National College of Ireland

Configuration Manual

MSc Research Project: CLOUD SECURITY AND COMPLIANCE: EVALUATING LONG TERM DATA COMPLIANCE

MSc in Cyber Security

PALLAVI KARUNAKAR SALIAN

Student ID: 23267381

School of Computing

National College of Ireland

Supervisor: Joel Aleburu

National College of Ireland

National College of Ireland

Project Submission Sheet

Student Name:	Pallavi Karunakar Salian		
Student ID:	23267381		
Programme:	MSc in Cyber Security	**Year:**	2024-25
Module:	Practicum-2		
Lecturer:	Joel Aleburu		
Submission Due Date:	11 Aug 2025		
Project Title:	Configuration Manual		
Word Count:	1061		

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.
<u>ALL</u> internet material must be referenced in the references section. Students are encouraged to use the Harvard Referencing Standard supplied by the Library. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action. Students may be required to undergo a viva (oral examination) if there is suspicion about the validity of their submitted work.

Signature:	Pallavi Karunakar Salian
Date:	11 Aug 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS:

1. Please attach a completed copy of this sheet to each project (including multiple copies).
2. Projects should be submitted to your Programme Coordinator.
3. **You must ensure that you retain a HARD COPY of ALL projects**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. Please do not bind projects or place in covers unless specifically requested.
4. You must ensure that all projects are submitted to your Programme Coordinator on or before the required submission date. **Late submissions will incur penalties.**
5. All projects must be submitted and passed in order to successfully complete the year. **Any project/assignment not submitted will be marked as a fail.**

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

AI Acknowledgement Supplement

[Insert Module Name]

[Insert Title of your assignment]

Your Name/Student Number	Course	Date
Pallavi Karunakar Salian	MSc in Cyber Security	11 Aug 2024

This section is a supplement to the main assignment, to be used if AI was used in any capacity in the creation of your assignment; if you have queries about how to do this, please contact your lecturer. For an example of how to fill these sections out, please click here.

AI Acknowledgment

This section acknowledges the AI tools that were utilized in the process of completing this assignment.

Tool Name	Brief Description	Link to tool

Description of AI Usage

This section provides a more detailed description of how the AI tools were used in the assignment. It includes information about the prompts given to the AI tool, the responses received, and how these responses were utilized or modified in the assignment. **One table should be used for each tool used**.

[Insert Tool Name]	
[Insert Description of use]	
[Insert Sample prompt]	[Insert Sample response]

Evidence of AI Usage

This section includes evidence of significant prompts and responses used or generated through the AI tool. It should provide a clear understanding of the extent to which the AI tool was used in the assignment. Evidence may be attached via screenshots or text.

Additional Evidence:

[Place evidence here]

Additional Evidence:

[Place evidence here]

CLOUD SECURITY AND COMPLIANCE: EVALUATING LONG TERM DATA COMPLIANCE
CONFIGURATION MANUAL

Tools and Technologies Used

Category	Tools and Servies Used
Programming Language	Python
Data Processing	pandas, numpy
Visualization	matplotlib, seaborn
Preprocessing	LabelEncoder, StandardScaler
ML Models	AdaBoost, Random Forest, SGDClassifier
Evaluation Metrics	accuracy_score, classification_report, F1, etc.
Imbalanced Handling	SMOTE (from imblearn)
Outlier Removal	IQR-based custom filtering
Utilities	warnings, train_test_split
Dataset Format	CSV file (read using pandas)

BEGIN

1. Setup
Import necessary libraries
– pandas, numpy
– seaborn, matplotlib.pyplot
– scikit-learn modules (model_selection, preprocessing, ensemble, linear_model, metrics)
– imblearn.over_sampling.SMOTE
Suppress warnings

2. Load and Inspect Data
data ← read CSV "final(2).csv"
Display first few rows
Display data info() and memory usage
Check for missing values and duplicates

3. Explore Categorical Features
categorical_cols ← columns with dtype == object
FOR each col IN categorical_cols:
Print unique values of cols

```
# 4. Outlier Removal (IQR method)
numeric_cols ← columns with numeric types
Q1, Q3 ← compute 25th and 75th percentiles of numeric_cols
IQR ← Q3 – Q1
outlier_mask ← any value < Q1 – 1.5*IQR OR > Q3 + 1.5*IQR
data_cleaned ← data where no row has outlier_mask == True

# 5. Visualize Target Distribution
Plot countplot of data_cleaned["Prediction"] with annotations

# 6. Encode Categorical Variables
label_encoder ← new LabelEncoder()
FOR each col IN ['Protcol','Flag','Family','SeddAddress','ExpAddress','Threats','IPaddress','Port','Prediction']:
data_cleaned[col] ← label_encoder.fit_transform(data_cleaned[col])

# 7. Correlation Analysis
Compute corr_matrix of features (excluding target)
Plot heatmap of corr_matrix with annotations

# 8. Prepare Training and Test Sets
X ← data_cleaned without 'Prediction'
y ← data_cleaned['Prediction']
X_train, X_test, y_train, y_test ← train_test_split(X, y, test_size=0.2, random_state=42)

# 9. Handle Class Imbalance
smote ← new SMOTE(random_state=42)
X_train_res, y_train_res ← smote.fit_resample(X_train, y_train)

# 10. Feature Scaling
scaler ← new StandardScaler()
X_train_scaled ← scaler.fit_transform(X_train_res)
X_test_scaled ← scaler.transform(X_test)

# 11. Train and Evaluate Models
Define models:
– ada_model ← AdaBoostClassifier(n_estimators=50, random_state=42)
– rf_model ← RandomForestClassifier(n_estimators=100, random_state=42)
– sgd_model ← SGDClassifier(max_iter=1000, tol=1e-3, random_state=42)

FOR each model IN [ada_model, rf_model, sgd_model]:
model. Fit(X_train_scaled, y_train_res)
y_pred ← model. Predict(X_test_scaled)
Print classification_report(y_test, y_pred)
Plot confusion_matrix(y_test, y_pred)
Compute accuracy, precision, recall, f1_score

# 12. Compare Models
Build comparison table with metrics for each model
Identify best model by highest Accuracy, Precision, Recall, F1

END
```

Explanation of the pseudo-code

The pseudo-code lays out a standard supervised-learning pipeline in twelve logical phases:

Setup
Starts by importing all required libraries (data handling, visualization, modeling, metrics, and SMOTE for imbalance) and silencing irrelevant warnings.

Data Loading and Inspection
Reads in the CSV, displays a data snapshot, checks schema and memory, and verifies there are no missing or duplicate records.

Categorical Feature Exploration
Identifies object-typed columns and prints their unique values so you can understand all levels before encoding.

Outlier Removal
Uses the interquartile-range (IQR) method on every numeric field to flag and drop rows with extreme values, helping prevent skewed learning.

Target Distribution Visualization
Draws a bar chart of the encoded target ("Prediction") classes to reveal any imbalance that may hurt model performance.

Label Encoding
Transforms all categorical columns (including ports and addresses) into integer labels so that tree-based and linear models can process them.

Correlation Analysis
Computers and plots a heatmap of feature correlations (excluding the target) to spot redundant variables or strong predictors.

Train/Test Split
Randomly allocates 80 percent of the data for training and 20 percent for testing to assess generalization.

Imbalance Handling with SMOTE
Oversamples the minority classes in the training set to produce a balanced learning sample.

Feature Scaling
Standardizes numeric features so models that assume normal inputs (like SGD) behave well.

Model Training and Testing
Trains AdaBoost, Random Forest and SGD classifiers. On the test set, it predicts on each and prints precision/recall/F1 then draws a confusion matrix.

Model Comparison
Tabulates accuracy, precision, recall, and F1 for all models in a DataFrame, then selects the top performer (Random Forest) by each metric.

REFERENCES:

Kaggle (2023). Generative mathematical models for ransomware attack prediction using Chi-Square feature selection for enhanced accuracy. *Signal, Image and Video Processing*, [online] 19(10). doi:https://doi.org/10.1007/s11760-025-04396-x.

Attou, H., Guezzaz, A., Benkirane, S. and Azrour, M. (2025). Cloud Security Enhancement Through RBFNN and AdaBoost-Based Model. [online] doi:https://doi.org/10.13140/RG.2.2.12954.96965.

Matharaarachchi, S., Domaratzki, M. and Muthukumarana, S. (2024). Enhancing SMOTE for imbalanced data with abnormal minority instances. *Machine Learning with Applications*, 18, p.100597. doi:https://doi.org/10.1016/j.mlwa.2024.100597.

Rathna, S. (2024). Cloud-Centric Detection Framework for Ransomware and Zero-Day Attacks Using LSTM Networks. *International Journal of Engineering Research and Science & Technology*, 20(3). doi:https://www.researchgate.net/profile/Rathna-

Suruli-Andavar/publication/392760250_Cloud-Centric_Detection_Framework_for_Ransomware_and_Zero-Day_Attacks_Using_LSTM_Networks/links/6851720c24267473b7785e19/Cloud-Centric-Detection-Framework-for-Ransomware-and-Zero-Day-Attacks-Using-LSTM-Networks.pdf.