National College of Ireland

Configuration Manual

MSc Research Project
Cloud Computing

Anisha Khalam
Student ID: X23401061

School of Computing
National College of Ireland

Supervisor: Dr Giovani Estrada

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Anisha Khalam
Student ID:	X23401061
Programme:	Cloud Computing
Year:	2018
Module:	MSc Research Project
Supervisor:	Dr Giovani Estrada
Submission Due Date:	20/12/2018
Project Title:	Configuration Manual
Word Count:	761
Page Count:	7

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:	Anisha Khalam
Date:	10th December 2025

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):	

Configuration Manual

Anisha Khalam
X23401061

1 System Requirements

1.1 Hardware Requirements

1.1.1 Minimum Requirements:

- CPU:4 cores (Intel Core i5 or AMD Ryzen 5 equivalent)
- RAM: 8 GB
- Storage: 10 GB free space
- Network: Stable internet connection (for dataset download)

1.1.2 Tested Configurations

- Configuration 1 (Development):
  - CPU: Intel Core i7-10700K (8 cores)
  - RAM: 16 GB DDR4
  - GPU: None (CPU-only training)
  - OS: Windows 11
  - Training Time: 1 hours per task (forecasting)
- Configuration 2 (AWS Deployment):
  - Instance Type: AWS EC2 t3.small and AWS EC2 t2.large
  - CPU: 2 vCPUs
  - RAM: 2 GB
  - OS: Amazon Linux 2

1.2 Software Requirement

- Operating System
  - Windows 10/11 (64-bit)
  - Ubuntu 20.04/22.04 LTS
  - macOS 11+ (Big Sur or later)

  - Amazon Linux 2 (for AWS deployment)

- Required Software
  - Python 3.11.x (tested with 3.11.9)
  - pip (Python package manager)
  - Git (for repository management)

- Optional
  - Anaconda/Miniconda (recommended for environment management)
  - CUDA Toolkit 11.8+ (if using GPU)

2 Environment Setup

2.1 Python Installation

2.1.1 Windows

- Download Python 3.11.9 from python.org
- During installation, check "Add Python to PATH"
- Verify installation: python –version

2.1.2 Linux/Ubuntu

- sudo apt update
- sudo apt install python3.11 python3.11-venv python3.11-dev
- python3.11 –version

2.2 Virtual Environment Setup

- Navigate to project directory
- Create virtual environment Windows:venv\Scripts\activate, Linux:venv/bin/activate
- Verify activation (prompt should show (venv))

3 Installation Instructions

3.1 Clone Repository

- Git repository (Khalam; 2025): git clone <https://github.com/Anisha151098/vm_predictor>
- cd GRU

3.2 Install Dependencies

- Install Core Dependencies
  - Ensure virtual environment is activated
  - Install all required packages
    - pip install tensorflow==2.17.0 (Abadi et al.; 2016)
    - pip install numpy==2.2.3
    - pip install pandas==2.2.3
    - pip install scikit-learn==1.6.1
    - pip install matplotlib==3.10.1
    - pip install seaborn==0.13.2
    - pip install requests
    - pip install tqdm
    - pip install h5py

- Create requirements.txt
  - pip freeze > requirements.txt

4 Running the Code

- Activate environment:for Windows: venv\Script\activate. For Linux\Ubuntu: source venv\bin\activate
- Run forecasting training: python train_azure_forecast_1hr.py, this will try to download dataset (Azure; 2017)
- This will trains 5 models with forecasting labels and the results are saved in results_azure_forecast_1hr
- It takes around 2 to 3 hours to train all models.
- create_individual_graphs.py will generate graphs.

5 AWS Deployment

- Before starting ensure that an active AWS with billing is provided, IAM user with appropriate permission

5.1 EC2 Setup

- Open AWS Management console and sign in
- Navigate to EC2→ Launch Instance
- Name the instance (Eg:GRU\Attention), select Amazon Linux as the Operating System. Select Instance type as t2.medium. Create a key pair of .pem format for login and download to the local system. Click launch instance. Review and launch instance (Services; 2025b).

5.1.1 IAM policy For EC2 Instance

- Navigate to IAM. From AWS Management Console, go to services select Security, Identity & Compliance → select IAM
- IAM → Roles → Create role → Select service or usecase as EC2 → Next
- Now select the policies needed, search AmazonEC2ReadOnlyAccess and CloudWatchReadOnlyAccess (Services; 2025a). Enable this permissions → Next
- Give a role name → Create role
- To **Create inline policy** Go to the role created → In permissions policies, click on Add permissions → Create inline policy. In policy editor, click on json, then add this json script to consolidate the idle instances forecasted.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:StopInstances"
            ],
            "Resource": "*"
        }
    ]
}
```

- Next → Give policy name → Create policy

5.2 Launch 3 t3.micro instances

- Navigate to: EC2 → Instances → Launch Instance
- Configuration: Name: ec21 (repeat for ec22, ec23) AMI: Amazon Linux Instance type: t3.micro Key pair: Create new or use existing Network settings: Default VPC
- Launch instance
- Repeat 3 times total

5.3 Workload setup for instance

5.3.1 Variable Spikes Pattern for ec21

- SSH into ec21(ssh -i vm-predictor-key.pem ec2-user@)
- Install stress tool - sudo yum install stress -y
- Create workload script

```
    cat << 'EOF' > /home/ec2-user/workload_ec21.sh

#!/bin/bash
while true; do
  # 5 minutes idle (0-2% CPU)
  sleep 300

  # 2 minutes spike (60-80% CPU)
  stress --cpu 1 --timeout 120s

  # 3 minutes moderate (20-30% CPU)
  stress --cpu 1 --timeout 180s --cpu-method fibonacci
done
EOF
```

- Run:

```
chmod +x /home/ec2-user/workload_ec21.sh
```

- Run this to keep the scrpt run in background:

```
nohup /home/ec2-user/workload_ec21.sh > /dev/null 2>&1 &
```

5.3.2 Sustained High Load in ec22

- ssh to the instance
- Install stress :sudo yum install stress -y
- Run this for sustained load:

```
        cat << 'EOF' > /home/ec2-user/workload_ec22.sh
#!/bin/bash
stress --cpu 1 --cpu-method ackermann &
EOF
```

- Run this to keep the workload run in background:

```
        chmod +x /home/ec2-user/workload_ec22.sh
nohup /home/ec2-user/workload_ec22.sh > /dev/null 2>&1 &
```

5.4 Connect EC2 via SSH and Deployment set up in EC2

- On the local machine, Navigate to the directory where the pem file of the instance downloaded.
- Connect the instance via ssh
- Verify the trained model exist in same directory
- Run extract_model.py from the git repo.
- Verify the dependencies installation
- Install pip install boto3==1.35.0
- Verify AWS credentials configured
- nano ec2_forecast_exact_architecture.py, copy and paste ec2_forecast_exact_architecture.py code from the git repo
- Run this code python ec2_forecast_exact_architecture.py
- Forecast result will be shown in the command prompt.

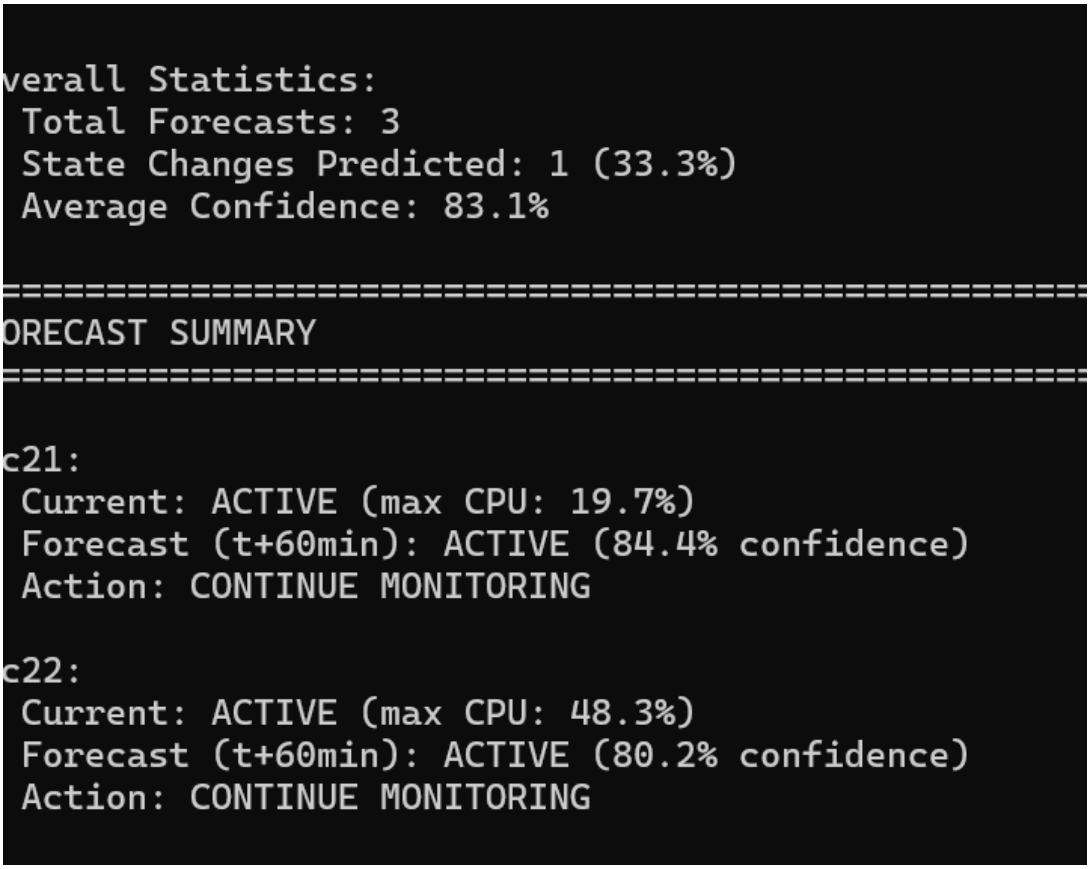

Figure 1: Sneak peek of output obtained

References

Abadi, M. et al. (2016). Tensorflow: A system for large-scale machine learning, *Proceedings of OSDI*, pp. 265–283.
URL: *https://www.tensorflow.org/*

Azure, M. (2017). Azure public dataset v1.
URL: *https://github.com/Azure/AzurePublicDataset*

Khalam, A. (2025). Vm state predictor - gru-attention architecture. GitHub repository.
URL: *https://github.com/Anisha151098/vm_predictor*

Services, A. W. (2025a). Amazon cloudwatch documentation.
URL: *https://docs.aws.amazon.com/cloudwatch/*

Services, A. W. (2025b). Amazon ec2 documentation.
URL: *https://docs.aws.amazon.com/ec2/*