Time Series Forecasting & Benchmarks Repository
Software Requirements Specification

Aiden Duval, Zane Globus-O'Harra, Erin Stone, Kareem Taha, Zachary Weisenbloom
Team 1 ("zeakz")
https://github.com/zfgo/CS_422_p1
CS 422 Spring 2023 Project 1 – 2023-05-05

Table of Contents

1 Revision History

Date	Author	Description
2023-04-13	Zane G-O, Aiden D	Add initial document to Google Drive, add table of contents, and add section and subsection titles. Add 2, 2.1, start 2.2, 2.4, 2.6.
2023-04-15	Zane G-O	Add 2.3, finish 2.2, 2.4. Create and finish use case 2.6.0.1. Create and start use cases 2.6.1 (use cases for contributors: 2.6.1.1), and 2.6.2 (use cases for participants: 2.6.2.1, 2.6.2.2).
2023-04-16	Zane G-O	Finish use cases. Start 3.1, 3.2.
2023-04-19	Zane G-O	Finish 3.1 (including all subsubsections) and 3.2 (including all subsubsections). Add 2.5 (including all subsubsections), 3.3, 3.4, 3.5, 4, and 5.
2023-05-04	Zane G-O	Small format changes.

2 Concept of Operations

The concept of operations (ConOps) “describes system characteristics for a proposed system from the users’ viewpoint” (IEEE Std 1362-1998). The ConOps communicates overall system characteristics to all stakeholders, and should be readable by any stakeholder familiar with the application domain.

2.1 Current System or Situation

The fields of data science (DS), machine learning (ML), and artificial intelligence (AI) have exploded in recent years. The engineers in these fields analyze many types of data, creating models and algorithms to classify, categorize, predict, etc. One type of data is the time series (TS), which has time, t, as the independent variable, and the magnitude of some other variable, x, with respect to the time domain. The recording of this data happens at equally spaced instances of time, Δ. TS forecasting, using models created by DS or ML engineers, aims to determine an estimate for the value of $x_{N+\Delta}$.

With the rise in popularity of DS, ML, and AI, so too has TS forecasting risen in popularity, especially with an abundance of data from Internet of Things (IoT) devices. However, many engineers in this field find themselves going back and forth across the internet, spending a great deal of time searching for places to download TS data.

2.2 Justification for a New System

Despite the importance and rise of popularity of TS forecasting and analysis, there is no central repository where TS data and benchmarks are stored. The DS and ML engineers need such a location where they can obtain data to test their models, and then compare the results of their models against other methods.

Another issue is that the current repositories that hold TS data are spread across the web, and the data that they hold has many different characteristics. Some of theses differences include:

- Contain a limited number of TS,
- The TS has very little data,
- Scalar TS,
- Non-hierarchical TS,
- Not free,
- No possibility to benchmark results outside of a competition,
- Aims other than TS forecasting,
- Non-standard data formats,
- Redundant pointers to the same data, and
- Some do not actually contain TS data.

It would hence be desirable to create a central repository that would fix these issues and be a main hub for TS forecasting data.

2.3 Operation Features of the Proposed System

The new system would allow for contributors to upload TS data containing a training set and a testing set to the web. Then, DS and ML engineers, or "participants," would be able to browse through the uploaded datasets and download any training dataset that interested them. The participants would then perform experiments on the data they downloaded, either by creating a model, or by developing a new forecasting algorithm for the TS. These experiments would produce a "solution" dataset, which the participant could then upload to the system, which would then be compared to the test set. The participant would be presented with a list of metrics (and graphs?) comparing their solution dataset to the actual test dataset.

2.4 User Classes

Users on the website are distinguished into 3 categories:

- ***Contributor***: Uploads TS data partitioned by train/test. It is up to the contributor to ensure that the data is in the proper format, is properly cleaned/filtered, etc.
- ***Participant***: Also known as the DS and ML engineer, the participant downloads TS training data, trains a ML-model or uses a forecasting algorithm on it, then uploads the solution data that the model produces.
- ***Administrator***: Database maintenance.

2.5 Modes of Operation

There will be at least 4 modes of operation: 1 mode of operation that all users will share, 1 mode of operation for contributors, and 2 modes of operation for participants.

2.5.1 Global Modes of Operation

There will be one global mode of operation that all users will share.

1. All users need to create an account or sign in to use the TS repository application. This mode will allow users to sign in or create an account from the start up page. From there, the modes of operation will diverge.

2.5.2 Contributor Modes of Operation

The contributor will have one mode of operation.

1. This mode of operation will allow contributors to upload a forecasting task and TS data sets to the TS repository, and make use of the application's functionality that supports this mode of operation.

2.5.3 Participant Modes of Operation

The participant will have two modes of operation, depending on whether they are downloading TS data or uploading a solution.

1. The first participant mode of operation will allow participants to download TS training data that is associated with a forecasting task. The participant will be able to view all of the created forecasting tasks, as well as sort them by keyword. Once the participant chooses a forecasting task that interests them, they will be allowed to download the TS training data that a contributor uploaded for that task.
2. The second participant mode of operation will allow participants to upload TS solution data that they have forecasted based on a forecasting task and using TS training data that they have already downloaded via the first participant mode of operation. After uploading their TS solution data, the participant will be displayed comparison metrics, statistics, and graphs comparing their solution data to the TS test data that was uploaded by the contributor for that forecasting task.

2.6 Operational Scenarios (a.k.a. "Use Cases")

Use Case 2.6.0.1: User creates an account
Brief Description: This use case describes how a user would be able to start using the system by creating an account.
Actors: A user who has not used the system before.
Preconditions:

1. The user has a desire to upload TS data or has a desire to download data upon which to test their models.
2. The user has internet access.
3. The user has an email address.

Steps to Complete the Task:

1. The user follows the prompts to create an account on the initial screen, and enters their email along with a password.
2. The user selects whether they are a *contributor* or a *participant*.

Postconditions: The user has now successfully created an account, and can begin either uploading TS data if they chose to be a contributor, or can begin downloading training datasets if they chose to be a participant.

2.6.1 Contributor Use Cases

Use Case 2.6.1.1: Contributor uploads forecasting task and TS data

Description: This use case describes the sequence necessary for a contributor to upload time-series data.
Actors: Contributors.
Preconditions:

1. The contributor has an account.
2. The contributor has internet access. ~~can navigate to & access the web-application~~
3. The contributor is logged in as a contributor.
4. The contributor has a forecasting task.
5. The contributor has TS data that is associated with the forecasting task in (4), and has already partitioned the TS data into train/test sets.
6. The TS data is in a format compatible with the system.

Steps to Complete the Task:

1. The contributor navigates to the "upload data" page.
2. The contributor inputs the forecasting task.
3. The contributor uploads the TS data.
4. If multiple TS datasets are associated with one forecasting task,
    a. The contributor specifies the TS set metadata that is associated with all of the uploaded files.
5. The contributor specifies the TS metadata associated with each file.

Postconditions: The contributor has uploaded a forecasting task with associated TS data, that is now freely available for any participant to download.

2.6.2 Participant Use Cases

Use Case 2.6.2.1: Participant downloads TS training data

Description: This use case describes the sequence necessary for a participant to download TS training data.
Actors: Participants.
Preconditions:

1. The participant has an account.
2. The participant has internet access.
3. The participant is logged in as a participant.
4. The participant has available hard drive space to download the TS training data.

Steps to Complete the Task:

1. The participant navigates to the "download data" page.

2. The participant selects a forecasting task that they wish to complete.
3. The participant downloads the TS training data associated with the task in (2).

Postconditions: The participant has selected a forecasting task and downloaded the associated TS training data.

Use Case 2.6.2.2: Participant uploads solution data

Description: This use case describes the sequence necessary for a participant to upload TS solution data created from a forecasting task and the associated TS training data.
Actors: Participants.
Preconditions:

1. The participant has an account.
2. The participant has internet access.
3. The participant is logged in as a participant.
4. The participant has completed a forecasting task (see Use Case 2.6.2.1) and created TS solution data.

Steps to Complete the Task:

1. The participant navigates to the "upload solution data" page.
2. The participant selects the forecasting task for which they have created solution data.
3. The participant uploads their TS solution data.
4. The participant is displayed comparison metrics, statistics, and graphs comparing their solution data to the TS test data set that was uploaded by the contributor for that forecasting task. The participant can do what they choose with these metrics.

Postconditions: The participant can now view statistics comparing their TS solution data to the TS test data that was uploaded for that forecasting task by a contributor.

3 Specific Requirements

3.1 External Interfaces (Inputs and Outputs)

3.1.1 User creates an account

1. ***Description of purpose***: To create an account on the TS repository application so that users can be authenticated and identify themselves as contributors, participants, or administrators.
2. ***Source of input or destination of output***: The input will be the user's personal information, as well as their decision to register as a contributor or participant. The output will be a successfully created account on the TS repository application.
3. ***Valid ranges of inputs and outputs***: All authentication for the username must be unique (the account can not already exist). The password can be any valid string of characters (to be determined on implementation). The output will create a single account for the new user.
4. ***Units of measure***: Not applicable.
5. ***Data formats***: Unknown/unsure.

3.1.2 Contributor uploads forecasting task and TS data

1. ***Description of purpose***: To create a forecasting task with associated data for data scientists and machine learning engineers to forecast.
2. ***Source of input or destination of output***: Input will be a forecasting task (defined by a forecasting period and number of forecasts), as well as a set of files containing TS data (both training datasets and test datasets). The destination of output will be the backend database where the forecasting task and files will be stored. The output for the contributor will be a message telling them they have successfully uploaded their forecasting task and TS data.
3. ***Valid ranges of inputs and outputs***: A forecasting task must be input, defined by forecasting period and number of forecasts within a valid range (unsure what this range is). Files must be in the proper format, and properly formatted (the proper formatting is the contributor's responsibility).
4. ***Units of measure***: Not applicable.
5. ***Data formats***: For files that the contributor wishes to upload, file types will likely be CSV, XLSX, XLS, and JSON.

3.1.3 Participant downloads TS training data

1. ***Description of purpose***: The purpose is for the participant to download TS training data associated with a forecasting task so that they can forecast the training data using the forecasting task's parameters to create solution data.
2. ***Source of input or destination of output***: The source of input will be the user selecting the files that they wish to download. The destination of output will be the user, who has downloaded whatever TS data files that they wish to download.
3. ***Valid ranges of inputs and outputs***: The input range will be a mouse click from the user, clicking to download the files that they desire. The output range will be any of the files stored in the database, in a format desired by the user.
4. ***Units of measure***: Not applicable.
5. ***Data formats***: For files that the participant wishes to download, file types will likely be CSV, XLSX, XLS, and JSON.

3.1.4 Participant uploads solution data

1. ***Description of purpose***: To create comparison metrics, statistics, and graphs between the participant's solution and the already uploaded TS test data that the contributor uploaded for the forecasting task.
2. ***Source of input or destination of output***: The source of input will be the user uploading their solution data. The destination output will be the comparison metrics, statistics, and graphs that are computed and displayed to the user.
3. ***Valid ranges of inputs and outputs***: The input range will be a single solution file (or set of solution files?), of the correct file types, and properly formatted. The valid range of outputs will be numbers and graphs.
4. ***Units of measure***: Possible percentages for the metrics, and variable units of measure for the graphs, depending on the units of the TS.

5. ***Data formats***: For files that the participant wishes to upload, file types will likely be CSV, XLSX, XLS, and JSON. Graphs will be images (PNG or JPG, displayed on the web page).

3.2 Functions

3.2.1 Account Creation Functions

1. ***Validity check on login inputs***: This system shall verify that no user is using the same username or email as an already registered user, ensuring that each account is unique, and registered to only one email.
2. ***Sequence of operations in processing inputs***: After the user has filled out the required authentication information, they will choose to be either a contributor or participant. Depending on this decision, the user will be able to access only a certain set of web pages (these web pages are the "outputs" for the account creation use case).
3. ***Responses to abnormal situations***: The validity check will either lead to the creation of a new account, or asking the user to create a new account again, but with different information that does not conflict with previously stored user information.
4. ***Relationship of outputs to inputs***:
    a. ***Input/output sequences***: The user creates an account (registering their email, and creating a username and password). The user then selects if they are a contributor or participant. The user is then displayed with the output web pages that depend on their decision to be a contributor or participant.
    b. ***Formulas for input to output conversion***: Not applicable.

3.2.2 Forecasting Task Creation Functions

1. ***Validity check on forecasting task inputs***: The system shall verify that the forecasting task inputs are in the proper format and are in a valid range.
2. ***Sequence of operations in processing inputs***: The system first performs a validity check. If invalid, the forecasting task is rejected. Otherwise, the forecasting task is stored in the system's database.
3. ***Responses to abnormal situations***: If the forecasting task fails the validity check, then the user must enter a new and valid forecasting task.
4. ***Relationship of outputs to inputs***:
    a. ***Input/output sequences***: The user enters the forecasting task. A validity check is performed. The forecasting task is stored in the database, and the user can also upload TS data files that are associated with that forecasting task.
    b. ***Formulas for input to output conversion***: Not applicable.

3.2.3 File Upload Functions

1. ***Validity check on file inputs***: The system shall verify that the files that the user wishes to upload have the proper file type, as specified by the data formats in 3.1.2 and 3.1.4. The system will not

check the contents of the file: it is the user's job to ensure that the contents of the files that they are uploading are properly formatted.

2. ***Sequence of operations in processing inputs***: The system will upload the file. The system will then convert the file to a single, consistent format/file type which will be how the files are stored in the system's database.
3. ***Responses to abnormal situations***: If, for some reason, the user does not ensure that the contents of the files that they are uploading are properly formatted, and the system is unable to convert the files to a single, consistent format/file type in the database, the system will raise an error and the files will be rejected and not added to the database.
4. ***Relationship of outputs to inputs***:
    a. ***Input/output sequences***: The user uploads a file. A validity check is performed. The files are temporarily saved while they are being converted to a single, consistent format/file type. This converted file is saved in the database. The original file is deleted—the only copy of the file is the converted version which is saved in the database.
    b. ***Formulas for input to output conversion***: Formulas for converting files from CSV, XLSX, XLS, and JSON to the desired single, consistent format/file type will be needed.

3.2.4 File Download Functions

1. ***Validity check on download selection inputs***: The system shall ensure that only files that are saved in the database and associated with a forecasting task shall be displayed to the user. The user selects their preferred file type, specified by the data formats in 3.1.3, and inputs a mouse click to download the TS data files upon which they wish to create a forecast. These inputs will be "drop down menus" or something similar, and thus we will not need to validate the *user's* inputs, only ensure that the displayed files actually exist and are associated with a valid forecasting task.
2. ***Sequence of operations in processing inputs***: The user inputs a mouse click selecting the files they wish to download and specifying a file type that they wish to use. The system retrieves the file from the database, converting it to the format that the user desires, and providing them with the files.
3. ***Responses to abnormal situations***: Only files available for download will be displayed to the user, eliminating any situation where the user would attempt to download an unavailable file. Only the specified data formats will be available, eliminating any situation where the user would attempt to download a file in a format where conversion functionality has not been implemented.
4. ***Relationship of outputs to inputs***:
    a. ***Input/output sequences***: The user inputs a mouse click selecting the files they wish to download and specifying a file type that they wish to use. The system converts the files from the single, consistent format/file type that was used to store them in the database to the file type of the user's choice. The files are temporarily saved, and made available for the user to download. The files that have been temporarily saved are deleted after the user downloads them
    b. ***Formulas for input to output conversion***: Formulas for converting files from the single, consistent format/file type that is used in the database to CSV, XLSX, XLS, and JSON will be needed.

3.3 Usability Requirements

- ***Efficiency of use***:
    - The hierarchy of web pages is relatively efficient and simple: there is one login page, and depending on the user's choice of contributor or participant, the corresponding pages will then be displayed.
    - There will likely be some tediousness involved with uploading multiple files, especially if a contributor wishes to upload a large amount of files. This is due to the fact that the user will need to specify TS metadata for every file that they upload.
    - The user will likely not need to worry about the types of files they are uploading—most data sets are already in CSV, XLSX, XLS, or JSON format, which are the data formats that our system will accept when a user is uploading files.
    - The user does not need to worry about converting files to their preferred data format themselves after downloading data—our system will provide functionality to convert data to either CSV, XLSX, XLS, or JSON format at the user's discretion.
- ***Intuitiveness***:
    - The user interface of our system should be intuitive, with each form and selection box being clearly labeled and pleasantly designed.
    - It should be intuitive and easy for the user to upload and download files.
- ***Low perceived workload***:
    - As previously mentioned in "Efficiency of use," there will be some tediousness involved with uploading multiple files, especially if the contributor wishes to upload a large amount of files. While it may not be feasible to upload a number of files on the order of 10^2, it should be reasonable to expect that the user wishes to upload a number of files on the order of 10^1. Our first approach was to have one web page with every file listed and an enormous form where the user would need to fill in the metadata for every file all at once. To reduce the perceived workload of filling out metadata for every file, we will instead present the user with a separate web page for every file, with a small form on it to fill in the metadata.

3.4 Performance Requirements

Multiple users should be able to upload and download files at the same time. This means that the software and database needs to handle concurrent requests. In concrete terms, it is reasonable that the system should be able to handle 100 requests concurrently without a visible change in response time to the user.

As mentioned in 3.5, we think that it is unlikely that users will use this system to upload a forecasting task with an associated TS set that contains over 100 files. While this number of files could certainly be handled, it would be necessary to change our approach for uploading large numbers of files in terms of the efficiency of use. That being said, the system should be able to process the majority of uploads and downloads of files—say, 95% of these transactions—within 1 second.

The system should also be able to quickly create comparison metrics, statistics, and graphs for large sets of data. In measurable terms, 95% of the comparison metrics, statistics, and graphs should be created and presented to the user within 1 second.

3.5 Software System Attributes

In this software requirements specification, we will focus on three main system attributes: usability, portability, and maintainability. We will do our best to take care to incorporate these attributes to the best of our ability when designing the system.

- ***Usability***: The system should be usable. Usability is important because it allows the system to be easily understood by naive users, or users that are unfamiliar with the system or with technology in general. Usability also means that the system operates without fault or error, despite the user's ignorance of the system's function. Lastly, the system should be attractive, but not overbearing. This is important to ensure that the usability requirements (see section 3.3) are met.
- ***Portability***: The system should be portable. Portability is important because it allows the system to be used much more widely: by more users and in more settings. To achieve this, we will make a web application, accessible from anywhere by anyone, so long as the user has internet access. The system will be hosted using a web hosting service, and can be used as long as the user has the correct URL address.
- ***Maintainability***: The system should be maintainable. Maintainability is important because it reduces the effort for software engineers to locate and fix errors in the operational programs of the system. To create a maintainable software, we will ensure that we have correct and thorough documentation of our system's programs. Further, we will attempt to design our system such that it is stable, easy to change, and easy to test.

4 References

IEEE Std 1362-1998 (R2007). (2007). IEEE Guide for Information Technology—System Definition—Concept of Operations (ConOps) Document. https://ieeexplore.ieee.org/document/761853

ISO/IEC/IEEE Intl Std 29148:2011. (2011). Systems and software engineering—Life cycle processes—Requirements engineering. https://ieeexplore.ieee.org/document/6146379

Flores, Juan. (2023). CS 422—Software Methodologies 1—Project 1: Time Series Forecasting and Benchmarks.

Sethi, Ravi. (2023). *Software Engineering: Basic Principles and Best Practices*, 1st ed., Cambridge University Press.

van Vliet, Hans. (2008). *Software Engineering: Principles and Practice*, 3rd ed., John Wiley & Sons.

5 Acknowledgements

The SRS template was provided by Dr. Juan Flores, who changed the original template document created by Dr. Anthony Hornof.

The provided example submission of the SRS document, authored by Ronny Fuentes, Kyra Novitzky, Jack Sanders, Stephanie Schofield, Callista West (https://github.com/JackSanders1998/CIS422Proj2).