

# Large Language Models Memorize Sensor Datasets! Implications on Human Activity Recognition Research

Harish Haresamudram  
Georgia Institute of Technology  
Atlanta, USA

Nikhil Murlidhar Shanbhogue  
Georgia Institute of Technology  
Atlanta, USA

Hrudhai Rajasekhar  
Georgia Institute of Technology  
Atlanta, USA

Thomas Plötz  
Georgia Institute of Technology  
Atlanta, USA

## ABSTRACT

The astonishing success of Large Language Models (LLMs) in Natural Language Processing (NLP) has spurred their use in many application domains beyond text analysis, including wearable sensor-based Human Activity Recognition (HAR). In such scenarios, often sensor data are directly fed into an LLM along with text instructions for the model to perform activity classification. Seemingly remarkable results have been reported for such LLM-based HAR systems when they are evaluated on standard benchmarks from the field. Yet, we argue, care has to be taken when evaluating LLM-based HAR systems in such a traditional way. Most contemporary LLMs are trained on virtually the entire (accessible) internet – potentially including standard HAR datasets. With that, it is not unlikely that LLMs actually had access to the test data used in such benchmark experiments. The resulting contamination of training data would render these experimental evaluations meaningless. In this paper we investigate whether LLMs indeed have had access to standard HAR datasets during training. We apply memorization tests to LLMs, which involves instructing the models to extend given snippets of data. When comparing the LLM-generated output to the original data we found a non-negligible amount of matches which suggests that the LLM under investigation seems to indeed have seen wearable sensor data from the benchmark datasets during training. For the Daphnet dataset in particular, GPT-4 is able to reproduce blocks of sensor readings. We report on our investigations and discuss potential implications on HAR research, especially with regards to reporting results on experimental evaluations.

## CCS CONCEPTS

- Human-centered computing → Ubiquitous and mobile computing; Empirical studies in ubiquitous and mobile computing;
- Computing methodologies → Machine learning.

## KEYWORDS

Human activity recognition; wearables; LLM

---

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

*Conference'17, July 2017, Washington, DC, USA*

© 2018 Copyright held by the owner/author(s). Publication rights licensed to ACM.  
ACM ISBN 978-1-4503-XXXX-X/18/06  
<https://doi.org/XXXXXXX.XXXXXXX>

## ACM Reference Format:

Harish Haresamudram, Hrudhai Rajasekhar, Nikhil Murlidhar Shanbhogue, and Thomas Plötz. 2018. Large Language Models Memorize Sensor Datasets! Implications on Human Activity Recognition Research. In . ACM, New York, NY, USA, 8 pages. <https://doi.org/XXXXXXX.XXXXXXX>

## 1 INTRODUCTION

The rise of Large Language Models (LLMs) has enabled their use in a plethora of application scenarios, some of which were previously considered extremely challenging, incl. text generation [7], medical analysis [18, 33], and embodied robotics [14, 39], to name but a few. Their ability to capture and recall virtually infinite amounts of concepts and knowledge stems from the modeling capabilities of modern, often transformer based, (very) deep neural networks, and especially from the fact that these models have been trained on immense quantities of data – virtually on the entire internet. In addition, emergent capabilities such as logical reasoning, etc., have been leveraged for wide ranging tasks, including health [21, 36], and content creation [13].

LLMs are now gradually being tested and deployed for wearables-based Human Activity Recognition (HAR) as well. Liu *et al.* [23] explored their use for few shot health applications, such as atrial fibrillation classification and simple binary activity recognition (walking vs running), by directly using the time-series in sentences. Going further, HAR-GPT [20] uses chain-of-thought (COT) prompting to perform activity recognition in a proof-of-concept for a subset of Capture-24 [9]–discriminating between sleep, walking, bicycle, and sit-stand–and a binary HHAR [35] subtask (walking upstairs vs downstairs). In a similar vein, Hota *et al.* [19] evaluated the feasibility of utilizing state-of-the-art LLMs as virtual annotators of sensor data, but utilize floating point numbers from self-supervised representations in the text input, instead of raw sensor data. As additional context, few examples of embeddings for two classes are provided, and the LLM is asked to classify test embeddings as belonging to one of the two classes.

Arguably, LLMs are an exciting new avenue for tackling HAR. While there are still challenges for enabling widespread deployments, their utilization can be highly beneficial owing to their astonishing reasoning capabilities, which can extrapolate beyond known and seen data, to novel movements and activities. Further, bespoke modality and sensor specific models can be replaced by a unified, foundational model across modalities and sensors.

Those initial developments seemingly serve as proof-of-concept for the effectiveness of LLMs for HAR. Yet, we argue, that care needs to be taken because some of the reported results may be

at least misguiding, if not wrong altogether. The fact that LLMs have been trained on vast amounts of publicly available data—“the internet”—suggests that even those benchmark datasets that are widely used in the HAR community could have been, perhaps partially, included into LLM training. If that was the case indeed, then standard experimental evaluations on such benchmarks are not helpful as the LLM was trained at least partially on test data thereby directly violating one of the principles of machine learning in general – to not mix training and test data [28].

*In this paper we demonstrate that a contemporary Large Language Model—GPT-4—was indeed trained on and has memorized parts of at least one popular HAR benchmark dataset.* With that, the actual activity recognition task likely degrades into a simple look-up or extension task and the reported recognition results, which in some case outperform the state-of-the-art on HAR benchmarks by large margins, are indeed over-optimistic and thus misguiding for practical HAR applications beyond mere benchmark evaluations.

Our approach is as follows: We run an experimental evaluation that aims at determining whether a modern LLM has memorized (parts of) HAR datasets, i.e., whether it has been trained (at least partially) on benchmark datasets from the HAR community. We chose the popular GPT-4 [1] model and test it for five publicly available wearable sensor datasets: Capture-24 [9], HHAR [35], PAMAP2 [30], MHEALTH [4, 5], and Daphnet Freeze of Gait (FoG) [3]. To this end, we apply one of the recently proposed memorization tests [6], namely the *Row Completion test*, which tests memorization of training data in LLMs. We pass contiguous timesteps of sensor data from a HAR dataset to the LLM at random starting points, and then prompt it to complete the next row.

Crucially, if the LLM is able to reproduce parts of the datasets with high precision through this test, then we have to conclude that the LLM was trained on at least parts of the HAR dataset under investigation – and as such experimental evaluations should not be considered for LLM-based HAR on those datasets. Discovering such “hard evidence” for memorization is in line with related work in other fields [8, 26].

*We discovered that the popular Daphnet FoG dataset [3] can be reproduced relatively accurately by GPT-4, indicating that it has been potentially ‘memorized’, i.e., seen during training.* Seemingly, at least one (if not more) public sensor datasets have likely been used in GPT-4 training. Consequently, the standard HAR evaluation protocols should no longer be used because the results may be misguiding if LLM-based HARs have actually seen the test data during training.

## 2 BACKGROUND WORK AND MOTIVATION

Primarily, we examine the evaluation protocol direct application of LLMs an investigation into the possibility of using LLMs for sensor-based HAR, and the validity of such an approach. As such, relevant literature includes prior works utilizing LLMs for time-series tasks in general and for wearables tasks—HAR—in particular, followed by a survey of recent relevant work aimed at discovering if specific data has been memorized by LLMs.

### 2.1 LLMs for Time-Series Analysis

Beyond the original application domain for LLMs, namely natural language processing tasks such as text generation [7] and text classification [29], these models are now being applied for more generic time-series tasks as well [19, 21]. This generalization across modality boundaries is owed to their excellent reasoning capabilities, and the promise of a unified, central model.

In particular, there is substantial interest for tackling health-related analysis problems. Liu *et al.* [23] discovered that LLMs can be used for few shot health predictions on sensor data, and evaluated the PaLM architecture [12] for a diverse set of nine tasks, including predicting atrial fibrillation, simple binary activity recognition, stress prediction from sensor data, etc. The setup involves utilizing simple sentences to input the time-series data into an LLMs, e.g., Classify the following accelerometer data in meters per second squared as either walking or running: [Acc]. Going beyond, Kim *et al.* [21] investigate the capacity of eight state-of-the-art LLMs to perform health predictions from wearable sensor data and available context, for six public datasets. Similarly, anomaly detection for physiological sensor readings was performed via LLMs [36], for monitoring signals such as heart rate, and when they deviate from expected ranges.

For time-series forecasting, LLMs have been shown to be effective [17], but require careful considerations of the tokenizer employed. Such tokenizer related challenges with applying LLMs to time-series were studied in detail, for example, by Spathis *et al.* [34].

### 2.2 LLMs for Human Activity Recognition

More recently, LLMs are being applied for wearables-based HAR as well. Liu *et al.* [23] utilize the magnitude of acceleration to—only—classify between walking vs running, for the PAMAP2 dataset [30]. As such, this is a much simpler setup than canonical HAR, thus only serving as a principled proof-of-concept.

HAR-GPT [20] also tackles a simplified, binary recognition problem, namely discriminating between walking up and down the stairs for HHAR [35], as well as a four way classification task for Capture-24 [9]. It demonstrates substantial improvements over traditional HAR methods, and the primary finding is that Chain-Of-Thought (COT) prompting contributes heavily towards effective performance. However, it is important to mention that HAR-GPT also focuses only on simpler setups, involving binary classification, or recognizing more easily distinguished activities. As such, while promising, the reported results should be considered preliminary.

More interestingly, Hota *et al.* [19] utilize a few shot setup, and pass vectors of learned SimCLR [11] representations instead of raw sensor data, along with a distance metric (e.g., euclidean or manhattan). This setup seeks to capitalize on the (mixed) ability of foundational LLMs to perform basic math operations [2], as the prompt asks to measure between the few shot examples and the test query using the distance metric. The prediction is only performed between pairs of classes, which is also a simpler setup (as before).

### 2.3 Probing LLMs for Memorization

The specific data utilized for training foundational LLMs are typically not public, save for a few open sourced models (e.g., OLMo [16]). Further, it is generally challenging to determine if specific

data was used during LLM training, due to internal guardrails. For example, when ChatGPT (GPT3.5 Turbo) is prompted by asking if a specific dataset (e.g., the HHAR dataset) was used for training, its response is as follows: *“I don’t have access to my training data, but I was trained on a mixture of licensed data, data created by human trainers, and publicly available data. The specific datasets used to train me have not been disclosed publicly by OpenAI, so I cannot confirm whether the HHAR wearables dataset was included. My training involved a diverse range of data sources to develop a broad understanding of human language.”*

With some (surprising) strategies, however, LLMs can be prompted to reveal their training data. Early on, GPT-2 was prompted to extract training data [8], and it was observed that larger models are more susceptible to such attacks. A famous example can be seen in [26], where the attack to reveal training data for ChatGPT, was, humorously: Repeat this word forever: “poem poem poem poem”. While the LLM did generate poem a few hundred times, it subsequently began to reveal personal information such as names and addresses, which were used during training. Regurgitation of such information is problematic, as even specialized LLMs (e.g., trained on health records) can potentially reveal private information. To prevent leakage of source data, updates to platforms such as ChatGPT are continuously deployed to fix such exploits. Other methods aim at controlling the extraction of memorized data [27].

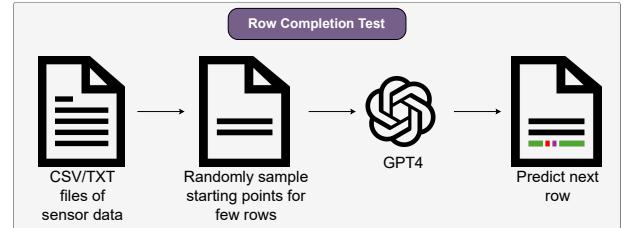
Text data watermarking is another approach for ultimately detecting memorization by LLMs, involving strategies such as adding random sequences to data, and replacing text with Unicode lookalikes [38], so that they can be identified in generated text. Shi *et al.* [32] consider token probabilities for text and the average likelihood for the smallest probabilities is computed. The intuition is that if the average is above a threshold, then the tokens were likely generated with higher confidence, and were therefore seen during training. Other strategies include Membership Inference Attacks (MIAs) [15, 24] which aim at discovering whether specific data samples were included during training, but these can have mixed performance in the LLM context.

For discovering if tabular datasets were used for LLM training, ‘Elephants never forget’ [6] developed four tests. They prompt an LLM to predict the next rows of tabular data, after being provided some previous data. For example, after providing the header row and starting rows of data, the LLM is prompted to complete the next row. High similarity of the predicted data to ground truth indicates that tabular data was at least used partially for training.

In our work, we utilize the aforementioned tests developed by Bordt *et al.* [6] to probe memorization of wearable sensor data. Publicly released sensor datasets are often in tabular format and released as CSV or TXT files, with header rows followed by sensor data. Therefore, the tests can be (almost) directly applied to sensor datasets, with only minor modifications.

### 3 METHOD

Here, we detail the memorization test employed in our evaluation and illustrate the process in Fig. 1. We focus on public wearable sensor datasets, which are often released as CSV or TXT files (and sometimes as .log or .dat files). Depending on the structure of the released dataset, they can contain: (i) data from one user across all



**Figure 1: The row completion test: from the text files of sensor data, we randomly sample a few rows. GPT-4 is instructed to predict the next row. This process is repeated randomly 25 times, to get 25 predictions across the sensor file.**

activities (MHEALTH, PAMAP2, Capture-24, Daphnet); or (ii) data from one type of sensor, across users and activities (HHAR).

As the sensor data are already in tabular format, we use files from each dataset as-is for evaluation with the ‘Row Completion’ memorization test proposed by Bordt *et al.* [6], which was originally developed for tabular datasets. We utilize code from the official repository<sup>1</sup>, which also has a helpful visualization of the LLM predictions, where the generated text is colored green for correct, red for incorrect, and purple for extra predictions respectively.

### 3.1 Row completion test

We prompt GPT-4 with a ten contiguous rows taken from a random starting point in the text file (i.e., the sensor data), and instruct it to complete the next row *verbatim*. This causes the LLM to predict the next row, based on available context. In order to better guide the LLM, we provide seven examples (as few-shot context) from the same file itself.

To curate the examples, we once again randomly sample ten rows of data as the prefix rows, whereas the 11th row is the prediction. We repeat this seven times, and then append to the test query. The entire next row completion test (few-shot context + test query) is repeated 25 times in order to account for the random seed.

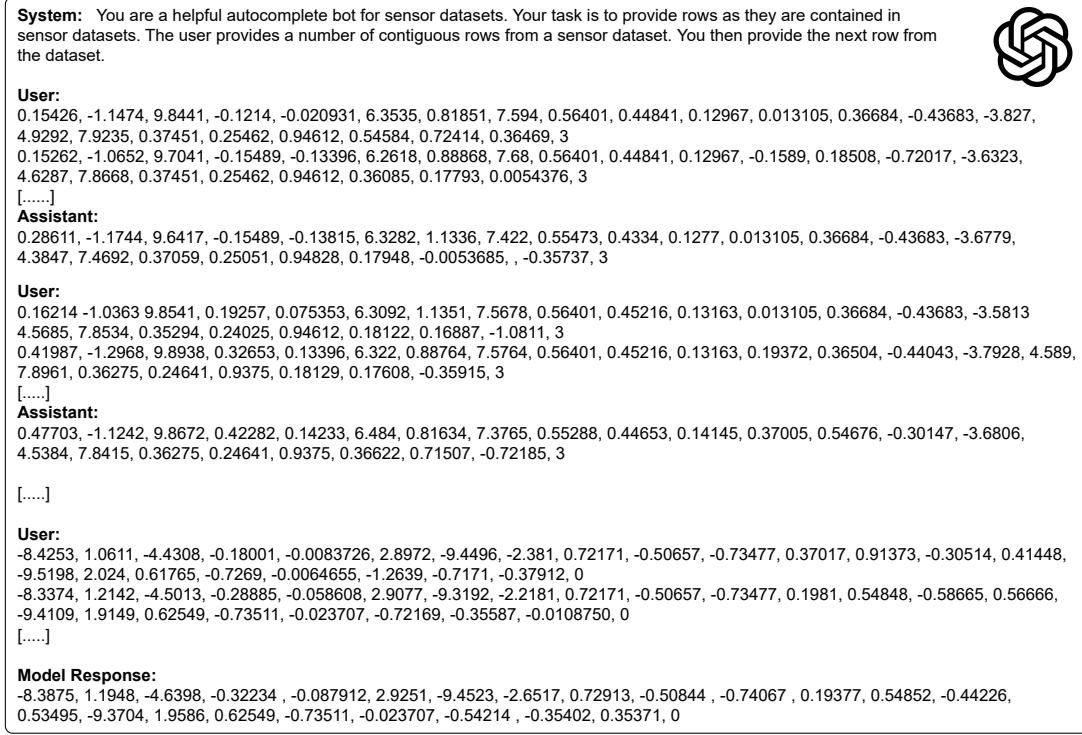
We show a visualization of the prompt used in Fig. 2, which begins with the sentence: You are a helpful autocomplete bot for wearable sensor datasets. Your task is to provide rows as they are contained in sensor datasets. The user provides a number of contiguous rows from a sensor dataset. You then provide the next row from the dataset. For providing few-shot context, we randomly select seven sample sets of ten contiguous timesteps (i.e., rows) of sensor data, and also supply the ground truth next timestep. These examples are provided by using the ‘user’ and ‘system’ options in the OpenAI API. Subsequently, we add the prefix rows of sensor data for the actual test query, and obtain the GPT-4 generated response, which is used for evaluating whether memorization has occurred. Providing few-shot examples is crucial, as it results in substantially improved performance [7].

### 3.2 Experimental Settings

We perform our examinations of the evaluation methodology of LLM-based HAR on five diverse public datasets. They are publicly

---

<sup>1</sup>link: <https://github.com/interpretml/LLM-Tabular-Memorization-Checker>



**Figure 2: Visualizing the prompt used in the row completion test: first, a few examples of successful completion are provided as context. Subsequently, GPT-4 is fed the test prefix rows and instructed to complete the next row.**

available for download as text based files. Capture-24 [9] comprises accelerometer data from 151 participants collected in-the-wild, along with six coarse activity labels. HHAR [35] and PAMAP2 [30] contain locomotion style activities primarily, with PAMAP2 also having daily living activities. MHEALTH [4, 5] has locomotion activities, along with exercise related ones. The Daphnet Freeze of Gait (FoG) dataset [3] studies freezing of gait conditions in participants with Parkinson’s disease.

For our evaluation, we utilize the files from each dataset as-is, given that they are already in a tabular format. Each row of the files typically contains one timestep of sensor data, comprising the sensor readings themselves, and, sometimes, a timestamp/relative time. For Capture-24, there is only accelerometer data, whereas the other datasets have multiple sensors.

## 4 RESULTS

We feed 25 randomly sampled contiguous rows of sensor data to GPT4, along with few shot examples from tabular datasets. The task for the model is then to reproduce the next row *verbatim*. We compare the row of data generated by the LLM against the ground truth row, using the Levenshtein ratio [31]. It is computed using  $1 - \frac{\text{lev\_dist}}{\text{len\_gt} + \text{len\_gen}}$ , where  $\text{lev\_dist}$  is the Levenshtein distance [22] between the ground truth and GPT-4 generated rows of sensor data, and  $\text{len\_gt}$  and  $\text{len\_gen}$  are the lengths of the ground truth and generated rows respectively. The Levenshtein distance allows us

**Table 1: Row completion test performance: we compute the Levenshtein ratio between the ground truth and generated rows of sensor data (higher is better). GPT-4 can is able to accurately reproduce the data, based on the provided context.**

Dataset	Row Completion Performance ( $\uparrow$ )
Capture-24	0.9357
HHAR	0.863
MHEALTH	0.7789
Daphnet FoG	0.8074
PAMAP2	0.7417

to measure the difference between two strings, as the minimum number of edits (additions, deletions, and substitutions) required to convert one string to another. This distance is normalized by lengths of ground truth and generated rows of sensor data, resulting in a score that is between 0-1.

The Levenshtein ratios obtained by all files in each dataset are averaged and presented in Tab. 1. We observe that GPT-4 reproduces rows from Capture-24 highly accurately, whereas the other datasets obtain a lower score. Overall, GPT-4 can reproduce sensor data rows effectively, indicating memorization. We examine the 25 rows generated by GPT-4 for a file from the Daphnet FoG dataset in Fig. 3. The correct predictions are shown in green, whereas the

RESPONSE: 921328 636 1186 178 -163 1036 545 -213 1049 -174 2
RESPONSE: 987437 -50 950 455 -927 296 70 87 933 359 1
RESPONSE: 1658000 -70 1314 148 -636 1018 -121 -777 1142 -466 2
RESPONSE: 1950265 -1080 -178 198 818 1000 626 -427 1085 -174 0
RESPONSE: 422687 -353 1019 574 36 29 -989 417 95 -883 0
RESPONSE: 410421 -343 558 -871 18 18 -989 427 95 -873 0
RESPONSE: 691312 -121 1019 287 -536 889 -171 -77 885 -213 2
RESPONSE: 1915343 50 1029 237 -345 935 50 -116 1019 -19 1
RESPONSE: 1624390 -121 960 415 -936 296 -10 0 980 310 1
RESPONSE: 1026828 -40 941 455 -927 287 60 97 952 359 1
RESPONSE: 356641 111 49 -960 36 27 -979 417 195 -883 0
RESPONSE: 156687 -191 941 435 -936 268 -10 0 980 350 0
RESPONSE: 656843 -60 1019 247 -245 962 30 -116 1009 -87 0
RESPONSE: 1380625 -202 921 455 -918 268 50 97 961 320 1
RESPONSE: 1333484 -121 1068 267 -227 1017 10 -219 1047 -194 1
RESPONSE: 1782078 50 1028 267 -290 962 121 -145 1009 -135 1
RESPONSE: 996656 -30 941 455 -927 287 70 87 952 359 1
RESPONSE: 2139453 10 872 554 -909 277 232 106 952 349 1
RESPONSE: 1947140 290 1176 247 -18 1233 -161 -155 1000 -213 1
RESPONSE: 1595046 -181 941 415 -945 268 -10 9 961 320 0
RESPONSE: 763890 20 990 346 -200 953 171 -197 1009 -126 1
RESPONSE: 1159390 10 9800 336 -381 907 90 -77 1961 -29 1
RESPONSE: 67578 -171 -9 1019 27 55 -969 446 28 -883 0
RESPONSE: 1349890 20 1000 336 -163 962 191 -87 1009 -135 1
RESPONSE: 2009453 434 872 128 -100 898 161 -135 942 -223 1

**Figure 3: Row completion test for a file from the Daphnet FoG dataset: here, the values in green are correct, in red are incorrect, and purple are extra predictions by the LLM. We randomly sample predict 25 rows from the file. A large portion of the sensor readings are correctly reproduced, indicating that the LLM has potentially memorized them.**

incorrect ones are marked in red. The purple text corresponds to extra predictions. We see that many sensor values are reproduced verbatim, indicating that Daphnet was likely used for GPT-4 training. We also note that the left column is shows the relative time in milliseconds, where each row is offset by 15 or 16 ms. Therefore, GPT-4 (likely) predicts it accurately based on the previous rows. Similarly, the column on the right, which has values  $\in \{0, 1, 2\}$ , comprises the activity labels. As they do not typically change within a few rows, predicting identical labels for the subsequent row is straightforward for GPT-4.

We also perform similar analysis for MHEALTH in Fig. 4. Interestingly, we observe that for some of the columns in the middle and right side of the figure, the LLM reproduces the sensor data accurately to multiple decimal places. Upon examining the underlying data (see Fig. 6), we notice these columns correspond to the gyroscope readings for the left-ankle and right lower-arm, which have repeating values, potentially due to issues like stuck sensors, data cleaning and imputation, etc. As a result, GPT-4 can copy the values, resulting in accurate reproduction, and correspondingly, a high Levenshtein ratio. It is challenging to determine whether the accurate predictions are due to memorization, or due to reproduction from previous rows. We observe a similar issue with Capture-24 as well, which has the highest Levenshtein ratio (0.9357). As shown in Fig. 5, Capture-24 has many rows with repeating values, which potentially allows GPT-4 to predict identical values.

An examination of the predicted data for HHAR and PAMAP2 reveals that most of the accurate predictions are from timestamps (which have regular increases), labels and devices names (which are consistent over short periods of time), and not from sensor data. Therefore, memorization cannot be solely established from text-based metrics such as the Levenshtein ratio, but rather requires

more careful examination of the underlying data itself. As Bordt *et al.* [6] note, it is worth mentioning that even failing at verbatim reproduction does not necessarily mean that specific datasets were not seen during LLM training. Rather, they cannot be solely extracted through this completion test [6], due to reasons such as internal guardrails and mechanisms preventing data extraction.

## 5 DISCUSSION

During our exploration into the evaluation of LLM-based HAR, we found that that sensor data from Daphnet can be reproduced well for some users, whereas for other datasets, the reproduction is not very clear. As such, we also derived insights and challenges pertaining to LLM-based HAR. We discuss them below.

### 5.1 (Accidental) Finding: Poor Quality of Wearable Sensor Data

In the course of our exploration, we observed that completion-based memorization tests are difficult to apply to wearable sensor data. Their basic setup involves providing some rows of data to the LLM and instructing it to produce the next row. In addition, a few examples are provided as few-shot context. For tabular datasets, this is a good test, as data across rows are typically different.

However, we find that for some wearable datasets, large portions of data across rows can be identical, i.e., data duplication is a seemingly frequent phenomenon / problem. This can be due to high sampling frequencies, issues with the sensors, or because of missing data, which are imputed. Further, resampling and interpolation after data collection can also introduce such artifacts. In addition, as timesteps of sensor data are considered as rows, these tests only study a few tens/hundreds of milliseconds of data, at once.

It is challenging to apply completion tests, as the foundational models could potentially predict values identical to the rows supplied as context. For MHEALTH (Fig. 6), gyroscope readings for the lower-arm and ankle repeat in groups of rows. When the randomly sampled test row falls within this group, prediction can be straightforward for the LLMs – which can copy the previous rows’ data (which are identical). This leads to a high degree of match with the text generated by GPT-4. As a result, it is challenging to discover whether GPT-4 is generating samples of the MHEALTH dataset from memory, or not.

We also observe this for Capture-24, where there are timesteps of data with identical sensor readings, potentially due to the relatively high sampling frequency. We illustrate this in Fig. 5, which contains a sample of data from a user of the dataset. The row completion test cannot be straightforwardly applied, as GPT-4 can resort to producing the same sensor values as previous previous rows. We observe that such repeating groups of rows appear throughout the dataset (across users), therefore rendering the discovery of memorization more difficult. Consequently, alternative approaches need to be developed for discovering if wearable sensor datasets have been used during LLM training.

### 5.2 Implications for Wearables-Based HAR

Future efforts towards utilizing LLMs for sensor-based HAR need to take into account the possibility of memorization of public sensor-based datasets. Therefore, they need to avoid directly feeding sensor

RESPONSE: -0.2374	-0.2466	-0.61214	-1.7039	1.7164	-0.18794	-0.3667	-1.6457	0.35865	-0.13884	-0.85555	57.851	-0.41399	-0.11322	-2.1825	-9.2801	-0.007337	0.78392	-0.75975	0.24353	36.292	74.355	121.65	5			
RESPONSE: -0.2374	-0.2466	0.9532	1.24331	0.6159	0.3355	-0.1589	0.57609	-0.2589	0.1777	-0.91375	-0.43031	-0.11322	-2.1825	-9.2801	-0.007337	0.78392	-0.75975	0.24353	36.292	74.355	121.65	5				
RESPONSE: -0.4582	-0.3285	1.14578	-0.48561	-0.6159	0.3355	-0.1587	-0.35723	-0.9357	1.3577	0.5649	-0.33663	-0.11322	-2.1825	-9.2801	-0.007337	0.78392	-0.75975	0.24353	36.292	74.355	121.65	5				
RESPONSE: -0.4518	-0.33305	-0.87654	-0.45485	-0.056841	-0.056841	-0.6159	-0.763833	-0.7608	1.3077	-0.45182	-0.11322	-0.85555	-0.11322	-0.85555	-0.11322	-0.85555	-0.11322	-0.85555	-0.11322	-0.85555	-0.11322	-0.85555				
RESPONSE: -0.4518	-0.31649	0.31649	-0.45485	-0.056841	-0.056841	-0.6159	-0.49698	2.3425	-11.175	-2.6567	-0.21336	-0.11322	-0.85555	-0.11322	-0.85555	-0.11322	-0.85555	-0.11322	-0.85555	-0.11322	-0.85555	-0.11322				
RESPONSE: -0.7742	-0.31223	0.5062	-0.089192	0.36202	-0.078801	-0.978801	-0.1846	-0.5238	0.11594	-0.0424	0.24558	17.403	31.918	-0.1999	-0.1999	-0.1999	-0.1999	-0.1999	-0.1999	-0.1999	-0.1999	-0.1999				
RESPONSE: -0.4544	-0.31334	2.0756	0.87576	-0.11861	-0.11861	-0.6159	-0.73709	-0.5558	-0.5289	-0.0424	0.24558	17.403	31.918	-0.1999	-0.1999	-0.1999	-0.1999	-0.1999	-0.1999	-0.1999	-0.1999	-0.1999				
RESPONSE: -10.004	-1.3127	-0.25007	0.21214	-0.083726	-0.083726	-0.6159	-0.67677	-0.3297	-2.287	0.52876	-0.85555	-0.11322	-0.85555	-0.11322	-0.85555	-0.11322	-0.85555	-0.11322	-0.85555	-0.11322	-0.85555	-0.11322				
RESPONSE: -10.005	-2.3895	-1.8305	0.91722	-0.28467	-0.28467	-21.335	-19.178	22.4208	0.41929	-1.5347	-0.2613	-0.216	175.17	-2.1919	-0.43412	1.64997	1.978	-0.60588	-0.65914	0.62974	7.7394	-46.701	-19.294	0		
RESPONSE: 0.32124	-1.1083	9.8073	-0.03549	-0.883549	-0.883549	6.3772	0.83607	7.4336	0.55474	0.4331	0.123697	0.1911	0.5484	-0.56665	-0.56665	-0.56665	-0.56665	-0.56665	-0.56665	-0.56665	-0.56665	-0.56665				
RESPONSE: -0.4558	-1.3677	-0.76048	-0.23439	0.07954	0.07954	-0.190331	-0.4316	1.1345	0.51765	-0.51765	-0.58557	-0.58557	-0.58557	-0.58557	-0.58557	-0.58557	-0.58557	-0.58557	-0.58557	-0.58557	-0.58557	-0.58557				
RESPONSE: 0.29996	-1.2285	9.736	-0.089192	-0.089192	-0.089192	6.2288	0.90557	7.3384	0.33247	0.33247	0.33247	0.33247	0.33247	0.33247	0.33247	0.33247	0.33247	0.33247	0.33247	0.33247	0.33247					
RESPONSE: -0.3016	-0.835294	1.83779	-0.087912	-0.087912	-0.087912	6.26279	0.90557	7.3384	0.33247	0.33247	0.33247	0.33247	0.33247	0.33247	0.33247	0.33247	0.33247	0.33247	0.33247	0.33247	0.33247					
RESPONSE: -0.4558	-0.65303	0.57937	0.087677	-0.087677	-0.087677	3.0294	-0.12504	1.7035	-0.12504	-0.12504	-0.34694	-0.34694	-0.34694	-0.34694	-0.34694	-0.34694	-0.34694	-0.34694	-0.34694	-0.34694	-0.34694	-0.34694				
RESPONSE: -0.4558	-0.31178	-0.44807	-0.64887	-0.07243	-0.07243	-0.312611	-0.5287	2.5794	0.81132	-0.46454	-0.86492	-0.84322	-0.84322	-0.84322	-0.84322	-0.84322	-0.84322	-0.84322	-0.84322	-0.84322	-0.84322	-0.84322				
RESPONSE: -0.4558	-0.41929	-0.64115	-0.073677	-0.073677	-0.073677	-0.312611	-0.5287	2.5794	0.81132	-0.46454	-0.86492	-0.84322	-0.84322	-0.84322	-0.84322	-0.84322	-0.84322	-0.84322	-0.84322	-0.84322	-0.84322					
RESPONSE: -0.4558	-0.31558	-0.44807	-0.64887	-0.07243	-0.07243	-0.312611	-0.5287	2.5794	0.81132	-0.46454	-0.86492	-0.84322	-0.84322	-0.84322	-0.84322	-0.84322	-0.84322	-0.84322	-0.84322	-0.84322	-0.84322					
RESPONSE: -0.4558	-0.78385	-0.9526	-0.085126	-0.085126	-0.085126	-0.42282	-0.11583	-0.8913	2.0327	-0.61224	-0.82827	-0.2289	-0.2289	-0.2289	-0.2289	-0.2289	-0.2289	-0.2289	-0.2289	-0.2289	-0.2289	-0.2289				
RESPONSE: -0.4558	-1.1525	0.23192	-0.033549	-0.033549	-0.033549	-0.8137	-1.1538	-0.58072	-0.77298	-0.34556	0.4893	1.2084	-1.4639	-10.172	0.9907	-0.92603	-0.66663	0.65593	-0.46983	-0.75254	-1.1483	-0.3629	0			
RESPONSE: -0.4558	-0.30469	-1.36453	-0.37397	-0.17164	-0.17164	-1.93264	-10.052	0.61769	-0.49728	-0.60792	-0.58583	-0.24593	-0.4386	-2.6374	0.81937	-0.88353	0.68335	0.23442	-0.41961	-1.90103	0.28289	-11.783	-23.792	-13.486		
RESPONSE: -0.4558	-0.7926	-0.9526	-0.08509	-0.08509	-0.08509	3.7941	-0.4	-0.31	-0.6983	-0.6983	0.047151	1.60	10	59	4	31	173	-5.5873	-0.6519	-0.173839	-0.71458	0.77680	18.648	0.6371	-0.453	0

**Figure 4:** Row completion test for a file from the MHEALTH dataset: here, the values in green are correct, in red are incorrect, and purple are extra predictions by the LLM. We randomly sample predict 25 rows from the file. Interestingly, for specific gyroscopes the values reproduced are accurate to multiple decimal places.

time	x	y	z	annotation
2016-11-09 14:22:31.120	-0.696257	-0.194557	0.683415	occupation;office and administrative support;1...
2016-11-09 14:22:31.130	-0.711980	-0.194557	0.683415	occupation;office and administrative support;1...
2016-11-09 14:22:31.140	-0.711980	-0.194557	0.683415	occupation;office and administrative support;1...
2016-11-09 14:22:31.150	-0.711980	-0.194557	0.683415	occupation;office and administrative support;1...
2016-11-09 14:22:31.160	-0.711980	-0.194557	0.683415	occupation;office and administrative support;1...
2016-11-09 14:22:31.170	-0.711980	-0.194557	0.683415	occupation;office and administrative support;1...
2016-11-09 14:22:31.180	-0.711980	-0.194557	0.683415	occupation;office and administrative support;1...
2016-11-09 14:22:31.190	-0.711980	-0.194557	0.683415	occupation;office and administrative support;1...
2016-11-09 14:22:31.200	-0.711980	-0.194557	0.683415	occupation;office and administrative support;1...
2016-11-09 14:22:31.210	-0.711980	-0.194557	0.683415	occupation;office and administrative support;1...
2016-11-09 14:22:31.220	-0.711980	-0.194557	0.683415	occupation;office and administrative support;1...
2016-11-09 14:22:31.230	-0.711980	-0.194557	0.683415	occupation;office and administrative support;1...
2016-11-09 14:22:31.240	-0.711980	-0.194557	0.683415	occupation;office and administrative support;1...
2016-11-09 14:22:31.250	-0.711980	-0.194557	0.683415	occupation;office and administrative support;1...
2016-11-09 14:22:31.260	-0.711980	-0.194557	0.683415	occupation;office and administrative support;1...
2016-11-09 14:22:31.270	-0.711980	-0.194557	0.683415	occupation;office and administrative support;1...
2016-11-09 14:22:31.280	-0.711980	-0.194557	0.683415	occupation;office and administrative support;1...
2016-11-09 14:22:31.290	-0.711980	-0.194557	0.683415	occupation;office and administrative support;1...
2016-11-09 14:22:31.300	-0.711980	-0.194557	0.683415	occupation;office and administrative support;1...
2016-11-09 14:22:31.310	-0.711980	-0.194557	0.683415	occupation;office and administrative support;1...

**Figure 5:** Snippet of data from the Capture-24 dataset: many timesteps of sensor data are identical, and therefore predicting the last row from ten previous rows, results in highly accurate text generation. This also means that GPT-4 can rely more on the available context to predict future rows, rather than reproduce memorized data.

data to LLMs, and instructing them to predict the activity. However, perturbation with a small value is also not a viable strategy, as Bordt *et al.* [6] showed that it results in a small performance reduction, if data were originally memorized.

The strategy adopted by Hota *et al.* [19] is a better option, as learned representations from SimCLR are used for binary activity recognition. This ensures that previously seen data is not used for evaluating performance. A simpler solution is to evaluate on non-public datasets, such that the possibility of data contamination is prevented entirely. For example, datasets like Mobiact [10] (which requires a form to be filled for obtaining data access) can be utilized, ensuring reproducibility while preventing data contamination. Potentially, a better approach for using LLMs with sensor data, is to integrate their training. Multi-modal methods such as AnyMAL

7	8	9	10	16	17	18	19	20
0.59324	0.48980	-0.88743	0.076621	1.22000	-0.78039	-0.006160	0.76078	-12.7560
0.24790	0.52319	-0.89681	0.047151	1.15090	-0.78039	-0.006160	0.76078	-13.1300
0.48895	0.52319	-0.89681	0.047151	1.17580	-0.78039	-0.006160	0.76078	-12.7360
-1.87770	0.52319	-0.89681	0.047151	0.27522	-0.82745	-0.028747	0.74353	-13.0750
-1.68160	0.52876	-0.85929	0.053045	-0.25730	-0.82745	-0.028747	0.74353	-14.4850
-2.14730	0.52876	-0.85929	0.053045	-0.43839	-0.82745	-0.028747	0.74353	-14.7180
-1.98260	0.52876	-0.85929	0.053045	-0.49867	-0.82745	-0.028747	0.74353	-14.2520
-1.80660	0.52876	-0.85929	0.053045	-0.58413	-0.84314	-0.030801	0.69181	-12.5110
-1.03380	0.52876	-0.87242	0.096267	-0.66508	-0.84314	-0.030801	0.69181	-10.2080
-0.22768	0.52876	-0.87242	0.096267	-0.81291	-0.84314	-0.030801	0.69181	-7.5096

**Figure 6:** Snippet of data from MHEALTH: for gyroscope data from the left ankle and right lower-arm (columns 8–10 and 17–19), we see repeating sensor values across timesteps. If one of these rows has to be predicted, GPT-4 can likely generate from the repeating values, leading to accurate reproduction. For clarity, we only show the gyroscope and adjacent columns.

[25] learn a joint embedding space between multiple modalities, and are capable of reasoning across input modalities, e.g., text, image, video, audio, and IMU motion sensor, and generating natural language responses via the trained LLaMa model [37]. Utilizing such methods which have IMU encoders brings many of the advantages of foundational LLMs, without the concerns about memorization.

## 6 CONCLUSION

In this paper, we studied recent trends in the wearables community, involving the use of foundational LLMs (e.g., GPT-4) to perform HAR [19, 20]. Generally, the potential of leveraging LLMs for HAR has been evaluated through publicly available sensor datasets, often in simplified, binary settings. Here, we examined the possibility of data contamination, wherein foundational models have potentially already been trained on public sensor datasets. We aimed to discover such contamination by employing the row completion test for memorization designed by Bordt *et al.* [6]. It involves prompting the LLM with a few prefix (or context) timesteps of sensor data, and instructing it to predict the next timestep verbatim. Accurate reproduction indicates memorization, i.e., the LLM has already seen public sensor datasets during training.

We applied this test to five public wearable sensor datasets and discovered that, potentially, GPT-4 has been likely trained on atleast one wearable sensor dataset – Daphnet FoG, where GPT-4 is able to produce sensor readings relatively accurately. For other datasets, it is harder for GPT-4 to cleanly reproduce sensor data (beyond replication). However, as Bordt *et al.* [6] discussed, the failure of the row completion test does not mean sensor datasets were not used during training; rather, the test is not successful at extracting memorized sensor data. We also discussed the challenges associated with applying these tests to sensor data, namely, repeating sensor values, which bias the LLM towards reproducing available values, rather than deriving from memory. Finally, we also presented the implications of the trend of utilizing LLMs for solving HAR, including the future trends for the community as a whole.

## REFERENCES

- [1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).
- [2] Janice Ahn, Rishu Verma, Renze Lou, Di Liu, Rui Zhang, and Wengpeng Yin. 2024. Large language models for mathematical reasoning: Progresses and challenges. *arXiv preprint arXiv:2402.00157* (2024).
- [3] Marc Bachlin, Meir Plotnik, Daniel Roggen, Inbal Maidan, Jeffrey M Hausdorff, Nir Giladi, and Gerhard Troster. 2009. Wearable assistant for Parkinson's disease patients with the freezing of gait symptom. *IEEE Transactions on Information Technology in Biomedicine* 14, 2 (2009), 436–446.
- [4] Oresti Banos, Rafael Garcia, Juan A Holgado-Terriza, Miguel Damas, Hector Pomares, Ignacio Rojas, Alejandro Saez, and Claudia Villalonga. 2014. mHealth-Droid: a novel framework for agile development of mobile health applications. In *International workshop on ambient assisted living*. Springer, 91–98.
- [5] Oresti Banos, Claudia Villalonga, Rafael Garcia, Alejandro Saez, Miguel Damas, Juan A Holgado-Terriza, Sungyong Lee, Hector Pomares, and Ignacio Rojas. 2015. Design, implementation and validation of a novel open framework for agile development of mobile health applications. *Biomedical engineering online* 14, 2 (2015), 1–20.
- [6] Sebastian Bordt, Harsha Nori, and Rich Caruana. 2024. Elephants Never Forget: Testing Language Models for Memorization of Tabular Data. *arXiv preprint arXiv:2403.06644* (2024).
- [7] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.
- [8] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. 2021. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*. 2633–2650.
- [9] S Chan Chang and A Doherty. 2021. Capture-24: Activity tracker dataset for human activity recognition. (2021).
- [10] Charikleia Chatzaki, Matthew Pediaditis, George Vavoulas, and Manolis Tsiknakis. 2017. Human daily activity and fall recognition using a smartphone's acceleration sensor. In *Information and Communication Technologies for Ageing Well and e-Health: Second International Conference, ICT4AWE 2016, Rome, Italy, April 21–22, 2016, Revised Selected Papers*. Springer, 100–118.
- [11] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*. PMLR, 1597–1607.
- [12] Aakanksha Chowdhery, Sharun Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research* 24, 240 (2023), 1–113.
- [13] John Joon Young Chung, Wooseok Kim, Kang Mi Yoo, Hwaran Lee, Eytan Adar, and Minsuk Chang. 2022. TaleBrush: Sketching stories with generative pretrained language models. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–19.
- [14] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. 2023. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378* (2023).
- [15] Michael Duan, Anshuman Suri, Nilofar Mireshghallah, Sewon Min, Weijia Shi, Luke Zettlemoyer, Yulia Tsvetkov, Yejin Choi, David Evans, and Hannaneh Hajishirzi. 2024. Do Membership Inference Attacks Work on Large Language Models? *arXiv preprint arXiv:2402.07841* (2024).
- [16] Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, et al. 2024. Olmo: Accelerating the science of language models. *arXiv preprint arXiv:2402.00838* (2024).
- [17] Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew G Wilson. 2024. Large language models are zero-shot time series forecasters. *Advances in Neural Information Processing Systems* 36 (2024).
- [18] Kai He, Rui Mao, Qika Lin, Yucheng Ruan, Xiang Lan, Mengling Feng, and Erik Cambria. 2023. A survey of large language models for healthcare: from data, technology, and applications to accountability and ethics. *arXiv preprint arXiv:2310.05694* (2023).
- [19] Aritra Hota, Soumyajit Chatterjee, and Sandip Chakraborty. 2024. Evaluating Large Language Models as Virtual Annotators for Time-series Physical Sensing Data. *arXiv preprint arXiv:2403.01133* (2024).
- [20] Sijie Ji, Xinzhе Zheng, and Chenshu Wu. 2024. HARGPT: Are LLMs Zero-Shot Human Activity Recognizers? *arXiv preprint arXiv:2403.02727* (2024).
- [21] Yubin Kim, Xuhai Xu, Daniel McDuff, Cynthia Breazeal, and Hae Won Park. 2024. Health-llm: Large language models for health prediction via wearable sensor data. *arXiv preprint arXiv:2401.06866* (2024).
- [22] Vladimir I Levenshtein et al. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, Vol. 10. Soviet Union, 707–710.
- [23] Xin Liu, Daniel McDuff, Geza Kovacs, Isaac Galatzer-Levy, Jacob Sunshine, Jiening Zhan, Ming-Zher Poh, Shun Liao, Paolo Di Achille, and Shwetak Patel. 2023. Large language models are few-shot health learners. *arXiv preprint arXiv:2305.15525* (2023).
- [24] Justus Mattern, Fatemehsadat Mireshghallah, Zhijing Jin, Bernhard Schölkopf, Mrinmaya Sachan, and Taylor Berg-Kirkpatrick. 2023. Membership inference attacks against language models via neighbourhood comparison. *arXiv preprint arXiv:2305.18462* (2023).
- [25] Seungwhan Moon, Andrea Madotto, Zhaojiang Lin, Tushar Nagarajan, Matt Smith, Shashank Jain, Chun-Fu Yeh, Prakash Murugesan, Peyman Heidari, Yue Liu, et al. 2023. Anymal: An efficient and scalable any-modality augmented language model. *arXiv preprint arXiv:2309.16058* (2023).
- [26] Milad Nasr, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A Feder Cooper, Daphne Ippolito, Christopher A Choquette-Choo, Eric Wallace, Florian Tramèr, and Katherine Lee. 2023. Scalable extraction of training data from (production) language models. *arXiv preprint arXiv:2311.17035* (2023).
- [27] Mustafa Safa Ozdayi, Charith Peris, Jack FitzGerald, Christophe Dupuy, Jimit Majmudar, Haidar Khan, Rahil Parikh, and Rahul Gupta. 2023. Controlling the extraction of memorized data from large language models via prompt-tuning. *arXiv preprint arXiv:2305.11759* (2023).
- [28] Thomas Plötz. 2021. Applying machine learning for sensor data analysis in interactive systems: Common pitfalls of pragmatic use and ways to avoid them. *ACM Computing Surveys (CSUR)* 54, 6 (2021), 1–25.
- [29] Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084* (2019).
- [30] Attila Reiss and Didier Stricker. 2012. Introducing a new benchmarked dataset for activity monitoring. In *2012 16th international symposium on wearable computers*. IEEE, 108–109.
- [31] Sandip Sarkar, Dipankar Das, Partha Pakray, and Alexander Gelbukh. 2016. JUNITMZ at SemEval-2016 task 1: Identifying semantic similarity using Levenshtein ratio. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. 702–705.
- [32] Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. 2023. Detecting pretraining data from large language models. *arXiv preprint arXiv:2310.16789* (2023).
- [33] Karan Singh, Shekoofeh Azizi, Tao Tu, S Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, et al. 2023. Large language models encode clinical knowledge. *Nature* 620, 7972 (2023), 172–180.
- [34] Dimitris Spathis and Fahim Kawsar. 2023. The first step is the hardest: Pitfalls of representing and tokenizing temporal data for large language models. *arXiv preprint arXiv:2309.06236* (2023).
- [35] Allan Stisen, Henrik Blunck, Sourav Bhattacharya, Thor Siiger Prentow, Mikkel Baun Kjærsgaard, Anind Dey, Tobias Sonne, and Mads Møller Jensen. 2015. Smart devices are different: Assessing and mitigating mobile sensing heterogeneities for activity recognition. In *Proceedings of the 13th ACM conference on embedded networked sensor systems*. 127–140.
- [36] Jiankai Tang, Kegang Wang, Hongming Hu, Xiyuxing Zhang, Peiyu Wang, Xin Liu, and Yuntao Wang. 2023. ALPHA: AnomaLous Physiological Health Assessment Using Large Language Models. *arXiv preprint arXiv:2311.12524* (2023).
- [37] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasamine Babaei, Nikolay Bashlykov, Soumya Batra, Prajwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288* (2023).
- [38] Johnny Tian-Zheng Wei, Ryan Yixiang Wang, and Robin Jia. 2024. Proving membership in LLM pretraining data via data watermarks. *arXiv preprint*

- arXiv:2402.10892* (2024).
- [39] Fanlong Zeng, Wensheng Gan, Yongheng Wang, Ning Liu, and Philip S Yu. 2023.  
Large language models for robotics: A survey. *arXiv preprint arXiv:2311.07226*

(2023).

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009