Sleep Quality Prediction from Lifelog Data Using LLM-based Imputation

JongYeol Hyun
a Seoul School of Integrated Sciences & Technologies
jacobgreen4477@gmail.com

YuChul Byun
Sogang University
Graduate School of AI·SW
byc3230@sogang.ac.kr

Dongkeun Bak
Sungkyunkwan University
dongkeun94@skku.edu

***Abstract*— This study proposes a methodology to enhance the classification performance of traditional machine learning models for lifelog data analysis by leveraging Large Language Models (LLMs) for missing value imputation. The reasoning capabilities of LLMs are used throughout the imputation process, and the resulting features are fed into an ensemble-based predictor. In our evaluation, the proposed approach** improved **classification accuracy on lifelog data compared with conventional methods.**

Keywords—Lifelog, Sleep, Sleep Quality, Stress Level, LLM, Missing Value Imputation, Machine Learning Ensemble

I. INTRODUCTION

With the rapid proliferation of smart devices and the increasing use of wearable technology, lifelog data—capturing diverse aspects of users' daily lives—is being generated at an unprecedented scale. Such data has become an essential resource in domains including personalized services, healthcare, and behavioral analysis. It provides objective indicators of quality of life by encompassing variables such as physical activity, heart rate, and sleep patterns. In particular, variations in physiological signals during daily activities and sleep offer valuable insights into sleep quality, emotional states, and stress levels.

Despite its potential, lifelog data often suffers from missing values and noise due to the inherent limitations of real-world data collection. Even with continuous monitoring via smartphones and smartwatches, missing or anomalous entries frequently occur. This problem is exacerbated by inconsistent device usage—for instance, users may not wear smartwatches during sleep—resulting in incomplete or fragmented sleep-related records. Such issues limit the reliability of downstream analyses and predictive modeling.

To address these challenges, this study proposes an imputation framework that leverages the inference capabilities of LLMs. By imputing missing values in sensor-derived lifelog data, we aim to enhance the informational quality of input features and thereby improve the accuracy of sleep quality prediction. Unlike conventional statistical methods that rely on numerical similarity or distance measures, LLMs can incorporate contextual knowledge, temporal dependencies, and domain-specific sleep rules. This enables them to generate plausible estimates for variables such as bedtime and wake time by considering weekday/weekend patterns, seasonal variations, and individual behavioral routines.

Through this approach, logically consistent and statistically valid sequences can be reconstructed, effectively mitigating the adverse effects of missingness. As a result, the usable sample size increases and predictive models can exploit a more complete dataset. Ultimately, this study contributes a robust imputation strategy that enhances both the accuracy and generalizability of sleep quality classification.

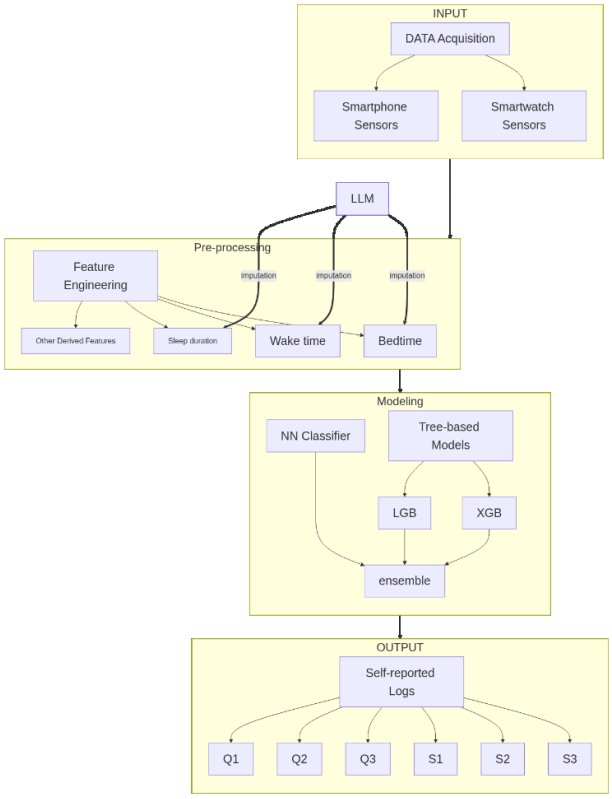

Fig. 1. Overall architecture of the LLM-based imputation and prediction model.

II. RELATED WORK

With the advancement of smart devices, research on utilizing various types of sensor data collected from individuals has become increasingly active. A study by Ribeiro et al. (2022) published in JMIR mHealth and uHealth provides a comprehensive review of "Lifelog" research, which involves collecting multimodal sensor data—such as images, audio, location, physical activity, and physiological signals—via wearables and smartphones to analyze personal experiences and behaviors. The paper highlights the use of multimodal data and retrieval/classification techniques in lifelog applications, including memory augmentation and behavioral understanding, and emphasizes that vision-based logs offer particularly rich information [1].

Meanwhile, the rapid advancement of LLMs and growing interest in their capabilities have spurred a wide range of research exploring their applications across various domains. In a study by Ding et al. (2024) published on arXiv, the authors propose a framework for addressing missing data in

recommendation systems using LLMs. By leveraging LLMs to predict and impute missing values, the framework overcomes the limitations of traditional mean or regression-based methods. The study demonstrates that LLM-based imputation improves recommendation accuracy across diverse classification and regression tasks, including single/multi-class classification and score prediction, outperforming conventional statistical techniques [2].

Research is also actively exploring the use of LLMs to automate de-identification of medical clinical data. Singh et al. (2025), in their study on arXiv, propose the RedactOR framework, which performs fully automated de-identification of multimodal Electronic Health Records (EHR), including both structured and unstructured text, as well as clinical voice data [3]. This study suggests that generative AI can be effectively integrated into real-world healthcare data pipelines.

In another study that proposed a framework for predicting and detecting individuals' physical and psychological states using lifelog data, the authors established a process that includes lifelog data collection, feature extraction, labeling, and model training. Based on this framework, they developed models to detect sleep quality, personality traits, mood, and depression. Experiments using real-world data confirmed that daily activity logs have a meaningful correlation with personal health conditions such as sleep and mood, and the overall results were promising [4].

Recent studies apply LLMs directly to imputation for time-series and tabular data. Jacobsen and Tropmann-Frick demonstrate that, with parameter-efficient fine-tuning (e.g., LoRA), LLMs can achieve performance competitive with dedicated deep-learning imputers (e.g., SAITS) on time-series imputation tasks [5]. In addition, GATGPT integrates an LLM with a spatiotemporal heterogeneous graph to jointly address anomaly detection, imputation, and prediction, and discusses the trade-offs between prompt-based and fine-tuned approaches [6].

Extending this line of research, our study proposes an LLM-based imputation framework tailored to the sleep lifelog setting, where missing values in key variables such as bedtime and wake time frequently arise due to device sparsity and inconsistent usage. By integrating contextual knowledge—including weekday/weekend patterns, seasonal effects, and domain-specific sleep guidelines—LLMs generate plausible and temporally consistent imputations that go beyond conventional statistical methods. These enriched features not only reconstruct logically coherent sleep sequences but also enhance the predictive performance of downstream ensemble classifiers, thereby offering a robust methodology for sleep quality prediction.

III. METHODOLOGY

A. Datasets

For this study, we utilized the competition-provided subset of the 2024 ETRI Lifelog Dataset, which consisted of 450 training samples and 250 public test samples. This subset was derived from the original dataset and distributed specifically for the challenge task.

The original 2024 ETRI Lifelog Dataset was collected by the Electronics and Telecommunications Research Institute (ETRI) from 10 participants using smartphones, smartwatches, sleep sensors, and self-reporting applications. In total, this process yielded approximately 700 participant-days of multimodal records, capturing both daily activities and sleep behaviors. The data collection protocol followed methodologies consistent with previous ETRI Lifelog studies [7–9].

The dataset includes 12 sensor-derived variables from smartphones and smartwatches (Table I), such as charging status, activity recognition, ambient sound, Bluetooth/Wi-Fi signals, GPS trajectories, illuminance, screen status, app usage, heart rate, ambient light, and step count. These multimodal features comprehensively represent both physical activity patterns and environmental context in everyday life.

TABLE I. LIFELOG DATA

Category	Data Item	Description
Smartphone-based Data	mACStatus	Charging status
	mActivity	Activity classification via Google Activity Recognition API
	mAmbience	Ambient sound labels and probabilities
	mBle	Nearby Bluetooth device information
	mGps	GPS coordinates
	mLight	Illuminance measured by smartphone
	mScreenStatus	Screen usage status
	mUsageStats	App usage logs and usage time
	mWifi	Nearby Wi-Fi device information
Smartwatch-based Data	wHr	Heart rate readings
	wLight	Ambient light
	wPedo	Step data

In addition, six sleep-related health indicators were derived from a combination of sleep sensor measurements and self-reported surveys (Table II). These indicators (Q1–Q3, S1–S3) capture both subjective and objective aspects of sleep health, fatigue, and stress. Each target was labeled on a daily per-participant basis and discretized into either binary (0/1) or ternary (0/1/2) levels, depending on the indicator. For example, Q1 denotes self-reported overall sleep quality after waking, while S1–S3 represent adherence to established sleep guidelines for total sleep time (TST), sleep efficiency (SE), and sleep onset latency (SOL), respectively. Classification criteria were predefined according to clinical and behavioral standards, and detailed guidelines are documented in the dataset release [10].

TABLE II. SELF-REPORTED INDICATORS(TARGETS)

Indicators	Description
Q1	Overall sleep quality as perceived by a subject immediately after waking up.
Q2	Physical fatigue of a subject just before sleep.
Q3	Stress level experienced by a subject just before sleep.
S1	Adherence to sleep guidelines for total sleep time (TST).
S2	Adherence to sleep guidelines for sleep efficiency (SE).
S3	Adherence to sleep guidelines for sleep onset latency (SOL, or SL).

B. Missing Data Types

Sensor data with missing values can be broadly categorized into three types. These scenarios represent different challenges in modeling lifelog data, as temporal continuity and contextual information are crucial for predicting sleep-related outcomes.

1) The previous day's data (D-1) is available, but the current day's input data (D-Day) is missing: In this case, the model must generate predictions without essential information required for accurate inference.

2) The previous day's data is missing, while the current day's input data is available: This limits the model's ability to reflect contextual patterns from prior data.

3) Both the previous and current day's input data are missing: In this case, the model cannot leverage learned temporal patterns, which may significantly degrade prediction performance.

C. Bedtime and Wake Time Inference

In this competition, only sensor data were provided, and unlike prior challenges [11], information on bedtime and wake time was unavailable. Nevertheless, previous studies have demonstrated that these variables—together with their difference (i.e., sleep duration)—are critical indicators of sleep quality and stress [12]. To compensate for this absence, a baseline model was first trained to evaluate feature importance. Based on the results and the observed frequency of missing values, mScreenStatus emerged as the most influential feature and was therefore selected as the primary source for inferring sleep-related variables.

Bedtime and wake time were derived from mScreenStatus using the following procedure:

1) Time Window Filtering: Screen activity records were restricted to the range 21:00–11:00, which typically encompasses the main sleep period.

2) Noise Reduction: Isolated screen-on events within screen-off intervals were removed, and short awake segments (≤2 minutes) surrounded by sleep blocks were reclassified as sleep to reduce fragmentation

3) Sleep Block Detection: Continuous screen-off intervals were detected, and the longest interval was designated as the primary sleep episode.

4) Feature Calculation: The start of the longest block was defined as bedtime (21:00–02:00), the end of the block as wake time (03:00–11:00), and their difference as sleep duration (excluded if <100 minutes).

Applying this rule-based procedure led to a substantial number of missing values, largely due to the inherent sparsity of sensor activity during sleep. In certain cases, entire days contained only target labels without any corresponding logs, underscoring a data quality challenge that degrades predictive performance. To address this limitation, LLMs were employed to impute missing bedtime and wake time entries by leveraging contextual patterns, statistical tendencies, and domain-specific knowledge. These imputed values were then incorporated as features for downstream prediction tasks.

D. LLM-Based Missing Value Imputation Process

The LLM-based missing value imputation procedure in this study was designed as a structured pipeline. The first step focused on selecting and configuring an appropriate model for stable inference.

1) Model Selection

All training and inference tasks were performed in a Google Colab Pro+ environment equipped with an NVIDIA A100 GPU to ensure sufficient computational capacity. For the imputation process, the Qwen/Qwen3-8B model was employed with bfloat16 precision, and inference was conducted using the vLLM library for efficient memory management and fast decoding [13].

The experiment was conducted under a standardized configuration to ensure consistency and reproducibility. Specifically, the maximum token length was fixed at 37,000, the temperature parameter was set to 0 in order to eliminate sampling randomness, and the random seed was set to 42 to guarantee reproducibility across runs.

2) Exploratory Data Analysis for Prompt Engineering

Exploratory analysis revealed several domain-specific behavioral patterns, which were explicitly embedded into the prompts:

- Weekend effect: Later bedtimes and wake times on Fridays and Saturdays.
- Seasonal effect: Earlier wake times during July and August.
- Sleep compliance effect (S1 = 2): Higher adherence to total sleep time guidelines was associated with delayed wake times.

These contextual insights were incorporated into the prompt design to guide the model in producing realistic imputations rather than relying on simple statistical averages.

3) Prompt Engineering for System Message

The system message provided global constraints and task context to the LLM. It specified:

- Task definition: Imputation of missing bedtime and wake time for sleep health prediction.
- Time representation: Decimal-hour format (e.g., 23.50 = 23:30, 0.75 = 00:45)
- Valid time ranges: Bedtime = 21.0–2.0, Wake time = 3.0–11.0 (decimal hours).
- Domain rules: Later wake times on weekends, earlier wake times in summer months, adjustments for S1 compliance.
- Data integrity: Preserve all observed values, fill only missing fields, and avoid empty outputs.

4) Prompt Engineering for User Message

The user message contained task-specific instructions and contextual information:

- Imputation sequence: (i) fill missing bedtime, (ii) impute wake time conditional on bedtime.
- Statistical references: Weekday- and month-specific averages to ensure temporal consistency.

5) Missing Value Generation and Validation

The results of LLM-based missing value generation, as illustrated in Figure 2, demonstrate the model's ability to infer bedtime with contextual precision using temporal information and weekday/month-specific averages. For instance, in the case of a missing value for July 10, 2024 (Wednesday), the LLM recognized that the date corresponded to a weekday in July and referred to the average bedtime for Wednesdays in July—approximately 23.3—to fill in the missing data.

During this process, the model identified that the date was not a weekend, thereby applying no weekend adjustment and instead directly using the average value. It also ensured that the generated value fell within the predefined valid bedtime range (21.0 to 2.0, based on a 24-hour decimal format).

Unlike traditional approaches that mechanically insert mean values, the LLM integrates contextual conditions (weekday, month, weekend status) and statistical references (average bedtime, time range constraints) to infer missing values. Notably, the model correctly interprets circular time logic, understanding that values beyond 24:00 (e.g., 1.5 = 1:30 AM) are still valid within the 21.0 to 2.0 range. This contextual awareness and adaptive generation highlight the distinct advantage of using LLMs over conventional imputation methods.

Let's take the first missing bedtime entry: 2024-07-10, which is a Wednesday in July. Looking at the statistics for July, the average bedtime for Wednesday is 23.2999999999999997. Since the user mentioned that the average bedtime for Wednesday in July is around 23.3, I can use that as a reference. However, I should also check if there are any other factors, like if it's a weekend or a holiday, but since it's a weekday, I can use the average.

Wait, but the user also mentioned that on weekends, people tend to go to bed later. However, July 10th is a Wednesday, so it's a weekday. Therefore, using the average bedtime for Wednesday in July (23.3) makes sense. But I need to ensure that the value is within the 21.0 to 2.0 range. Wait, 23.3 is within 21.0 to 2.0? Wait, no. Wait, the range for bedtime is 21.0 to 2.0, which is a circular range. So 21.0 to 2.0 would mean times from 9 PM to 12 AM, **but that doesn't make sense**. Wait, the user said that the average bedtime is in the range of 21.0~2.0 or 3.0~11.0 for wake time. Wait, maybe there's a misunderstanding here. Let me recheck the instructions.

The user mentioned that the average bedtime is in the range of 21.0~2.0, which is a circular time range. So 21.0 is 9 PM, 2.0 is 2 AM. So the bedtime is between 9 PM and 2 AM. Wait, that seems odd. But according to the example given, 1.4776 is 1 hour 28 minutes, which is 1:28 AM. So the bedtime is in the range of 21.0 (9 PM) to 2.0 (2 AM), which is a 3-hour window. That seems like a very short window for bedtime. Wait, maybe there's a mistake here. Alternatively, perhaps the range is 21.0 to 2.0, meaning that the bedtime is between 9 PM and 2 AM, but that's a 5-hour window. Wait, 21.0 is 9 PM, 2.0 is 2 AM, so the range is from 9 PM to 2 AM, which is 5 hours. But the user also mentioned that the average bedtime is in 21.0~2.0 or 3.0~11.0 for wake time. Wait, maybe the bedtime is in 21.0~2.0 (9 PM to 2 AM) and wake time is in 3.0~11.0 (3 AM to 11 AM). **That seems possible.**

Fig. 2. LLM inference log for bedtime and wake time imputation(abridged).

Building upon the final imputed values, this study performed additional feature engineering to enhance the predictive capability of the model. Prior research has demonstrated that variables such as bedtime, wake time, and total sleep duration are highly influential in predicting sleep quality, as they serve as indirect indicators of an individual's sleep behavior [12].

In line with these findings, we derived key temporal features, including bedtime, wake time, and sleep duration (bedtime – wake time). Furthermore, to capture longitudinal patterns and behavioral consistency, time-series features such as lag variables (e.g., previous day's bedtime and wake time) and rolling statistics (e.g., 3-day and 7-day moving averages of sleep duration) were constructed. These features were subsequently integrated into the training process to improve the robustness and accuracy of the predictive model.

TABLE III. LIST OF ADDITIONAL DERIVED FEATURES

Feature Name	Definition
Sleep Duration	The duration between bedtime and wake time.
Lag Features	Previous day's bedtime, wake time, and sleep duration
Rolling Features	Average sleep duration over the past 3 and 7 days

IV. EXPERIMENTS

A. Valid Dataset

From the 450 training samples, a validation set of 40 samples (approximately 8.9%) was carefully selected. The choice of this size was motivated by the following considerations:

The choice of Valid Size = 40 was primarily motivated by the following considerations:

1) Limited sample size of the dataset: Since the dataset is based on survey responses, the total number of training samples (450) is relatively small. To secure sufficient data for model training, it was necessary to minimize the proportion allocated to validation.

2) Balanced sampling across participants: The validation set was constructed by extracting four samples per participant from 10 participants, resulting in a total of 40 validation samples. This ensured that the validation set retained participant-level diversity rather than being biased toward specific individuals.

3) Distributional similarity to the public test set: The selected validation samples were chosen from periods temporally adjacent to the competition's public test dataset. This design aimed to maximize similarity in data distribution and temporal characteristics, thereby improving the reliability of validation results.

This validation design enabled both effective utilization of limited training data and reliable estimation of model generalization performance.

As illustrated in Fig. 3 and Fig. 4, the validation set exhibits a distribution of mean wake time and mean bedtime that is consistent with both the training and test sets. Although individual differences across participants (e.g., early vs. late sleepers) are clearly preserved, the overall temporal patterns of the validation data closely align with those of the test data. This further confirms that the validation design successfully

captured participant-level diversity while maintaining distributional similarity to the public test set.

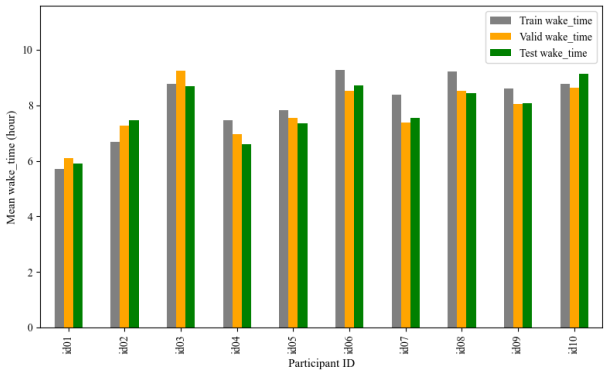

Fig. 3. Comparison of mean wake time by participant ID.

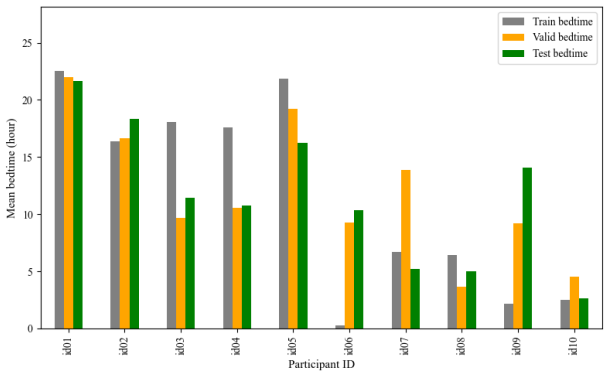

Fig. 4. Comparison of mean bedtime by participant ID.

B. Model Ensemble

To maximize predictive performance, we designed an ensemble comprising Light Gradient Boosting Machine (LightGBM; hereafter LGB), Extreme Gradient Boosting (XGBoost; hereafter XGB), and a Tabular Prior-Data Fitted Network (TabPFN). The ensemble prediction was computed as a weighted average of the three model outputs. [14-16]

A systematic grid search was employed under the constraint that the three model weights sum to one. Specifically, the search space was defined with a step size of 0.1 for each weight, resulting in 66 valid combinations (i.e., all integer multiples of 0.1 such that the sum of the weights for LGB, XGB, and TabPFN equals one). For each candidate weight set, the ensemble F1 score was evaluated on the validation dataset.

Table IV summarizes the top 5 combinations. The best-performing configuration was LGB(0.5), XGB(0.2), and TabPFN(0.3), which achieved an F1 score of 0.7218 on the validation dataset.

TABLE IV. TOP 5 COMBINATIONS FOR ENSEMBLE MODELS

Rank	LGB	XGB	TabPFN	Score
1	**0.5**	**0.2**	**0.3**	**0.7218**
2	0.4	0.1	0.5	0.7202
3	0.5	0.3	0.2	0.7176
4	0.3	0.2	0.5	0.7167
5	0.5	0.1	0.4	0.7163

C. Evaluation of Imputation Effectiveness

To quantitatively assess the effectiveness of missing value imputation, four strategies were compared: (1) retaining missing values without imputation (Original), (2) mean imputation, (3) KNN imputation, and (4) an LLM-based imputation. Each method was evaluated in terms of F1 score across six prediction targets (Q1, Q2, Q3, S1, S2, and S3). Experiments were conducted on a validation dataset of size 40, designed to ensure consistency in evaluating model generalizability.

The predictive model was implemented as an ensemble of LGB, XGB, and TabPFN, with initial weights set to (0.4, 0.3, 0.3). The optimal ensemble weights for each imputation method were determined through grid search.

As shown in Table V, the LLM-based Imputation method achieved the highest performance, with an average F1 score of 0.7218 and a top weighted F1 score of 0.7238. In contrast, the Original, Mean, and KNN approaches yielded lower average scores, confirming the superiority of the proposed method.

TABLE V. SUMMARY OF F1 SCORES AND OPTIMAL ENSEMBLE WEIGHTS (VALID SIZE: 40)

Method	F1	Top1 Weights (LGB, XGB, Tab)	Top1 F1
Original	0.6841	(0.1, 0.2, 0.7)	0.6998
Mean Impute	0.6684	(0.2, 0.0, 0.8)	0.7152
KNN Impute	0.6989	(0.2, 0.1, 0.7)	0.7116
LLM Impute	0.7218	(0.3, 0.0, 0.7)	0.7238

Table VI presents the F1 scores by target. The LLM-based Imputation method improved **S2 (sleep efficiency)** to 0.7749, which is approximately a 10 percentage-point increase over the Original method, and also achieved the highest score for S1 (0.524). Furthermore, it consistently outperformed other methods in Q1 and Q3, confirming the robustness of the approach.

TABLE VI. TARGET-WISE F1 SCORES (VALID SIZE: 40)

Target	Original	Mean Impute	KNN Impute	LLM Impute
Q1	0.716	0.693	0.697	**0.721**
Q2	0.780	0.811	**0.840**	**0.840**
Q3	0.715	0.680	0.715	**0.748**
S2	0.670	0.619	0.699	**0.775**
S3	**0.748**	0.723	0.723	0.723
S1	0.476	0.484	0.519	**0.524**

The results indicate that imputation strategies have a considerable influence on model performance, with the

LLM-based approach generally achieving better results across most indicators.

As shown in Tables V and VI, the LLM method was particularly effective in handling targets sensitive to contextual dependencies, such as S1 and S2. Unlike statistical or distance-based methods, the LLM-based approach leverages temporal order, weekday/weekend patterns, and domain-specific sleep guidelines to generate plausible missing values. This enables the reconstruction of logically consistent and statistically valid data, thereby enhancing the performance of downstream predictive models.

When compared to other strategies, mean imputation provided only modest improvements, while KNN imputation produced moderate gains. In contrast, the LLM-based method achieved higher F1 scores across most metrics, demonstrating consistent advantages.

These findings suggest that generative language models, by effectively capturing domain context, can serve as a reliable alternative for missing value imputation.

V. CONCLUSION

This study proposed a method for imputing missing values in sensor-based time-series lifelog data by leveraging the contextual understanding and reasoning capabilities of Large Language Models (LLMs). In experiments focusing on sleep quality prediction, the LLM-based imputation demonstrated improved performance compared to traditional methods such as mean substitution and K-Nearest Neighbors (KNN).

Unlike conventional approaches that simply replace missing values with averages or rely on similarity-based estimations, the proposed method incorporated diverse domain knowledge, including statistical summaries, weekday/weekend patterns, seasonal factors, prior-day context, and adherence to sleep guidelines. This enabled context-aware and temporally consistent imputations that more appropriately reflect actual human sleep behavior.

The experimental results suggest that LLM-based imputation can serve as a promising alternative for improving the quality of lifelog data, showing enhancements in both predictive performance and the logical consistency of reconstructed data. In particular, by embedding domain-specific insights obtained through exploratory data analysis into prompts, the model was guided to perform logical imputations rather than relying on simple averages.

Furthermore, the study confirmed that performance improvement is achievable through prompt-based inference even without additional training, indicating the potential for efficient utilization of LLMs in lifelog data analysis.

REFERENCES

[1] Ribeiro, R., Trifan, A., & Neves, A. J. (2022). Lifelog retrieval from daily digital data: narrative review. JMIR mHealth and uHealth, 10(5), e30517.

[2] Ding, Z., Tian, J., Wang, Z., Zhao, J., & Li, S. Data Imputation using Large Language Model to Accelerate Recommender System.

[3] Singh, P., Dzialo, C., Kim, J., Srivatsa, S., Bulu, I., Gadde, S., & Kenthapadi, K. (2025). RedactOR: An LLM-Powered Framework for Automatic Clinical Data De-Identification. *arXiv preprint arXiv:2505.18380*.

[4] Li, J., Ma, W., Zhang, M., Wang, P., Liu, Y., & Ma, S. (2021). Know yourself: physical and psychological self-awareness with lifelog. Frontiers in Digital Health, 3, 676824.

[5] M. Jacobsen and M. Tropmann-Frick, "Imputation Strategies in Time Series Based on Language Models," Datenbank-Spektrum, vol. 24, pp. 197–207, Oct. 2024.

[6] GATGPT: Language Model with Spatiotemporal Heterogeneous Graph for Spatiotemporal Tasks of Anomaly Detection, Imputation, and Prediction, arXiv:2412.09255, 2024 (preprint).

[7] S. Chung, S. Oh, S. Lee, and H. T. Jeong, "Real-world multimodal lifelog dataset for human behavior study," ETRI Journal, vol. 44, no. 3, pp. 426–437, Jun. 2022.

[8] S. W. Oh, J. H. Kim, Y. J. Kim, and H. T. Jeong, "Human Understanding AI Paper Challenge 2024—Dataset Design," arXiv preprint arXiv:2403.16509, 2024.

[9] S. W. Oh, S. Chung, J. H. Kim, Y. J. Kim, and H. T. Jeong, "Understanding Human Daily Experience Through Continuous Sensing: ETRI Lifelog Dataset 2024," arXiv preprint arXiv:2508.03698, 2025.

[10] ETRI, "ETRI Lifelog Dataset 2024," 2025. [Online]. Available: https://nanum.etri.re.kr/share/human/ETRILifelogDataset2024

[11] ETRI_Lifelog_Dataset_2020, 2021. URL https://nanum.etri.re.kr/share/schung/ETRILifelogDataset2020.

[12] T.-A. Song, S. R. Chowdhury, M. Malekzadeh, et al., "AI-driven sleep staging from actigraphy and heart rate," Plos one, vol. 18, no. 5, e0285703, 2023.

[13] K. Kwon, T. Dao, M. Zaharia, and P. Liang, "vLLM: Easy, fast, and cheap LLM serving with PagedAttention," arXiv:2309.06180, 2023

[14] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proc. KDD, 2016, pp. 785–794.

[15] G. Ke, Q. Meng, T. Finley, et al., "LightGBM: A highly efficient gradient boosting decision tree," in Adv. Neural Inf. Process. Syst., 2017.

[16] N. Hollmann, M. Sixt, et al., "TabPFN: A transformer that solves small tabular classification problems in a second," in Int. Conf. Learn. Representations (ICLR), 2023.