SRS

Release 1.0

Dux D-zine

Oct 10, 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 simply 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 costumer's time series' 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. When a user requests a time series, they will have several options for what metadata (if any) they would like to specify for the data set. This could include options such as domain of data, maximum/minimum length, and number of variables.

After the user has specified what kind of time series data they are looking for, our application will provide them with a training set of data points. 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 in the time series (which will be specified when they receive the initial data points). 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 analysts to build more accurate and precise predictive models.

Researchers
Researchers in the field of artificial intelligence will find our repository useful in their research because it will give them access to the kind of specific and high quality data required in an academic setting.

Students
By having the ability to specify metadata specifications for time series data sets, students will find our application useful because it will allow them to learn about time series analysis without having to wrangle large and messy data sets.

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 request time series data sets according to their specifications, receive the training data in the format of their choosing, 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 our application installed

Steps to Complete the Task:

1. The user will specify properties that the data must fit
    a. This will include domain, size boundaries, etc.
2. They will receive a list of potential data sets in the repository
3. The user will choose the file type that they wish to download the data as
4. The user will download the data into their computer's "Downloads" folder
5. Finally, 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 select the type of data they intend to upload to the application
2. They will select the file that matches the data type specified and confirm that they wish to upload it
3. They will receive a calculated "rating" of their predictions
4. [*Possibly*] if their prediction is in the top 5 based on the score they receive, they will have the option to leave their name and information about the predictive model they used

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 updated their application if an update is necessary.

CHAPTER

TWO

SPECIFIC REQUIREMENTS

2.1 External Interfaces (Inputs and Outputs)

2.1.1 Meta Data Specifications (Input)

The user will have several options to choose from for requirements they want their TS data set to satisfy. The purpose of this will be to narrow down which repository data sets to show the user or to tell them that the repo does not have data that fits their specifications. There will be specific ranges of acceptable values for numerical meta data, i.e. number of data points (1 to 999999) and number of variables (1 to 999). For other specifications such as domain, hierarchical class, scalar/vector, set class, etc. the options will be discrete and the user will choose from only acceptable values. This class of inputs will come from the user selecting their meta data options on the application's GUI.

2.1.2 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 a variety of formats including csv, JSON, and txt. 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 TS 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). The size of the data will also be within a certain range (up to 1MB).

2.1.3 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 can upload this data in any of the file formats listed above (i.e. csv, JSON, txt). 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 Data Set Filtering

When the user inputs the specifications that the TS data must fulfill, the program will filter out data sets that do not meet their requirements before displaying a list of TS data sets available for download. If no data sets exist that meet their specifications, then the GUI will display a message that indicates that no such data exists in the repository.

2.2.2 Preparing Data for Download

Once the user has selected which data set they wish to download, they will have to select the format they wish to download it as. At this point, our application will query the backend and generate a document in the specified format.

2.2.3 Input Validation

When a user inputs their predicted data points they will first have to specify which type of data they are uploading. Then, our system will know what kind of formatting is needed for that type of file upload. When a file is uploaded, the predicted data file will be compared to the expected file format. If the uploaded file is ill-formatted, the user will get an error message and be asked to re-upload.

There will also be input validation associated with meta data specifications. This will mostly consist of ensuring that numerical values are within allowed ranges which are listed above.

2.2.4 Prediction "Scoring"

If the user submits an acceptable file of predicted data points, the application will run our scoring algorithm and compare the validation set we have reserved 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 30 seconds for data filtration and TS repository display after they enter their specifications for a search. 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 95% of the time.

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 like this if it is to be used in research.

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.

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.