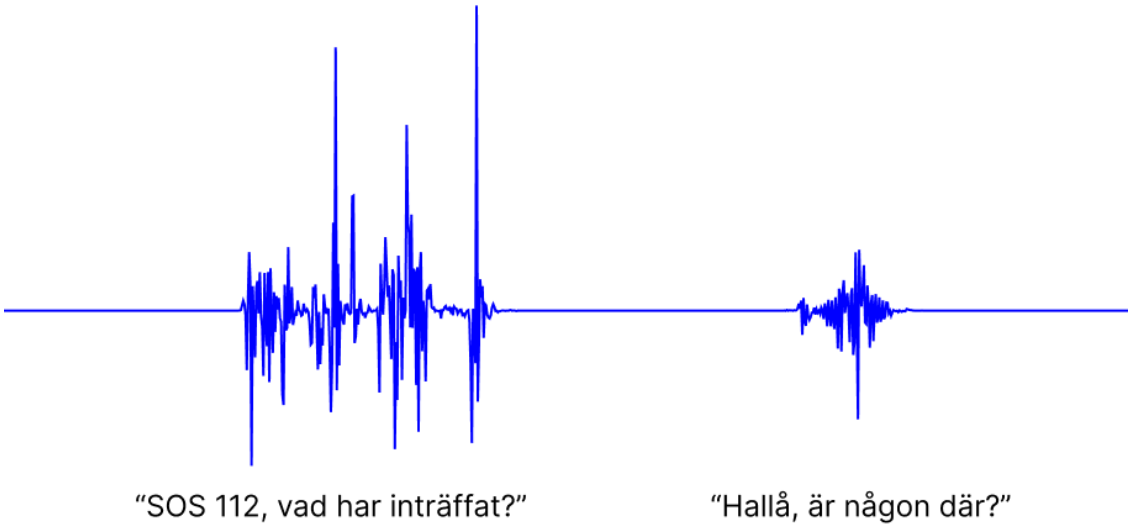


Early Classification of Silent Emergency Calls

Exploratory Audio-Based Detection of Pocket-Call-Like Acoustic Patterns

Master’s thesis in Data Science and AI

PONTUS GRANLI HOLMBERG

KRISTOFFER REIMERTZ

DEPARTMENT OF MATHEMATICAL SCIENCES

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2026
www.chalmers.se

Master's thesis 2026

Early Classification of Silent Emergency Calls

Exploratory Audio-Based Detection of Pocket-Call-Like Acoustic Patterns

PONTUS GRANLI HOLMBERG
KRISTOFFER REIMERTZ

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Mathematical Sciences
Division of Applied Mathematics and Statistics
Chalmers University of Technology
Gothenburg, Sweden 2026

Early Classification of Silent Emergency Calls
Exploratory Audio-Based Detection of Pocket-Call-Like Acoustic Patterns
PONTUS GRANLI HOLMBERG
KRISTOFFER REIMERTZ

Supervisor: Klas Modin, Mathematical Sciences
Supervisor: Vigfus Valgeirsson, Omda
Examiner: Klas Modin, Mathematical Sciences

Master's Thesis 2026
Department of Mathematical Sciences
Division of Applied Mathematics and Statistics
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Early operator actions when receiving a silent emergency call.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2026

Early Classification of Silent Emergency Calls
Exploratory Audio-Based Detection of Pocket-Call-Like Acoustic Patterns
PONTUS GRANLI HOLMBERG
KRISTOFFER REIMERTZ
Department of Mathematical Sciences
Chalmers University of Technology

Abstract

Silent calls to the Swedish emergency number 112 place unnecessary strain on emergency-call operations and may delay responses to critical incidents. This thesis investigates whether accidental silent emergency calls can be identified using only caller-side audio. In particular, the work explores the classification of a manually defined subgroup of ”pocket calls”.

Two audio feature representations were evaluated: handcrafted acoustic feature representations and pretrained self-supervised BEATs embeddings. Logistic regression and XGBoost classifiers were evaluated across multiple experimental settings, including delayed-speech scenarios, respiratory-audio analysis, and early detection using short audio intervals.

The results showed that classification of the complete ”Silent 112” operator-assigned category resulted in poor and unstable performance, while significantly stronger performance was achieved for the ”pocket call” subgroup. Across all experiments, BEATs embeddings outperformed handcrafted features. The strongest performance was achieved using BEATs embeddings together with logistic regression on a 10-second audio window, achieving a recall of 0.92 with no false positives on the held-out test set. Additionally, reliable classification became increasingly feasible after approximately 6 to 8 seconds of available audio.

Although the findings are preliminary and limited by the relatively small number of manually identified ”pocket calls”, the results demonstrate the potential of caller-side acoustic analysis for supporting automated identification of accidental silent emergency calls.

Keywords: classification, emergency, machine learning, silent 112.

Acknowledgements

We would like to express our sincere gratitude to our supervisor at Omda, Vigfus Valgeirsson, as well as to the other people involved from Omda, for support throughout the project. We would also like to thank SOS Alarm and its personnel for granting us access to their facilities and for their assistance during the work. Lastly, we would like to thank our supervisor at Chalmers, Klas Modin, for his valuable insights and guidance.

Pontus Granli Holmberg, Gothenburg, June 2026
Kristoffer Reimertz, Gothenburg, June 2026

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

BEATs	Bidirectional Encoder representation from Audio Transformers
FNR	False Negative Rate
FPR	False Positive Rate
LR	Logistic Regression
OOF	Out-Of-Fold
PR-AUC	Precision-Recall Area Under the Curve
RMS	Root Mean Square
SSL	Self-Supervised Learning
VAD	Voice Activity Detection
XGBoost	Extreme Gradient Boosting
ZCR	Zero Crossing Rate

Contents

List of Figures

List of Tables

1
Introduction

Emergency response centers receive millions of calls each year. In Sweden alone, the national emergency number (112) receives approximately 3.5 million calls annually [1, 2, 3, 4, 5, 6]. However, approximately one-third of all calls constitute non-emergencies, and by far the most common type are so called ”Silent 112” calls, meaning that the operator is unable to establish contact with the caller [5]. As seen in Figure 1.1 these calls account for roughly 60% of all non-emergencies, which corresponds to approximately 600,000 calls. Typically, these calls are made on accident, unbeknownst to the caller; often caused due to devices having built-in easy-dialing features to the emergency number, or sensors within a device falsely indicating an accident has occurred, triggering the call [7, 8]. This becomes an issue for operators as they must continuously handle new incoming calls, and with the abundance of silent calls being addressed, this contributes to longer waiting times, possibly putting other help seekers at risk [8, 9]. In addition to this, operators must be able to identify if a silent call is actually being made by a person seeking help but incapable of speaking, adding possibly extra stress and prolonged decision making times. There could very well be situations where the caller is unable to speak due to illness or a perpetrator being in their proximity [7, 10]. In such cases, the caller might resort to non-lexical cues such as respiratory sounds or rustling noises [10].

1.1 Background and Problem Formulation

This study is conducted with guidance from Omda Emergency AB. Omda is a supplier of software products related to healthcare and emergency services, with SOS Alarm being among its clients. Omda seeks to explore how machine learning methods could be utilized to better manage silent calls and the risk they pose for SOS Alarm. Within the scope of the collaboration, real call data has been obtained from SOS Alarm.

All calls are supplemented with metadata, containing information about call duration, general location, time of day among other things. Additionally, associated with each call is a label containing information about the call type, with ”Silent 112” being one of the options. The data has been manually labeled by operators, though this is not necessarily among their primary work duties, implying that a non-negligible fraction of the calls are mislabeled. Moreover, within the ”‘Silent 112” class, there is considerable variability in the sounds emitted across calls. The one common denominator is that the operator fails to establish any form of contact.

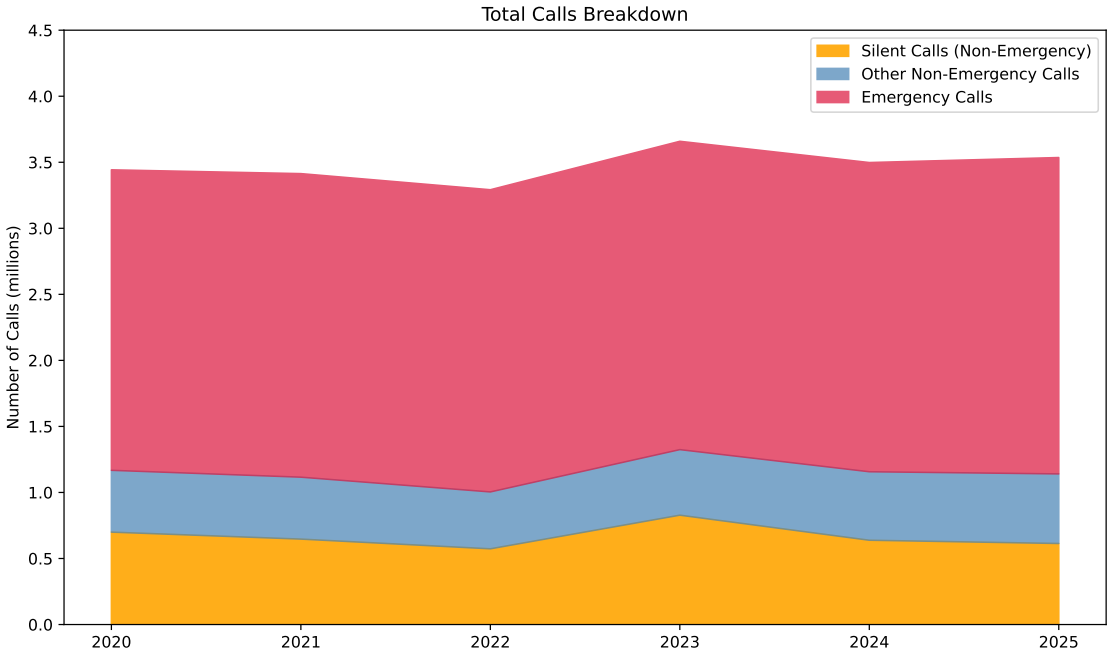


Figure 1.1: Breakdown of emergency and non-emergency calls from 2020 to 2025 [1, 2, 3, 4, 5, 6]. ”Silent 112” calls make up a large share of non-emergency calls.

The ”Silent 112” label is rather deceiving as it encompasses both identifiable sounds and ambiguous sounds, as well as complete silence, sometimes all occurring within a single call.

The heterogeneity in the ”Silent 112” data makes it an ill-suited classification target, especially in the context of emergency call handling, where misclassifications can have severe complications for human life. To mitigate this risk, the scope of the project narrowed down to a potential subgroup of particular interest, namely ”pocket calls”. These calls can loosely be defined as sounds emitting from a device rubbing against the inside fabric of a pocket. This might still sound ill-defined and it is difficult to say objectively what characterizes a ”pocket call” (or if such a distinction can even be made). To justify the construction of this subgroup, a limited part of the study is devoted to classification of the broader category of ”Silent 112” calls, followed up with a subsequent focus on ”pocket calls”.

A major focal point concerns the intended end product. A practical use case might relate to identifying ”pocket calls” in the period between call initiation and operator answer, or include this information to assist the operator in assessing the call. The national requirement stipulates that this waiting time must not exceed eight seconds on average [5]. Compliance with this requirement varies from year to year; from 2023 to 2025 the average waiting time fluctuated from 10.9, 6.6 to 7.3 seconds [4, 5, 6]. Thus, focusing on classifying calls within a time range of 5-10 seconds seems reasonable. A major caveat, however, is that the audio files did not include the call-in waiting time period.

1.2 Related Work

To the best of our knowledge, little research exists on the matter of employing machine learning models targeting silent calls specifically. Existing approaches to emergency classification rely on direct interaction with the caller through speech [11, 12, 13]. Despite the limited studies, a notable proof-of-concept model constructed in 2021 showed promising results by treating involuntary calls as an audio classification task, indirectly targeting silent calls [14]. The audio files were converted to spectrograms, and used to train a classifier, yielding a precision score of 84% and a recall score of 67%. Though, there is a lack of scientific support as there is no mention of what model is used and the methodological information is scarce and no follow-up has been reported since. Furthermore, the model was trained on just 300 calls with three categories: *urgent*, *non-urgent* and *invalid* (involuntary) calls. Each category consisted of 100 calls, all of which were 30 seconds long.

In the absence of speech, the problem at hand inherently relies on the acoustic context associated with pocket calls, lending credence to the belief that this can be framed as an audio classification problem. A commonly used benchmark for such a task is the ESC-50 dataset; a dataset comprised of 50 sound categories, including human sound events and environmental sounds. The baselines provided for ESC-50 were achieved by extracting the zero-crossing rate and mel-frequency cepstral coefficients (MFCCs) and utilizing k-NN, SVM and random forest ensemble; yielding 32.2%, 39.6% and 44.3% accuracies respectively [15]. Though seemingly simple in nature and providing favorable baselines, such shallow, hand-crafted approaches have limited improvement capabilities, as stated by [16] in 2015. As a matter of fact the authors strongly recommended exploring alternative paradigms. Interestingly, this coincided with the simultaneous rise of deep learning, enabling precisely such a shift. An additional strength of deep learning methods lies in their ability to process raw data directly in conjunction with uncovering deep-rooted complex patterns from the data, allowing for significant improvements over previous methods where human expertise was at the core [17].

One concern with deep learning is that it requires a large amount of annotated data to train a well performing model [18, 19]. To address this limitation, self-supervised learning (SSL) has become a viable strategy to tackle this issue. The core idea is to learn meaningful representations in the form of latent embeddings on a substantial volume of unlabeled data [19]. These representations can subsequently be applied for various downstream tasks. This approach has shown great potential when it comes to audio classification tasks, with BEATs (Bidirectional Encoder representation from Audio Transformers) achieving 98.1% accuracy on the ESC-50 dataset, comfortably surpassing the human baseline of 81.3% [20, 15].

A major caveat of the aforementioned methods is that they have been evaluated on a classification task with 50 categories. This is an important distinction from the problem at hand in this thesis, which places its focus on distinguishing a specific, subjectively defined ”pocket call” or the whole ”Silent 112” group type versus emer-

gency calls, whose audio content may differ greatly. However, the pretrained audio embeddings from BEATs may still provide useful representations that can be used to train a classifier specific to this task.

1.3 Aim

The aim of this thesis is to investigate machine learning-based detection of silent emergency calls, with a specific focus on calls exhibiting pocket-call-like characteristics. In particular, the thesis evaluates how different audio representations and segmentation strategies affect performance.

1.4 Research Questions

The thesis is guided by four research questions:

1. To what extent can accidental silent emergency calls be distinguished from genuine emergency calls using machine learning on early-stage emergency call audio?
2. How does specifying the classification task from the broad ”Silent 112” operator category to a manually defined ”pocket call” category affect classification performance?
3. How do different audio representations affect the detection of accidental emergency calls?
4. How does model performance change as the amount of available call duration is progressively reduced?

1.4.1 Limitations

As mentioned in section 1.1 the data contains mislabeled data. This is something we have been able to confirm by listening to a significant amount of calls and inspecting their assigned labels. As part of the process of identifying ”pocket calls”, our main objective became to construct subgroups within the ”Silent 112” category, with ”pocket calls” being our main focus. However, this presents another limitation as the identified subgroups are i. subjectively identified by us and ii. correctly labeling the data within a correct subgroup proved difficult, meaning our own relabeled sets contain errors. Furthermore, we only focused on relabeling calls within the ”Silent 112” category, meaning that the other classes still contain mislabeled data.

A major shortcoming of the data is that the audio from the waiting time period that the caller waits during the call-ins is not provided. We believe that this is the most interesting part of the call, as it may help determine whether a call is accidental before an operator has even answered the call.

1.5 Data Protection and Ethical Considerations

The dataset includes extremely sensitive personal health information, which requires serious consideration of privacy and data protection. These considerations lie outside the scope of this work and it is worth emphasizing that no model incorporates personal information in its assessment. Moreover, the report will not reveal any information regarding any individual calls, nor when the data was collected.

1.5.1 Risk Assessment under the European Union AI Act

The European Union AI Act regulates the use of AI within the union. The purpose of the act is to address the potential dangers that AI pose on the fundamental rights and livelihoods on the member states' citizens. The framework specifies both the risks with AI and procedures to manage these by assigning the AI systems to one of four risk categories: *Minimal or no risk*, *Transparency risk*, *High risk* and *Unacceptable risk*, sorted from least to most severe [21]. AI systems falling under the latter category are strictly prohibited.

In Article 6(2), ANNEX III [22] it is stated that "AI systems intended to evaluate and classify emergency calls by natural persons or to be used to dispatch, or to establish priority in the dispatching of, emergency first response services, including by police, firefighters and medical aid, as well as of emergency healthcare patient triage systems" are to be considered high risk. The risk category assessment is motivated in Recital 58 as the matter concerns critical decision making procedures regarding human safety [23]. However, Article 6(3) declares that an AI system is exempt from the high risk category if it does not materially influence the evaluator's final verdict [24].

It is not unreasonable to think that if an AI system were to be developed in the current context, it would fall under the high risk category, which means it would be necessary to comply with strict regulations as stated by [21]. This includes adopting preemptive measures such as risk assessment, avoiding discriminatory bias, human supervision, clear documentation, and ensuring that the system is robust and highly accurate, among other things.

2
Theory

The following chapter introduces underlying theories for the models employed, as well as the foundations for transforming audio to useful representations that can be processed by machine learning models. Additionally, theoretical approaches to measuring model performance are presented.

2.1 Audio Representations

In order for audio signals to be processed by machine learning models, they need to be converted to useful representations. These representations can be modeled explicitly using handcrafted-based features, or learned from the raw signal by utilizing neural-based networks.

2.1.1 Handcrafted Features

Handcrafted audio features are low-level descriptors computed from either time, frequency or spectral representations of signals [16]. A key aspect when it comes to extracting the features is the assumption of the signal being stationary [25]. In principle, this means operating over short time frames as the properties of the signal are diluted over time. The signal is typically split into 10-30 ms long overlapping frames [25]. Although this assumption is considered in the context of speech processing, the same principle holds for more general audio classification tasks [26]. Some of the more useful audio descriptors in this context include root mean square (RMS) energy, zero-crossing rate (ZCR), spectral centroid, roll-off, flatness and bandwidth [26]. These are described in Table 2.1.1.

2.1.2 Self-Supervised Embeddings

Self-supervised learning (SSL) is a machine learning technique in which a model learns to create meaningful representations from massive amounts of unlabeled data by solving pretext tasks [33]. These tasks are designed by humans and depend on the domain (natural language processing, computer vision, etc). Once a model is pretrained in this manner it can be used as a feature extractor to extract rich representations (embeddings) from the input audio to be used as features for downstream tasks.

A notable difference between SSL audio models is their pretraining domain. Popular models like wav2vec 2.0 [34] and HuBERT [35] are pretrained on speech corpora and

Table 2.1: Overview of commonly used handcrafted audio features.

Feature	Definition	Description
Root Mean Square (RMS) Energy	$\mathrm{RMS}(\mathbf{x}) = \sqrt{\frac{1}{N}\sum_{n=1}^{N} x[n]^2}$	Reflects the energy level of a signal and can be useful for estimating loudness [27, 28].
Zero-Crossing Rate (ZCR)	$\mathrm{ZCR}(\mathbf{x}) = \frac{1}{2N}\sum_{n=1}^{N} \lvert\mathrm{sign}(x[n]) - \mathrm{sign}(x[n-1])\rvert$	Measures how frequently the waveform crosses the zero-amplitude axis [29, 30]. The crossing rate can give an indication whether the signal is periodic or not as periodic signals generally have a low zero-crossing rate [29].
Spectral Centroid	$\mathrm{SC}(\mathbf{x}) = \frac{\sum_{k=1}^{K} F[k]\,k}{\sum_{k=1}^{K} F[k]}$	Represents the weighted average of the frequency bins positions, indicating where the energy is located [27, 31]. A high centroid value suggests that the signal is dominated by high-frequency elements [32, 27, 31].
Spectral Roll-off	$\mathrm{SR(x)} = \min\left\{k : \sum_{j=1}^{k} F[j] \geq \alpha \sum_{j=1}^{K} F[j]\right\}$	Provides the frequency point below which a pre-specified distribution α of the spectral magnitude is located [31].
Spectral Flatness	$\mathrm{SF}(\mathbf{x}) = \frac{\left(\prod_{k=1}^{K} F[k]\right)^{1/K}}{\frac{1}{K}\sum_{k=1}^{K} F[k]}$	Gives an indication of the noisiness of a signal and is computed as the quotient of the geometric and arithmetic average of the spectral magnitudes [29].
Spectral Bandwidth	$\mathrm{SB}(\mathbf{x}) = \frac{\sum_{k=1}^{K} \lvert k - \mathrm{SC}(\mathbf{x})\rvert\, F[k]}{\sum_{k=1}^{K} F[k]}$	Represents the frequency range weighted by the distances from the spectral centroid [27].

Here, $\mathbf{x}$ denotes a complete audio frame, $x[n]$ is the n-th sample in a frame of length N, k denotes the frequency bin, $F[k]$ is its magnitude, K is the number of frequency bins, and α is the roll-off threshold. The sign function is defined as $\mathrm{sign}(x[n]) = 1$ if $x[n] \geq 0$ and -1 otherwise.

therefore produce representations optimized for such tasks. Although these models achieve *state-of-the-art* (SOTA) performance on speech tasks, they have limited exposure to non-speech events.

BEATs [20] is an SSL model that has been pretrained on general audio from the AudioSet-2M dataset, which contains approximately 2 million 10-second clips, spanning 527 classes of sound events [36]. These classes include various general audio events, but most notably human related events like breathing, crying, whispering, coughing. BEATs is trained using a semantic prediction objective such that the representation encodes what kind of sound is present rather than focusing on redundant acoustic noise and signal-level details. This combination of general audio domain and semantic representation makes BEATs a relevant model for involuntary call detection, where the distinctive signals between a voluntary and non-voluntary silent call are non-speech audio events.

The BEATs framework, as illustrated in Figure 2.1, employs an acoustic tokenizer on 128-dimensional Mel-filter bank features, which are then split into 16×16 grid patches and flattened into a sequence $\mathrm{X} = \{\mathrm{x}_t\}_{t=1}^{T}$. The resulting tokens $\hat{\mathrm{Z}} = \{\hat{\mathrm{z}}_t\}_{t=1}^{T}$ acts as target labels during the SSL pretraining phase. Specifically, the audio SSL model employs the Vision transformer architecture, where audio patches are fed into a linear projection layer and transformer encoder. However, only 25% of the patches are passed through as masking is applied to the input patches at positions $\mathcal{M} = \{1, \ldots, T\}^{0.75T}$, i.e. only patches $\mathrm{X}^U = \{\mathrm{x}_t : t \in \mathcal{M}\}_{t=1}^{T}$ are fed into the system, producing representations $\mathrm{R}^U = \{\mathrm{r}_t : t \in \mathcal{M}\}_{t=1}^{T}$. These are subsequently combined with the unmasked patch features, $\{\mathrm{r}_t : t \in \mathcal{M}\}_{t=1}^{T} \cup \{0 : t \notin \mathcal{M}\}_{t=1}^{T}$ and passed on to another transformer acting as label predictor, resulting in predicted labels $\mathrm{Z} = \{\mathrm{z}_t\}_{t=1}^{T}$. Finally, cross-entropy loss is used to compute the correct label $\hat{z}$ given the unmasked patches $\mathrm{X}^U : \mathcal{L}_{\mathrm{MAM}} = -\sum_{t \in \mathcal{M}} \log p(\mathrm{z}_t = \hat{z}_t \mid \mathrm{X}^U)$.

After one audio SSL model has been trained, it is subsequently used for training the tokenizer by encoding X to the sequence $\hat{\mathrm{O}} = \{\hat{\mathrm{o}}_t\}_{t=1}^{T}$ through knowledge distillation, as seen in Figure 2.2. The tokenizer's objective is to predict the teacher model' output through a transformer estimator, i.e. $\mathrm{O} \approx \hat{\mathrm{O}}$, with a cosine similarity loss. It's worth noting that the tokenizer's output is quantized to discrete labels by mapping the represenations to a codebook vector, and in the first iteration the discrete labels are obtained through a random-projection tokenizer. By the third iteration, BEATs is able to outperform all previous SOTA models on the AudioSet dataset.

2.1.3 Voice Activity Detection

Voice activity detection concerns the task of identifying human speech. Historically, these conventional algorithms relied on handcrafted features features such as energy thresholds, zero-crossing rate and spectrum analysis [37]. While impressive results have been achieved, challenges persist regarding robustness across different environments. Modern approaches utilizing neural-based networks have shown major im-

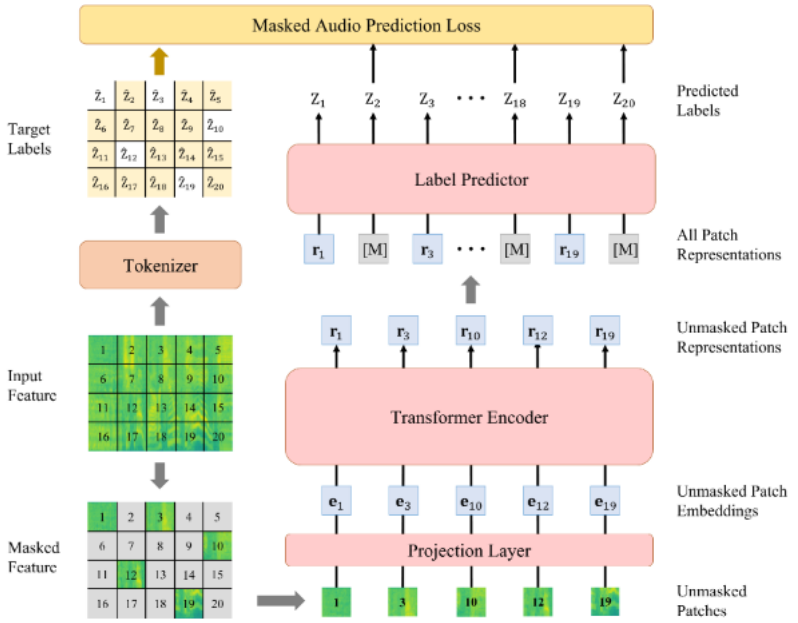


Figure 2.1: Training setup of the audio SSL model. The model is trained to predict the correct tokens through masked learning. Adapted from Chen et al. [20]. Licensed under CC BY 4.0; cropped.

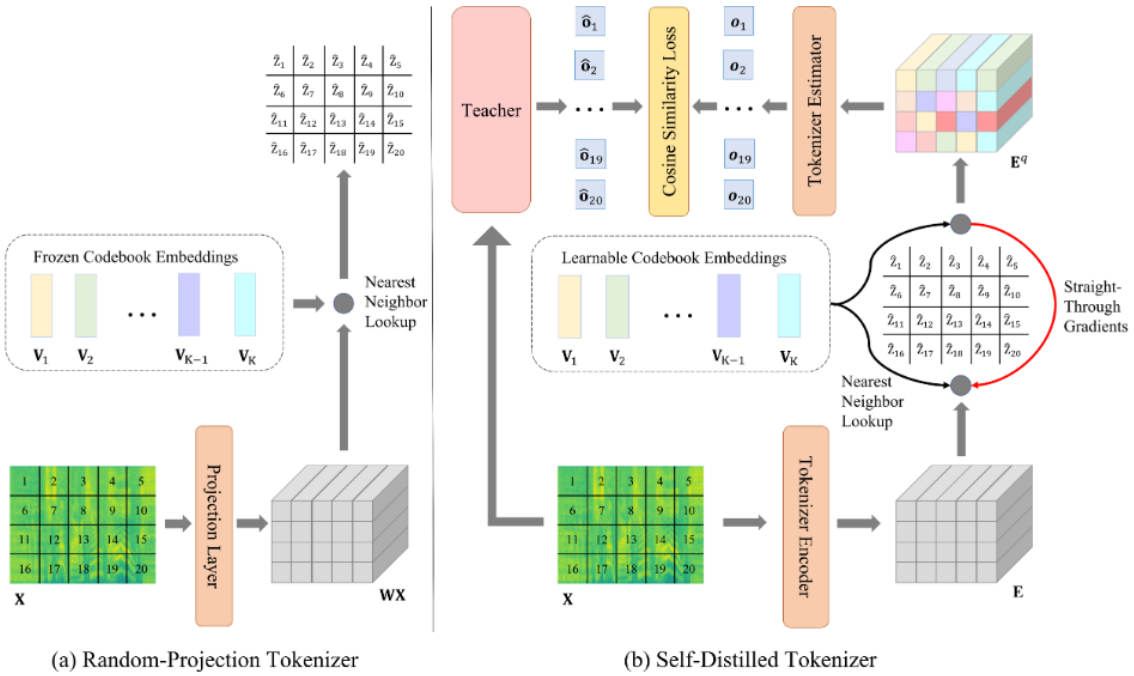


Figure 2.2: Illustration of tokenization procedure. In the first iteration (a) a random-projection tokenizer generates tokens. In subsequent iterations (b) the audio SSL model trains the tokenizer encoder through knowledge distillation, resulting in more informative tokens. Reproduced from Chen et al. [20]. Licensed under CC BY 4.0.

provements compared with conventional methods, not least in noisy environments. [38]. Silero VAD is an example of a recently developed neural-based VAD model; employing multi-head attention (MHA) with Short-Time Fourier transform (STFT) features [39]. The evaluation metrics, provided from eight datasets, including both clean and noisy speech in various settings showcase strong capabilities. The model achieves AUC scores of around 0.94-0.98 across all datasets, except for one where it achieves a score of 0.79 [40].

2.2 Classification Models

Classification models are supervised learning methods used to predict a discrete class label from a set of input features. Numerous models exist; here two commonly used classifiers are presented: logistic regression and XGBoost.

2.2.1 Logistic Regression

Binary logistic regression [41] is a classifier that is often appropriate to use when interpretability is important. The model provides an interpretable description of the relationship between a binary outcome variable Y and a set of predictor variables x. More formally, let

$$\pi(x) = \Pr(Y = 1 \mid X = x), \tag{2.1}$$

denote the conditional probability that the outcome equals 1. In contrast, $1 - \pi(x)$ denotes the conditional probability that the outcome equals 0. The log-odds of the outcome are assumed to be a linear function of the predictors:

$$\log\left(\frac{\pi(x)}{1-\pi(x)}\right) = \beta_0 + \beta^T x. \tag{2.2}$$

Solving for $\pi(x)$ gives the logistic function:

$$\pi(x) = \frac{\exp(\beta_0 + \beta^T x)}{1 + \exp(\beta_0 + \beta^T x)}, \tag{2.3}$$

which is equivalent to $\pi(x) = \frac{1}{1+\exp[-(\beta_0+\beta^T x)]}$. This corresponds to applying the sigmoid function to $\beta_0 + \beta^T x$.

2.2.2 XGBoost

XGBoost (eXtreme Gradient Boosting) is a scalable machine learning algorithm based on gradient boosting. *Boosting* is a technique in which multiple "weak learners", i.e. models with limited predictive ability are combined to form a single prediction rule [42]. Decision trees are commonly adopted as weak learners [43]. The algorithm is formulated as follows [44]: given N samples $\{(a_i, b_i)\}_{i=1}^{N}$ where a_i denotes the observed features, and b_i the associated labels, the boosting procedure is carried out over T iterations. In iteration t each a weak learner h_t is trained with respect to a distribution D_t over the training examples. This distribution determines

the relative importance of each example in the current iteration. The weighted error of h_t is given by:

$$\epsilon_t = \Pr_{i\sim D_t}\left[h_t(a_i) \neq b_i\right] = \mathbb{E}_{i\sim D_t}\left[\mathbf{1}\left\{h_t(a_i) \neq b_i\right\}\right] = \sum_{i=1}^{N} D_t(i)\mathbf{1}\left\{h_t(a_i) \neq b_i\right\}. \quad (2.4)$$

The errors are used to update D_t to D_{t+1}, meaning that examples misclassified by h_t are assigned higher weight in distribution D_{t+1}. Consequently, the next learner h_{t+1} is trained with respect to D_{t+1}. Despite the fact that each learner might only perform slightly better than random guessing under their respective distributions, the resulting aggregation of the T predictions demonstrates strong predictive capability. In the case of binary classification with $h_t(a_i) \in \{-1, +1\}$, the final classifier is defined as:

$$H(a) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(a)\right), \quad (2.5)$$

where α_t is a weight adjusts how much influence each learner has.

Gradient boosting extends the idea of boosting by optimizing over more general loss functions [**j**, 45]. The algorithm is as follows [46, 45]: a model F_{t-1} is updated by introducing a new weak learner h_t, and constructing the update as $F_t(a) = F_{t-1}(a) + \eta h_t(a)$, where η is the learning rate. Hence after T iterations:

$$F_T(a) = F_0(a) + \sum_{t=1}^{T} \eta h_t(a). \quad (2.6)$$

By computing the loss function $\mathcal{L}(F_t) = \sum_{i=1}^{N} \mathcal{L}(b_i, F_t(a_i))$ and computing the negative gradient, so called pseudo-residuals r_{it} are obtained [47]:

$$r_{it} = -\left[\frac{\partial L(b_i, F(a_i))}{\partial F(a_i)}\right]_{F=F_{t-1}}. \quad (2.7)$$

The weak learner's h_t objective is to correct for these error such that $h_t \approx r_{it}$, i.e. the training set of interest is $(\{a_i, r_{it}\})_{i=1}^{N}$ [45, 46].

XGBoost follows the same ensemble approach; the main difference is that XGBoost introduces optimization techniques that make tree boosting scalable [46]. It makes efficient use of parallelization and better utilization of cache access and out-of-core computation. As for the algorithmic aspect, during the tree splitting procedure histograms are utilized to find split points with a greedy approach. XGBoost also incorporates sparsity-aware split finding, meaning it can handle missing data efficiently. Moreover, mechanisms for controlling overfitting are also built into the model, such as subsampling, shrinking the learning rate, and regularization of leaf weights [48]. Empirically, XGBoost has shown strong capabilities in supervised machine learning tasks that involve tabular data, often outperforming deep models [49, 50].

2.3 Validation Strategies

Reliable model validation is important to ensure that a machine learning model generalizes well to unseen data. The choice of validation strategy depends on the structure and characteristics of the underlying dataset, such as class balance and sample independence. This section outlines the cross-validation techniques used and details how grouping constraints are applied to prevent data leakage in correlated datasets.

2.3.1 Cross-Validation

Cross-validation is a resampling-based validation strategy commonly used to evaluate and estimate the generalization of machine learning models [51]. In k-fold cross validation, the dataset is partitioned into k subsets, or "folds", of equal size. The model is then trained on $k - 1$ folds and evaluated on the remaining fold. This process is repeated until each fold has acted as the validation fold. *Stratified* k-fold cross validation is a cross-validation technique that ensures that each fold contains the same class proportions as the original dataset, which is crucial for imbalanced data where a standard fold might entirely lack minority class samples.

By collecting the model's predictions on each validation fold, a complete set of *out-of-fold* (OOF) predictions is generated. Because every sample is predicted using a model that never saw it during training, aggregating these OOF predictions allows for an unbiased performance estimation across the entire training dataset.

2.3.2 Grouped Validation

Standard cross-validation assumes that the samples are independent. However, if multiple samples originate from the same source call, this assumption might not hold. In segment-based audio classification, segments extracted from the same source call may share similar acoustic patterns. If segments of the same call are distributed across both training and validation folds, there is a risk of data leakage.

Grouped cross-validation addresses this issue by ensuring that all samples originating from the same source call are assigned to the same fold [52]. This is achieved by assigning a group identifier based on the source call and ensuring that all segments belonging to the same group are assigned to the same fold.

2.4 Performance Evaluation Metrics

Evaluation metrics are important aspects for giving a comprehensive overview of model performance, especially in the case of imbalanced datasets. Informative metrics in these settings include confusion matrices, precision, recall, specificity false positve rate (FPR), false negative rate (FNR), precision-recall (PR) curves, area under the precision-recall curves (PR-AUC) and F1-scores.

In the binary classification setting, the classes are denoted as *positive* and *negative* [53]. The resulting confusion matrix forms a 2×2 grid, comparing the model's prediction with the true values. It is a visual summarization, with a layout as shown in Table 2.2. The true positive and true negative cells displays the correctly classified data points, whereas false positives indicate negatives misclassified as positives, and vice versa for the false negatives.

Table 2.2: Table showcasing confusion matrix in binary classification setting. It is desirable to have high true positive and true negative rates.

	Predicted class	
True class	Negative	Positive
Negative	True Negative (TN)	False Positive (FP)
Positive	False Negative (FN)	True Positive (TP)

An important distinction is the rate at which data is correctly classified in relation to its misclassifications. Recall, precision and specificity provide useful insight to model performance in regards to this [53, 54]. These are defined as:

$$\text{Precision} = \frac{TP}{TP + FP}, \tag{2.8}$$

$$\text{Recall} = \frac{TP}{TP + FN}, \tag{2.9}$$

$$\text{Specificity} = \frac{TN}{TN + FP}. \tag{2.10}$$

Specificity measures the proportion of negative samples correctly identified as negatives. However, in safety critical applications it might be more intuitive to track the rate of error instead. The False Positive Rate (FPR) is the complement to specificity and represents the proportion of actual negative samples that are incorrectly classified as positives [55]:

$$\text{FPR} = \frac{FP}{TN + FP}. \tag{2.11}$$

Similarly, the complement to recall, false negative rate (FNR), measures the proportion of positive samples that are incorrectly classified as negative, and is defined as [55]:

$$\text{FNR} = \frac{FN}{TP + FN}. \tag{2.12}$$

Furthermore, the F1-score metric is a measurement that combines both precision and recall by computing the harmonic mean of the two [54]. The score is defined as:

$$F1 = \frac{2 \cdot Precion \cdot Recall}{Precision + Recall}. \tag{2.13}$$

PR curves are suitable in instances where the positive class is heavily outnumbered [53]. It gives an overview of how different classification thresholds values affect the precision and recall rates. Moreover, by computing the PR-AUC score, an overview of the overall performance of the classifier across all threshold can be obtained. The higher the area, the more appropriate is the classifier.

2.5 Statistical Uncertainty

Performance metrics estimated on finite sampled datasets are subject to statistical uncertainty. This is especially important when evaluating small test sets, as a single classification outcome can drastically change the reported performance metric. Confidence intervals provide an estimate of the range within which the true performance metric is likely to lie.

2.5.1 Wilson Score Intervals

The Wilson score interval is a method used to compute confidence intervals for binomial proportions, such as accuracy or recall [56]. Compared to normal approximation methods, Wilson intervals are more accurate for small sample sizes, or when proportions are close to 0 or 1. The lower and upper bounds of the Wilson score interval are calculated as follows:

$$\mathrm{CI} = \frac{\hat{p} + \frac{z^2}{2n} \pm z\sqrt{\frac{\hat{p}(1-\hat{p})}{n} + \frac{z^2}{4n^2}}}{1 + \frac{z^2}{n}}. \tag{2.14}$$

Where $\hat{p}$ is the observed proportion, n is the sample size, z is the z-score corresponding to the desired confidence level.

2.5.2 Bootstrap Confidence Intervals

Bootstrap confidence intervals estimate uncertainty by repeated resampling with replacement of the evaluation data [57]. Bootstrap confidence intervals are particularly useful for performance metrics such as F1-score, ROC-AUC, and PR-AUC, for which confidence intervals cannot easily be calculated using analytical formulas. Stratified bootstrapping ensures that the proportion of classes stays the same as the target dataset.

Bootstrap is usually performed for thousands of iterations. For each iteration, a random sample of equal size to the target dataset is generated by randomly drawing observations with replacement. Performance metrics are then computed for each resampled dataset, from which the empirical distributions are used to calculate the intervals from the quantiles of these distributions.

3
Methodology

The following chapter presents an in-depth overview of the dataset used, the methods utilized, the analysis procedures, and the motivations for the actions taken. The objective of the work was not only to maximize classification performance, but to investigate how task formulation, negative-class construction, and representation strategy affect the detectability of accidental emergency calls.

3.1 Dataset Description

The dataset amounted to 9752 real calls received by SOS Alarm. All calls lasting less than three seconds were discarded due to the short time frame, resulting in 9337 files. The data mirrored real life proportions, with ”Silent 112” making up approximately 17% of the data. In order to obtain a representative collection reflecting human behavior, the data, which were randomly sampled, included both weekdays and weekends and all times of the day.

3.2 Dataset Limitations and Problem Reformulation

Several limitations were identified in the available dataset during the initial stages of the project. In particular, differences between the intended outcome and the available audio data, together with the broad nature of the operator-assigned categories, motivated a more narrow and distinct problem formulation of the classification task, rather than trying to model using the operator-assigned labels. This section describes the dataset constraints, the exploratory analysis process, and a more defined problem definition that is more aligned with the operational objective.

3.2.1 Operational Constraints and Early Detection

The intended objective of the system is to identify accidental emergency calls as early as possible. In practice, this would ideally involve analysis of the audio captured immediately after the call was initiated, before the operator has connected to the caller, which is accessible but not currently stored. The available data therefore only capture audio after the operator has connected to the call. As a consequence, the dataset did not contain the earliest stage of audio from when the caller initiated the call, which may include acoustic cues that are highly relevant for distinguishing

Table 3.1: Table showcasing the different categories within the dataset.

Class	Description	Amount
Silent 112	The operator is unable to establish contact with the caller	1291
Emergency call	Urgent situation that requires prompt assistance from emergency services	6742
Police 114 14	Caller is referred to 114 14 when the need for help is not considered urgent	212
Did not know they had called 112	When the caller themselves does not know they called 112	384
Nuisance call	When the caller intentionally calls to disturb the 112 service	239
Referred to other help	Caller is referred to support outside the 112 organization	205
Other, not 112	Calls with no urgent need for help and no referral to other services	264
	Total	**9337**

accidental calls from real emergencies. This introduces a mismatch between the intended objective and the available training data. In an attempt to approximate early-stage call classification, experiments were designed using only the initial 10 seconds of the available audio.

3.2.2 Limitations of the ”Silent 112” Category

The operator-labeled ”Silent 112” category contains silent emergency calls of heterogeneous acoustic characteristics and varying underlying causes. The primary defining characteristic of these calls is that the operator did not establish any contact with the caller, rather than the presence of a specific acoustic pattern. As a consequence, the category includes calls with diverse audio characteristics, such as near-silent audio, background speech or environmental noise, irregular noise, and subtle acoustic activity.

An initial experiment was therefore conducted where a model was trained using the broad ”Silent 112” category as the positive class and the real emergency category as the negative class. Particular emphasis was placed on evaluating model behavior on challenging negative samples, such as real emergencies that do not contain speech in the initial moment of a call. These experiments served both as a baseline evaluation and as motivation for a more refined reformulation of the target classification task.

3.2.3 Manual Audio Inspection and Definition of Target Class

To better understand the dataset, a manual exploratory inspection of call samples was conducted. The calls were reviewed individually to identify recurring acoustic patterns and to qualitatively assess whether acoustically consistent subcategories could be identified. All categories presented in Table 3.1 were analyzed, with particular emphasis on the ”Silent 112” and ”Did not know they had called 112” categories. The latter category generally contained calls where the caller verbally confirmed that the call had been accidental.

During the inspection process, observations related to background noise, speech presence, environmental sounds, device handling, and other forms of acoustic activity were documented. Particular attention was given to identifying calls that exhibited characteristics consistent with accidental dialing behavior.

The manual inspection process was primarily intended to support qualitative understanding of the dataset and guide reformulation of the classification problem. Based on these observations, a more narrowly defined target class representing irregular noise was constructed. This subcategory was referred to as ”pocket calls”, since the irregular noise by which they are characterized resembles the friction noise of a microphone rubbing against a fabric.

This subcategory of silent emergency calls was constructed using a majority of calls from the ”Silent 112” category together with some confirmed cases of accidental ”pocket calls” from the ”Did not know they had called 112” category. In total, 117 samples with pocket-call characteristics were found in the ”Silent 112” category, and 11 samples were found in the ”Did not know they had called 112” category, resulting in 128 calls total. Among these, 7 calls exhibited distinct irregular noise and were later verbally confirmed by the caller to be an accidental pocket call.

Manual inspection revealed substantial acoustic heterogeneity within the operator assigned ”Silent 112” category. Multiple distinct call types were observed, including nearly silent recordings, calls containing environmental noise, background speech, and calls exhibiting characteristics consistent with accidental pocket dialing behavior. The observed subcategories identified within ”Silent 112” are presented in Table 3.2.

The reason why identifying ”pocket calls” seems feasible lies in the intrinsic characteristics of the sounds emitted during these calls. Pocket calls may generate continuous irregular friction noise patterns caused by movement of the device against the fabric, regardless of whether the operator has answered the call. This may not hold for other silent emergency call types, where the caller may remain silent while waiting for the operator to answer, making these calls more difficult to distinguish acoustically from accidental pocket calls.

These observations motivated the more defined ”pocket call” classification of accidental emergency calls. The exploratory analysis combined qualitative audio record-

Table 3.2: Acoustically observed subcategories identified within the ”Silent 112” category during manual inspection.

Class	Description	Amount
Complete silence	The audio from the caller side is devoid of any sounds	168
Background voices	Speech can be overheard but it’s not directed into the microphone	271
Pocket calls	The audio contains continuous noise for most or all of the call	128
Keypad sounds	Silence is accompanied with occasional keypad button presses	26
Silent call with noises	Broad category containing various sounds that cannot be specified into any of the other “Silent 112” categories	690
Respiratory sounds	A person can be overheard breathing	8

ing inspections with quantitative feature-space analysis, described in section 3.5. The ”pocket call” target class was then also compared against the operator-assigned ”Silent 112” category to evaluate whether it better aligns with the operational objective of detecting accidental silent emergency calls.

3.2.4 Voice Activity Detection Analysis

A point of interest was to inspect the distributions of how long time it takes until a caller in an emergency situation starts to interact with the operator. A simple exploratory speech detection algorithm utilizing Silero VAD was constructed to get a rough estimation of this, which is described here.

Let $\tilde{x}[l]$ denote an individual audio sample at index l, where $l = 0, \ldots, L-1$ and L denotes the total number of samples in the signal. The complete audio signal is $\mathbf{x} = \{\tilde{x}[l]\}_{l=0}^{L-1}$. With sampling frequency $f_s = 16$ kHz, Silero VAD uses a frame length of $W = 512$ samples [58]. The signal is divided into $N = \lfloor L/W \rfloor$ non-overlapping frames of length W. An audio frame at index n is defined as:

$$\mathbf{x}_n = \{\tilde{x}[nW + k]\}_{k=0}^{W-1}, \qquad n = 0, \ldots, N-1. \tag{3.1}$$

Each frame corresponds to:

$$\frac{W}{f_s} = \frac{512}{16000} = 32\,\text{ms}, \tag{3.2}$$

of audio. For each frame, a speech detection rule is applied $s_n = \mathbf{1}(p(\mathbf{x}_n) > \tau)$, where $p(\mathbf{x}_n)$ denotes the speech probability and τ is an adjustable threshold. Furthermore,

once a segment j has been identified, the complete speech segment S_j is ended only when four consecutive frames fall below a negative threshold $\tilde{\tau}$ [58]. That is, an endpoint is registered at frame n if $p(\mathbf{x}_{n+r}) < \tilde{\tau}, r = 0, 1, 2, 3$. The duration of S_j in seconds is

$$d_j = |S_j| \frac{W}{f_s}, \tag{3.3}$$

where $|S_j|$ denotes the number of frames in the segment. The first valid speech segment is defined as $j^* = \min\{j : d_j > 0.15\}$. The onset time for the first detected speech is:

$$t^* = \min(S_{j^*}) \frac{W}{f_s}. \tag{3.4}$$

The minimal duration of 0.15 seconds, which corresponds to at least five audio frames, was arbitrarily set, with in mind that it should capture short utterances. Instead, the main voice detection restriction was implemented by setting the thresholds $\tau = 0.9$ and $\tilde{\tau} = 0.75$.

Though the hyperparameter values haven't been properly evaluated, the algorithm's performance was quickly assessed by listening to a very small subset of calls. For each call, our perception of speech onset time was compared with the the algorithm's output, and it aligned well in all cases. The resulting speech-onset estimates were used to construct delayed-speech emergency call subsets described in the following subsection.

3.3 Audio Preprocessing and Segmentation

All call data were processed at a sampling rate of 16 kHz, which matches the sampling rate expected by the pretrained BEATs model used later for feature extraction. The call data contained separate audio channels for the caller and the emergency operator. Since the objective of the work was to analyze the caller acoustic behavior, only the caller audio channel was retained for further processing. Since the focus of this work was to analyze the earliest portion of each call, only the first 10 seconds of each call were kept, as motivated in subsection 3.2.1. Additional experiments using shorter durations were performed as part of the early-detection analysis presented in section 3.10.

One representation strategy consisted of extracting a single feature vector from the initial audio interval of each call. Calls longer than 10 seconds were truncated to the first 10 seconds, whereas calls shorter than 10 seconds were retained in their entirety. Feature extraction was then performed on the available audio interval. This representation strategy served as the primary baseline representation and was used in several analyses, including comparison between the original "Silent 112" category and the manually defined "pocket call" subset, as well as comparison between the handcrafted features and BEATs embeddings.

In addition to extracting a single feature vector from the initial audio interval, segment-based processing strategies were analyzed in order to enable localized acoustic detection. Rather than representing the entire analyzed audio interval using a single feature vector, the audio was divided into shorter overlapping audio segments prior to feature extraction. Segment-based processing reduces the risk of short duration acoustic events are diluted when feature representations are computed over longer audio intervals. Overlapping windows are standard practice in segment-based audio processing and reduce sensitivity to segment boundaries while increasing the likelihood that short events are fully captured within at least one segment.

An experiment was conducted to analyze the segment window and overlap sizes. The sizes included in this experiment were chosen to cover shorter and longer sizes, and were:

- 1: Segment size: 3.0, overlap 2.0
- 2: Segment size: 5.0, overlap 4.0
- 3: Segment size: 1.0, overlap 0.5

Predictions were initially generated at the segment level. Call-level predictions were obtained by aggregating segment probabilities, as described later in subsection 3.6.4.

3.4 Feature Extraction

In this project, two different feature representations were investigated: handcrafted audio features and pretrained self-supervised audio embeddings. Handcrafted features were primarily used as a baseline feature representation and for the exploratory analysis described in section 3.5, due to their intuitive nature compared to self-supervised embeddings. In addition to representation quality, computational efficiency is an important consideration for real-time emergency call analysis. The computational cost of feature extraction was evaluated for both handcrafted features and self-supervised embeddings of BEATs. The analysis focused on measuring feature extraction latency using multiple unseen 10-second audio samples. The mean and standard deviation extraction times for both handcrafted acoustic features and BEATs embeddings were then measured for 50 unseen samples. This analysis was intended to investigate whether self-supervised representations introduce practical limitations for real-time emergency call analysis.

3.4.1 Handcrafted Acoustic Features

Handcrafted acoustic features were extracted in order to provide interpretable low-level representations of the audio signals. Unlike pretrained self-supervised embeddings, these features describe spectral and temporal characteristics. The selected handcrafted features were chosen to capture a wide range of properties of the call signals, and are described in detail in subsection 2.1.1. The selected features are:

- RMS Energy
- Zero-crossing rate
- Spectral Centroid

- Spectral Roll-off
- Spectral Flatness
- Spectral Bandwidth

Feature extraction was performed either on the entire truncated 10-second calls, or on individual audio segments of the 10 second truncated call, depending on the experiment setup. Audio signals were processed at a sampling rate of 16 kHz using short-time frame-based analysis with a frame length of 25 ms and a hop length of 10 ms.

For an audio signal segmented into T overlapping frames, a six dimensional feature vector $f_t \in \mathbb{R}^6$ was extracted for each frame t. To obtain a fixed-dimensional representation to be used for classification, statistical pooling was applied across the temporal dimension. Specifically, the mean and standard deviation of each feature were computed across all extracted frames:

$$\mu = \frac{1}{T}\sum_{t=1}^{T} f_t, \qquad \sigma = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(f_t - \mu)^2} \tag{3.5}$$

The pooled statistics were then concatenated to form the final handcrafted feature representation $z = [\mu; \sigma]$, where $z \in \mathbb{R}^{12}$. In contrast to the high-dimensional BEATs embeddings, the handcrafted feature representation provided a compact and interpretable description of the acoustic characteristics of the calls. This enabled analysis of whether simple low-level acoustics features were sufficient for distinguishing emergency calls from non-emergency calls.

3.4.2 BEATs Embeddings

In addition to handcrafted features, pretrained self-supervised audio embeddings extracted using the BEATs framework were explored. BEATs is a transformer-based audio representation model trained on large-scale audio corpora using self-supervised learning objectives, as described in subsection 2.1.2. Specifically, the best performing version of BEATs, *BEATs iter3+*, was used as the pretrained frozen encoder to extract embeddings.

Unlike handcrafted audio features, BEATs embeddings provide rich, high-dimensional learned representations capable of capturing complex acoustic structure and contextual audio information. The use of pretrained embeddings was particularly relevant in this project due to the limited number of available positive training samples. In contrast to other popular self-supervised learning audio models, BEATs was specifically designed as a foundational model for general purpose audio tasks, and not just speech-related tasks. This fits the problem of this project, as we seek to analyze audio with absence of speech.

BEATs embeddings were extracted from either the entire 10-second truncated call or from individual audio segments of the 10-second truncated call, depending on the

experimental setup. For an input audio clip of N seconds, the BEATS encoder produced a sequence of T frame-level embeddings $R = \{r_t\}_{t=1}^{T}$, where each embedding $r_t \in \mathbb{R}^{768}$. The number of frame-level embeddings T depended on the duration of the input audio. To obtain a fixed-dimensional representation required for classification, the frame-level embeddings were temporally aggregated using mean and standard deviation pooling across the time dimension T:

$$\mu = \frac{1}{T}\sum_{t=1}^{T} r_t, \qquad \sigma = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(r_t - \mu)^2} \tag{3.6}$$

The resulting pooled statistics were then concatenated to form the final feature vector $z = [\mu; \sigma]$, where $z \in \mathbb{R}^{768}$. This produced a fixed 1536-dimensional representation for each audio segment or call, regardless of the original audio duration.

3.5 Exploratory Feature-Space Analysis

Using the handcrafted features presented in subsection 3.4.1, an exploratory analysis of the feature distributions was performed in order to investigate the degree of separability between different call categories. The purpose of this analysis was to evaluate whether different call types exhibited distinguishable characteristics in the handcrafted feature space.

Particular emphasis was placed on comparisons involving delayed-speech emergency calls, since these represent operationally challenging scenarios in which genuine emergency calls may initially resemble accidental silent calls. The exploratory analysis focused on the comparisons summarized in Table 3.3. The subcategories "Delayed-speech emergency calls (5 s)" and "Delayed-speech emergency calls (10 s)" are shortened to "Delayed-5s" and "Delayed-10s", respectively.

Table 3.3: The different positive and negative classes considered.

Positive Class	Negative Class
Silent 112	Delayed-5s
Silent 112	Delayed-10s
Pocket Calls	Delayed-5s
Pocket Calls	Delayed-10s

3.5.1 Distribution Overlap Analysis

To measure the degree of similarity between feature distribution across different call categories, a distribution overlap analysis was performed for each handcrafted feature independently. Distribution overlap provides a measure of how much two distributions overlap and can therefore be used as an indicator of feature separability. The overlapping coefficient [59] is defined as

$$\Delta(f_1, f_2) = \int \min\left[f_1(x),\ f_2(x)\right]\, dx \tag{3.7}$$

where $f_1(x)$ and $f_2(x)$ denote the estimated probability density functions of two feature distributions. The coefficient measures the shared area between the two distributions and takes values in the range [0, 1]. A value of 0 indicates complete separation, whereas a value of 1 indicates identical distributions. Therefore, lower overlap values suggest that the feature provides greater separation between the two call categories.

3.6 Classification Models

In this project, classification experiments were conducted using two supervised learning approaches: gradient boosted decision trees implemented through XGBoost and logistic regression. Both classifiers were evaluated under different audio representation and prediction frameworks in order to analyze how they affected classification behavior.

The objective of the experiments was not only to evaluate classifier performance, but also to investigate how different audio representation strategies affected robustness and early detection capability.

All classifiers trained in these experiments are probabilistic in nature and produce probability estimates for the positive class, that is, an involuntary emergency call. Depending on the experiment configuration, predictions were generated either from a single feature representation extracted from the initial 10 second audio interval, as described in section 3.3, or from multiple segment representations of the initial 10 second interval that were then aggregated into a call-level prediction.

The segment-based approach was motivated by several considerations. Firstly, calls may contain short localized acoustic events that risk being diluted when feature representations are extracted across longer audio intervals. By operating on shorter overlapping windows, a segment-level approach should increase the likelihood that informative acoustic patterns are preserved within at least one segment. Additionally, the segment-based approach provides multiple segment-level predictions within the analyzed audio interval. This enables analysis of how model confidence evolves as additional portions of the audio interval become available.

The two representation strategies differ primarily in how the analyzed audio interval is represented. The single-representation approach summarizes the entire 10-second interval using a single feature vector, whereas the segment-based approach represents the same interval using multiple overlapping feature vectors that are subsequently aggregated into a call-level prediction.

3.6.1 XGBoost

XGBoost was selected as the primary nonlinear classification model due to its strong performance on tabular data and its robustness in settings with limited training

data. This was considered very relevant for this project given the limited number of positive training samples available.

For all experiments, the classifier was trained as a binary probabilistic model using the logistic objective function with the negative log-likelihood evaluation metric. Class imbalance was handled using the *scale_pos_weight* parameter, computed as the ratio between negative and positive training samples within each experiment:

$$\text{scale\_pos\_weight} = \frac{N_{\text{negative}}}{N_{\text{positive}}} \tag{3.8}$$

This form of class weighting only scales the positive class, heavily penalizing misclassifications of the positive involuntary "pocket call" class.

To reduce overfitting and improve generalization, the row subsampling parameter and the feature subsampling parameter were both fixed to 0.8 for all experiments. To avoid an excessively large parameter search, only the learning rate, maximum depth, and number of estimators were tuned, as described later in section 3.7.

3.6.2 Logistic Regression

In addition to XGBoost, experiments were also conducted using regularized logistic regression, through the Scikit-Learn library. Logistic regression was included as an alternative linear model due to its simplicity, interpretability, and strong performance on high-dimensional embedding representations, where semantic and contextual structure is often linearly separable. Unlike the tree-based model, logistic regression constructs a linear decision boundary in the feature space. This provides a useful comparison against the non-linear modelling of XGBoost.

The classifier was trained using L2 regularization, and class imbalance was handled through the class weighting parameter, which assigns weights inversely proportional to class frequencies:

$$w_c = \frac{n_{\text{samples}}}{n_{\text{classes}} \cdot n_c} \tag{3.9}$$

where n_c denotes the number of samples belonging to class c. Since all experiments considered a binary classification task ($n_{\text{classes}} = 2$), the minority class received a higher weight, causing misclassification of positive samples to be penalized more heavily. The optimization algorithm used was the LBFGS solver, and only the regularization strength parameter was tuned to avoid excessive hyperparameter search.

3.6.3 Single-Representation Classification

In the single-representation classification approach, each call was represented using a single feature representation extracted from the initial audio interval. Unless stated otherwise, the first 10 seconds of caller audio were used as the input window.

Feature representations were generated using the extraction procedures described in section 3.4. The resulting feature vector was then used as input to the classifier, producing a single probability estimate for each call sample.

This approach provides a simple, global representation of the analyzed audio interval and acts as a baseline against which the segment-based approach can be compared. This approach preserves the complete temporal context of the analyzed interval within a single feature representation.

3.6.4 Segment-Based Classification

For the segment-based approach, rather than representing the analyzed audio interval using a single feature representation, the audio was divided into multiple shorter overlapping segments prior to feature extraction, as described in section 3.3. Feature representations were then extracted independently for each segment using the procedures described in section 3.4. The classifier then produced segment-level probabilistic predictions.

Since segment-level classification produces multiple predictions for each call sample, aggregation strategies were required in order to obtain call-level predictions. Two strategies were analyzed. The first required all segment probabilities to exceed a predefined threshold before a call was classified as positive. The second aggregated segment probabilities by averaging them to obtain a single call-level probability estimate.

3.7 Training and Validation

Model development and experimental validation were primarily conducted using cross-validation on the training data. A held-out test set was reserved for final evaluation of the selected model configuration in order to reduce the risk of overfitting experimental decisions to the evaluation data.

3.7.1 Train-Test Splits

In this project, two experimental task formulations were investigated. Initial experiments used the original operator-assigned ”Silent 112” category as the positive class, while later experiments used the manually identified ”pocket call” subset introduced in section 3.2. The ”Emergency Call” category, as presented in Table 3.1, was used as the negative class for both setups.

Since the two formulations used different positive-class definitions and dataset compositions, separate train/test splits were constructed for each setup. Approximately 80% of the samples were assigned to the training set and 20% to the held-out test set using stratified sampling based on the corresponding file-level labels. Shuffling was enabled prior to splitting, and a fixed random seed of 1 was used for reproducibility. This split was selected to maximize the training data while preserving

an independent hold-out test set for final evaluation. The resulting train- and test split sizes are presented in Table 3.4.

Table 3.4: The two dataset configurations and their corresponding dataset splits.

Configuration	Train Positive	Train Negative	Test Positive	Test Negative
Silent 112	1033	5394	258	1348
Pocket Call	102	5394	26	1348

Once constructed, the held-out test sets remained fixed throughout the following experiments and were not repeatedly used during exploratory experimentation, feature scaling, feature selection analysis, negative-class composition experiments or threshold tuning.

3.7.2 Cross-Validation Strategy

Cross-validation was used throughout the experiments in order to obtain robust performance estimates, which was especially relevant under the limited data conditions. For experiments based on full-call representations, a stratified k-fold cross-validation was applied to preserve class distributions across folds.

For segment-based experiments, stratified *group* k-fold cross validation was used to ensure that all segments originating from the same call recording was assigned to the same fold. This prevented data leakage between training and validation data caused by correlated segments from the same call samples appearing in multiple folds.

All experiments used five cross-validation folds with shuffling enabled. Out-of-fold predictions were generated during cross-validation such that each training sample received a prediction from a model that had not been trained on that sample. This enabled unbiased analysis of model behavior on all training data, which was important under the limited data conditions.

The resulting out-of-fold predictions were used for confusion matrix analysis, feature representation comparison, segmentation experiments, threshold optimization, and analysis of false positives occurring within the ”Delayed-speech real emergency (5 s)” and ”Delayed-speech real emergency (10 s)” subcategories of real emergency calls, introduced in section 3.8.

3.7.3 Model and Threshold Tuning

The model hyperparameters were optimized using out-of-fold predictions generated from the training data. A limited grid-search strategy was used to limit complexity, reduce the risk of overfitting under the limited data conditions, and computational constraints. Logistic regression experiments varied the regularization parameter C, while XGBoost experiments varied tree depth and number of estimators. Model

configurations were primarily ranked using PR-AUC, since this metric is threshold-independent and suitable for imbalanced binary classification. The hyperparameters included in the grid-search is presented in Table 3.5.

Table 3.5: Table showcasing the model hyperparameter search.

Model	Hyperparameter	Tested Values
Logistic Regression	Regularization strength, C	0.1, 1.0, 10.0
XGBoost	Maximum depth	4, 6
XGBoost	Learning rate	0.05
XGBoost	Number of estimators	100, 200, 300

After selecting the best performing model configurations, probability thresholds were optimized separately using threshold sweeps on the out-of-fold predictions. This enabled analysis of the tradeoff between recall and false positive behavior across different thresholds. Decision thresholds between 0.1 and 0.9 were analyzed in increments of 0.05 using the out-of-fold predictions generated from the training data.

3.8 Negative Class Composition

The negative class consisted of calls from the ”Emergency call” category. However, real emergency calls are in most cases easy to distinguish from silent calls, as they often contain speech early on from when the operator connected to the call. Therefore, emphasis of the negative class in the training set was placed on calls with limited speech activity during the initial portions of the call, since these represent challenging cases that may acoustically resemble ”Silent 112” calls.

To identify such calls, the results from subsection 3.2.4 were used. Based on the speech detection results, two subsets of real emergency calls were constructed:

- **Delayed-speech emergency calls (5 s)**: Real emergency calls without detected speech during the first five seconds.
- **Delayed-speech emergency calls (10 s)**: Real emergency calls without detected speech during the first ten seconds.

In total, 364 ”Delayed-speech emergency calls (5 s)” and 45 ”Delayed-speech emergency calls (10 s)” were found. The substantial reduction in sample count between the 5 second and 10 second subsets indicates that most emergency callers began speaking within the first ten seconds of the call. After the train-test split described in section 3.7, the number of samples in the training set was 295 for ”Delayed-speech emergency calls (5 s)” and 34 for ”Delayed-speech emergency calls (10 s)”. Experiments were conducted using varying proportions of ordinary emergency calls

within the negative training class in order to investigate how class construction affected model behavior. In particular, a varying number of real emergency calls were combined with the "Delay-speech real emergency (5 s)" and "Delay-speech real emergency (10 s)" subcategories of real emergency calls. The experimental setup is presented in Table 3.6.

Table 3.6: Experimental setup of negative class composition within the training set. "Delayed-speech emergency calls (N s)" is shortened to Delayed-Ns.

Composition	Ordinary emergency call	Delayed-5s	Delayed-10s	Total
Full	5063	294	37	5394
Large	2500	294	37	2831
Medium	1000	294	37	1331
Min	400	294	37	731
Tiny	100	294	37	431
None	0	294	37	331

The experiments were intended to analyze how the inclusion of easy negatives in the training data influenced the behavior and generalization of the model.

3.9 Evaluation Setup

The final evaluation was conducted using the best performing logistic regression and XGBoost model configurations identified during the cross-validation experiments on the training data.

All final experiments were evaluated on the test set that had remained fixed throughout the model development. This ensured that the performance metrics reported reflected generalization to previously unseen data. Evaluation focused on metrics relevant to the imbalanced classification setting and the operational requirements of the task. These included recall, precision, F1-score, PR-AUC, false-positive rate (FPR), and false-negative rate (FNR). In addition to the overall false-positive rate, separate analysis was performed on the subcategories "Delayed-speech real emergency (5 s)" and "Delayed-speech real emergency (10 s)" of real emergency calls, which were introduced in section 3.8, since false positives within these categories represent operationally important cases. For the "Delayed-speech real emergency (5 s)" case, the available audio is limited to five seconds such that any voice is not included.

To measure statistical uncertainty, 95% confidence intervals were computed for selected performance metrics. Wilson score intervals were used strictly for proportion based metrics such as recall and false positive rate, while bootstrap confidence intervals using 2000 resampling iterations with replacement were used for all evaluation metrics including PR-AUC. The bootstrap procedure used a stratified resampling approach to preserve class proportions across iterations.

3.10 Early Detection Experiments

In practical emergency call scenarios, it is desirable to identify accidental emergency calls as soon as possible to reduce operator workload. Therefore, an additional set of experiments was carried out to evaluate how classification performance changed when additional call data progressively became available.

Early detection performance was evaluated by truncating the available audio from the beginning of each call. Separate experiments were conducted using the first 2, 4, 6, 8 and 10 seconds of audio. For each duration, the same preprocessing pipeline and feature extraction procedures described earlier in this chapter were used. Following the setup in section 3.9, the experiments utilized the best-performing configurations of the logistic regression and XGBoost models, evaluating performance on the test set with the same metrics.

4
Results and Discussion

This chapter presents the experimental results obtained throughout the project. The analysis begins with experiments based on the original operator-assigned ”Silent 112” category in order to investigate the limitations of modeling on the initial labels. The results then motivate a more defined target class based on manually identified ”pocket calls”.

The following subsections analyze feature space characteristics, compare handcrafted audio features against pretrained BEATs embeddings, investigate the influence of negative class training composition, and compare single-window and segment-based classification strategies. The chapter concludes with early detection experiments on tuned models and final evaluations on the held-out test set.

4.1 Initial Experiments Using Original Operator Labels

Initial experiments were conducted using the original operator-assigned ”Silent 112” category as the positive class. The purpose of these experiments was to serve as an exploratory baseline for understanding the limitations of the original label and to motivate a more defined target class. For these experiments, both logistic regression and XGBoost were evaluated using the single-representation classification approach presented in subsection 3.6.3, with handcrafted acoustic features and BEATs embeddings. The precision, recall and F1 scores correspond to the positive ”Silent 112” class. These results are presented in Table 4.1.

Table 4.1: Classification performance metrics of model trained on the broad ”Silent 112” category.

Model	Features	F1	Precision	Recall	PR-AUC
LR	Handcrafted	0.58	0.47	0.77	0.71
LR	BEATs	0.86	0.85	0.87	0.91
XGBoost	Handcrafted	0.76	0.80	0.73	0.83
XGBoost	BEATs	0.89	0.89	0.89	0.94

Although these classification metrics in themselves look promising, inspecting the performance on hard negatives revealed the main consequences of this approach, as seen in Table 4.2.

Table 4.2: Classification performance on operationally relevant hard negatives.

Model	Features	FP rate	FN rate	FP Delayed-5s	FP Delayed-10s
LR	Handcrafted	0.17	0.23	0.41	**0.92**
LR	BEATs	0.027	0.13	0.078	**0.73**
XGBoost	Handcrafted	0.035	0.27	0.15	**0.78**
XGBoost	BEATs	0.022	0.11	0.078	**0.92**

4.2 Exploratory Analysis

Although the initial experiments using the original ”Silent 112” category produced strong overall classification performance, further analysis revealed that the operational objective was not effectively achieved. In particular, several real emergency calls with limited early speech activity were incorrectly classified as accidental silent calls, indicating substantial acoustic heterogeneity within the original target category.

To better understand these limitations, an exploratory analysis was conducted to investigate whether a more operationally relevant and acoustically coherent target formulation could be identified. The analysis combined qualitative manual inspection of call recordings with quantitative feature-space analysis.

4.2.1 Caller Speech Detection

Following the algorithm as described in subsection 3.2.4, an approximate distribution of the time it taken until contact was established between the operator and person in need of emergency assistance was constructed. The result is shown in Figure 4.1. Evidently, it is not uncommon for a help seeker to begin speaking only after some time has passed. While there is a sharp decline thereafter, a non-negligible amount of calls have considerable delays. This may further justify the narrowed down scope of the project and the difficulties of distinguishing silent calls from emergency calls.

4.2.2 Handcrafted Feature Distribution Analysis

The handcrafted feature distribution analysis was conducted to investigate whether the ”pocket call” target class exhibited greater acoustic separability from delayed-speech emergency calls than the broader operator-assigned ”Silent 112” category. For each handcrafted feature, the overlap between feature distributions was estimated for both ”Delayed-speech emergency call (5 s)” and ”Delayed-speech emergency call (10 s)” subsets. Lower overlap values indicate greater estimated separability between the compared distributions.

Table 4.3 presents the estimated overlapping coefficients for the ”Delayed-5s” comparisons. In general, the ”pocket call” category exhibited consistently lower overlap

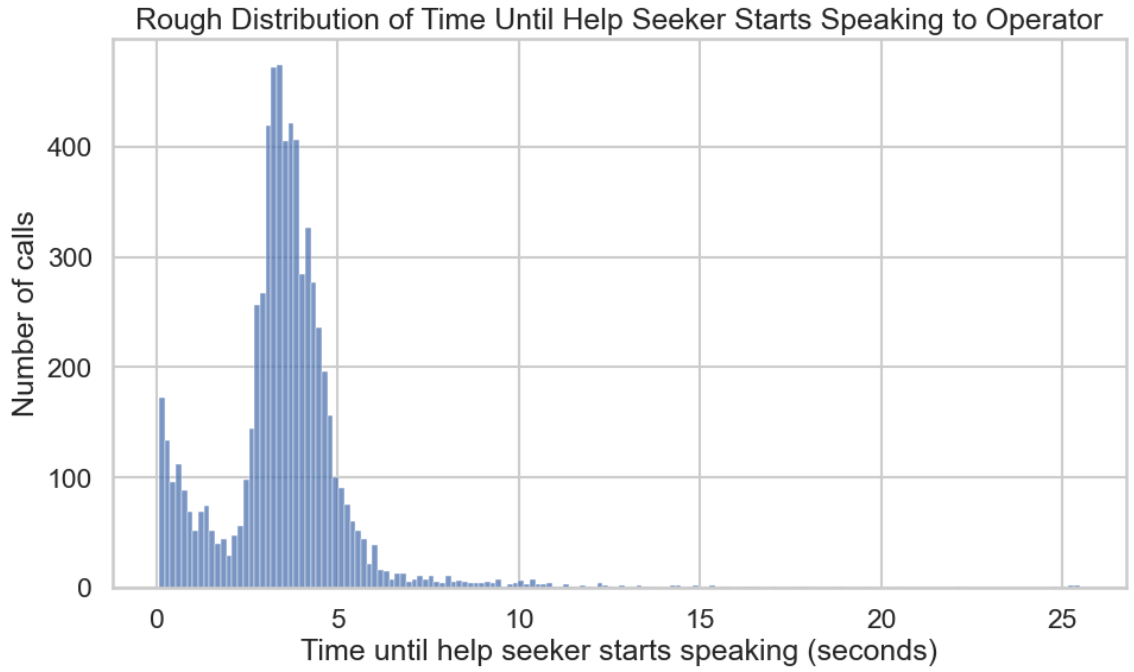


Figure 4.1: Distribution showing time elapsed before help seeker begins, showing that most begin speaking within approximately five seconds, with a small trailing tail of longer delays.

values across all handcrafted features, indicating greater estimated acoustic separability from delayed-speech emergency calls than the ”Silent 112” category.

Table 4.3: Delayed-5s overlap comparison.

Feature	Silent 112 Overlap	Pocket Call Overlap
Spectral Roll-off (mean)	0.75	0.52
Spectral Bandwidth (mean)	0.76	0.53
Spectral Roll-off (std)	0.79	0.57
Spectral Centroid (mean)	0.79	0.58
ZCR (mean)	0.80	0.61
Spectral Bandwidth (std)	0.76	0.65
RMS (mean)	0.89	0.67
RMS (std)	0.80	0.67
Spectral Centroid (std)	0.84	0.68
ZCR (std)	0.85	0.69
Spectral Flatness (std)	0.98	0.97
Spectral Flatness (mean)	0.99	0.98

The largest reductions in overlap were observed for spectral domain features. In particular, the overlap for spectral roll-off (mean) decreased from 0.75 for ”Silent 112” to 0.52 for ”pocket calls”, while spectral bandwidth (mean) decreased from 0.76 to 0.53. Although the distributions are still not cleanly separable, the reduced overlap

suggests that ”pocket calls” exhibit more distinct spectral characteristics with improved separability from challenging real emergency calls with delayed speech than the broader ”Silent 112” category.

Figure 4.2 illustrates the selected feature distributions corresponding to the features with the lowest overlap values. The top row presents comparisons between the ”Silent 112” category and ”Delayed-speech emergency calls (5 s)”, while the bottom row presents the corresponding distributions for the ”pocket Call” category. The x-axis corresponds to the feature value, measured in Hertz for both spectral bandwidth and spectral roll-off, while the y-axis shows the estimated density.

The results indicate that the manually constructed ”pocket call” class may have more separable acoustic characteristics than the broad ”Silent 112” category. This supports the hypothesis that narrowing the target class towards a more defined category of calls containing irregular friction-like noise improves the acoustic separability between accidental silent emergency calls and real emergency calls with delayed speech.

Additional overlap analysis using the ”Delayed-speech emergency calls (10 s)” subset is provided in Appendix A. Similar tendencies were observed, but the smaller sample size of the subset compromises the stability and reliability of the overlap estimates.

4.3 Pocket Call Classification Experiments

Following the exploratory analysis and formulation of a more defined target class, classification experiments were conducted using the single-representation classification of the initial 10 seconds of each call. These experiments evaluated both handcrafted acoustic features and pretrained BEATs embeddings in order to analyze how feature representation influenced classification performance. These experiments also acted as a direct comparison against the broad original ”Silent 112” label in order to investigate whether the more defined ”pocket call” category improved robustness, especially against false positives within real emergency calls with limited speech. Both logistic regression and XGBoost classifiers were evaluated.

Table 4.4: Single-window classification performance metrics.

Model	Features	F1	Precision	Recall	PR-AUC
LR	Handcrafted	0.31	0.19	0.93	0.39
LR	BEATs	0.93	0.92	0.95	0.98
XGBoost	Handcrafted	0.53	0.52	0.54	0.56
XGBoost	BEATs	0.89	0.89	0.88	0.97

Table 4.4 shows the single-window classification performance for the evaluated feature representations and classifiers. Overall, models trained using pretrained BEATs

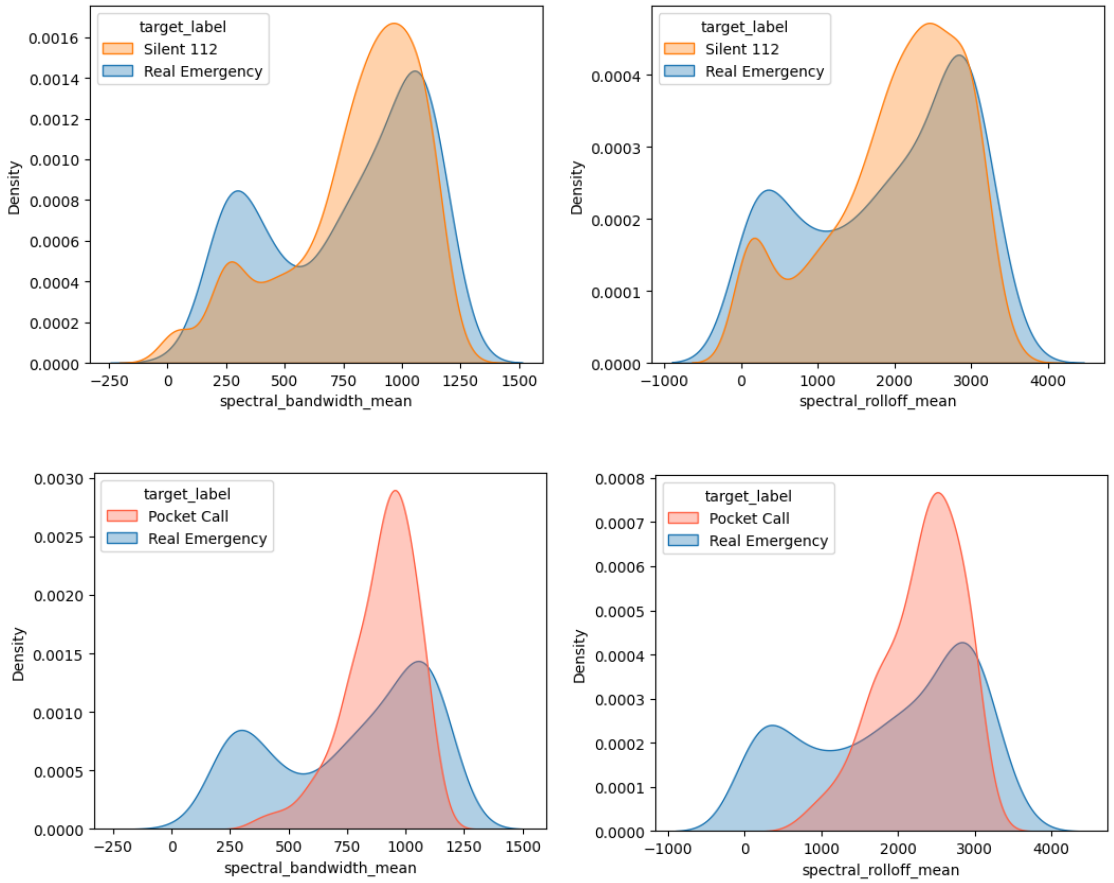


Figure 4.2: Kernel density estimates of selected handcrafted feature distributions for comparisons against Delayed-speech emergency calls (5 s). Top row: operator-assigned Silent 112 category versus delayed-speech emergency calls. Bottom row: "pocket call" category versus delayed-speech emergency calls.

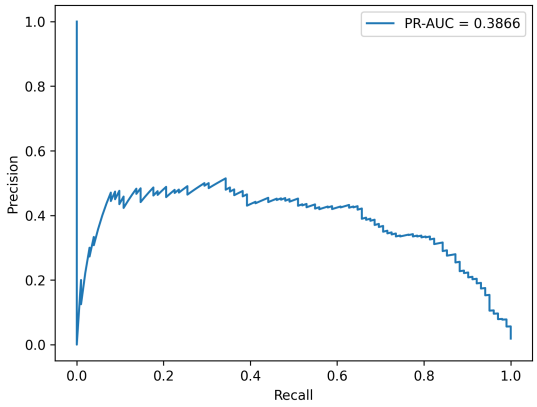
embeddings substantially outperformed models trained on handcrafted acoustic features. The largest improvement was observed for logistic regression, where the F1-score increased from 0.31 using handcrafted features to 0.93 using BEATs embeddings. Similar improvements were observed in precision, PR-AUC, and recall.

Interestingly, logistic regression achieved comparable or better performance than XGBoost when using BEATs embeddings. This suggests that the pretrained embedding representation already provides a high degree of linear separability between pocket calls and real emergency calls. In addition to the performance metrics presented in Table 4.4, operationally relevant metrics are also presented in Table 4.5. Although overall false positive rates were low for all feature representations and classifiers, substantially higher false positive rates were observed for operationally relevant delayer-speech emergency call subsets. In particular, the false positive rate for delayed-speech emergency calls without speech during the first 10 seconds remained notably higher than the overall false positive rate, especially for the handcrafted feature representations.

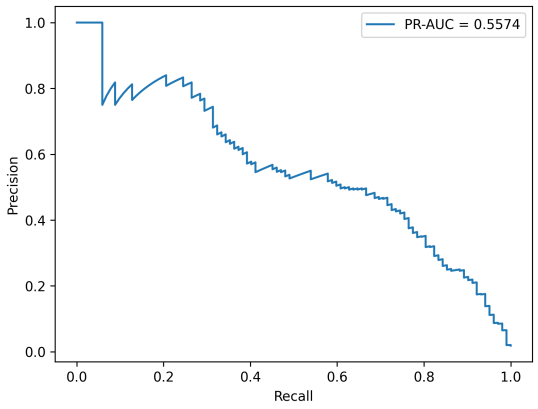
Table 4.5: Single-representation classification performance on operationally relevant hard negatives.

Model	Features	FP rate	FN rate	FP Delayed-5s	FP Delayed-10s
LR	Handcrafted	0.078	0.069	0.25	**0.89**
LR	BEATs	0.0017	0.049	0.0068	**0.081**
XGBoost	Handcrafted	0.0093	0.46	0.024	**0.24**
XGBoost	BEATs	0.0020	0.1176	0.0068	**0.11**

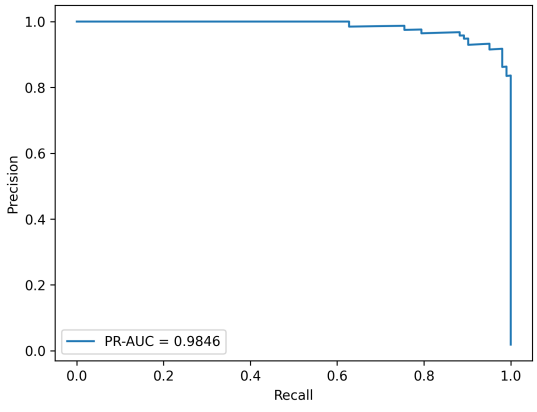
The results supports the hypothesis that narrowing the target class towards acoustically characteristic "pocket calls" improves robustness against delayed-speech emergency calls, especially when pretrained BEATs embeddings are used as feature representations. The precision-recall curves shown in Figure 4.3 highlights the performance difference between handcrafted features and pretrained BEATs embeddings.



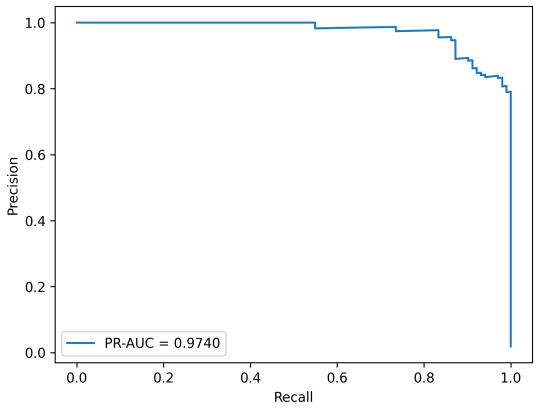
(a) Logistic regression with handcrafted features.



(b) XGBoost with handcrafted features.



(c) Logistic regression with BEATs embeddings



(d) XGBoost with BEATs embeddings.

Figure 4.3: Precision-recall curves corresponding to the model and feature configurations presented in Table 4.4.

4.4 Negative-Class Composition Experiments

While the single-window experiments demonstrated promising classification performance, many negative samples corresponded to ordinary emergency calls containing clear speech activity and were therefore comparatively easy to separate from accidental silent calls. Additional experiments were therefore conducted to investigate how the composition of the negative class influenced model behavior under more operationally challenging conditions.

The results demonstrate a tradeoff between overall false positive reduction and preservation of accidental call detection performance, as presented in Figure 4.4. Increasing the number of ordinary voiced emergency calls in the negative training class consistently reduced the overall false positive rate. However, excessive inclusion of such easy negative samples increased the false negative rate on accidental calls and did not significantly improve performance on the operationally challenging delayed-speech emergency call subsets. Exact metric values corresponding to Figure 4.4 are presented in Table 4.6.

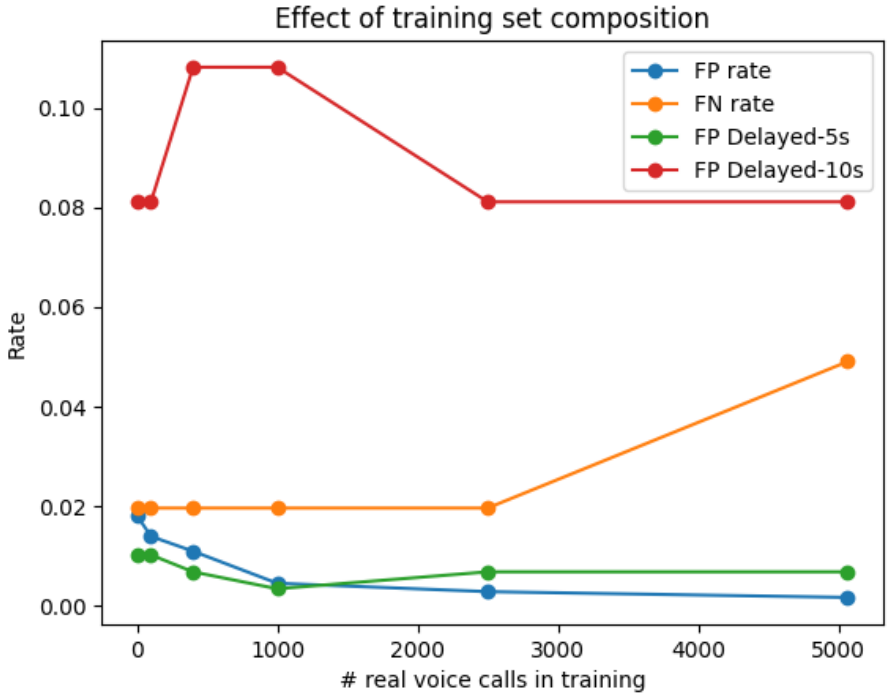


Figure 4.4: Effect of ordinary voiced emergency call inclusion in the negative training class on classification performance.

The configuration using 2500 ordinary voiced emergency calls was selected for subsequent experiments, as it provided a balance between low overall false positive rates and preserved robustness on delayed speech emergency calls while not increasing false negative rates.

These experiments suggests that negative class composition to some extent influences model behavior for accidental call detection. In particular, the results indicates that training with large numbers of acoustically easy emergency calls may encourage the classifier to more strongly rely on speech presence rather than learning the characteristics specific to accidental pocket calls. It is worth noting that the differences are quite small, with most metrics only varying by a few percentage points.

Table 4.6: Classification performance in regards to the number of ordinary voiced emergency calls included in the negative training class.

No. voiced real calls	FN rate	FP rate	FP Delayed-5s	FP Delayed-10s
0	0.020	0.018	0.010	0.081
100	0.020	0.014	0.010	0.081
400	0.020	0.011	0.0068	0.11
1000	0.020	0.0045	0.0034	0.11
2500	0.020	0.0028	0.0068	0.081
5066	0.049	0.0068	0.0068	0.081

4.5 Segment-Based Classification and Aggregation

Since the single-representation classification approach compressed the entire audio interval into a single fixed-dimensional feature vector, short-duration acoustic events may be diluted during temporal pooling. Segment-based classification was therefore investigated in order to preserve localized temporal information and analyze whether this approach could improve performance of accidental call detection. This is done as described in subsection 3.6.4, where strict aggregation classified a call as positive only if all segment predictions exceeded the decision threshold, whereas mean aggregation used the average segment probability across the call.

All experiments in this section used a fixed decision threshold of 0.5. Threshold optimization was investigated separately in section 4.6.

The results in Table 4.7 indicate that mean aggregation consistently outperformed strict aggregation across both logistic regression and XGBoost models. The difference is clear in recall, where strict aggregation produced substantially lower rates for all segment durations. This suggests that strict aggregation was sensitive to prediction variability within segments. Since a single segment of low confidence could strongly influence the final call-level decision, the resulting predictions became highly conservative with limited practical utility for accidental call detection. In contrast, mean aggregation resulted in considerably improved recall while maintaining low false positive rates.

To evaluate whether segmentation itself provided benefits beyond temporal localization, the strongest segmented configurations were compared against the corre-

Table 4.7: Segment-based classification performance for varying segment durations and aggregation strategies.

Representation	Model	Aggregation	F1	Recall	FP Rate
Short	LR	Strict	0.13	0.07	0.00
Short	LR	Mean	0.79	0.67	0.00070
Medium	LR	Strict	0.41	0.25	0.00
Medium	LR	Mean	0.81	0.70	0.0011
Long	LR	Strict	0.78	0.65	0.00040
Long	LR	Mean	0.89	0.86	0.0025
Short	XGBoost	Strict	0.08	0.04	0.00
Short	XGBoost	Mean	0.66	0.49	0.00
Medium	XGBoost	Strict	0.48	0.31	0.00
Medium	XGBoost	Mean	0.87	0.78	0.00070
Long	XGBoost	Strict	0.81	0.68	0.00
Long	XGBoost	Mean	0.92	0.90	0.0025

sponding single window representations. Table 4.8 compares the strongest aggregation configurations with the single-window approach. The single-window approach slightly out-edges the segmentation approach in performance for both logistic regression and XGBoost. This may indicate that the pretrained BEATs embeddings already capture sufficient information for this task, reducing the benefit of temporal segmentation.

Table 4.8: Comparison between full-call and segmented representations using the strongest configurations.

Representation	Model	Aggregation	F1	Recall	FP Rate Delayed-10s
Full	LR	-	0.95	0.98	0.081
Long	LR	Mean	0.96	0.96	0.0055
Full	XGB	-	0.94	0.94	0.081
Long	XGB	Mean	0.92	0.89	0.0055

Although segment-based classification did not consistently outperform the full 10 second single-representation approach, the experiments provided useful insights into how aggregation strategies influence classification behavior. In particular, the results demonstrated that conservative aggregation strategies may reduce recall while producing fewer false positives, while mean aggregation preserved detection performance while keeping false positive rates low.

Despite the potential advantages of localized analysis provided by the segmentation classification approach, the full 10 second single-representation approach achieved the strongest overall performance across both logistic regression and XGBoost. However, all representation approaches are carried forward into the subsequent hyper-

parameter optimization and threshold tuning phases to analyze if such tuning alters the outcome.

4.6 Hyperparameter and Threshold Optimization

Following the representation and aggregation experiments, hyperparameter optimization and threshold tuning experiments were conducted using BEATs embeddings for logistic regression and XGBoost. The objective was to investigate whether classifier specific hyperparameter tuning and threshold selection could further improve operational performance.

For logistic regression, the regularization parameter C was varied across multiple configurations. For XGBoost, experiments investigated combinations of maximum tree depth and number of estimators. Performance was evaluated using cross-validated F1-score, recall, PR-AUC, and false positive rates. Table 4.9 summarizes the strongest performing and most operationally informative hyperparameter configurations across both classifiers.

The hyperparameter experiments produced relatively small performance differences across configurations. For logistic regression, the strongest performing configurations were consistently obtained using full 10 second single window representations with BEATs embeddings. The logistic regression configuration with $C = 10.0$ achieved an F1-score of approximately 0.94 with a recall of 0.99 while maintaining low false positive rates. Interestingly, the long segment logistic regression configuration with $C = 0.1$ achieved an F1 score of 0.93 while slightly reducing false positive rates compared to the strongest single window configuration. However, short- and medium segment representations exhibited noticeably reduced recall.

For XGBoost, the strongest configurations were again obtained using full 10 second single window representations with BEATs embeddings. The best performing XGBoost models achieved F1-scores of approximately 0.95 with recall around 0.94. Although several XGBoost configurations performed well on false positive rates, and even achieved slightly higher F1-scores than logistic regression, these models also produced lower recall.

While some configurations achieved marginally lower false positive rates, model selection was ultimately guided by operational considerations. With this in mind, reducing false positives is important to avoid incorrectly flagging genuine emergency calls as accidental calls. However, the observed reductions in false positive rates were generally small and often accompanied by a more noticeable reduction in recall. Given the operational cost of missing true pocket calls, configurations with higher recall were preferred when the increase in false positive rate was limited. Based on this trade-off, the final selected configurations were logistic regression with $C = 10.0$ and XGBoost with a maximum tree depth of 6 and 200 estimators, both using the full 10-second single-window BEATs representation.

Table 4.9: Table showcasing the model hyperparameter search.

Model	Representation	Hyperparameter	F1	Recall	FPR	FPR Delayed-10s
Logistic Regression	Single window	$C = 10.0$	0.94	0.99	0.0039	0.11
Logistic Regression	Single window	$C = 1.0$	0.94	1.00	0.0046	0.11
Logistic Regression	Long segments	$C = 0.1$	0.93	0.94	0.0029	0.054
Logistic Regression	Single window	$C = 0.1$	0.87	1.00	0.011	0.32
XGBoost	Single window	Max. depth = 6, No. estimators = 200	0.95	0.94	0.059	0.054
XGBoost	Single window	Max. depth = 6, No. estimators = 300	0.95	0.94	0.059	0.054
XGBoost	Long segments	Max. depth = 6, No. estimators = 200	0.93	0.92	0.0042	0.16
XGBoost	Single window	Max. depth = 4, No. estimators = 100	0.90	0.89	0.0024	0.054

To further analyze classifier behavior and decision threshold, precision recall curves were generated for the top performing model configurations ranked by PR-AUC.

The precision-recall curves presented in Figure 4.5 show that logistic regression using full 10 second single window representations outperformed segmented approaches across most decision thresholds. The top three single window configurations achieved PR-AUC values above 0.99, indicating strong separability between ”pocket calls” and ordinary emergency calls. In contrast, segmented representations showed earlier precision degradation at high recall levels.

The XGBoost precision-recall curve presented in Figure 4.6 exhibited a similar overall behavior. Single window representations again achieved the highest PR-AUC

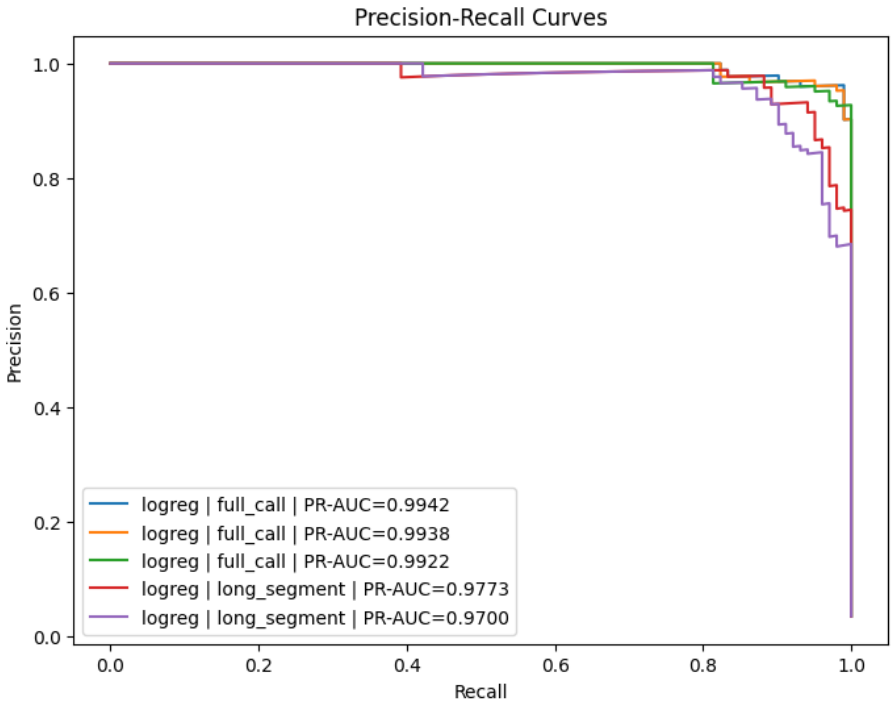


Figure 4.5: Precision-recall curves for the top performing logistic regression configurations ranked by PR-AUC.

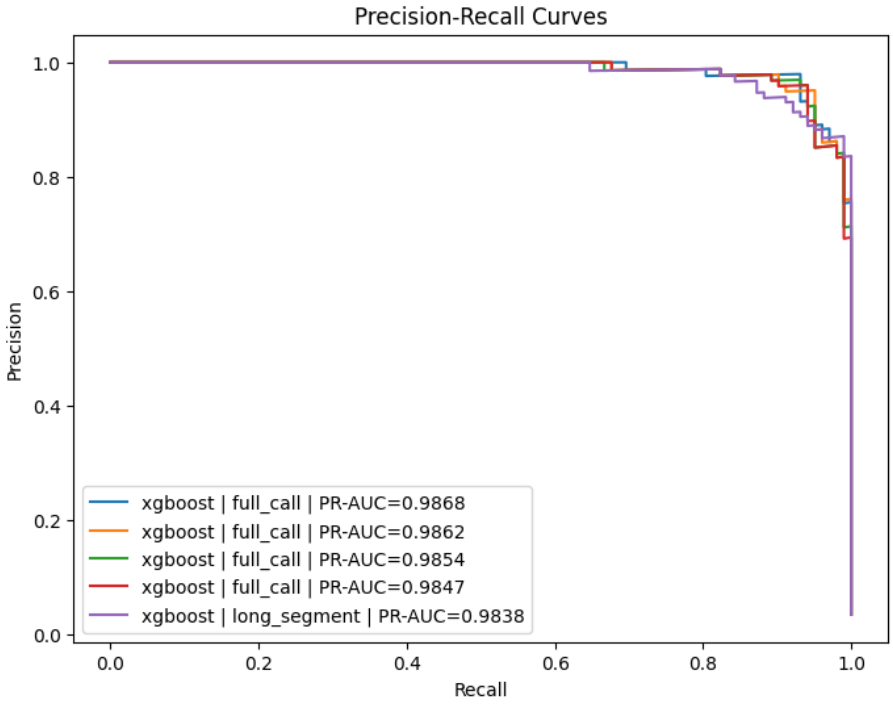
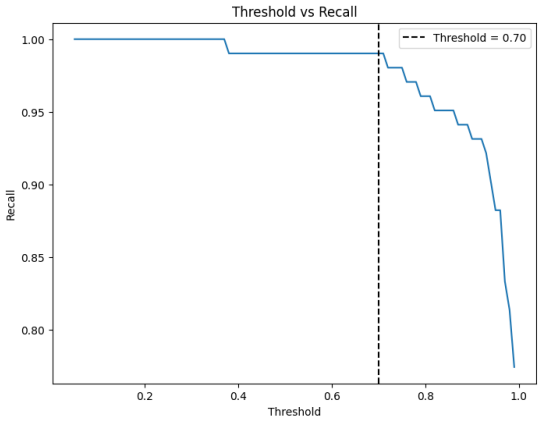


Figure 4.6: Precision-recall curves for the top performing XGBoost configurations ranked by PR-AUC.

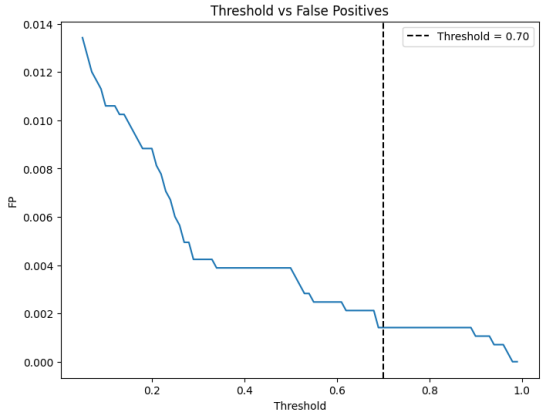
values, while segmented representations produced slightly earlier precision degrada-

tion at high recall values.

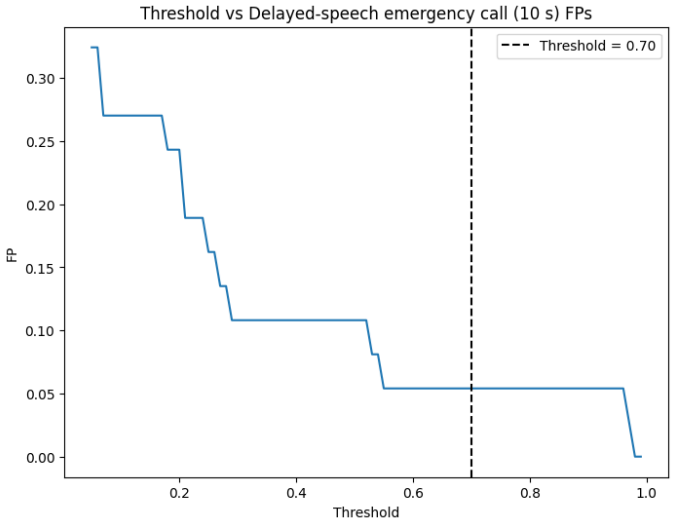
Threshold sweeps were conducted on the strongest performing logistic regression and XGBoost hyperparameter configurations to investigate trade-offs between recall and false positive rates.



(a) Threshold vs Recall.



(b) Threshold vs FPR.



(c) Threshold vs FPR Delayed-10s.

Figure 4.7: Threshold sweep for the strongest performing logistic regression configuration using full 10 second single window BEATs embeddings. The black line indicates the 0.70 decision threshold.

Figure 4.7 shows the threshold sweep analysis for logistic regression. Increasing the decision threshold significantly reduced false positive rates while maintaining high recall. Thresholds around 0.70 produced the strongest operational trade-off, achieving recall values above 0.98 while simultaneously reducing overall and delayed-speech false positive rates. At this decision threshold, the classifier maintained an F1-score of 0.98 while reducing the false positive rate to approximately 0.0014.

A similar threshold sweep was performed for XGBoost, with a resulting threshold of 0.75 producing a balanced operational trade-off between the recall and false positive rate. The corresponding threshold sweep curves are provided in Figure A.1

Overall, the experiments demonstrated that hyperparameter optimization gave slight performance improvements, but did not alter the ranking of the evaluated representations. Full 10 second single window representations consistently remained the strongest performing representation across both classifiers. Threshold optimization produced significant operational improvements, reducing false positive rates while maintaining high recall for ”pocket calls”.

4.7 Held-Out Test Set Evaluation

Following hyperparameter optimization and threshold tuning, the final selected logistic regression and XGBoost models were evaluated on the held-out test set containing unseen emergency and non-emergency calls. The intended outcome of this section was to assess how well the selected models generalized to previously unseen data. In addition to overall performance, further analyses were conducted to investigate model robustness under delayed-speech emergency call conditions, emergency calls containing only respiratory audio, and early detection scenarios with limited available audio.

Performance was evaluated using precision, recall, F1-score, PR-AUC, false positive rate (FPR) and false negative rate (FNR). Confidence intervals were also computed using both Wilson score intervals and bootstrap confidence intervals in order to estimate uncertainty of the reported metrics.

4.7.1 Full Test Set Performance

Table 4.10 presents the final held-out test set performance for the selected logistic regression and XGBoost models using the full 10-second BEATs representation. The full test set contained 1348 real emergency calls, where 70 samples corresponded to ”Delayed-speech emergency calls (5 s)” and 8 samples to ”Delayed-speech emergency calls (10 s)”, and 26 ”pocket calls”.

Table 4.10: Final held-out test set performance for logistic regression and XGBoost.

Model	Precision	Recall	F1	PR-AUC	FPR	FPR Delayed-10s
LR	1.00	0.92	0.96	0.97	0.00	0.00
XGB	0.96	0.88	0.92	0.98	0.0007	0.00

The logistic regression model achieved strong overall performance on the unseen test set, obtaining a recall of 0.92 and an F1-score of 0.96. No false positives were

produced on the negative test samples. The model achieved a PR-AUC of 0.97, indicating strong separation capability between emergency and non-emergency calls. Among the 26 ”pocket calls”, two were misclassified as false negatives.

Confidence interval analysis indicated relatively stable performance despite limited number of ”pocket call” samples in the test set. The Wilson score interval for recall ranged from 0.76 to 0.98 while the bootstrap confidence interval ranged from 0.81 to 1.00. The bootstrap confidence interval for PR-AUC ranged from 0.91 to 1.00, suggesting consistently high ranking performance across resampled test sets.

The XGBoost model achieved a recall of 0.88 and an F1-score of 0.92 on the same test set. One false positive was produced, and the model achieved a PR-AUC of 0.98. Although the PR-AUC was slightly higher than logistic regression, the XGBoost model produced a lower recall, missing three emergency calls compared to two for logistic regression.

For XGBoost, the Wilson score interval for recall ranged from 0.71 to 0.96, while the bootstrap confidence interval ranged from 0.73 to 1.00. The bootstrap confidence interval for PR-AUC ranged from 0.93 to 1.00. Similarly to logistic regression, the intervals remained relatively stable overall, with the intervals for ”pocket calls” specific metrics being broader likely due to limited sample size.

The confusion matrix for the logistic regression model is shown in Figure A.2, while the corresponding confusion matrix for XGBoost is shown in Figure A.3. These provide a detailed breakdown of the classification outcomes.

Overall, both models generalized well to unseen data and achieved strong performance. However, logistic regression demonstrated more favorable operational characteristics due to its higher recall and absence of false positives.

4.7.2 Delayed-speech Emergency Evaluation

To further investigate robustness under operationally difficult conditions, an additional evaluation was conducted using delayed-speech emergency calls. In these experiments, only the first five seconds of audio were made available to the models before classification. In total, 72 such calls were available in the test set. This setup simulated situations in which emergency related speech occurs late in the call. Table 4.11 presents the evaluation metrics for this experiment setup. Only 11 real emergency calls with delayed speech up to 10 seconds were available in the test data, which is why we chose to go with the shorter five second duration.

Table 4.11: Performance under delayed-speech conditions using only 5 seconds of available audio.

Model	Precision	Recall	F1	PR-AUC	FPR	FPR Delayed-5s
LR	0.81	0.96	0.88	0.96	0.086	0.086
XGB	0.76	0.85	0.80	0.91	0.10	0.10

Both models experienced performance degradation under the delayed-speech condition compared to the full 10 second evaluation. The logistic regression model maintained comparatively stable recall and F1-score performance despite the restricted audio duration. Although false positive rates increased slightly, the model continued to preserve a good tradeoff between false positive rates and recall. The XGBoost model exhibited a larger performance reduction under the same conditions, particularly in recall.

The confusion matrices for the delayed-speech evaluation are shown in Figure A.4 and Figure A.5. These provide a detailed breakdown of the classification outcomes under delayed-speech conditions.

For the delayed-speech evaluation, wider confidence intervals were observed compared to the full test set evaluation, likely due to smaller number of samples. For logistic regression, the Wilson score interval for recall ranged from 0.81 to 0.99, while the bootstrap confidence interval ranged from 0.89 to 1.00. The corresponding false positive rate confidence intervals remained comparatively wider, with the Wilson score interval of the overall false positive rate ranging from 0.040 to 0.18, and the bootstrap confidence interval ranged from 0.029 to 0.16. This reflects the increased uncertainty associated with the limited number of negative delayed-speech samples.

4.7.3 Evaluation on Operationally Important Non-Speech Calls

An additional robustness evaluation was conducted on a small held-out subset of emergency calls that contained respiratory audio with no clear speech. These calls were presented in Table 3.2, and consisted of audible breathing sounds and were excluded from training in order to assess generalization to unseen operationally important non-verbal emergency calls. These calls were deemed operationally important because they represent emergency situations with limited verbal information, where incorrectly classifying the call as accidental would be a critical error.

Among the eight respiratory audio emergency calls included in this evaluation subset, both logistic regression and XGBoost correctly classified seven calls as non-accident. Although this evaluation subset was limited in size, the results indicate that both models were able to distinguish respiratory audio emergency calls from ”pocket calls” with relatively high specificity, even in the absence of speech. The Wilson score interval for specificity ranged from 0.53 to 0.98 for both classifiers, indicating a significant uncertainty, likely due to the small sample size. These results

should primarily be interpreted as an exploratory robustness analysis rather than a conclusive performance evaluation.

4.7.4 Early Detection Performance

Early detection performance was also evaluated to investigate how early emergency calls could be reliably identified under the current circumstances when only the initial portion of audio was available to the models. In this experiment, the available audio duration was progressively limited to 2, 4, 6, 8, and 10 seconds from the beginning of each call. The objective was to analyze how performance evolved as additional audio data became available.

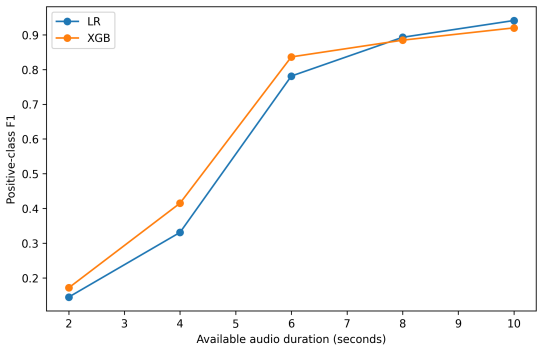
Figure 4.8 presents the early detection performance trends for logistic regression and XGBoost across different audio durations. Both models demonstrated significantly reduced performance when only the earliest 2 to 4 seconds of audio were available. Under these conditions, the logistic regression model achieved an F1-score of 0.16, while XGBoost achieved an F1-score of 0.17. Although recall remained relatively high for both models at short durations, F1-score was low due to a large number of false positives. This indicates that the initial seconds of emergency calls often lack enough information for reliable classification of this specific task.

Performance improved significantly as more audio became available. At 6 seconds, logistic regression achieved an F1-score 0.81 with a recall of 0.96, while XGBoost achieved an F1-score of 0.84 with a recall of 0.88. At 8 seconds, both models reached performance levels close to the full 10-second evaluation. Logistic regression achieved an F1-score of 0.89 and maintained a recall of 0.96, while XGBoost achieved an F1-score of 0.88 with a recall of 0.88.

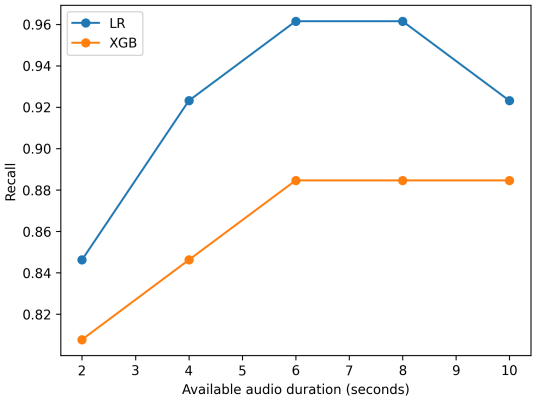
The delayed-speech false positive analysis also highlights the difficulty of early emergency detection. Both models exhibited elevated false positive rates for delayed-speech emergency calls at short durations. However, these rates decreased steadily as additional audio became available, with the subset of ”Delayed-speech emergency calls (5 s)” sharply decreasing around the six second mark. This likely corresponds to the point at which speech begins to appear in these types of calls.
Overall, early detection experiments demonstrated that reliable emergency call classification under the current setup became feasible after approximately 6-8 seconds of available call audio.

4.8 Future Work

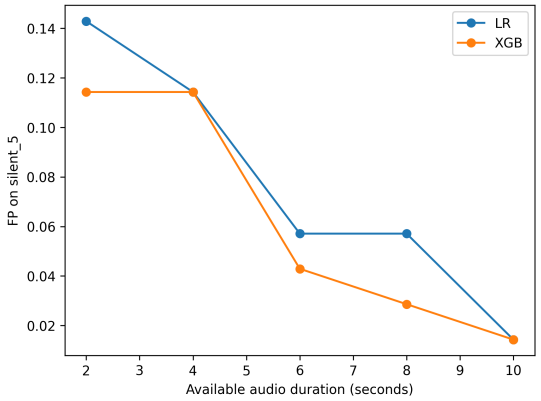
There are several directions for future work that could extend the findings of this study. One important limitation concerns the available ”pocket call” data. Although the collected dataset contained a significant amount of ordinary emergency calls, only 128 calls were identified as ”pocket calls”. Also, the ”pocket call” category was constructed through manual inspection of the audio recordings, introducing subjectivity into the labeling process. Future work should therefore investigate the



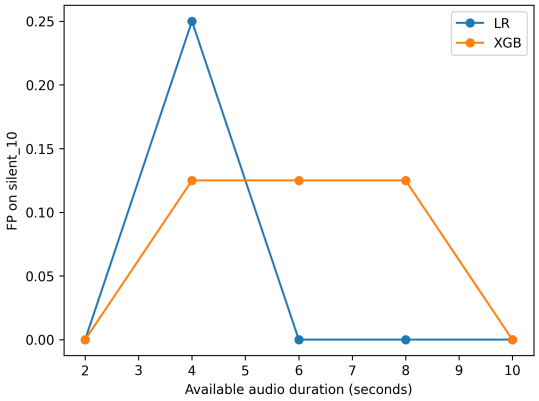
(a) F1-score versus available audio duration.



(b) Recall versus available audio duration.



(c) FPR on delayed-speech emergency calls (5 s) versus available audio duration.



(d) FPR on delayed-speech emergency calls (10 s) versus available audio duration.

Figure 4.8: Early detection performance for logistic regression (LR) and XGBoost (XGB) using progressively longer portions of the initial call audio. Shown are F1-score, recall, and false positive rates on delayed-speech emergency calls.

collection of larger and more systematically annotated ”pocket call” datasets in order to improve the robustness and generalizability of the findings. It may also be valuable to investigate whether additional subgroups of silent emergency calls could be identified, beyond just ”pocket calls”.

Another important direction concerns the analyzed audio interval. The current approach only utilized audio captured after the emergency call had been answered by the operator. Future work could investigate whether audio occurring between call initiation and call pickup contains important audio cues and acoustic patterns that may improve earlier ”pocket call” detection.

This work was conducted without the use of any metadata. Future work could investigate whether additional contextual information, such as time of day or historical call patterns, could be incorporated to improve classifier performance.

Finally, this study primarily focused on the analysis and feasibility of ”pocket call”

detection rather than real-world deployment considerations. Future work could therefore investigate how such systems may be integrated into operational emergency workflows and how automated decisions should be handled. Since the current approach is based on caller-side audio analysis, one possible direction could involve systems that provide feedback to callers when a call is classified as a potential "pocket call". In such a system, a reduction in call priority could be reversed by manual caller confirmation in cases of incorrect classification. However, the suitability of such systems would depend greatly on the operational requirements of the emergency dispatch system and requires careful consideration.

5
Conclusion

This study investigated the feasibility of identifying accidental silent 112 calls using only caller-side audio. Two different feature representations were evaluated, conventional handcrafted acoustic features and pretrained self-supervised BEATs embeddings. Logistic regression and XGBoost classifiers were evaluated across multiple experiments, including temporal windowing and segment aggregation strategies, delayed-speech scenarios, and early detection settings.

The findings showed that attempting to classify the entire "Silent 112" category as a single class resulted in poor and unstable performance. In contrast, substantially stronger results were obtained when focusing specifically on the "pocket call" subgroup. Across nearly all experiments, audio embeddings extracted from the pretrained BEATs model consistently outperformed conventional handcrafted features, indicating that pretrained self-supervised audio representations capture more informative acoustic structure for this task. The strongest overall performance was achieved using BEATs embeddings together with logistic regression on a 10-second audio window, achieving high recall and low false positive rates.

The experiments also demonstrated that reliable classification became increasingly feasible as additional audio context became available. Although very short audio intervals produced unstable predictions and high false positive rates, performance improved substantially after approximately 6 to 8 seconds of available audio.

Despite the promising results, the findings should be interpreted with caution. The study was conducted on a limited number of manually identified pocket calls, and the subgroup labeling process involved subjective manual inspection of calls. Extensive validation, larger datasets annotated by experts in the field, and operational considerations would therefore be required before considering integration into real emergency call systems.

Overall, the results indicate that the caller-side acoustic analysis has potential as a supporting tool to identify accidental silent emergency calls. Although current work should be viewed as preliminary, it provides a foundation for continued research on automated analysis of silent emergency calls.

[1] SOS Alarm, "Verksamhetsrapport 2020 112 i Sverige," SOS Alarm Sverige AB, Stockholm, Sweden, 2020. [Online] Available: https://www.sosalarm.se/globalassets/dokument/112-rapporter/112-rapporten-2020.pdf.

[2] SOS Alarm, "Verksamhetsrapport 2021 112 i Sverige," SOS Alarm Sverige AB, Stockholm, Sweden, 2021. [Online] Available: https://www.sosalarm.se/globalassets/dokument/112-rapporter/112-rapporten-2021.pdf.

[3] SOS Alarm, "Verksamhetsrapport 2022 112 i Sverige," SOS Alarm Sverige AB, Stockholm, Sweden, 2022. [Online] Available: https://www.sosalarm.se/globalassets/dokument/112-rapporter/112-rapporten-2022.pdf.

[4] SOS Alarm, "Verksamhetsrapport 2023 112 i Sverige," SOS Alarm Sverige AB, Stockholm, Sweden, 2023. [Online] Available: https://www.sosalarm.se/globalassets/dokument/112-rapporter/112-rapporten-2023.pdf.

[5] SOS Alarm, "Verksamhetsrapport 2024 112 i Sverige," SOS Alarm Sverige AB, Stockholm, Sweden, 2024. [Online] Available: https://www.sosalarm.se/globalassets/dokument/112-rapporter/112-rapport_2024.pdf.

[6] SOS Alarm, "Verksamhetsrapport 2025 112 i Sverige," SOS Alarm Sverige AB, Stockholm, Sweden, 2025. [Online]. Available: https://www.sosalarm.se/globalassets/dokument/112-rapporter/112-rapport-2025.pdf.

[7] T. Nilsson, "Tysta nödsamtal överbelastar SOS Alarm," *Sveriges Radio*, Feb. 11, 2026. [Online] Available: https://www.sverigesradio.se/artikel/tysta-nodsamtal-overbelastar-sos.

[8] "Allt fler felaktiga larmsamtal med ny teknik," *SOS Alarm*, Sep. 18, 2023. [Online] Available: https://www.sosalarm.se/om-oss/pressrum/nyheter/2023/allt-fler-felaktiga-larmsamtal-med-ny-teknik/.

[9] O. Blushtein, M. Siman-Tov, and R. Magnezi, "Identifying and minimizing abuse of emergency call center services through technology," *American Journal of Emergency Medicine*, vol. 38, no. 5, pp. 916–919, May 2020. [Online] Available: https://doi.org/10.1016/j.ajem.2019.07.015.

[10] E. Stokoe and E. Richardson, "Asking for help without asking for help: How victims request and police offer assistance in cases of domestic violence when perpetrators are potentially co-present," *Discourse Studies*, vol. 25, no. 3, pp. 383–408, Mar. 2023. [Online] Available: https://doi.org/10.1177/14614456231157293.

[11] S. Rafi et al., "Out-of-hospital cardiac arrest detection by machine learning based on the phonetic characteristics of the caller's voice," *Studies in Health Technology and Informatics*, vol. 294, pp. 445–449, May 2022. [Online] Available: https://doi.org/10.3233/SHTI220498.

[12] K. Gormley, K. Lockhart, and J. Isaac, "Using natural language processing in facilitating pre-hospital telephone triage of emergency calls," *British Paramedic Journal*, vol. 7, no. 2, pp. 31–37, Sep. 2022. [Online] Available: https://doi.org/10.29045/14784726.2022.09.7.2.31.

[13] M. Abi Kanaan, J.-F. Couchot, C. Guyeux, D. Laiymani, T. Atechian, and R. Darazi, "A methodology for emergency calls severity prediction: from preprocessing to BERT-based classifiers," in *Proc. 19th IFIP Int. Conf. Artificial Intelligence Applications and Innovations (AIAI)*, León, Spain, Jun. 2023,

pp. 329–342. [Online] Available: https://doi.org/10.1007/978-3-031-34111-3_28.

[14] "How we detect involuntary calls to the 112," *ML2Grow*, Oct. 11, 2021. [Online] Available: https://ml2grow.com/how-we-detect-involuntary-calls-to-the-112/.

[15] K. J. Piczak, "ESC: Dataset for environmental sound classification," in *Proc. 23rd ACM Int. Conf. Multimedia (MM'15)*, Brisbane, Australia, Oct. 2015, pp. 1015–1018. doi: 10.1145/2733373.2806390. [Online] Available: https://doi.org/10.1145/2733373.2806390.

[16] D. Barchiesi, D. Giannoulis, D. Stowell, and M. D. Plumbley, "Acoustic scene classification: Classifying environments from the sounds they produce," *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 16–34, May. 2015. [Online] Available: https://doi.org/10.1109/MSP.2014.2326181.

[17] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015. [Online] Available: https://doi.org/10.1038/nature14539.

[18] Z. Zhao, L. Alzubaidi, J. Zhang, Y. Duan, and Y. Gu, "A comparison review of transfer learning and self-supervised learning: Definitions, applications, advantages and limitations," *Expert Systems with Applications*, vol. 242, Art. no. 122807, May 2024. [Online] Available: https://doi.org/10.1016/j.eswa.2023.122807.

[19] F. Chen, Z. Zhu, C. Sun, and L. Xia, "Evaluating metric and contrastive learning in pretrained models for environmental sound classification," *Applied Acoustics*, vol. 232, p. 110593, Mar. 2025. [Online] Available: https://doi.org/10.1016/j.apacoust.2025.110593.

[20] S. Chen et al., "BEATs: Audio pre-training with acoustic tokenizers," in *Proc. 40th Int. Conf. Machine Learning (ICML)*, Honolulu, HI, USA, Jul. 2023, in *Proc. Mach. Learn. Res.*, vol. 202, pp. 5178–5193. [Online] Available: https://proceedings.mlr.press/v202/chen23ag.html.

[21] "AI Act," *Shaping Europe's Digital Future*, Jan. 27, 2026. [Online] Available: https://digital-strategy.ec.europa.eu/en/policies/regulatory-framework-ai.

[22] "Annex III: High-Risk AI Systems Referred to in Article 6(2)," *EU Artificial Intelligence Act*. [Online] Available: https://artificialintelligenceact.eu/annex/3/ [Accessed: Apr. 25, 2026].

[23] "Recital 58," *EU Artificial Intelligence Act*. [Online] Available: https://artificialintelligenceact.eu/recital/58/ [Accessed: Apr. 25, 2026].

[24] "Article 6: Classification Rules for High-Risk AI Systems," *EU Artificial Intelligence Act*. [Online] Available: https://artificialintelligenceact.eu/article/6/ [Accessed: Apr. 25, 2026].

[25] Z.-H. Tan and B. Lindberg, "Low-complexity variable frame rate analysis for speech recognition and voice activity detection," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 5, pp. 798–807, Oct. 2010. [Online] Available: https://doi.org/10.1109/JSTSP.2010.2057192.

[26] M. F. McKinney and J. Breebaart, "Features for audio and music classification," in *Proc. 4th Int. Conf. Music Information Retrieval (ISMIR)*, Baltimore,

MD, USA, Oct. 2003, pp. 151–158. [Online] Available: https://ismir2003.ismir.net/papers/McKinney.pdf.

[27] C. Constantinescu and R. Brad, "An overview on sound features in time and frequency domain," *Int. J. Adv. Stat. IT&C Econ. Life Sci.*, vol. 13, no. 1, pp. 45–58, Dec. 2023. [Online] Available: https://doi.org/10.2478/ijasitels-2023-0006.

[28] L. Chen, X. Zhou, Q. Tu, and H. Chen, "Enhancing speech and music discrimination through the integration of static and dynamic features," in *Proc. Interspeech 2024*, Kos, Greece, Sep. 2024, pp. 4318–4322. [Online] Available: https://doi.org/10.21437/Interspeech.2024-1596.

[29] G. Peeters, "A large set of audio features for sound description (similarity and classification) in the CUIDADO project," IRCAM–Centre Pompidou, Paris, France, CUIDADO I.S.T. Project Report, Apr. 2004. [Online] Available: http://recherche.ircam.fr/equipes/analyse-synthese/peeters/ARTICLES/Peeters_2003_cuidadoaudiofeatures.pdf.

[30] Y. A. Ibrahim, J. C. Odiketa, and T. S. Ibiyemi, "Preprocessing technique in automatic speech recognition for human computer interaction: An overview," *Annals. Computer Science Series*, vol. 15, no. 1, pp. 186–191, 2017. [Online] Available: https://anale-informatica.tibiscus.ro/download/lucrari/15-1-23-Ibrahim.pdf.

[31] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Trans. Speech Audio Process.*, vol. 10, no. 5, pp. 293–302, Jul. 2002. [Online] Available: https://doi.org/10.1109/TSA.2002.800560.

[32] U. Nam and J. Berger, "Addressing the Same but Different–Different but Similar Problem in Automatic Music Classification," in *Proc. 2nd Annu. Int. Symp. Music Information Retrieval (ISMIR)*, Bloomington, IN, USA, Oct. 2001, pp. 21–22. [Online] Available: https://ismir2001.ismir.net/posters/nam.pdf.

[33] D. Bergmann, "What is self-supervised learning?," *IBM Think*. [Online] Available: https://www.ibm.com/think/topics/self-supervised-learning [Accessed: Mar. 12, 2026].

[34] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020, pp. 12449–12460. [Online] Available: https://proceedings.neurips.cc/paper/2020/hash/92d1e1eb1cd6f9fba3227870bb6d7f07-Abstract.html.

[35] W.-N. Hsu et al., "HuBERT: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 29, pp. 3451–3460, 2021. [Online] Available: https://doi.org/10.1109/TASLP.2021.3122291.

[36] J. F. Gemmeke et al., "Audio Set: An ontology and human-labeled dataset for audio events," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, LA, USA, Mar. 2017, pp. 776–780. [Online] Available: https://doi.org/10.1109/ICASSP.2017.7952261.

[37] J. Ramirez, J. M. Gorriz, and J. C. Segura, "Voice Activity Detection. Fundamentals and Speech Recognition System Robustness," in *Robust Speech*

Recognition and Understanding, M. Grimm and K. Kroschel, Eds. London, U.K.: IntechOpen, 2007, ch. 1, pp. 1–22. [Ebook] Available: https://www.intechopen.com/chapters/104.

[38] D. Rho, J. Park, and J. H. Ko, "NAS-VAD: Neural Architecture Search for Voice Activity Detection," in *Proc. Interspeech 2022*, Incheon, Korea, Sep. 2022, pp. 3754–3758. doi: 10.21437/Interspeech.2022-975. [Online] Available: https://www.isca-archive.org/interspeech_2022/rho22_interspeech.html.

[39] A. Veysov and D. Voronin, "One Voice Detector to Rule Them All," *The Gradient*, Feb. 19, 2022. [Online] Available: https://thegradient.pub/one-voice-detector-to-rule-them-all/ [Accessed: Mar. 12, 2026].

[40] Silero Team, "Quality Metrics," *Silero VAD Wiki*, GitHub, Mar. 26, 2026. [Online] Available: https://github.com/snakers4/silero-vad/wiki/Quality-Metrics [Accessed: May 8, 2026].

[41] D. W. Hosmer, Jr., S. Lemeshow, and R. X. Sturdivant, *Applied Logistic Regression*, 3rd ed., Hoboken, NJ, USA: John Wiley & Sons, 2013. [Ebook] Available: https://doi.org/10.1002/9781118548387.

[42] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, Aug. 1997. [Online] Available: https://doi.org/10.1006/jcss.1997.1504.

[43] R. Appel, T. Fuchs, P. Dollár, and P. Perona, "Quickly boosting decision trees–pruning underachieving features early," in *Proc. 30th Int. Conf. Machine Learning (ICML)*, Atlanta, GA, USA, Jun. 2013, in *Proc. Mach. Learn. Res.*, vol. 28, no. 3, pp. 594–602. [Online] Available: https://proceedings.mlr.press/v28/appel13.html.

[44] Y. Freund and R. E. Schapire, "A Short Introduction to Boosting," *Japanese Society for Artificial Intelligence*, vol. 14, no. 5, pp. 771–780, Sep. 1999. [Online] Available: https://cseweb.ucsd.edu/~yfreund/papers/IntroToBoosting.pdf.

[45] R. Johansson, "An intuitive explanation of gradient boosting," Chalmers University of Technology. [Online] Available: https://www.cse.chalmers.se/~richajo/dit866/files/gb_explainer.pdf [Accessed: May 8, 2026].

[46] M. Wiens, A. Verone-Boyle, N. Henscheid, J. T. Podichetty, and J. Burton, "A tutorial and use case example of the eXtreme Gradient Boosting (XGBoost) artificial intelligence algorithm for drug development applications," *Clinical and Translational Science*, vol. 18, no. 3, Art. no. e70172, Mar. 2025. [Online] Available: https://doi.org/10.1111/cts.70172.

[47] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *The Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, Oct. 2001. [Online] Available: https://doi.org/10.1214/aos/1013203451.

[48] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD)*, San Francisco, CA, USA, Aug. 2016, pp. 785–794. [Online] Available: https://doi.org/10.1145/2939672.2939785.

[49] R. Shwartz-Ziv and A. Armon, "Tabular data: Deep learning is not all you need," *Information Fusion*, vol. 81, pp. 84–90, May. 2022. [Online] Available: https://doi.org/10.1016/j.inffus.2021.11.011.

[50] V. Borisov, T. Leemann, K. Seßler, J. Haug, M. Pawelczyk, and G. Kasneci, "Deep neural networks and tabular data: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 6, pp. 7499–7519, Jun. 2024. [Online] Available: https://doi.org/10.1109/TNNLS.2022.3229161.

[51] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning: with Applications in R*, 2nd ed., New York, NY, USA: Springer, 2021. [Ebook] Available: https://doi.org/10.1007/978-1-0716-1418-1.

[52] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, no. 85, pp. 2825–2830, 2011. [Online] Available: https://jmlr.org/papers/v12/pedregosa11a.html.

[53] J. Davis and M. Goadrich, "The relationship between Precision-Recall and ROC curves," in *Proc. 23rd Int. Conf. Machine Learning (ICML)*, Pittsburgh, PA, USA, Jun. 2006, pp. 233–240. [Online] Available: https://doi.org/10.1145/1143844.1143874.

[54] O. Rainio, J. Teuho, and R. Klén, "Evaluation metrics and statistical tests for machine learning," *Scientific Reports*, vol. 14, Art. no. 6086, Mar. 2024. [Online] Available: https://doi.org/10.1038/s41598-024-56706-x.

[55] J. P. Jones II, L. J. Farrell, and L. M. Benson, Eds., *Informational Scientific Primers for Officers of the Court: Intended to Strengthen Use of Forensic Science Evidence*, National Institute of Standards and Technology, Gaithersburg, MD, USA, NIST Special Publication 1500-28, Mar. 2025. [Online] Available: https://doi.org/10.6028/NIST.SP.1500-28.

[56] E. B. Wilson, "Probable inference, the law of succession, and statistical inference," *Journal of the American Statistical Association*, vol. 22, no. 158, pp. 209–212, Jun. 1927. [Online] Available: https://doi.org/10.1080/01621459.1927.10502953.

[57] B. Efron and R. J. Tibshirani, *An Introduction to the Bootstrap*, New York, NY, USA: Chapman & Hall/CRC, 1994. [Ebook] Available: https://doi.org/10.1201/9780429246593.

[58] Silero Team, "utils_vad.py," *Silero VAD*, [Source code], GitHub, 2024. [Online] Available: https://github.com/snakers4/silero-vad/blob/master/src/silero_vad/utils_vad.py [Accessed: Apr. 26, 2026].

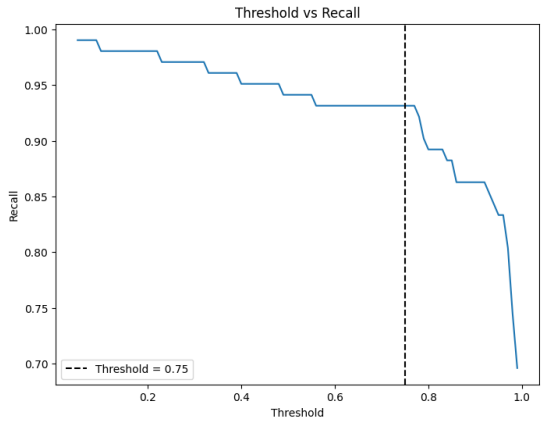
[59] H. F. Inman and E. L. Bradley, Jr., "The overlapping coefficient as a measure of agreement between probability distributions and point estimation of the overlap of two normal densities," *Communications in Statistics - Theory and Methods*, vol. 18, no. 10, pp. 3851–3874, 1989. [Online] Available: https://doi.org/10.1080/03610928908830127.

A

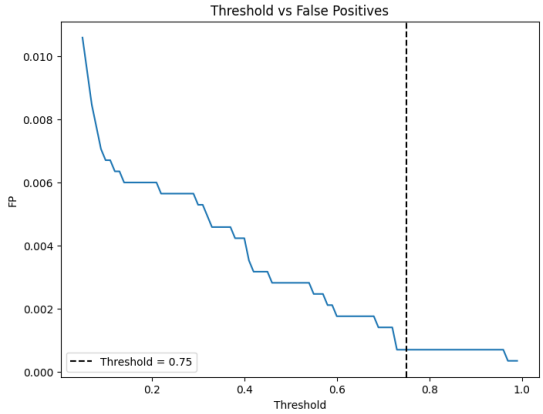
Appendix

Table A.1: Delayed-10s overlap comparison

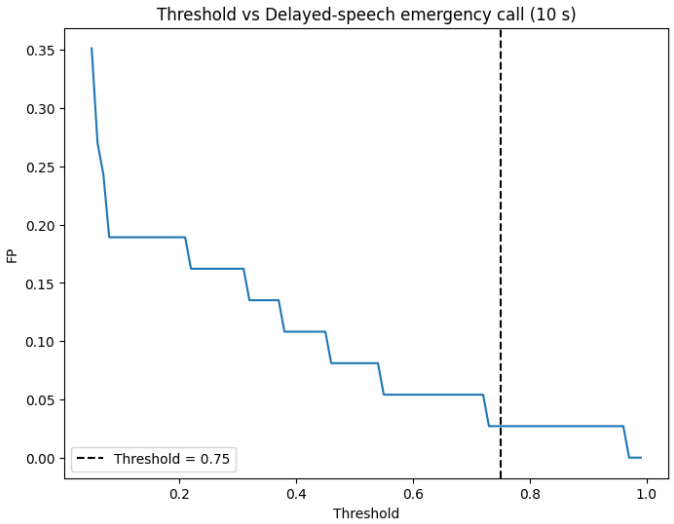
Feature	Silent 112 Overlap	Pocket Call Overlap
RMS (mean)	0.89	0.24
RMS (std)	0.80	0.29
Spectral Bandwidth (mean)	0.77	0.34
Spectral Roll-off (mean)	0.75	0.44
Spectral Roll-off (std)	0.79	0.45
ZCR (mean)	0.80	0.46
Spectral Bandwidth (std)	0.76	0.47
Spectral Centroid (mean)	0.79	0.49
ZCR (std)	0.85	0.52
Spectral Centroid (std)	0.84	0.52
Spectral Flatness (std)	0.98	0.71
Spectral Flatness (mean)	0.99	0.93



(a) Threshold vs Recall.



(b) Threshold vs FPR.



(c) Threshold vs FPR Delayed-10s.

Figure A.1: Threshold sweep for the strongest performing XGBoost configuration using full 10 second single window BEATs embeddings. The black line indicates the 0.75 decision threshold.

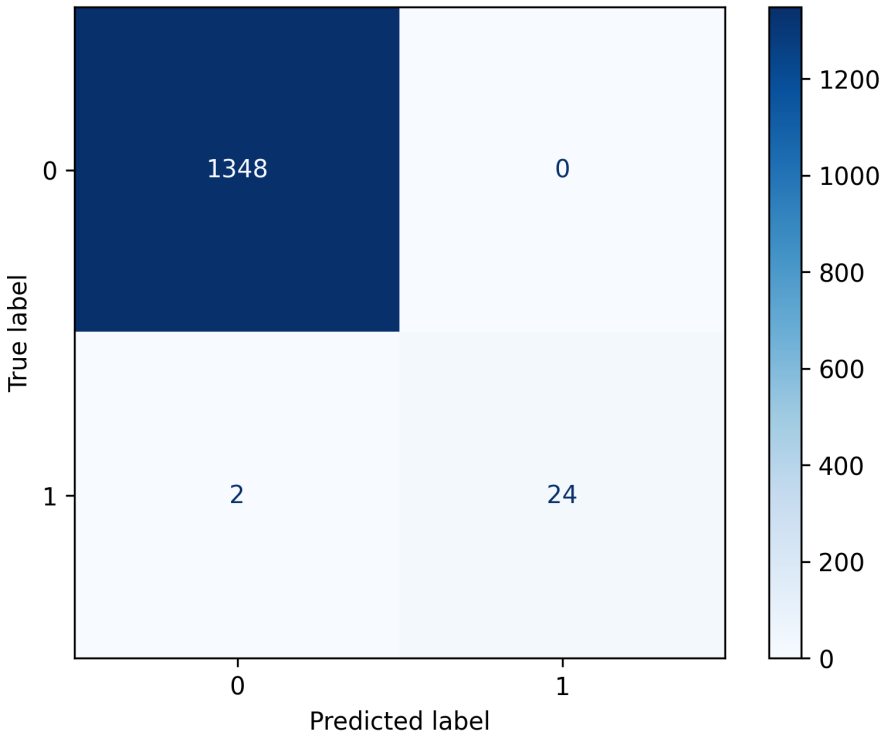


Figure A.2: Confusion matrix for the final logistic regression model on the held-out test set. The model correctly classified 24 of 26 pocket calls and produced no false positives.

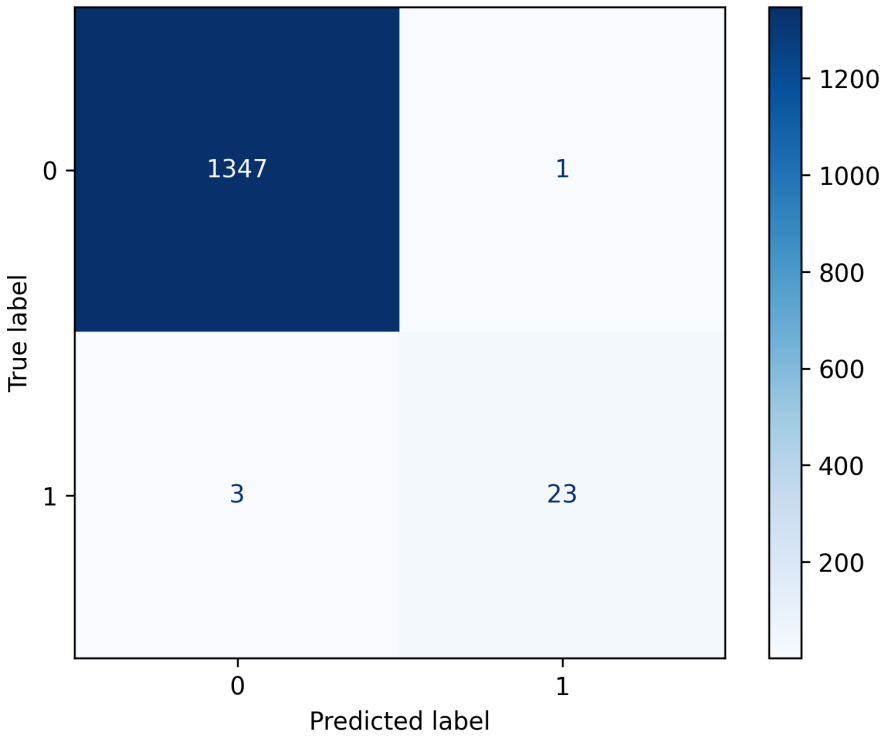


Figure A.3: Confusion matrix for the final XGBoost model on the held-out test set. The model correctly classified 23 of 26 pocket calls and produced one false positive.

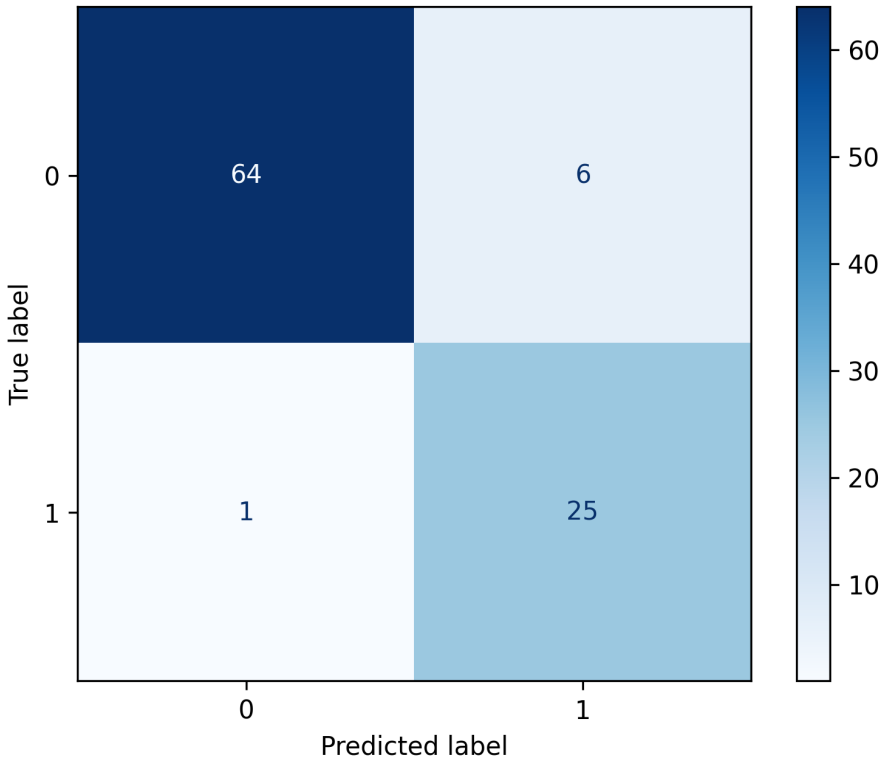


Figure A.4: Confusion matrix for the final logistic regression model under delayed-speech conditions on the test set, using only the first 5 seconds of available audio.

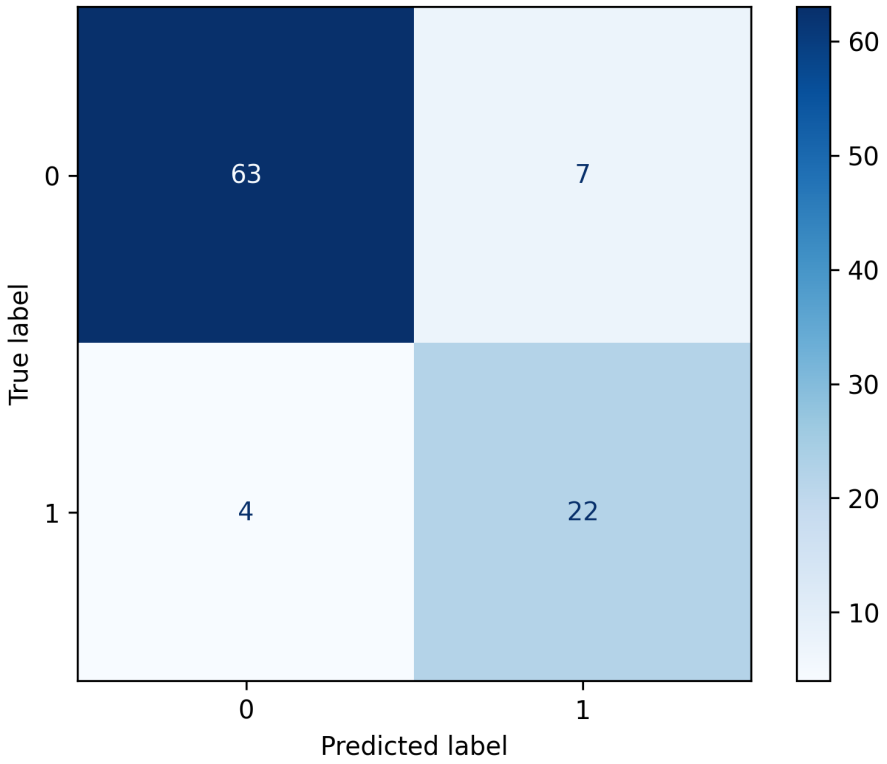


Figure A.5: Confusion matrix for the final XGBoost model under delayed-speech conditions on the test set, using only the first 5 seconds of available audio.

DEPARTMENT OF MATHEMATICAL SCIENCES
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY