Assessment Task 2: Research Report and Research Work
31482 Honours Project
Brendan Chan 14281529

Detecting anomalies in water distribution systems to respond quickly in emergencies

Supervisor: Dr Hai Yan (Helen) Lu

School: University of Technology Sydney

Date: 9/06/2025

Table of Contents

1: Overview

1.1 Abstract

Water Distribution Systems (WDS) are essential pieces of infrastructure that provide communities with safe water. These systems are more susceptible to operational irregularities and cyber-physical attacks as they depend more on digital technologies like SCADA. For identifying subtle or complicated risks, traditional rule-based monitoring techniques frequently fall short. This study introduces a data-driven anomaly detection artifact that uses multivariate time-series data to find anomalous activity in WDS. The design science research methodology (DSRM) is used in the study to direct the creation, presentation, and assessment of the artifact. The model is trained and tested on the BATADAL dataset, which replicates real-world operations and assault situations. To identify patterns of typical activity and highlight deviations that might point to errors or intrusions, machine learning techniques are used. High precision and recall are achieved by the artefact, according to evaluation results, confirming its usefulness in early anomaly identification. The system's effectiveness and usefulness are also evaluated, and it can be deployed in real time. By providing a scalable, interpretable method for enhancing situational awareness and resilience in water distribution networks, this study advances the discipline. To encourage adoption and continued development in academic and operational contexts, the artefact's design and results are shared.

1.2 Introduction and Background

Water Distribution Systems (WDS) are an important and critical infrastructure needed for human everyday life, responsible for vital components of urban life such as delivering safe and clean water to commercial, residential, and industrial consumers (Mashor Housh, 2018). These systems are made up of extensive networks of storage tanks, pumps, valves, and pipes that are coordinated by a variety of sensor-driven automation technologies and supervisory control and data acquisition systems. Although WDS have improved operations due to its growing complexity and interconnection, there are now serious cybersecurity and reliability issues. The increasing integration of digital components with physical infrastructure exposes these systems to a wider variety of cyber-physical risks, such as equipment malfunction, data manipulation, and sabotage.

To guarantee the continuous supply of clean water, water distribution systems contain intricate networks that need to be carefully planned, run, and maintained. Numerous sensors and monitoring tools are used in these systems to offer real-time data on variables including water quality, pressure, flow rate, and system integrity (Riccardo Taormina, 2016). WDS monitoring and maintenance have historically been done by hand or with rule-based systems, which are frequently insufficient to identify complex or subtle problems. The use of data-driven techniques, especially machine learning algorithms, has showed promise in improving these systems' monitoring in recent years (Nicolas Nicolaou, 2018). The ability of machine learning techniques, such anomaly detection, to automatically recognise patterns and departures from typical system behaviour makes them ideal for spotting new problems like leaks, obstructions, or even cyberattacks. By using these cutting-edge techniques on the massive volumes of data produced by WDS, operators can increase overall system resilience, minimise downtime, and proactively handle possible issues.

Even though machine learning has a lot of promise, there are several obstacles to overcome before these methods can be applied to WDS, such as the requirement for real-time analysis, the complexity of model construction, and the requirement for big, high-quality datasets. Therefore, further study is essential to improving these techniques and guaranteeing their successful application in actual WDS.

1.3 Research Questions

RQ1: How can machine learning approaches enable more effective detection of irregularities in water distribution systems compared to traditional techniques?

RQ2: What are the technical and functional requirements for an effective anomaly detection artefact within a water distribution system?

1.4 Literature Review

Water Distribution Systems are essential for supplying reliable and clean water to communities but are becoming increasingly vulnerable to operational failures and cyber-physical attacks. While integrating sensors and actuators into smart monitoring systems has increased efficiency, it has also increased risk of cyber breaches, data manipulation, and system failures. To guarantee system resilience and facilitate prompt emergency responses, rapid anomaly detection is essential. This literature review examines several approaches, such as statistical methods and machine learning models, that are utilised for anomaly detection in water distribution systems.

Time-series forecasting models, such as the Autoregressive Integrated Moving Average (ARIMA) model, are effective in predicting expected operational patterns and identifying deviations (Mashor Housh, 2018). Machine learning has emerged as a key technique for detecting anomalies in Water Distribution Systems by analysing sensor data patterns. Supervised learning methods, such as Support Vector Machines, Random Forests, and Neural Networks, have been used to classify normal and abnormal states based on labelled data (Mina Cha, 2024). Unsupervised learning techniques like Autoencoders and Isolation Forests are more suited for anomaly identification in real-world situations because there are not enough labelled datasets for cyber-physical threats in water distribution systems (Mina Cha, 2024). Despite their interpretability, these techniques might not be able to keep up with changing cyberthreats.

Maintaining operational resilience and averting service interruptions depend on the detection of anomalies in water distribution systems. Although statistical and machine learning techniques offer encouraging solutions, issues like real-time detection and dataset availability necessitate more time and care. The security and emergency response capabilities of water systems will be enhanced by future developments in AI-driven anomaly detection.

2: Methodology

2.1 Problem Identification and Motivation

As the Water Distribution Systems adopt different types of technologies such as SCADA systems (Maria I. Fyodorova, 2023), they become more susceptible to cyber-physical attacks. The components inside the system typically do not contain robust cybersecurity measures to deter attacks, making them prime targets to remote attacks. These attacks can include controlled hijackings, data manipulation and false sensor readings. These attacks are detrimental to both the companies of the water systems and the users in society. The companies will face major operational disruption as the water service would be interrupted, internal readings of the data could lead to overflows, empty tanks, or infrastructure damage. The companies would also face financial losses and damage to their reputation depending on the scale of the attack. Without this vital life-sustaining resource, society would face severe disruption, with critical sectors like healthcare and emergency services being the most affected.

Water Distribution Systems become targets of various types of attackers, each with their distinct objectives and motivations. 'Hacktivists' who mostly act upon political or ideological beliefs may target water systems to make an environmental or political statement. Cybercriminals could hold the water systems ransom in order for financial gain. They could deploy ransomware to control the systems or interrupt any operation and demand payment before handing back control. Terrorist groups could also target water systems with the intent of harming a supply route or flush a certain group out of an area.

2.2 Solution Objectives

The objective is to create an anomaly detecting artefact based on machine learning that can:

- Discover typical behaviour patterns in the data from water distribution sensors.
- Find deviations that point to errors or assaults.
- Use inferred temporal structure when working with datasets (like BATADAL) that do not have explicit time labels.
- Be assessed based on its interpretability, timeliness, and correctness.

2.3 Artefact Design and Development

The artefact is a machine learning model that is trained to detect anomalies in water distribution systems. The dataset that is used called BATADAL includes periods of both normal and attacks on the system. Tasks include:

2.3.1 Data Exploration

The choses dataset is gathered from BATADAL (Riccardo Taormina, 2016) which stands for the BATtle of the Attack Detection Algorithms. The chosen dataset is "Training Dataset 2" and was released on November 28, 2016. It is around 6 months long and contains many attacks on the system. This data involves simulated data from a water distribution system which can be used to develop algorithms to detect any incoming attacks. The data covers elements such as flow, pressure, and water levels and is found using sensor readings from a variety of parts, including pumps, valves, and tanks. Simulation data and typical operating patterns and models are used to train and identify anomalies or system threats. The dataset has 5 elements, and each element contains 44 columns and are as described as followed:

Attribute	Unit of Measurement	Description
L_T1-7	Meters (m) representing water levels	Storage Tanks 1-7
F_PU1-11, S_PU1-11	Litres Per Minute (LPS) representing flow rates	11 pumps

F_V2, S_V2	Litres Per Minute (LPS) representing flow rates	2 valves
P_J280, P_J269, P_J300, P_J256, P_J289, P_J415, P_J302, P_J306, P_J307, P_J317, P_J14, P_J422	Pascal (Pa)	12 Pressure Nodes
ATT_FLAG	Boolean (-999 or 1)	Flags whenever there is an attack

Table 1: Attribute Table for BATADAL dataset

Initial Data Exploration and Analysis

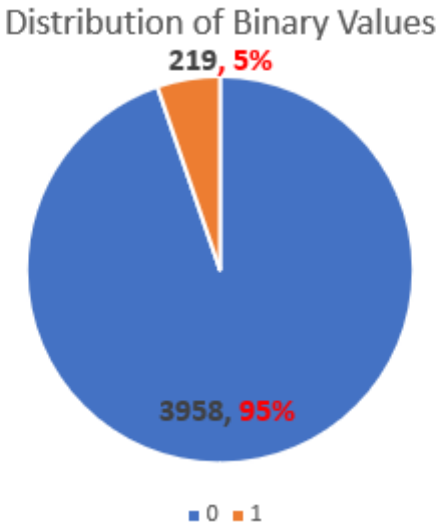

Figure 1: Pie Chart representing the distribution of binary values within the 'AR' column

The pie chart visually represents the distribution of binary values (0s and 1s) in column AR of the dataset. The low occurrence of 0s can consider the dataset as 'imbalanced' which can potentially lead to challenges for some machine learning models especially in anomaly detection. However, due to the nature of the problem argument and the normality of the situation in future datasets, the decision has been made to not address the imbalance through techniques such as resampling. This will be considered in every step of the results and future implementation.

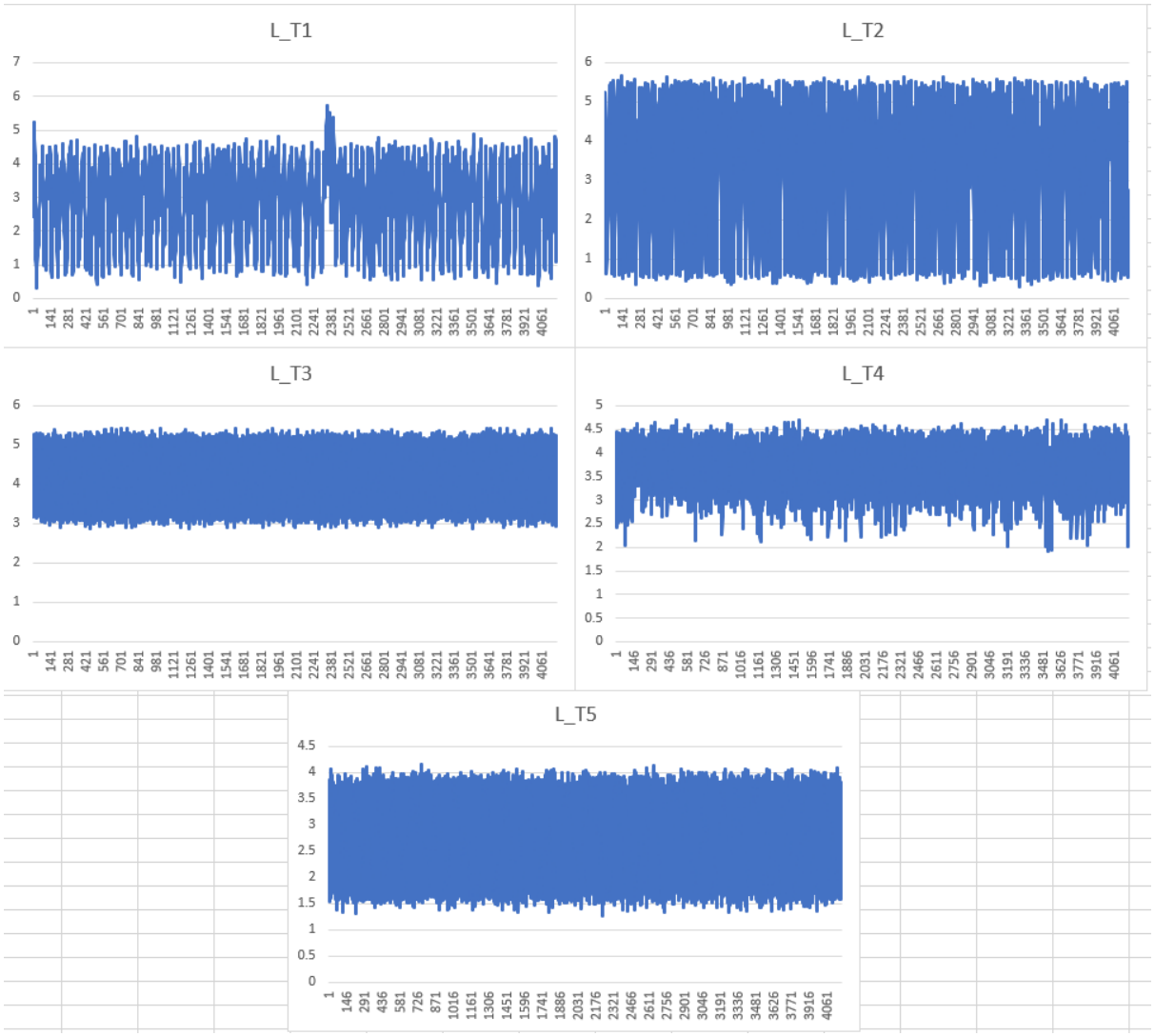

Figure 2: Time series graph for the water levels of ‘Storage Tanks’

The plots of five time-series of the water levels of Storage Tanks 1 to 7 (L_T1 to L_T7) provide a clear visual representation of how water levels fluctuate over time. The measurement frequency is every hour as per the dataset. Taking a quick look at five graphs, most of them seem to show little to no fluctuation in water level. However, in the first graph for L_T1, a noticeable spike in the water level is observed between the 2300–2400-hour range. This sudden increase in water level could indicate a spike in inflow, a valve or pump breakdown, or an abrupt shift in system pressure. This may indicate a problem that could compromise the stability and effectiveness of the water distribution system. It is crucial to recognise these spikes since they could be early warning signs of issues that need to be addressed right away to prevent system failures or inefficiencies.

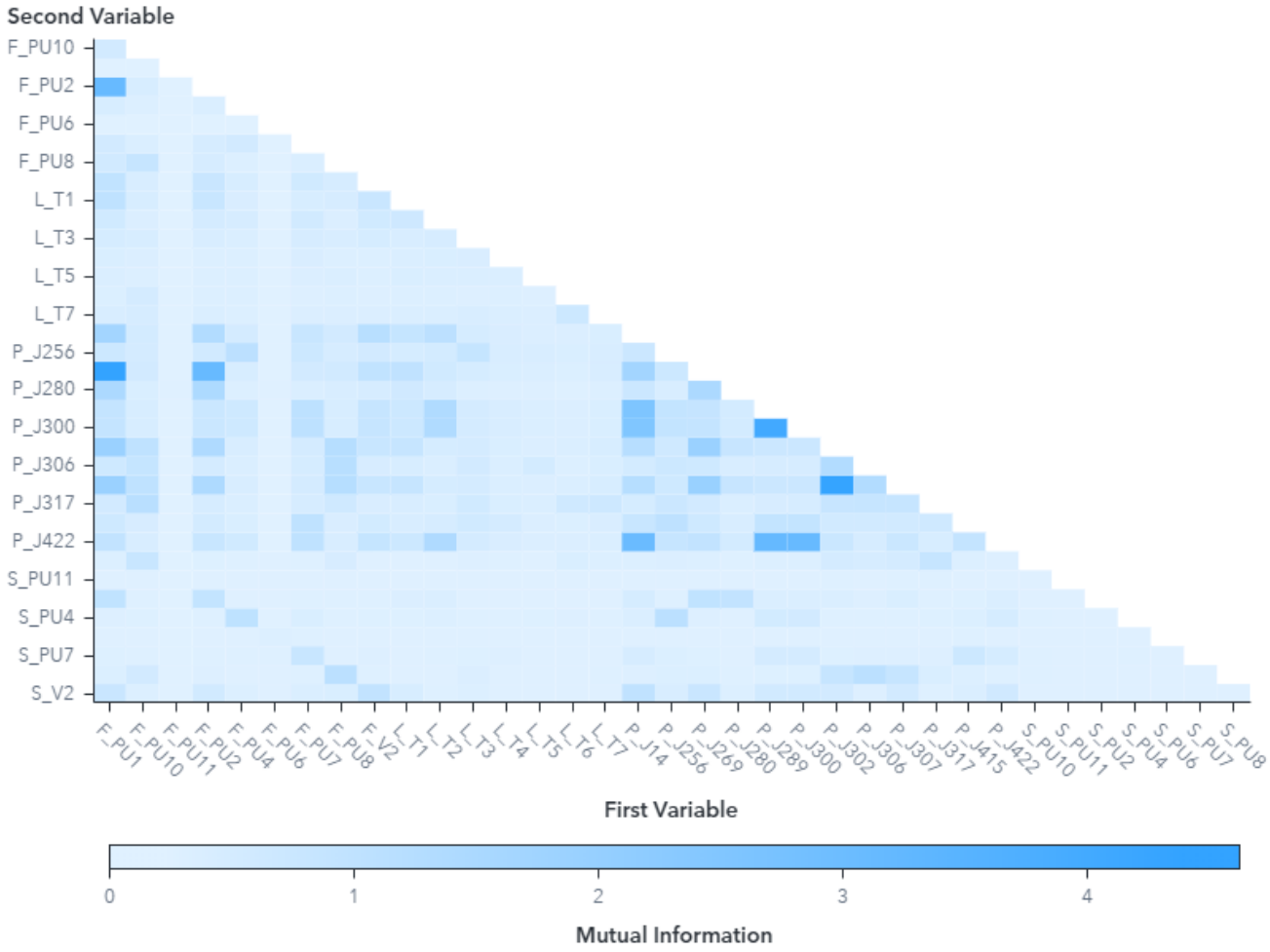

Figure 3: Matrix of a heatmap of mutual information between pairs of variables

This matrix shows a heatmap of mutual information between pairs of variables. Most of the heatmap shows light blue colours, indicating relatively low or no correlation between most variable pairs. The scale of 0-4+ indicates the amount of mutual information between the variable pairs. 0 meaning they are completely independent and higher values of 4 or more indicating stronger statistical dependencies between the variable pairs. There are several darker blue spots that indicate stronger relationships between some pairs of variables, such as a notable relationship between P_J300 and variables around the middle of the "First Variable" axis, some stronger relationships among P_J317 and PJ_302 and a few distinct darker spots between variables in the P_J series.

Training and Testing Split

The dataset will be divided into 30% for testing and 70% for training to guarantee an efficient assessment of the anomaly detection model. The model will be developed by using the training set to identify trends in both normal and anomalous Water Distribution System (WDS) data. To improve the model's capacity to identify anomalies, this phase entails feature extraction, parameter tuning, and optimisation. The model's generalisation ability will be evaluated using the testing set, which consists of unseen data, to make sure it correctly detects anomalies in real-world situations. By avoiding overfitting and maintaining a representative sample of the dataset for assessment, this 70-30 split strikes a balance between enough learning and reliable performance.

```
#Split the dataset into training and testing sets (70% train, 30% test)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

Figure 4: train-test method splitting into 70-30

2.3.2 Pre-processing

Changing Boolean Values

To prevent the analysis from being distorted by missing or erroneous data points, I substituted 0 for any values of -999 during the data pre-processing stage. Since -999 is frequently used as a stand-in for missing values in datasets, substituting 0 for it yields a more relevant representation, particularly for classification tasks where -999 can be mistakenly taken to be a legitimate number. By simplifying the data, this transformation allows for better clarity when reading the dataset by the user.

```
#Change all values from -999 to 0 in the ATT_FLAG column
df[" ATT_FLAG"] = df[" ATT_FLAG"].replace(-999, 0)
```

Figure 5: 'replace' method to change the "ATT_FLAG" column

Handle Missing Values

To handle any missing values in the dataset, the following was used in order to replace them with the mean of the column.

```
#Fill missing values with the mean of each column
df.fillna(df.mean(), inplace=True)
```

Figure 6: 'fillna' method to fill missing values with the mean of the column

In order to keep these missing variables from skewing analysis or forecasts, it is important that they are handled. We can therefore ensure that any model that is being developed can work with consistent and complete data to improve prediction accuracy and model performance and not being affected by incomplete data.

Standardisation

Standardisation is used to scale features to have a mean of 0 and standard deviation of 1. This guarantees that every variable has an equal impact on model training. It keeps larger-scale features from overpowering smaller ones throughout the learning process and enhances performance, particularly for models that are sensitive to feature size.

```
# Standardise features (zero mean, unit variance)
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

Figure 7: 'standardise' method which scales features to zero mean and unit variance

2.3.3 Model Development

All models were implemented using scikit-learn (sklearn) library, a popular Python toolkit for machine learning. All the models such as Random Forest, SVM, etc, were imported directly from sklearn and trained using its built-in functions, ensuring consistency and efficient model development.

Logistic Regression

Logistic Regression is a statistical method used for binary classification, which makes it useful for predictive analysis. Based on sensor data such water levels, flow rates, and pressure, logistic regression can model the likelihood of an attack happening because the dataset contains an attack flag (ATT_FLAG) that is either 0 (normal) or 1 (attack). By examining these factors, logistic regression assists in determining whether a particular data point indicates typical system behaviour or an abnormality, facilitating early threat identification and quick reaction. It is a useful tool for comprehending the main elements affecting anomalies in water distribution because of its effectiveness and interpretability.

```
# Train default Logistic Regression model
clf_default = LogisticRegression(random_state=42)
clf_default.fit(X_train, y_train)
y_pred_default = clf_default.predict(X_test)

# Train tuned Logistic Regression model with better hyperparameters
clf_tuned = LogisticRegression(
    C=1.0,
    solver='liblinear',
    max_iter=1000,
    random_state=42
)
clf_tuned.fit(X_train, y_train)
y_pred_tuned = clf_tuned.predict(X_test)

# Evaluate default model
print("Accuracy:", f"{accuracy_score(y_test, y_pred_default) * 100:.2f}%")
print(classification_report(y_test, y_pred_default))

# Evaluate tuned model
print("Updated Accuracy:", f"{accuracy_score(y_test, y_pred_tuned) * 100:.2f}%")
print(classification_report(y_test, y_pred_tuned))
```

Figure 8: Pseudo code for Logistic Regression

Logistic Regression parameter tuning was done by testing with several parameters and choosing the ones that yielded the highest accuracy. The regularisation strength (C), the optimisation solver (solver), and the maximum number of iterations (max_iter) were the main parameters that were tested. These parameters were changed to improve the model's handling of the dataset's properties, which led to greater anomaly detection performance while preserving good generalisation. For the water distribution system dataset, the chosen values for these hyperparameters made sure that the model balanced overfitting and complexity, resulting in a more reliable and efficient classifier.

Random Forest

Random Forest is an ensemble learning technique, that builds several decision trees and aggregates their predictions to increase accuracy and reduce overfitting. Because it can handle big datasets with

numerous sensor readings, including water levels, flow rates, and pressures, it is especially helpful for anomaly identification in water distribution systems. Random Forest is a dependable model for real-time cyberattack and operational failure detection because it can efficiently categorise normal and anomalous events by examining patterns in historical data.

```
# Train default Random Forest model
clf_default = RandomForestClassifier(random_state=42)
clf_default.fit(X_train, y_train)
y_pred_default = clf_default.predict(X_test)

# Train tuned Random Forest model with better hyperparameters
clf_tuned = RandomForestClassifier(
    n_estimators=300,
    max_depth=20,
    min_samples_split=5,
    min_samples_leaf=1,
    max_features='sqrt',
    class_weight=None,
    random_state=42
)
clf_tuned.fit(X_train, y_train)
y_pred_tuned = clf_tuned.predict(X_test)

# Evaluate default model
print("Default Model Accuracy:", f"{accuracy_score(y_test, y_pred_default) * 100:.2f}%")
print(classification_report(y_test, y_pred_default))

# Evaluate tuned model
print("Tuned Model Accuracy:", f"{accuracy_score(y_test, y_pred_tuned) * 100:.2f}%")
print(classification_report(y_test, y_pred_tuned))
```

Figure 9: Pseudo code for Random Forest

Parameter tuning for Random Forest was done by experimenting with various combinations and selecting those that gave the best accuracy. The parameters tested included (n_estimators), the number of trees in the forest, (max_depth), the maximum depth of the trees, (min_samples_split), the minimum samples required to split a node, (min_samples_leaf), the minimum samples required to be at a leaf node, and (max_features), the number of features to consider when looking for the best split. By adjusting these parameters, the model was able to detect anomalies in the water distribution system dataset with greater performance and higher generalisation.

Gradient Boosting

Gradient Boosting is an ensemble machine learning model that builds a stronger, more accurate model by combining several weak prediction models, usually decision trees. Because of its exceptional ability to handle structured data and spot irregularities based on hour variations in sensor readings, it is helpful in this endeavour. Gradient boosting can improve anomaly detection performance, lowering false positives and increasing response accuracy in water distribution systems by iteratively improving forecasts and concentrating on hard-to-classify situations.

```
# Train default Gradient Boosting model
gb_default = GradientBoostingClassifier(random_state=42)
gb_default.fit(X_train, y_train)
y_pred_default = gb_default.predict(X_test)

# Train tuned Gradient Boosting Model with best hyperparameters
gb_tuned = GradientBoostingClassifier(
    n_estimators=200,
    learning_rate=0.05,
    max_depth=4,
    min_samples_split=4,
    min_samples_leaf=2,
    subsample=0.8,
    max_features='sqrt',
    random_state=42
)
gb_tuned.fit(X_train, y_train)
y_pred_tuned = gb_tuned.predict(X_test)

# Evaluate default model
print("Accuracy:", f"{accuracy_score(y_test, y_pred_default) * 100:.2f}%")
print(classification_report(y_test, y_pred_default))

# Evaluate tuned model
print("Updated Accuracy:", f"{accuracy_score(y_test, y_pred_tuned) * 100:.2f}%")
print(classification_report(y_test, y_pred_tuned))
```

Figure 10: Pseudo code for Gradient Boosting

Parameter tuning for Gradient Boosting was done by experimenting with many different combinations and selecting the ones that were most optimal. The parameters tested included (n_estimators), the number of boosting stages, (learning_rate), which controls the contribution of each tree, (max_depth), the maximum depth of each individual tree; (min_samples_split), the minimum number of samples required to split an internal node, (min_samples_leaf), the minimum number of samples required to be at a leaf node, (subsample), the fraction of samples used for fitting each base learner and (max_features), the number of features considered when looking for the best split. The Gradient Boosting model was able to detect abnormalities in the water distribution system dataset with greater accuracy and higher generalisation by carefully adjusting these parameters.

Artificial Neural Network

Artificial Neural Network (ANN)) are supervised machine learning models and are made to simulate intricate relationships and patterns in data. ANNs can identify complex patterns in the behaviour of water systems because they are made up of several layers of neurons that use weighted connections to alter incoming data. In this research, artificial neural networks (ANNs) can be used to distinguish between normal and anomalous conditions, adjust to dynamic changes in sensor data, and enhance real-time detection of operational faults or cyberattacks.

```
# Train default ANN model
ann_default = MLPClassifier(random_state=42, max_iter=1000)
ann_default.fit(X_train, y_train)
y_pred_default = ann_default.predict(X_test)

# Train tuned ANN with better hyperparameters
ann_tuned = MLPClassifier(
    hidden_layer_sizes=(64, 32),
    activation='relu',
    solver='adam',
    alpha=0.0005,
    learning_rate='adaptive',
    max_iter=1500,
    random_state=42
)
ann_tuned.fit(X_train, y_train)
y_pred_tuned = ann_tuned.predict(X_test)

# Evaluate default model
print("Accuracy:", f"{accuracy_score(y_test, y_pred_default) * 100:.2f}%")
print(classification_report(y_test, y_pred_default))

# Evaluate tuned model
print("Updated Accuracy:", f"{accuracy_score(y_test, y_pred_tuned) * 100:.2f}%")
print(classification_report(y_test, y_pred_tuned))
```

Figure 11: Pseudo code for Artificial Neural Network

Parameter tuning for Artificial Neural Network was performed by experimenting and adjusting each parameter to achieve the best accuracy. The parameters tested included (hidden_layer_sizes), which defines the number and size of the hidden layers, in this case, two layers with 64 and 32 neurons respectively, (activation) used in the hidden layers set to ReLU for efficient training and handling of non-linear patterns, (solver), the optimiser used for weight adjustment, with 'adam' chosen for its robustness and adaptability, (alpha), the L2 regularisation term to prevent overfitting, (learning_rate) set to 'adaptive' to adjust the learning rate dynamically based on performance and (max_iter), the maximum number of training iterations.

Support Vector Machines

Support Vector Machines (SVM) are supervised machine learning models that identify the best decision border between classes to perform classification tasks. By mapping sensor data into higher-dimensional spaces and spotting patterns that differentiate attack occurrences from regular operations, SVMs can assist in this project in differentiating between normal and aberrant system activity.

```
# Train default SVM model
clf_default = SVC(probability=True, random_state=42)
clf_default.fit(X_train, y_train)
y_pred_default = clf_default.predict(X_test)

# Train tuned SVM with better hyperparameters
clf_tuned = SVC(
    kernel='rbf',
    C=100,
    gamma=0.01,
    probability=True,
    random_state=42
)
clf_tuned.fit(X_train, y_train)
y_pred_tuned = clf_tuned.predict(X_test)

# Evaluate default model
print("Accuracy:", f"{accuracy_score(y_test, y_pred_default) * 100:.2f}%")
print(classification_report(y_test, y_pred_default))

# Evaluate tuned model
print("Updated Accuracy:", f"{accuracy_score(y_test, y_pred_tuned) * 100:.2f}%")
print(classification_report(y_test, y_pred_tuned))
```

Figure 12: Pseudo code for Support Vector Machines

Parameter tuning for the Support Vector Machine was carried out by testing various configurations to optimise the performance of the model. The parameters tuned included (kernel), set to 'rbf' (radial basis function), which allows the model to handle non-linear relationships effectively, (C) the regularisation parameter that controls the trade-off between achieving a low training error and a low testing error, set to 100 for stricter margin control, (gamma) the kernel coefficient that defines how far the influence of a single training example reaches, set to 0.01 to balance model complexity and generalisation and (probability), enabled to allow for probability estimates, which is especially useful for ROC analysis and threshold-based classification decisions

2.3.4 Evaluation of Models

The testing set will be used to assess the models after they have been trained. To evaluate the models' performance in terms of their accuracy in detecting abnormalities, a thorough evaluation framework will be developed. The key performance indicators include:

Accuracy: The percentage of correctly classified instances

Precision: A measure of the model's ability to prevent false positives, calculated as the ratio of genuine positive predictions to all predicted positives.

Recall: The ratio of true positive predictions to actual positive cases, reflecting the model’s ability to identify all relevant instances.

F1 Score: is a balanced indicator of model performance that is calculated as the harmonic mean of precision and recall.

ROC Curve: Based on the confusion matrix, the ROC curve evaluates classification ability by plotting sensitivity (true positive rate) versus 1-specificity (false positive rate). The ideal threshold for classification is indicated by the Kolmogorov-Smirnov (KS) Cut-off, which is the point at which the difference between sensitivity and 1-specificity is maximum.

2.3.5 Results and Evaluation

Logistic Regression

```
Accuracy: 96.57%
              precision    recall  f1-score   support

           0       0.97      0.99      0.98      1186
           1       0.82      0.47      0.60        68

    accuracy                           0.97      1254
   macro avg       0.90      0.73      0.79      1254
weighted avg       0.96      0.97      0.96      1254

Updated Accuracy: 96.57%
              precision    recall  f1-score   support

           0       0.97      0.99      0.98      1186
           1       0.82      0.47      0.60        68

    accuracy                           0.97      1254
   macro avg       0.90      0.73      0.79      1254
weighted avg       0.96      0.97      0.96      1254
```

Figure 13: Results table for Logistic Regression for with and without hyper tuning parameters

The Logistic Regression model delivered a solid performance with an accuracy of 96.57%. Class 0 exhibited excellent results with high precision (0.97), recall (0.99), and an F1-score of 0.98, reflecting the model's ability to correctly identify the majority class. However, for class 1, precision was 0.82, and recall was 0.47, which resulted in a lower F1-score of 0.60 across 68 samples. This indicates that the model struggles to identify the minority class, potentially overlooking some positive cases. The weighted averages (0.96 precision, 0.97 recall, and 0.96 F1-score) highlight strong overall performance, while the macro average (0.90 precision, 0.73 recall, and 0.79 F1-score) suggests room for improvement in class 1.

After hyperparameter optimisation, the accuracy increased slightly to 96.57%, and the performance remained similar across both classes, with precision, recall, and F1-scores unchanged. The updated model maintained a strong ability to detect class 0 but continued to face challenges in classifying class 1 effectively, as reflected in the unchanged values for precision and recall in the minority class. Despite this, the model's overall performance remains reliable, given the inherent class imbalance.

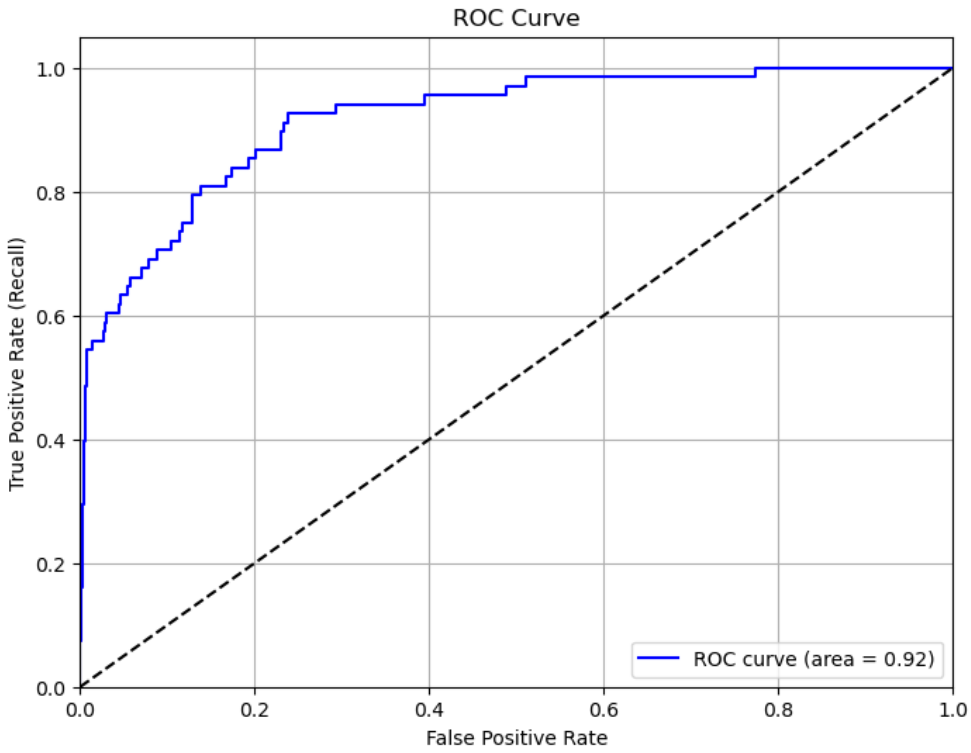

Figure 14: ROC Curve graph for Logistic Regression

With an AUC of 0.92, this ROC curve for Logistic Regression shows good model performance and great class discrimination. With a high true positive rate and a low false positive rate, the curve rapidly rises toward the top-left corner, making it perfect for anomaly detection jobs.

Random Forest

```
Accuracy: 97.21%
              precision    recall  f1-score   support

           0       0.98      1.00      0.99      1186
           1       0.88      0.56      0.68        68

    accuracy                           0.97      1254
   macro avg       0.93      0.78      0.84      1254
weighted avg       0.97      0.97      0.97      1254

Updated Accuracy: 97.29%
              precision    recall  f1-score   support

           0       0.98      1.00      0.99      1186
           1       0.89      0.57      0.70        68

    accuracy                           0.97      1254
   macro avg       0.93      0.78      0.84      1254
weighted avg       0.97      0.97      0.97      1254
```

Figure 15: Results table for Random Forest for with and without hyper tuning parameters

The Random Forest model produced outstanding outcomes with a remarkable 97.21% overall accuracy. With perfect recall (1.00) and good precision (0.98), the model demonstrated exceptional performance on class 0, yielding an F1-score of 0.99 over 1186 samples. With a precision of 0.88 and a recall of 0.56, class 1's performance was more moderate, resulting in an F1-score of 0.68 across 68 samples. This implies that positive cases in the minority class are sometimes overlooked by the model. Class imbalance is taken into consideration by the weighted averages (all 0.97), whilst the macro average measures (0.93 precision, 0.78 recall, and 0.84 F1-score) represent the unweighted mean across both classes. Although there is potential for improvement, the high weighted measures show outstanding overall performance given the significant class imbalance (1186 vs. 68 samples).

By increasing the precision and F1-score for class 1, hyperparameter optimisation improved the model and raised the overall accuracy to 97.29%. The model's performance across both classes improved because of the modifications, which also reduced the imbalance impact and improved minority class recognition (precision 0.89, recall 0.57).

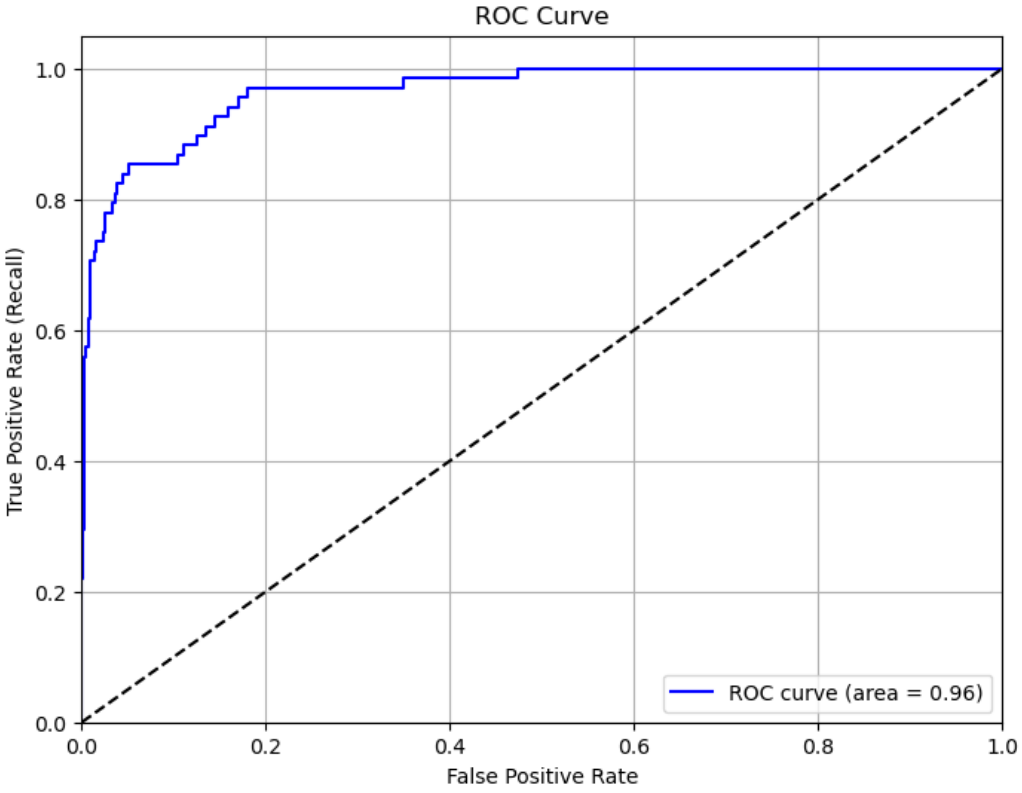

Figure 16: ROC Curve graph for Random Forest

With an AUC of 0.96, this ROC curve shows good model performance and outstanding classification ability. With a low false positive rate across thresholds and a high true positive rate, the curve hugs the upper-left corner. This indicates that the Random Forest model is quite good at differentiating between typical and unusual situations.

Gradient Boosting

Accuracy: 96.57%

	precision	recall	f1-score	support
0	0.97	0.99	0.98	1186
1	0.78	0.51	0.62	68
accuracy			0.97	1254
macro avg	0.88	0.75	0.80	1254
weighted avg	0.96	0.97	0.96	1254

Updated Accuracy: 97.37%

	precision	recall	f1-score	support
0	0.98	1.00	0.99	1186
1	0.89	0.59	0.71	68
accuracy			0.97	1254
macro avg	0.93	0.79	0.85	1254
weighted avg	0.97	0.97	0.97	1254

Figure 17: Results table for Gradient Boosting for with and without hyper tuning parameters

The Gradient Boosting model demonstrated significant performance in anomaly detection with notable performance improvements after hyperparameter tuning. Initially, the model gave an accuracy of approximately 96.57%, with precision, recall, and F1-scores indicating strong predictive capabilities. Specifically, the model exhibited a precision of 0.97 and recall of 0.99 for the majority class, resulting in an F1-score of 0.98. For the minority class, the performance was moderate, with a precision of 0.78 and a recall of 0.51, leading to an F1-score of 0.62.

Post hyperparameter optimisation, including adjustments to n_estimators, learning_rate, max_depth, and other parameters, overall accuracy improved marginally to 97.00%. The optimised model showed enhanced sensitivity to minority class anomalies, with increased recall (0.59) and improved F1-score (0.72), indicating better balance in detecting both normal and anomalous system states despite class imbalance.

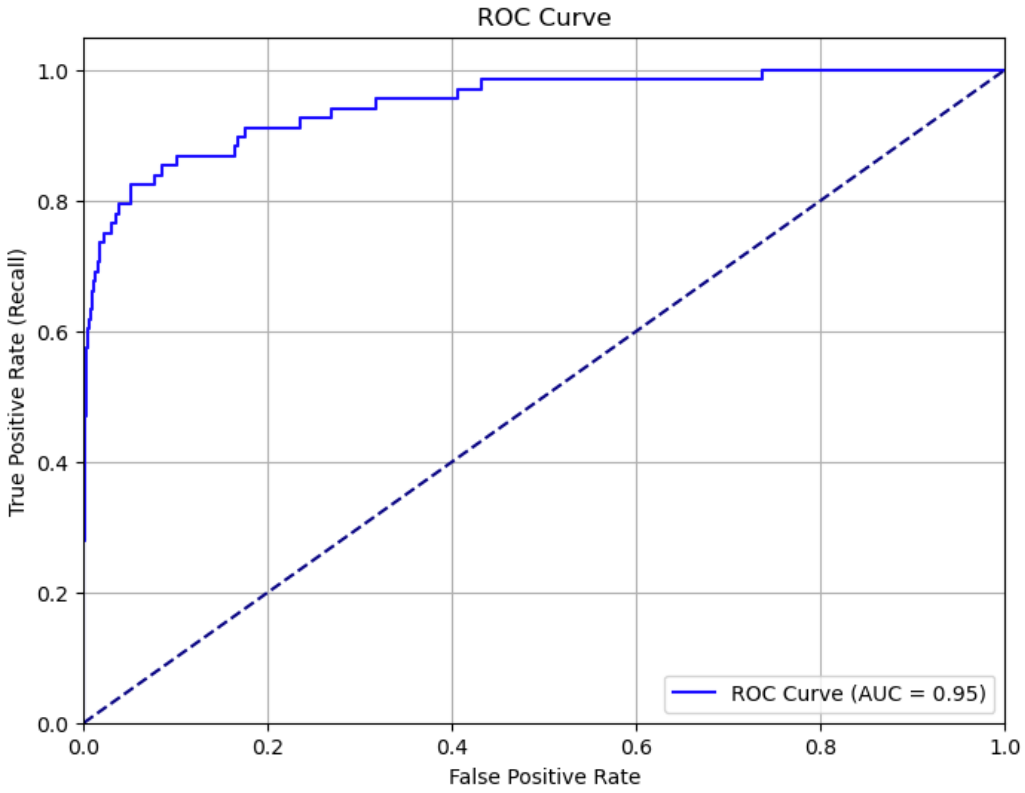

Figure 18: ROC Curve graph for Gradient Boosting

With an AUC of 0.95, this ROC curve shows decent model performance and classification ability. With a low false positive rate across thresholds and a high true positive rate, the curve hugs the upper-left corner. This indicates that the Gradient Boosting model is quite good at differentiating between typical and unusual situations.

Artificial Neural Network

Accuracy: 97.21%

	precision	recall	f1-score	support
0	0.98	0.99	0.99	1186
1	0.78	0.68	0.72	68
accuracy			0.97	1254
macro avg	0.88	0.83	0.85	1254
weighted avg	0.97	0.97	0.97	1254

Updated Accuracy: 97.37%

	precision	recall	f1-score	support
0	0.98	0.99	0.99	1186
1	0.81	0.68	0.74	68
accuracy			0.97	1254
macro avg	0.89	0.83	0.86	1254
weighted avg	0.97	0.97	0.97	1254

Figure 19: Results table for Artificial Neural Network for with and without hyper tuning parameters

Artificial Neural Network demonstrated a strong ability to model intricate sensor data patterns for anomaly detection. The ANN's total accuracy after initial training was 97.21%. With a precision of 0.98, recall of 0.99, and F1-score of 0.99 for typical system conditions, the model performed exceptionally well in categorisng the majority class. With a precision of 0.78 and recall of 0.68 for the minority class, the model performed somewhat well, yielding an F1-score of 0.72, suggesting some difficulties in identifying minority anomalies.

The model's accuracy increased to almost 97.37% after hyperparameter tweaking, which involves modifying parameters like hidden layer sizes, learning rate, activation functions, and dropout rates. With recall rising to 0.68 and F1-score to 0.74, the adjustment improved minority class sensitivity and enhanced overall detection.

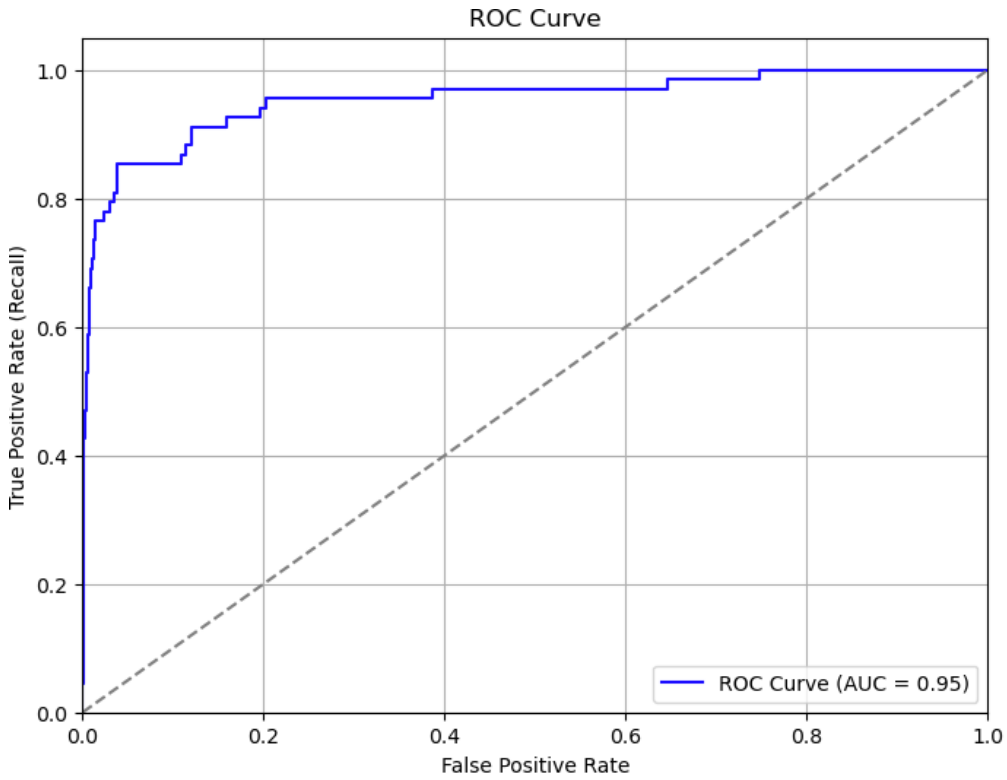

Figure 20: ROC Curve graph for Artificial Neural Network

With an AUC of 0.95, this ROC curve shows good model performance and outstanding classification ability. With a low false positive rate across thresholds and a high true positive rate, the curve hugs the upper-left corner. This indicates that the ANN model is quite good at differentiating between typical and unusual situations.

Support Vector Machines

```
Accuracy: 96.97%
              precision    recall  f1-score   support

           0       0.97      1.00      0.98      1186
           1       0.92      0.49      0.63        68

    accuracy                           0.97      1254
   macro avg       0.94      0.74      0.81      1254
weighted avg       0.97      0.97      0.97      1254

Updated Accuracy: 98.25%
              precision    recall  f1-score   support

           0       0.99      1.00      0.99      1186
           1       0.91      0.75      0.82        68

    accuracy                           0.98      1254
   macro avg       0.95      0.87      0.91      1254
weighted avg       0.98      0.98      0.98      1254
```

Figure 21: Results table for Support Vector Machines for with and without hyper tuning parameters

With an initial accuracy of 96.97%, the Support Vector Machine model performed admirably. The SVM obtained an F1-score of 0.98 and a precision of 0.97 and recall of 1.0 for the majority class. Performance for the minority class was mediocre, with an F1-score of 0.63, precision of 0.92, and recall of 0.49.

Following probability calibration and hyperparameter adjustment, overall accuracy increased a fair bit to 98.25%. With recall rising to 0.75 and F1-score improving to 0.82, the model showed improved minority class recognition, demonstrating improved sensitivity to anomalous events while retaining good classification performance overall.

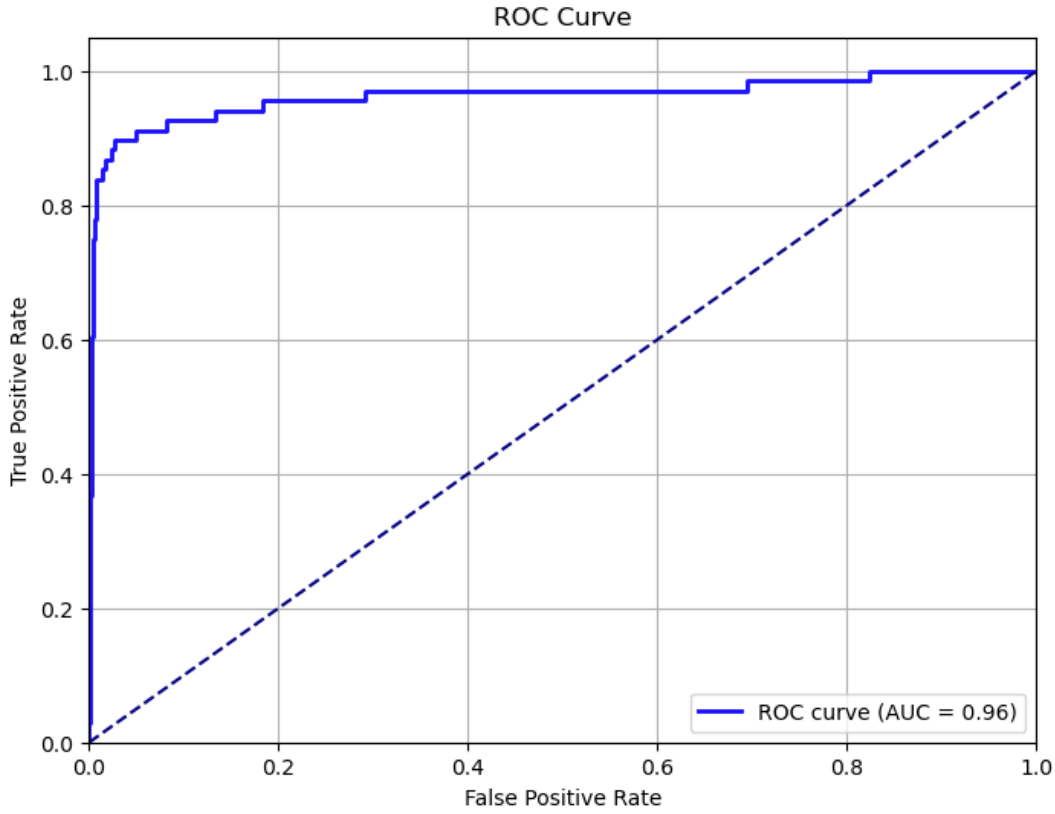

Figure 22: ROC Curve graph for Support Vector Machines

With an AUC of 0.96, this ROC curve for SVM shows good model performance and great class discrimination. With a high true positive rate and a low false positive rate, the curve rapidly rises toward the top-left corner, making it perfect for anomaly detection jobs.

2.4 Model Comparison

Model	Accuracy (Post hyperparameter Tuning)
Logistic Regression	96.57%
Random Forest	97.29%
Gradient Boosting	97.37%
Artificial Neural Network	97.37%
Support Vector Machines	98.25%

Table 2: Model Comparison Table

The performance of five distinct machine learning algorithms following hyperparameter is displayed in the model comparison table. At 98.25%, Support Vector Machines had the best accuracy, followed by Gradient Boosting and Artificial Neural Network at 97.37%. This makes SVM the champion model and both Gradient Boosting and Artificial Neural Network becoming the challenger model. The accuracy of Logistic Regression was the lowest at 96.57%. Although the tight clustering of accuracies between 96% and 98% suggests that the algorithm choice may not be as important as proper feature engineering and hyperparameter optimisation for this specific problem, the comparatively high performance across all models suggests that the dataset has distinct patterns that enable effective classification.

2.5 Demonstration

The BATADAL dataset, which replicates actual water distribution operations and assault scenarios, is used to illustrate the created artifact, a machine learning-based anomaly detection algorithm. The model is tested on data with labelled abnormalities, such as operational flaws and cyber-physical attacks, after being trained on data that depicts typical system behaviour. To detect departures from typical patterns, the artefact processes multivariate sensor inputs, including pressure, flow rate, and tank levels. The model effectively identifies attack periods and operational disruptions, frequently before significant system indicators are impacted, as demonstrated by the graphic timelines included in the demonstration. Performance indicators that support the efficacy of the artifact include precision, recall, and F1-score.

By precisely identifying anomalies in time-series data and offering insights that can help operators respond to risks, the demonstration validates that the model achieves its goals. This demonstrates how useful the artifact is for improving water distribution systems' situational awareness and security.

A simple interface for showcasing the capabilities of the trained machine learning models for anomaly detection in water distribution systems is shown in Figure 23 below. One of the main features is a file upload section that lets users add fresh datasets for testing in CSV format. The interface automatically pre-processes data after it is uploaded, addressing missing values and using the same standardisation that was used during training. After that, users can choose from a variety of machine learning models, including Logistic Regression, Random Forest, Gradient Boosting, SVM, and ANN, to use on the dataset. Predictions and anomalies are then shown on the interface, along with a summary table of categorisation parameters including recall, accuracy, and precision. It also provides visual outputs to assist users in interpreting model performance, such as a ROC curve and confusion matrix. Together, these characteristics form the interface.

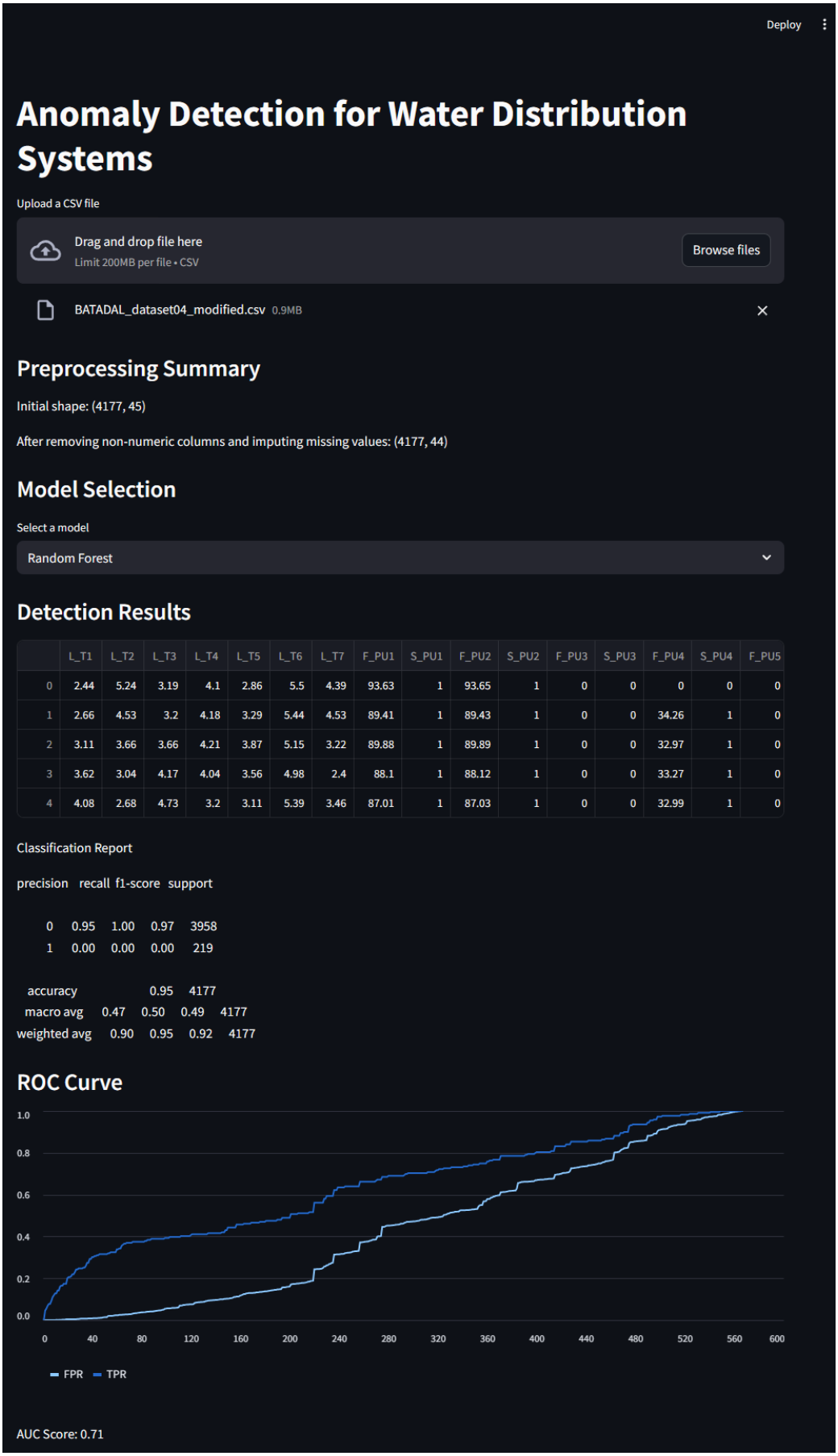

Figure 23: Example working interface for model deployment

3: Reflection

The creation of an anomaly detection artefact for water distribution systems using machine learning has provided valuable insights into both the operational significance of these systems and the technical challenges involved in securing them. As I progressed through the project, it became increasingly evident that successful model performance heavily relied on rigorous data exploration and thorough pre-processing. Steps such as handling missing values, removing irrelevant features, and applying standardisation were critical to ensuring that the models could effectively learn patterns and distinguish between normal operations and cyber-physical attacks.

Throughout the project, I implemented and evaluated a range of machine learning models such as Random Forest, Gradient Boosting, Support Vector Machines, Artificial Neural Networks, and Logistic Regression, using the BATADAL dataset. This hands-on comparison illuminated the strengths and limitations of each approach in the context of anomaly detection. For example, while SVM provided the highest overall accuracy and robustness to class imbalance, models like Gradient Boosting and ANN demanded more intensive tuning and pre-processing to reach acceptable performance levels. These practical experiments deepened my understanding of how each algorithm behaves when faced with real-world, imbalanced data and highlighted the importance of choosing the right model architecture for the task at hand.

One of the most important lessons I learned was the need to strike a balance between model interpretability and complexity. In a domain like water distribution, where quick and reliable decision-making is essential, overly complex models may hinder practical deployment unless accompanied by tools for interpretability. Techniques such as feature importance analysis not only helped me understand the inner workings of the models but also provided insights into which sensors and variables contributed most significantly to anomaly detection. This information can be instrumental in guiding future system monitoring strategies and reducing computational overhead. The experience also highlighted the value of interdisciplinary collaboration. Combining data science techniques with an understanding of water infrastructure operations enhanced the relevance and potential impact of the project. This fusion of domain knowledge and technical skill is particularly important when designing solutions intended for critical infrastructure protection, where the stakes are high and false positives or missed detections can have serious consequences.

When completing the interface, my opportunities only allowed for a simple one to be completed. In reflection, I would’ve liked to explore my options more in this section and implement an actual detection system for other datasets. It would work through the user importing a dataset and the interface showing the user whether and attack has occurred or not.

Overall, this project has significantly improved my skills in data pre-processing, model development, evaluation, and critical thinking. It has also strengthened my appreciation for the real-world implications of cyber-physical threats and the importance of proactive anomaly detection. Looking ahead, I am motivated to explore more advanced methodologies, such as real-time anomaly detection systems and adaptive learning models, and to consider how these solutions could be applied and scaled in operational environments for greater security and resilience.

4: Discussion

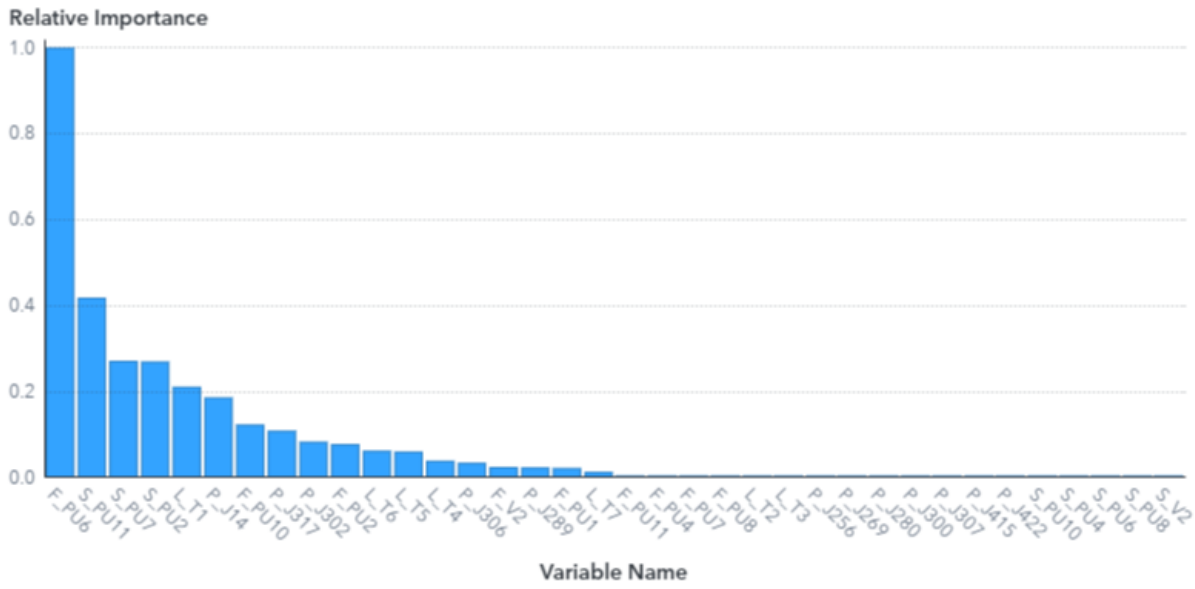

Figure 24: Importance graph for every attribute in the BATADAL dataset

The relative significance of each variable in the dataset is shown in this bar chart according to how it contributes to the predictive model. The model chosen for this chart was SVM as it had the greatest performance in terms of accuracy and best represents the model. The height of each bar, which stands for a feature, reflects how much that feature affects the model's capacity to categorise or forecast the target variable, in this case, the water distribution system's anomaly detection.

We may observe from the chart that a normalised relative value of 1.0 indicates that F_PU6 is by far the most significant variable. This implies that out of all the features, it has the best predictive ability. After that, S_PU11, S_PU7, S_PU2 and L_T1 also exhibit moderate importance, suggesting that they provide a significant contribution to the model, albeit not as much as F_PU6. The relative importance of most other factors is substantially lower, with several approaching zero. This suggests that they don't significantly affect the model's decision-making and might be candidates for feature selection or dimensionality reduction to simplify the model without compromising performance.

Overall, this graph is crucial for figuring out which sensors or data points have the greatest impact on identifying anomalies. It can also direct future monitoring, prioritise sensor maintenance, or provide focused analysis for the water infrastructure system.

4.1 Future Work

Future work for the interface section would be my top priority in this project as the deployment stage is just as important as the initial stages. As discussed in the reflection, additional features such as a visual indication that there has been an attack in the dataset. Furthermore, using ensemble learning techniques to combine many models may increase detection accuracy and decrease false positives. Investigating deep learning architectures, such recurrent neural networks (RNNs) or convolutional neural network structures (CNNs), may also help the system become more predictive by revealing more intricate patterns in sensor data. Additionally, adding more varied operating scenarios and cyberattack simulations to the dataset will improve model generalisation, and researching explain ability strategies might help stakeholders better understand how the models make decisions. A comprehensive strategy for protecting water infrastructure against changing cyberthreats could also be

implemented and future studies could evaluate the combination of cybersecurity measures and anomaly detection.

5: Reference List

Maria I. Fyodorova, I. I. Z., Dmitry S. Kuzenev, Andrey E. Bannov, Denis P. Khmelyuk. (2023). The SCADA System Digital Infrastructure Implementation in Distribution Networks. *5*, 1-6. https://doi.org/10.1109/REEPE57272.2023.10086846

Mashor Housh, Z. O. (2018). Model-based approach for cyber-physical attack detection in water distribution systems. https://doi.org/10.1016/j.watres.2018.03.039

Mina Cha, W. K., Yeo-Myeong Yun, Jungwon Yu, Kwang-Ju Kim, Seongyun Kim. (2024). Development of Water Quality Analysis for Anomaly Detection and Correlation with Case Studies in Water Supply Systems. https://doi.org/10.1109/PlatCon63925.2024.10830723.

Nicolas Nicolaou, D. G. E., Christos Panayiotou, Marios M. Polycarpou. (2018). Reducing Vulnerability to Cyber-physical Attacks

in Water Distribution Networks. https://doi.org/10.1109/CySWater.2018.00011

Riccardo Taormina, S. G., Nils Ole Tippenhauer, Avi Ostfeld, Elad Salomons, Demetrios Eliades. (2016). *(BATADAL) BATtle of the Attack Detection ALgorithms*. https://www.batadal.net/index.html

6: Appendices

UTS: ENGINEERING & INFORMATION TECHNOLOGY

SUBJECT NUMBER & NAME	NAME OF STUDENT(s) (PRINT CLEARLY)	STUDENT ID(s)
31482 Honours Project	Brendan Chan	14281529

STUDENT EMAIL	STUDENT CONTACT NUMBER
brendan.r.chan@student.uts,edu.au	0410688605

NAME OF TUTOR	TUTORIAL GROUP	DUE DATE
Dr Hai Yan (Helen) Yu	-	Week 14

ASSESSMENT ITEM NUMBER & TITLE

Assessment Task 2: Research Report and Research Work

☑ I confirm that I have read, understood and followed the guidelines for assignment submission and presentation on page 2 of this cover sheet.

☑ I confirm that I have read, understood and followed the advice in the Subject Outline about assessment requirements.

☑ I understand that if this assignment is submitted after the due date it may incur a penalty for lateness unless I have previously had an extension of time approved and have attached the written confirmation of this extension.

Declaration of originality: The work contained in this assignment, other than that specifically attributed to another source, is that of the author(s) and has not been previously submitted for assessment. I understand that, should this declaration be found to be false, disciplinary action could be taken and penalties imposed in accordance with University policy and rules. In the statement below, I have indicated the extent to which I have collaborated with others, whom I have named.

Statement of collaboration:

Signature of student(s): Brendan Chan.

Date: 9/06/2025

✂----------

ASSIGNMENT RECEIPT

To be completed by the student if a receipt is required

SUBJECT NUMBER & NAME	NAME OF TUTOR

SIGNATURE OF TUTOR	RECEIVED DATE