. Vib. Eng. Technol.* **2024**, *12*, 3101–3132. [\[CrossRef\]](#)
66. Yan, M.; Deng, J.; Zhang, S.; Chen, P. Feature Selection Method Based on Improved Differential Evolution and ReliefF. In Proceedings of the 2024 Guangdong-Hong Kong-Macao Greater Bay Area International Conference on Digital Economy and Artificial Intelligence, DEAI 2024, Dongguan, China, 19–21 January 2024; pp. 539–543. [\[CrossRef\]](#)
67. Zhang, S.; Wang, T.; Worden, K.; Sun, L.; Cross, E.J. Canonical-correlation-based fast feature selection for structural health monitoring. *Mech. Syst. Signal Process.* **2025**, *223*, 111895. [\[CrossRef\]](#)
68. Liu, Z. Ensemble Learning. In *Artificial Intelligence for Engineers*; Springer Nature: Cham, Switzerland, 2025; pp. 221–242. [\[CrossRef\]](#)
69. Chhillar, I.; Singh, A. An improved soft voting-based machine learning technique to detect breast cancer utilizing effective feature selection and SMOTE-ENN class balancing. *Discov. Artif. Intell.* **2025**, *5*, 4. [\[CrossRef\]](#)
70. Mahboubi, N.; Xie, J.; Huang, B. Point-by-point transfer learning for Bayesian optimization: An accelerated search strategy. *Comput. Chem. Eng.* **2025**, *194*, 108952. [\[CrossRef\]](#)
71. Iturbe-Araya, J.I.; Rifà-Pous, H. Enhancing unsupervised anomaly-based cyberattacks detection in smart homes through hyperparameter optimization. *Int. J. Inf. Secur.* **2025**, *24*, 45. [\[CrossRef\]](#)
72. Widardo, F.; Chowanda, A. Hyperparameter tuning for deep learning model used in multimodal emotion recognition data. *Bull. Electr. Eng. Inform.* **2025**, *14*, 261–267. [\[CrossRef\]](#)
73. Stumpf, S.; Rajaram, V.; Li, L.; Burnett, M.; Dietterich, T.; Sullivan, E.; Drummond, R.; Herlocker, J. Toward harnessing user feedback for machine learning. In Proceedings of the 12th International Conference on Intelligent User Interfaces, IUI07, Honolulu, HI, USA, 28–31 January 2007; pp. 82–91. [\[CrossRef\]](#)
74. Lee, T.Y.; Smith, A.; Seppi, K.; Elmqvist, N.; Boyd-Graber, J.; Findlater, L. The human touch: How non-expert users perceive, interpret, and fix topic models. *Int. J. Hum. Comput. Stud.* **2017**, *105*, 28–42. [\[CrossRef\]](#)
75. Liao, Q.V.; Gruen, D.; Miller, S. Questioning the AI: Informing Design Practices for Explainable AI User Experiences. In Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, CHI '20, Honolulu, HI, USA, 25–30 April 2020; pp. 1–15. [\[CrossRef\]](#)
76. Almaliki, M.; Ali, R. Persuasive and Culture-Aware Feedback Acquisition. In *Persuasive Technology*; Springer International Publishing: Berlin/Heidelberg, Germany, 2016; pp. 27–38. [\[CrossRef\]](#)
77. Deterding, S.; Dixon, D.; Khaled, R.; Nacke, L. From game design elements to gamefulness: Defining “gamification”. In Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments, MindTrek '11, Tampere, Finland, 28–30 September 2011; pp. 9–15. [\[CrossRef\]](#)
78. Herzig, P.; Ameling, M.; Schill, A. A Generic Platform for Enterprise Gamification. In Proceedings of the 2012 Joint Working IEEE/IFIP Conference on Software Architecture and European Conference on Software Architecture, Helsinki, Finland, 20–24 August 2012; pp. 219–223. [\[CrossRef\]](#)
79. Nicholson, S. A RECIPE for Meaningful Gamification. In *Gamification in Education and Business*; Reiners, T., Torsten and Wood, L.C., Eds.; Springer International Publishing: Cham, Switzerland, 2014; pp. 1–20. [\[CrossRef\]](#)
80. Johnson, D.; Deterding, S.; Kuhn, K.A.; Staneva, A.; Stoyanov, S.; Hides, L. Gamification for health and wellbeing: A systematic review of the literature. *Internet Interv.* **2016**, *6*, 89–106. [\[CrossRef\]](#)
81. Pløhn, T.; Aalberg, T. Using gamification to motivate smoking cessation. In Proceedings of the European Conference on Games Based Learning, Steinkjer, Norway, 8–9 October 2015; p. 431.
82. Simões, J.; Redondo, R.D.; Vilas, A.F. A social gamification framework for a K-6 learning platform. *Comput. Hum. Behav.* **2013**, *29*, 345–353. [\[CrossRef\]](#)
83. Lister, M. Gamification: The effect on student motivation and performance at the post-secondary level. *Issues Trends Educ. Technol.* **2015**, *3*, 2. [\[CrossRef\]](#)
84. Abramovich, S.; Schunn, C.; Higashi, R.M. Are badges useful in education?: It depends upon the type of badge and expertise of learner. *Educ. Technol. Res. Dev.* **2013**, *61*, 217–232. [\[CrossRef\]](#)
85. Ahn, J.; Pellicone, A.; Butler, B.S. Open badges for education: What are the implications at the intersection of open systems and badging? *Res. Learn. Technol.* **2014**, *22*, 563. [\[CrossRef\]](#)
86. Domínguez, A.; Saenz-de Navarrete, J.; De-Marcos, L.; Fernández-Sanz, L.; Pagés, C.; Martínez-Herráiz, J.J. Gamifying learning experiences: Practical implications and outcomes. *Comput. Educ.* **2013**, *63*, 380–392. [\[CrossRef\]](#)

87. Hanus, M.D.; Fox, J. Assessing the effects of gamification in the classroom: A longitudinal study on intrinsic motivation, social comparison, satisfaction, effort, and academic performance. *Comput. Educ.* **2015**, *80*, 152–161. [[CrossRef](#)]
88. Goehle, G. Gamification and Web-based Homework. *PRIMUS* **2013**, *23*, 234–246. [[CrossRef](#)]
89. Almaliki, M. Misinformation-Aware Social Media: A Software Engineering Perspective. *IEEE Access* **2019**, *7*, 182451–182458. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.