

MSE 2017 Project Final Report

Optimization of Wave Energy Converters

Mengyu Li

The University of Adelaide

Adelaide, SA 5005

a1680972@student.adelaide.edu.au

ABSTRACT

Renewable energy has attracted much attention in recent years. It can bring us significant benefits, such as its good economic efficiency and non-pollution. Traditional energy can pollute the environment and be used up easily, so the importance of renewable energy is obvious. The research on wave energy has become increasingly popular [2]. In these two semesters, our group has done some contributions to this area on both research side and software side. In the first semester, we mainly focus on the research side. We have built several machine learning models to predict the captured wave energy. In the second semester, we mainly focus on the software side, we build several applications which integrate with our machine learning models that built in last semester. This report will describe my personal contributions in our project this semester.

In this semester, I have done an Android app, the front-end of two web applications and tried a new machine learning model using tensorflow [1]. Last semester we mainly focus on building different machine learning models that can predict the absorbed wave energy and in this semester, all our applications can use our machine learning models to do predictions. Our models are not only for researchers or programmers now, anyone can try our models by using our Android app or web applications. The Android app is a simple game that allows users to explore how to maximise absorbed wave energy. The background and details of this application will be described later in this report. The web applications allow users to experience how a model is created step by step. It is a good opportunity for non-professionals to experience the entire process of building a machine learning model and use it.

KEYWORDS

Software engineering, Android app, Web Application, Machine learning Models, Angular

1 INTRODUCTION

Our task in this semester is to build applications that allow users to use our machine learning models, which we built in last semester. To describe the functions of our applications, especially the Android app, I need to briefly introduce the background of our machine learning models. Let us assume that we want to set up a wave energy farm. Such energy farm will involve many devices which is called wave energy converters. We simplify a wave energy converter as a buoy, which can capture the energy from the movement of waves. The main purpose of this project is to capture energy by several buoys. To maximise the captured energy, we have more than one buoy in a limited area. Our models are responsible for calculating the overall captured energy of these buoys. The input

data of our model is the coordinates of these buoys and the output data is just the overall energy absorption. The training data can be got in CSV files. These machine learning models have been built in last semester. They are generated by many different software packages with different programming languages, such as Scikit-learn in Python, Multi-poly Regression in Matlab and Multi-layer perceptron in Weka. However, it is difficult for people who do not have any coding knowledge to run these models or train these models. Therefore, we built several applications that run these machine learning models in the back-end and provide a user interface for users.

Building software applications become the major task for our project in this semester. We built an Android application and two web applications. The Android application is very easy to use. Users can change the coordinates of four buoys and get the captured energy instantly. The web applications allow users to train their own models using their own dataset. The details of these applications will be described later in this report. We met many problems when building these applications and we continued to search good methods to solve these problems. The solutions that are related to my personal work will be provided in this report. In addition, I also build a multi-layer perceptron (MLP) model in this semester. This model is built using tensorflow, which is a popular open-source library for machine learning. I thought MLP models are more suitable for our project, so I tried to build a new MLP model this semester. Unfortunately, the result is not better than our previous models, even though I have tried many different parameters. This report will focus on the software side.

Contribution This report will describe my personal contributions in building our software applications. I have built a hybrid Android app, the front-end of two web applications and an MLP model. The machine learning model is not the main objective in this semester, so I will not introduce my efforts on it, but provide more details of our software applications.

Outline This report is organised as follows: In section 2, I will describe how I built the Android application. Section 3 will describe the front-end of our web application. Besides, I will introduce Angular, which is a front-end framework that I used in our project. To explain why I choose this framework, a comparison between Angular and other frameworks are also provided. In addition, this section will also introduce how front-end and back-end connect each other. Section 5 will provide my personal reflection on this final-year project. Finally, I will make a conclusion in the last section.

2 ANDROID APP

I built an interesting Android application in this semester. We show this Android application on the university's open day and Ingenuity show. I provide the idea that we can build a simple game on Android device, because I think it can attract many high school's students' attention. In addition, it provides a good opportunity for students and visitors, who do not have any professional knowledge, to experience our machine learning models. We can introduce how we build our models when they use our applications. I also created a web page that can show a rank list. Such web page can show the scores that users got in the game.

2.1 Functions

This Android app allows users to move four icons on the screen by sliding their fingers. Users can input their usernames and then change the position of four buoys. The coordinates will be sent to our cloud server and the server will calculate the overall captured energy of these four buoys using our machine learning models. After that, the result will be sent back to the Android app and be shown on the tablets' screen instantly. A history-data list is also provided for users. To make the app more attractive, I also created wave animations. Users can know the direction of waves through this wave animations and then modify the positions of buoys to let these four buoys capture more energy. We can see from the figure 1 that the wave comes from left, so if we put these four buoys in vertical direction, the captured energy would be larger.

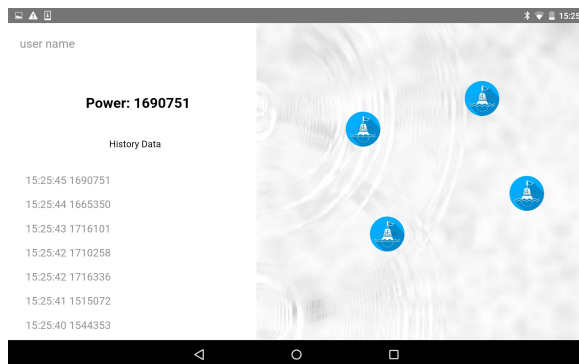


Figure 1: Screenshot of the Android app

In addition, I also built a simple web page to show a rank list. Many users will use our Android app on the open day and they will get different results. Therefore, a rank list is needed to show these results. This simple web page shows the top 6 results and the worst result. It was shown on a big screen during the university's open day. The four buoys on this simple web page will be moved to the corresponding positions as users moved in the Android app. Such movement can let users know what good layouts look like. The screenshot of this web page is shown in figure 2.



Figure 2: Screenshot of the Simple Web Page

The entire process of the Android application and the simple web page can be seen in figure 3. We use Ajax [9] to exchange data between front-end and back-end.

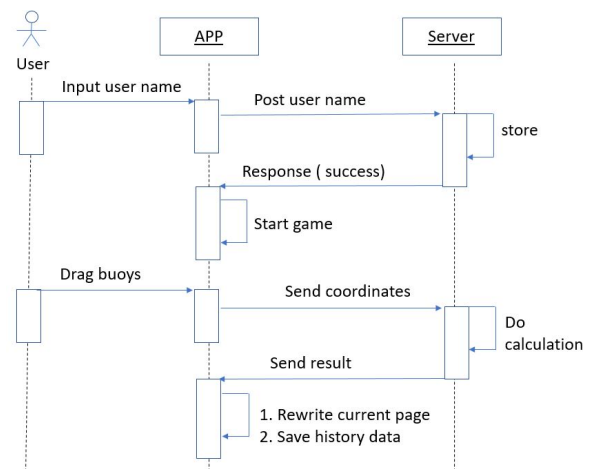


Figure 3: Process of the Android application

2.2 HTML5

This mobile app is a hybrid app. It uses a new technique known as HTML5 [13]. We run it on Android device, but actually it can be used on both Android system and iOS system. The architecture is shown as figure 4. Most basic functions can be realised by HTML5, such as local storage and data exchange. Some other native functions, such as Bluetooth, must be written in Java and objective C, but these native functions can be encapsulated as interfaces and be invoked by JavaScript. Native functions of Android system and iOS can be united and used in the same way. There are many advantages of hybrid app. Firstly, it can use both native functions and HTML5 functions. Compared with traditional app, the hybrid app can use HTML5 functions which are simpler to build. Many public third-party JavaScript libraries can be used in the hybrid app. Secondly, the hybrid can use HTML + CSS to build a more flexible app UI. Lastly but the most importantly, the hybrid app can be run on both Android system and iOS system. Developers can use the same way to invoke functions in different systems. There are also many disadvantages of the hybrid app. Developers need to spend

much time on encapsulating native functions. If there are too many native functions that need to be used, it would be a big trouble. Fortunately, we do not need any native functions in our project. All functions that we need can be built in JavaScript, including the wave animation. I built this animation using a third-party JavaScript library, Velocity.js [3].

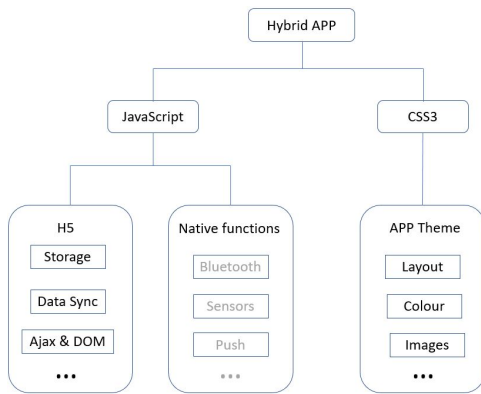


Figure 4: Architecture of mobile app

3 FRONT-END

3.1 Angular

I used Angular[10] to build our web application. Angular is a popular front-end framework which is supported by Google company. HTML is good for declaring static documents, but it cannot meet our requirements that the web view needs to refresh the data frequently. As a result, I select Angular to build our front-end, because it can dynamically rewrite our web view. There are many benefits that Angular can bring to our project. The Angular application is actually a "single-page" application. The "single-page" application does not mean that our website only has one web page. It means the website can dynamically rewrite the current page, rather than loading entire new pages from a server. Our applications can load all major resources from the server at the first time when users visit our web applications, such as some descriptions and necessary icons. It can largely relieve server's burden, because the front-end will not load same resources repeatedly. I made two sequence diagrams to show the differences between Angular and traditional method. Figure 4 shows the process of traditional methods, we can see that if users wish to visit a.html and b.html, the browser need to send two requests to the server.

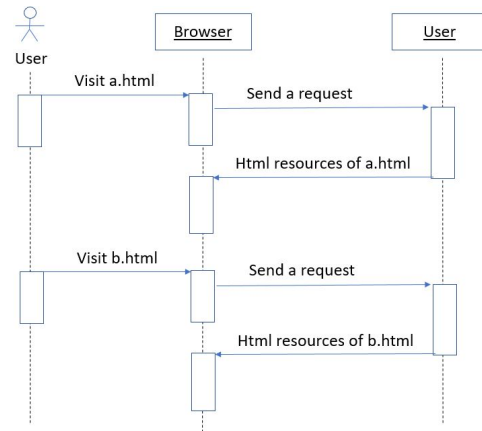


Figure 5: Process of traditional method

However, if we use Angular to build our web applications, the process will be different. The front-end sends only one request to our server, thus relieving our server's burden. The b.html can be shown by dynamically rewriting the current page. Figure 5 shows the process of Angular.

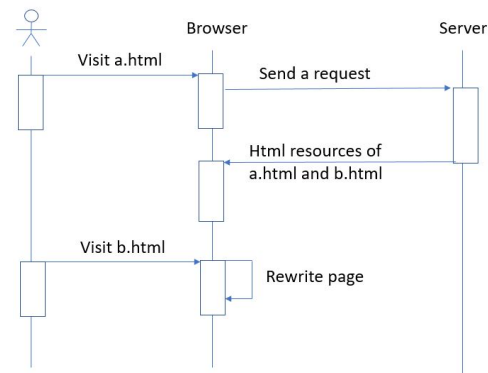


Figure 6: Process of Angular

I also did a simple experiment to demonstrate the performance of Angular. I create two simple pages: a.html and b.html. Firstly, I used a traditional method to build these two pages and used the developer tool of Chrome to record their loading speed. The result is shown in figure 6 and figure 7.

Name	Size	Time	Waterfall
a.html	344 B	7 ms	
b.html	139 B	4 ms	

Figure 7: Process of Angular

Name	Size	Time	Waterfall
b.html	347 B	7 ms	
	142 B	4 ms	

Figure 8: Process of Angular

The total consumed time is $7\text{ms} + 7\text{ms} = 14\text{ms}$. I run this simple code 20 times and the average time consumption is around 14.5ms. Then I tried to use Angular to build these two pages. The web view is the same, but the loading speed is different. The result is shown in figure 8.

Name	Size	Time	Waterfall
a.html	1.6 KB	5 ms	
/a	1.4 KB	2 ms	

Figure 9: Process of Angular

On the one hand, we can see that the total loading time is only 5ms. I run it 20 times as well and the average loading time is around 5.2ms. The web application has loaded a.html and b.html at the same time, so the total time-consumption is only 5ms. As described above, the Angular has loaded all major resources at the first time when users visit the web applications. Therefore, we can know that the loading speed of Angular is significantly faster than traditional methods ($7\text{ms} + 7\text{ms} < 5\text{ms}$). On the other hand, the loading size of Angular seems to be larger than traditional method ($344\text{B} + 347\text{B} < 1.6\text{KB}$). However, if users visit these two pages more than one time, the result will be different. For example, if users click on the hyperlink of these two pages N times, the loading size of traditional methods will be $(344\text{B} + 347\text{B}) * N$, while the loading size of Angular will still be 1.6KB. In this experiment, the Angular script can dynamically rewrite the page rather than sending a request to the server, thus relieving our server's burden and improving the performance of our web applications.

3.2 Compare with other frameworks

Nowadays, many great JavaScript frameworks have been produced by world-class companies, such as [6] and [11]. For example, Angular.js is developed by Google and React.js is maintained by Facebook. When I begin to build a web app, the first question that I need to consider is the selection of the framework. This decision will influence my continued work, so I made a comparison among these popular frameworks and did some research on them. Before I analysed the technical detail of each framework. I need to know how many developers are using each framework. More developers mean that we can find solutions more easily from technology communities, such as stackoverflow.com. We will face many problems during the entire developing process. To check the number of developers of each framework, I searched some useful information from GitHub and StackOverflow. The data is shown in table 1.

Table 1: Figure for different front-end frameworks

	Angular	Ember	React
GitHub Stars	44.3K	14.9K	29.7K
GitHub Contributors	1341	539	581
StackOverflow Followers	127K	16.9K	6.1K
Size	563KB	428KB	131KB

In addition, I looked at the Google Trend and the result is shown as figure 10.

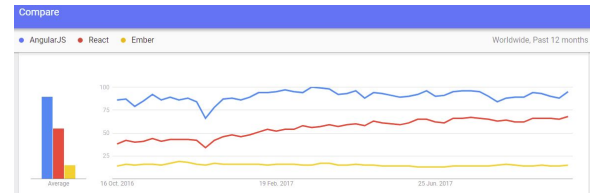


Figure 10: Google Trend

From the table and the screenshot above, we can find that AngularJS is more popular than React.js and Ember.js [15]. during all time. React.js is not as popular as Angular.js. However, we can see from the Google Trend that React.js experienced an increase over the past two years. On the contrary, Ember.js is far smaller than both AngularJS and React and. Besides, it does not have any increase in the past two years, so I will not consider Ember.js anymore.

Both AngularJS and React are JavaScript libraries for building user interfaces. I compare these two frameworks from two aspects: flexibility and maintainability. The conclusion of this comparison may be not accuracy, because it involves my personal feelings and personal developing experience.

Flexibility Both these two JavaScript libraries have great flexibility in use. As a flexible programming language, JavaScript can be used anywhere including browsers, cloud servers, smart phones and even hardware. In our project we mainly use JavaScript in our web app. To include JavaScript libraries in our web apps, we only need one-line code, which looks like this:

```
<script src="https://cdn/angular.min.js"></script >
```

After adding this one-line code, we can use these libraries in our web app directly. Therefore, both these two languages are easy to install. Programmers do not need to consider any dependency and do not need any special integrated development environment. In addition, they both can integrate with other good existing libraries and frameworks easily. From this point of view, both these two frameworks are friendly to developers. However, when it comes to components control, the differences between these two frameworks are obvious. Angular adopts model-view-controller (MVC) [11], which applications into three interconnected parts: Model, View and Controller. MVC separates the controller from the web view. This is a flexible design, because it allows programmers to develop their applications parallelly. In addition, by adopting MVC, the JavaScript code can be reused more easily, thus promoting the efficiency of development. I attach a simple code snippet to describe it more clearly.

```

1 <body ng-controller="myController">
2   <div>Hello {{ something }}</div>
3 </body>

```

HTML

```

1 app.controller('myController', function($scope) {
2   $scope.something = "World!";
3 });

```

Script

This is a simple demo to show how Angular works. All these codes are run in the browser. The HTML code is used to render the web view, while the script code is responsible for controlling. This is a flexible design for me, because when I build web view, such as modifying the CSS code, I do not need to consider JavaScript functions. When I build functions, I do not need to worry about web view. On the contrary, the React.js is not as flexible as Angular, because it adopts "virtual DOM" [5], which integrate the view code (HTML) with controlling code (JavaScript). To understand "virtual DOM", DOM means Document Object Model, which can be simply seen as HTML code for web developers. Sometimes it can be called HTML DOM as well. "Virtual DOM" means that HTML code is generated by JavaScript dynamically. A simple demo is shown below:

```

1 <div id="example"></div>
2 <script>
3   ReactDOM.render(
4     <h1>Hello, world!</h1>,
5     document.getElementById('example')
6   );
7 </script>

```

Script

We can find that the HTML code and JavaScript are closely interwoven. It is not flexible for developers, because when we need to modify the web view or controller, we have to change HTML code and JavaScript at the same time. It will increase the burden on developers. After comparing these two simple demos, I personally think that Angular is more flexible than React.

Maintainability Maintainability is another crucial factor that we need to consider when we evaluate different frameworks. This section will describe the maintainability of Angular and React from my point of view. Angular use modules to control apps. We can simply consider a modular as a container for the different parts of our web app. Each module is independent of other modules, so it is convenient for developers to maintain the whole app. Angular can naturally help us divide the large application into different modules and each module can be further divided into different controllers. The architecture is plotted in figure 11.

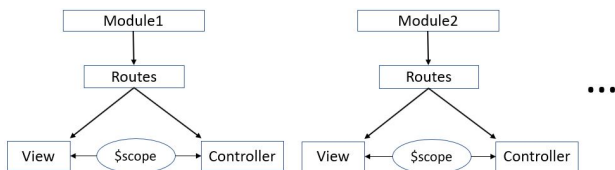


Figure 11: Architecture of Angular in our project

This structure makes Angular applications become easier to be maintained. When developers modify one of these modules, they do not need to worry that the module they are building will influence other modules. Functions in each module can be realized independently and will not influence other functions in other modules. In addition, when programmers conduct unit tests, they only need to load related modules, thus making the testing work more conveniently. Therefore, I believe that the app which is based on Angular can be maintained efficiently.

React use JavaScript XML (JSX) to render web view. JSX is an XML-like syntax that can be used to generate HTML code. It can be known as HTML in JavaScript. When using JSX, developers do not edit HTML code directly. By contrast, they need to edit JSX using JavaScript and then browsers will transfer JSX to HTML code. From this point of view, JSX is a kind of "raw code". It is unreadable for web view builders, because JSX cannot generate web view directly, so it is not convenient for developers to maintain their applications. In chrome, which is a famous browser produced by Google company, developers can modify the HTML code under development mode and the web view will be changed instantly. However, if we use JavaScript code to produce HTML code, just like JSX in React, the web view cannot be changed in real-time. If we hope so, we need to use other third-party tools, such as gulp. As a result, the web view is a little difficult to maintain in React. In addition, the mix of HTML code and JavaScript code will make maintenance work become more difficult, because developers always modify the HTML code and JavaScript code at the same time. React does not have a good controller to help developers maintain functions.

After comparing these two libraries. I think the maintainability of Angular is better than React. Angular developers can test and fix each module handily. The web view and controller are also separated in Angular applications, which is also convenient for continued maintaining. On the contrary, the JSX in React, which is a kind of "raw HTML", makes the front-end become more fragmented and unreadable. Both the two simple demos, which has been provided above, print "hello world" in a web page, but in different ways. Obviously the Angular is more readable. After comparing several different frameworks, I think the maintainability of Angular is better, so eventually I decided to build our web applications using Angular.

3.3 Ajax

This section will introduce how the front-end and back-end communicate with each other and why I choose this communicating method. The connection between clients' browsers and our cloud servers is a critical issue. In our Android application, when users drag buoys, the front-end will frequently will send coordinates to cloud server and get results back. In our project, we have two applications that are needed to consider such connection. The first one is our main website and the second is our mobile app. There are many ways to build connections between clients' browser and cloud server, such as Ajax, web socket and Message Queuing Telemetry Transport(MQTT) [7]. This section mainly describes Ajax and explains why I use it.

Ajax Ajax means "Asynchronous JavaScript and XML". It allows browsers to exchange data with our server flexibly. In our project,

we use Ajax as our main communicating method, because it can bring our project several benefits. The first benefit is that Ajax can update website contents without refreshing the page [14], thus improving user experience and speeding up the loading process. In our project, clients' browsers need to get messages from sever frequently. The second benefit is that Ajax can help us separate front-end and cloud server completely. My group members have built several Application Programming Interfaces (API) and we use API to transfer messages between our front-end and server. Figure 12 shows how Ajax and API work in our project.

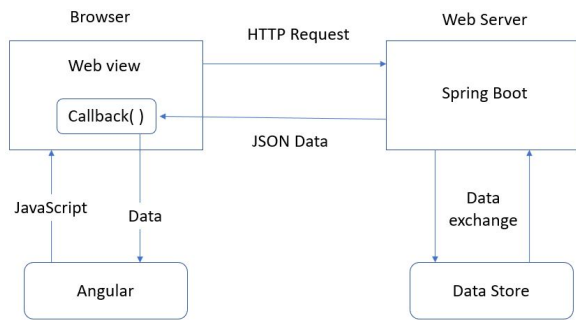


Figure 12: Ajax work

The front-end post an HTTP Request to web server, and then the sever will do some calculations and exchange data with database. After that, the server will send the result, which can be known as "response", back to the front-end. After the front-end receive data, it will invoke callback() function Such function allows front-end to save the data as variables in JavaScript and do further work, such as rewriting current page and doing some calculations.

3.4 MVC in our project

In this semester, I have built the front-end for three new modules using Angular. They are:

- Dataset
- Model
- Prediction

The back-end side is built by my group member, so I just introduce front-end side in this report. The front-end code contains three parts: view, controller and model (MVC) [8]. These three parts are known as MVC framework. The view side is responsible for transferring HTML code to web view. The controller side is responsible for dynamically rewriting current page and exchange data with the back-end. The model side is used as a router. Such router is responsible for switching between different pages when users click on different buttons or icons. The architecture of the front-end is briefly shown as Figure 13. We have three modules and each module has an MVC sub-framework. These three sub-frameworks can be switched by a router, which is realised in pages.module.js.

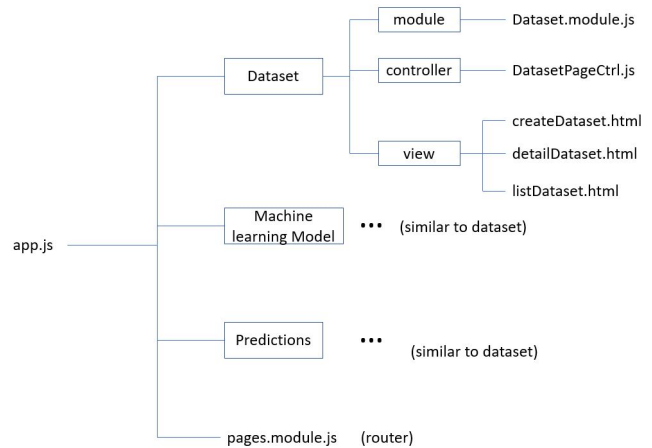


Figure 13: Structure of Angular in our project

Several new templates have been built for our web view in this semester. All these templates are beautified by Bootstrap.min.css, which is a free and open-source Cascading Style Sheets(CSS). The web view is not so important in our project, so I just select Bootstrap to build the web view of our web application, though there are also many other popular CSS frameworks. Bootstrap [4] is an open-source CSS library that can help us beautify our web view. It can be included by one-line code:

```
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" />
```

HTML

I have built three templates for our web applications using Bootstrap. The first template is responsible for showing tables in our web applications, which is used in listDataset.html, listModel.html and listPrediction.html. The screenshot of this template can be seen in figure 14.

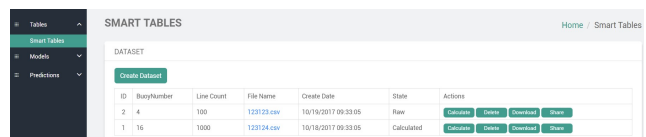


Figure 14: Structure of Angular in our project

There are four buttons in each row: calculate, delete, download and share. When users click on one of these four buttons, the front-end will invoke the corresponding API using Ajax and then dynamically rewrite current page. Take figure 14 as an example, if users click on the "Delete" button in the second row, the front-end will post a request, which includes the dataset-id of the dataset in the second row, to our server to notify the back-end that the user wishes to delete this dataset. After our backend successfully deletes this dataset from database, it will send a response back to the front-end and then the front-end will rewrite the current page. In this example, the front-end will refresh the data in the table and

users will find the second dataset disappeared. The entire flow is provided in figure 15.

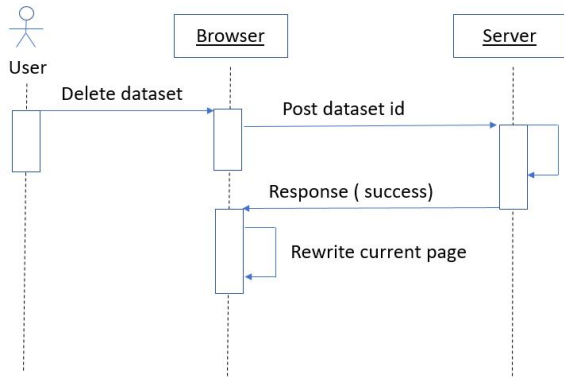


Figure 15: Flow of delete a dataset

The second template is responsible for showing detail information of a specific machine learning model or dataset, such as some parameters of a model. Such data is obtained in JSON format and I need to convert it to readable data format. The screen-shot of this template is provided in figure 16.

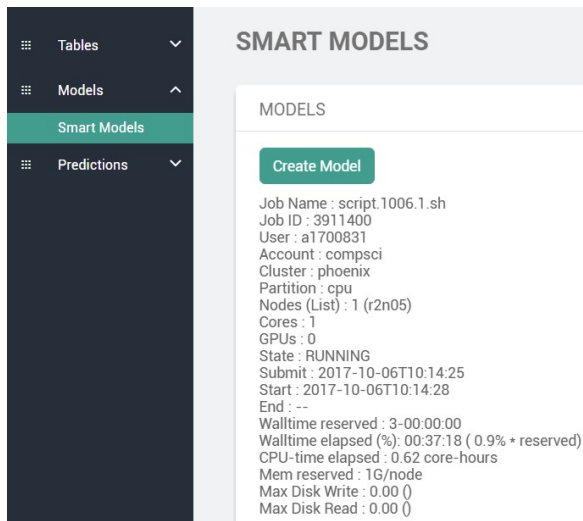


Figure 16: Structure of Angular in our project

The third template is responsible for providing a form for users. In our project, users will submit forms frequently. For example, when they select a machine learning model, they need to set parameters of the model and then click on "submit" button to upload these parameters. There is a challenge in building these forms. Different users may choose different models and different models have different parameters, which means I cannot use a static form, because it cannot meet all requirements. Some of the forms have to be generated dynamically according to different parameters. After

users submit the form, the front-end will send these parameters to our cloud server using Ajax.

Another new page that I have built in this semester is login.html. The screen-shot can be seen from the figure 18. I use the Bootstrap.css to beautify such page.

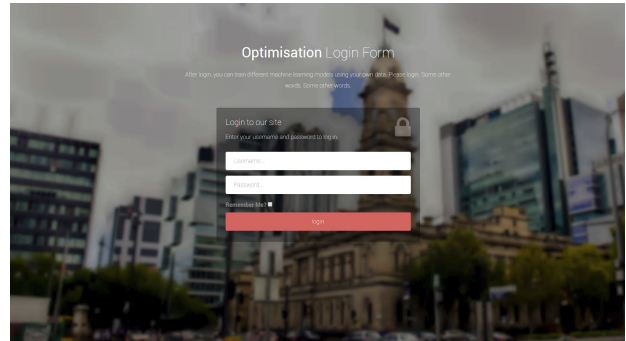


Figure 17: Screenshot for login.html

I tested the front-end code during three periods. The first period is during the process of running. I used chrome's developer tools to check whether I have any bugs in my code. This is a useful tool that can detect many errors of our web application. Figure 18 is an example to show how it finds bugs.

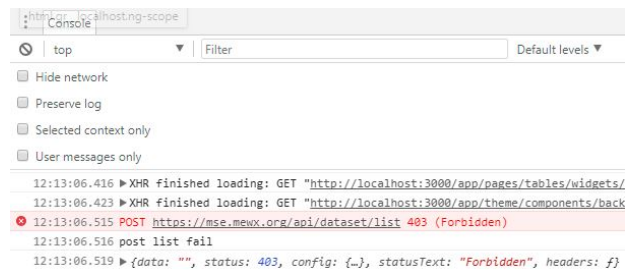


Figure 18: Chrome developer tool

We can see from the figure 18 that there is a 403 forbidden when the front-end tried to post a message to the back-end. Such error message will not appear in the web view that makes users confused, but the chrome developer tool can help us detect such error easily. The second period is during the process of coding. In this period, the plug-in of sublime called JSHint[12] can help me check grammar errors. The sublime is a popular editor that have a vast number of excellent plug-ins and JSHint is one of them. JSHint can highlight the grammar errors when developers editing JavaScript code. The Third period is when I finished my code work. I used unit-test to detect bugs in my code. As I have described above, Angular applications are module-based applications, which means that it is easy for developers to test a big application part by part. For each module in the front-end code, I built a testing module which contains a set of input parameters and expected output data. If the real output values are the same as real output values, the original module can be seen as correct. If not, I will check this

Table 2: Detected errors

Error type	Dataset	Module	Prediction
404	0	0	0
403	5	3	0
500	2	1	1
JS Grammar	3	6	4
Loading too long time	2	1	0
xxx is not a function	1	2	0
Too much recursion	1	0	0
browser not compatible	1	1	1
HTML Grammer	1	2	0

module individually. Table 2 shows the error that I detected at the beginning. Some of these errors are generated by the server, such as 500 internal error, and they can be detected when front-end send a request to the back-end.

4 REFLECTION

In this final-year project, I have also learnt many useful knowledges on both machine learning and software engineering. For machine learning, I have tried many models using Matlab and Python. I grasped many useful tools and packages, such as tensorflow in Python and libsvm in Matlab. Also, I have learnt many useful models, such as multi-poly regression [16] and multi-layer perceptron. These skills will also be useful for my future work. For software, I have learnt how to build front-end using Angular and have a deep understanding on front-end architecture. Besides, I have also learnt how to develop mobile applications using html5. I have learnt so many skills in my final-year project. In addition, I also learnt how to use GitHub to manage my project code. I know how to set milestone, how to negotiate with our clients and how to work with other group members. Through this final-year project, I also got familiar with some useful tools, such as slack and zenHub. To sum up, I really get a lot from this project. For future work, the research on the machine learning models should be continued, because our existing models are not the best. Specifically, there are three aspects can be considered. The first one is the error rate. The decrease of error rate is always worth pursuing. Currently our best machine learning model is multi-layer perceptron (MLP), which is found by my group mate in last semester. To achieve a better result, we can try other models or just improve the parameters of the MLP in the future. The second aspect is that we can consider more factors in the real environment. In our final-year project, we only consider the coordinates of the energy converters and ignore other factors. Apparently, there are many other factors needed to be considered in reality, such as the size and the submergence depth of wave energy converters. Our existing models do not take these factors into consideration. The third aspect that can be improved in the future is our applications. We designed many functions, but some of them cannot be finished within limited time.

5 CONCLUSION

In this final-year project, I have made some contributions to the wave energy converters. In the first semester I mainly focus on

research side, while in the second semester I mainly focus on the software side. For research side, I have built several machine learning models in the semester 1. The details of this part have been described in the last semester's report. For software part, I have built the front-end of our web application and an Android app. These applications can invoke the machine learning models which we built in the first semester, so our work has a logical progression. In this semester, we design a poster and show the applications in the university's open day and ingenuity show. By showing these applications to the public, we introduce to others the meaning of wave energy converters and how we build our machine learning models for this project.

REFERENCES

- [1] Chen Abadi M, Barham P. 2016. TensorFlow: A System for Large-Scale Machine Learning.
- [2] Falcão A Clément A, McCullen P. 2002. Wave energy in Europe: current status and perspectives.
- [3] Flanagan D. 2006. In *JavaScript: the definitive guide*.
- [4] Rahman S F. 2014. In *Jump Start Bootstrap*.
- [5] Psaila G. 2008. In *Virtual DOM: An Efficient Virtual Memory Representation for Large XML Documents*.
- [6] Salazar G A Gómez J, García L J. 2013. In *BioJS: an open source JavaScript framework for biological data visualization*.
- [7] Stanford-Clark Hunkeler U, Truong H L. 2008. In *A MQTT-S—A publish/subscribe protocol for Wireless Sensor Networks*.
- [8] Deacon J. 2009. In *Model-view-controller (mvc) architecture*.
- [9] Garrett J J. 2005. Ajax: A new approach to web applications.
- [10] Deshmukh Jadhav M A, Sawant B R. 2015. In *A Single page application using angularjs[J]*. *International Journal of Computer Science and Information Technologies*.
- [11] Mehta D, Jain N, Mangal P. 2015. In *AngularJS: A modern MVC framework in JavaScript*.
- [12] Perez J, Kovalyov A, Kluge W. 2014. In *Jshint, a javascript code quality tool*.
- [13] Pilgrim M. 2010. In *HTML5: Up and Running: Dive into the Future of Web Development*.
- [14] Ricca F, Marchetto A, Tonella P. 2008. In *State-based testing of Ajax web applications*.
- [15] Pattabiraman K, Ocariza F, Bajaj K. 2013. In *An empirical study of client-side JavaScript bugs*.
- [16] Thorp J S Wang T, Pal A. 2015. In *Multi-polytope-based adaptive robust damping control in power systems using CART*.