A Supplementary Material

A.1 Models details

Time-LogCosh-GAN (TLCGAN) is a traditional GAN model that uses two separate random noise inputs, z_1 and z_2, generated in a time-series format [16]. This model was trained using 10-fold stratified data over 200 epochs, with a learning rate of $lr = 0.0001$ and a batch size of 5. *The Time-series GAN* (TimeGAN) is a generative model trained adversarially and jointly. It uses a learned embedding space with both supervised and unsupervised losses. It comprises four network components: an embedding function, a recovery function, a sequence generator, and a sequence discriminator. In this study, we set the maximum sequence length and the hidden dimension size to 50. The model underwent training over 200 epochs with all other parameters set to their default values. *Transformer-based Time-Series Generative Adversarial Network* (TTS-GAN) architecture comprises two primary components: a generator and a discriminator. Both components are built using the transformer encoder architecture. The encoder comprises two compound blocks. The first block utilizes a multi-head self-attention module, while the second block is a feed-forward MLP with a GELU activation function. The model was trained for 200 iterations, with the sequence length matching the size of the temporal windows from the utilized dataset. All other parameters were kept at their default values. *The Structured State Space Diffusion SSSD^{S4}* (SSSD) model [13] is a diffusion model designed for time-series data, drawing inspiration from a generative model for audio - DiffWave [33]. The model was trained using the default configuration for 1,000 iterations. *DroppelGANger (DGAN)* is a synthetic data generation framework based on generative adversarial networks (GANs) [10], consisting of a metadata generator and a min/max generator. We adjusted it to set the max sequence length and sample length equal to the temporal window length from the used dataset. The batch size was 5, and the learning rate was $lr = 10^{-4}$ for both the discriminator and the generator. The model was trained for 200 epochs.

A.2 Prompting details

As mentioned, we selected three data instances from the desired class and the same dataset fold, resulting in nd-arrays with a shape of (3, temporal window, 3). We used these arrays as the context for the prompt given to the LLMs (Figure 2). All LLMs received the same samples as context to ensure fairness.

A.3 On the datasets imbalacement

Figure 6 displays two histograms representing the distribution of the two datasets, MHAD2 and MHEALTH. The x-axis likely represents different categories or classes, while the y-axis represents the count of observations within each class. In both datasets, some classes are significantly overrepresented (e.g., class 11 in MHAD1 and class 4 in MHAD2). This imbalance can lead to biased model

performance. For instance, a model might perform well in the majority classes but fail to learn adequately in the minority classes.

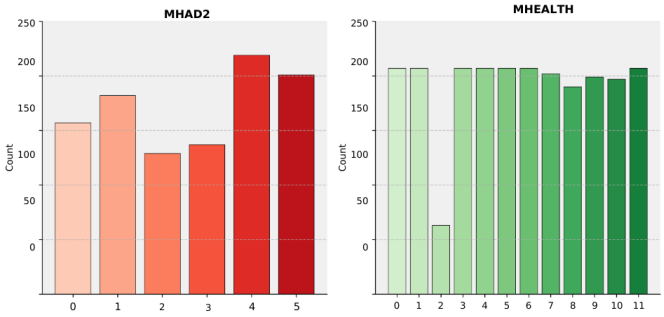

Fig. 6. The datasets are divided into stratified 10-folds, with the bars representing the mean number of samples per class across all folds.

For the MHAD2 dataset, the imbalance is more pronounced, with class 4 having the highest count (around 220), followed by class 5 with a slightly lower count. On the other hand, classes 2 and 3 have much lower counts (around 100), while classes 0 and 1 show an intermediate count. The imbalance could likely bias models towards predicting classes 4 and 5 more frequently, as they dominate the dataset. Minority classes like 2 and 3 could be underrepresented in the model's learning process, leading to poor classification for those specific classes. The imbalance in the MHEALTH model is less severe compared to MHAD2. Most classes (except class 2 and class 7) have around 200 examples. Class 2 is significantly underrepresented, with less than 50 examples, and class 7 has around 180 examples. Although there is still some imbalance, especially with class 2, it is not as extreme as in MHAD2. The model is likely to perform more evenly across most classes, though it might struggle to correctly classify examples of class 2 due to the low representation.

A.4 On overfitting

The Figure 7 is the complet eversion of the figure 4 in the main paper.

A.5 Factors affecting the LLMs

The Figure 8 is the complet eversion of the figure 5 in the main paper.

A.6 LLMs data's distribution

The original distribution of the data from fold zero and category zero can be compared with the synthetic data for the same fold and category. The synthetic

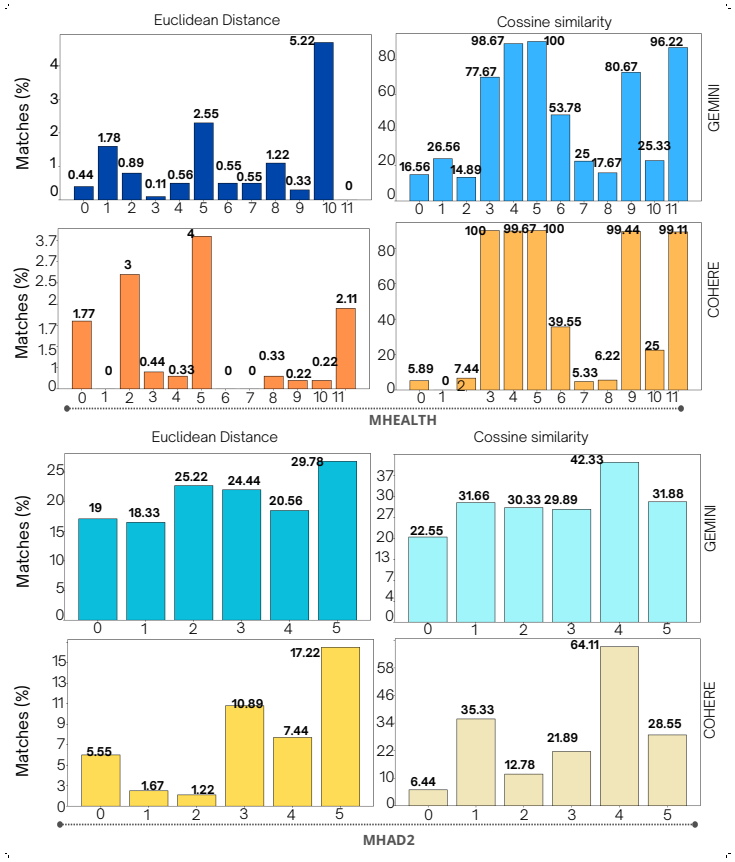

Fig. 7. The percentage of samples generated by the LLMs that match the training samples was evaluated. Using the Euclidean distance, we compared each generated sample to the three training samples used in the prompt. The same process was repeated using cosine similarity. More details about this experiment can be found in Section 4.4. The x-axis in the plots corresponds to the category, and the y-axis corresponds to the percent of the matches. The figure illustrates that, overall, less than 6% of the synthetic data generated for the MHEALTH dataset and 29% for the MHAD2 dataset are exact copies of the training samples. Meanwhile, the cosine similarity indicates that, at most, 60% of the synthetic data falls within the same distribution as the training data for MHAD2. For MHEALTH, there are notable variations.

data generally follow the distribution pattern of the real data, although discrepancies can be observed. For example, in the case of Gemini (see Figure 10), the synthetic data has a higher minimum value and a lower maximum than the original. In contrast, for Cohere and GPT-4, the minimum values are lower, as well as the maximum values. Notably, the data generated by Cohere and Gemini are quite similar, while GPT's data stands out with more divergence (see Figure 9). The real and synthetic data medians are quite similar for all three models. This

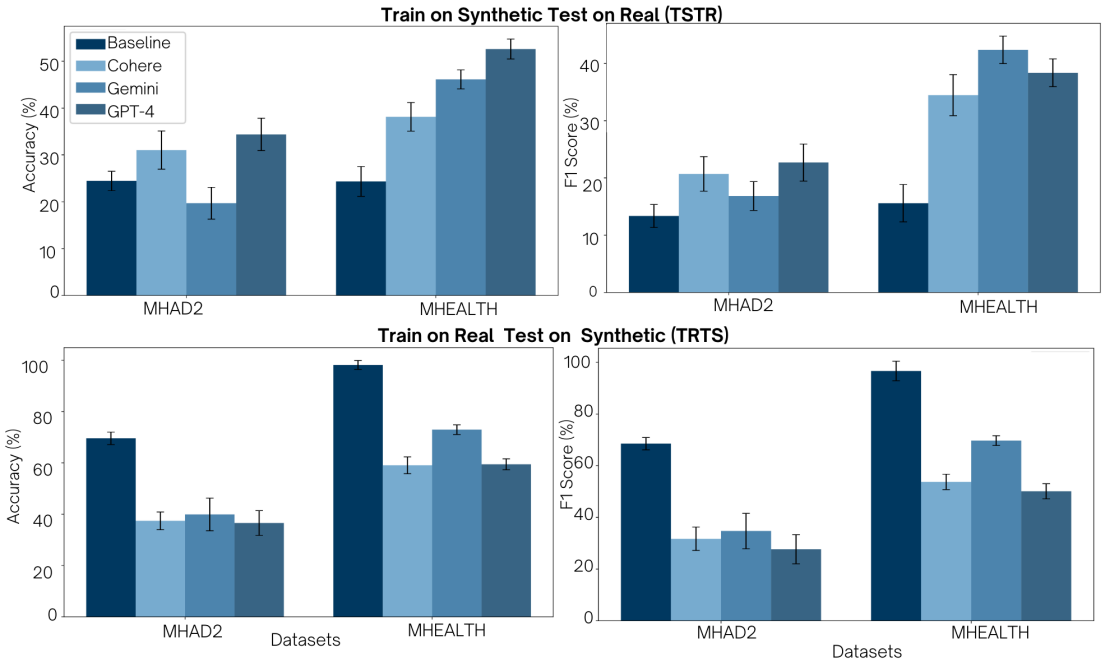

Fig. 8. The thin lines centered on the bars represent the confidence intervals, while the bars depict the assessed metrics. The y-axis shows the metric values, and the x-axis indicates the evaluated datasets. A notable disparity between accuracy and F1 scores highlights how class imbalance affects model performance. The Gemini model stands out, having the smallest gap between the two metrics.

suggests that the synthetic data generators have captured the central tendency of the real data. The IQR, which represents the spread of the middle 50% of the data, also seems comparable between real and synthetic data maintaining the overall variability of the real data.

However, the violin plots reveal differences in the distribution shapes between real and synthetic data. For instance, the real data for Gemini have a slightly wider distribution than the synthetic data. In contrast, Cohere's synthetic data has a slightly narrower distribution than the real data.

A.7 LLMs vs. Traditional models

TRTS and TSTR protocols In TRTS, we used the number of synthetic samples as the number of real test samples. Table 3 presents the performance of both methods. GPT-4 is not the best-performing model on both evaluation protocols. Still, it enhances the baseline by approximately 10% on the MHAD2 TSTR evaluation and almost 28% on the MHEALTH TSTR evaluation. It significantly outperforms SSSD (with an improvement of around 14%) and TTS-GAN (with

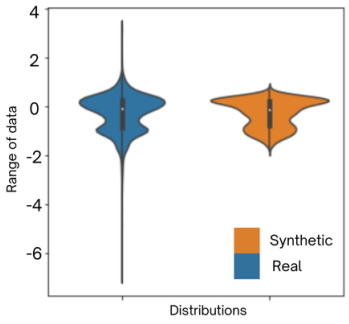

Fig. 9. Comparative analysis of GTP-4's synthetic (orange) and real (blue) distributions using the MHAD2 Dataset.

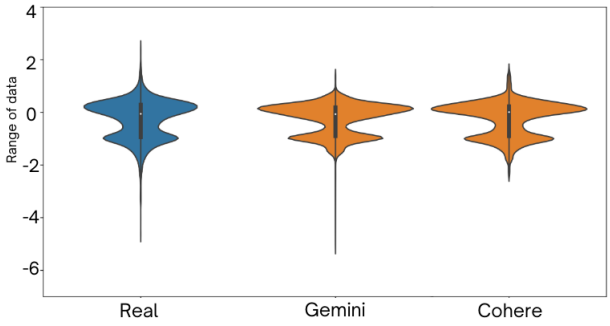

Fig. 10. Comparative analysis of synthetic (orange) and real (blue) distributions using the MHAD2 Dataset.

a gain of roughly 8%) on MHAD2 and TimeGAN (with a gain of roughly 32%), DGAN (more than 40%), and TTS-GAN (more than 30%) on MHEALTH. Regarding TRTS, it does not exceed the baseline but still proves superior to most models in the TSTR protocol. TLCGAN showed the best performance, exceeding the baseline in both scenarios in MHAD2. In MHEALTH, it was the second best on TSTR and the best on TRTS, even without outperforming the baseline.

Mixed Our analysis will concentrate on the data generated by GPT-4 (our primary focus) and TLCGAN (the baseline). Upon introducing synthetic samples to the MHAD2 training set (refer to figure 3), GPT-4 outperforms the baseline in two of the three demonstrated scenarios. In these scenarios, considering accuracy, it surpasses both TimeGAN and SSSD. However, when considering the arithmetic mean, TTS-GAN also falls within this category. TLCGAN demonstrated the highest performance, with nearly 3% more accuracy and a higher average. Concerning the MHEALTH dataset, GPT-4 outperforms the baseline in all scenarios (see table 4). Within these, it surpasses TLCGAN in two out

Table 3. Results of each model under the TRTS protocol.

Eval.	Dataset	Model	Accuracy	Recall	F1
TRTS	MHAD2	#Baseline	69.55±2.42	68.40±2.48	68.53±2.41
		TLCGAN	**69.17 ±4.59**	**68.20±4.81**	**68.17±4.94**
		DGAN	*55.83*±3.54	*54.81*±3.74	*53.96*±4.17
		TimeGAN	*68.33*±6.62	*65.96*±7.27	*60.96*±8.05
		SSSD	*17.92*±0.94	*19.38*±1.16	*10.58*±2.25
		TTS-GAN	20.73±3.55	23.29±3.98	12.21±3.78
		GPT-4	36.56±4.82	36.06±4.47	27.66±5.63
	MHEALTH	#Baseline	98.12±1.75	96.82±3.71	96.66±3.76
		TLCGAN	**78.26 ±3.77**	**78.28 ±3.78**	**74.79 ±4.53**
		DGAN	70.03±2.73	70.09±2.73	68.44±3.00
		TimeGAN	25.99±4.21	25.83±4.23	18.66±3.53
		SSSD	70.03 ±2.73	70.10±2.73	68.44±3.00
		TTS-GAN	8.92 ±0.49	8.80 ±0.48	1.88±0.47
		GPT-4	59.43 ±2.12	59.44 2.17	50.10 ±2.93
TSTR	MHAD2	#Baseline	24.46±2.07	23.81±2.26	13.37±2.01
		TLCGAN	**47.08 ±4.50**	**40.98 ±3.42**	**35.50 ±4.11**
		DGAN	45.63±6.16	38.36±4.93	32.77±6.42
		TimeGAN	45.52±2.83	37.64±2.83	32.45±3.18
		SSSD	20.73±2.73	19.80±2.53	8.92±2.17
		TTS-GAN	26.15±5.30	22.36±3.59	14.79±3.50
		GPT-4	34.38±3.46	29.86±2.20	22.69±3.23
	MHEALTH	#Baseline	24.33±3.17	22.87±2.80	15.61±3.26
		TLCGAN	55.25±3.37	51.81±1.59	49.23±2.20
		DGAN	11.61±2.73	14.13±1.91	7.40±2.11
		TimeGAN	20.80±2.56	19.22±2.03	13.17±2.22
		SSSD	**63.47 ±1.66**	**58.13 ±2.34**	**55.93 ±2.72**
		TTS-GAN	12.94 ±3.30	13.14 ±2.78	6.19 ±2.76
		GPT-4	52.61 ±2.13	52.50 ±2.78	38.37 ±2.42

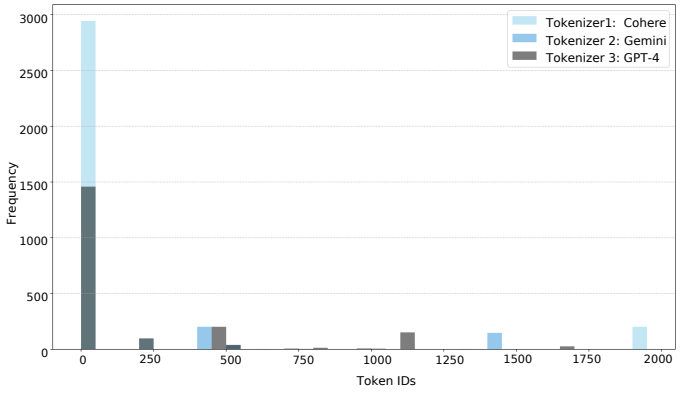

Fig. 11. Token distribution across different LLMs. Note that GPT-4 generates fewer tokens for the same sample compared to other models, indicating a more compact tokenization strategy.

Table 4. Results of each model under the Mixed protocol.

Dataset	Model	Added	Accuracy	Recall	F1
MHEALTH	Baseline	–	94.74±0.87	95.20±0.79	95.14±0.80
	TLCGAN	100	95.51±1.09	95.96±1.03	95.78±1.08
		150	95.42±0.57	95.88±0.54	95.76±0.58
		300	95.83±0.33	96.25±0.32	96.07±0.45
	DGAN	100	95.04±0.48	95.56±0.42	95.39±0.46
		150	95.34±0.75	95.82±0.71	95.72±0.74
		300	95.34±0.75	95.82±0.71	95.75±0.74
	TimeGAN	100	95.09±0.97	95.60±0.85	95.40±1.13
		150	95.39±0.63	95.85±0.59	95.75±0.58
		300	95.83±0.94	95.08±0.91	95.16±0.90
	SSSD	100	95.09±0.73	95.59±1.03	95.50±0.70
		150	95.72±0.81	96.1±0.78	96.13±0.79
		300	96.12±0.57	96.43±0.60	96.32±0.62
	TTS-GAN	100	95.30 ±1.03	95.75 ±0.93	95.68 ±0.95
		150	95.62 ±1.19	96.07 ±1.07	95.88 ±1.15
		300	96.04 ±0.91	96.44 ±0.83	96.41 ±0.84
	GPT	100	95.26 ±1.05	95.68 ±0.93	95.60 ±0.98
		150	95.46 ±0.87	95.86 ±0.84	95.78 ±0.82
		300	96.13 ±1.29	96.54 ±1.13	96.44 ±1.15
Dataset	**Model**	**Added**	**Accuracy**	**Recall**	**F1**
MHAD2	Baseline	–	69.55±2.42	68.40±2.48	68.53±2.41
	TLCGAN	100	70.36±3.18	69.69±3.606	69.39±3.77
		150	72.23±3.14	71.34±3.05	71.66±3.12
		300	70.45±2.79	69.61±2.97	69.33±3.05
	DGAN	100	70.98±1.88	70.27±2.23	70.39±2.4
		150	70.63±2.25	69.99±2.57	69.31±3.07
		300	68.75±4.08	68.52±3.99	67.84±4.71
	TimeGAN	100	70.63±3.11	69.88±3.29	70.08±3.46
		150	67.86±3.89	67.35±4.20	66.64±4.62
		300	69.02±3.15	68.91±3.39	68.36±3.57
	SSSD	100	68.93±2.65	67.97±3.00	68.20±2.83
		150	61.61±4.86	60.60±5.22	59.88±5.64
		300	67.32 ±2.83	66.69 ±3.26	66.13 ±3.23
	TTS-GAN	100	70.45±2.91	69.92±3.22	69.55±3.15
		150	70.36±2.68	69.27±3.19	69.77±3.05
		300	68.84±2.51	68.10±2.71	67.77±3.09
	GPT	100	70.54±2.59	69.57±3.06	69.79±3.14
		150	70.80±2.98	69.81±3.19	69.84±3.68
		300	68.57±2.23	67.88±1.39	67.53±2.88

of three scenarios, thus becoming the top-performing model. This holds both in terms of average and accuracy, implying that GPT's performance is notably high since the average was maintained and the datasets are unbalanced. The performance of both models varies according to the amount of data added, and depending on the dataset, it may negatively affect model performance as more synthetic data is added. Too many synthetic samples might introduce noise or overfitting, diminishing the benefits of augmentation (see table 4 for the complete results in this evaluation). Even though it is not the top-performing model in all scenarios, GPT-4 is competitive compared to other models. Like the others, augmenting data improves the model's performance to a certain extent, suggesting a consistency in the results produced when employing this model.

A.8 About the tokens

Insights from Tokenization and Model Performance Figure 11 confirms our understanding of how LLMs tokenize time-series data. Notably, the number of tokens varies across models which impacts their performance.