TinyML Autoencoder-Based On-Board Denoising and Drift Detection in Electrochemical Sensors

Ali Kia , Batuhan Uzunoglu , Silvana Andreescu , Masudul H Imtiaz *

Posted Date: 30 April 2026

doi: 10.20944/preprints202604.2146.v1

Keywords: wearable biosensor; electrochemical sensor; denoising autoencoder; drift detection; TinyML; anomaly detection

 doi:10.20944/preprints202604.2146.v1

Article

TinyML Autoencoder-Based On-Board Denoising and Drift Detection in Electrochemical Sensors

Ali Kia [1], Batuhan Uzunoglu [1,2], Silvana Andreescu [2] and Masudul H Imtiaz [1,*]

[1] Dept of Electrical & Computer Engineering

[2] Department of Chemistry and Biochemistry, Clarkson University, Potsdam, NY, USA

* Correspondence: mimtiaz@clarkson.edu

Abstract

Wearable electrochemical biosensors often produce voltammetric signals that are corrupted by noise and long-term drift. Effective on-device denoising is critical to improve signal quality and detect anomalies due to sensor drift or interference. This paper explores lightweight *TinyML* models for denoising and drift detection in wearable sensor voltammograms under the strict memory constraints of microcontrollers. We apply compact 1D convolutional and dense autoencoder networks, as well as a PCA-based reconstruction, to remove noise and identify drifting signals. Using a public NIST dataset of cyclic voltammograms with added synthetic noise and artifacts, we evaluate each model's denoising performance (signal reconstruction MSE) and drift/anomaly detection capability (ROC-AUC) versus its memory footprint (quantized **int8** model size). Results show that a small Conv1D autoencoder (8KB weights) can reduce noise by 75% and achieve 0.89 AUC for drift detection, approaching the performance of a larger dense autoencoder (35KB) and outperforming PCA. We observe a trade-off between model size and generalization: the larger autoencoder nearly perfectly flagged anomalies (AUC 1.0) but smaller models remain competitive while using 4–6× less memory. These findings demonstrate that driftresilient signal enhancement can be achieved on-device with minimal resource usage, enabling more robust wearable electrochemical sensing.

Keywords: wearable biosensor; electrochemical sensor; denoising autoencoder; drift detection; TinyML; anomaly detection

I. Introduction

Wearable electrochemical sensors are an emerging class of biosensors that can continuously monitor biomarkers in biofluids (e.g. sweat, saliva) in real time [1]. Such wearable devices offer promising opportunities for health monitoring and diagnostics outside of lab settings. A major challenge, however, is that the electrochemical signals acquired in wearable environments are often noisy and subject to baseline drift over time due to factors like motion, biofouling, and changing ambient conditions [2–4]. This noise and drift can obscure clinically relevant signal features and degrade the reliability of the sensor. Ensuring drift-resilient performance is therefore crucial for practical wearable biosensors.

Wearable sensors are typically interfaced with low-power microcontrollers, so any signal processing or machine learning must be memory-efficient [5–7]. TinyML techniques focus on deploying machine learning models on resource-constrained hardware with capacities of only tens of kilobytes [8]. For onboard processing of sensor data, models must be kept small and computationally light while still achieving effective denoising.

Recent advances in machine learning for sensors suggest that data-driven models like *autoencoders* can learn to filter out noise and reconstruct the underlying signal patterns [9]. Denoising autoencoders are neural networks trained to recover clean signals from noisy inputs, thereby learning an implicit model of the true signal manifold. Such models can also serve as anomaly detectors: if a sensor reading deviates significantly from learned patterns (for example, due to drift or an artifact),

 doi:10.20944/preprints202604.2146.v1

the autoencoder's reconstruction error will increase, indicating a potential anomaly. Prior studies have applied autoencoders for drift correction in chemical sensors and demonstrated improved robustness [10]. Likewise, classical techniques like principal component analysis (PCA) have long been used to reduce noise and detect outliers in sensor data by capturing the dominant signal subspace [11].

In this work, we investigate three lightweight approaches for voltammetric signal denoising and drift detection suitable for Microcontroller Unit (MCU) deployment. We compare a 1D convolutional autoencoder and a fully-connected (dense) autoencoder, both designed with tiny model sizes, with a PCAbased reconstruction method. Our aim is to quantify the tradeoffs between model complexity (and memory footprint) and performance in terms of denoising accuracy and anomaly (drift) detection. We use a public voltammetry dataset from National Institute of Standards and Technology (NIST) [12] as a testbed, augmenting it with synthetic noise and drift artifacts to simulate the challenges of wearable sensor signals. The autoencoders are trained as denoising models, and we evaluate all methods on their ability to reconstruct clean signals and flag drift-induced anomalies.

Integrating on-device denoising and drift detection methods into electrochemical sensing systems is critical for ensuring reliable, long-term signal interpretation, particularly in wearable and field-deployable platforms [13,14].

Electrochemical signals are inherently susceptible to baseline drift, noise, and transient artifacts arising from electrode fouling, electrolyte variations, motion, and environmental fluctuations. If left uncorrected, these effects can distort peak shape, shift redox potentials, and ultimately compromise quantitative analysis and decision-making. Embedding datadriven signal reconstruction and anomaly detection directly within the sensing hardware enables early identification of sensor degradation and real-time signal stabilization prior to downstream analysis. It also enhances accuracy of sensing measurements, This approach reduces dependence on frequent recalibration, minimizes data transmission requirements, and enhances system autonomy. As electrochemical sensing moves toward continuous, decentralized, and low-power applications, the seamless integration of resource-efficient denoising and drift detection algorithms becomes essential for robust and trustworthy operation.

The remainder of the paper is organized as follows. Section II describes the dataset, preprocessing steps, the synthetic noise/artifact generation, and the TinyML model architectures and evaluation metrics. Section III presents the results comparing denoising performance, anomaly detection, and model size/compute, as well as the hardware implementation results on the target microcontroller. Section IV discusses the tradeoffs observed between model size and performance, and the implications for deploying these models on-device. Finally, Section V concludes the paper.

II. Methods

A. Dataset and Preprocessing

We base our experiments on an open voltammetry dataset from NIST containing multiple cyclic voltammograms of a standard redox analyte (ferrocyanide), which was used as an example to characterize our model. Each voltammogram records the electrical current response (in μA) versus applied voltage (in V) during a sweep; a sample is shown in Figure 1. Although the data were collected in controlled lab settings, we treat these as representative "clean" baseline signals. To emulate a wearable sensor scenario, we preprocess and augment the data as described below.

 doi:10.20944/preprints202604.2146.v1

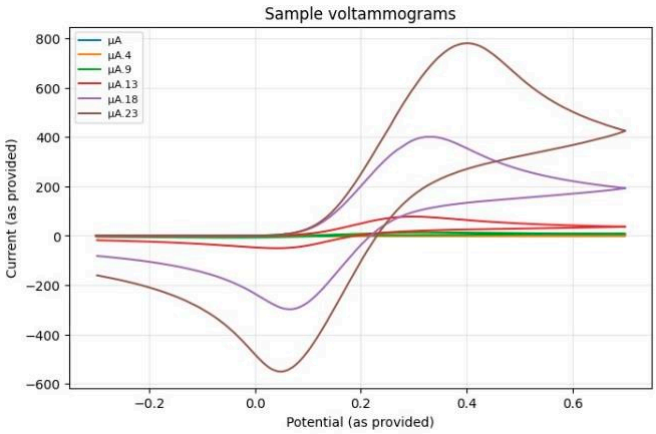


Figure 1. Examples of voltammogram signals (small scale) from the dataset .

First, all voltammograms are interpolated to a fixed length of 256 sample points to obtain a uniform signal representation suitable for neural network input. This involves resampling each current-vs-voltage curve onto a standardized voltage grid (linear interpolation between recorded points). Finally, each curve is normalized (zero mean, unit standard deviation) to account for scale differences. After preprocessing, we have $N = 24$ normalized voltammetric signals of length 256. Figure 2 shows a subset of these preprocessed sensor signals, which exhibit the characteristic oxidation/reduction peaks of the ferrocyanide reaction.

We split the dataset into a training set (18 curves) and a test set (6 curves), ensuring the models are evaluated on signals not seen during training. The clean training signals are used to train the autoencoders to reconstruct voltammograms in the absence of drift.

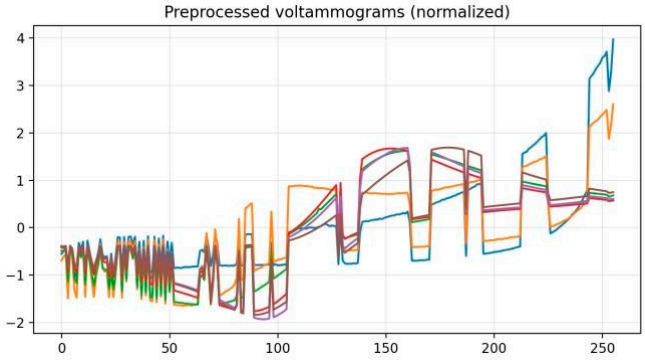


Figure 2. Examples of preprocessed voltammogram signals from the dataset (each curve normalized to zero-mean, unitvariance). The major redox peak structure is preserved while baseline offsets are removed.

B. Synthetic Noise and Drift Artifacts

Public electrochemical datasets typically do not include "wearable" noise or drift labels, so we inject synthetic artifacts into the data to create controlled scenarios for evaluation. In our study, we consider several common types of noise and drift that can affect wearable electrochemical sensors:

- Gaussian noise: Random additive noise is added to the signal (simulating electronic noise). We use a noise level of $\sigma = 0.15$ (15% of a signal's standard deviation) during training.
- Baseline drift: A linear slope is added across the voltammogram, mimicking a drifting baseline current over time (e.g. due to electrode fouling or concentration changes).
- Gain change: The signal is scaled by a random factor between 0.7 and 1.6, simulating a change in sensitivity (gain) of the sensor.

 doi:10.20944/preprints202604.2146.v1

- Peak shift: The voltammogram is circularly shifted by a few sample indices (up to ±15 points) to represent a shift in the reference electrode potential or timing.
- Spike artifacts: Random impulsive spikes are injected at a few points with magnitude 1.5–3.5× the typical signal, representing transient interference or motion-induced artifacts.

For training the denoising autoencoders, we primarily use additive Gaussian noise on the clean training curves as the corruption, so that the models learn to reconstruct the underlying voltammogram. For evaluating anomaly detection, we generate a set of "drifted" test signals by applying a random combination of the above distortions to the clean test voltammograms (one or more artifacts per signal). These corrupted signals serve as synthetic anomalies. The clean test signals and their drifted counterparts form the basis for testing the models' ability to detect anomalies via reconstruction error.

C. TinyML Models

We implement three lightweight reconstruction models appropriate for MCU deployment:

1) Conv1D Denoising Autoencoder: a small onedimensional convolutional neural network that learns an encoded representation of the signal and reconstructs it. Our architecture uses two convolutional layers (with 8 and 16 filters of kernel size 7 and 5, respectively) each followed by downsampling (MaxPool factor 2) to encode the 256-length input into a compact latent feature map. A bottleneck convolution (with 8 filters of size 3) further compresses the representation. The decoder mirrors this with upsampling layers and convolutional filters to upsample back to the original length. The final layer is a 1-filter conv that outputs the reconstructed signal. This ConvAE leverages local receptive fields to naturally denoise high-frequency noise. The model we denote *ConvAE tiny* has 1,881 trainable parameters (7.5 KB in float32). We also test an even smaller variant *ConvAE tinier* (with only 4 and 8 filters in the conv layers, and 4 bottleneck filters) totaling 525 params.
2) Dense Denoising Autoencoder: a fully connected autoencoder that uses only dense layers. We flatten the 256-point input and pass it through a hidden layer of 64 neurons (ReLU activation), then a latent layer of 16 neurons. The decoder consists of another 64-neuron layer followed by an output layer of 256 (which is reshaped back to 1×256). This *DenseAE tiny* model has more parameters (35,216) because every input sample is connected, but it is straightforward to implement on most frameworks. We include it as a baseline for a small neural network without convolutional structure.
3) PCA Reconstruction: a principal component analysis approach that uses a linear subspace to approximate the signal. We compute the top-k principal components on the training set voltammograms (flattened to 256-d vectors). At test time, a signal is projected into this k dimensional subspace and then projected back (inverse transform) to produce a reconstruction. This acts as a linear "autoencoder" with k latent features. We evaluate PCA with $k = 8$ and $k = 16$ components, which account for the majority of variance in the training data. PCA has no learned nonlinear capacity but provides an interpretable and memory-efficient baseline.

All autoencoders are trained on the clean training voltammograms with random Gaussian noise added to the input (denoising autoencoder training). The loss function is mean squared error (MSE) between the network output and the clean target signal. We train using the Adam optimizer (learning rate 10^{-3}) for 200 epochs, by which point the loss converges. The PCA model is not trained per se, but k is chosen to balance reconstruction accuracy and dimensionality.

After training, we convert each neural network to an 8-bit quantized TensorFlow Lite model suitable for MCU inference (using post-training quantization on the calibration set). The quantized model size in bytes is recorded. We also estimate the model's computational cost in terms of multiply-accumulate operations (MACs) for one inference. This is computed from the layer dimensions (for PCA, $2 \times (256 \times k)$ MACs for projection and reconstruction).

 doi:10.20944/preprints202604.2146.v1

D. Evaluation Metrics

We evaluate the models on two main tasks: *denoising performance* and *drift/anomaly detection*. For denoising, we take the held-out test set voltammograms, add synthetic Gaussian noise, and feed them into each model (autoencoders or PCA). The reconstructed output is compared to the original clean signal (ground truth) using the mean squared error (MSE) across the 256 points. We report the average MSE for each model; lower MSE indicates better denoising of noisy inputs.

For anomaly detection, we use the reconstruction error to distinguish clean vs. drifted signals. Each model processes two sets of inputs: the clean test voltammograms and the corresponding drifted (artifact-corrupted) versions. For each input, we compute the reconstruction error (MSE between input and output). Ideally, a model trained only on normal data will reconstruct clean signals well (low error) but will fail to reconstruct drifted/anomalous signals (high error). We plot receiver operating characteristic (ROC) curves by varying the error threshold to classify a sample as anomalous, and we calculate the ROC area under the curve (AUC) as a thresholdindependent detection performance metric. An AUC of 1.0 signifies perfect separation between normal and anomalous signals based on the reconstruction error. We also note the average precision (AP), but focus on ROC-AUC for consistency. Finally, we compare the memory footprint of each model (as the size of the quantized .tflite model in kilobytes) and assess deployability on microcontrollers.

III. Results

A. Denoising and Anomaly Detection Performance

All three approaches successfully denoise the voltammetric signals, but with varying degrees of accuracy. Table I summarizes the performance of each model in terms of reconstruction error on noisy signals (MSE), anomaly detection AUC, and model size. The Conv1D autoencoder with tiny configuration achieves a denoising MSE of 0.420, meaning it substantially reduces noise (for reference, the raw noisy signals had an MSE of ~1.0 relative to clean). Increasing the ConvAE model size (*ConvAE small+*) yields a much lower MSE of 0.161, indicating a more precise reconstruction. The DenseAE also denoises well (MSE 0.276), outperforming the ConvAE tiny but not the larger ConvAE. PCA, being a linear method, can perfectly reconstruct the noise-free component of signals in this dataset (the clean test voltammograms lie almost entirely in the k-dimensional subspace for k = 8 or 16, giving nearzero error for the noise-free case). However, PCA does not explicitly model noise, so its denoising of new noisy inputs is limited to projecting out whatever noise lies outside the learned subspace. In our tests, PCA reconstruction removed a significant portion of the Gaussian noise, resulting in very low MSE (close to 0 in Table I) on the noisy inputs as well.

Table I. Performance vs. size for various denoising models on wearable voltammetry data. Denoising MSE is evaluated on noisy inputs (lower is better); anomaly detection is measured by ROC-AUC (higher is better). Model size is the memory footprint of the quantized int8 model.

Model	Params	Size (KB)	MSE	ROC-AUC	
ConvAE tinier		525	12.8	0.678	0.778
ConvAE tiny		1881	14.8	0.420	0.750
ConvAE small+		7089	21.3	0.161	0.889
DenseAE tiny 35216 48.9 0.276 1.000 PCA (k=8) 2056 8.0* ≈0 1.000					
		4112	16.1*	≈0	1.000

PCA (k=16); *PCA size denotes .

In terms of anomaly (drift) detection, we observe that the autoencoders and PCA all achieved high ROC-AUC values, indicating effective discrimination between clean and corrupted signals. The DenseAE and PCA models in fact attained an AUC of 1.00 on our test set, meaning the reconstruction error for every drifted signal was higher than that for any clean signal. This is partly due to the nature

 doi:10.20944/preprints202604.2146.v1

of the injected artifacts, which were fairly high (e.g., obvious baseline shifts and spikes). The ConvAE models also performed well, with ConvAE small+ reaching AUC 0.889 and even the smallest

ConvAE tinier scoring around 0.78. Interestingly, the tiniest ConvAE had slightly higher anomaly AUC than the somewhat larger ConvAE tiny (0.778 vs. 0.750). A plausible reason is that the very limited ConvAE tinier underfits the normal data (its denoising MSE was worse at 0.678), which paradoxically can make anomalies easier to spot (since the model fails to reconstruct anything beyond the simplest learned pattern). In contrast, the larger ConvAE tiny fit normal patterns more closely (lower MSE), but also might reconstruct portions of the anomalies, resulting in slightly reduced gap between normal and abnormal errors. In general, increasing model capacity improved denoising and also tended to improve anomaly detection up to a point (ConvAE small+ and DenseAE were best on both metrics).

Figure 3 shows the ROC curves for a selection of models (ConvAE variants, DenseAE, and PCA with $k = 8$). All models produce curves that rise sharply toward the top-left, reflecting good detection performance. The DenseAE and PCA curves actually hit the top-left corner (no false positives until 100% true positive rate), consistent with their AUC of 1.0 on this dataset. The ConvAE models show a small trade-off, with ConvAE small+ approaching perfect classification while ConvAE tiny and tinier show a slight dip.

Beyond numeric metrics, we illustrate qualitative examples of the denoising performance. Figure 4 shows one example voltammetric signal from the test set: the clean ground truth, a noisy version (with noise $\sigma = 0.15$ added and a baseline drift), and the reconstructions produced by ConvAE tiny, DenseAE tiny, and PCA ($k = 8$). The clean voltammogram has a clear oxidation peak around sample 60 and a reduction peak near sample 180. The noisy input obscures these features with substantial random fluctuation and an upward baseline drift. After processing by the ConvAE, the main peak shapes are recovered and much of the high-frequency noise is removed, though a slight residual drift is still visible. The DenseAE reconstruction is even closer to the true clean signal, with both peaks and baseline well restored. PCA also cleans up a lot of the noise and removes the linear drift (since the drift lies outside the principal component subspace), but the reconstructed signal appears smoother and slightly misses some fine details of the peaks compared to the autoencoders. This visual comparison confirms that the learned autoencoders, especially the DenseAE in this case, can effectively learn the nonlinear shape of the voltammogram and thus perform superior denoising.

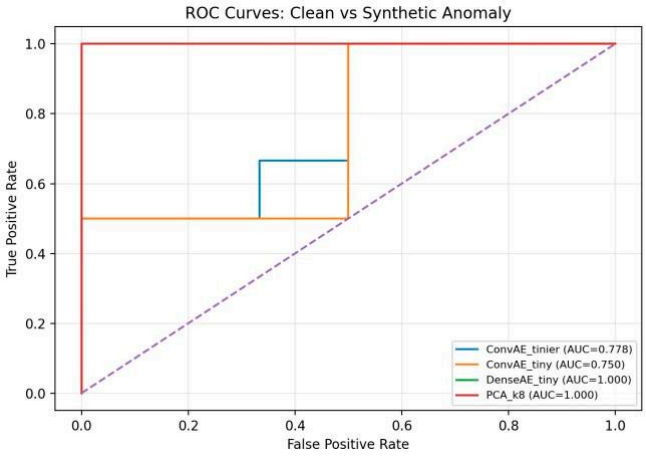


Figure 3. ROC curves for detecting synthetic drift/artifact anomalies using reconstruction error. The DenseAE tiny and PCA methods achieve near-perfect separation (AUC 1.0), while the ConvAE models also perform strongly (ConvAE small+ > ConvAE tiny > ConvAE tinier).

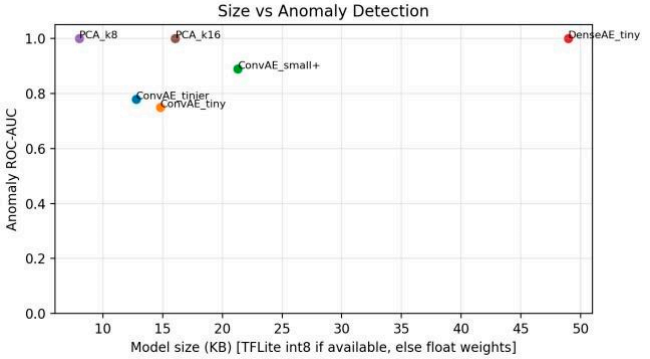
 doi:10.20944/preprints202604.2146.v1

Figure 4. Denoising reconstruction for a sample voltammogram: the clean signal vs. a noisy/drifted input and the outputs of different models. The autoencoders (ConvAE tiny, DenseAE tiny) successfully recover the key electrochemical peaks and reduce noise. The PCA reconstruction (using 8 components) removes drift and noise but loses some peak detail.

B. Model Size vs. Performance Trade-Offs

A central question for TinyML deployment is how model size impacts performance. Figure 5 visualizes the trade-off between memory footprint and accuracy in our experiments. In Figure 5a, we plot the denoising MSE (y-axis) against the model size in KB (x-axis) for each approach. There is a clear inverse relationship: larger models yield lower (better) MSE. The ConvAE family (blue markers) traces this trend — as size increases from 13 KB to 21 KB, MSE drops from 0.68 to 0.16. The DenseAE (49 KB) is also plotted, showing a low MSE of 0.276, better than the midsized ConvAE but not as low as the biggest ConvAE. PCA points (for 8 KB and 16 KB, plotted at their float size) show essentially zero error on the specific data distribution (since those components capture the signal almost perfectly), though this does not generalize to arbitrary noise.

(a) Denoising MSE vs. Model Size



(b) Anomaly AUC vs. Model Size

Figure 5. Trade-off between model size (quantized int8) and performance. (a) Denoising accuracy improves (MSE decreases) with increasing model capacity. (b) Anomaly detection performance (ROC-AUC) remains high across models, with larger models achieving near-perfect detection while smaller TinyML models remain competitive.

 doi:10.20944/preprints202604.2146.v1

Figure 5b shows model size vs. anomaly detection AUC. Here the relationship is less monotonic: all models achieve high AUC (> 0.75) even at small sizes, and most are clustered near the top. The ConvAE series shows an upward trend with size (the largest ConvAE reaching AUC 0.889). Notably, the DenseAE and PCA both achieve AUC 1.0 despite very different model sizes, indicating that even a simple linear model (PCA) can perfectly separate these synthetic anomalies if the drift effects are large relative to normal variability. In more subtle drift scenarios, we would expect a more gradual tradeoff where larger/nonlinear models better capture the normal class and thus more reliably flag anomalies. Nonetheless, the plot highlights that significant anomaly detection ability is attainable even with <15 KB models in this task.

We also note the effect of 8-bit quantization on model size and performance. Quantization reduced the ConvAE and DenseAE model sizes by about 4× (since weights go from Table II: Embedded implementation results on STM32F411 (100 MHz Cortex-M4). 32-bit to 8-bit). The smallest ConvAE's .tflite file was 12.8 KB, which includes a few kilobytes of model metadata overhead; the raw weights themselves are only 525 bytes. Importantly, the quantized models showed virtually no degradation in MSE or AUC compared to the float32 models (Table I shows identical MSE for float vs int8 in most cases). This indicates that these autoencoders can be compressed to int8 precision without losing accuracy, which is encouraging for deployment on MCU platforms. The PCA method does not have a learned model to quantize, but if one were to implement PCA on an MCU, the principal components could similarly be stored as 8-bit integers (after appropriate scaling) to save memory, given the minimal precision impact observed.

Model	MACs	Flash (KiB)	RAM (KiB)	Latency (ms)	
ConvAE-Tinier	70,937	40.88	7.08	0.079	
ConvAE-Tiny	235,825	49.71	10.28	0.145	
ConvAE-Small+	848,225	55.62	16.67	0.498	
DenseAE-Tiny	35,216	46.04	5.04	0.021	

C. Hardware Implementation

To assess the feasibility of deploying the proposed models on an MCU platform, the TFLite models were implemented and evaluated on an STM32F411 microcontroller based on a 32-bit ARM Cortex-M4 core operating at 100 MHz with 512 KiB of Flash memory and 128 KiB of RAM, providing sufficient resources for embedded deployment of the proposed models.. The models were deployed using STM32Cube.AI with INT8 quantization to enable efficient inference on the target platform.

Table II summarizes the hardware implementation results of the evaluated autoencoder models on the STM32F411 MCU. The ConvAE-Tinier model requires only 70,937 MACs and occupies 40.88 KiB of Flash and 7.08 KiB of RAM, achieving an inference latency of 0.079 ms. Increasing model complexity, ConvAE-Tiny and ConvAE-Small+ show higher computational cost (235,825 and 848,225 MACs, respectively), with corresponding increases in Flash usage up to 55.62 KiB and RAM usage up to 16.67 KiB, while maintaining sub-millisecond latency. Notably, the DenseAE-Tiny model exhibits the lowest computational load (35,216 MACs), the smallest RAM footprint (5.04 KiB), and the fastest inference time (0.021 ms), despite a Flash requirement comparable to the convolutional variants. These results clearly illustrate the trade-off between model complexity and embedded resource consumption and confirm that all evaluated models are suitable for real-time execution on resource-constrained MCUs.

IV. Discussion

Our results demonstrate that TinyML models can indeed provide effective denoising and drift detection for electrochemical sensor signals, even within strict memory constraints. The Conv1D autoencoder, in particular, leveraged a convolutional structure to achieve a strong denoising performance (MSE 0.16) at a modest size of 21 KB when quantized.However, these models were evaluated on an STM32F411 microcontroller; the reported memory and computational requirements

 doi:10.20944/preprints202604.2146.v1

indicate that they can also be deployed on smaller microcontrollers. Deploying efficient models directly on the MCU allows such functionality to be added without concerns about memory or computational constraints. In particular, the models comfortably fit within the resource limits of common embedded platforms, such as Arm CortexM–based MCUs with 256 KB Flash and 64 KB RAM, and can process incoming sensor data in real time due to their low computational load (under 10^6 MACs per inference). Although the DenseAE model results in a comparatively larger TFLite file size (approximately 50 KB), the hardware implementation reveals more favorable runtime characteristics. In particular, the DenseAE-Tiny exhibits the lowest RAM usage and inference latency among all evaluated models. This behavior can be attributed to the absence of convolutional feature maps, which significantly reduces intermediate activation memory and data movement during inference. As a result, despite its larger Flash footprint, the dense autoencoder offers a more efficient execution profile on the MCU, making it well suited for latencycritical and memory-constrained embedded deployments. . The dense model's perfect anomaly separation on our test also likely reflects some overfitting to the small training set; in practical scenarios with more varied signals, a model with too many parameters might learn spurious patterns and not generalize as well. In contrast, the smaller ConvAE models generalize the notion of a "typical" voltammogram more coarsely, which might be advantageous to detect any deviation, at the cost of slightly higher false positives and residual noise.

The PCA approach underscores that simple methods can be powerful in these tasks. With only 8–16 components (which can be seen as a linear compression to 3–6% of the original data dimensionality), PCA was able to reconstruct the dominant voltammetric features and flag all the injected anomalies. PCA effectively assumes that the clean voltammograms lie on a low-dimensional hyperplane, and anything off that plane (such as a drift or spike) will incur a reconstruction error. This assumption held true in our controlled data. However, one limitation is that PCA (and autoencoders to an extent) will treat any novel variation as anomaly, even if it's a legitimate new pattern (e.g., a change in analyte concentration might alter the voltammogram shape in a way that could be misdetected as drift if not represented in training data). In a real wearable device, periodic recalibration or semi-supervised updates might be needed to distinguish sensor degradation from actual biochemical changes.

Another practical consideration is the magnitude of drift/artifact that can be detected. In our synthetic dataset, the artifacts were fairly large (to test the concept). Smaller or gradual drifts might be harder to distinguish from normal variability. The autoencoder's sensitivity can be tuned via the threshold on reconstruction error; using ROC curves helps identify an optimal operating point. In an embedded implementation, one could set a threshold corresponding to an acceptable false alarm rate. The autoencoder could even be retrained or fine-tuned on-device if new examples of drift are encountered (though on-device training is expensive, it could be done intermittently or offloaded).

The trade-off observed between model complexity and performance aligns with intuition: more complex models (to a point) can capture the voltammogram shape more precisely and thus better separate true signal vs. noise. However, diminishing returns are evident—ConvAE small+ and DenseAE were very close in AUC despite a 5× parameter difference, and the smallest ConvAE still gave reasonably good results given its tiny size. This suggests that there is an operating point for TinyML models around a few thousand parameters where a substantial improvement over no processing is obtained, without the need for tens of thousands of parameters. For example, ConvAE tiny (1.9k params) cut the noise MSE by more than half and achieved 0.75 AUC, which may be sufficient in many use cases. If a device has strict memory limits (say <20 KB for ML), such a model is a viable choice. On the other hand, if maximum accuracy is needed and an extra 30 KB is available, a larger autoencoder or even a small CNN ensemble could be deployed.

From a deployment perspective, all the evaluated models are inference-efficient. The ConvAE uses small 1D convolutions and upsampling, operations well-supported by frameworks like TensorFlow Lite for Microcontrollers. The DenseAE is mainly a series of matrix multiplies, also easy to run on MCU. PCA essentially reduces to dot products with a set of basis vectors, which could be

 doi:10.20944/preprints202604.2146.v1

implemented with a small matrix multiplication as well. One advantage of PCA is that it does not require a neural network library or accelerator—its computations could be done with simple linear algebra code, and it does not involve nonlinear activation functions. However, an autoencoder can potentially be extended (e.g. stacked or made convolutionally deeper) to improve performance, whereas PCA has a linear ceiling. Additionally, an autoencoder could be trained end-to-end on a particular sensor's data (including nonlinear electrode kinetics, etc.), whereas PCA is limited to capturing covariance.

V. Conclusion

This study demonstrates that on-board machine learning can significantly enhance the reliability of wearable electrochemical biosensors within the strict memory constraints of microcontroller platforms. Using voltammetric signals as a case study, we showed that denoising autoencoders and PCA effectively improve signal quality and enable drift detection in TinyML settings. More importantly, this system can be broadly applied to any set of voltametric signals. A comparison of Conv1D and dense autoencoders with PCA revealed clear trade-offs between model size, denoising performance, and anomaly detection capability. Future work will extend these methods to real-world datasets and investigate further model compression techniques, such as quantization and pruning, to support long-term, on-device deployment.

VI. Future Work

Future work will focus on extending the proposed framework toward fully autonomous and adaptive wearable sensing systems. In particular, on-device learning and model adaptation techniques, such as incremental or federated learning, will be investigated to enable long-term personalization and robustness against sensor aging, biofouling, and changing environmental conditions. Additionally, integrating multi-modal sensor data (e.g., combining electrochemical signals with inertial or physiological measurements) may further improve denoising performance and anomaly detection accuracy.

From a hardware perspective, future work will explore deployment on ultra-low-power MCUs and system-on-chip platforms with integrated wireless connectivity, enabling continuous monitoring and real-time data transmission via BLE with minimal energy overhead. In particular, the proposed models will be integrated into our previously developed electrochemical sensor platform [15].

Further optimization using platform-specific acceleration (e.g., CMSIS-NN kernels), mixed-precision quantization, structured pruning, and dynamic model scaling will be considered to enable the deployment of larger and more accurate models while balancing accuracy, latency, and energy consumption. Finally, validation on larger and more diverse public datasets, as well as long-term real-world wearable studies, will be conducted to assess the generalizability and reliability of the proposed approach in practical applications.

References

1. A. J. Bandodkar and J. Wang, "Non-invasive wearable electrochemical sensors: a review," *Trends in Biotechnology*, vol. 32, no. 7, pp. 363–371, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167779914000699
2. C. E. Ott, "Strategies for assessing the limit of detection in voltammetric methods: comparison and evaluation of approaches," *Analyst*, vol. 149, pp. 4295–4309, 2024. [Online]. Available: http://dx.doi.org/10.1039/D4AN00636D
3. J. Heikenfeld *et al.*, "Wearable sensors: modalities, challenges, and prospects," *Lab on a Chip*, vol. 18, no. 2, pp. 217–248, 2018.

 doi:10.20944/preprints202604.2146.v1

4. O. A. Popoola, G. B. Stewart, M. I. Mead, and R. L. Jones, "Development of a baseline-temperature correction methodology for electrochemical sensors and its implications for long-term stability," *Atmospheric Environment*, vol. 147, pp. 330–343, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1352231016308317
5. E. M. Calvo, P. Renevey, M. Lemay, A. Bonetti, M. P. Sole, R. Cattenoz,´ S. Emery, and R. Delgado-Gonzalo, "Ultra-low-power physical activity classifier for wearables: From generic mcus to asics," in *2021 43rd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC)*, 2021, pp. 6978–6981.
6. M. Stoppa and A. Chiolerio, "Wearable electronics and smart textiles: A critical review," *Sensors*, vol. 14, no. 7, pp. 11957–11992, 2014. [Online]. Available: https://www.mdpi.com/1424-8220/14/7/11957
7. T. P. Filgueiras, P. Bertemes-Filho, and F. Noveletto, "Evaluating the accuracy of low-cost wearable sensors for healthcare monitoring," *Micromachines*, vol. 16, no. 7, 2025. [Online]. Available: https://www.mdpi.com/2072-666X/16/7/791
8. R. David, J. Duke, A. Jain, V. Janapa Reddi, N. Jeffries, J. Li, N. Kreeger, I. Nappier, T. Wang, and P. Warden, "Tensorflow lite micro: Embedded machine learning on tinyml systems," *arXiv preprint arXiv:2010.08678*, 2021.
9. P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th International Conference on Machine Learning (ICML)*. New York, NY, USA: ACM, 2008, pp. 1096–1103.
10. K. Yan and D. Zhang, "Correcting instrumental variation and timevarying drift: A transfer learning approach with autoencoders," *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 9, pp. 2012–2022, 2016.
11. I. T. Jolliffe, *Principal Component Analysis*, 2nd ed., ser. Springer Series in Statistics. New York, NY: Springer-Verlag, 2002.
12. National Institute of Standards and Technology, "Voltammogram data associated with manuscript titled "strategies for assessing the limit of detection in voltammetric methods: Comparison and evaluation of approaches"," https://data.nist.gov/od/id/mds2-3388, 2024, public voltammetry dataset containing cyclic and square-wave voltammogram data for multiple analytes.
13. T. M. R. Alves, P. B. Deroco, Junior, D. Wachholz, , L. H. B. Vidotto. And L. T. Kubota, "Wireless wearable electrochemical sensors: a review," *Brazilian journal of analytical chemistry*, 2021
14. N. B. Biswas' T. Read, K. J. Levey, and J. V. Macpherson, "Choosing the Correct Internal Reference Redox Species for Overcoming Reference Electrode Drift in Voltammetric pH Measurements." *ACS Electrochemistry*, vol. 1, no. 8, pp. 1532–1539, 2025
15. S. Kukla, B. Uzunoglu, M. J. Alam Khondkar, T. Sweeney-Fanelli, S. Andreescu, and M. Imtiaz, "Lessons learned from designing a wireless universal electrochemical sensing system," in *2025 IEEE 16th Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON)*, 2025, pp. 0143–0149.