ROMANIAN JOURNAL OF INFORMATION SCIENCE AND TECHNOLOGY
Volume 29, Number 3, 2026, 225–236
doi: 10.59277/ROMJIST.2026.3.02

A Multi-Stage Poisoning Detection Pipeline for Embedded Voice-Command Systems

Sebastian-Alexandru DRAGUSIN[1,*], Denisa TOMA[1], Robert-Nicolae BOSTINARU[1], and Nicu BIZON[2]

[1]Doctoral School, National University of Science and Technology POLITEHNICA Bucharest, 060042 Bucharest, Romania

[2]Faculty of Electronics, Communication and Computers, National University of Science and Technology POLITEHNICA Bucharest, Pitesti University Centre, 110040 Pitesti, Romania

Email: sebastian.dragusin@upb.ro*, denisa.toma@stud.etti.upb.ro, robert.bostinaru@stud.etti.upb.ro, nicu.bizon1402@upb.ro

* Corresponding author

Abstract. Embedded voice-command systems are increasingly adopted in cyber-physical and industrial environments, yet their machine-learning pipelines remain vulnerable to data poisoning attacks that can silently degrade recognition performance or induce targeted misclassifications. This paper proposes a multi-stage poisoning detection pipeline tailored for embedded voice-command applications, combining fast integrity screening with progressively stronger detection models. First, an audio integrity layer analyzes waveform and spectrogram consistency to flag abnormal patterns indicative of injected perturbations or corrupted samples. Second, a lightweight machine-learning detector based on Random Forests operates on compact statistical descriptors extracted from spectrogram representations, enabling efficient on-device or edge-level screening. Third, a spectrogram-based Convolutional Neural Network provides high-sensitivity discrimination between clean and poisoned samples, acting as a robust validation stage in higher-assurance deployments. To increase resilience, the training procedure integrates regularization and controlled augmentation to mitigate overfitting to spurious artifacts and reduce poisoning sensitivity. The proposed approach is designed to be deployable under embedded constraints by separating low-cost screening from deeper analysis, and it offers a structured methodology for securing voice-command pipelines against poisoning threats.

Key-words: Anomaly detection; convolutional neural networks; cyber-physical systems; data poisoning; embedded systems; spectrogram analysis; voice-command recognition.

1. Introduction

Embedded voice-command systems are increasingly integrated into cyber-physical and industrial environments, where speech interaction can simplify operator workflows and enable hands-free control [1]. In these settings, however, the adoption of Machine Learning (ML) pipelines expands the attack surface beyond classical software vulnerabilities, because the quality and integrity of training data directly influence operational behavior [2].

This issue is particularly acute in embedded deployments, where models may be deployed for long periods, data acquisition can occur in heterogeneous environments, and update/validation cycles may be constrained by operational requirements [3].

Within this threat landscape, data poisoning represents a high-impact class of attacks in which an adversary corrupts a fraction of the training dataset to reduce generalization or to induce predictable errors at inference time. The poisoning mechanisms relevant to voice-command recognition include label flipping (e.g., swapping the label of a command), backdoor-style poisoning (embedding a hidden signal that triggers misbehavior only when present), and carefully controlled corruption rates that balance stealth with effectiveness. In voice recognition scenarios, poisoning can be engineered so that the system confuses semantically distinct commands for instance, causing "stop" to be interpreted as "start", with direct safety implications in embedded control loops [4].

A practical defense is complicated by the fact that poisoned samples can be subtle, and artifacts may manifest differently depending on representation and processing stage. For voice-command datasets, integrity screening can be performed by analyzing spectrogram consistency and detecting abnormal frequency or amplitude patterns, complemented by statistical or anomaly-detection techniques that highlight outliers before they enter the training pipeline.

This paper proposes a multi-stage poisoning detection pipeline for embedded voice-command systems, built around three complementary stages that align with embedded constraints. First, a signal/representation integrity layer performs rapid consistency checks over waveform-derived characteristics and spectrogram structures to identify samples that deviate from the expected acoustic profile. Second, a lightweight ML detector based on Random Forests (RF) performs efficient screening using compact statistical descriptors extracted from spectrograms, namely the mean, standard deviation, minimum, and maximum values, enabling a low-cost decision mechanism suitable for device- or edge-level deployment. Third, a deep verification stage uses a Convolutional Neural Network (CNN) trained to discriminate between "Original" and "Altered" samples using spectrogram inputs, offering higher sensitivity for subtle perturbations and enabling a robust secondary validation in higher-assurance deployments.

Beyond detection, the paper emphasizes robustness-oriented training as a complementary mitigation strategy. The proposed CNN-based detector employs a standardized preprocessing pipeline in which spectrograms are resized to 128×128 to ensure uniform training, and the model is optimized using Adam with binary cross-entropy for binary discrimination between clean and altered instances. Robustness is strengthened through the explicit use of dropout, L_2 regularization, and data augmentation, together with controlled exposure to both altered and original samples during training to reduce sensitivity to noise-like perturbations and limit overfitting to spurious artifacts. In addition, the broader operational perspective motivates continuous monitoring and feedback mechanisms, where systematic auditing of new data and model responses can support long-term resilience as data distributions and attack strategies evolve.

From an applied perspective, the proposed pipeline is intended to be compatible with industrial integration requirements, where security measures must be aligned with existing protocols

and operational workflows, and their impact on productivity must be evaluated through controlled testing and revalidation cycles. This motivates a staged architecture that separates low-cost screening from deeper analysis, allowing embedded and industrial deployments to trade off computational cost against assurance without compromising operational constraints.

Despite the growing body of research on poisoning detection, most existing approaches focus on isolated detection mechanisms or computationally intensive models that are difficult to deploy under embedded constraints. In practical voice-command systems, lightweight statistical screening, fast anomaly filtering, and high-sensitivity verification must operate jointly to ensure both efficiency and robustness. This motivates the development of a structured multi-stage detection architecture that progressively refines decision confidence while remaining compatible with resource-constrained embedded environments.

The remainder of this paper is organized as follows. Section 2 presents the methodological framework, including the threat model, poisoning attack construction, and the proposed multi-stage detection pipeline. Section 3 provides the experimental evaluation, detailing the dataset, validation protocol, statistical and learning-based performance analysis, and a discussion of the approach's limitations. Section 4 concludes the paper by summarizing the main findings and outlining directions for future research.

2. Methodology

This section presents the methodological framework of the study by formalizing the threat model, detailing the poisoning attack construction, and describing the proposed multi-stage detection pipeline tailored to embedded voice-command systems.

2.1. Threat model

The considered operational context is an embedded AI-enabled system that performs specialized tasks under hardware and software constraints, including limited memory, low processing power, and practical difficulty in applying security updates or patches. Such devices are often deployed in distributed and sometimes intermittently connected environments, which makes the deployment of heavyweight security controls challenging [5]. Within this context, data poisoning is treated as an emerging threat particularly relevant to embedded systems [6] that use AI for anomaly detection or for automating critical processes, because corrupted training data can lead to sub-optimal performance and incorrect decisions that directly affect system functionality and safety [7].

To ensure full control over acquisition conditions and labeling, the voice-command dataset used in the poisoning experiments was recorded by the authors, and its class balance was validated by inspecting the per-command frequency distributions before and after dataset contamination (8 commands as shown in Fig. 1 in the supplementary material [8]). In the original dataset, the command counts are closely balanced across classes, left: 133, stop: 117, yes: 120, down: 129, up: 132, no: 129, go: 120, right: 116, while in the poisoned variant the distribution remains comparable, with moderate shifts consistent with controlled corruption, left: 134, stop: 114, yes: 120, down: 119, up: 143, no: 129, go: 124, right: 117. This verification step is important because it confirms that the poisoning process does not trivially reveal itself through extreme class imbalance, thereby keeping the experimental setup aligned with realistic stealth assumptions.

2.2. Attack construction

The adversary is defined by the ability to manipulate the training dataset by modifying existing samples and/or introducing malicious samples so that the resulting learned model produces erroneous predictions or becomes intentionally vulnerable. Poisoning actions include label alteration (label flipping), changing the label of a voice command so that the model learns contradictory mappings (such as interpreting "go" as "stop", as shown in Fig. 2 in the supplementary material [8]), as well as injecting subtle noise or irrelevant information that induces false correlations and degrades command recognition performance (Fig. 3 in the supplementary material [8]). A more covert objective is achieved via backdoor-style poisoning, where a small, manipulated subset embeds a hidden behavior that is activated only when specific patterns are present at inference, while the model appears to behave normally in most other cases. Stealth is explicitly linked to the proportion of corrupted data, since excessive poisoning increases detectability, whereas overly small poisoning may not achieve the intended impact [9].

2.3. Proposed multi-stage detection pipeline

From the defensive perspective, protection against poisoning requires detection mechanisms and robust training capable of preventing and correcting negative effects in voice-command pipelines [10]. Practical defensive elements include observing discrepancies between original and corrupted signals through waveform/spectrogram analysis, using dimensionality reduction Principal Component Analysis (PCA) and reconstruction-error reasoning to isolate anomalous samples [11], and employing learning-based detectors (e.g., RF on contaminated vs. clean data) to improve anomaly recognition. Robustness is further supported by integrating adversarial training, as well as stabilizing techniques such as regularization and dropout, combined with periodic re-evaluation and feedback loops that reintroduce detected suspicious patterns into the learning process to improve long-term resilience [12].

The proposed multi-stage detection pipeline combines complementary controls that progressively reduce the probability that poisoned (i.e., subtly manipulated) audio samples contaminate the training set: an initial data-integrity/anomaly screening flags outliers that deviate from the expected acoustic patterns, a distribution-level verification compares statistical properties between reference and candidate samples, and discontinuity-based checks capture abrupt information/feature shifts that are atypical for genuine voice-command data; the remaining samples can then be filtered using ML-based detectors (e.g., RF operating on spectrogram-derived summary features such as mean, standard deviation, and extrema), optionally complemented by CNN-based screening and robust-training practices to increase resilience against Poison Generative pre-Trained Transformer (PoisonGPT)-style manipulation.

In the proposed multi-stage architecture, the detection process follows a sequential decision flow in which each component contributes complementary evidence regarding the integrity of candidate samples. The signal-integrity, distribution-based, and discontinuity analyses operate as preliminary screening stages, designed to detect statistical and structural inconsistencies at low computational cost. Following these stages, a lightweight classification algorithm based on RF performs supervised anomaly screening using compact spectrogram-derived descriptors, enabling efficient discrimination under embedded constraints. The CNN acts as a secondary, higher-sensitivity classifier, applied to refine the decision when deeper verification is required. The relationship between classifiers is therefore hierarchical and complementary: the RF provides fast initial classification, while the CNN serves as a high-confidence validation stage. The

final decision is derived through a conservative sequential logic, where a sample is considered trustworthy only if consistency is maintained across all stages, whereas any classifier signaling anomalous behavior triggers rejection or further inspection.

At the signal level, an anomaly-based filter is applied by comparing Short Time Fourier Transform (STFT) spectrograms of a reference (clean) utterance against a candidate (potentially altered) utterance; the anomaly score is defined as the mean absolute element-wise deviation between the two spectrograms [13]. Formally, the score is computed as:

$$\text{AnomalyScore} = \frac{1}{N} \sum_{i=1}^{N} |s_{\text{original}}[i] - s_{\text{altered}}[i]| \tag{1}$$

where s_{original} is the STFT spectrogram of the original signal, s_{altered} is the STFT spectrogram of the altered signal (via noise injection or label-driven manipulation scenarios), and N is the total number of spectrogram elements.

At the distribution level, a core statistical component is the Kolmogorov-Smirnov (KS) test [14], which quantifies the maximum separation between the empirical Cumulative Distribution Functions (CDFs) of the reference and candidate signals [15]. The KS statistic is defined as:

$$D_{n,m} = \sup_x |F_n(x) - F_m(x)| \tag{2}$$

where $F_n(x)$ and $F_m(x)$ denote the CDFs of the original and altered samples, respectively, and n, m are the corresponding sample sizes; the associated p-value serves as a decision indicator, with small values (commonly below 0.05) signaling statistically significant deviations consistent with poisoning-induced distortions.

Complementary to KS, a discontinuity-based detector is used to capture abrupt local changes in the waveform that are consistent with injected perturbations [16]. Consecutive absolute differences are computed for the original (3) and altered signals (4):

$$\Delta_{\text{original}} = |x_{i+1} - x_i| \tag{3}$$

$$\Delta_{\text{altered}} = |y_{i+1} - y_i| \tag{4}$$

followed by a discontinuity score defined as the mean absolute deviation between these difference sequences:

$$D = \frac{1}{N} \sum_{i=1}^{N-1} |(\Delta_{\text{altered}})_i - (\Delta_{\text{original}})_i| \tag{5}$$

A threshold rule is then applied: if $D > T$, the sample is treated as altered (anomalous); otherwise, it is treated as non-altered, where T denotes a predefined threshold.

At the ML stage, poisoning detection is further strengthened through ML-based detectors trained to capture subtle anomalies that may not be reliably exposed by classical statistical tests. In this component, a RF classifier is used as a supervised discriminator between altered and non-altered samples, leveraging the fact that poisoned audio can introduce small but systematic shifts in time-frequency structure that become measurable after feature extraction. The RF ensemble is built via bagging, where each decision tree is trained on a randomly sampled subset of the training set and a randomly sampled subset of features, and the final decision is obtained by majority voting across trees [17]. Denoting the training set as $D = \{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$,

each tree T_i is trained on a bootstrap subset $D_i \subset D$, producing a prediction $h_i(x)$ for an instance x, and the aggregated classifier is then computed as:

$$H(x) = \text{majority}\{h_1(x), h_2(x), \ldots, h_m(x)\} \tag{6}$$

where m is the number of trees and $\text{majority}\{\cdot\}$ denotes the majority vote.

The RF classifier was trained using an ensemble size of 100 decision trees, with a fixed random seed of 42 to ensure reproducibility. Each tree was learned using bootstrap sampling from the training set, and the ensemble decision was computed by majority voting as in (6). The input to the RF consists of a compact four-dimensional feature vector extracted from the STFT spectrogram expressed in dB: mean, standard deviation, maximum, and minimum, computed over the full spectrogram matrix. These parameters are therefore obtained deterministically from the time-frequency representation, while the RF model parameters (tree splits and thresholds) are learned during training by optimizing the split criterion across nodes on the bootstrapped samples.

In the proposed pipeline, the input variables to the RF are compact spectrogram-derived descriptors: mean, standard deviation, maximum, and minimum, selected to maintain low computational cost while retaining discrimination power between clean and poisoned patterns. Complementary unsupervised strategies are also relevant in this stage: clustering algorithms such as K-means [18] and Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [19] can group acoustically similar commands and highlight outliers whose structures deviate from the dominant clusters, providing an additional mechanism to separate potentially poisoned samples from valid data in voice-command datasets. At the learning-based stage, automated integrity verification is extended using models trained to discriminate "original" versus "altered" time-frequency patterns [20]. The CNN-based component operates on spectrogram inputs preprocessed to a fixed resolution (128×128) and uses 2D convolutions with Rectified Linear Unit (ReLU) activations; a standard sigmoid output layer supports binary decisioning, with probabilities below 0.5 interpreted as "Original" and above 0.5 as "Altered". The convolutional mapping and key nonlinearities are expressed in (7)-(8), and (9), respectively, as:

$$Z_{i,j} = (F * X)_{i,j} = \sum_m \sum_n X_{i+m,j+n} \cdot F_{m,n} \tag{7}$$

$$f(x) = \max\{0, x\} \tag{8}$$

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{9}$$

and the training objective employs binary cross-entropy:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^{N} [y_i \log(p_i) + (1 - y_i) \log(1 - p_i)] \tag{10}$$

where $y_i \in \{0, 1\}$ is the ground-truth label and p_i is the predicted probability.

The CNN classifier operates on STFT spectrograms computed with a hop length of 512 samples and an FFT size of 2048, converted to a dB scale and then preprocessed to a fixed 128×128 representation by zero-padding/cropping and max-absolute normalization. The network architecture consists of three convolutional blocks: Conv2D(32, 3×3) + MaxPool(2×2), followed by Conv2D(64, 3×3) + MaxPool(2×2), followed by Conv2D(64, 3×3), then Flatten $\rightarrow$ Dense(64,

ReLU) $\rightarrow$ Dense(1, Sigmoid) for binary classification. Model parameters (convolutional kernels and dense weights) are learned by backpropagation using the Adam optimizer, minimizing the binary cross-entropy objective in (10). Training is performed for 20 epochs with a train/test split of 75%/25%, and validation performance is monitored on the held-out test set.

From an optimization perspective, the CNN training corresponds to minimizing the binary cross-entropy objective in (10) over the network parameters using the Adam optimization algorithm, which performs adaptive gradient-based updates to improve convergence stability. In the RF component, optimization is implicitly achieved through impurity-based split selection during tree construction, where node partitions are chosen to maximize class separation across bootstrapped samples.

Finally, robustness is strengthened through robust training practices, including dropout, L_2 regularization, and controlled data augmentation so that the detector remains stable under noise and other poisoning-relevant perturbations; optimization is performed with Adam under the same binary classification objective. In the experimental evidence summarized in the referenced prevention studies, anomaly filtering is associated with approximately 80% reduction of errors caused by poisoned data, while robust training is reported to increase model accuracy by about 15% under corrupted-data conditions; additionally, adversarial regularization is reported to reach approximately 85% accuracy while reducing the false-positive rate to around 5% in the considered scenarios.

To mitigate potential overfitting, the learning-based models were trained using controlled regularization mechanisms, including dropout and L_2 penalties, together with a fixed training/testing split and validation on unseen samples. The consistent separation between clean and altered samples across stages indicates stable model behavior rather than memorization of training data.

The overall classification process follows a structured multi-stage procedure for assessing the integrity of candidate audio samples. The input signal is first transformed into a time–frequency representation using STFT, followed by statistical screening through anomaly scoring, distribution verification (KS test), and discontinuity analysis. Compact statistical descriptors extracted from the spectrogram are then provided to the Random Forest classifier for lightweight anomaly screening, while the preprocessed spectrogram is analyzed by the CNN for higher-sensitivity validation. A conservative decision rule is finally applied, accepting the sample only if all stages indicate consistency and rejecting it otherwise.

To provide a clearer procedural view of the proposed multi-stage detection framework and its relationship with the associated mathematical formulations, the overall processing flow is summarized in Algorithm 1.

Algorithm 1: Multi-Stage Poisoning Detection Pipeline

Input: Audio sample x
Output: Clean / Altered decision

1. Compute STFT spectrogram S of the input signal.
2. Compute anomaly score using spectrogram deviation.
3. Perform distribution verification using the KS statistic.
4. Compute discontinuity score from waveform differences.
5. If any statistical test indicates anomaly $\rightarrow$ mark the sample as *suspicious*.
6. Extract statistical features (mean, standard deviation, maximum, minimum) from S.
7. Apply the RF classifier using the decision rule.
8. Preprocess the spectrogram (resize to 128×128 and normalize) and apply the CNN classifier.
9. Combine statistical and learning-based decisions using a conservative fusion rule.
10. If all stages are consistent $\rightarrow$ *Clean*, otherwise $\rightarrow$ *Altered*.

---

3. Experimental Evaluation and Results

This section reports an experimental validation of the proposed multi-stage poisoning-detection pipeline introduced in Section 3, using a controlled voice-command dataset and two representative poisoning mechanisms (label manipulation and subtle noise injection). The evaluation is structured to mirror the pipeline stages, so that each result figure (Figs. 4–9 in the supplementary material [8]) directly operationalizes one detection or mitigation component and illustrates how poisoned samples can be progressively isolated before model training.

The evaluation was conducted using the voice-command dataset recorded by the authors under controlled acquisition conditions, as described in Section 2.1. The dataset contains eight command classes and includes both clean recordings and synthetically altered samples generated through controlled perturbation and label manipulation procedures. To evaluate the classification models, the dataset was partitioned into training and testing subsets using a 75%/25% split with a fixed random seed of 42 to ensure reproducibility.

3.1. Stage 1: Signal/spectrogram integrity screening via anomaly scoring

The first stage applies a fast integrity screen based on an STFT spectrogram deviation score, computed as the mean absolute element-wise difference between a reference (clean) spectrogram and a candidate (potentially altered) spectrogram. This screening mechanism is designed to remain lightweight for embedded/edge constraints while retaining sensitivity to systematic poisoning-induced perturbations that manifest in the time-frequency domain. A representative example is retained to illustrate the effect, as shown in Fig. 4 in the supplementary material [8], the altered sample is flagged as anomalous, yielding an STFT-based deviation score of 0.64 for the illustrated "stop" command, which is consistent with the intended use of Stage 1 as a coarse pre-filter that rejects clearly deviant candidates before invoking subsequent (and more computationally expensive) verification stages.

3.2. Stage 2: Distribution-level verification with KS testing

After integrity screening, a statistical verification stage compares candidate and reference signal distributions using the KS test, which quantifies the maximum separation between empirical CDFs. This step is intended to capture poisoning-induced distributional drift even when perturbations are subtle in the raw waveform. Fig. 5 in the supplementary material [8] reports representative outcomes for specific commands: the altered "no" sample yields a statistically decisive deviation with $p \approx 0$, whereas the clean "stop" sample yields $p = 1.00$, indicating no detectable distribution shift under the same procedure. This contrast supports the role of Stage 2 as a low-cost statistical gate that can provide strong evidence of contamination prior to higher-complexity learning-based verification stages.

3.3. Stage 3: Discontinuity-based checks for abrupt feature shifts

To complement distribution testing, a discontinuity-based score is applied to capture abrupt, atypical changes that may arise from poisoning artifacts, such as local waveform irregularities or sudden energy redistribution that is inconsistent with the expected phonetic structure of genuine commands. The underlying motivation is that certain manipulations may not always produce a strong global distributional shift, yet they can still introduce sharp local distortions that remain

detectable through first-order difference behavior. In the revised presentation, Fig. 6 in the supplementary material [8] illustrates this effect using representative commands: the altered "go" sample exhibits a non-zero discontinuity score of $D = 5.45 \times 10^{-2}$, whereas the clean/control "no" sample yields $D = 0$ under the same computation, indicating the absence of the targeted abrupt-change signature. This stage therefore provides an additional structural criterion, orthogonal to distribution matching, that strengthens the pipeline when poisoning manifests primarily as localized irregularity rather than broad statistical drift.

3.4. Learning-based detection: RF screening and CNN verification

Following the statistical stages, learning-based detectors are applied to capture subtle anomalies that may evade classical tests. A lightweight RF classifier operates on compact spectrogram-derived descriptors (mean, standard deviation, minimum, maximum), enabling efficient screening at the device/edge level, while a CNN provides a high-sensitivity verification stage when stronger assurance is required.

In the RF screening demonstration (Fig. 7 in the supplementary material [8]), the classifier operating on low-dimensional spectrogram descriptors (mean, standard deviation, minimum, and maximum) provides a practical separation between clean and altered samples. Specifically, the altered "up" instance is flagged as "Detected Anomaly," while the corresponding clean/control example "yes" is labeled "Detected Normal." Although Fig. 7 in the supplementary material [8] is presented as an illustrative qualitative output rather than a full metric table, it supports the intended role of RF as a lightweight, low-cost discriminator suitable for early-stage screening prior to deeper verification.

The CNN-based stage provides a more decisive separation, as illustrated in Fig. 8 in the supplementary material [8] through explicit detection scores computed on spectrogram inputs. In the representative clean example, the original "up" command is assigned a near-zero anomaly probability (0.00%), whereas the poisoned instance is classified with near-certain confidence, with the altered "yes" sample yielding 99.97%. This large margin between clean and altered scores highlights strong discriminative capacity at the time-frequency level and supports the CNN component as a high-sensitivity validator following earlier low-cost screening stages.

3.5. Robustness-oriented training as mitigation

Beyond detection, robustness-oriented training is treated as a complementary mitigation mechanism intended to reduce model sensitivity to poisoning artifacts by stabilizing learning through regularization, dropout, controlled augmentation, and systematic exposure to both clean and altered patterns during training. In the revised presentation, Fig. 9 in the supplementary material [8] provides a representative example of the resulting prediction behavior: for the altered "no" command, the model produces a predicted probability of 0.51, while the paired clean/original example is presented for direct visual comparison in both waveform and STFT domains. This bounded response is consistent with the intended effect of robustness-oriented training, namely, to dampen perturbation-driven confidence shifts and reduce over-reactive predictions under poisoning-relevant distortions, while leaving the final decision to the combined evidence of the preceding multi-stage detection pipeline.

In the case of voice commands, such a cycle may involve real-time analysis of acoustic characteristics and recording the model's reactions to new data, thus adjusting the training parameters to maintain the robustness of the model even in the face of new types of attacks.

3.6. Limitations

While the proposed multi-stage poisoning detection pipeline demonstrates consistent separation between clean and altered samples under controlled experimental conditions, several limitations should be acknowledged. First, the evaluation was conducted on a controlled dataset with synthetically generated poisoning, which may not fully capture the diversity and adaptive nature of real-world adversarial scenarios. Second, the learning-based components were configured to balance robustness and computational feasibility for embedded deployment rather than to achieve maximum standalone classification performance. Third, the study focuses on representative poisoning mechanisms (label manipulation and additive perturbation), while more advanced and adaptive attack strategies may require further extension of the detection framework. Finally, large-scale statistical validation across multiple datasets and deployment environments remains an important direction for future investigation.

4. Conclusions

This paper addressed the practical risk that training-data poisoning poses to embedded voice-command pipelines, where subtle contamination can either degrade general recognition or induce targeted misclassifications while remaining difficult to notice during routine dataset curation. In contrast to single-layer countermeasures commonly discussed in the literature, where defenses may focus on either statistical drift, trigger reconstruction, or model-level sanitization, the proposed contribution is a multi-stage, deployable screening workflow that aligns with embedded constraints by separating low-cost screening from high-sensitivity verification. The resulting design provides a structured way to reduce attack surface in the training pipeline without assuming that a single detector is sufficient against diverse poisoning strategies reported in recent speech backdoor research.

Experimentally, the multi-stage pipeline separates clean from altered samples using complementary indicators. In Stage 1, STFT-based integrity screening flags altered commands with a representative anomaly score of 0.64 (altered "stop"). In Stage 2, distribution verification produces a decisive split, with $p \approx 0$ for an altered "no" sample versus $p = 1.00$ for a clean "stop" control. In Stage 3, discontinuity analysis captures localized irregularities, yielding $D = 5.45 \times 10^{-2}$ for an altered "go" sample and $D = 0$ for a clean "no" control. The learning-based components further strengthen decision confidence: the RF stage provides practical Normal/Anomaly separation on low-dimensional spectrogram descriptors, while the CNN verifier exhibits a high-margin separation with 0.00% for a clean "up" sample and 99.97% for an altered "yes" sample. Robustness-oriented training complements detection by bounding predictive behavior under perturbation, with a representative altered "no" output of 0.51, consistent with dampened sensitivity to poisoning artifacts.

The pipeline enforces a conservative acceptance principle in which trust is granted only when multiple independent indicators align and is withheld when any stage signals inconsistency. This layered structure improves resilience by reducing reliance on a single detection cue and by increasing the difficulty of sustaining stealth across heterogeneous validation mechanisms.

Future work targets a more operationally realistic validation on a physical embedded platform, with end-to-end acquisition from microphone hardware and evaluation under over-the-air conditions (room acoustics, reverberation, speaker-microphone distance) to assess robustness beyond offline waveform manipulation. A priority direction is to expand the attack suite to cover ad-

ditional poisoning families emphasized in the literature for speech pipelines (e.g., compression-based triggers, latent-space rearrangement, and frequency-offset perturbations), then quantify detection performance under mixed and adaptive attackers. In parallel, the pipeline should be profiled for embedded feasibility, including latency, memory footprint, and energy cost, and extended with secure data provenance mechanisms and automated dataset auditing to support continuous learning scenarios where new audio batches are routinely integrated. These steps would convert the current controlled validation into a deployment-grade assurance workflow suitable for safety-relevant voice-command control in cyber-physical embedded systems.

References

[1] S.-A. DRAGUSIN, N. BIZON, R. N. BOSTINARU and D. TOMA, *Command recognition system using convolutional neural networks*, Proceedings of 2024 16th International Conference on Electronics, Computers and Artificial Intelligence, Iaşi, Romania, 2024, pp. 1–10.

[2] S.-A. DRAGUSIN, N. BIZON, R. N. BOSTINARU, F. M. ENESCU, R. M. TEODORESCU and C. SAVULESCU, *Analysis of vulnerabilities in communication channels using an integrated approach based on machine learning and statistical methods*, Proceedings of 2024 16th International Conference on Electronics, Computers and Artificial Intelligence, Iaşi, Romania, 2024, pp. 1–12.

[3] S.-A. DRAGUSIN, N. BIZON and R. N. BOSTINARU, *Comprehensive analysis of cyber-attack techniques and vulnerabilities in communication channels of embedded systems*, Proceedings of 2024 16th International Conference on Electronics, Computers and Artificial Intelligence, Iaşi, Romania, 2024, pp. 1–12.

[4] B. H. ABBASI, Y. ZHANG, L. ZHANG and S. GAO, *Backdoor attacks and defenses in computer vision domain: A survey*, arXiv preprint, 2025, pp. 1–22.

[5] S.-A. DRAGUSIN, N. BIZON and R. N. BOSTINARU, *A brief overview of current encryption techniques used in embedded systems: Present and future technologies*, Proceedings of 2023 15th International Conference on Electronics, Computers and Artificial Intelligence, Bucharest, Romania, 2023, pp. 1–8.

[6] M. ALJANABI, A. H. OMRAN, M. M. MIJWIL, M. ABOTALEB, E.-S. M. EL-KENAWY, S. Y. MOHAMMED and A. IBRAHIM, *Data poisoning: Issues, challenges, and needs*, IET Conference Proceedings **44**, 2023, pp. 359–363.

[7] J. FAN, Q. YAN, M. LI, G. QU and Y. XIAO, *A survey on data poisoning attacks and defenses*, Proceedings of 2022 7th International Conference on Data Science in Cyberspace, Guilin, China, 2022, pp. 48–55.

[8] S.-A. DRAGUSIN, D. TOMA, R.-N. BOSTINARU, and N. BIZON, *Supplementary material of the paper Sebastian-Alexandru Dragusin, Denisa Toma, Robert-Nicolae Bostinaru and Nicu Bizon, "A Multi-Stage Poisoning Detection Pipeline for Embedded Voice-Command Systems"*, Romanian Journal of Information Science and Technology, 2026. Accessed: Mar. 29, 2026. [Online]. Available: https://jeeeccs.net/sdragusin/Supplementary-material-A-Multi-Stage-Poisoning-Detection-Pipeline-for-Embedded-Voice-Command-Systems.docx.

[9] P. ZHAO, W. ZHU, P. JIAO, D. GAO and O. WU, *Data poisoning in deep learning: A survey*, arXiv preprint, 2025, pp. 1–20.

[10] K. LI, C. BAIRD and D. LIN, *Defend data poisoning attacks on voice authentication*, IEEE Transactions on Dependable and Secure Computing **21**(4), 2023, pp. 1754–1769.

[11] R. N. BOSTINARU, N. BIZON, S.-A. DRAGUSIN, G. V. IANA and D. TOMA, *Dimensionality reduction with principal component analysis for fire and non-fire audio classification: A new approach*,

Proceedings of 2025 17th International Conference on Electronics, Computers and Artificial Intelligence, Târgovişte, Romania, 2025, pp. 1–5.

[12] R. BERNHARD, P. A. MOELLIC and J. M. DUTERTRE, *Impact of low-bitwidth quantization on the adversarial robustness for embedded neural networks*, Proceedings of 2019 2nd International Conference on Cyberworlds, Kyoto, Japan, 2019, pp. 308–315.

[13] N. THEWES, P. STEINHAUER, P. TRAMPERT, M. PAULY and G. SCHNEIDER, *Explainable anomaly detection for sound spectrograms using pooling statistics with quantile differences*, arXiv preprint, 2025, pp. 1–20.

[14] G. GHATAK, H. MOHANTY and A. U. RAHMAN, *Kolmogorov-Smirnov Test-based actively adaptive Thompson sampling for non-stationary bandits*, IEEE Transactions on Artificial Intelligence **3**(1), 2022, pp. 11–19.

[15] M. A. SEMENOVA and D. S. KHALIN, *Research of statistic distributions of nonparametric goodness-of-fit tests by large samples*, Proceedings of 2018 14th International Scientific-Technical Conference on Actual Problems of Electronic Instrument Engineering, Novosibirsk, Russia, 2018, pp. 260–264.

[16] S. W. HO and R. W. YEUNG, *On the discontinuity of the Shannon information measures*, IEEE Transactions on Information Theory **55**(12), 2009, pp. 5362–5374.

[17] R. PRIMARTHA and B. A. TAMA, *Anomaly detection using random forest: A performance revisited*, Proceedings of 2017 4th International Conference on Data and Software Engineering, Palembang, Indonesia, 2017, pp. 1–6.

[18] R. KUMARI, SHEETANSHU, M. K. SINGH, R. JHA and N. K. SINGH, *Anomaly detection in network traffic using K-mean clustering*, Proceedings of 2016 3rd International Conference on Recent Advances in Information Technology, Dhanbad, India, 2016, pp. 387–393.

[19] T. M. THANG and J. KIM, *The anomaly detection by using DBSCAN clustering with multiple parameters*, Proceedings of 2011 26th International Conference on Information Science and Applications, Jeju Island, South Korea, 2011, pp. 1–5.

[20] B. I. HAIRAB, M. SAID ELSAYED, A. D. JURCUT and M. A. AZER, *Anomaly detection based on CNN and regularization techniques against zero-day attacks in IoT networks*, IEEE Access, 2022, pp. 98427–98440.