

Improving the Auto scaling mechanism in Cloud computing environment using Support Vector regression and Bi-LSTM

MSc Research Project
Cloud Computing

Jackson Peter
Student ID: x20183305

School of Computing
National College of Ireland

Supervisor: Divyaa Manimaran Elango

National College of Ireland
Project Submission Sheet
School of Computing



Student Name:	Jackson Peter
Student ID:	x20183305
Programme:	Cloud Computing
Year:	2021
Module:	MSc Research Project
Supervisor:	Divyaa Manimaran Elango
Submission Due Date:	31/01/2022
Project Title:	Improving the Auto scaling mechanism in Cloud computing environment using Support Vector regression and Bi-LSTM
Word Count:	677
Page Count:	3

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	31st January 2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	<input type="checkbox"/>
Attach a Moodle submission receipt of the online project submission , to each project (including multiple copies).	<input type="checkbox"/>
You must ensure that you retain a HARD COPY of the project , both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	<input type="checkbox"/>

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Improving the Auto scaling mechanism in Cloud computing environment using Support Vector regression and Bi-LSTM

Jackson Peter
x20183305

1 Objective

The main objective of this document is to provide an overview of steps or tasks to implement and evaluate the coding part of my research submission, from an external machine or environment. The following sections will describe the various modules required to run the project code.

2 Essential Module requisite

The basic requirement for executing the scripts is Python language and the version should be 3.8 or latest builds. Python code can be run on any machine irrespective of the operating system. However, machines with Linux distribution is preferred. In my implementation, I have deployed the code on Ubuntu 20.04. Furthermore, pip, the package manager should be enabled for initiating the extra packages.

3 Extra Library packages

In addition to the basic packages, few additional libraries are necessary for running the code. The 'TensorFlow' and 'scikit-learn' packages are associated with the machine-learning and deep-learning algorithms implemented for the performance evaluation after the training. The 'math' and 'pandas' functions are required for the data set generation which work in conjunction with 'sin' function. 'Matplotlib' library is required for generating line graphs and bar graphs post the code execution phase on all 3 algorithms. The 'NumPy' package is deployed for the calculations on matrix data. The aforementioned libraries can be installed with a single command as mentioned in the following section or can be installed separately.

```
pip install tensorflow sklearn numpy pandas matplotlib
```

4 Evaluation

For the evaluation of proposed architecture, one need to familiarize with the main python files of the implementation and general dataset. The dataset 'data.csv' file is having

5000 cycles of values created with sin function. It is generated to match the real-world traffic which will continuously fluctuate between a minimum and maximum value. The 'wave.py' file has created to generate the synthetic dataset to match with usual cloud computing traffic by merging three sin functions and the resulted data is used for train the machine-learning and deep-learning algorithms.

Once the data is generated, the process will focus on the main python file which is 'components.py'. In this file, all the major implementation operations are defined to make use by all the 3 algorithms. The file describes the warm-up time and loads the historical data. The different calculations are defined based on the load and the capability of the processing server to predict the next cycles. The other files are 'ml-models.py' and 'dl-models.py' which are responsible for the training of machine-learning and deep-learning procedures respectively.

5 Results

Finally, one has to check the performance of each algorithm against the trained and need to analyze the different parameters based on the output values. The three different algorithms included in the research were Linear Regression (LR), Support Vector Regression (SVR) and Bi-directional LSTM (Bi-LSTM). The below commands are executed for the evaluation purpose.

```
python3 components.py LR
python3 components.py SVR
python3 components.py BI
```

On executing each command, a line graph and a bar graph would get generated with the output values against the received traffic load. The below images represents the each algorithm's output value.

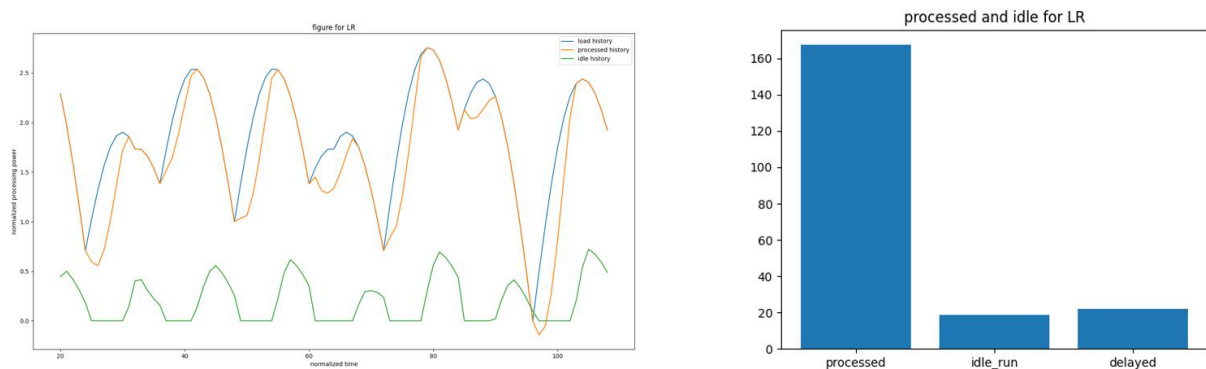


Figure 1: LR - Total Processed, Idle and delayed Load

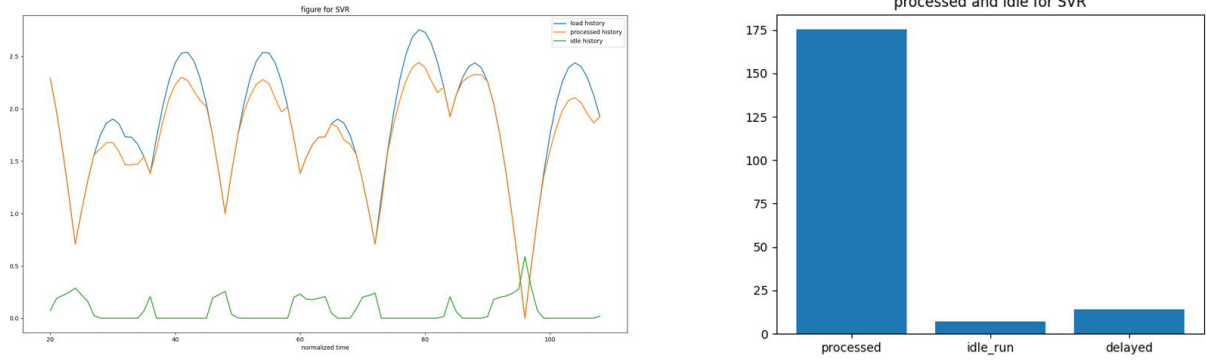


Figure 2: SVR - Total Processed, Idle and delayed Load

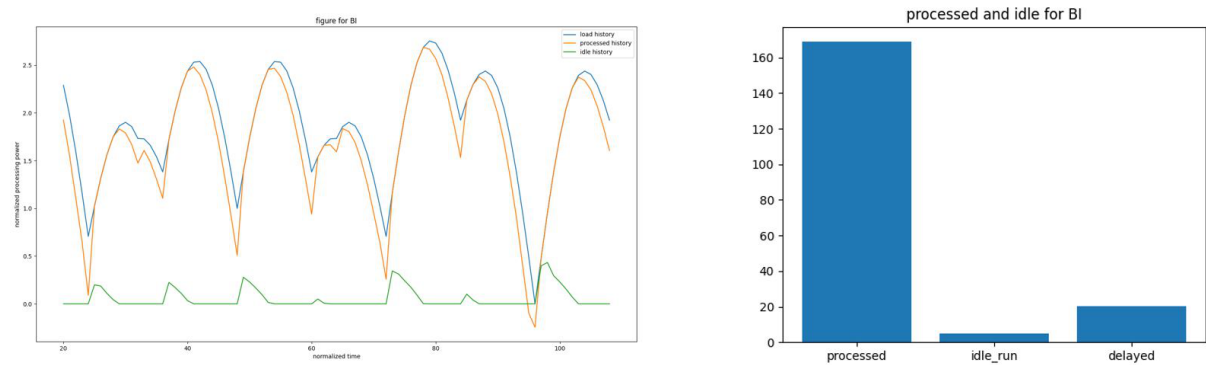


Figure 3: BI - Total Processed, Idle and delayed Load

6 Conclusion

To conclude and finish the procedure, the output values are captured after a typical run and comparison can be visualised as shown in below table. From the generated prediction and output values, it is clear that SVR and Bi-LSTM algorithms are more efficient than LR algorithm. SVR processes more incoming load whereas Bi-LSTM excels with minimal idle cycles.

	LR	SVR	Bi-LSTM
Total Processed	167.44	175.17	168.83
Idle Cycle	18.60	6.848	4.886
Delayed	21.76	14.04	20.38

Figure 4: Output Comparison - LR, SVR and Bi-LSTM