National College of Ireland

Configuration Manual

MSc Research Project
Programme Name

Harshitha Reddy Bachupalli Kollan
Student ID: X23300531

School of Computing
National College of Ireland

Supervisor: Aaloka Anant

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Harshitha Reddy Bachupalli Kollan
Student ID:	X23300531
Programme:	Msc. Data Analytics (MSCDAD_JAN25I) **Year:** 2025- 2026
Module:	MSc (Research) Practicum Part 2
Lecturer:	Aaloka Anant
Submission Due Date:	11-12-2025
Project Title:	Early Warning System for Distributed File System Failures using Temporal Log Analysis.
Word Count:	855…… **Page Count: 6**

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: Harshitha Reddy Bachupalli Kollan

Date: 11-12-2025

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):	

Early Warning System for Distributed File System Failures Using Temporal Log Analysis

Harshitha Reddy Bachupalli Kollan
X23300531

1 Introduction

The following configuration manual includes the technical specifications that would be necessary to install, set up and operate the early-warning prediction system that is created in the context of this research project. The system conducts a temporal modelling of the HDFS log data to forecast block failures before they happen by observing behaviour trends that change with time.
To compare the prediction of the two machine learning methods, the project undergoes two model types: XGBoost as a non-sequential baseline model and an LSTM with the ability to learn sequences of events.
This manual aims to allow any researcher, lecturer or external examiner to reproduce all the modelling pipeline and re-run the experiments to confirm the results discussed in the dissertation. It also defines all software requirements, directory layouts, data location, system dependencies, execution process and retraining the models. This is created with the aim of making the whole system reproducible and capable of being run in any computing environment that is compatible.

2 System Requirements

The present project was developed and executed fully within a Python-based data analytics environment by using Visual Studio Code (VS Code). The entire modelling, preprocessing, and evaluation work is only based on open-source libraries to ensure that it is completely reproducible.

- Windows 11 on an AMD Ryzen processor and with 816 GB of RAM available locally.
- Dependencies were managed and the version of the packages was isolated with a Python virtual environment (MLVE).
- All code was run within the VS Code with a Jupyter Notebook extension, which allowed the notebook (experimenty.ipynb) to run in an interactive environment.
- The system was written in python 3.12.

Data processing, modelling and visualisation required the following libraries: pandas and NumPy to manipulate data; scikit-learn to perform feature vectorisation, class weighting, and evaluation metrics, XGBoost to model non-sequential baselines, TensorFlow/Keras to implement the LSTM network and matplotlib and seaborn to generate evaluation plots.
To make sure that all the packages are in version, all packages were installed in the MLVE environment. All the necessary libraries were installed in the virtual environment using pip so as to avoid dependency conflicts.

2.1 Software Used

- Windows 11
- Python 3.12.x
- Jupyter extension of Visual Studio Code.
- MLVE (python -m venv created virtual environment).

2.2 Required Python libraries

```
Pip install pandas numpy scikit-learn xgboost tensorflow keras matplotlib seaborn
```

3 Dataset Description

The HDFS log data that is accessed by the LogHub repository is used in the early-warning system. The data set has multiple copies of the Hadoop Distributed File System (HDFS) logs, each of which is a collection of raw text files that document block actions, times, interfaces between nodes, and failures. This importation was done with pandas and regular expressions were used to extract the timestamps, block IDs and message content.
LogHub event templates were applied to decode free-text messages into event identifiers that can be used to form a temporal model.
In this project, HDFS_2k dataset was mainly utilized because it is the dataset that has structured and template-mapped versions of the logs that are needed to train both models. Other datasets (HDFS_v1, HDFS_v2 and HDFS v3 Tracebench) were successfully investigated to understand the log structure though, they were not utilized in the final experiments.

4 Directory Structure

The project directory will have the following format. With this layout, logs, processed datasets, model files, and evaluation outputs are properly arranged and can be accessed easily in order to be reproducible.

```
LOG_DETECT/
│
├── archive/
│   ├── HDFS_2k/
│   │   ├── HDFS_2k.log
│   │   ├── HDFS_2k_log_structured.csv
│   │   └── HDFS_2k_log_templates.csv
│   │
│   ├── HDFS_v1/
│   │   ├── preprocessed/
│   │   ├── HDFS.log
│   │   └── README.md
│   │
```

```
│   ├── HDFS_v2/
│   │   ├── node_logs/
│   │   └── README.md
│   │
│   ├── HDFS_v3_TraceBench/
│   │   ├── preprocessed/
│   │   ├── tracebench/
│   │   └── README.md
│
├── experimenty.ipynb
├── best_lstm.keras
├── block_info_filtered.csv
├── block_info_sample.csv
├── early_warning_comprehensive_eval.png
└── early_warning_comprehensive_eval_lstn.png
```

All of the modelling is executed in experimenty.ipynb, and no extra .py files are needed.

5 Code Execution

1. **Activate the virtual environment**
   Open VS Code and launch the integrated terminal.
   Activate MLVE using: Mlven\Scripts\activate

2. **Install the required libraries**
   All dependencies must be installed within the virtual environment as :
   pip install pandas numpy scikit-learn xgboost tensorflow keras matplotlib seaborn tqdm

3. **Open the project notebook**
   In VS Code navigate to project directory and open experimenty.ipynb. The entire project pipeline is presented in the window.

   Make sure that the MLVE environment is selected as Python interpreter on the top right-hand side of VS Code.

4. **Run all notebook cells sequentially**

   Execute each cell in order. The notebook performs:
   - Parsing and cleaning of raw HDFS logs
   - Timestamp and block ID extraction
   - Event template mapping
   - Temporal segmentation Sliding-window.
   - Feature vector construction
   - XGBoost model and LSTM model training.
   - Calculation of PR-AUC, ROC-AUC, recall, precision and lead-time.
   - Generation of evaluation plots

5. **Review the outputs**

   The notebook automatically generates:

- Precision–Recall curves
- ROC curves
- Plots of lead-time distribution.
- Feature and occlusion importance plots
- Saved LSTM model (best_lstm.keras).

6. **Locate exported files**
   Generated plots and model files appear in the project root directory.

6 References

HTTPS://GITHUB.COM/LOGPAI/LOGHUB
HTTPS://WWW.TENSORFLOW.ORG/
HTTPS://WWW.PYTHON.ORG/