Fully Test-Time Adaptation for Feature Decrement in Tabular Data

Zi-Jian Cheng[1,2] , **Zi-Yi Jia**[1,2] , **Kun-Yang Yu**[2,3] , **Zhi Zhou**[2,3] and **Lan-Zhe Guo**[1,2*]
[1]School of Intelligence Science and Technology, Nanjing University, China
[2]National Key Laboratory for Novel Software Technology, Nanjing University, China
[3]School of Artificial Intelligence, Nanjing University, China
{chengzj,yuky,zhouz,guolz}@lamda.nju.edu.cn, jiazy@smail.nju.edu.cn

Abstract

Tabular data is widely adopted in various machine learning tasks. Current tabular data learning mainly focuses on closed environments, while in real-world applications, open environments are often encountered, where distribution shifts and feature decrements occur, leading to severe performance degradation. Previous studies have primarily focused on addressing distribution shifts, while feature decrements, a unique challenge in tabular data learning, have received relatively little attention. In this paper, we present the first comprehensive study on the problem of **Fully Test-Time Adaptation for Feature Decrement in Tabular Data**. Through empirical analysis, we identify the suboptimality of existing missing-feature imputation methods and the limited applicability of missing-feature adaptation approaches. To address these challenges, we propose a novel method, LLM-IMPUTE, which leverages Large Language Models (**LLMs**) to **impute** missing features without relying on training data. Furthermore, we introduce **A**ugmented-**T**raining **LLM** (ATLLM), a method designed to enhance the robustness of feature decrements by simulating feature-decrement scenarios during the training phase to address tasks that can not be imputed by LLM-IMPUTE. Extensive experimental results demonstrate that our proposal significantly improves both performance and robustness in missing feature imputation and adaptation scenarios.

1 Introduction

Tabular data [Altman and Krzywinski, 2017], a highly structured data format, organizes information in rows and columns, where each row represents an independent sample or instance, and each column corresponds to a specific feature or attribute [Sahakyan *et al.*, 2021]. Tabular data is extensively utilized in real-world applications. For instance, tabular data supports financial tasks such as credit scoring [West, 2000] and stock market prediction [Zhu *et al.*, 2021]. Tabular data also facilitates medical applications including disease diagnosis [Yıldız and Kalayci, 2024] drug development [Meijerink *et al.*, 2020]. To fully leverage its potential, machine learning research has designed various models tailored to tabular data, ranging from traditional tree-based models (e.g., CatBoost [Prokhorenkova *et al.*, 2018] and XGBoost [Chen and Guestrin, 2016]) to emerging deep-learning models (e.g., SwitchTab [Wu *et al.*, 2024] and TabPFN [Hollmann *et al.*, 2025]). These models have demonstrated exceptional performance across diverse tabular tasks.

Most existing tabular machine learning models are trained and evaluated in closed environments where data distribution and feature spaces between training and testing phases are consistent. However, real-world applications of tabular data often occur in open environments [Zhou, 2022], where distribution shifts and feature decrement between training and testing phases are prevalent [Guo *et al.*, 2025]. For example, in natural disaster prediction, training data may originate from a specific region, while the testing phase involves data from different regions. Similarly, in recommendation systems, user interests evolve dynamically over time, necessitating that models dynamically adjust their recommendation strategies to adapt to different data distributions. These challenges in open environments highlight the limitations of models developed for closed environments, necessitating the development of more robust and adaptable tabular models.

A prominent research direction addressing challenges in open-environment is fully test-time adaptation (FTTA). FTTA aims to enhance the performance of pre-trained models during the fully test-time phase, where training data is unavailable. Recent research has proposed various FTTA algorithms. For example, Tent [Wang *et al.*, 2021] adapts models by updating batch normalization parameters. FTAT [Zhou *et al.*, 2025] introduces a confident distribution optimizer, a local consistency weighter, and a dynamic model ensembler to optimize the adaptation process. However, these FTTA algorithms primarily focus on distribution shifts [Shao *et al.*, 2024], assuming consistent feature space between training and testing phases and fail to address feature decrement inherent to tabular data, which restricts their effectiveness in open environments where feature decrements occur.

Feature decrement refers to the reduction in feature dimensions during the testing phase compared to those available in the training data. For instance, in weather prediction systems, certain primary sensors may cease transmitting data without

[*] Corresponding author.

replacement by new sensors, resulting in a degradation of features. Current methods for addressing feature decrements can be categorized into two types: missing-feature imputation and missing-feature adaptation. Missing-feature imputation aims to impute missing feature values to maintain consistent input dimensions, while missing-feature adaptation enables themselves to dynamically adjust to feature-decrement scenarios. However, the majority of missing-feature imputation methods depend on training data to impute, rendering them unsuitable for FTTA scenarios. Furthermore, existing missing-feature adaptation approaches often face significant challenges in achieving effective adaptation to FTTA scenarios. This indicates significant limitations of both approaches when handling feature decrements in FTTA scenarios.

It is evident that the existing FTTA algorithms, primarily designed for distribution shifts, cannot be applied in feature decrements, and the feature decrement methods cannot be applied in FTTA scenarios. Hence, it is urgent to research FTTA in feature-decrement scenarios to propose more robust and adaptive solutions. To this end, this paper conducts a systematic investigation into FTTA for feature decrement in tabular data for the first time.

In this paper, we introduce the problem of **Fully Test-Time Adaptation for Feature Decrement in Tabular Data** and conduct a systematic empirical investigation. First, we define the problem of fully test-time feature decrements and evaluate existing methods in fully test-time feature-decrement scenarios. We compare missing-feature imputation methods, finding no significant performance improvement compared with random-value imputation, indicating that imputed values from missing-feature imputation methods are not optimal. We also observe that missing-feature adaptation approaches exhibit poor robustness in fully test-time feature-decrement scenarios, with performance significantly degrading as the degree of feature decrement increases. To address the suboptimality of existing missing-feature imputation methods, we propose **LLM-IMPUTE**, a method leveraging LLMs to generate imputed values without relying on training data. To tackle the limited applicability of missing-feature adaptation approaches, we introduce **ATLLM**, a model specifically designed for feature-decrement scenarios. ATLLM enhances robustness through an augmented-training module tailored to feature-decrement scenarios. LLM-IMPUTE and ATLLM form a comprehensive framework, significantly improving FTTA algorithms' performance and robustness in feature-decrement scenarios.

Our contributions are summarized as follows:

- **Problem.** We find that current FTTA algorithms are tailored to distribution shifts and inapplicable in feature decrements, while current methods designed for feature decrements remain unevaluated in fully test time.
- **Analysis.** We conduct extensive empirical analysis and identify the suboptimality of missing-feature imputation methods and the limited applicability of missing-feature adaptation approaches.
- **Method.** We propose LLM-IMPUTE, by utilizing LLMs to impute missing features without training data, and introduce ATLLM to enhance robustness of feature decrements by simulating feature-decrement scenarios.
- **Evaluation.** Comprehensive experiments on 9 datasets demonstrate that proposed FTTA methods exhibit significant improvements in performance and robustness in feature decrements over 11 comparison models.

2 Related Work

2.1 Fully Test Time

Fully test time is first introduced in Tent [Wang *et al.*, 2021], which aims to enhance the performance of pre-trained models when training data is unavailable during the test phase. Various FTTA algorithms have been proposed. For instance, Tent [Wang *et al.*, 2021] achieves adaptation by updating the batch normalization parameters of the model. EATA [Niu *et al.*, 2022] further enhances this approach by incorporating active sample selection and weighting strategies to improve adaptation efficiency. FTAT [Zhou *et al.*, 2025] introduces a confident distribution optimizer, a local consistency weighter, and a dynamic model ensembler to refine the adaptation process. However, current FTTA algorithms predominantly addresses distribution-shift scenarios and overlooks challenges posed by feature decrements. To address this gap, we focus on the problem of FTTA for feature decrement in tabular data. To the best of our knowledge, our work is the first to systematically define and analyze this problem.

2.2 Tabular Machine Learning

Tabular data refers to structured and heterogeneous data, which is widely utilized in domains such as medical diagnostics, financial analytics, and social sciences [Borisov *et al.*, 2022; Kadra *et al.*, 2021; Shwartz-Ziv and Armon, 2022]. Current models designed for tabular data can be broadly categorized into two main types: tree-based models and deep-learning models. Tree-based models excel at handling irregular patterns and non-informative features within the objective function, offering robust performance in scenarios where data lacks rotational invariance [Grinsztajn *et al.*, 2022]. While with the rapid advancement of deep learning, numerous deep-learning-based models tailored for tabular data have emerged, including SwitchTab [Wu *et al.*, 2024] and TabPFN [Hollmann *et al.*, 2025]. Although in closed environments both types achieve excellent performance [Jia *et al.*, 2024], they have not been fully evaluated in feature-decrement scenarios.

2.3 Feature Engineering on LLMs

Recent studies have explored feature engineering for tabular datasets by leveraging LLMs. This field primarily focuses on two aspects: feature generation and feature selection. (1) Feature generation is a critical process in feature engineering that aims to derive meaningful features from raw data without manual intervention. For example, CAAFE [Hollmann *et al.*, 2024] introduces a context-aware feature engineering framework that utilizes LLMs to generate semantically meaningful features based on the task description. In contrast, OcTree [Nam *et al.*, 2024] eliminates the need for manually defining the search space and leverages the optimization and inference capabilities of LLMs to discover effective feature generation rules. (2) Feature selection involves identifying

Method Type	Method	Representative Models
Specific Missing-feature Imputation Methods	Treat missing-feature values as feature minima	CatBoost
	Left subtree split	XGBoost, LightGBM
	Missing-feature values as a separate category	TabTransformer
Common Missing-feature Imputation Methods	Impute missing features by 0	Most deep-learning models

Table 1: Representative missing-feature imputation methods.

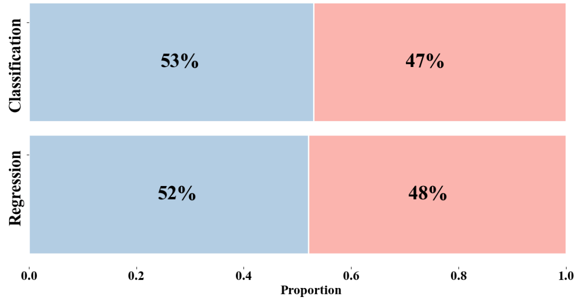

Figure 1: Methods performance comparison: missing-feature imputation vs. random-value imputation.

and selecting the most relevant features from a dataset. With extensive prior knowledge, LLMs can analyze and determine which feature is crucial for a specific goal. For instance, LMPriors [Choi *et al.*, 2022] prompts an LLM to evaluate each candidate feature by predicting its relevance to the target variable [Li *et al.*, 2024]. However, there is a notable lack of approaches that leverage LLMs to impute missing features.

3 Problem Formulation and Analysis

In this section, we first present a detailed definition and formalization of FTTA for feature decrement in tabular data. Then, we provide a comprehensive analysis on this problem, exploring the impact of fully test-time feature-decrement scenarios on missing-feature imputation and missing-feature adaptation approaches.

3.1 Problem Formulation

Formally, the goal of tabular prediction tasks is to train a machine-learning model $f : \mathcal{X} \to \mathcal{Y}$, where $\mathcal{X}$ is the input space and $\mathcal{Y}$ is the output space. We define the set of features in $\mathcal{X}$ as C.

Feature Decrement. We define the feature sets C for training and testing data as C^{train} and C^{test}, respectively. If there's no feature decrement, the feature sets from training and testing data are identical, i.e., $C^{train} = C^{test}$. While in feature-decrement scenarios, some features in testing data are missing, i.e., $C^{test} \subsetneqq C^{train}$.

Fully Test Time. Fully test time refers to the scenario where, during the testing phase, only the trained model f and testing data are available, while training data is inaccessible.

FTTA for Feature Decrement in Tabular Data. This paper tackles the research problem of FTTA for feature decrement in tabular data, focusing on how to improve the robustness of tabular data learning algorithms for the feature decrement problems, without relying on training data during the test phase.

3.2 Analysis

We begin the study by analyzing the performance and robustness of existing missing-feature imputation methods and missing-feature adaptation approaches in fully test-time feature decrement scenarios.

Missing-feature imputation methods aims to impute missing features to maintain consistency between dimensions of training and test inputs. Current missing-feature imputation methods adopt one of two primary approaches for imputing missing features to ensure normal prediction functionality. The first approach involves utilizing the model's self-imputation module specifically designed to handle missing features. For instance, CatBoost [Prokhorenkova *et al.*, 2018] processes missing values by treating them as the minimum value of the feature. Similarly, other tree-based models, such as XGBoost [Chizat *et al.*, 2020] and LightGBM [Badirli *et al.*, 2020], typically follow a default strategy of assigning missing values to the left child node. TabTransformer [Huang *et al.*, 2020], leveraging its deep learning architecture, imputes missing features by treating them as a distinct category. The second approach involves imputing missing features with a constant value, commonly zero. This approach is widely used in various deep-learning models as a simple method for imputing missing data. Table 1 provides a comparison of representative models employing these two missing-feature imputation methods.

Missing-feature adaptation approaches can directly predict test data with missing features, eliminating the need for imputing missing values, thereby avoiding potential biases or errors introduced by imputation methods. These approaches are designed to adapt to feature-decrement scenarios by leveraging the available features, without requiring additional preprocessing steps such as imputing. By focusing on the intrinsic relationships within the observed data, missing-feature adaptation approaches can maintain model performance even when some features are absent.

However, these methods have not been evaluated in FTTA scenarios. Therefore, we conduct empirical experiments to assess their efficacy and identify two observations.

Observation 1: The suboptimality of missing-feature imputation methods. To address the question of whether current missing-feature imputation methods are effective in FTTA scenarios, we compare model performance under two imputation strategies: missing-feature imputation methods

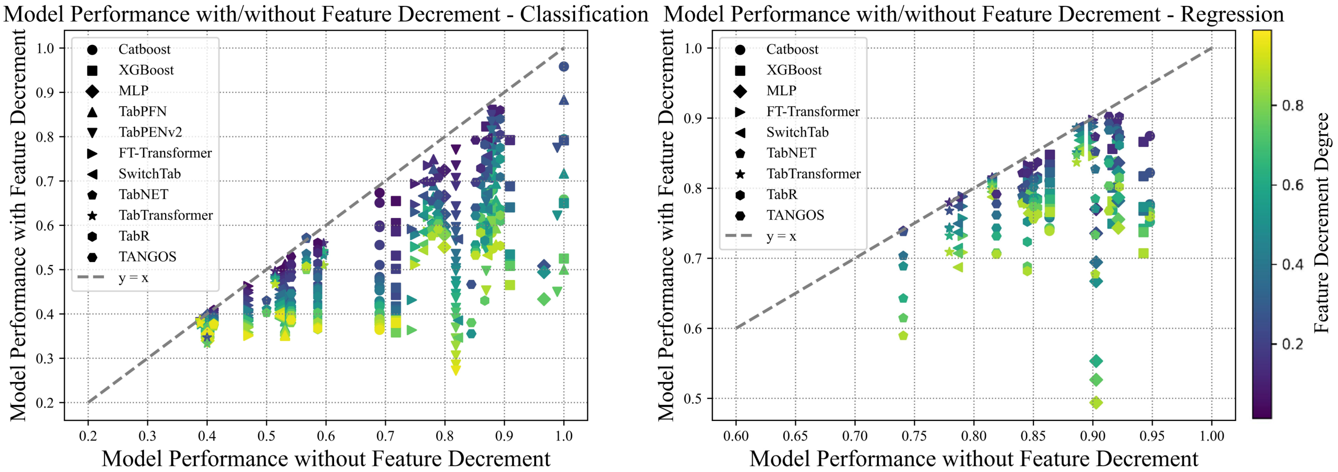

Figure 2: Missing-feature adaptation approaches performance decreases from different degrees of feature decrement.

and random-value imputation. We choose four types of missing-feature imputation methods to evaluate, including left subtree split, feature minima, separate category and zero. Random-value imputation means to impute missing values with randomly generated numbers. Experimental results, depicted in Figure 1, reveal minimal performance differences between these two strategies. It suggests that generated imputation values by missing-feature imputation methods do not significantly alleviate adverse effects of feature decrement on model performance. In essence, these imputed values are nearly equivalent to random values, underscoring their suboptimality. It is evident that rather than being specifically designed to address feature decrements, missing-feature imputation methods appear to function primarily as a technical requirement to align training and testing inputs.

<u>Observation</u> 2: The limited applicability of missing-feature adaptation approaches. Current missing-feature adaptation approaches can be divided into tree-based models and deep-learning models. We choose CatBoost, XGBoost and 8 current deep-learning models, including SwitchTab, TabPFN, etc. To evaluate whether these approaches can efficiently handle feature decrements in FTTA scenarios, we compare their performance with and without feature decrements. Experimental results, illustrated in Figure 2, can be summarized as follows: (1) Model performance degradation with feature decrement in FTTA scenarios. Scatter points below the $y = x$ line indicate a decrease in model performance from without feature decrement to with feature decrement, confirming that feature decrement negatively impacts model performance in FTTA scenarios. (2) Correlation between performance degradation and feature-decrement degree. The color gradient of the scatter points, where brighter points represent larger deviations from the $y = x$ line, reveals a strong positive correlation between the degree of feature decrement and the extent of performance degradation in FTTA scenarios. Larger deviations are associated with more significant declines in model performance. (3) Limited robustness of models. While TabTransformer (represented by star-shaped markers) shows relatively smaller deviations from the $y = x$ line, it is still not immune to feature decrement. This highlights the limited applicability of existing missing-feature adaptation approaches to adapt to FTTA scenarios.

These findings highlight limitations of existing missing-feature imputation methods and adaptation approaches in FTTA scenarios, and emphasize the need for solutions tailored to fully test-time feature-decrement scenarios. Therefore, we shift our focus to FTTA algorithms and explore the potential of LLMs for addressing fully test-time feature decrements. LLMs possess two key advantages as FTTA algorithms for handling feature decrements:

- **Diverse and Rich Prior Knowledge.** Pre-trained on large-scale datasets, LLMs possess a vast repository of knowledge encompassing factual information, linguistic conventions, cultural contexts, and common sense [Zhao *et al.*, 2023]. Additionally, LLMs demonstrate a nuanced understanding of complex concepts and terminology across various domains [Chang *et al.*, 2024]. This extensive knowledge base enables LLMs to infer relevant patterns and relationships from limited information, effectively addressing feature-decrement challenges by leveraging contextual understanding.
- **Input without the Need for Imputing.** Unlike models which require fixed-dimension input, LLMs accept text inputs of variable lengths [Lamb *et al.*, 2024]. This enables LLMs to handle missing features during testing without requiring imputations. For instance, if some features are unavailable during testing, LLMs can still process the input, albeit with shorter text sequences compared to the training phase. This inherent flexibility allows LLMs to maintain their predictive capabilities even with incomplete data.

Therefore, we propose two FTTA algorithms based on LLMs to handle problems of the suboptimality of existing

You are a data retrieval expert who can call on any resource. Given the task background description and the specific meaning of the feature names, you need to develop values for each feature to fill in when feature values are missing.

Background: < Task Description > .
Goal: < Task Target > .
Features: < Features Description > .

Please analyse this task and solve it step by step.

Step 1: Analyse. Based on the common sense , state in sentences the causal relationship or trend between each feature and the task description, and give a range of values for features that are usually prevalent in real life.

Step 2: Impute. Based on the information above and the answers from Step 1, identify a value for each feature to impute when this feature is missing. The value imputed should be reasonable and within the range given in Step 1.

Output format: "Feature" : Value

Answer:

Figure 3: A prompt template for the LLM-IMPUTE method.

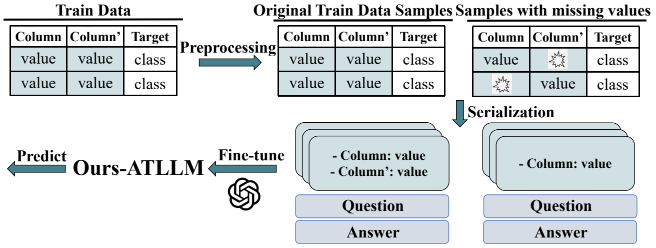

Figure 4: Overview of ATLLM.

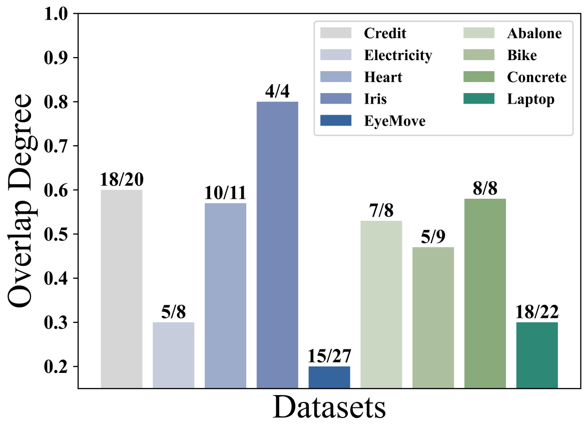

Figure 5: Overlap degree between feature ranges from LLM-IMPUTE and actual datasets. The numbers above bars indicate how many of imputed values provided by LLM-IMPUTE are within feature ranges of the actual dataset.

missing-feature imputation methods and the limited applicability of missing-feature adaptation approaches.

4 Methodology

In this section, we introduce the LLM-IMPUTE approach to address the inadequacies of current missing feature imputation methods and develop the ATLLM framework to broaden the applicability of missing feature adaptation strategies. The significance of LLM-IMPUTE and ATLLM in research and real-world applications is detailed in Appendix C. Existing imputation methods often yield suboptimal results. To achieve more accurate imputations, we utilize LLM APIs, which, being closed-source, are accessible even when training data are unavailable. Figure 3 illustrates the process by which LLM-IMPUTE imputes missing features. To enhance the accuracy of LLM-generated imputations, we mimic the reasoning process of human experts in tabular prediction tasks by employing a step-by-step guidance mechanism based on prompt learning and chains of thought [Wei *et al.*, 2022]. The input comprises three key elements:

4.1 LLM-IMPUTE

Task Description. The background information and target of a given task are first integrated with descriptions of features (as shown in red text in Figure 3). Notably, because there is no access to training data, we refrain from providing information such as data distributions or the minimum/maximum value of each feature, which are commonly included in feature-engineering methods.

Inference Process. The inference process of LLMs is divided into two steps (see green text in Figure 3).

- **Step 1 - Analyse.** The first step directs LLMs to utilize its internal knowledge, in conjunction with the information provided in the dataset, to infer the relationships between features and the task at hand and to give a realistic range of feature values. This step ensures that LLMs fully leverage prior knowledge about the task.
- **Step 2 - Impute.** In the second step, LLMs are required to output a specific value for each feature, based on the value range derived in Step 1. This output is then used to impute the missing features. In this step, LLMs must analyze the relevance of the range of values proposed in the first step and select an appropriate and generalized value from within that range.

Output Format. To facilitate the parsing and utilization of imputed values, we guide LLMs in structuring its responses through explicit instructions (see blue text in Figure 3).

Models	Credit		Electricity		Heart		Iris		Eyemovements		Abalone		Bike		Concrete		Laptop	
	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	nRMSE	MAE	nRMSE	MAE	nRMSE	MAE	nRMSE	MAE
Catboost	+5.26	+7.02	+1.07	+31.3	+0.83	+0.35	+0.00	-2.72	+2.07	+1.54	-0.014	-0.215	-0.059	-62.39	-0.065	-4.367	-0.003	-6.380
XGBoost	+8.08	+11.5	+2.27	+1.11	+4.85	+8.44	+3.70	+3.84	+3.39	+4.40	-0.056	-1.689	-0.022	-9.301	-0.076	-5.631	-0.146	-874.4
MLP	+7.07	+12.2	+1.78	+14.5	+12.7	+12.1	+18.4	+25.6	+1.79	+2.77	-0.044	-1.031	-0.026	-18.95	-0.196	-24.02	-0.017	-28.39
TabPFN	+3.16	+0.05	+1.27	+14.9	+6.01	-1.29	+2.53	-1.44	+1.50	+4.90	\	\	\	\	\	\	\	\
TabPFNv2	+2.01	-1.88	+0.88	+3.73	+2.06	+10.9	+6.82	+6.79	-3.73	+11.1	\	\	\	\	\	\	\	\
FT-Transformer	+16.4	+2.41	+1.16	+23.3	+4.39	+18.6	+17.3	+23.2	+2.33	+0.00	-0.016	-0.129	-0.019	-12.59	-0.031	-0.037	-0.005	+1.333
SwitchTab	+10.5	+7.32	-4.37	+4.41	+1.65	+11.3	+18.3	+21.4	-11.2	+12.6	-0.044	-0.581	-0.008	-6.994	-0.047	-3.946	-0.001	-47.14
TabNET	+7.03	+2.33	+0.49	+32.8	+0.78	-2.75	+3.64	+7.71	+0.84	+2.61	-0.003	-0.948	-0.035	-27.77	-0.072	-54.40	-0.170	-768.4
TabTransformer	-0.23	+16.4	+0.59	+15.3	+2.57	+15.6	+4.48	+41.1	+1.67	+2.14	-0.023	-0.005	-0.007	+0.255	-0.039	+0.125	-0.001	-2.340
TabR	+4.51	+8.61	-12.4	+25.2	+5.69	+3.18	+6.79	+8.06	+0.06	+3.09	-0.016	-0.175	-0.032	-14.93	-0.062	-4.205	-0.021	-39.69
TANGOS	+12.9	+0.00	+1.51	+34.0	+12.4	-5.53	+37.7	+48.0	+0.75	+7.62	-0.012	+0.015	-0.033	-19.27	-0.058	-4.557	-0.004	-83.34

Table 2: Model performance improvement by LLM-IMPUTE. The classification task is the more metrics' improvement (accuracy and F1 score) the better, while the regression task is the more metrics' decrease (nRMSE and MAE) the better.

4.2 ATLLM

Due to the heavy reliance on the prior knowledge of LLMs, LLM-IMPUTE exhibits limited effectiveness when applied to anonymized datasets such as Jannis from AutoML [Grinsztajn *et al.*, 2022], because it is incapable to analyze features which lack semantic meanings. To address this limitation, we further construct ATLLM as a complementary approach to LLM-IMPUTE. ATLLM has an augmented-training module by simulating feature-decrement scenarios in the training phase to improve its robustness. An illustrative overview of the ATLLM is presented in Figure 4. ATLLM is comprised of three main components:

Preprocessing. The preprocessing step augments training data by adding samples with manually missing features while preserving the original training data. These augmented samples are generated by manually removing certain features from the original training data. This component is specifically designed to simulate real-world feature-decrement scenarios, where any features may be missing in the testing phase.

Serialization. After preprocessing, training data are converted into sentences suitable for LLMs' input. ATLLM utilizes the List Template (" - Feature: value. ") format to structure the input text, leveraging LLMs' strong ability to read and parse lists effectively [Hegselmann *et al.*, 2023].

Fine-tuning and Predicting. The serialized sentences are then used to fine-tune Llama3-8B, a model released by Meta AI in April 2024. During fine-tuning, the number of epochs is set to 30 to ensure that the model has ample opportunity to learn and converge. The learning rate is set to $1e^{-5}$ to prevent overfitting and enable the model to converge effectively. Finally, the fine-tuned ATLLM is put into the downstream tabular task for prediction.

5 Experiments

5.1 Experimental Settings

In this section, we introduce datasets, models, and evaluation metrics used in experiments.

Datasets. To effectively simulate feature-decrement scenarios in tabular data, we select a variety of open-source and reliable datasets from OpenML and Kaggle's extensive dataset library. These datasets encompass three primary tasks: binary classification, multi-class classification, and regression, and span a range of fields such as finance and healthcare. A summary of the key attributes of the datasets is provided in Appendix A.

Models. To demonstrate the effectiveness of our proposed methods, we compare them against three categories of models: tree-based models, deep-learning models, and LLM. In Appendix B, we provide full hyperparameter grids for tree-based and deep-learning models and the prompt for LLM. Detailed experiment results are shown in Appendix D.

- **Tree-Based Models.** We evaluate XGBoost [Chen and Guestrin, 2016] and CatBoost [Prokhorenkova *et al.*, 2018] as representatives of gradient-boosted decision trees. Both models aim to correct the errors of previous iterations by adding additional decision trees, thereby reducing prediction errors.
- **Deep-Learning Models.** Deep-learning models we evaluate include MLP, FT-Transformer [Gorishniy *et al.*, 2021], Switchtab [Wu *et al.*, 2024], TabPFN [Hollmann *et al.*, 2023], TabPFNv2 [Hollmann *et al.*, 2025], TabNet [Arik and Pfister, 2021], TabR [Gorishniy *et al.*, 2024], TabTransformer [Huang *et al.*, 2020], and Tangos [Jeffares *et al.*, 2023].
- **LLM.** We select Llama3-8B as the representative of LLMs for evaluation. For this model, we construct the input text using the **List Template** format.

Evaluation Metrics. We utilize accuracy and F1 score for classification tasks, where higher values are preferred. For regression tasks, we employ normalized Root Mean Square Error (nRMSE) and Mean Absolute Error (MAE), where lower values are preferred.

5.2 LLM-IMPUTE Results

Overall Performance. As illustrated in Table 2, LLM-IMPUTE significantly enhances the performance of various models in feature-decrement scenarios, underscoring its effectiveness as a robust solution for imputing missing features. The experimental results demonstrate that LLM-IMPUTE achieves an average improvement of approximately 5% in overall model performance across 9 datasets.

Rationality. Figure 5 shows the overlap degree between feature ranges from LLM-IMPUTE and actual datasets, and the percentage of generated values of LLM-IMPUTE that fall within actual feature ranges. Despite the fact that feature

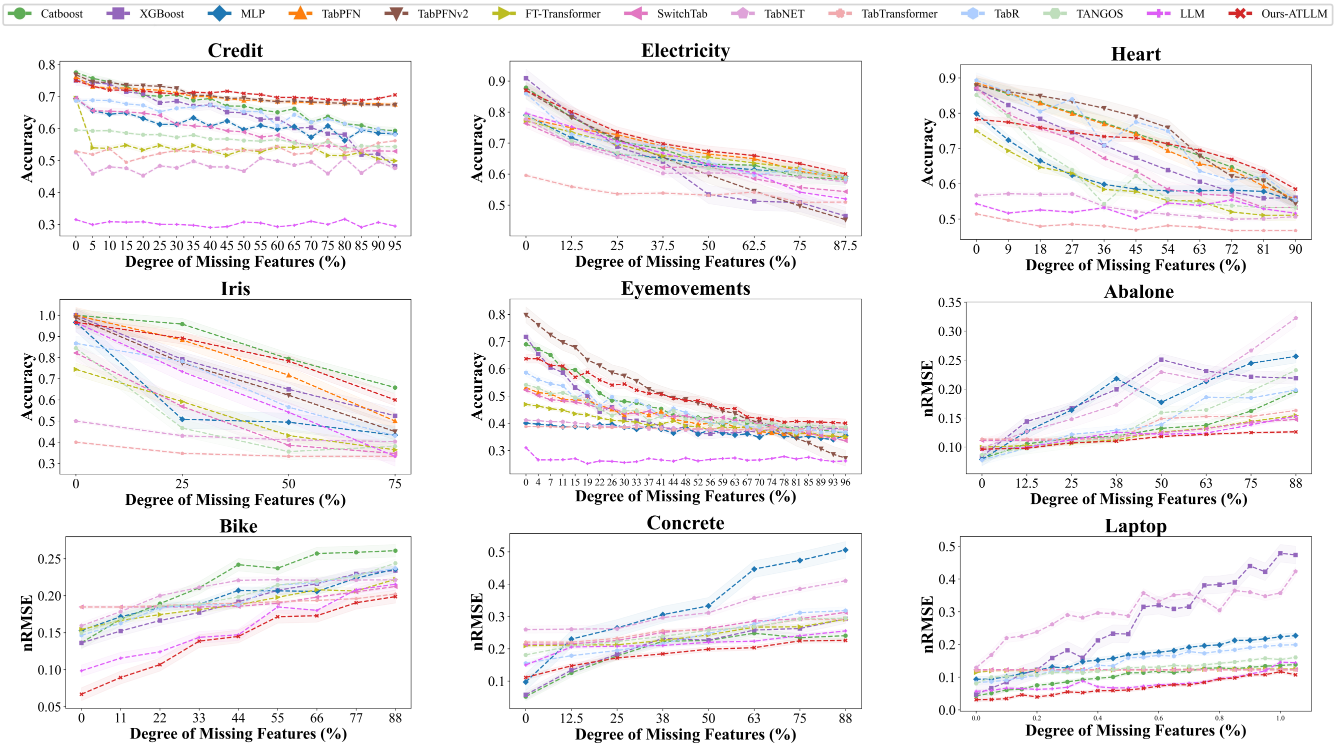

Figure 6: The performance of ATLLM and comparison models under different degrees of feature decrement.

ranges predicted by LLM-IMPUTE in Step 1 do not perfectly match actual ranges (i.e., the overlap degree is not high), most of the imputed values generated by LLM-IMPUTE still fall within realistic bounds and exhibit practical relevance.

Superiority. LLM-IMPUTE addresses the issue of the suboptimality of existing missing-feature imputation methods by generating values that are both valid and common-sense values. Furthermore, as LLM-IMPUTE only requires a single API call per tabular dataset, it helps reduce time costs while simultaneously improving model performance.

5.3 ATLLM Results

Overall Performance. Figure 6 illustrates the performance comparison between ATLLM and missing-feature adaptation approaches in feature-decrement scenarios across various datasets. When the degree of feature decrements is substantial, ATLLM consistently outperforms all the comparison models. Experimental results demonstrate that ATLLM surpasses existing approaches, with a 3% improvement in classification accuracy and a 5% reduction in regression nRMSE.

Robustness. We observe that TabPFNv2 exhibits excellent performance, especially when the degree of feature-decrement is low. However, its robustness is limited, as its performance deteriorates significantly with increasing degree of feature decrements. In contrast, ATLLM maintains relatively stable performance across varying degrees of feature decrements. This stability demonstrates that ATLLM has superior robustness compared to other models.

Superiority. The comparison between LLM and ATLLM reveals that the performance of ATLLM with no feature decrement is also improved. This suggests that the augmented-training module not only enables ATLLM to better handle feature-decrement scenarios but also deepens its understanding of the task during the training phase. Furthermore, ATLLM demonstrates superior performance on datasets with a higher number of features.

6 Conclusion

In this paper, we make the first attempt to address the problem of FTTA for feature decrements in tabular data, a unique and critical challenge in tabular data learning. Existing FTTA algorithms are primarily designed to handle distribution shifts and fail to effectively address feature decrements. Meanwhile, existing methods for feature decrement, such as missing feature imputation and adaptation, suffer from suboptimal performance and limited applicability. To tackle these challenges, we propose two novel FTTA approaches: LLM-IMPUTE, which leverages LLMs for training-free missing feature imputation, and ATLLM, which achieves better robustness by simulating feature-decrement scenarios during training, to further address tasks that can not be imputed, as an complementarity of LLM-IMPUTE method. Comprehensive experimental results demonstrate that our proposal significantly improves both performance and robustness in FTTA scenarios for feature decrement in tabular data.

Acknowledgements

This research was supported by the National Science Foundation of China (62306133, 624B2068), and the Key Program of Jiangsu Science Foundation (BK20243012). We would like to thank the reviewers for their constructive suggestions.

References

[Altman and Krzywinski, 2017] Naomi Altman and Martin Krzywinski. Tabular data. *Nature Methods*, 14(4):329–331, 2017.

[Arik and Pfister, 2021] Sercan Ö Arik and Tomas Pfister. Tabnet: Attentive interpretable tabular learning. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, pages 6679–6687, 2021.

[Badirli *et al.*, 2020] Sarkhan Badirli, Xuanqing Liu, Zhengming Xing, Avradeep Bhowmik, Khoa Doan, and Sathiya Keerthi Keerthi. Gradient boosting neural networks: Grownet. *arXiv preprint arXiv:2002.07971*, 2020.

[Borisov *et al.*, 2022] Vadim Borisov, Tobias Leemann, Kathrin Seßler, Johannes Haug, Martin Pawelczyk, and Gjergji Kasneci. Deep neural networks and tabular data: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 35(6):7499–7519, 2022.

[Chang *et al.*, 2024] Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45, 2024.

[Chen and Guestrin, 2016] Tianqi Chen and Carlos Guestrin. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794, 2016.

[Chizat *et al.*, 2020] Lenaic Chizat, Pierre Roussillon, Flavien Léger, François-Xavier Vialard, and Gabriel Peyré. Faster Wasserstein distance estimation with the Sinkhorn divergence. *Advances in Neural Information Processing Systems*, pages 2257–2269, 2020.

[Choi *et al.*, 2022] Kristy Choi, Chris Cundy, Sanjari Srivastava, and Stefano Ermon. LMPriors: Pre-Trained Language Models as Task-Specific Priors. *Advances in Neural Information Processing Systems 2022 Foundation Models for Decision Making Workshop*, 2022.

[Gorishniy *et al.*, 2021] Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting deep learning models for tabular data. *Advances in Neural Information Processing Systems*, pages 18932–18943, 2021.

[Gorishniy *et al.*, 2024] Yury Gorishniy, Ivan Rubachev, Nikolay Kartashev, Daniil Shlenskii, Akim Kotelnikov, and Artem Babenko. TabR: Tabular Deep Learning Meets Nearest Neighbors. In *Proceedings of the 12th International Conference on Learning Representations*, 2024.

[Grinsztajn *et al.*, 2022] Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux. Why do tree-based models still outperform deep learning on typical tabular data? *Advances in Neural Information Processing Systems*, pages 507–520, 2022.

[Guo *et al.*, 2025] Lan-Zhe Guo, Lin-Han Jia, Jie-Jing Shao, and Yu-Feng Li. Robust semi-supervised learning in open environments. *Frontiers of Computer Science*, 19(8):198345, 2025.

[Hegselmann *et al.*, 2023] Stefan Hegselmann, Alejandro Buendia, Hunter Lang, Monica Agrawal, Xiaoyi Jiang, and David Sontag. TabLLM: Few-shot classification of tabular data with large language models. In *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics*, pages 5549–5581, 2023.

[Hollmann *et al.*, 2023] Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. TabPFN: A Transformer That Solves Small Tabular Classification Problems in a Second. In *Proceedings of the 11th International Conference on Learning Representations*, 2023.

[Hollmann *et al.*, 2024] Noah Hollmann, Samuel Müller, and Frank Hutter. Large language models for automated data science: Introducing caafe for context-aware automated feature engineering. *Advances in Neural Information Processing Systems*, 2024.

[Hollmann *et al.*, 2025] Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637:319–326, 2025.

[Huang *et al.*, 2020] Xin Huang, Ashish Khetan, Milan Cvitkovic, and Zohar Karnin. Tabtransformer: Tabular data modeling using contextual embeddings. *arXiv preprint arXiv:2012.06678*, 2020.

[Jeffares *et al.*, 2023] Alan Jeffares, Tennison Liu, Jonathan Crabbé, Fergus Imrie, and Mihaela van der Schaar. TANGOS: Regularizing Tabular Neural Networks through Gradient Orthogonalization and Specialization. In *Proceedings of the 11th International Conference on Learning Representations*, 2023.

[Jia *et al.*, 2024] Lin-Han Jia, Lan-Zhe Guo, Zhi Zhou, and Yu-Feng Li. Realistic evaluation of semi-supervised learning algorithms in open environments. In *Proceedings of the 12th International Conference on Learning Representations*, 2024.

[Kadra *et al.*, 2021] Arlind Kadra, Marius Lindauer, Frank Hutter, and Josif Grabocka. Well-tuned simple nets excel on tabular datasets. *Advances in Neural Information Processing Systems*, pages 23928–23941, 2021.

[Lamb *et al.*, 2024] Tom A Lamb, Adam Davies, Alasdair Paren, Philip HS Torr, and Francesco Pinto. Focus on this, not that! steering llms with adaptive feature specification. *arXiv preprint arXiv:2410.22944*, 2024.

[Li *et al.*, 2024] Dawei Li, Zhen Tan, and Huan Liu. Exploring large language models for feature selection: A datacentric perspective. *arXiv preprint arXiv:2408.12025*, 2024.

[Meijerink *et al.*, 2020] Lotta Meijerink, Giovanni Cinà, and Michele Tonutti. Uncertainty estimation for classification and risk prediction on medical tabular data. *arXiv preprint arXiv:2004.05824*, 2020.

[Nam *et al.*, 2024] Jaehyun Nam, Kyuyoung Kim, Seunghyuk Oh, Jihoon Tack, Jaehyung Kim, and Jinwoo Shin. Optimized Feature Generation for Tabular Data via LLMs with Decision Tree Reasoning. *Advances in Neural Information Processing Systems*, 2024.

[Niu *et al.*, 2022] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient Test-Time Model Adaptation without Forgetting. In *Proceedings of the 39th International Conference on Machine Learning*, pages 16888–16905, 2022.

[Prokhorenkova *et al.*, 2018] Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. CatBoost: unbiased boosting with categorical features. *Advances in Neural Information Processing Systems*, pages 6639–6649, 2018.

[Sahakyan *et al.*, 2021] Maria Sahakyan, Zeyar Aung, and Talal Rahwan. Explainable artificial intelligence for tabular data: A survey. *IEEE access*, 9:135392–135422, 2021.

[Shao *et al.*, 2024] Jie-Jing Shao, Xiao-Wen Yang, and Lan-Zhe Guo. Open-set learning under covariate shift. *Machine Learning*, 113(4):1643–1659, 2024.

[Shwartz-Ziv and Armon, 2022] Ravid Shwartz-Ziv and Amitai Armon. Tabular data: Deep learning is not all you need. *Information Fusion*, 81:84–90, 2022.

[Wang *et al.*, 2021] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully Test-Time Adaptation by Entropy Minimization. In *Proceedings of the 9th International Conference on Learning Representations*, 2021.

[Wei *et al.*, 2022] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, pages 24824–24837, 2022.

[West, 2000] David West. Neural network credit scoring models. *Computers & operations research*, 27(11-12):1131–1152, 2000.

[Wu *et al.*, 2024] Jing Wu, Suiyao Chen, Qi Zhao, Renat Sergazinov, Chen Li, Shengjie Liu, Chongchao Zhao, Tianpei Xie, Hanqing Guo, and Cheng Ji. Switchtab: Switched autoencoders are effective tabular learners. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence*, pages 15924–15933, 2024.

[Yıldız and Kalayci, 2024] A Yarkın Yıldız and Asli Kalayci. Gradient boosting decision trees on medical diagnosis over tabular data. *arXiv preprint arXiv:2410.03705*, 2024.

[Zhao *et al.*, 2023] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.

[Zhou *et al.*, 2025] Zhi Zhou, Yu-Kun Yang, Lan-Zhe Guo, and Yu-Feng Li. Fully Test-time Adaptation for Tabular Data. In *Proceedings of the 39th AAAI conference on Artificial Intelligence*, 2025.

[Zhou, 2022] Zhi-Hua Zhou. Open-environment machine learning. *National Science Review*, 9(8):nwac123, 2022.

[Zhu *et al.*, 2021] Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. Tat-qa: A question answering benchmark on a hybrid of tabular and textual content in finance. *arXiv preprint arXiv:2105.07624*, 2021.