Configuration Manual

MSc Research Project
MSc Cloud Computer

Chinmay Dhanawade
Student ID: X23200197

School of Computing
National College of Ireland

Supervisor: Prof. Sudarshan Deshmukh

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Chinmay Dhanawade
Student ID:	X23200197
Programme:	MSc Cloud Computer
Year:	2024
Module:	MSc Research Project
Supervisor:	Prof. Sudarshan Deshmukh
Submission Due Date:	12/12/2024
Project Title:	Configuration Manual
Word Count:	1242
Page Count:	8

Signature:	
Date:	27th January 2025

Hybrid models Cloud-based Enhancements for Air Quality Prediction Systems

Chinmay Dhanawade
X23200197

12/12/2024

1 System Requirements

Before starting the setup process, ensure that your system meets the following requirements:

- An internet connected computer.
- An active AWS account.
- Knowledge of Python programming.
- Knowledge about cloud computing concepts.
- Ram Available 16 GB
- 100 GB Disk space
- LAN/ WAN Connection
- VRAM need 512MB to 2 GB
- OS Win/Mac/ Linux

2 Cloud Environment Setup

2.1 AWS Account Creation

To begin using AWS services, you need to create an account:

- Visit the AWS website.
- Then click on 'Create an AWS Account.'
- There, enter your email address, password, and account details and follow the prompts.
- Verify and sign in to your AWS Management Console.

Services (2020a)

2.2 Setting Up AWS Cloud9

Amazon Cloud9 will be used to create the application and to control the resources.

- Jump into the AWS Management Console –¿ "Cloud9."
- Click "Create Environment"
- Enter a unique name for the environment
- Provide a description (optional)
- Configure environment settings :
  - Select **Create a new EC2 instance for environment (direct access)**.
  - Choose an instance type (e.g., `t2.micro` for small workloads).
  - Select the platform (Amazon Linux, Ubuntu, etc.).
- Set an automatic environment shutdown timeout if required (default is 30 minutes).
- Review and click "Create"
- Access your Cloud9 environment

Services (2023a)

2.3 Setting Up AWS Elastic Beanstalk

AWS Elastic Beanstalk will be used to deploy the Flask application:

- From the AWS Management Console, go to "Elastic Beanstalk."
- Click on "Create Application."
- Enter an application name, for example, AirQualityPredictionApp.
- Select a platform, which in this case is Python, and select the appropriate version.
- Click on "Create environment" under Web server environment.

Set up environment settings as necessary and click "Create environment." Services (2020b) Services (2021)

2.4 Setting up AWS Codepipeline

- Log in to the AWS Management Console
- Navigate to CodePipeline
- Click "Create Pipeline"
- Enter a unique pipeline name
- Service Role :Select **Create a new service role** (recommended for new users) or choose an existing role if applicable.

- Add source stage : For **GitHub**, connect to your account and select a repository and branch.
- Add build stage (optional)
- Add deploy stage : Choose a deployment provider: **Amazon ECS**, **AWS Lambda**, **AWS Elastic Beanstalk**, or **S3** for deployment.
- Review and click "Create pipeline"
- Monitor your pipeline:Once created, your pipeline will automatically start running based on the source triggers.

Services (2023b)

3 Development Environment Setup

Make sure Python is installed on your local computer:
Download Python from python.org.

3.1 Installing Necessary Software

- Install Python. The installation procedure varies depending on your operating system. Setting up the Python Environment.
- For creating a virtual environment for the project:
  - Open a CLOUD9 instance.
  - A virtual environment is created
- Install libraries
- pip install pandas numpy scikit-learn statsmodels flask boto3 matplotlib seaborn keras

Foundation (2020)

4 Model Development

- This SARIMA model will be implemented using historical air quality data to capture the linear trends very effectively.
- ***Random Forest Model Implementation***
  - The Random Forest model shall be used for non-linear relationships in the air quality data.Both of them need to be trained, then integrated into a final hybrid model.
- ***Hybrid Model Integration***

  - Once the two models are built, they need to be combined into a hybrid model which effectively combines the two models' predictions. Integration could be in terms of averaging predictions or weighted averages of model performance during validation.

Johnson (2022)

5 Deployment Process

The steps to deploy the application are:

- Create a Flask application in your project directory with necessary files (app.py, requirements.txt, etc.).
- Start Elastic Beanstalk in your project directory: eb init -p python-3.x AirQualityPredictionApp –region us-west-2
- Create an environment and deploy: eb create air-quality-env –single –instance_type t2.micro

This deployment process makes sure that your application is online and ready for users to interact with
Services (2021)

6 Setting Up Git

Before you push code to Git, ensure that you have Git installed on local machine. It can download and install it from git-scm.com.

6.1 Setting Up a Git Repository

- Access Your Project Directory: Open your terminal or command prompt and change into the directory containing your project files.: cd path/to/your/project
- Initialize Git: Run the following command to initialize a new Git repository: git init
- Add Files: Add all project files in the staging area: git add.
- Commit Changes: Commit all files added with a descriptive message.: git commit -m "Initial commit of air quality prediction system"

6.2 Creating a Remote Repository

To push your local repository to a remote server, say GitHub, follow these steps:

- Create New Repository on GitHub
- Go to your GitHub account

- Click on the "+" icon at the top right and select "New repository."
- Name your repository (e.g., air-quality-prediction), add a description, and make it public or private.
- Click "Create repository."
- Link Local Repository to Remote
- Copy the remote repository URL from GitHub.
- Connect your local repository to the remote one: git remote add origin https://github.com/youruse quality-prediction.git 3. Pushing Code to GitHub Connect your local repository to the remote one and you can now push changes: Push Your Code: You can then push your local commits to the remote repository using: git push -u origin master

This command uploads all the committed changes from your local master branch to the remote master branch in GitHub. GitHub (2020)

6.3 Jupyter Notebook

Jupyter Notebook is an open-source web application that allows you to create and share documents containing live code, equations, visualizations, and narrative text. Follow these steps to install and set up Jupyter Notebook using Anaconda:

1. **Download Anaconda:** Visit the official Anaconda website at `https://www.anaconda.com/` and download the Anaconda distribution suitable for your operating system (Windows, macOS, or Linux).
2. **Install Anaconda:** (Anaconda; 2023)
   - For Windows: Run the installer and follow the on-screen instructions. Ensure you check the option to add Anaconda to your PATH environment.
   - For macOS and Linux: Open the downloaded installer file and follow the installation prompts in the terminal.
3. **Verify Installation:** Open a terminal or command prompt and type:

   ```
   conda --version
   ```

   This command should return the installed version of Conda.
4. **Launch Jupyter Notebook:** (?) After installation, launch Jupyter Notebook by running the following command in the terminal or Anaconda Prompt:

   ```
   jupyter notebook
   ```

   This will open the Jupyter Notebook interface in your default web browser.
5. **Create a New Notebook:** Click `New` in the top-right corner and select `Python 3` to create a new Python notebook.

6.4 Setting Up Google Colab

Google Colab is a cloud-based notebook environment that allows you to write and execute Python code in an interactive, collaborative way. To set up Google Colab, follow these steps (Colaboratory; 2023):

1. **Access Google Colab:** Open your web browser and navigate to `https://colab.research.google.com/`.
2. **Sign In with Google Account:** Log in using your Google account. If you do not have one, create a Google account at `https://accounts.google.com/`.
3. **Create a New Notebook:** Once logged in, click on `File > New Notebook` to create a new Colab notebook. Alternatively, click the `+ New Notebook` button on the homepage.
4. **Upload Notebook Files:** If you want to run an existing notebook file, click `File > Upload Notebook` and upload the required `.ipynb` file.
5. **Install Libraries:** Install any additional Python libraries using the `!pip install` command in a code cell. For example:

   ```
   !pip install pandas
   ```

6. **Save and Share:** Google Colab automatically saves your work in your Google Drive. To share the notebook, click `Share` in the top-right corner and configure sharing permissions.

7 Deployment Process

7.1 Upload Your Application

Choose one of the following methods:

- **ZIP File Upload**: Create a ZIP file containing your application and upload it directly in the Elastic Beanstalk console.
- **GitHub Integration**: Use the repository and branch you linked earlier for deployment.

7.2 Deploy Application:

Once uploaded, click **Deploy** in the Elastic Beanstalk console.

7.3 Monitor Deployment:

- AWS Elastic Beanstalk will handle provisioning, deployment, and scaling.
- Check the deployment logs and monitor status in the **Events** section.

7.4 Access Your Application

Once successfully deployed, Elastic Beanstalk provides a unique URL (e.g., `http://<your-enviro`
to access your application.

7.5 Post-Deployment Steps

- Testing -Open the application URL and test its functionality. Enter sample inputs for AQI predictions to confirm the app works as expected.
- Performance Monitoring
- Enable Logging - Go to the **Logs** section in the Elastic Beanstalk console. Enable enhanced logging to troubleshoot any issues with the application.
- Scaling and Updates - Adjust instance scaling if necessary for performance improvements.

8 Testing and Evaluation

Once deployed, test the application by accessing it through its URL given by Elastic Beanstalk.

Evaluation Metrics:

Evaluate the model based on RMSE and MAE metrics using predictions against actual AQI values.

8.1 Troubleshooting Common Issues

- While setting up and running the system, you may encounter a number of common issues
- Deployment Errors: Ensure that all dependencies are listed in requirements.txt.
- Data Access Issues: Check IAM permissions for accessing S3 buckets or other resources.
- Model Performance: Hyperparameter tuning or different modeling approaches if performance is poor

9 Future Enhancements

Consider implementing additional features such as:

- Real-time data streaming from IoT devices.
- Personalized experiences using user authentication.
- Advanced visualization tools to have a better insight into air quality trends.

Following this configuration guide, you should be able to deploy and run your air quality prediction system efficiently within a cloud environment by adopting hybrid modeling approaches that combine statistical methods with machine learning techniques for better accuracy in real-time applications. This comprehensive guide will be the roadmap for the successful implementation of an air quality prediction system that exploits modern cloud technologies while being scalable, flexible, and accurate in its predictions to make a big difference in public health outcomes and environmental management strategies.

References

Anaconda, I. (2023). Download anaconda distribution. Accessed: 2023-12-10. **URL:** *https://www.anaconda.com/*

Colaboratory, G. (2023). Welcome to colaboratory. Accessed: 2023-12-10. **URL:** *https://colab.research.google.com/*

Foundation, P. S. (2020). Installing python, `https://www.python.org/downloads`. Accessed 11 Dec. 2024.

GitHub, I. (2020). Setting up git and creating a repository, `https://git-scm.com/book/en/v2/Getting-Started-Installing-Git`. Accessed 11 Dec. 2024.

Johnson, L. (2022). Hybrid modeling approaches in environmental data analysis, *Environmental Science & Technology* **56**(1): 123–135.

Services, A. W. (2020a). Aws account creation, `https://aws.amazon.com`. Accessed 11 Dec. 2024.

Services, A. W. (2020b). Setting up aws elastic beanstalk, `https://aws.amazon.com/elasticbeanstalk`. Accessed 11 Dec. 2024.

Services, A. W. (2021). Deploying applications with aws elastic beanstalk, `https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/Welcome.html`. Accessed 11 Dec. 2024.

Services, A. W. (2023a). Setting up aws cloud9, `https://aws.amazon.com/cloud9`. Accessed 10 Dec. 2024.

Services, A. W. (2023b). Setting up aws codepipeline, `https://aws.amazon.com/codepipeline`. Accessed 11 Dec. 2024.