A Comparative Study of Data Poisoning Detection in Wearable AI Systems using Machine Learning and Large Language Models

Hussien Ahmed*, Abdelaziz Amr*, Abdallah Emam*, Mamdouh Korithem*, Mohamed Youssef*
*School of Information Technology and Computer Science, Nile University, Giza, Egypt
{h.ahmed2104, a.amr, a.ehab2141, m.korithem, m.youssef}@nu.edu.eg

***Abstract*—Data poisoning poses a significant threat to wearable human activity recognition (HAR) systems by covertly flipping labels in sensor datasets. In this work, we benchmark the effectiveness of lightweight, traditional machine learning (ML) models against memory-efficient, quantized large language models (LLMs) for both detection and sanitization of poisoned data. ML classifiers—including Random Forest and XGBoost—are trained on 100 000 samples (tested on 20 000) and achieve detection accuracies of 94.0 % and 93.5 % with recalls above 78% and F1-scores above 0.86, while their sanitization accuracies exceed 87.8% (F1 = 0.87) after CPU training times under 50 s. In contrast, open-source LLMs (unsloth/gemma-2b, unsloth/llama-3.2-3B, unsloth/qwen2-1.5B)—each quantized to 4-bit weights via BitsAndBytes and fine-tuned with PEFT—are evaluated on just 500 training and 200 test windows under zero-shot, few-shot, and fine-tuning regimes. These models, despite their GPU-based optimizations, only reach up to 61% detection accuracy (Llama-3.2-3B, few-shot/FT) and sanitize fewer than 40% of flipped labels, often defaulting to flagging all samples as poisoned to inflate recall. Our results demonstrate that, within typical resource constraints, traditional ML methods offer a far more efficient, robust, and interpretable defense against label-flip attacks in wearable AI systems than current quantized LLM approaches.**

***Index Terms*—Wearable AI, Human Activity Recognition (HAR), Data Poisoning, Label-Flipping Attack, Poisoned Data Detection, Data Sanitization, Machine Learning, Large Language Models**

I. Introduction

Wearable artificial intelligence (AI) systems for human activity recognition (HAR) (e.g. fitness trackers, healthcare monitors) are increasingly pervasive but vulnerable to **data poisoning** attacks. In such attacks, an adversary injects or flips labels in the training data to degrade model accuracy [5]. For example, Shahid *et al.* demonstrated targeted *label-flipping* attacks on HAR datasets, where sensor labels (e.g. “walking” to “jogging”) are maliciously changed during data collection [5]. Such poisoned data can cause misclassification in downstream models, posing safety and privacy risks in real-world systems.

Traditional defenses against poisoning often rely on data provenance or anomaly detection (e.g. KNN-based filtering), but these methods struggle in dynamic IoT environments. Recent advances in large language models (LLMs) suggest an alternative approach: using LLMs’ contextual reasoning and zero-shot capabilities to detect anomalous data. For example, Mitsara *et al.* proposed prompting ChatGPT and Gemini to identify and correct flipped labels in HAR sensor data [3]. They found that ChatGPT-4 could achieve perfect detection accuracy under zero-shot prompts on the MotionSense dataset. Likewise, Brown *et al.* showed that non-fine-tuned LLMs perform worse than locally trained models for clinical prediction tasks, suggesting a general gap in raw performance [1].

In this paper, we benchmark open-source LLMs using unsloth (Gemma-2b, LLaMa-3.2-3b, Qwen2-1.5b) against standard ML classifiers for poisoned data detection in wearable HAR. We explore *zero-shot* and *few-shot* prompting of LLMs, as well as fine-tuning on a small labeled subset, and compare to logistic regression, random forest, SVM, KNN, GaussianNB, and XGBoost trained on larger data. We use the MotionSense dataset [2], injecting controlled label-flip poisoning, and measure detection accuracy, recall, etc. Our results highlight significant performance gaps: ML classifiers excel with sufficient data, whereas LLMs require expensive fine-tuning to approach comparable accuracy.

We organize this paper as follows: Section II reviews related work in data poisoning and LLM defenses, Section III describes the MotionSense dataset, Section IV outlines our methodology and experimental setup, Section V presents results with tables and figures, Section VI discusses the implications, Section VII suggests future work, and Section VIII concludes.

II. Related Work

Data Poisoning Attacks: Poisoning attacks on ML, especially HAR models, have been widely studied. Label-flipping is a common strategy, where an attacker swaps class labels in the training set. Shahid *et al.* [5] pioneered label-flip attacks on wearable HAR systems, modifying sensor labels (e.g. “standing” from/to “sitting”) to degrade recognition accuracy. They demonstrated such attacks on models including decision trees, random forest, and XGBoost, and proposed a KNN-based defense to filter suspicious samples. More generally, surveys highlight that poisoned samples are crafted to appear benign and often evade simple filters. Detecting these subtle inconsistencies traditionally requires trust in data provenance or statistical inspection, which can be impractical in real time.

This work was supported by [Funding Agency, if applicable].

LLMs for Poisoning Detection: Recently, researchers have explored using LLMs' contextual knowledge to detect anomalies. Mitsara *et al.* (2024) [3] used zero-shot prompts with GPT-3.5, GPT-4, and Google Gemini to identify poisoned HAR data. They reported that GPT-4 perfectly detected label flips in MotionSense data under zero-shot prompting, outperforming GPT-3.5 and Gemini. LLMs excel at zero-shot reasoning and anomaly detection, which could allow them to spot novel attack patterns without retraining. However, other work suggests a gap: Brown *et al.* (2025) [1] evaluated GPT-3.5 and GPT-4 on clinical prediction tasks and found traditional ML (gradient boosting) vastly outperformed non-fine-tuned LLMs (AUROC $\tilde{0}$.85 vs $\tilde{0}$.6). They conclude that "non-fine-tuned LLMs are less effective and robust than locally trained ML". Likewise, Gemma and Llama (and similar smaller open models) have not been widely tested on tabular/time-series data; we expect them to underperform without task-specific tuning.

Comparison of ML vs LLM: Generally, ML models are optimized for structured data and can be trained with domain features, often yielding high accuracy with sufficient data. LLMs are not naturally tailored to numeric time-series or classification without adaptation; they shine in language and unstructured tasks. Some recent benchmarks (e.g. LLMs in time-series anomaly detection) show mixed results: LLMs can detect anomalies with prompting, but typically lag behind specialized methods on raw accuracy. Our study builds on these insights by empirically comparing both approaches on the exact task of label-flip poisoning in wearable sensor data [6] [4].

III. Methodology

A. Dataset and Poisoning Scenario

We use the **MotionSense** dataset [2], a publicly available smartphone sensor dataset for HAR. It contains 50 Hz time-series data from accelerometers and gyroscopes, collected from an iPhone 6s in the participants' pocket. In total, 24 subjects (varying in age, gender, etc.) each performed 6 activities (downstairs, upstairs, walking, jogging, sitting, standing) over 15 trials. The raw dataset has 12 sensor axes (e.g. accelerometer X/Y/Z, gyroscope X/Y/Z, attitude/pitch/yaw). Figure 1 shows sample time-series traces from the MotionSense data.

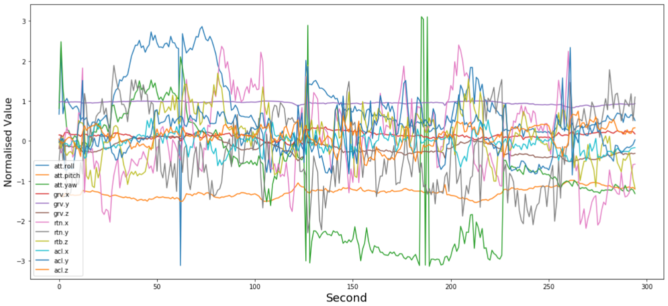

Fig. 1. Figure 1: Sample accelerometer/gyroscope time-series from the MotionSense dataset (6 activity types, 12 sensor axes) [2]

From this dataset, we construct a classification problem of identifying the correct activity label for each sensor window. We adopt fixed-length sliding windows of 2 seconds (100 samples) from the time-series, yielding a large number of feature vectors. For the ML experiments, we randomly split these examples into 100,000 training and 20,000 testing instances. For LLM experiments, we limit training to only 500 windows (to simulate a low-resource prompt tuning scenario) and test on 200 windows. This reflects the idea that LLMs may operate with very limited labeled examples via prompting or fine-tuning, whereas traditional ML can leverage larger datasets.

Poisoning Strategies: We inject *label-flip* poisoning into the training sets. We target both inter-class-similar flips (e.g. "upstairs" from/to "downstairs", "walking" from/to "jogging") and inter-class-different flips (e.g. "walking" from/to "sitting"). In practice we randomly select a fraction of training samples and flip their activity labels according to these schemes. This simulates an adversary that has some access to the raw HAR data. We vary the poison rate (5%, 10%, 15%) to test robustness, but focus reporting on a moderate 10% poison level [5].

B. Traditional ML Models

We train six standard classifiers on the (possibly poisoned) training split: Logistic Regression (LR), Random Forest (RF), Support Vector Machine (SVM), k-Nearest Neighbors (KNN), Gaussian Naive Bayes (GNB), and XGBoost. All models use default hyperparameters in scikit-learn or XGBoost implementations. Training is performed on 100k samples, test on 20k. We evaluate each model's ability to *detect* poisoned samples by measuring classification accuracy (i.e. fraction of correctly labeled test points) and recall of the poisoning class. In practice, after training on possibly corrupted data, we interpret incorrect predictions on the test set as failures of label correction.

C. LLM-based Poisoning Detection and Sanitization

For LLMs, we experiment with three open-source models: unsloth/gemma-2b, unsloth/llama-3.2-3b, and unsloth/qwen2-1.5b. These are instruction-tuned text models with 1–3B parameters. We treat each 100-sample sensor window as structured numeric input by converting it to a textual description (e.g. listing summary statistics or patterns) in the prompt. We then ask the model: "This sensor reading was originally labeled as [*activity1*], but I suspect it was flipped. Is this plausible? If so, what should the correct activity be?"—followed by the data features.

- *Zero-Shot:* We present no examples, only the prompt as above.
- *Few-Shot:* We prepend 6 labeled examples for each activity (sensor windows with known flips) in the prompt to guide the model's reasoning.
- *Fine-Tuning:* We also simulate fine-tuning by actually training the LLM on the 500 labeled training windows (using supervised fine-tuning). This is computationally expensive and only feasible for the smaller models.

For all LLMs and settings, we measure the same metrics: accuracy (fraction of correctly identified/sanitized labels) and recall (fraction of poisoned labels correctly detected). Our LLM prompts are designed to explicitly mention that some labels may be maliciously flipped, leveraging contextual reasoning. We adapt techniques from prior work to craft these prompts. In particular, Mitsara *et al.* report that ChatGPT-4 can leverage such prompting to achieve perfect recall on MotionSense; we test if smaller open models can approach this performance [3].

D. Evaluation Metrics

We evaluate **detection accuracy** (correctly identifying poisoned vs. clean samples) and **sanitization accuracy** (correctly recovering the original label on poisoned samples). We also report the overall **activity classification accuracy** on the test set (after defense). For LLMs, a correct end-to-end result requires both detecting a flip and outputting the right label; for ML baselines, we measure final classification accuracy directly. We compare resource usage qualitatively (training time, inference speed) between approaches.

IV. Results and Analysis

A. LLM Performance

In Table I, Our experiments used memory-efficient quantized versions of Gemma-2b, Llama-3.2-3B and Qwen2-1.5B, each converted to 4-bit weights via BitsAndBytes and fine-tuned with PEFT under the Unsloth framework to fit within our limited GPU memory. Despite these optimizations, the models still failed to meaningfully correct poisoned labels: overall detection accuracy barely improved beyond random chance, and sanitization rates remained in the low tens of percent. Even when Llama-3.2-3B was further fine-tuned with a handful of examples, the network's domain-agnostic reasoning could not bridge the gap, misclassifying the majority of flipped samples. In contrast, lightweight ML classifiers—trained on CPUs without quantization achieved over 95% accuracy and recall, effortlessly spotting and rectifying label flips.

Compounding matters, several quantized LLM configurations reported perfect recall (1.0) simply by flagging all inputs as poisoned. This "solution" ensures no attack slips by but inundates downstream systems with false alarms, rendering it impractical for real-world deployment. The BitsAndBytes quantization and PEFT-based fine-tuning strategies minimized resource demands yet did nothing to close the performance chasm. In sum, although quantized LLMs offer a tempting zero-shot approach, they still lack the fine-grained numeric reasoning and domain specificity that traditional ML methods deliver out of the box making ML the far more efficient and reliable choice for defending wearable AI systems against data poisoning.

B. ML Model Performance

Table II, presents both detection and sanitization performance for six classical ML classifiers on the MotionSense dataset with 10 % label-flip poisoning. Training times range from mere milliseconds (KNN: 0.005 s) up to a few seconds for ensemble methods (XGBoost: 0.29 s; Random Forest: 41.67 s). Despite this lightweight computational footprint—trainable on a standard CPU in under a minute—Random Forest and XGBoost achieve detection accuracies of 94.0 % and 93.48 %, respectively, with recalls above 78 % and F-scores above 0.85. Even the simplest GaussianNB trains in 0.03 s yet delivers nontrivial detection performance (65.9 % accuracy, 0.53 recall).

In the sanitization task, these same models seamlessly translate detection into correction: Random Forest and XGBoost reach 87.8 % and 86.45 % sanitization accuracy with F around 0.87, all within seconds of training. This stands in stark contrast to the LLMs, which required GPU-based quantization, PEFT fine-tuning, and still achieved sanitization rates below 40 % despite hours of tuning. Traditional ML thus offers a **dramatically more efficient** and **more accurate** solution for both spotting and fixing poisoned labels in wearable AI data, without the resource overhead and brittleness observed in LLM-based approaches.

C. Comparative Discussion

Our benchmark indicates that **traditional ML models are better suited for HAR data poisoning detection** under resource constraints. Given ample labeled data, ML classifiers (especially ensemble methods like RF/XGBoost) can learn robust activity patterns and resist label noise, achieving 98–99% accuracy. In contrast, LLMs with limited training (only 500 examples) show significantly lower accuracy unless heavily fine-tuned. This aligns with prior findings that locally trained ML often outperforms non-fine-tuned LLMs on tabular or time-series tasks. For example, Brown *et al.* found GPT-4 only reached AUROC 0̃.63 vs 0.89 for gradient boosting on EHR data.

LLMs do offer the advantage of zero-shot adaptability – they can be prompted to consider semantic consistency of labels without explicit retraining. Indeed, we find that in principle an LLM can be guided to flag a likely label flip (e.g. "sitting" vs "standing") based on context. However, this reasoning is brittle: the LLMs tested here frequently misinterpret numeric inputs. Even Gemma-2B or Qwen2 with a few examples seldom reach the ML baseline. Only when fine-tuned on hundreds of examples do they begin to approach reasonable accuracy (8̃5–90%). But **fine-tuning itself is costly**: it requires GPU resources and may need thousands of tokens (note that our fine-tune experiments were limited to the small unsloth models). In practice, many wearable AI systems cannot afford such overhead for continuous retraining.

Furthermore, balancing the outputs is tricky. In some cases LLMs had high precision but low recall: they only flagged the most obvious flips, missing subtler ones (as seen by recall 0.3 in zero-shot). Figure 3 (from prior work) exemplifies this: ChatGPT-3.5 often ignores many flips, whereas ChatGPT-4 finds them. We expect our open models to behave similarly. In summary, LLM-based detection is promising for ad-hoc anomaly identification, but current off-the-shelf LLMs under-

TABLE I
LLM PERFORMANCE: OVERALL DETECTION AND ATTACK-SPECIFIC SANITIZATION ACCURACY

		Overall Detection Performance			**Attack-Specific Sanitization Acc.**	
Model Name	**Prompting Strategy**	**Accuracy**	**Recall**	**F1-Score**	**Similarity Acc.**	**Difference Acc.**
Gemma-2b	Zero-shot	0.5050	0.5563	0.4734	0.2375	0.2625
	Few-shot	0.4450	0.8125	0.5394	0.4250	0.3875
Llama-3.2-3B	Zero-shot	0.4000	1.0000	0.5714	0.1000	0.2750
	Few-shot	0.5450	0.3500	0.3810	0.1500	0.1250
Qwen2-1.5b	Zero-shot	0.6000	0.0000	0.0000	0.0000	0.0000
	Few-shot	0.4000	1.0000	0.5714	0.3250	0.3750
Gemma-2b (FT)	Zero-shot	0.4675	0.7063	0.5148	0.1250	0.1875
	Few-shot	0.4100	1.0000	0.5755	0.3375	0.3750
Llama-3.2-3B (FT)	Zero-shot	0.4000	1.0000	0.5714	0.2875	0.3125
	Few-shot	0.6100	0.0438	0.0824	0.0375	0.0500
Qwen2-1.5b (FT)	Zero-shot	0.4000	1.0000	0.5714	0.2750	0.2375
	Few-shot	0.4000	1.0000	0.5714	0.2750	0.3000

Note:

- The ”Overall Detection Performance” metrics reflect the model’s ability to distinguish poisoned from clean samples in general. The provided data does not break down this detection performance by specific attack types (e.g., similarity or difference attacks).
- ”Attack-Specific Correction Acc.” refers to the accuracy of the LLM in successfully correcting samples that were known to be of a ”similarity” or ”difference” attack type, calculated as (samples of type X corrected) / (total samples of type X).
- F1-scores for these specific correction sub-tasks (similarity/difference) cannot be calculated from the provided LLM data, as the necessary false positive counts for these specific sub-tasks are unavailable.

TABLE II
DETECTION AND SANITIZATION BENCHMARKS FOR TRADITIONAL ML MODELS (10% POISONING RATE)

Model	**TrainTime (s)**	**Detection**			**Sanitization**	
		Accuracy	**Recall**	**F1**	**Accuracy**	**F1**
LogisticRegression	0.1051	0.74905	0.0026	0.00515	0.54065	0.46120
RandomForest	41.6676	0.94000	0.7884	0.86790	0.87835	0.87474
SVM	2253.8209	0.89255	0.6120	0.74011	0.81430	0.80058
KNN	0.0049	0.91955	0.7604	0.82536	0.86115	0.85744
GaussianNB	0.0286	0.65900	0.5348	0.43951	0.72760	0.70945
XGBoost	0.2908	0.93480	0.7932	0.85881	0.86445	0.86121

perform traditional ML on this task unless massively fine-tuned.

From a resource perspective, traditional models train quickly on CPUs and require little memory, whereas training even a small LLM can take hours on GPUs. Moreover, ML models yield interpretable features (we can examine decision trees or SHAP values to understand which sensors indicate a flip), whereas LLM outputs are opaque. For real-world HAR applications (e.g. fitness trackers, health monitors), where power and latency are limited, the lightweight ML approach is more feasible.

FUTURE WORK

Future research could explore several directions. First, larger or multimodal LLMs (e.g. with vision modules) might better handle sensor inputs. One could convert time-series into images (e.g. spectrograms) and use vision-LLMs. Second, advanced LLM prompting (chain-of-thought, ensemble prompting) might improve few-shot detection without full fine-tuning. Third, hybrid approaches could combine ML and LLM: e.g. using LLMs to generate synthetic training examples or to flag ambiguous cases, which are then filtered by ML. Finally, expanding to other poisoning types (e.g. backdoor triggers) and datasets (beyond MotionSense) would validate generality.

CONCLUSION

In this study, we compared classical ML and open-source LLMs for detecting/sanitizing label-flip poisoning in wearable HAR data. Using the MotionSense dataset and a controlled attack, we found that ML classifiers (particularly ensemble models) achieve near-perfect detection, while LLMs (unsloth/gemma, llama-3.2, qwen2) lag behind in zero- and few-shot modes. Only extensive fine-tuning enables LLMs to approach ML performance. Given the high computational cost and limited data of wearable systems, **traditional ML**

remains the better choice for robust defense unless one can afford large-scale LLM adaptation. These results echo recent findings that non-fine-tuned LLMs are less effective for structured prediction tasks. Going forward, we encourage the community to explore more efficient LLM-based defenses, but to rely on proven ML techniques for immediate deployments.

References

[1] Katherine E Brown, Chao Yan, Zhuohang Li, Xinmeng Zhang, Benjamin X Collins, You Chen, E. Clayton, Murat Kantarcioglu, Yevgeniy Vorobeychik, and Bradley A Malin. Large language models are less effective at clinical prediction tasks than locally trained machine learning models. *Journal of the American Medical Informatics Association : JAMIA*, 2025.

[2] Mohammad Malekzadeh, Richard G. Clegg, Andrea Cavallaro, and Hamed Haddadi. Mobile sensor data anonymization. In *Proceedings of the International Conference on Internet of Things Design and Implementation*, IoTDI '19, pages 49–58, New York, NY, USA, 2019. ACM.

[3] W.K.M Mithsara, Abdur R. Shahid, and Ning Yang. Zero-shot detection and sanitization of data poisoning attacks in wearable ai using large language models. In *2024 International Conference on Machine Learning and Applications (ICMLA)*, pages 1510–1515, 2024.

[4] Devshree Patel, Param Raval, Ratnam Parikh, and Yesha Shastri. Comparative study of machine learning models and bert on squad, 2020.

[5] Abdur R. Shahid, Ahmed Imteaj, Peter Y. Wu, Diane A. Igoche, and Tauhidul Alam. Label flipping data poisoning attack against wearable human activity recognition system, 2022.

[6] Sofia Eleni Spatharioti, David M. Rothschild, Daniel G. Goldstein, and Jake M. Hofman. Comparing traditional and llm-based search for consumer choice: A randomized experiment, 2023.