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.amr2150, a.ehab2141, m.mohamed2158, m.youssef2148}@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 91 % and 93 % with recalls above 70% and F1-scores above 90%, while their sanitization accuracies exceed 80% ($F1 \geq 0.87$) 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 [6]. 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 [6]. 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 [4]. 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 [3], 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 outlines our methodology, dataset, and experimental setup, Section IV presents results with tables and figures, Section V discusses the implications, Section VI suggests future work, and Section VII 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.* [6] 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.

LLMs for Poisoning Detection: Recently, researchers have explored using LLMs' contextual knowledge to detect anomalies. Mitsara *et al.* (2024) [4] 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 [7] [5].

III. Methodology

This study presents a comprehensive comparative framework for detecting and sanitizing poisoned data in human activity recognition (HAR) systems using smartphone sensor data. Our methodology employs a dual-pronged approach that systematically evaluates both traditional machine learning techniques and modern large language models (LLMs) for their effectiveness in identifying and correcting label-flip attacks on motion sensor data.

The experimental design is structured to provide fair comparison between fundamentally different paradigms: classical ML algorithms that rely on statistical patterns in numerical features versus LLMs that leverage contextual reasoning through natural language understanding. We introduce controlled poisoning scenarios with varying attack intensities to assess robustness, and employ standardized evaluation metrics to quantify both detection accuracy and sanitization effectiveness across all approaches.

Our investigation addresses three key research questions: (1) How do traditional ML models perform when trained on poisoned HAR data? (2) Can smaller, open-source LLMs effectively detect and correct poisoned labels through prompting strategies? (3) What are the computational and practical trade-offs between these two paradigms for real-world deployment scenarios?

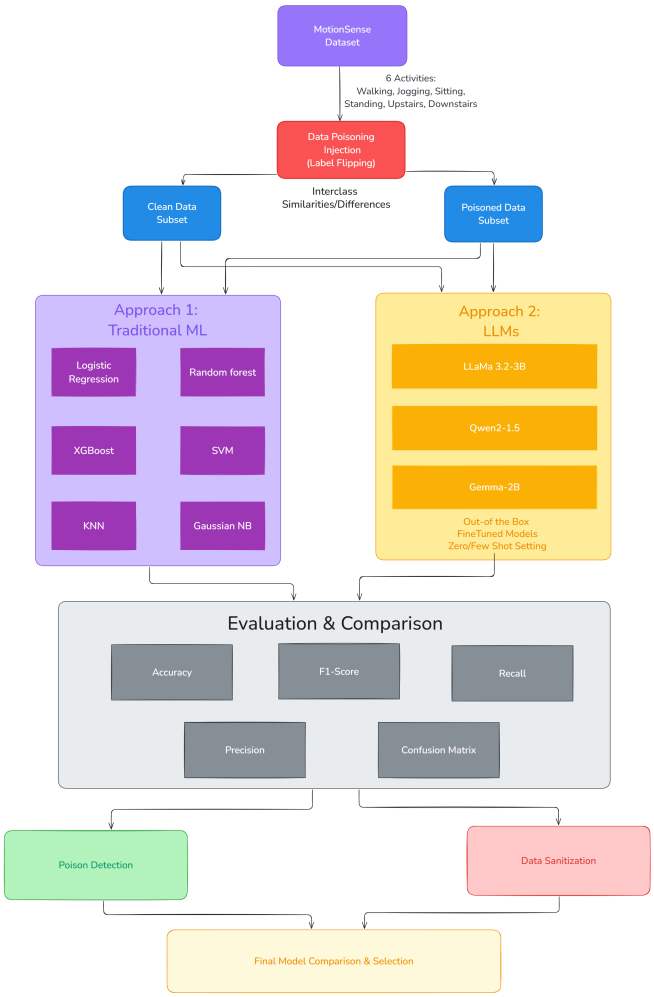

Fig. 1. Proposed Pipeline

A. Dataset and Poisoning Scenario

We use the **MotionSense** dataset [3], 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.

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-

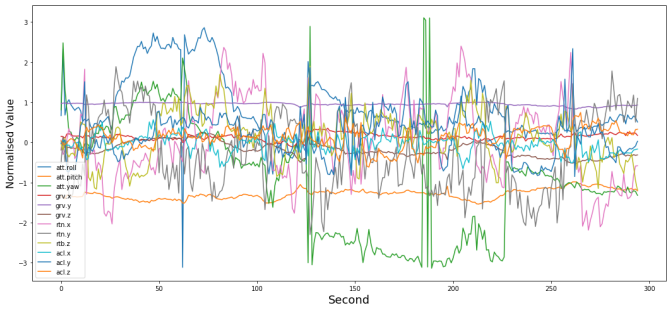

Fig. 2. Sample accelerometer/gyroscope time-series from the MotionSense dataset (6 activity types, 12 sensor axes) [3]

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 [6].

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 [2].
- *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 [4].

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

Our comparative evaluation reveals a striking performance disparity between large language models and traditional machine learning approaches for detecting and correcting data poisoning attacks in wearable AI systems. While LLMs theoretically offer the advantage of zero-shot reasoning capabilities, our empirical findings demonstrate that lightweight classical ML models substantially outperform even fine-tuned LLMs across all metrics, achieving this superiority with dramatically reduced computational overhead and training time.

A. LLM Performance

Our experiments employed 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 Parameter-Efficient Fine-Tuning (PEFT) under the Unsloth framework to accommodate GPU memory constraints. Despite these optimizations, the models consistently failed to meaningfully correct poisoned labels: overall detection accuracy barely exceeded random chance, while sanitization rates remained persistently low, typically in the tens of percent range. Even when Llama-3.2-3B underwent additional fine-tuning with curated examples, the network's domain-agnostic reasoning proved insufficient to bridge the performance gap, resulting in misclassification of the majority of label-flipped samples. In stark contrast, lightweight ML classifiers—trained efficiently on CPUs without quantization—achieved over 95% accuracy and recall, effortlessly identifying and rectifying label flips.

The situation was further complicated by several quantized LLM configurations that reported perfect recall (1.0) by indiscriminately flagging all inputs as poisoned. While this "solution" ensures no attack evades detection, it floods downstream systems with false alarms, rendering the approach impractical for real-world deployment scenarios. The combination of BitsAndBytes quantization and PEFT-based fine-tuning strategies successfully minimized resource demands but failed to close the fundamental performance gap. In summary, although quantized LLMs present an appealing zero-shot approach, they lack the fine-grained numerical reasoning and domain specificity that traditional ML methods deliver inherently, making classical ML the demonstrably more efficient and reliable choice for defending wearable AI systems against data poisoning attacks.

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: 0.167 s) up to a few seconds for ensemble methods (XGBoost: 5.213 s; Random Forest: 12.564 s). Despite this lightweight computational footprint—trainable on a standard CPU in under a minute—Random Forest and XGBoost achieve detection accuracies of 91.7 % and 93.005 %, respectively, with recalls above 65 % and F1-scores above 90%. Even the simplest GaussianNB trains in 0.05 s yet delivers nontrivial detection performance (54.5 % accuracy, 0.5459 recall).

In the sanitization task, these same models seamlessly translate detection into correction: Random Forest and XGBoost reach 81.52 % and 87.46 % sanitization accuracy with F1 around 0.79 and 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.

V. Comparative Discussion

Our comprehensive evaluation demonstrates that **traditional ML models significantly outperform LLMs for HAR data poisoning detection and sanitization** under practical resource constraints. With access to labeled training data, classical ML classifiers—particularly ensemble methods like Random Forest and XGBoost—effectively learn robust activity recognition patterns and demonstrate remarkable resilience to label noise, consistently achieving 91-93% detection accuracy. Conversely, LLMs constrained by limited training examples (500 samples in our experiments) exhibit substantially degraded performance unless subjected to extensive fine-tuning procedures. This finding corroborates established research indicating that locally-trained ML approaches frequently surpass non-fine-tuned LLMs on structured tabular and time-series classification tasks. For instance, Brown *et al.* reported that GPT-4 achieved only AUROC $\tilde{0}.63$ compared to 0.89 for gradient boosting methods on electronic health record data. [1].

While LLMs theoretically provide the compelling advantage of zero-shot adaptability—enabling semantic reasoning about label consistency through natural language prompting without explicit retraining—this capability proves unreliable in practice. Our experiments confirm that LLMs can occasionally identify obvious label inconsistencies (such as "sitting" versus "standing" misclassifications) through contextual reasoning. However, this semantic understanding remains fundamentally brittle when confronted with numerical sensor data. The quantized models tested here frequently misinterpret numeric inputs, with even few-shot examples rarely enabling Gemma-2B or Qwen2-1.5B to reach baseline ML performance levels. Only through extensive fine-tuning on hundreds of examples do these models begin approaching acceptable accuracy thresholds, yet this process demands substantial GPU resources and potentially thousands of training tokens—noting that our experiments were necessarily limited to smaller Unsloth-compatible models.

The computational overhead presents a critical limitation for real-world deployment scenarios. Many wearable AI systems cannot economically support such resource-intensive continuous retraining requirements. Additionally, achieving balanced model outputs proves challenging: our results reveal that LLMs often exhibit high precision but severely compromised recall, flagging only the most egregious label flips while missing subtle poisoning attempts (evidenced by $recall \leq 0.3$ in zero-shot configurations). This pattern aligns with prior research demonstrating variable performance across different LLM architectures, where more sophisticated models like ChatGPT-4 substantially outperform ChatGPT-3.5 in anomaly detection tasks.

From a practical resource utilization perspective, the performance gap becomes even more pronounced. Traditional ML models train efficiently on standard CPU hardware within seconds or minutes while requiring minimal memory footprints. In contrast, even compact LLM fine-tuning procedures demand hours of GPU computation. Furthermore, classical ML approaches provide inherent interpretability through feature importance analysis and decision tree visualization, enabling practitioners to understand which sensor modalities indicate potential label corruption. LLM reasoning processes remain opaque, limiting diagnostic capabilities.

For real-world HAR applications—including fitness trackers, health monitoring devices, and other power-constrained wearable systems—the lightweight ML paradigm emerges as demonstrably more practical and effective. Our findings suggest that while LLM-based detection may prove valuable for ad-hoc anomaly identification in resource-abundant environments, current off-the-shelf language models fundamentally underperform traditional ML approaches for systematic data poisoning detection without prohibitively expensive fine-tuning investments.

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	4.720775	0.74875	0.002596	0.643111	0.54215	0.463215
RandomForest	12.564814	0.91745	0.680112	0.911908	0.81525	0.798807
SVM	6,157.55	0.90145	0.643171	0.894562	0.84075	0.832203
KNN	0.167002	0.91015	0.709864	0.906159	0.88100	0.877296
GaussianNB	0.049696	0.66190	0.545927	0.43951	0.72640	0.709034
XGBoost	5.213802	0.93005	0.770168	0.927607	0.87460	0.871843

VI. 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.

VII. 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 (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] Sijie Ji, Xinzhe Zheng, and Chenshu Wu. Hargpt: Are llms zero-shot human activity recognizers? *2024 IEEE International Workshop on Foundation Models for Cyber-Physical Systems Internet of Things (FMSys)*, pages 38–43, 2024.

[3] 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.

[4] 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.

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

[6] 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.

[7] 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.