System Requirements Specification

Release 1.1

Dux D-zine

Oct 13, 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 needed in a wide 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 also 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 insulin effectiveness [*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 data. To train predictive models, one often needs several datasets of high-quality data. 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 is then intended to use this data set in any way they wish to build a predictive model for the remaining data points (which will be specified when they receive the initial csv file). The user will then input their predicted values for the remaining points at which point our program will rate the effectiveness of the predictions.

1.4 User Classes

Time Series Analysts
Our application will be useful to analysts in industry who do work with time series. The increased availability and specificity of time series data through the use of our repository will allow professionals to build more accurate and precise predictive models.

Researchers
Researchers in the field of artificial intelligence will also find our repository useful. It will give them access to the kind of specific and high quality data required in a research setting without having to scour several different repositories across the web.

Students
By standardizing the process of getting and working with time series data, students will be able to study time series analysis without having to go through the hassle of wrangling and cleaning up data. They will also be able to learn to build better models through the "high score" feature of our app, which allows easy comparison with others working in AI.

Contributers
This class of users will (for now) be restricted to the DUX D-zine team as we build and expand the repository. They will be able to add time series data sets into the repository's backend framework and control updates and new releases of the application. This will allow us to adapt the repository according to feedback and also add more data for the benefit of the user classes listed above.

1.5 Modes of Operation

There will be a single mode of operation for all of our user classes. Within this mode users will be able to select time series data, receive the training data in a csv file, and then test their predictions against a validation set kept hidden by the 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 using our application.

Actors: A user

Preconditions:

1. The user must have system requirements satisfied
2. The user must have the application installed

Steps to Complete the Task:

1. The user will select which time series data set they wish to work with
2. They will then download the data into their computer's "Downloads" folder in the form of a .csv file
3. Then, they will be given the option to upload predictive data for the remaining data points in the time series

Postconditions

The user will now be on a screen that allows them to upload predictions for the rest of the data points. If they would like to test their predictive model, they can continue in the application as prompted.

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 system requirements satisfied
2. User must have our application installed
3. User must have made predictions based on TS data previously retrieved from the repository according to the steps detailed above.

Steps to Complete the Task:

1. The user will prepare their predictions in a specified format as a csv file
2. They will then upload the file to our application
3. The user will have the option to leave their information (e.g. name, GitHub account) 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 notified by the GUI and their name will be added to the "high-score" page

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.

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 users of the application.

Actors: A contributer

Preconditions:

1. Contributer must have system requirements satisfied
2. Contributer must have our application installed
3. Contributer must have access to the application's data base

Steps to Complete the Task:

1. Contributer will format TS data in the manner necessary to interact with the backend framework
2. Then they will specify meta data properties of that data set
3. They will upload it to the data base, making it available to users
4. If the new addition of data requires additional features in the front end application, the contributer may make a new release of the application and push the updated application to end users

Postconditions

The new TS data set will be available in the application's repository. Potentially users will have to update their application depending on the nature of the data added.

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 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 1MB for this size.

2.1.2 Predictive Points (Input)

The user 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 file formatted in a specific way which will be described in the user documentation. The size of this input will vary depending on the number of validation points in that specific TS data set.

2.2 Functions

2.2.1 Preparing Data for Download

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

2.2.2 Input Validation

When a file is uploaded, the predicted data 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 15 seconds for loading the app and being able to see and interact with the GUI. 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 1 minute 95% of the time. The scoring algorithm will complete and display the rating of a user's prediction within 30 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. In the academic world it is very important to understand the methodology of an application and for that methodology to remain consistent.

This application will be able to run on all 3 major operating systems (Linux, MacOS, and Windows). 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 OS. To achieve compatibility on all three OSs, we will ensure that any dependencies included in our design will available on all major operating system types and clearly defined in the requirements section of the documentation. Furthermore, we will have documentation that guides user's through using the app on each of the three major OSs.

CHAPTER

THREE

REFERENCES

1. Bagnall, T. O. (2022). UEA & 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.