Urban Sound Classification: A Feature-Based Machine Learning Approach

Technical Report

Kostis Matzorakis
George Manthos
Spiros Batziopoulos

MSc in Artificial Intelligence
University of Piraeus & NCSR Demokritos

February 2026

Abstract

This report presents a comprehensive study on urban sound classification using feature-based machine learning approaches. We evaluate five classical machine learning algorithms (Logistic Regression, Support Vector Machines, Random Forest, XGBoost, and Multi-Layer Perceptron) on the UrbanSound8K dataset, which contains 8,732 labeled audio excerpts across 10 urban sound categories. Our methodology employs handcrafted acoustic features, including Mel-Frequency Cepstral Coefficients (MFCCs) and spectral features, rather than end-to-end deep learning approaches. A critical contribution of this work is the systematic comparison of two evaluation protocols: (1) the dataset-compliant 10-fold cross-validation using predefined folds, and (2) a conventional random train-test split. Our results reveal a striking 19 percentage point difference in accuracy between these protocols (70.40% vs. 89.64% for XGBoost), demonstrating the severe impact of data leakage when related audio excerpts from the same recording appear in both training and test sets. XGBoost with enhanced temporal features (including delta and delta-delta coefficients) achieves the best performance at 70.40% ± 4.94% accuracy under proper evaluation. We implement an efficient feature caching system using MD5 hashing that provides 10-100× speedup for iterative experiments. This work emphasizes that evaluation methodology is as critical as model architecture, and provides a reproducible baseline for feature-based audio classification research.

Keywords: Audio Classification, Machine Learning, Feature Engineering, UrbanSound8K, XGBoost, Evaluation Methodology, Data Leakage, MFCCs

1. Introduction

Urban sound classification is an important task in environmental audio analysis with applications spanning smart city infrastructure, public safety systems, and accessibility tools. The ability to automatically recognize and categorize sounds such as car horns, sirens, construction equipment, and ambient urban noise enables systems to respond intelligently to auditory events in real-time.

While recent advances in deep learning have achieved impressive results on audio classification tasks using end-to-end learning from raw waveforms or spectrograms, feature-based machine learning approaches remain valuable for several reasons: (1) interpretability – features map to known acoustic properties, (2) data efficiency – they work effectively with smaller datasets, (3) computational efficiency – no GPU required for training or inference, and (4) reproducibility – fixed pipelines with standard libraries enable consistent comparisons.

1.1 Project Objectives

This project aims to develop and evaluate a feature-based machine learning system for urban sound classification with the following specific objectives:

- Implement a robust audio preprocessing and feature extraction pipeline for the UrbanSound8K dataset
- Compare five classical machine learning algorithms across different feature configurations
- Systematically evaluate the impact of evaluation methodology on measured performance
- Demonstrate the critical importance of proper train-test splitting to prevent data leakage
- Develop an efficient feature caching system to accelerate iterative experimentation
- Establish a reproducible baseline for future research and comparison

1.2 Report Organization

The remainder of this report is organized as follows: Section 2 provides background on audio features and machine learning algorithms. Section 3 details our methodology including preprocessing, feature extraction, model selection, and evaluation protocols. Section 4 describes our implementation and system architecture. Section 5 presents experimental results. Section 6 discusses key findings and implications. Section 7 concludes with summary and future directions.

2. Background and Related Work

2.1 UrbanSound8K Dataset

The UrbanSound8K dataset is a widely-used benchmark for urban sound classification research. It contains 8,732 labeled sound excerpts (≤4 seconds) from field recordings, categorized into 10 classes: air conditioner, car horn, children playing, dog bark, drilling, engine idling, gun shot, jackhammer, siren, and street music. The dataset was created by Salamon et al. (2014) specifically to support research in urban sound classification.

A critical feature of UrbanSound8K is its predefined 10-fold structure, where each fold is ensured to contain excerpts from different original recordings. This design prevents data leakage during cross-validation, as multiple excerpts extracted from the same source recording share recording-specific characteristics (microphone properties, environmental acoustics, background noise patterns) that would artificially inflate performance if present in both training and test sets.

2.2 Audio Feature Extraction

2.2.1 Mel-Frequency Cepstral Coefficients (MFCCs)

MFCCs are the most widely-used features in audio and speech processing. They approximate the human auditory system's response to sound by applying a mel-scale filter bank (mimicking frequency perception) followed by a discrete cosine transform for decorrelation. The resulting coefficients compactly represent the spectral envelope, capturing timbre information essential for sound discrimination.

The MFCC extraction process involves: (1) framing the signal with overlapping windows, (2) computing the power spectrum via FFT, (3) applying triangular mel-scale filters, (4) taking logarithm to compress dynamic range, and (5) applying DCT to obtain cepstral coefficients. We extract 13 MFCCs per frame, then aggregate across time using mean and standard deviation, yielding 26 features per audio file.

2.2.2 Temporal Dynamics: Delta and Delta-Delta Coefficients

Static MFCCs capture spectral characteristics at each frame but lack temporal context. Delta coefficients (Δ) represent first-order derivatives approximating the rate of change of MFCCs over time. Delta-delta coefficients ($\Delta\Delta$) are second-order derivatives capturing acceleration. Including these temporal features significantly improves classification performance by encoding dynamic patterns in the audio signal.

2.2.3 Spectral Features

Beyond MFCCs, we extract complementary spectral features that capture different aspects of the audio signal:

• **Spectral Centroid:** The center of mass of the spectrum, indicating brightness (higher for cymbals, lower for bass)

• **Spectral Rolloff:** The frequency below which 85% of spectral energy lies, distinguishing tonal from noisy sounds

• **Zero-Crossing Rate:** How often the signal changes sign, correlating with noisiness and presence of high frequencies

• **Spectral Contrast:** Peak-to-valley ratio in sub-bands, capturing harmonic versus noisy structure

2.3 Machine Learning Algorithms

We evaluate five classical machine learning algorithms representing different learning paradigms:

Algorithm	Type	Key Characteristics
Logistic Regression	Linear	Fast, interpretable, probabilistic outputs
Support Vector Machine	Kernel-based	RBF kernel maps to high-dimensional space
Random Forest	Tree ensemble	Bagging, feature importance, robust to outliers
XGBoost	Gradient boosting	Sequential error correction, state-of-art on tabular data
Multi-Layer Perceptron	Neural network	Learns hierarchical representations

3. Methodology

3.1 Data Preprocessing Pipeline

All audio files undergo a standardized preprocessing pipeline to ensure features are comparable across recordings:

Resampling to 16 kHz: Converts all recordings to a uniform sample rate. 16 kHz captures frequencies up to 8 kHz (Nyquist limit), sufficient for environmental sounds while reducing computational cost.

Peak Normalization: Divides by the maximum absolute amplitude to normalize to [-1, 1] range. This removes recording volume differences, preventing the model from learning volume rather than sound content.

Length Normalization to 4 seconds: Zero-pads shorter clips or truncates longer ones to fixed duration. Required for consistent feature vector dimensionality across samples.

3.2 Feature Extraction Configurations

We evaluate two feature configurations to assess the impact of temporal dynamics:

Feature Set	Dimensionality	Components
Baseline	46	MFCCs (26) + Spectral Centroid (2) + Spectral Rolloff (2) + Zero-Crossing Rate (2) + Spectral Contrast (14)
XGB (Enhanced)	~100	Baseline (46) + Delta MFCCs (26) + Delta-Delta MFCCs (26) + Log-Mel Spectrogram (2)

Feature extraction parameters: n_mfcc=13, n_fft=2048, hop_length=512, n_mels=128. All features are aggregated across frames using mean and standard deviation to produce fixed-length vectors regardless of audio duration.

3.3 Feature Caching System

Feature extraction is computationally expensive, especially when repeated across multiple experiments with different models and hyperparameters. We implement an MD5-based caching system that stores extracted features on disk, indexed by a hash of the file path, feature set identifier, and extraction parameters. On subsequent runs, features are loaded from cache if available, providing 10-100× speedup. This optimization is essential for iterative experimentation and hyperparameter tuning.

3.4 Model Configurations

We configure each algorithm with parameters selected to balance performance and computational efficiency:

Model	Key Hyperparameters
Logistic Regression	`max_iter=1000, random_state=42`
SVM	`kernel=RBF, C=10, gamma=scale`
Random Forest	`n_estimators=200, max_depth=20, min_samples_split=5`
XGBoost	`n_estimators=400, learning_rate=0.05, max_depth=6, subsample=0.9`
MLP	`hidden_layers=(128, 64), activation=relu, max_iter=500`

3.5 Evaluation Protocols

A critical contribution of this work is the systematic comparison of two evaluation methodologies. This comparison reveals fundamental insights about data leakage and its impact on reported performance.

3.5.1 Protocol 1: Compliant Cross-Validation (Recommended)

This protocol follows the UrbanSound8K creators' recommended evaluation procedure. We perform 10-fold cross-validation using the dataset's predefined folds. In each iteration, one fold serves as the test set while the remaining nine folds form the training set. This process repeats 10 times, with each fold serving exactly once as the test set. Performance metrics are reported as mean ± standard deviation across all folds.

Critical advantage: The predefined folds ensure that excerpts from the same original recording never appear in both training and test sets simultaneously. This prevents the model from memorizing recording-specific artifacts (microphone characteristics, room acoustics, background noise patterns) rather than learning generalizable sound patterns. This protocol produces realistic performance estimates that generalize to truly unseen recordings.

3.5.2 Protocol 2: Random Train-Test Split (For Comparison)

This protocol represents a conventional machine learning approach found in many textbooks: randomly shuffle all samples and split into 80% training and 20% test sets (with stratification to maintain class balance). An additional 12.5% of training data is held out for validation.

Critical flaw: Random splitting ignores the fact that UrbanSound8K contains multiple excerpts extracted from individual source recordings. When related excerpts appear in both training and test sets, the model can exploit recording-specific patterns, leading to artificially inflated performance that does not reflect real-world generalization capability.

Why include this protocol? We deliberately evaluate this flawed methodology to demonstrate the severe impact of data leakage and emphasize that proper evaluation design is as critical as model architecture. The performance difference between protocols quantifies the cost of improper evaluation.

3.6 Performance Metrics

For both protocols, we compute standard classification metrics using weighted averaging to account for class imbalance: accuracy, precision, recall, and F1-score. For Protocol 1, we additionally report standard deviations across folds to quantify performance variance.

4. Implementation

4.1 System Architecture

The project is implemented in Python with a modular architecture promoting code reuse and maintainability. The system comprises distinct modules for each pipeline stage:

- **config/config.py:** Centralized configuration (paths, audio parameters, feature parameters, CV settings)
- **src/data_loader.py:** Metadata loading and fold-based data splitting
- **src/pre_processing.py:** Audio loading, resampling, normalization, and length fixing
- **src/feature_extraction.py:** MFCC and spectral feature computation
- **src/cache_manager.py:** MD5-based feature caching and directory management
- **src/models.py:** ModelFactory pattern for consistent model instantiation
- **src/train.py:** Training orchestration with optional cross-validation
- **src/evaluate.py:** Metrics computation, visualization, and reporting
- **notebooks/:** Jupyter notebooks for experiments and analysis

4.2 Key Libraries and Dependencies

The implementation leverages established scientific computing libraries: librosa (audio processing), scikit-learn (machine learning models and metrics), xgboost (gradient boosting), pandas (data manipulation), numpy (numerical computation), matplotlib/seaborn (visualization), and joblib (caching).

4.3 Reproducibility Measures

We implement several measures to ensure experimental reproducibility:

- Fixed random seed (RANDOM_STATE=42) throughout all experiments
- Centralized configuration file documenting all parameters
- Feature caching with content-based hashing ensures identical feature extraction
- Complete requirements.txt specifying exact library versions
- Modular code structure enabling independent verification of each stage
- Version control via GitHub repository

5. Experimental Results

5.1 Protocol 1: Compliant Cross-Validation Results

Table 1 presents results from the dataset-compliant 10-fold cross-validation evaluation. These represent realistic performance estimates on truly unseen recordings.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
XGBoost	**70.40 ± 4.94**	**70.64 ± 6.14**	**70.40 ± 4.94**	**69.54 ± 5.57**
MLP	68.83 ± 4.74	68.57 ± 5.59	68.83 ± 4.74	67.83 ± 5.18
SVM	68.40 ± 3.74	68.09 ± 4.51	68.40 ± 3.74	67.49 ± 4.18
Random Forest	68.15 ± 3.05	69.47 ± 3.57	68.15 ± 3.05	67.52 ± 3.21
Logistic Reg.	67.66 ± 4.18	67.98 ± 4.73	67.66 ± 4.18	67.33 ± 4.56

Table 1: Protocol 1 (Compliant 10-Fold CV) Results

Key observations: XGBoost achieves the highest accuracy at 70.40% with temporal features (delta and delta-delta MFCCs). All models cluster within a relatively narrow 3-percentage-point range (67.66% to 70.40%), suggesting the task difficulty under proper evaluation. Standard deviations of 3-6% indicate moderate variance across folds, reflecting the dataset's inherent complexity and the natural variation when testing on different subsets of recordings.

5.2 Protocol 2: Random Split Results

Table 2 shows results from the conventional random train-test split evaluation. Note the dramatically higher performance compared to Protocol 1.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
XGBoost	**89.64**	**89.83**	**89.64**	**89.65**
SVM	88.90	89.06	88.90	88.91
MLP	88.55	88.61	88.55	88.55
Random Forest	86.03	86.52	86.03	86.10
Logistic Reg.	77.96	78.19	77.96	77.98

Table 2: Protocol 2 (Random Split) Results - ■■ DATA LEAKAGE

■■ **WARNING:** These results appear excellent but are fundamentally misleading. They reflect the model's ability to recognize recording-specific patterns rather than generalizable sound characteristics.

5.3 Comparative Analysis: The Impact of Data Leakage

Table 3 presents the performance difference between the two protocols, quantifying the effect of data leakage.

Model	Protocol 1 (Correct)	Protocol 2 (Leakage)	Absolute Difference	Relative Increase
XGBoost	70.40%	89.64%	+19.24%	+27.3%
SVM	68.40%	88.90%	+20.50%	+30.0%
MLP	68.83%	88.55%	+19.72%	+28.6%
Random Forest	68.15%	86.03%	+17.88%	+26.2%
Logistic Reg.	67.66%	77.96%	+10.30%	+15.2%

Table 3: Performance Difference Between Evaluation Protocols

The performance gap ranges from 10.30 percentage points (Logistic Regression) to 20.50 percentage points (SVM), with an average increase of approximately 19 percentage points across all models. This represents a relative performance inflation of 15-30% depending on the algorithm.

6. Discussion

6.1 The Critical Importance of Evaluation Methodology

The most significant finding of this study is the dramatic impact of evaluation protocol on measured performance. The ~19 percentage point difference between protocols is not a marginal effect – it fundamentally changes conclusions about model capabilities.

Mechanism of data leakage: UrbanSound8K excerpts are extracted from longer field recordings. Multiple excerpts from the same recording share: (1) microphone transfer characteristics, (2) recording environment acoustics and reverberation, (3) background noise patterns, and (4) recording quality and artifacts. When random splitting places related excerpts in both training and test sets, the model can exploit these recording-specific signatures as shortcuts, achieving high test accuracy without learning transferable sound discrimination.

Real-world implications: A model showing 89.64% accuracy under Protocol 2 would likely achieve only ~70% accuracy when deployed on recordings from new microphones and environments. The 19% gap represents the cost of improper evaluation – researchers and practitioners would be misled about actual generalization capability, leading to deployment failures and eroded trust in ML systems.

6.2 Model Performance and Feature Engineering

XGBoost superiority: XGBoost achieves the highest accuracy (70.40%) under proper evaluation, consistent with its strong performance on tabular feature-based tasks in general. The gradient boosting approach effectively captures complex interactions between handcrafted acoustic features. The 2-3 percentage point advantage over other models, while modest, is consistent across folds.

Value of temporal features: The enhanced feature set including delta and delta-delta coefficients provides richer temporal context compared to static MFCCs alone. These derivatives capture the dynamics of spectral evolution, which is crucial for discriminating between sounds with similar spectral envelopes but different temporal patterns.

Performance clustering: The relatively small variance across models (67.66-70.40%) suggests that, given properly engineered features, model choice has less impact than evaluation methodology. This underscores that feature quality and evaluation correctness often matter more than sophisticated algorithms.

6.3 Feature Caching and Computational Efficiency

The MD5-based feature caching system provides 10-100× speedup on repeated experiments, transforming the development cycle from hours to minutes. This acceleration enables: (1) rapid iteration during feature engineering, (2) extensive hyperparameter search, (3)

comparison of many model variants, and (4) repeated verification runs. The computational benefit compounds across the research lifecycle – a small upfront investment in caching infrastructure yields substantial long-term productivity gains.

6.4 Limitations

Several limitations warrant discussion:

- **Feature selection:** Our handcrafted features, while interpretable, may miss patterns that end-to-end deep learning could discover automatically from raw audio or spectrograms.
- **Hyperparameter optimization:** Model hyperparameters were manually selected rather than systematically optimized via grid search or Bayesian methods, potentially leaving performance gains on the table.
- **Dataset scope:** Results are specific to UrbanSound8K. Generalization to other audio domains (medical sounds, animal vocalizations, industrial machinery) requires validation.
- **Class imbalance:** While we use weighted metrics, class imbalance in the dataset may still affect per-class performance, particularly for minority classes.
- **Temporal resolution:** Fixed 4-second excerpts may lose important temporal structure in longer sound events.

6.5 Comparison with Literature

Our XGBoost result (70.40% accuracy) with handcrafted features aligns with reported baselines in the literature for feature-based approaches on UrbanSound8K. State-of-the-art deep learning methods (e.g., convolutional neural networks on mel-spectrograms) typically achieve 75-85% accuracy, demonstrating the performance gap between handcrafted features and learned representations. However, our approach offers advantages in interpretability, computational efficiency, and data requirements that may be valuable in resource-constrained or safety-critical applications requiring explainability.

7. Conclusions and Future Work

7.1 Summary of Contributions

This study makes several contributions to audio classification research and practice:

- **Demonstrated critical impact of evaluation methodology:** The systematic comparison of evaluation protocols reveals a 19 percentage point performance gap attributable solely to data leakage, emphasizing that proper train-test splitting is as important as model architecture.
- **Established reproducible baseline:** Complete implementation with fixed random seeds, centralized configuration, and feature caching provides a verified baseline for future comparisons.
- **Quantified feature engineering value:** Enhanced features with temporal dynamics (delta coefficients) improve all models, demonstrating the continued relevance of domain knowledge in audio ML.
- **Validated efficient implementation:** Feature caching system enables 10-100× speedup, making extensive experimentation practical on standard hardware.
- **Identified best practices:** XGBoost with enhanced features achieves 70.40% accuracy under proper evaluation, serving as a strong classical ML baseline.

7.2 Lessons Learned

The most important lesson from this project transcends the specific domain of audio classification: **evaluation methodology critically determines the validity and interpretability of machine learning research.** A perfectly implemented model with state-of-the-art architecture becomes scientifically meaningless if evaluated on leaked data. Researchers must understand their data's structure (recordings, sessions, subjects, environments) and design evaluation protocols that match intended deployment scenarios.

Secondary lessons include: (1) handcrafted features with domain knowledge remain competitive for modest datasets, (2) computational optimization (caching) enables better science by removing infrastructure barriers, and (3) simple models often perform competitively when given quality features, reducing the rush toward complex architectures.

7.3 Future Work

Several promising directions for future research:

- **Deep learning comparison:** Train CNN models on raw spectrograms using the same compliant evaluation protocol to rigorously compare with feature-based approaches under identical conditions.

- **Hybrid approaches:** Combine handcrafted features with learned representations (e.g., using deep features as input to XGBoost) to potentially capture benefits of both paradigms.
- **Systematic hyperparameter optimization:** Apply grid search or Bayesian optimization to find optimal configurations for each model, potentially improving performance by several percentage points.
- **Per-class analysis:** Investigate which sound classes are most confused and why, potentially revealing opportunities for class-specific feature engineering or data augmentation.
- **Ensemble methods:** Combine predictions from multiple models to potentially achieve higher accuracy and robustness.
- **Cross-dataset evaluation:** Test models trained on UrbanSound8K on other urban sound datasets to assess transferability and identify domain-specific versus general features.
- **Real-time deployment:** Optimize the pipeline for low-latency inference to enable real-time urban sound monitoring applications.

7.4 Final Remarks

This project demonstrates that rigorous methodology and attention to evaluation details are as crucial as algorithmic sophistication in machine learning research. The dramatic 19% performance gap between proper and improper evaluation protocols serves as a cautionary tale for researchers across all ML domains. As the field continues to advance toward more complex models and larger datasets, maintaining methodological rigor becomes increasingly important to ensure that reported results reflect true capabilities rather than artifacts of evaluation design.

Our implementation, achieving 70.40% accuracy with XGBoost and handcrafted features under proper evaluation, establishes a solid baseline for urban sound classification. While deep learning may achieve higher absolute performance, the interpretability, efficiency, and reproducibility of feature-based approaches ensure their continued relevance in research and practical applications.

References

Salamon, J., Jacoby, C., & Bello, J. P. (2014). A dataset and taxonomy for urban sound research. In *Proceedings of the 22nd ACM international conference on Multimedia* (pp. 1041-1044).

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785-794).

Pedregosa, F., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830.

McFee, B., et al. (2015). librosa: Audio and music signal analysis in python. In *Proceedings of the 14th Python in Science Conference* (Vol. 8, pp. 18-25).

Davis, S., & Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4), 357-366.

Piczak, K. J. (2015). Environmental sound classification with convolutional neural networks. In *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)* (pp. 1-6). IEEE.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5-32.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273-297.

Appendix A: Complete Hyperparameter Configurations

This appendix provides complete hyperparameter specifications for reproducibility.

Component	Parameter	Value
Audio Processing	Target Sample Rate	`16000 Hz`
	Audio Duration	`4.0 seconds`
	Normalization	`Peak normalization to [-1, 1]`
Feature Extraction	n_mfcc	`13`
	n_fft	`2048`
	hop_length	`512`
	n_mels	`128`
Cross-Validation	Number of Folds	`10 (predefined)`
	Random State	`42`
Random Split	Test Size	`0.20`
	Validation Size	`0.125`
	Stratification	`By class`
Logistic Regression	max_iter	`1000`
	random_state	`42`
SVM	kernel	`rbf`
	C	`10`
	gamma	`scale`
Random Forest	n_estimators	`200`
	max_depth	`20`
	min_samples_split	`5`
	random_state	`42`

XGBoost	n_estimators	`400`
	learning_rate	`0.05`
	max_depth	`6`
	subsample	`0.9`
	random_state	`42`
MLP	hidden_layer_sizes	`(128, 64)`
	activation	`relu`
	max_iter	`500`
	random_state	`42`