MSE 2017 Project Individual Report

Optimization of Wave Energy Converters

Mengyu Li
The University of Adelaide
Adelaide, SA 5005
a1680972@student.adelaide.edu.au

ABSTRACT

Renewable forms of energy have become a hot area of research in recent years, such as wind energy and solar energy. The traditional energy, like petroleum, can cost environmental pollution easily, so the potential value of the renewable energy is obvious. There are many benefits of renewable energy, such as non-pollution, renewable and economical. Therefore, the research on renewable energy using computing techniques [1][2][3] are becoming increasingly popular recently. This area has drawn many researchers' attention, including the optimisation research team at the University of Adelaide.

In our project this semester, we mainly focus on wave energy. Our team do research on the submerged wave energy converters [4] from scratch. We simplify a wave energy converter as a buoy, which will be put below the surface of the ocean and captures energy from waves. However, only a single buoy cannot obtain enough energy, so we need to consider how to deploy multi-buoys in the ocean. The objective is to maximise the overall energy absorption. To achieve this goal, there are many factors need to be considered, such as the distance between of each buoy, the submerged-depth of each buoy. In our project this semester, we mainly focus on the layout of four buoys since the layout will influence the absorbed energy significantly [4].

KEYWORDS

Machine Learning, renewable Energy, Optimisation

1 INTRODUCTION

The layout of multi-buoys can be various and our work is to find out the best one that can make buoys capture the largest energy. As mentioned in [5], the positions of buoys are essential because the positions will significantly influence the overall energy. However, there are many tough problems need to be solved when we try to optimise the layout of buoys. The first problem we meet is that the given model, which is operated in the Matlab, requires too long time to calculate the result. We use i7-2720QM (single thread) to run the original model and it took approximately 46 seconds to calculate a single result (energy) for four buoys. This speed will cost too long time if we wish to optimise the layout, since we will use evolutionary algorithms to do optimisation in the next semester and such algorithm needs to do large calculations. Therefore, we have to solve the time-consuming problem before we do optimisation. To solve this issue, our tutor suggested us to train a new model using machine learning and use such new model to replace the original model.

Finding a new machine learning model becomes the major task for our project in this semester. We kept finding a good model that can replace the original model. The output (energy) of the new model should be as accurate as possible. In other words, the error rate of the machine learning should be small, otherwise the optimisation would not be accurate as well. In this semester, we only focus on the model for four buoys. To find an ideal machine learning model, we did research on the following two problems this semester: 1.what features we should choose as the input of our training model. 2.how to reduce the error rate of our machine learning model.

Apart from the machine learning models, we also built a simple web app, which can display our research results and allow others to use our models. Users can upload their own data and select the machine learning method. Then the web app will provide them with a model that can fit their training data. Users can download this model or just use it online.

Contribution. This essay will describe my personal contributions in finding machine learning models and building the web app. On the research side, I have done the following works: 1. trained three machine learning models: Non-linear fit, Boosted Regression Tree [6][7][8] and Multi-Poly Regress [9] 2. conducted an experiment to demonstrate the features that we should use as input data. 3. tried to optimise the original model. In web application side, I am responsible for: building the UI of our web app.

Outline. The essay is organised as follows. In section 2, I will introduce the three machine learning models. The advantages and disadvantages of each method will be provided. In section 3, I briefly introduce the experiment that I conducted to demonstrate the features that we should select. Section 4 will briefly introduce my effort in optimising the original model. In section 5, the UI of the web application will be provided. Finally, I will make a conclusion and describe my future work in next semester in section 6.

2 MACHINE LEARNING MODELS

The purpose of finding out new models is to replace the original model and cutting the running time. In this semester, I only consider the model for four buoys and only use the positional information of these four buoys as input data. More specifically, I ignored all other factors, such as the submerged depth, the wave speed and the wave angles. I narrowed down the research to the coordinates of the four buoys, that means my model can only predict the absorbed energy for four buoys always under the same circumstance.

Tool. The major tool I used is Matlab, which is a very famous and useful tool for machine learning and Mathematic computation.

Method. Three methods: nonlinear regression, boosted regression tree and multi-poly regression.

Benchmark. To make a comparison between each machine learning mode, we use Root Relative Squared Error (RRSE) [10] as a benchmark. Different group members may train different models, but we use the same benchmark to evaluate the models. The formula of RRSE is shown as below:

$$RRSE = \sqrt{\frac{\sum_{i=1}^{N}(\widehat{\theta_i} - \theta_i)^2}{\sum_{i=1}^{N}(\bar{\theta} - \theta_i)^2}} \quad (1)$$

θ means the real energy, which is generated by the original model. $\widehat{\theta}$ means the predicted energy that is generated by our new model, while $\bar{\theta}$ means the average value of θ. N means the size of data set. We use this RRSE to evaluate the accuracy rate of our new models. The less the RRSE, the more accurate our new models. Our clients did not specify a target RRSE this semester.

2.1 Non-linear regression

Nonlinear models have become essential machine learning models, which have attracted much attention in recent years. As described in [11] [12] [13], the nonlinear model can help researchers find the nonlinear relationships between the input data and output data. This method may not fully suitable for our project, but I still tried it and present the relevant efforts here. At the beginning of our project, I just use a built-in method in Matlab to train the nonlinear model. Firstly, I define a simple equation as follows:

$$E = b_0 + b_1x_1 + b_2y_1 + b_3x_2 + b_4y_2 + b_5x_3 + b_6y_3 + b_7x_4 + b_8y_4 \quad (2)$$

E means the energy, which is the output of the original model. x_n and y_n represent the coordinate of the n-th buoy. b_n is the coefficient that we need to figure out using Matlab. The results are as follows:

b_0 = 1.541414699230119 * 1.0e+06
b_1 = -0.000010909444929 * 1.0e+06
b_2 = -0.001166222084215 * 1.0e+06
b_3 = 0.000011206081888 * 1.0e+06
b_4 = -0.000257229960103 * 1.0e+06
b_5 = 0.000008148138227 * 1.0e+06
b_6 = 0.000264111518500 * 1.0e+06
b_7 = -0.000015969619420 * 1.0e+06
b_8 = 0.001164598580767 * 1.0e+06

This is an initial model function. After defining this model function, I use the Matlab to calculate the most appropriate value of Bn. The size of the whole data set is 100000. I set 70% of them as training data and 30% of them as testing data. The RRSE of this nonlinear model is around 80%. The error rate is high, because this model function is too simple to fit the training data. However, such model is a good beginning of the whole project. In this semester, I also tried many other model functions which are more complex and achieve better RRSE.

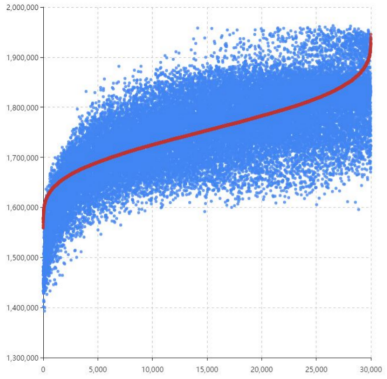

Figure 1: The plot for nonlinear fit

In this semester, the size of training data that used by nonlinear regression is much larger than that used by other two methods, so you will see that the points in the plot for nonlinear regression are denser than that for other two methods. The Figure 1 shows the plot for the simple nonlinear-fit model. The y-axis represents the energy. The x-axis represents each layout. Red points represent predicted energies which are generated by the non-linear model. These values have been sorted in ascending order. The blue points represent real energy values which are calculated by the original model. The distance between the red point and blue point can reflect the fitting degree.

Advantage The new model of this method is portable. This method can provide researchers with a white box, since it can be described as a nonlinear equation. Researchers can use such equation in any other circumstance and run it using any other programming language. It is convenient for researchers for who wish to do further research works. My group members tried some other methods, such as multi-layer perceptron. These methods are hard to get out of their original tool or programming language.

Disadvantage Finding a suitable model function is the main tough point in this method. The equation described above is the simplest model function and obviously, it cannot fit the training data very well. The Matlab can help us figure out the coefficient Bn, but cannot provide a suitable function model. Some other function models, such as high-power equation, are hard to design. In some case, such model functions depend on the experience of developers.

2.2 Boosted Regression Tree

Boosted Regression Tree (BRT) [6] is the second method that I have tried using Matlab. This method combines binary regression trees [8] and adopts a gradient boost technique. In this project, I used this model as a black box. There are many parameters need be modified in this method, such as the number of trees and leaves. Tuning such parameters is a problem for researchers. In my case, I just use a simple nested loop to try different values of parameters and pick up the best one by comparing their RRSE.

TUNE PARAMETERS OF BOOSTED REGRESSION TREE

```
1  bestRRSE = infinity
2  bestTREES = 0
3  bestLEAVES = 0
4  for num_of_trees ← 10 to 100
5       do for num_of_leaves ← 10 to 100
6            training()
7            RRSE ← testing()
8            if RRSE < bestRRSE
9              then
10                  RRSE ← bestRRSE
11                  bestTREES ← num_of_trees
12                  bestLEAVES ← num_of_leaves
```

The data set is same as that I used in the nonlinear method. I tried different data size for training and the RRSE is changed significantly. I always use 70% of the whole data as training data and 30% as testing data. The RRSE of this model is around 50%-60%, which is much better than the nonlinear model.

Figure 2 shows the predicted values and real values. Compared with the nonlinear model, the blue points are much closer to red points, which means this model is can fit the data better than the nonlinear model.

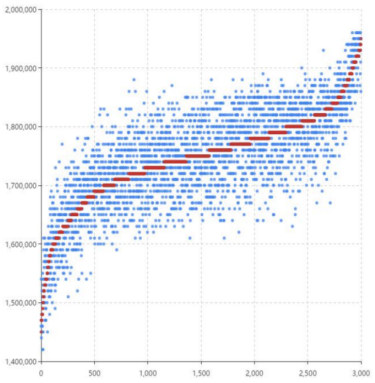

Figure 2: The plot for boosted regression tree

Table 1 below shows the different RRSE of boosted regression tree when I use different size of training data. The training data are rounded generated by the original model.

Advantage BRT have essential advantages of tree-based methods [14]. With each tree added, the new tree can correct the previous trees' errors. Such method can fit complex nonlinear relationships among each data set. It is a classic method in machine learning, so I give it a try although it does not fully suitable for our project.

Disadvantage There are three basic parameters in BRT, which are trees, leaves and learning rate. Each of these parameters needs to be tuned by programmers to achieve a good fit. A high number of trees or leaves often lead to overfitting [14], which means the model can fit the training data very well, but cannot fit the testing data. The values of these parameters depend on the researchers' personal experience. For me, I just keep trying different values and compare the value of RRSE. Another disadvantage of this method is the time consumption of the predicting phase. When I set the 100 trees and 50 leaves and run the Matlab with 7000 training data, the Matlab will cost 8 minutes to generate the model and 4 minutes to do prediction using 3000 testing data.

2.3 Multi-Poly Regression

Multi-poly regression is a basic but effective method to fit multi-dimensional data using a high degree polynomial. This theory of this method is similar to the first method, that finds an appropriate model function to fit the data. However, researchers do not need to set a fixed model function in advance, such as $y=b_0+b_1x_1+b_2x_2$. The approach can generate a high degree polynomial as model function automatically. The only parameter that needs to be modified is the polynomial order. The best RRSE that achieved by this method is around 31% with 10000 training data and a 6-order polynomial.

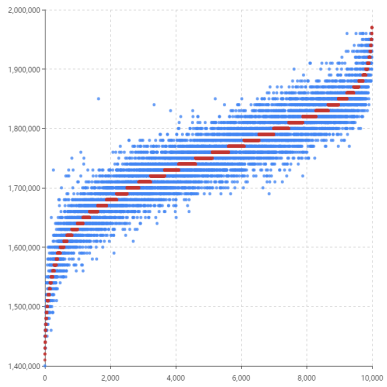

Figure 3: The plot for multi-poly regression

Advantage According to the basic Mathematical knowledge, the high-order equation can calculate the distances between two points naturally. [4] shows that the distance among each two buoys is an essential factor that will influence the absorbed energy. Therefore, the multi-poly regression can take this key factor into consideration and fit the training data better than other methods.

The multi-poly regression method in Matlab does not need any more tool box. The result is portable, because the model can be described using a single high-order polynomial. Researchers can use such equal in any other environment. In addition, there is only one parameter that needs to be tuned. It is convenient for the researchers with less experience on machine learning. In my project, I have tried from one to seven degrees and find six degrees is the best. If the polynomial has more than six degrees, the model will over fit. The RRSE for five degrees, six degrees and seven degrees are shown as below.

Disadvantage According to the original model code, the angle between each two points is another essential factor that may influence the captured energy. However, it is difficult for a multi-poly model to calculate the angle, such as sin or cos. Therefore, such method can only get the RRSE around 30%.

Table 1: RRSE for boosted regression tree in different size of training data

data size	1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
RRSE	66.35%	69.26%	62.15%	62.72%	60.13%	63.16%	62.53%	58.20%	58.20%	56.31%

Table 2: RRSE for multi-poly regression using different order-equations

	4	5	6	7
RRSE	48%	41%	30%	38%

Researchers do not need to figure out the model functions by themselves. However, it brings another limitation that users cannot make contributions on modifying the model functions. If researchers wish to do some further optimisation on the given equation, they can only modify the equation after the model generated. I suggested that we can use evolutionary computation to modify it, but have no enough to do it in this semester.

2.4 Summary for three methods

I have tried three machine learning methods: non-linear regression, boosted regression tree and multi-poly regression. Not all these methods are suitable for this project, but they provided many useful ideals for me and for my future work. For example, the first method nonlinear method (as shown in the graph below) shows me the potential of a good model function. To find out a better model function, I use the multi-poly regression and successfully obtain a six-order equation. Then I use this equation as model function and achieved a much better result.

In figure 4, the coordinates of each red point is (predicted energy, real energy). The predicted energy is calculated by my machine learning model, while the real energy is calculated by original mode.If these two value are the same, the red point should located on the 45% black line. In this figure, we can see that the red points are more close to 45% black line in the third one, which means that multi-poly regression can fit data better.

Table 3 below shows their best RRSE separately. The RRSE of Nonlinear Regression and Boosted Regression Tree are much higher than Multi-Poly Regression, but it does not mean that the Multi-Poly Regression is better than other two methods. It only represents that multi-poly regression can fit the training data in our project better, but the result can be different in other situations.

3 FEATURES SELECTION

To train an idea model, I considered about extracting features from the eight coordinates. Selecting appropriate features is important for training model [15]. At the beginning, I only use the coordinates of four buoys as training data, but the error rate is high. Therefore, I thought several methods to optimise the raw data and conducted an experiment to demonstrate my idea. Our group members may have different ideas on the selection of the features. The next two subsections will provide my efforts on this part.

3.1 Swap buoys

I considered three methods to swap the four buoys: 1. Horizontal: sort four buoys by their distance between y-axis. 2. Vertical: sort four buoys by their distance between x-axis. 3. Basepoint: sort them by their distance between the point (0, 0). I thought about this idea because I notice that in the first method nonlinear-fit method, the Bn in front of each coordinate is different. That means different buoys should have different weights. The random sequence may lead to an average weight of each buoy, which is absurd. From the physical point of view, each buoy should capture different power because of occlusion by each other.

The table below shows the RRSE of boosted regression tree under different swapping method:

We can see from the table that the vertical-swapping method obtains the best RRSE. The gap between each swapping method is not obvious in BRT, but it can demonstrate the necessity of such swapping buoys. This swapping method achieves a remarkable success in multi-layer perceptron, which is a modern machine learning method adopted by my group member. When using the vertical-swapping method, the RRSE of multi-layer perceptron drop from approximate 50% to 20%.

3.2 Swap buoys

In this semester, I conducted an experiment to demonstrate that the overall energy will not be changed if four buoys translate to the same distance. The translation of layouts will not influence the result in the original model. As is shown in the picture below, the overall energy captured by layout A, B, C and D are same.

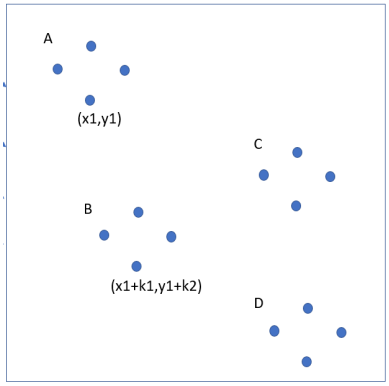

Figure 5: layouts after translating

The steps of my experiment is provided as pseudocode below:

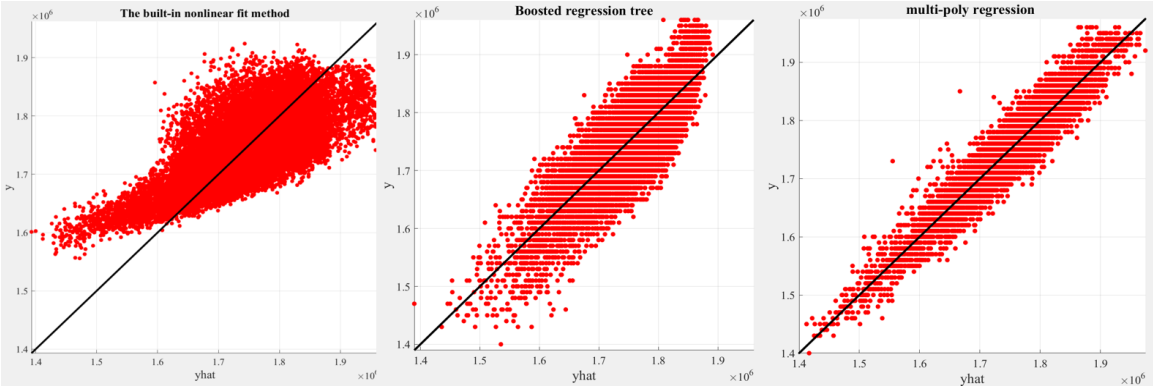

Figure 4: The plot for nonlinear regression, boosted regression tree and multi-poly regression method

Table 3: RRSE for three different methods

	Nonlinear Regression	Boosted Regression Tree	Multi-Poly Regression
RRSE	78%	61%	30%

Table 4: RRSE for three different methods

	horizontal	vertical	basepoint
RRSE	58.3%	56.8%	62.1%

The steps of my experiment

1 Generate 5000 training dataset randomly: $Dataset1$
2 $Dataset2 \leftarrow Dataset1$
3 **for** $dataset \leftarrow Dataset2$
4 **do** Generate random number k_1 and k_2;
5 X_2 plus/minus random number K_1
6 Y_2 plus/minus random number K_2
7 X_3 plus/minus random number K_1
8 Y_3 plus/minus random number K_2
9 X_4 plus/minus random number K_1
10 Y_4 plus/minus random number K_2
11
12 CalculateEnergy ($Dataset_1$): $Energy_1$
13 CalculateEnergy ($Dataset_2$): $Energy_2$

In step3, each layout is translated to a random distance and a new dataset is generated. Then I calculate the energy of the new dataset (Energy2) and the original dataset (Energy1) and make a comparison between them. The result shows that the dataset of Energy1 and Energy2 are the same. Therefore, we can demonstrate that the translation will not influence the overall absorbed energy. The table below shows my experimental data. The line A-H represent the coordinates of four buoys. The line J-Q represent the coordinates of buoys after translation. The line I and line R represent the original energy and new energy

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	174	61	153	162	260	118	92	95	1.61E+06	236	282	215	383	322	339	154	316	1.61E+06
2	155	112	242	139	38	265	155	166	1.75E+06	283	272	370	299	166	425	283	326	1.75E+06
3	171	84	39	221	257	10	198	135	1.91E+06	367	127	235	264	453	53	394	178	1.91E+06
4	194	103	127	188	190	227	115	41	1.80E+06	187	87	120	172	183	211	108	25	1.80E+06
5	206	183	71	186	271	183	183	116	1.63E+06	335	265	200	268	400	265	312	198	1.63E+06
6	90	162	112	70	63	6	54	266	1.92E+06	85	159	107	67	58	3	49	263	1.92E+06
7	261	212	2	5	189	224	21	121	1.77E+06	376	236	117	29	304	248	136	145	1.77E+06
8	170	227	249	143	104	188	58	77	1.68E+06	331	360	410	276	265	321	219	210	1.68E+06
9	263	162	44	88	273	262	203	215	1.76E+06	245	161	26	87	255	261	185	214	1.76E+06
10	16	132	116	235	83	98	175	151	1.70E+06	15	128	115	231	82	94	174	147	1.70E+06
11	199	113	4	105	170	242	121	75	1.68E+06	196	110	1	102	167	239	118	72	1.68E+06
12	136	231	100	118	157	54	21	179	1.80E+06	120	217	84	104	141	40	5	165	1.80E+06
13	148	166	143	2	82	208	79	84	1.85E+06	148	165	143	1	82	207	79	83	1.85E+06
14	232	185	12	73	74	64	163	127	1.67E+06	222	176	2	64	64	55	153	118	1.67E+06
15	225	9	163	0	227	177	35	141	1.74E+06	269	85	207	76	271	253	79	217	1.74E+06
16	112	215	142	52	242	187	8	81	1.72E+06	112	209	142	46	242	181	8	75	1.72E+06
17	125	206	91	60	201	105	39	243	1.74E+06	213	275	179	129	289	174	127	312	1.74E+06
18	124	50	106	243	72	200	167	14	1.77E+06	120	40	102	233	68	190	163	4	1.77E+06
19	79	140	209	179	134	104	127	241	1.69E+06	184	263	314	302	239	227	232	364	1.69E+06

Figure 6: The data for layouts after translating

Such finding can help us reduce the dimension of the training data. At the beginning, we have nine-dimension training data: the eight coordinates and one energy. After conducting an experiment on the original model, I suggested that we can reduce one buoy in our training data, which means we only need seven-dimension training data. More specifically, the original data set is $x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4$, Energy. Then I simplify the data set as $0, 0, x_2 - x_1, y_2 - y_1, x_3 - x_1, y_3 - y_1, x_4 - x_1, y_4 - y_1$, Energy. This decrease will not influence the result of the original model and I have demonstrated it in the experiment above. The objective in this semester is to find a new model to replace the original one, so I did research on the characters of the original model. After reducing the dimensions from nine to seven, the accuracy rate of a machine-learning model can be increased significantly [15], because of the more dimensions, the more complex inside the model.

Another experiment on the given model is to rotate the layout. The result shows that the energy will be changed after rotating. The experimental data is shown as figure 7. The line A - line H represent the original layout and the line J - line Q show the layouts

after rotating by a random degree. The line I and line R represent the energy of the original model and new model respectively.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	174	61	92	95	260	118	153	162	1.61E+06	114.28	144.7	26.819	129.5	156.38	238.89	42.453	218.75	1.75E+06
2	155	112	242	139	155	166	38	265	1.75E+06	140.92	129.27	224.18	166.21	134.63	182.91	6.9108	267.62	1.72E+06
3	257	10	171	84	198	135	39	221	1.91E+06	257.02	9.5263	171.15	83.685	198.25	134.63	39.407	220.93	1.91E+06
4	115	41	194	103	127	188	190	227	1.80E+06	73.806	97.255	105.99	192.38	3.3938	226.85	34.775	293.97	1.85E+06
5	183	116	271	183	206	183	71	186	1.63E+06	196.95	90.312	293.17	144.86	228.76	153.61	95.39	174.75	1.63E+06
6	63	6	112	70	90	162	54	266	1.92E+06	61.873	13.294	103.09	82.564	70.526	171.38	22.662	270.48	1.94E+06
7	2	5	21	121	261	212	189	224	1.77E+06	1.5124	5.1684	9.3259	122.45	239.52	236	166.7	241.06	1.80E+06
8	58	77	249	143	104	188	170	227	1.68E+06	79.059	55.16	281.11	58.529	157.28	146.37	232.13	162.92	1.75E+06
9	44	88	263	162	203	215	273	262	1.76E+06	74.293	64.502	304.95	49.182	269.76	121.09	352.41	137.77	1.66E+06
10	83	98	16	132	175	151	116	235	1.70E+06	75.799	103.67	6.5447	132.8	163.78	163.1	98.944	242.67	1.70E+06
11	121	75	4	105	199	113	170	242	1.68E+06	120.82	75.295	3.744	105.01	198.72	113.48	169.41	242.41	1.68E+06
12	157	54	100	118	21	179	136	231	1.80E+06	159.58	45.802	105.97	112.67	30.236	177.67	147.77	223.65	1.82E+06
13	143	2	79	84	148	166	82	208	1.85E+06	143.01	1.5359	79.272	83.743	148.54	165.52	82.675	207.73	1.85E+06
14	74	64	12	73	163	127	232	185	1.67E+06	81.348	54.355	20.938	70.955	177.46	105.86	253.1	154.88	1.54E+06
15	163	0	225	9	35	141	227	177	1.74E+06	160.57	28.027	220.1	47.553	10.235	144.92	193.19	213.39	1.70E+06
16	142	52	8	81	242	187	112	215	1.72E+06	140.81	55.133	6.2035	81.157	237.8	192.32	107.21	217.43	1.73E+06
17	91	60	201	105	135	206	39	243	1.74E+06	94.401	54.493	206.88	93.886	137.01	198.22	53.353	240.26	1.74E+06

Figure 7: The data for layouts after rotating and translating

4 MY EFFORTS ON OPTIMISING THE ORIGINAL MODEL

This section will describe my failure effort on optimising the original model. The major barrier of the original model is the long time-consumption. To solve this problem, I find out a loop that cost the most running time. This loop contains many physical and mathematical knowledge, such as calculus. From the programming point of view, I just treat this loop as a black box. The number of loops is 50 and my basic idea is to use the first 20 result to predict the final 50th result, thus speeding up the original model. The original model needs to loop for 50 times to get the result, if I can jump from 20th result to 50th result, the time-consuming problem will be solved. To achieve this goal, I recorded the result of each loop and use them as training data. The methods that I have tried are as follows: Polynomial, Fourier [15], Gaussian and Sum of Sine. They are all classic methods that provide by Matlab tool.

Polynomial To fit the 50-training data, the degree of polynomial must be set as eight. Such high degree leads to serious over-fitting. The model function is generated by Matlab as follows:

$$f(x) = p_1*x^8+p_2*x^7+p_3*x^6+p_4*x^5+p_5*x^4+p_6*x^3+p_7*x^2+p_8*x+p_9 \quad (3)$$

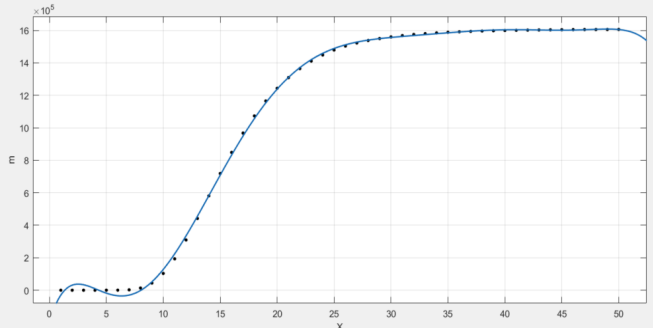

Figure 8: The training data and fitting curve of polynomial

The fitting condition can be shown by the figure 8. There are 50 points in this graph, which represent the result of each loop.

Fourier Fourier is a famous machine learning method, but it cannot fully fit our training data. The model function is generated by Matlab as follows:

$f(x) = a_0 + a_1cos(xw) + b_1sin(xw) + a_2cos(2xw) + b_2sin(2xw) + a_3cos(3xw) + b_3sin(3xw) + a_4cos(4xw) + b_4sin(4xw) + a_5cos(5xw) + b_5sin(5xw) + a_6cos(6xw) + b_6sin(6xw)$

Finally, all these efforts cannot work. There are several reasons: 1. The model function can only be used for a single dataset and it is not easy to find the model function for each dataset. 2.I have tried many model functions but the results were not satisfactory.

5 THE UI OF WEB APP

I am also responsible for building the front-end of our web app. The programming languages I used are html5+css+javascript. I have done the following four pages: home page, timeline page, models page and login page. Home page can show our project and our group members. Timeline page can show our efforts in this semester in the form of a time line. Models page allows users to select a machine learning model and upload their training data and testing data. After the back-end generate the model, users can drag buoys on the web page and the result will be shown. Login page is just used for register and login.

I attach some of the screen shots here. The code can be viewed at our group's github. This web page use the boostrap as framework. I add a html5-tag < *video* > to make our web more attractive.

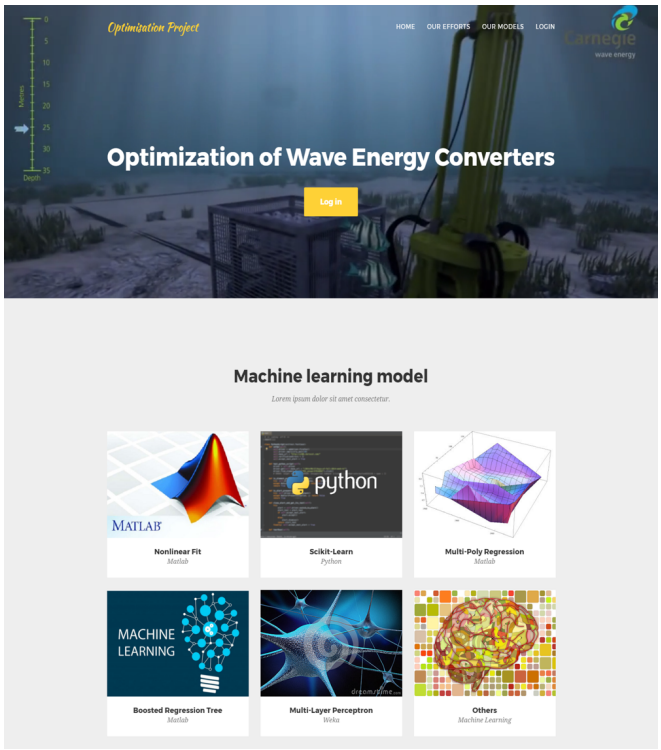

Figure 9: The home page of our web app

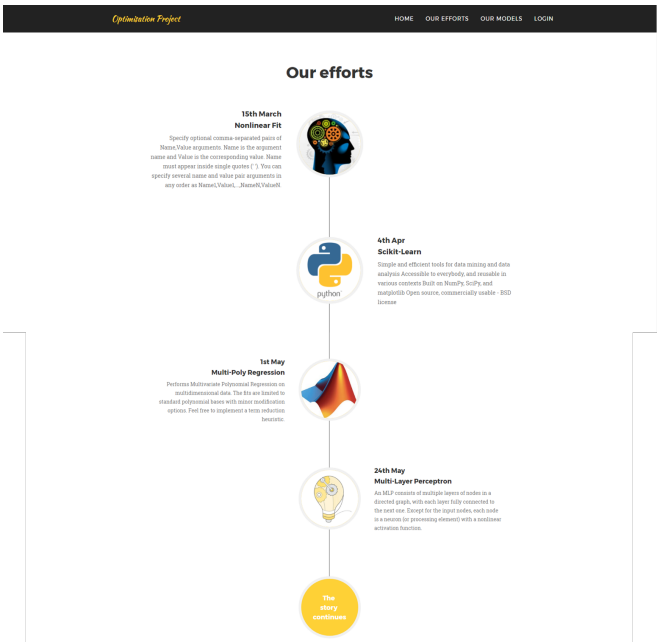

Figure 10: The timeline

Figure 11: The upload web page

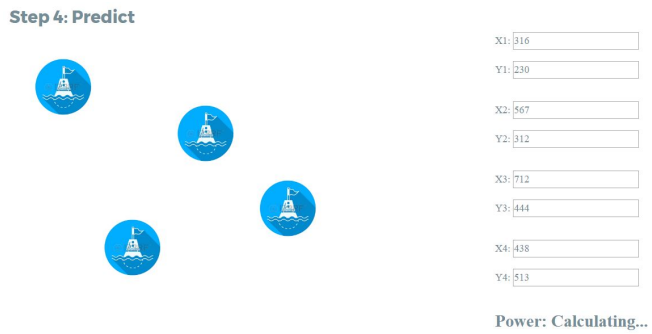

Figure 12: Drag buoys

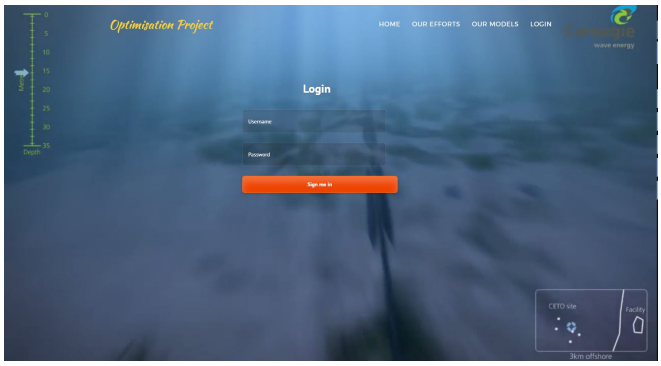

Figure 13: Drag buoys

6 CONCLUSION AND FUTURE WORK

In this report, I introduce the objective of our project in this semester and present my contributions on research side and software side. For research side, I have 1. adopted three different machine learning models to fit our training data, 2. conducted an experiment to demonstrate that we can reduce the dimensions of training data and lastly 3. tried to optimise the original model. On the software side, I showed my contributions on building the UI. My personal future work in next semester can be divided into two sides: the research side and the software side. For research side, I will 1. spend around four weeks to optimise the layout of multi-buoys to maximise the overall absorbed energy. 2. spend three weeks to optimise other factors that may influence the economic benefit, such as the cable length which is used to connect buoys with shore-side devices. 3. Do more research on optimisation methods and machine learning methods. For software side, there are two specific problems that need to be solved: 1. The calculation in back-end may cost too much time and the front-end have to wait for it, how to keep a long-time connection is a problem for developers. 2. The web application should not only work for our research project. It should be able to provide more services on model training and optimisation. For example, if users have some other data and wish to train a model or obtain an optimised value, the web app should be able to do this work.

REFERENCES

[1] Lin, Longbi, Ness B. Shroff, and R. Srikant. "Asymptotically optimal energy-aware routing for multihop wireless networks with renewable energy sources." IEEE/ACM Transactions on networking 15.5 (2007): 1021-1034.

[2] Ghandakly, A. A., Rodrigues, R. (2010, July). A GUI simulation system for integrating photovoltaic and wind units into power grids. In Proceedings of the 2010 Conference on Grand Challenges in Modeling Simulation (pp. 406-413). Society for Modeling Simulation International.

[3] Goiri, Í., Katsak, W., Le, K., Nguyen, T. D., Bianchini, R. (2013, March). Parasol and greenswitch: Managing datacenters powered by renewable energy. In ACM SIGARCH Computer Architecture News (Vol. 41, No. 1, pp. 51-64). ACM.

[4] Wu, J., Shekh, S., Sergiienko, N. Y., Cazzolato, B. S., Ding, B., Neumann, F., Wagner, M. (2016, July). Fast and effective optimisation of arrays of submerged wave energy converters. In Proceedings of the 2016 on Genetic and Evolutionary Computation Conference (pp. 1045-1052). ACM.

[5] Arbonès, D. R., Ding, B., Sergiienko, N. Y., Wagner, M. (2016, September). Fast and Effective Multi-objective Optimisation of Submerged Wave Energy Converters. In International Conference on Parallel Problem Solving from Nature (pp. 675-685). Springer International Publishing.

[6] Friedman J H. Greedy function approximation: a gradient boosting machine[J]. Annals of statistics, 2001: 1189-1232.

[7] Hara, K., Chellappa, R. (2013). Computationally efficient regression on a dependency graph for human pose estimation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3390-3397).
[8] Breiman, Leo; Friedman, J. H., Olshen, R. A., Stone, C. J. (1984). Classification and regression trees. Monterey, CA: Wadsworth Brooks/Cole Advanced Books Software. ISBN 978-0-412-04841-8.
[9] Cecen A. (2017). Multivariate Polynomial Regression. URL: https://cn.mathworks.com/matlabcentral/fileexchange/34918-multivariate-polynomial-regression Viewed on 28 May 2017.
[10] epsoft (2017). Root Relative Squared Error URL: http://www.gepsoft.com/gxpt4kb/Chapter10/Section1/SS07.htm Viewed on 28 May 2017.
[11] Duggleby R G. A nonlinear regression program for small computers[J]. Analytical biochemistry, 1981, 110(1): 9-18.
[12] Ratkowsky, D. A., Giles, D. E. (1990). Handbook of nonlinear regression models (p. 128). New York: M. Dekker.
[13] Motulsky, H. J., Ransnas, L. A. (1987). Fitting curves to data using nonlinear regression: a practical and nonmathematical review. The FASEB journal, 1(5), 365-374.
[14] Elith, J., Leathwick, J. R., Hastie, T. (2008). A working guide to boosted regression trees. Journal of Animal Ecology, 77(4), 802-813.
[15] Blum A L, Langley P. Selection of relevant features and examples in machine learning[J]. Artificial intelligence, 1997, 97(1): 245-271.
[16] Kim C J. Polynomial fit of interferograms[J]. Applied optics, 1982, 21(24): 4521-4525.