A Demonstration of TENDS: Time Series Management System based on Model Selection

Yuanyuan Yao, Shenjia Dai
Zhejiang University
yoyoyao,daishenjia@zju.edu.cn

Yilin Li
Zhejiang University
yilin.23@intl.zju.edu.cn

Lu Chen
Zhejiang University
luchen@zju.edu.cn

Dimeng Li
Alibaba Group
lidimeng.ldm@alibaba-inc.com

Yunjun Gao
Zhejiang University
gaoyj@zju.edu.cn

Tianyi Li
Aalborg University
tianyi@cs.aau.dk

ABSTRACT

The growth in sensor technologies, IoT devices, and information systems has opened up new opportunities for managing time series data across various domains. Despite significant progress, existing time series management systems face two crucial limitations: insufficient functionality and inadequate adaptability. This highlights the need for more comprehensive systems that not only improve data quality and analysis but also effectively manage the variety and volume of time series data.

This paper presents TENDS, a time series management system based on model selection. TENDS uniquely combines advanced functionalities for imputation, prediction, and anomaly detection, offering a comprehensive analysis of time series data. It features (i) an effective model selection mechanism to adapt to various data types and to improve efficiency; (ii) fourteen state-of-the-art prediction methods and three state-of-the-art imputation methods; and (iii) a dynamic expert knowledge base for anomaly detection, evolving continuously with new data to ensure accuracy. TENDS boasts a comprehensive suite of visualization tools. With its configurable offline and online interfaces, TENDS (i) provides extensive flexibility in model selection and parameter adjustment, (ii) facilitates easy visualization of training results, and (iii) supports real-time documentation and statistical analysis of time series.

PVLDB Reference Format:
Yuanyuan Yao, Shenjia Dai, Yilin Li, Lu Chen, Dimeng Li, Yunjun Gao, and Tianyi Li. A Demonstration of TENDS: Time Series Management System based on Model Selection. PVLDB, 17(12): 4357 - 4360, 2024.
doi:10.14778/3685800.3685874

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/yoyo185644/TENDS.

1 INTRODUCTION

Time series data [7], which is organized chronologically, is prevalent across various domains such as finance, healthcare, and transportation. Advances in sensor technology, IoT devices, and information systems are generating and accumulating vast amounts of time series data, creating both challenges and opportunities for analysis. For instance, AliCloud's TSDB [1]—a time series database—stores business data for numerous users in sectors, e.g., retail, e-commerce, finance, and supply chain management. Moreover, Alibaba Cloud's DAS [7] processes over 2 million prediction tasks in each two-hour window from 4 am to 2 pm daily, equating to a significant workload. These highlight the need for the development of time series management systems. Despite the advancements in this field [2–4, 6], two key limitations remain to be addressed.

Limitation I: Insufficient functionality. Existing time series management systems are limited in their scope, focusing on only a few tasks. AutoAI-TS [6] concentrates exclusively on time series prediction, while AWS Cloud [2] includes both prediction and anomaly detection. A truly comprehensive time series management system, however, should offer a wider range of functionalities, from enhancing data quality to conducting data analysis.

Limitation II: Inadequate adaptability. As shown in Fig. 1, time series data can exhibit various trends, including linear, stationary, turning point, spike, and regular patterns. Most existing systems are not equipped to effectively manage this diversity. AutoML-based systems [3, 6] adopt the concept of model selection to tackle different types of time series, where the systems evaluate the performance each model and select the the best-performing model. However, AutoML-based systems have two primary limitations.

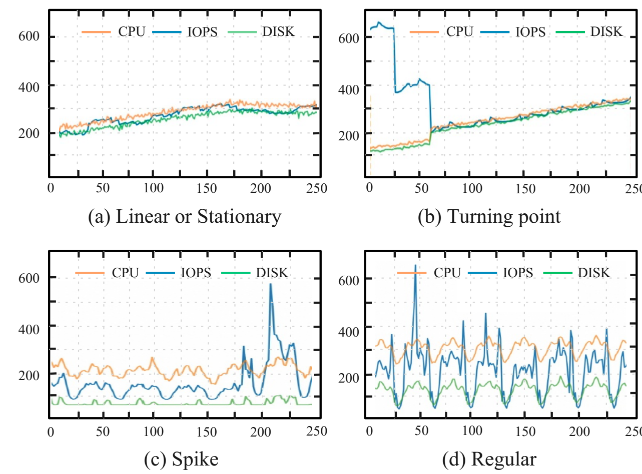

Figure 1: An case study of time series data.

Proceedings of the VLDB Endowment, Vol. 17, No. 12 ISSN 2150-8097.
doi:10.14778/3685800.3685874

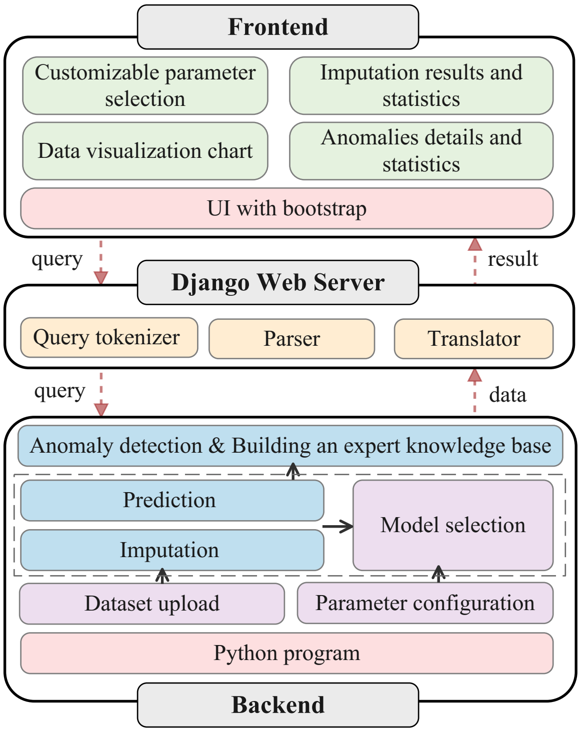

Figure 2: System overview. Frontend offers customizable options and displays through Bootstrap UI. Django Web Server is a web application framework for query processing. Backend handles business logic and data storage, and interacting with Frontend.

First, AutoML-based systems are specifically tailored for time series prediction, lacking broad functionality in handling other crucial tasks. Second, they become particularly inefficient when processing large-scale time series data.

We develop a **t**ime s**e**ries ma**n**agement system based on mo**d**el **s**election (TENDS) for comprehensive time series analysis and clear presentation of analysis results. To address the first limitation, TENDS incorporates advanced time series imputation, prediction and anomaly detection. For imputation, it simulates missing data and trains imputation models with iterative refinement on the simulated dataset. For prediction, it analyzes periodicity and stability in time series data to identify the data type and selects the most appropriate models for future forecasting, accommodating both multivariate and univariate data. For anomaly detection, TENDS maintains a dynamic expert knowledge base that evolves with new data. The knowledge base includes rules for identifying anomalies as well as strategies for managing them. To address the second limitation, TENDS adopts model selection for handling various types of time series. Enhancing model selection efficiency, TENDS employs the idea of SimpleTS [7], a method from our prior research, which clusters prediction algorithms by similar performance to streamline classification and thus significantly boost efficiency.

TENDS offers a suite of visualization tools to help users gain insights into time series data analysis. It equips users with fourteen state-of-the-art time series prediction methods and three state-of-the-art imputation methods. TENDS features a configurable offline interface, which offers extensive flexibility in parameter selection for model selection tasks and facilitates easy visualization of model training results. The online interface of TENDS assists users in documenting reasons for anomalies and provides statistical analysis tools for outliers and missing data. The key contributions of TENDS are summarized as follows.

- To our best knowledge, TENDS is the first time series data management system that offers imputation, prediction, and anomaly detection, featuring state-of-the-art methods in each category.
- TENDS integrates model selection into imputation and prediction to adapt to various types of time series and improve efficiency. Moreover, TENDS constructs a dynamic knowledge base to enhance anomaly detection.
- TENDS provides a comprehensive visualization toolkit. It features configurable offline and online interfaces for flexible model selection, training result visualization, real-time anomaly detection, and conducting statistical analyses, thereby deepening users' insights into the data.

2 SYSTEM OVERVIEW

Fig. 2 provides an overview of the TENDS architecture, which includes both the frontend and backend. The frontend offers a web interface and data visualizations for configuring time series data model training, displaying model training results, performing time series data analysis, and analyzing statistical results of anomalies. The backend handles data imputation, prediction, and anomaly detection. In particular, we adopt the idea of model selection for both imputation and prediction.

Imputation. First, TENDS randomly omits a subset of data points, determined by the imputation size setting. This is to simulate missing data for the training of imputation models selected by users. Second, TENDS defines evaluation metrics for assessing the imputation of missing values, including Mean Squared Error (MSE), Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), accuracy, R-squared and Sinkhorn distance [5] to assess the accuracy of filling in missing values and to compare the distribution similarity between data sequences. Next, TENDS trains the imputation models on the dataset with simulated missing values. The process continues with the iterative training of selected models, progressively refining the imputation for missing values. Finally, TENDS evaluates the imputed data using the predefined metrics and saves it for model selection.

Prediction. This process begins with prediction model training. First, TENDS sets up a weighted TS2Vec model for subsequent data classification in model selection. The TS2Vec model uses representation learning on the original data to reveal hidden information. TENDS then randomly samples a predefined portion of the training data to feed into user-selected models. Following this, TENDS trains the selected models according to the prediction window ratio of the dataset. Next, TENDS performs prediction tasks on each data piece in the training set using all models. Once a data piece has been processed for prediction tasks with all models, TENDS records and stores the most effective model, alongside its accuracy, precision, SMAPE metric, and prediction results, and each model's execution time. Furthermore, to compare artificial feature extraction with representation learning, TENDS saves features in the database, including turning points, variance, periodicity, stationarity, maximum value, minimum value, mean value, absolute variance, etc, for subsequent model selection.

Model selection. The goal is to identify the best models for both imputation and prediction among user-selected models, ensuring optimal performance for each time series type in online operations. First, TENDS clusters imputation and prediction models based on the data features and model performance derived from imputation and prediction, respectively. Second, TENDS trains a classifier using the clustering results. TENDS replace the best-performing model's label with clustered labels, ensuring balanced samples through upsampling. TENDS employs soft labeling during this step, which replaces traditional one-hot encoded labels with weighted vectors. This reduces overconfidence in correct labels and ensuring tight clustering of training examples, improving classifier accuracy.

Anomaly detection. TENDS enhances its anomaly detection by integrating expert judgment with data analysis, making the process not just data-driven but also wisdom-informed. The expert knowledge base is not static; it grows dynamically, enriched by ongoing contributions of expert feedback, criteria for judging anomalies, and insights into unusual patterns. The expert knowledge base encompasses not just rules for identifying specific anomalies but also suggestions for handling anomalies, response strategies, and practical operational experience.

Moreover, TENDS can categorize anomalies by their severity and type, tailoring specific response procedures for each. It employs predefined solutions for common anomalies and leverages its expert knowledge base for novel or complex situations. This ensures efficient and effective handling of various scenarios.

3 DEMONSTRATION

We proceed to introduce the implementation details of TENDS, including offline training and online management.

3.1 Offline training.

As shown in Fig. 3, the offline training section primarily involves pre-training and model selection for model imputation and prediction. We have divided the training interface into three modules: Training Setting, Training Processing, and Training Results.

Training Setting. Users have the options to: (i) upload datasets and choose machine learning models for model completion and prediction; (ii) define the training dataset size; (iii) set the prediction window size for training prediction models (e.g., setting a 10% prediction window means using the first 90% of data for training and the last 10% to test predictions); and (iv) set the imputed data percentage (e.g., setting it to 10% indicates randomly dropping 10% of the data in the training set for imputation model training).

Training Processing and Training Results. Once parameters are set and "Start Training" is clicked, TENDS activates the Training Processing module for imputation model training, prediction model training, and model selection. The process starts with training the models, then proceeds to classify and select them based on their accuracy and other factors. The module displays the progress of

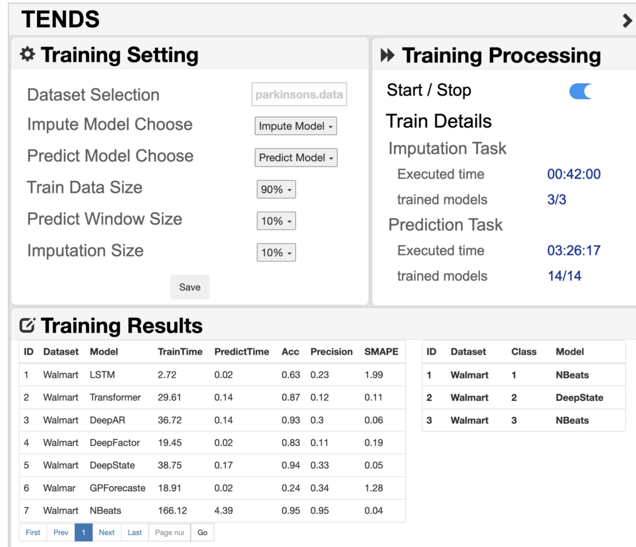

Figure 3: Offline training. Training Setting allows users choose models and parameters for imputation and prediction. Training Processing oversees and displays training progress (e.g., running for 50 minutes, completing 3 models). Training Results presents outcomes and model selection results.

model training, including execution time and the number of models trained, and presents the training results on the interface.

3.2 Online management.

Fig. 4 illustrates the online management section with five key modules: Task Setting, Task Control, Ongoing Task, Task Results, and Statistical Analysis.

Task Setting and Task Control. For imputation, TENDS offers a range of streaming methods. For prediction, TENDS allows the selection of either a specific model or the best model from previous model selection results. During online operations, TENDS retrieves streaming data, fills in missing values, and predicts the upcoming data for the "predict_window_size" using the selected model.TENDS updates the task details in the "Task Control" module every second, showing task status and executed time. For anomaly

Ongoing Task. The ongoing task displays a line chart, plotting streaming time series data against timestamps. Users can customize the timeframe to display—options include the last 24 hours, three days, or one week—with automatic updates every second. Multivariate data are differentiated by colored lines, while imputation and prediction are marked by yellow circles and dashed lines, respectively. TENDS compares actual and predicted values, highlighting anomalies with red circles.

Task Results. This module displays the results and statistics of imputation in the left table and those of anomaly detection in the right table. The left table lists timestamps ("Time"), variables with missing data ("Variable"), and the imputed values ("Impute"). The right table specifies anomaly times ("Time"), affected variables ("Variable"), original anomaly values ("Real"), and predicted values ("Predict").

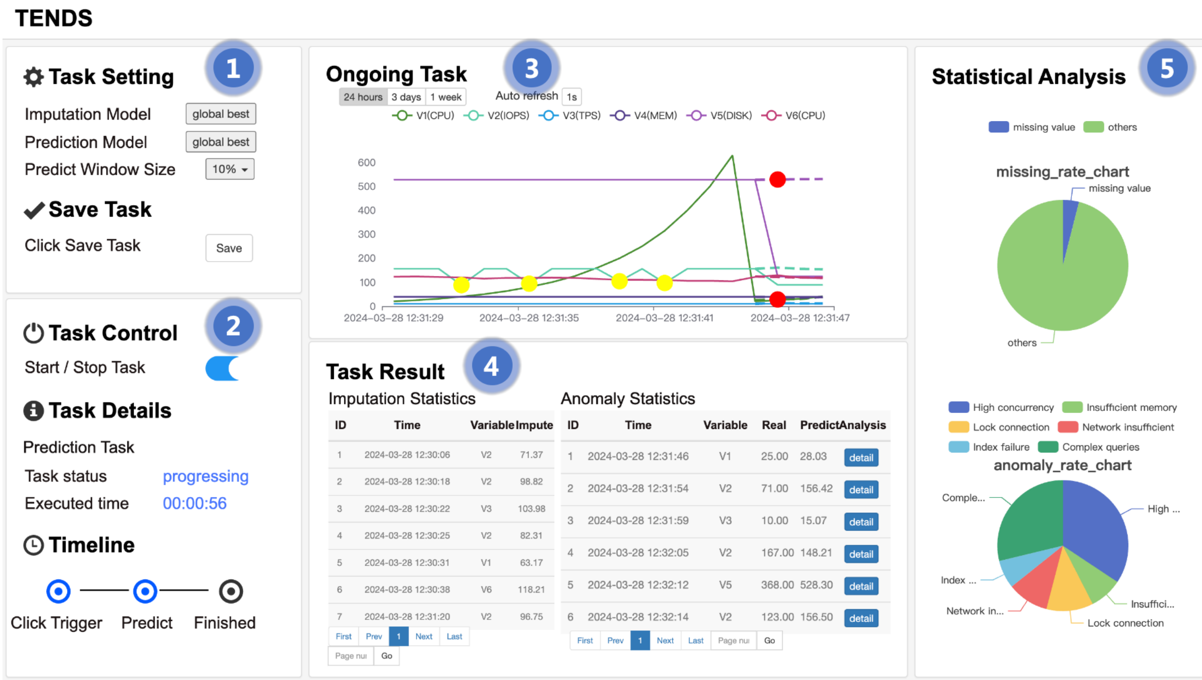

Figure 4: Online management. Task Setting selects imputation and prediction models. Task Control manages and tracks task progress. Ongoing Task displays a time-based graph for time series data, with yellow for imputed data and red for anomalies. Task Result shows specifics, (e.g., missing data at 01:15:20 in Variable 2 with an imputed value of 0.32). Statistical Analysis reveals the proportions of missing data and reasons of anomalies.

Experts can use the detail" button to annotate noteworthy observations, such as anomalies and their causes. This builds an expert database, improving the model's accuracy and reliability.

Statistical Analysis. TENDS uses pie charts to present the statistical and analytical findings for missing value and anomaly detection. First, TENDS can analyze the distribution of missing values over time in the dataset, aiding users in identifying patterns and trends of data absence. Second, TENDS can identify and display reasons for anomalies and their probabilities in a pie chart, including high concurrency, insufficient memory, lock contention, network latency, and complex queries in the scenarios of database anomaly detection.

3.3 Demonstration steps

The user operations of TENDS are divided into three steps.

Step 1. Users upload the dataset to TENDS and choose suitable prediction models based on the characteristics and requirements of the dataset. It then initiates the offline training process to train the selected models with the uploaded dataset, preparing them for online predictions.

Step 2. After completing step 1, TENDS switches to online prediction. Users can select the optimal model identified in the offline training phase or different models for online completion and prediction tasks. Moreover, users can specify a time period, which allows for the comparison of actual values with predicted ones and offers a clear insight into model performance.

Step 3. As part of the online prediction process, users can manually verify the system-generated anomaly judgments to ensure the accuracy and reliability of the model. Users can annotate abnormal data points with reasons and types based on their experience.

In summary, TENDS offers comprehensive functionalities for time series analysis, including imputation, prediction, and anomaly detection. It offers (i) flexible model and parameter selection, (ii) simple training result visualization, and (iii) real-time time series documentation and analysis.

ACKNOWLEDGEMENT

This work was supported in part by the NSFC under Grants No. (62102351, 62025206, U23A20296). Lu Chen is the corresponding author of the work.

REFERENCES

[1] 2024. Alicloud DAS. https://www.aliyun.com/product/hdm. 2024.03.01.
[2] 2024. AWS Auto-ML. https://docs.aws.amazon.com/managedservices/latest/userguide/forecast.html. 2024.03.01.
[3] Mustafa Abdallah, Ryan A. Rossi, Kanak Mahadik, Sungchul Kim, Handong Zhao, and Saurabh Bagchi. 2022. AutoForecast: Automatic Time-Series Forecasting Model Selection. In *CIKM*. 5–14.
[4] Lerina Aversano, Mario Luca Bernardi, Marta Cimitile, Martina Iammarino, and Debora Montano. 2023. Forecasting technical debt evolution in software systems: an empirical study. *Frontiers Comput. Sci.* 17, 3 (2023), 173210.
[5] Boris Muzellec, Julie Josse, Claire Boyer, and Marco Cuturi. 2020. Missing Data Imputation using Optimal Transport. In *ICML*, Vol. 119. 7130–7140.
[6] Syed Yousaf Shah, Dhaval Patel, Long Vu, Xuan-Hong Dang, Bei Chen, Peter Kirchner, Horst Samulowitz, and etc. 2021. AutoAI-TS: AutoAI for Time Series Forecasting. In *SIGMOD '21*. 2584–2596.
[7] Yuanyuan Yao, Dimeng Li, Hailiang Jie, Lu Chen, Tianyi Li, Jie Chen, Jiaqi Wang, Feifei Li, and Yunjun Gao. 2023. SimpleTS: An Efficient and Universal Model Selection Framework for Time Series Forecasting. *PVLDB* 16, 12 (2023), 3741–3753.