DATSR System Requirements Specification

Release 1.1

Dux D-zine

Oct 28, 2022

TABLE OF CONTENTS

CHAPTER

ONE

THE CONCEPT OF OPERATIONS (CONOPS)

1.1 Current System

Time series analysis and forecasting is a powerful tool in our modern data-driven world. Because of this, time series data is in high demand in both research and in a variety of industries including meteorology, finance, power, and agriculture. As we advance our predictive schemes with innovations like machine learning and stochastic modeling, the availability of data to train and test these systems has not been able to keep up with the increased demand. Our product will address this problem head on. We plan to engineer a repository with time series in mind that will accommodate the many facets of this special class of data while maintaining an intuitive user interface and optimized data management.

1.2 Justification for a New System

The need for time series data is here to stay. While buzzwords like neural networks and deep learning are now common place in science journalism, time series analysis has silently been a fundamental component of the artificial intelligence revolution of the past few decades. We see it being applied to everything from predicting effectiveness of COVID-19 lock-downs [2] to studying the effect of insulin [3]. Our repository will fill a need in the field of time series analysis and therefore advance research and industry alike.

It should be noted that there are existing repositories for time series data including UEA & UCR Time Series Classification Repository [1] and Wolfram Data Repository (Time Series) [4]. However, these repositories fall short of our costumer's needs in several ways. One major drawback of many of these systems is a lack of consistent data. To train predictive models, one often needs several datasets of high-quality data with consistent formatting. Some repos have ample data, but do not provide crucial functionality for organizing time series data such as hierarchical and set structures. We plan to design our new system so that we can avoid the shortcomings of other repositories and find a solution that caters to all of our costumers' needs.

1.3 Operational Features of the Proposed System

The primary function of our system will be to provide a repository of time series data to train and test analysis models. Users will be able to select from a list of time series data sets with a variety of sizes, structures, and domains.

After the user has selected which time series they wish to use, our application will provide them with a set of data points for training. The user can then, if they choose, build a predictive model with the training set and make predictions for the validation set. They will then input their predicted values for the remaining points so that our program can rate the effectiveness of the predictions based on a statistically calculated "score."

1.4 User Classes

Researchers and Time Series Analysts

Both researchers and time series analysts need access to a large amount of clean, well-formatted time series data so that they can build effective predictive models. Furthermore, our application will provide functionality that will allow this class of users to evaluate their predictive models and compare with others who have done the same. This kind of feedback will create a healthy community of both cooperation and competition that will advance the field for everyone.

Students and Experimenters

By standardizing the process of getting and working with time series data, our users will be able to study and try out time series analysis without having to go through the hassle of wrangling and cleaning up data. They will also be able to learn about popular predictive schemes through the "high score" feature of our app. Although this class of users is less likely to use the evaluation component of our application, they will still find ample benefit in the availability of lots of training sets.

Contributers

This class of users will be able to add time series data sets into the repository's backend framework. This is important for the altruistic contributors who would like to make more data available to their peers, but also for those who are simply looking for a clean and easy way to store their TS data. Furthermore, if there are users who want to see what kind of predictive models may work well for their specific application, they can upload data and then look at the high score page to see experimental results generated by the user classes described above.

1.5 Modes of Operation

There will be a single mode of operation for all of our user classes. This mode is entered when users go to our website and it allows for all the functionality that is described above. Users will be able to download time series training data, evaluate predictive schemes, contribute to the repository, and view other predictive results on the high score page.

Beyond this single mode of operations, developers will have access to the source code of the application which will allow them to make any improvements or fixes to DATSR if necessary. These fixes can then be released by updating the code on the sever that hosts our application.

1.6 Operational Scenarios

Use Case #1: Retrieving Time Series Data

Brief Description: This use case describes how a user would retrieve a training set of a time series data set using our application.

Actors: A user

Preconditions:

1. User must have access to a browser and must go to the URL to reach our site

Steps to Complete the Task:

1. The user will select which time series data set they wish to work with
2. On the page where they have selected the data set, they will be able to view meta data that pertains to the set
3. They will then download the data into their browser's "Downloads" folder in the form of a .csv file or .dat file

Postconditions

After this task is completed, the user will have access to the training set of their selected time series data set. They can now navigate to a screen that allows them to upload predictions for the rest of the data points if they so choose.

Use Case #2: Uploading Predictions

Brief Description: This use case describes how a user would upload data produced using a predictive model to get feedback on the accuracy of their predictions.

Actors: A user

Preconditions:

1. User must have access to a browser and must go to the URL to reach our site
2. User must have made predictions based on TS data previously retrieved from the repository as detailed in the steps above.

Steps to Complete the Task:

1. The user will prepare their predictions as a .csv or .dat file in a format specified in the user documentation
2. They will then upload the file to our application on the "Upload Data" page of the website
3. The user will have the option to leave their information (e.g. name, GitHub link) for the purpose of the "high-score" charts
4. They will receive a calculated "rating" of their predictions
5. If their prediction is in a specified top scorers range for the given data set, they will be be placed on the score board for that data set; if they specified a name and link but did not get in the top range their name will show up on the scoreboard, but below the high scores

Postconditions

The user now has some idea of the predictive ability of the model they are testing and potentially have improved their model with the additional training data. Furthermore, the score board will have been modified in response to their submission.

Use Case #3: Adding Data to the Repository

Brief Description: This case describes how a contributer would add a time series data set to the repository and make it available to other users of the application.

Actors: A contributer, a reviewer

Preconditions:

1. Contributor must have access to a browser and must go to the URL to reach our site
2. Contributer must have a time series data set ready in the format specified in the user documentation

Steps to Complete the Task:

1. The user will upload their file as a .csv or .dat to the website in the "Upload Data" page
2. They will then fill out the form and click "submit"
3. The data will be sent to the backend database for review
4. Once a reviewer has looked at the data and approved it, it will be made available to all users of the repository

Postconditions

The new TS data set will either be available in the application's repository or will have been rejected by the reviewer for not meeting the specifications/standards of DATSR.

CHAPTER

TWO

SPECIFIC REQUIREMENTS

2.1 External Interfaces (Inputs and Outputs)

2.1.1 Time Series Data (Output)

The user will be able to download time series data sets from the repository as that is the main purpose of the repository. The data will be available in the form of a .csv or .dat file. The output will be sourced from our backend database which will hold all the TS data. These atomic data sets will have metadata that describes them in addition to the time series data itself. The ranges that this data will fall into can be expressed in terms of the number of data points (from 1 to 999999) and the number of variables (from 1 to 999). Alternatively, the size of the data can be seen as size in memory and there will be an upper limit of 5MB for this size.

2.1.2 Predictive Points (Input)

Users will be able to upload predictions they have made for the validation set of the time series data. The purpose of this is so that they can test their predictive model and receive feedback for future improvement. They will have to upload the data as a .csv or .dat file formatted in the manner specified in the user docs. The size of this input will vary depending on the number of validation points in that specific TS data set.

2.1.3 New Data Sets (Input)

Contributers to the repository will be able to submit new TS data that they wish to be included on the site. To do this, they must upload the data as a .dat or .csv file in the format specified in the user documentation. They must also specify meta data so that the new TS set can be categorized in the database. The size of this input will follow the same constraints as the output time series data which have been detailed above.

2.2 Functions

2.2.1 Preparing Data for Download

Once the user has selected which data set they wish to download, the logic section and database section of our application will have to correctly format their data as a csv or dat file. The backend modules will query the database and process the information to format it in the manner specified in the user documentation.

2.2.2 Input Validation

When a file is uploaded, the file will be compared to the expected file format and either accepted or rejected. If the uploaded file is ill-formatted, the user will get an error message and be asked to re-upload. If uploaded data is rejected, the user will be pointed in the direction of the documentation that specifies data formatting standards.

2.2.3 Prediction "Scoring"

If the user submits an acceptable file of predicted data points, the application will run our scoring algorithm to compare the validation set we have kept from the user with their predictions. Then, without showing the user the hidden validation set, we will tell them their "score" for the prediction they have made. Although the validation set will remain hidden, users will be able to see how the scoring algorithm works in the user documentation.

2.3 Usability Requirements

Our application will include an intuitive GUI that guides users through the process of downloading time series data and testing predictive models against hidden validation sets. In addition to this, our application will include thorough user documentation that will help guide users not only through the application's UI, but also through the methodology and reasoning used in designing the system. For developers/contributers to the project, there will be documentation that further details the system on a technical level to aid future maintenance and updates.

2.4 Performance Requirements

We are aiming to have a statistically significant number of users (95%) experience wait times of less than 5 seconds for loading the website. Once the user moves on to trying to download the data set, we hope to prepare and send the file to the user in less than 30 seconds 95% of the time. The scoring algorithm will complete and display the rating of a user's prediction within 20 seconds also at a rate of 95%.

2.5 Software System Attributes

Key to our application is the consistency, reliability, and transparency of the scoring algorithm used with predictive data. A large part of the target market for our application are people in the realm of academia and high-tech industries. In the academic world it is very important to understand the methodology of an application and for that methodology to remain consistent. Furthermore, industry regulations often call for transparency especially in areas that commonly use time series analysis such as electric utilities and economics.

This application will be able to run on all major browsers. This is an important attribute for our project because we have a diverse set of possible users and it is unreasonable to expect them to have the same network stack. To achieve compatibility with most major browsers (e.g., Chrome, Firefox, Safari, Edge), we will make sure to only use web development frameworks and libraries that have widespread support and documentation.

CHAPTER THREE

REFERENCES

1. Bagnall, T. O. (2022). UEA &amp; UCR Time Series Classification Repository. Retrieved October 6, 2022, from https://www.timeseriesclassification.com/index.php
2. Singh S, Chowdhury C, Panja AK, Neogy S. Time Series Analysis of COVID-19 Data to Study the Effect of Lockdown and Unlock in India. J. Inst. Eng. India Ser. B. 2021;102(6):1275–81. doi: 10.1007/s40031-021-00585-7. Epub 2021 Apr 8. PMCID: PMC8031344.
3. Wang, Z., Tang, X., Swaminathan, S.K. et al. Mapping the dynamics of insulin-responsive pathways in the blood–brain barrier endothelium using time-series transcriptomics data. npj Syst Biol Appl 8, 29 (2022). https://doi.org/10.1038/s41540-022-00235-8
4. Wolfram Research, Inc. (2022). Time Series. Wolfram Data Repository. Retrieved October 6, 2022, from https://datarepository.wolframcloud.com/type/Time-Series/

CHAPTER FOUR

ACKNOWLEDGMENTS

The format of this SRS document was originally created by Professor Juan Flores. Reference to a completed SRS document was provided by Ronny Fuentes, Kyra Novitzky, Jack Sanders, Stephanie Schofield, Callista West with their Fetch project SRS.