MSE 2017 Project Final Report

Optimization of Wave Energy Converters

Chenwei Feng
The University of Adelaide
Adelaide, SA 5005
a1696585@student.adelaide.edu.au

1 INTRODUCTION AND MOTIVATION

The Paris Agreement is an international agreement which designed to slow the trend of global warming, as one of the most influential countries over the world Australia signed in Paris Agreement and would make contributions to the world environment. Australia Climate Change Authority published a report [3] demonstrated that to reduce the risk of dangerous climate change, it needs Australia to reduce the consumption of carbon on electricity systems by 2050. In 2016, a study [17] indicated that Australia consumed 248 terawatt hours of electricity between 2013 and 2014 and the large- scale renewable energy technology only provided almost 7% of this number. To boost the development of large-scale renewable energy technology and achieve the target of Paris Agreement, the Australia Climate Change Authority set the new goal in 2016, which was that by 2050 the electricity generation of large-scale renewable energy technology should increase to the 65% [16] of total electricity generation. In this case, it needs large-scale renewable energy technology has a rapid development to help Australia achieve the target.

At this stage, the wind and solar energy dominate renewable energy in Australia. However, Mark etc. considered that the wave energy is indispensable in the development of renewable energy. It not only because Australia has the most abundant wave energy resource in the World [5], the account of electricity generation can achieve the 1,800-terawatt hours, but also because comparing with the wind and solar energy, wave energy has some unique advantages, such as, less variable and more predictable. Also, it needs to notice that approximately 60% of the world's population lives within 60 kilometres of a coast, which means the distance between the wave energy sites and users is closer than other renewable energy. The closer distance can minimise the transmission issues, so that improve the efficiency of wave energy. How to develop and take advantage of the wave energy will the determine the environment of Australia in further, even the all-world climate.

As we know, the wave energy cannot be directly used, and it needs to be captured and converted to electrical power by particular devices. A device which can achieve this is called Wave Energy Converter (WEC) or buoy. The capacity of energy captured by a single buoy is limited. Therefore, multiple buoys are necessary for large-scale energy production. A set of buoys which work in close distance is called a wave energy farm. Optimising wave farms is a challenging problem, as the buoys interact in highly sophisticated ways with each other – which can result in negative as well as positive effect [13]. How to solve the position of each buoy in the buoy farm became a core issue. One Ocean Wave Energy Research Group came from the University of Adelaide proposed an accurate Matlab model [13] [1] [6], which can simulate the process of buoy farm working and get the account of converted electricity power. The group want to use this model to optimise the position of buoys; however, the running time of this model is an impediment. As the number of buoys increases, the running time of model will become unacceptable for optimisation work. As this reason, the primary purpose of our project is to use machine learning models to build a prediction model which can replace the Matlab model. At the same time, we need to develop a website, which contains our different machine learning models. Users who are interested in the position of buoys can use our web app to execute their experiments.

2 RELATED WORK

Machine learning algorithms widely applied in research area for prediction, and predicting the energy power is the most pivotal technology in power industry. For this reason, many studies related to use machine learning algorithms to predict the power output proposed during the recent decades.

In 2005, a model named auto-regressive integrated moving average (ARIMA) [10] was proposed by Guoyang etc. This model was used to forecast the speed and power of the wind, which got a good result. According to this study [10], Sfetsos conducted a survey [18] which compared the performance of ARIMA model and Artificial Neural Network (ANN) on the prediction of wind speed. This study indicated that ARIMA model has a better performance than ANN model when the scale of training dataset is not too large. As the increasing of the size of the dataset, the result of the ANN model was more accuracy, which means that the performance of the ANN model may go beyond the ARIMA model. The Multilayer Perceptron a class of feedforward artificial neural network, Li [15] take advantage of recurrent multilayer perceptron (RMLP) neural networks to predict the wind power and showed that the RMLP could be used to predict the wind power in changing wind conditions.

THERMO syphon solar heater is one device which can use solar energy to get hot water for human usage. Kalogirou et al. [14] used ANN to predict the performance of these devices. The performance measured regarding the useful energy extracted and the stored water temperature rise. The ANN was trained using the performance data for four types of systems. The output of ANN is the useful energy extracted from the system and the water temperature rise. Seven input units, 24 hidden layers and two output layers comprise the network model.

There are three studies about the wave energy estimation. The article [8] based on based on Gaussian process regression Upper Confidence Bound with Pure Exploration algorithm (GPUCB-PE) to predict and optimise the wave energy output with 40 boys. The study [12] used eight different ordinal and nominal classifiers and

one support vector regression algorithm to evaluate the wave energy estimation, this study mainly concerned wave height and energy flux prediction in a six time-horizon. The study [9] used the data which comes from 22 sites in worldwide to predict the wave energy. As the data from the real world, the predictions will be influenced by weather patterns and bathymetry and the prediction mainly determined by the probability distribution, if the distribution has heavy tails, the accuracy of the prediction will very low.

In conclusion, most of these previous studies are about wind energy and solar energy. There are a few studies about wave energy so that wave energy can be regarded as a new research area. Compare to previous studies, our machine learning model based on previous articles [13] [1] [6] does not need to simulate the process of wave energy conversion to compute the power output, and we do not concern the wave height. These works were done in Matlab model, the only work of our machine learning model is that according to the data which comes from the Matlab model to predict the wave energy

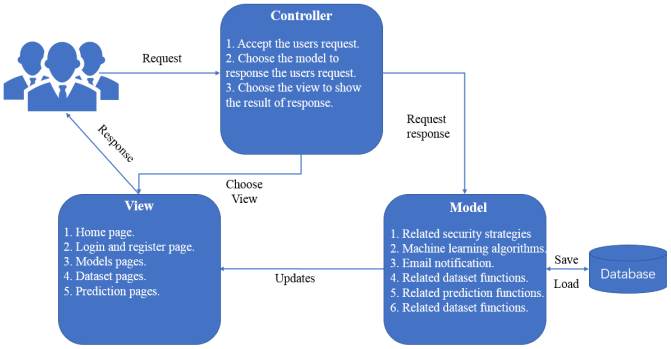
3 CONTRIBUTIONS

This part divided into three parts. The first part is the architecture of our website and the second part is the software engineering part, and the last one is the machine learning part.

3.1 Software architecture

We used the Model–view–controller (MVC) pattern to develop the website. The reason why we choose the MVC pattern is that the MVC is a dynamic programming design that simplifies subsequent modifications and extensions to the program. In addition, this pattern makes the program structure more intuitive by simplifying complexity and it is easy for the team to assign the jobs according to the features of team members. The architecture of our website mainly divided into two parts: the front-end and back-end. The back end contains web back-end and HPC back-end. The front-end page belongs to the view of MVC, Mengyu was responsible for this part. I and Yuanzhong were responsible for the web back-end, and Yuanzhong was also responsible for the HPC back-end. Figure 1 shows the architecture of our website. The front-end and back-end

Figure 1: The architecture of our website



are connected by the APIs. The APIs of our website can mainly divide into six parts, they are as follows:

- General API
- User related API
- Dataset related API
- Model related API
- Prediction related API
- Sharing related API

As all of this APIs is finished by Yuanzhong, so I would not detailed introduce this part in this report. The detail information can be found in Yuanzhong's report or the Wiki page [20] of our website.

3.2 Software engineer

In the second semester, we changed the language of back-end from PHP to Java. We choose the Spring Boot as the framework of the website because our team member Yuanzhong considers that the Spring Boot makes it easy to create stand-alone Spring applications. In other words, Spring Boot makes the encoding, configuration, deployment and monitoring easier. The Sprint Boot based on Java, this makes me need to re-develop the security strategies for our website. To better integrate with the spring framework, reduce the possibility of errors in the code integration process, I choose Spring Security which also comes from the spring framework to implement the security strategies. The detail seasons why I prefer the Spring Security 4.0 will be described in evolution section.

3.2.1 Authentication and Access Control.

The primary security of our website can boil down to two issues, that is authentication and authorization, the former means who are you, the latter represents what are you allowed to do. The Spring Security provides the example of the authentication and authorisation, and it is not very hard to implement these two functions if following the instance. However, I wanted to figure out how they work, so I conducted a more in-depth study about authentication in Spring Security. In this section I will introduce some core Java classes and interfaces in Spring Security for authentication, the dependence among them constitutes the architecture of authentication.

3.2.1.1 SecurityContextHolder.

SecurityContextHolder is used to store information about the security context. Such as who is the user of the current operation, whether the user has authenticated, and which role permissions he has etc. These are all stored in the SecurityContextHolder. SecurityContextHolder defaults to using the ThreadLocal policy to store authentication information of users. ThreadLocal means that this is a thread-bound strategy. Spring Security automatically binds authentication information to the current thread when the user logs in and automatically clears the current thread's authentication information when the user exits. Since the identification information is thread-bound, it can be got through use static methods anywhere. A typical example of obtaining the current login user's name shown below:

```
Object principal = SecurityContextHolder.getContext()
        .getAuthentication().getPrincipal();
if (principal instanceof UserDetails) {
  String username = ((UserDetails) principal).getUsername();
 } else {
  String username = principal.toString();
```

```
}
```

The getAuthentication () function gets the authentication information, then using the getPrincipal () function can get the Identity information. UserDetails is an interface of Spring Security, and its job is the encapsulation of identity information of users. UserDetails will be described in the following sections.

3.2.1.2 Authentication.

Let's see the source code first:

```
package org.springframework.security.core;
public interface Authentication extends Principal, Serializable {
        Collection<? extends GrantedAuthority> getAuthorities();
        Object getCredentials();
        Object getDetails();
        Object getPrincipal();
        boolean isAuthenticated();
        void setAuthenticated(boolean var1) throws
           IllegalArgumentException;

}
```

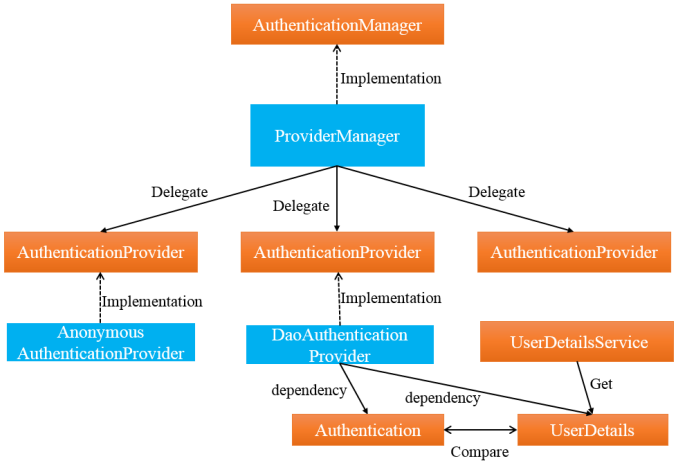
It can find that the Authentication is one of the interfaces, which inherits from the Principal class. With this interface, a list of permission information, passwords, user details, user identity information and authentication information can be got. The detail of this interface as follows:

- GetAuthorities (): a list of permission information, defaults to some implementation classes of the GrantedAuthority interface, usually a string of strings representing the permission information.
- GetCredentials (): password information, a password string entered by the user, usually removed after authentication, to ensure security.
- GetDetails (): detail information, the implementation interface in the web application is usually WebAuthenticationDetails, which records the visitor's IP address and the value of the session Id.
- GetPrincipal (): the most crucial identity information, in most cases, it returns the implementation class of the UserDetails interface, and it is one of the common interfaces in the framework. The UserDetails interface will be highlighted in the following section.

3.2.1.3 AuthenticationManager.

I think the Spring Security is not very friendly for the developers who are the first time use it. Because there is some similar Spring authentication class, such as AuthenticationManager, ProviderManager and AuthenticationProvider, there are many StackOverflow pages about questions about them, which means they always make developers confused. Combine the guidebook [2] and some references on the internet, and I consider that the AuthenticationManager is the core interface for authentication and it is the starting point of initiating authentication. In the real world, websites may allow users to log in using a username + password while allowing users to use email + passwords, mobile phone Numbers + passwords, or even let users log in with their fingerprints. Therefore, the AuthenticationManager is not for direct verification, and the most commonly used implementation of AuthenticationManager is ProviderManager, the ProviderManager maintains a List of List < AuthenticationProvider >, which holds multiple authentication mechanisms, which is the application of the Delegate mode. In other words, the start point of core authentication is always AuthenticationManager, ProviderManager is used to implement the interface AuthenticationManager, and ProviderManager has a chain of AuthenticationProvider instances, which provides different authentication mechanisms, such as name + password (UsernamePasswordAuthenticationToken). The Figure 2 shows the architecture of Authentication.

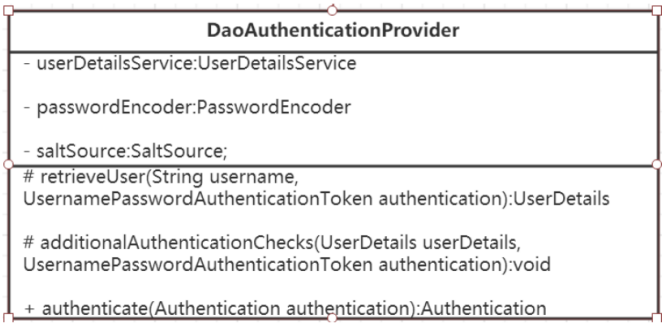
Figure 2: The architecture of Authentication



3.2.1.4 DaoAuthenticationProvider.

DaoAuthenticationProvider is the most commonly used implementation of AuthenticationProvider. As the name implies, Dao is the abbreviation of data access layer, which also means the implementation idea of this authentication device. The UML class diagram of DaoAuthenticationProvider is shown in Figure 3

Figure 3: UML class diagram of DaoAuthenticationProvider



According to this UML class diagram, in Spring Security, the submitted username and password are encapsulated into UsernamePasswordAuthenticationToken, and the task of UserDetailsService is to load the users, which according to the username. For the DaoAuthenticationProvider, the retrieveUser () can get the UserDetails, and the additionalAuthenticationChecks method is responsible for comparing the UsernamePasswordAuthenticationToken with UserDetails, if this method does not throw the exception, this

means the authentication is successful. The process of the comparison associates with PasswordEncoder and Salt, these two concepts had been introduced in the previous report [4].

3.2.1.5 UserDetails and UserDetailsService.
In the above section, the interface UserDetails is mentioned many times, it represents the most detail information of users and contains some necessary user information fields. The source code of this interface is shown as follow:

```
public interface UserDetails extends Serializable {
    Collection<? extends GrantedAuthority> getAuthorities();
    String getPassword();
    String getUsername();
    boolean isAccountNonExpired();
    boolean isAccountNonLocked();
    boolean isCredentialsNonExpired();
    boolean isEnabled();
}
```

The getCredentials () of Authentication and the getPassword () of UserDetails need to be treated differently. The former is the password certificate that the user submits, the latter one is the correct password of the user, and the authenticator is the comparison between these two.

UserDetailsService only load user information from a particular place (typically a database), the common implementation classes of UserDetailsService are JdbcDaoImpl and InMemoryUserDetailsManager; the former is loading users from the database, the latter load users from memory. As our website need save the information of users, the InMemoryUserDetailsManager class isn't obviously suitable for us, because it loads users from memory. What we needed is to load users from the database, so we used the JdbcDaoImpl class.

3.2.2 The process of authentication.
After introduced some important classes and interfaces in Spring Security for authentication, the process of authentication should be concluded as follows:

(1) The username and password are obtained by the filter, and then they encapsulated as UsernamePasswordAuthenticationToken, which is an instance of Authentication interface.
(2) The token from step 1 will be sent to the instance of AuthenticationManager to execute the verification.
(3) If verification is successful, AuthenticationManager will return an Authentication instance which is full of information, such as permission information, identity information, details information, but passwords are usually removed for security.
(4) By calling the SecurityContextHolder.GetContext(). SetAuthentication(...), pass in the returned authentication object to establish a security context.

3.2.3 Remember-Me.
Remember-Me refers to the site can remember the identity of logged in users between the session, specifically, it means after users' successful certification a certain period, they can no longer need to enter the username and password to log in, the system will automatically log in for users. The principle of Remember-Me is that Firstly after users successfully log in the system, the server will send a cookie to the client browser. Then if the next time users' browser accesses the server, the server can automatically detect the client cookies. At last, according to the cookie value trigger automatic login operation. Remember Me is a trade-off point, its target is to improve the usability, but it will reduce the security to some extent. Spring Security provides two ways to implement remember-me, the first way uses hashing to preserve the security of cookie-based tokens and the second way uses a database or other persistent storage mechanism to store the generated tokens [7]. The principle of the first way is that After users choose to the remember me and successfully log in, and Spring Security will generate a cookie and send it to the client browser. The cookie value is composed of the following:

base64(username+":"+expirationTime+":"+md5Hex(username+ ":"+expirationTime+":"+password+":"+key))

The generated cookie is valid only before the expiration time, and the username, password in this cookie must be same with that which was used by users to log in. It needs to note that if we use this mechanism to implement the remember me, it will have a big risk. That is after users get the cookie from the system, any user can use this cookie to automatically log in the system before the expiration time, which means if attackers steal users' cookies, then attackers can use this cookie to log in the system. What's worse, if users cannot realise that their cookies are stolen, then their account will remain in the stolen state until their change the password. Obviously, from the security view, this kind of implementation of remember me is not what we wanted. But we still want to implement the remember me, because it can improve the usability of our website. To make a balance between the security and usability, I choose the second way to implement the remember me, the concrete process of it is as follows.

(1) After the user selects the "remember me" and successfully log in, the system will save the username, the randomly generated serial number and the generated token into a database table, at the same time the combination of them create a cookie and this cookie is sent to the client browser.
(2) When next time the not logged in user access to the system, the system first checks the cookie, if the corresponding cookie contains the username, serial number and token are same with that saved in the database, then this user is regarded as verified. The system will generate a new token to replace the corresponding old token in the database, the serial number remains the same, at the same time, remove the old cookie. At last, the system will send a new cookie which contains the new re-generated token, the old serial number and username to the client.
(3) When the cookie is checked, if the username and the serial number contained in the cookie can match that in the database, but the token does not match, this is most likely because this cookie has misappropriated. And since the attacker has used your authenticated cookie to access to the system, and this action causes the old token to fail, because the new token is created. At this time, Spring Security can find the cookie stolen, and it will delete all the token records which are related to the current user in the database. In this way, the attacker could not use the

original cookie to log in again, at the same time, the system will remind users the possibility of its accounts stolen.

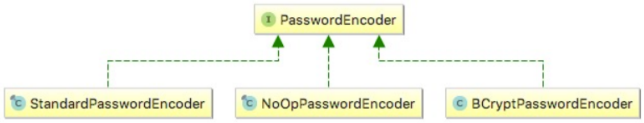
(4) If the corresponding cookie does not exist in the database, or the containing username and serial number are not consistent with that in the database, then the user will be directed to the login page.

In step (1), I used the universally unique identifier (UUID) provided by Java to get a unique randomly serial number. From the above process, it can find that this implementation still has potential risk, that is the attacker still can steal the cookie and access the system before the next time user log in the system. Although this is a risk in our system, I still think our system is secure enough. Because the attacker is hard to steal the cookie, we customise XSS protection in our website.

3.2.4 *Password Encrypt.*

In Spring Security, the interface of password encrypt is named PasswordEncoder, the class diagram of it shown in Figure 4. From the Figure, we can find it has three implementations. I used the BCryptPasswordEncoder to encrypt the users' password. The implementation of the BCryptPasswordEncode is supported by widely used "Bcrypt" algorithm to encrypt the passwords. Bcrypt is the deliberately slow algorithm that using a random 16-byte string, also called salt, to protect the passwords. It has a parameter named "strength", the function of this parameter is to tune the workload of this algorithm, the range of this parameter is from 4 to 31, and if the value is higher, it means this algorithm needs more workload to calculate the hash. Considering the performance of our web application, we want to make a balance between the security and performance, so we use the default value of strength, which is 10. Compare to some traditional encrypt algorithm, the advantage of Bcrypt is that it can slow the speed of hacking the password. If an attacker wants to hack a password encrypted by Bcrypt, it may need this attacker take few months.

Figure 4: The class diagram of PasswordEncoder



3.2.5 *Session Management.*

For Web applications, the primary principle of security is not to trust data from the client, to ensure that the data is validated and filtered, and then to use the data in the application. However, because of the stateless nature of HTTP, to maintain state between requests from the same user, the client must send a unique identifier (session ID) to represent its identity. While this violates the principle of security, the maintenance of the state gives us no alternative. This also makes the session a weak link in the Web application. Spring Security provides a good session management mechanism which allows developers to control the HTTP Sessions.

3.2.5.1 *When is the season created?*

Spring Security provides four ways for developers to control precisely when the session gets created and how Spring Security will interact with it:

- Always – If there is no session exist in the system, Spring Security will always create one.
- ifRequired – Spring Security only create a session when the system needed one.
- Never – Spring Security will never create a session, but it can use the sessions if they have existed.
- Stateless – Spring Security will never create and use the session.

The Java configuration is as follow:

```
    @Override
protected void configure(HttpSecurity HTTP) throws Exception {
    http.sessionManagement()
        .sessionCreationPolicy(SessionCreationPolicy.IF_REQUIRED)
}
```

It's crucial to understand that this configuration only controls what Spring Security does – not the entire application. For our website, I choose let Spring Security create a session when it needs one – this is "If Required".

3.2.5.2 *Session Timeout.*

After the session has timed out, if the user sends a request with an expired session id, they will be redirected to a URL configurable via the namespace. Similarly, if the user sends a request with a session id which is not expired, but entirely invalid, they will also be redirected to a configurable URL. The corresponding Java configuration is as follow:

```
HTTP.sessionManagement()
  .expiredUrl("/sessionExpired.html")
  .invalidSessionUrl("/invalidSession.html");
```

3.2.5.3 *Session Fixation Protection with Spring Security.*

Session fixation attack is that using the server's Session invariant mechanism to obtain authentication and authorization from others, then impersonate others execute some activities. In Spring Security, I choose to use the migrateSession option to protect the session. The working principle of the migrateSession is that on authentication a new HTTP Session is created, the old one is invalidated, and the attributes from the old session are copied over. Using this function can effectively prevent the Session Hijacking. The corresponding Java configuration is as follow:

```
HTTP.sessionManagement()
  .sessionFixation().migrateSession()
```

3.3 Machine Learning

The research work in this semester continued from the work of last semester. In the first semester, the size of the dataset was 10000, and the coordinates in the dataset are arranged by vertical, the lowest value of RRSE is 21.29% [4]. According to the previous work in the first semester, to reduce the value of RRSE in the second semester, I coded a function which can automatically test the different combinations of parameters. These parameters contain the number of the hidden layer, the number of nodes in each layer, the value of learning rate and momentum. I also set the search space

for these parameters: the amount of the hidden layer set as 3; the number of nodes in each layer configured from 1 to 20; the value of learning rate set from 0.0001 to 0.9, but it only contains the number likes 0.0001, 0.001, 0.1. Similar to the learning rate, the value of momentum was set from 0.01 to 0.9. For example, a combination of parameters can be that 3(hidden layers), 12,3,6(nodes on each layer), 0.05(learning rate) and 0.7(momentum). At last, I found that when the Multilayer Perceptron has three hidden layers, and the number of nodes in each layer is 19, 5, six respectively, the learning rate is 0.008, and the momentum is 0.6, the value of RRSE reduced to the 15.37%. Some studies, such as the study [18], had testified that increasing the scale of training data set can improve the accuracy of the model. So, to further reduce the RRSE, I used the new dataset which contains 1 million data with the same function and the parameters configuration, the RRSE decreased to the 13.87%. This number is the best value now, and compared to the multilayer perceptron in TensorFlow and random forest trees in Scikit Learning, the multilayer perceptron in Weka has the best performance in research work.

Also, I also used the deeplearning4j to train the model. Deeplearning4j is a Java-based toolkit for building, training and deploying Neural Networks, and it also works on the native GPU, as well as the distributed GPUs. To use the GPU, it needs to set the configuration, which shown as follow:

```
// GPU part.
DataTypeUtil.setDTypeForContext(DataBuffer.Type.HALF);

// Create the CUDA context instance and set the parameters
CudaEnvironment.getInstance().getConfiguration()

    // Whether to allow multiple GPU
    .allowMultiGPU(false)

    // Set the capacity of cached data in memory, in bytes:
    .setMaximumDeviceCache(2L * 1024L * 1024L * 1024L)

    // Whether to allow multiple GPU peer-to-peer (P2P) memory
      access
    .allowCrossDeviceAccess(false);
```

If we have multiple GPUs, the model can be trained under multiple GPU parallel training, it needs to change some settings, for example, in the above codes, the allowMultiGPU() and allowCrossDeviceAccess() need to be set as the true. Then add the logic of parallel training in logic:

```
// ParallelWrapper will take care of load balancing between GPUs.
ParallelWrapper wrapper = new ParallelWrapper.Builder(net)
    // DataSets prefetching options. Set this value with respect
      to number of actual devices
    .prefetchBuffer(24)

    // set the number of workers equal to some available devices.
      x1-x2 are good values to start with
    .workers(1)

    // rare averaging improves performance, but might reduce model
       accuracy
    .averagingFrequency(3)

    // if set to TRUE, on every averaging model score will be
      reported
    .reportScoreAfterAveraging(true)

    .build();
```

My target is to build a multilayer perceptron model with GPU based on the deeplearning4j. There were two ways to achieve the objective, and the first method is creating the model by myself, the second way is using the package provided by Weka. Firstly, I tried the first direction, but after one week's effort, the performance of the model is still bad. I think that's because I'm not very familiar with the deep learning, even if I take another week's time to research, the result may still bad. To improve the work efficiency, I used the package provided by Weka to train the model. This package needs the version of Weka is higher than 3.9.1, and the release of CUDA must be the 8.0. Based on this package, I build a model named DL4JmlpClassifier, which represents this model is multilayer perceptron based on the deeplearning4j with GPU, and then I also set an experiment to compare the performance between DL4JmlpClassifier and the multilayer perceptron based on CPU. The hardware for the experiment shown in table 1. In this experiment, the configuration is always kept same, which is that the neural network has three layers, the number of nodes in each layer is 19, 5 and 6 respectively. The learning rate set as 0.008, the momentum set as 0.6 and the batch size set as 100. The two models trained on the 11 datasets, which the size of them is from 5000 to 100000. The result is shown in Figure 5. Compared to CPU, the GPU computing has the faster speed. In my experiments, when the size of the dataset is smaller than 30000 the advantage of GPU computing on the speed was not significant, but after the scale of dataset beyond the 30000, the GPU computing had a better performance on the speed than CPU computing. Especially, when the size of the dataset is 100000, the GPU computing took almost 31 minutes to finish the task of training and testing; however, the CPU computing took nearly 46 minutes to complete the same work. In theory, GPU computing should also have the more accurate result than CPU computing, but the graphs in Figure 1 do not reflect that. The reason of it will be described in the evaluation part.

Table 1: The information of hardware.

GPU
NVIDIA GeForce GTX 770M/3G GDDR5
CPU
Intel Core i7-4700MQ Processor
Memory
DDRIII(L) 12GB

4 EVALUATION

This section also contains two parts, and one is for software part, another is for research part. In software part, I will describe two trade-off points in my contributions and the user experience of Spring Security from my side. In the research part, I will discuss the performance of multilayer perceptron based on deeplearning4j with GPU.

4.1 Software Part

In my contributions, there are two trade-off points, and one is the password encrypt algorithm. As described in Section 3.2.4, the BCryptPasswordEncoder was used as us encrypt algorithm. For this algorithm, the higher strength, the more secure, the highest strength of this algorithm is 31, the reason why I set the strength as ten rather than 31 is that the higher strength represents it needs

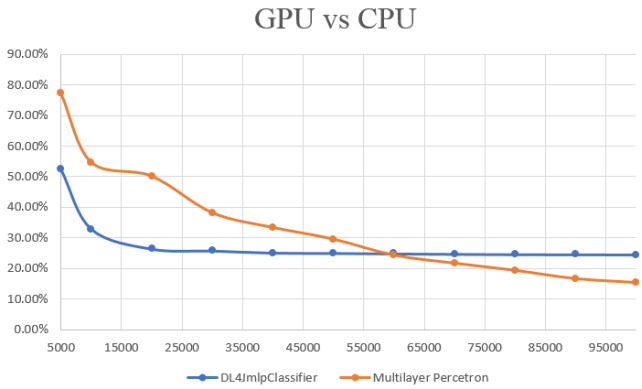


Figure 5: Comparison Between GPU and CPU

more time to calculate the hash, which means the response time of the system is longer. For example, after users entered the username, password and pressed the login button, the system needs more time to calculate the hash. Then the system can compare the hashed the password with that in the database, which means users need to wait few seconds to log in the system, this would reduce the performance of our website. To improve the performance and security, after I tried the different number of strength I choose the ten as the strength of this algorithm. Because the average wait time is almost 0.5s when the strength is ten and this time is suggested in the Spring Security document. This level of strength can make sure the passwords are secure, and at the same time, after users press the login button, they can log in the system immediately.

Another trade-off point is the remember me function, as described in section 3.1.3, remember me is a convenient function for users, which can let users log into the website without entering the username and password. The remember me function improves the usability of our website. However, this function also has the potential risk, that is the attacker still have the change to use cookies of victims to log into the system. The Http is a stateless protocol, if cookies and sessions are used to keep the state, this kind of risk is inescapable. So, the system needed the highest security level never use the remember me function, such as the bank web system. In bank system, no matter users close the web page or press the back button, they all need to log in again. In this way, the system is very secure, but the usability is terrible. Our website is not bank system, so we do not need the extremely high level of security, let users can efficiently use our website is the primary target. At the same time, I also use other functions in Spring Security to protect the cookies, such as XSS protection and set the HttpOnly attribute of the cookie as true. Both two measures can stop attackers use JavaScript to steal the cookies from clients' browsers [11].

In conclusion, our website improves performance and usability in the context of ensuring security. I also used other measures to protect the website, such as the Hibernate Validator was used to prevent the system from the SQL injection; users' passwords encrypted at the front end and back end, in this way, there is no plain text transmission on the Internet. Also, I customised the XSS protection mechanism and CSRF protection mechanism to against the XSS attack and CSRF attack. I believe our website is a secure system because I used the Spring Security framework. Compare to using PHP to create kinds of security mechanism I think the Spring Security is a good way to achieve the target. Because the security mechanisms provided by Spring Security come from the Spring Security team which contains 68 security experts [19] and it has 13 years history, the reasonability and security of these security mechanisms should higher than that which created by myself. Spring Security provides an available permissions framework and many manners of user identity authentication, which can save a lot of development work, such as username + password or email account + password. Also, Spring Security also provides the role judgment function, even if we didn't use this function but it is still very useful. What's more, Spring Security provides many security mechanisms, such as session management, XSS protection and CSRF protection etc. I think all these features should belong to the advantages of Spring Security. If I need to describe a disadvantage of Spring Security, it must be that Spring Security is not friendly to the developers who never used it before. In the framework of Spring Security, there are so many interfaces, and each interface has their implementations. Developers need to understand the dependence between these interfaces and implementations, then they can use them in own codes. However, understanding the dependence between many interfaces and implementations is not easy for the developers who never used it before, these developers need to take a lot of time to learn it. For example, as discussed in section 3.2.1, these contents took me almost two weeks to learn, understand and implement them, compare to using PHP to implement the same user authentication, the Spring Security cost much time. But I still think learning Spring Security is worth, because the more familiar you are with it, the easier it will be. For me, the Spring Security still needs more time to learn it, all the descriptions about Spring Security in this report are only from my side, it may have some inaccurate places.

4.2 Research Part

At the end of the first semester, the lowest value of RRSE was 21.29%, and the size of the dataset was 100000 [4]. In this semester, I fixed the algorithm that can automatically test the different combination of parameters, and then I got the new lowest value of RRSE, which was 15.37%. Kept the same configurations, I used the dataset that size is one million to training and testing, at last, I got the best model, the RRSE of this model is 13.87%. However, there are some limitations for the result. Firstly, the configuration of parameters. In my algorithm, according to the last semester research, I set the number of hidden layers as three and limit the number of nodes in each layer, which is smaller or equal to 20. These number limitations make the result as a local optimisation, which means the 13.87% may still not the global optimisation. Many studies had testified that the more layers and nodes, the result will be more accurate. However, the more layers mean the process of training will take more time. Another limitation is the size of the dataset, the size of the dataset is larger the more time is needed for training either. The size of the dataset interacts with the number of hidden layers and nodes to make the training time increased. For example, when I used the dataset which size is one million, and I set the neural

network has three layers, the number of nodes on each layer is 19, 5, 6 respectively, it took almost 3 hours to finish one time of training and testing. If I want to test 100 different combination of parameters it needs 300 hours to achieve that, in actual, the number of the various combinations of parameters is much bigger than 100, which means that too much time consumed to make it very hard to find the global optimisation.

For the DL4JmlpClassifier which is the multilayer perceptron based on deeplearning4j with GPU, the limitations are almost same. The difference is that the DL4JmlpClassifier is more complicated, for each layer in DL4JmlpClassifier there are 16 parameters that can be set. I set three hidden layers and one output layers, which means it has 16*16*16*16 different combinations, this made the DL4JmlpClassifier became harder to find the optimisation result. As this reason, based on the best configuration from the previous research I only set the number of hidden layers, the number of nodes in each layer, learning rate and momentum, other parameters used the default value. The result is shown in Figure 4, in theory, the performance of the DL4JmlpClassifier should better than the multilayer perceptron, the results of RRSE that before the size of the dataset is 60000 testifies the DL4JmlpClassifier have a better performance, especially, when the size of the dataset is small. The reason why the DL4JmlpClassifier performed worse than multilayer perceptron when the size of data is over the 60000 is that the configuration from the previous research did not work well on the DL4JmlpClassifier, but works well on the multilayer perceptron. I'm still not familiar with the deep learning, and it makes me cannot understand all meaning of the parameters, so I cannot set these parameters as the best value.

5 REFLECTION

In general, I think the MSE project A and B are the beneficial course that let me learn much knowledge from them. This knowledge not only contains the coding intellectual but also include much useful knowledge that needed in the real software industry.

Form the MSE project A, I learnt much knowledge about the machine learning algorithms. But I think the most useful knowledge is how to research a subject. I conclude some steps that how to research a subject as follow:

- Understand the tasks. I remembered the main job of the first week in the first semester is to understand the task of the project. We talked with supervisors and got the information about what are the requirements of this project. A good understanding of task can help us to prepare the work.
- Read related papers. All of team members are asked to read the related papers, through reading the papers, we had a deeper understanding of the project backgrounds. In addition, it needs to read other papers that related to the used technology of the project. From these papers, we can preliminary understand which technologies are suitable for this project.
- Preliminary research. According to the related papers, we can choose some technologies to execute the preliminary research. For example, at the early phase of our project, I preliminary research some machine learning algorithms and find the multilayer perceptron is suitable for our project.
- Deep research. After we choose the technology based on the last step, we need to have a deep research in the object with the chosen technology. Papers and online tutorials of related technology are always good tools to help the research.
- Record the information. This step is very important, during the long-time research, we may change the different parameters of algorithms and get the different results. We need to record all the result in document, it would help us to optimise the result.
- Evaluate the result. After the research, we should know the advantage and disadvantage of our research. It will help us to improve our research work.

For the MSE project B, I learnt much knowledge in Spring Security and deep learning. However, I think the most vital knowledge is about software engineering.

Firstly, I had learnt how to use the GitHub rightly. Before the project B, the only function that I can use in GitHub is code upload, this may sound strange, but it is true. As the most popular hosted platform for software projects, GitHub is used by many developers and projects, and it can say that rightly using the GitHub is an indispensable skill for software engineers. Fortunately, the project B has taught me how to use GitHub in the software industry. Such as how to code review, how to use issues and how to merge pull request etc.

Secondly, the project B has taught me how to manage my work. The burndown chart can show whether your team has a proper management of tasks. From our team's experience, it is not easy for making a good control of work, after meeting with Marian, I understand that if I want to have a proper management of work, I need to have a macro plan for the current sprint before this sprint begins and split the macro plan into detail tasks. How to measure whether the work is enough detail, I think Marian had given us a reasonable explanation. He said when you estimate the time of a task if the estimated time is too long, such as 15 hours or 20 hours, this means the work does not detail enough because we are impossible to finish 15 hours job or 20 hours job in one day. The reason why the first five sprints' burndown charts of our team looks not good is because we did not have a proper job management. We found the problem in time and adjusted our job management, and then we got a good result on burndown chars for the last three sprints.

The last, from the project B, I learnt the importance of team cooperation. Team cooperation contains many aspects, but I think the most significant is that when the team meets the problem the team members need have a good communication on this problem to help the team to solve this problem. For example, my team want to use the Spring Boot for the back-end of our website, however, I never use the Spring Boot before, so my concern is that I cannot make enough contributions for the team. I told my concern to my team members, after the communication they said if I meet the problems that I cannot solve, they will help me to solve them. And as our project need focus on both the software engineering side and research side at the same time, my team member also arranged me to research the deep learning neural network. In this way, our team can ensure the progress of the project and I have enough

contributions for the team. In addition, I think the burndown chart is also a good tool to measure whether the team has a good cooperation. The burndown chart records all the actions of team members, which means if there is only one member do not follow the work management, the whole team will not have a good performance on the burndown chart. Fortunately, I have great team members, Mengyu and Yuanzhong, although as we did not have a good work management at the beginning of this semester, which led to us did not have good performance on the burndown chart, after we adjust the strategy of work management we have a good performance on the burndown chart.

If I do this project next time, I think there will have big difference. Firstly, I would have more progresses in machine learning part. The lowest value of RRSE in the first semester is 21.29%, and that in this semester is 13.87%, I believe the next time, I can get the lower value of RRSE than before. What's more, I think the next I can more familiar with the deep learning neural network with GPU, and use it to reduce the value of RRSE and the training time. Secondly, I would do better in the back-end of our website. The next time, as I have already more familiar with the Spring Boot and Spring Security, I do not need to take too much to learn some basic knowledge, then I have more time to the more advanced knowledge and do more works on the back-end. The last, our team will have a high-level performance on the GitHub. I would manage the GitHub from the beginning of the project in next time. The management contains that monitoring whether team members standardly use the GitHub; Checking whether team members follow the work management. The purpose of the management is to let our team more professionally use the GitHub.

6 CONCLUSION

In this semester, our project still can be divided into two parts. For the software part, as the back-end of our website changed to Spring Boot, I choose the Spring Security to implement the security tactics for our website. At the same time, considering the usability and performance of our website, I adjusted the security tactics to make sure our website is security enough while having the good performance and usability. For the research part, the target of this semester is continuing to reduce the value of RRSE. I coded a function which can automatically test the different combinations of parameters. Then I found that when the Multilayer Perceptron has three hidden layers, and the number of nodes in each layer is 19, 5, 6 respectively, the learning rate is 0.008 and the momentum is 0.6, the value of RRSE reduced to the 15.37%. After this experiment, I used the new dataset which contains 1 million data with the same function and the parameters configuration to training a model and the RRSE of this model reduced to the 13.87%. This model is the best model in our project, and for the Open-day project, this model was used to predict the power according to the coordinates from the users. At the same time, I had a preliminary study on deep learning with GPU. I used the package provide by Weka based on the deeplearning4j to train a model when the size of the dataset is small this model has a good performance than multilayer perceptron with CPU, but as the size of the dataset is increasing the multilayer perceptron with CPU has the better performance. I think this is because I still not find the suitable parameters for deep learning neural network.

In conclusion, no matter project A or B, I had learnt many thing from these two courses. I'm sure these knowledge will help me a lot in my future study and work.

REFERENCES

[1] Dídac Rodríguez Arbonès, Boyin Ding, Nataliia Y Sergiienko, and Markus Wagner. 2016. Fast and effective multi-objective optimisation of submerged wave energy converters. In *International Conference on Parallel Problem Solving from Nature*. Springer, 675–685.

[2] Spring Security architecture guide. https://spring.io/guides/topicals/spring-security-architecture.

[3] Australia Climate Change Authority. Renewable energy target review report. December 2014.

[4] chenwei Feng. MSE 2017 Project Individual Report for semester 1.

[5] The climate change authority's special review on Australia's climate goals and policies:. towards a climate policy toolkit. 5 SEPTEMBER 2016.

[6] Boyin Ding, Leandro Souza Pinheiro da Silva, Nataliia Sergiienko, Fantai Meng, Jonathan David Piper, Luke Bennetts, Markus Wagner, Benjamin Cazzolato, and Maziar Arjomandi. 2017. Study of fully submerged point absorber wave energy converter-modelling, simulation and scaled experiment. (2017).

[7] Spring Security Document. https://docs.spring.io/spring-security/site/docs/current/reference/html/.

[8] Nicolas Vayatis Frederic Dias Dripta Sarkar, Emile Contal. Prediction and optimization of wave energy converter arrays using a machine learning approach.

[9] JeanRaymond Bidlot Gordon Reikard, Bryson Robertson. Wave energy worldwide: Simulating wave farms, forecasting, and calculating reserves.

[10] Yang X. Shasha W. Guoyang, W. Discussion about short-term forecast of wind speed on wind farm. In *Jilin Electric Power 181 (2005) 2124.*

[11] The introduction of HttpOnly. https://www.owasp.org/index.php/HttpOnly.

[12] P.A. Gutiérrez E. Alexandre C. Hervás-Martínez J.C. Fernández, S. Salcedo-Sanz. Significant wave height and energy flux range forecast with machine learning classifiers.

[13] Nataliia Y. Sergiienkoy Benjamin S. Cazzolatoy Boyin Ding Frank Neumann Markus Wagner Junhua Wu, Slava Shekh. 2016. Fast and Effective Optimisation of Arrays of Submerged Wave Energy Converters. In *Published on Proceedings of the Genetic and Evolutionary Computation Conference.*

[14] Panteliou S. Dentsoras A. Kalogirou, S.A. Artificial neural networks used for the performance prediction of a thermosiphon solar water heater. In *Renewable Energy 18 (1999) 8799.*

[15] S. Li. Wind power prediction using recurrent multilayer perceptron neural networks. In *Volume 4. (2003) 23252330.* In: Proceedings of the 2003 IEEE Power Engineering Society General Meeting.

[16] Tom Durrantb Julian O'Gradyc Ron K. Hoekec Kathleen L. McInnesc Uwe Rosebrocka Mark A. Hemera, Stefan Ziegerb. A revised assessment of Australia's national wave energy resource. Published in Renewable Energy.

[17] Department of Industry. Innovation and Science (2016) Energy in Australia. Canberra,January 2015.

[18] A. Sfetsos. A comparison of various forecasting techniques applied to mean hourly wind speed time series. In *Renewable Energy 21 (2000) 2335.*

[19] Spring Security team. https://spring.io/team.

[20] Yuanzhong Xia. The introduction of API,https://github.com/MewX/mse2017-buoy/wiki/API.