

Milestone Deliverable Review Report

Deep Funding Round: 4

Project code: DFR4-NEW9

Project title: Deep SYBIL: The Deep Learning Forecaster

Milestone number: 6

Milestone deliverable: Milestone deliverable: RNN

See the m6_deliverable directory in deep-sybil-public as an example to run the D-SYBIL service:

https://github.com/Tempor-ai/deep-sybil-public/tree/main/m6_deliverable

See the api_call_example.ipynb and other.ipynb notebooks as an examples of running D-SYBIL service on various datasets.

https://github.com/Tempor-ai/deep-sybil-public/blob/main/m6_deliverable/api_call_example.ipynb

See model_request.yaml for example of a custom model parameter

See this public link for the D-SYBIL URL details and demo videos of how the code being run as well as a code walkthrough (DO NOT SHARE OUTSIDE OF REVIEWER CIRCLE AND SNET):

https://drive.google.com/drive/folders/1xE_GX3HUOTRd63HJa7S138XaKldA89p9?usp=sharing

Date: 16/09/2024

Status: Accepted

Feedback (Why accepted, why rejected?):

The m6_deliverable directory for the Deep SYBIL project contains all the required components to run and evaluate the D-SYBIL forecasting service. It includes essential data files, model parameters, and example notebooks to guide API interactions. The Returns_short_interest_data_train.csv file holds the core dataset for model training, while temp_anom_w_forcing.csv introduces external forcing factors to improve forecast accuracy. Four Jupyter notebooks demonstrate API calls, showcasing how to integrate exogenous

variables like ILINet and STIR into the forecasting model. The `model_request.json` and `model_request.yaml` files include model parameters, with the YAML file using `random_state: 42` to ensure output reproducibility. An initial `url.yaml` file is included as a placeholder for future API configurations. Overall, the directory provides clear, reproducible steps for running and evaluating the Deep SYBIL service.

The `api_call_example.ipynb` notebook serves as a comprehensive guide for interacting with the Deep SYBIL API. It walks users through the process of making model requests and retrieving forecasts, using predefined parameters, including `random_state: 42`. The notebook ensures data is formatted correctly and API configurations are set up properly for seamless interaction with the Deep SYBIL service. It provides a step-by-step process for sending API requests, processing the results, and visualizing forecast outputs, making it easy to reproduce and understand the model's performance.

The notebook starts by verifying the Python environment, importing necessary libraries, and loading a time-series dataset of air passenger data from GitHub. It demonstrates how to preprocess the data, convert columns to the appropriate format, and use the data for model training. The example also covers creating model requests in both JSON and YAML formats, allowing users to customize preprocessors and base models. Finally, the notebook shows how to build the API payload, make the request, and capture the response in JSON format, ensuring successful API interaction.

To complete the deliverables SYBILs team presents a complete demo videos of how the code being run as well as a code walkthrough.

SYBILs deep-sybil-publicrepository shows activity last week and Jupyter Notebook 100.0%

If rejected, suggested changes: