National College of Ireland

Configuration Manual

MSc Research Project
Msc Cloud Computing

Rajkumar Malarkodi Paramasivam
Student ID: 23127180

School of Computing
National College of Ireland

Supervisor: Jitendra Kumar Sharma

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Rajkumar Paramasivam Malarkodi
Student ID:	23127180
Programme:	Msc Cloud Computing **Year:** 2024
Module:	Research Project
Lecturer:	Jitendra Kumar Sharma
Submission Due Date:	12/08/2024
Project Title:	Using step scaling policy with deep learning to enhance autoscaling
Word Count:	1811 **Page Count:** 10

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: ……

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
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.	□

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):	

1. Introduction

This Configuration Manual is used for preparing, configuring, and running a project. Its major goal is to make sure that users perform the project properly, and, for this reason, it describes system prerequisites and software needed along with analysis specifics. This paper describes a project that uses an advanced approach of machine learning models for better auto-scaling in cloud computing. The purpose of the project is to employ deep learning in the task to make the model even more scalable and accurate. The major goal of the paper is to provide a firm understanding and detailed guide on components of the project as well as comprehensive instruction on necessary software installation and configuration. Apart from that, the paper describes data processing and model development about the accomplished scope, which is comprehensive from hardware and software analysis to machine learning model running. This manual is designed for researchers and IT professionals who will work on the deployment and operation of the project. The main purpose is to facilitate a proper setup and operational process to enable people to work with.

2. Project Files Detail

The files associated with the current dissertation are crucial for implementing and understanding the development of the cloud auto-scaling system implemented with the enhancement of Deep Learning. The primary file used for the purpose is auto_scaling_project.ipynb, a Jupyter Notebook that exists as the central source for Deep Learning model implementation, as well as data parsing and analysis. The file is split into corresponding sections, including data preparation, model training, evaluation, and results sections, ensuring high interaction with users. The outer dataset file, vm_allocation_migration.csv, exists within the latter notebook, providing extensive information regarding virtual machine allocation, and migration metrics, such as the level of CPU, memory, and network usage. The corresponding notebook is reliant on external libraries and modules beyond the basic Anaconda data science toolkit, the relational links to which should be referenced. Further supplementary files next to the project include corresponding configuration files, beyond the documentation prioritizing explanation of steps required for project environment setting and analysis execution. All of the files represent a more structured approach to project initiation and the creation of an opportunity for others to replicate the study.

3. System Specification

3.1 Hardware Requirements

The following are the minimum hardware requirements for this project, which are designed to allow the efficient processing and analysis of large-scale data in cloud computing settings. Data-intensive operations and Deep Learning model computations on a minimum configuration of an Intel Core i5 processor with at least 8 GB of RAM can be effectively conducted. An SSD with at least 50 GB of available disk space is recommended to ensure rapid data delivery and processing. Thus, while the primary execution of the project is to be carried out on cloud platforms called AWS Sagemaker, in which scalable resources are provided, this hardware can secure a smooth run for the project's local development and testing. Such configuration of the hardware allows dealing effectively with the completion of computational tasks, data preprocessing, and model training.

3.2 Operating System

In this project, AWS SageMaker is used and it works within the general AWS environment. It is compatible with many operating systems, and within AWS, Amazon Linux is the default choice. It offers high performance and security as it is specifically designed for AWS. As such, it is compatible with various systems and tools necessary for training deep learning models. Meanwhile, Windows 10 or better is a viable alternative for local development due to user convenience However, all heavy computational operations are performed by AWS SageMaker, allowing users to train and deploy models in a robust, highly scalable, and flexible environment. Meanwhile, operating systems are handled by the system and are configured automatically. Overall, it is flexible, and any user comfortable with Jupyter Notebooks can quickly begin training and developing in the cloud.

4. Software Used

4.1 Programming Language

The primary programming language is Python and this decision was made due to this language’s vast support of data science and machine learning libraries. Because of such extensive support, Python is used in almost any machine learning course. Overall, Python has many packages and frameworks well-suited for creating complex deep-learning models and working with large datasets. Its relative simplicity and readability also allow it to develop solutions faster and conveniently run different approaches and parameter configurations in iterative experiments. In the context of AWS SageMaker, Python also integrates well with different machine learning and data processing tools, allowing to quickly train and deploy the

developed models. Therefore, the chosen programming language satisfies the requirements of the project.

4.2 Development Environment

The development environment for this project is AWS SageMaker. AWS SageMaker is a fully managed service that simplifies the process of building, training, and deploying machine learning models. It has integrated Jupyter notebooks to interactively develop and explore the data and models. The service also provides several supported machine-learning algorithms and frameworks, ranging from TensorFlow, PyTorch, and scikit-learn. SageMaker also provides tools for distributed training and automatic model tuning. Since SageMaker offers scalable infrastructure, the computations are not directly limited by the hardware resources, as would be the case for a local machine. By using SageMaker, the development process is streamlined and the deployment of the developed models is simplified via integration with other AWS services for, e.g., storing data.

4.3 Libraries and Frameworks

In order to help with machine learning and various data analysis tasks, the project is dependent on several critical libraries and frameworks. For instance, TensorFlow and PyTorch are relied on for the development and training of deep learning models and they provide a lot of flexibility as well as advanced technologies in terms of designing appropriate neural network architectures. Additionally, scikit-learn is increasingly used for data preprocessing, selecting key features, and estimating how well models perform and it provides all of the necessary tools in one package. Meanwhile, to operate on data easily and perform correct numerical operations, Python's Pandas and NumPy are essential. Lastly, Matplotlib and Seaborn are used to create important points and properly interpret results. In this way, the operation, modeling, and analysis of data are efficient in the AWS SageMaker environment.

4.4 Package Manager

This project is based on pip because it is the most commonly used package manager for Python. The work of engineers that use Python would be impossible without pip, as it is a simple tool created to simplify the work with Python packages, including ones from Python Package Index. Pip allows to easily access, install, and manage different libraries available through PyPI. It has streamlined the installation of such important packages as TensorFlow, PyTorch, scikit-learn, Pandas, NumPy, Matplotlib, and Seaborn, which are vital for machine learning, data processing, and visualization. Notably, pip ensures that all of the required

packages are up to date-and compatible with each other, making the development process more hassle-free and providing a more stable environment overall.

5. Project Development

5.1 Importing Library

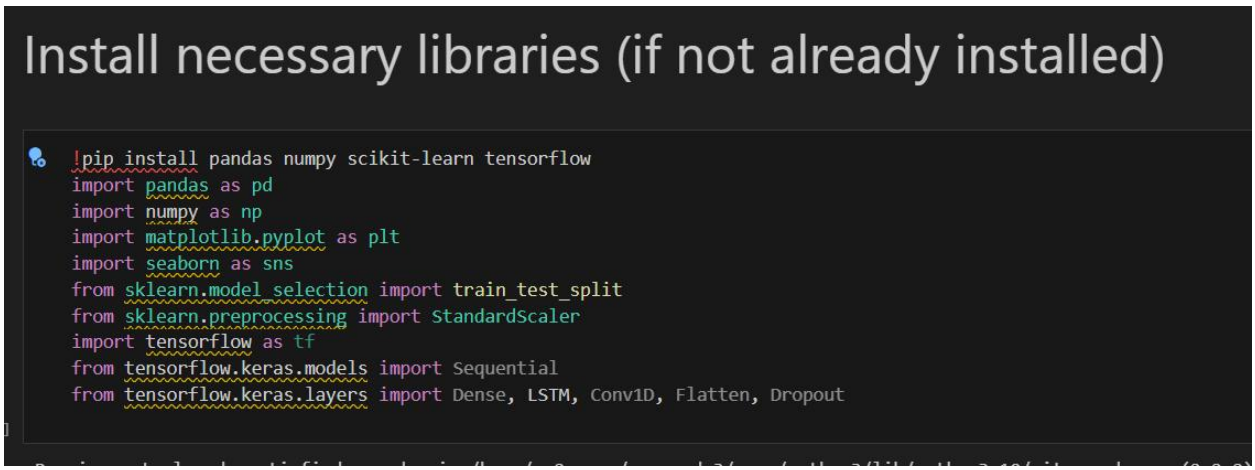

Figure 1: Importing Library

(Source: Acquired from Jupyter Notebook)

During the project development phase, importing libraries is the critical step for accessing various features used for visualization and analysis purposes. Information about libraries that must be accessed is imported using Python's import statement. For data manipulation and analysis, different Pandas and Numpy are imported. Data-visualization package is handled by Matplotlib and Seaborn. The package serves as a tool that supports the code for plotting and understanding data patterns. For the machine-learning process, Scikit-learn is imported for conducting the process of building and evaluating the model. Moreover, for implementing the deep-learning model, TensorFlow or PyTorch is used. Importing the mentioned libraries helps to access a broader range of tools and methods that are required for data visualization, improves the data processing and project outcome, and develops efficient working time.

5.2 Importing Files

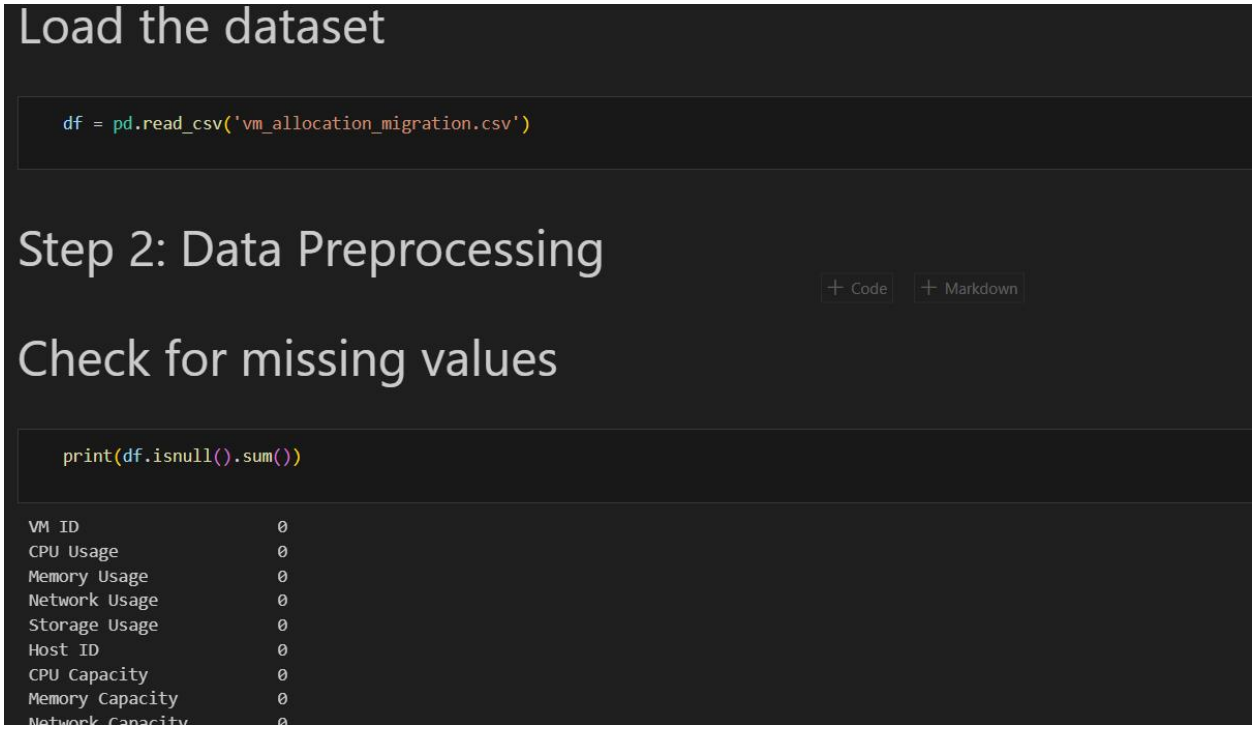

Figure 2: Importing Files

(Source: Acquired from Jupyter Notebook)

One of the key tasks in project development is importing the dataset. For AWS SageMaker and further data processing, it is essential first to upload the dataset to an S3 bucket. Alternatively, the data can be taken from the already uploaded file, and the boto3 library allows interaction with other AWS services. Then, with the assistance of the pandas library working with data in SageMaker, the dataset can be read into a data frame. By using string_io and pd.read_csv(), the dataset file is then uploaded into a data frame that will be used in this assignment for cleaning and analysis purposes. Such an approach will guarantee properly formatted data for further project development and modeling.

5.3 Data Cleaning and Processing

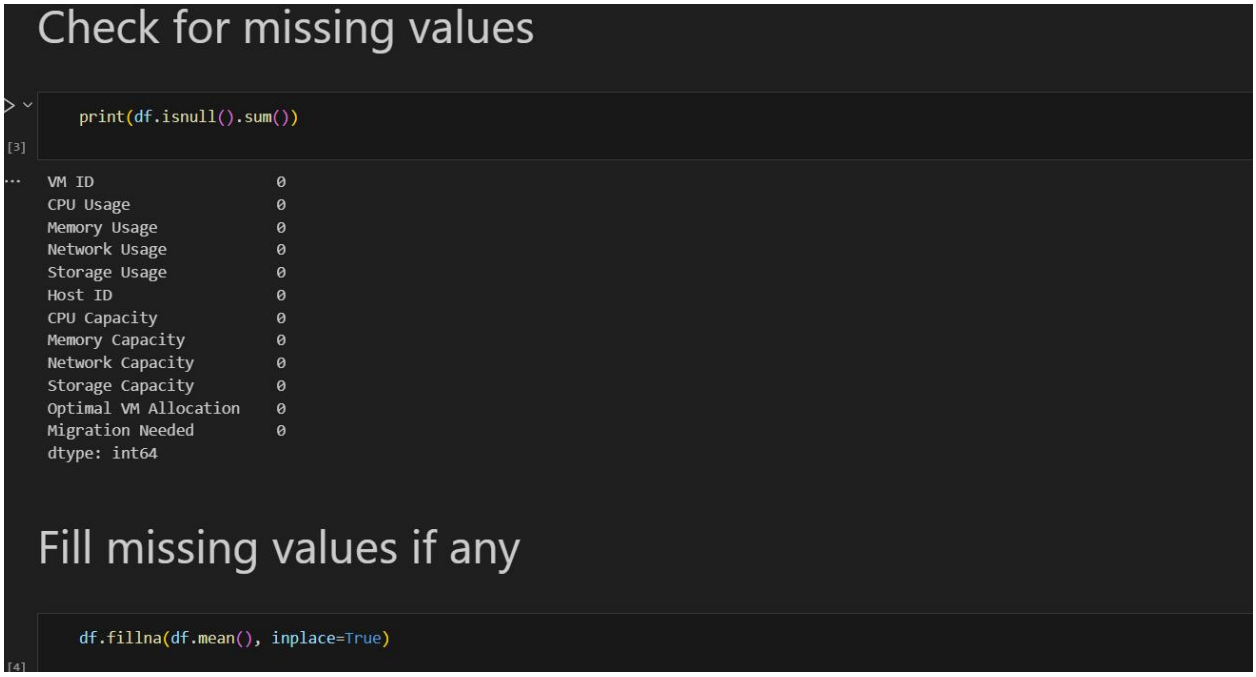

Figure 3: Data Cleaning and Processing

(Source: Acquired from Jupyter Notebook)

Data cleaning and processing are very important to have accurate model training. Checking whether the data includes missing values is the initial step. This can be done by executing isnull().sum() in the Pandas library. In this case, the procedure returns zero for all columns, but if there were some gaps in the data, these gaps would have been counted here. If there have been any missing values in the dataset, it is obligatory to consider them. However, in this case, it is necessary to fill missing data with appropriate values. This requirement can be met by replacing missing data with the mean in each column by applying fillna(df.mean(), inplace=True). This step is crucial to ensure that the dataset carrying out further statistical analyses and the training of the current model is not compromised by missing data. It will allow one to use the fill method to put missing data. These data will be replaced by the mean of all numbers within the same column. Then further model training and preprocessing actions, such as normalization or encoding, can be completed safely.

5.4 Modeling

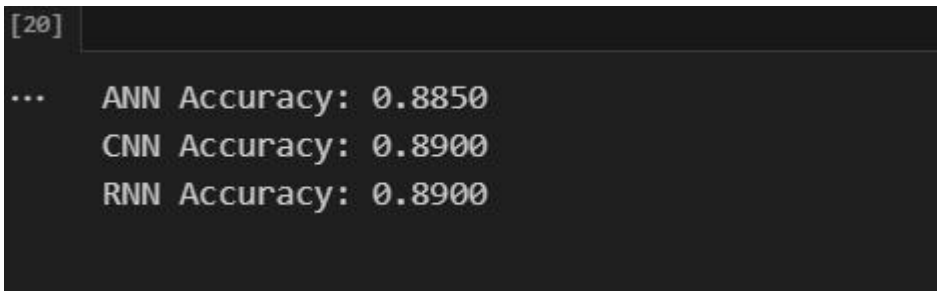

Figure 4: Modeling

(Source: Acquired from Jupyter Notebook)

Modeling is the essential phase of any machine learning system implementation as it allows one to evaluate the performance of different algorithms. In this way, in the current study, several models have been assessed to be able to predict outcomes based on the given dataset. Artificial Neural Network has shown a relatively high accuracy rate equal to 88.50%, which makes it possible to suppose that this model will be able to capture complicated patterns hidden in the dataset. At the same time, the Convolutional Neural Network and Recurrent Neural Network have both presented higher results among the evaluated models, 89.00%, showing their ability to extract relevant features and time-dependent sequences. Therefore, CNN's capacity to deal with spatial hierarchies and RNN's appropriateness to manage the sequences dependent on time constitute the possible reasons for the described approach's higher effectiveness. As a result, one might note the fact that in the case of modeling, one should take into account the type of algorithm one will have to use for data and the nature of the task to attain the desired result in a specific situation.

References

Agarwal, S., Rodriguez, M. A., & Buyya, R. (2024). A Deep RecurrentReinforcement Learning Method for Intelligent AutoScaling of Serverless Functions. *IEEE Transactions on Services Computing*.

Bali, A., El Houm, Y., Gherbi, A., & Cheriet, M. (2024). Automatic data featurization for enhanced proactive service autoscaling: Boosting forecasting accuracy and mitigating oscillation. *Journal of King Saud UniversityComputer and Information Sciences*, *36*(2), 101924.

Dogani, J., & Khunjush, F. (2024). Proactive autoscaling technique for web applications in containerbased edge computing using federated learning model. *Journal of Parallel and Distributed Computing*, *187*, 104837.