**New GasPipeline Multiclass Technical Report Function**

**Date**: Jan 22, 2025

**Tech Report Filename**: Multiclass\_GasPipeline\_Tech\_Report\_Function.docx

This page is to be printed on the back of the cover page. This text is hidden so it will not print.

Table of Contents

[1 Introduction 6](#_Toc192164927)

[2 Research Objectives 8](#_Toc192164928)

[3 Uncertainties and Research Challenges 9](#_Toc192164929)

[4 SCADA Architecture 10](#_Toc192164930)

[4.1 Sensor/Actuator: 11](#_Toc192164931)

[4.2 PLC/RTU/IED: 11](#_Toc192164932)

[4.3 SCADA Gateways: 12](#_Toc192164933)

[4.4 SCADA Server - MTU 12](#_Toc192164934)

[4.5 Data Historian: 12](#_Toc192164935)

[4.6 HMI: 13](#_Toc192164936)

[5 Gas Pipeline SCADA Testbed and Dataset 13](#_Toc192164937)

[5.1.1 Data Preprocessing and Exploratory Data Analysis (EDA) on Gas Pipeline SCADA System Dataset 18](#_Toc192164938)

[6 Methodology/Approach 20](#_Toc192164939)

[6.1 Feature Ranking 21](#_Toc192164940)

[6.1.1 Statistical Properties of Dataset (SP) 22](#_Toc192164941)

[6.1.2 WFI Using Random Forest 24](#_Toc192164942)

[6.1.3 WFI Using XGboost 25](#_Toc192164943)

[6.1.4 Feature Reduction Analysis 26](#_Toc192164944)

[6.2 Selective Feature Selection (SPFS) 27](#_Toc192164945)

[6.3 Recursive Feature Elimination (RFE) 28](#_Toc192164946)

[6.4 Cross-Validation 30](#_Toc192164947)

[6.5 Metrics Evaluation 30](#_Toc192164948)

[6.6 Averaging Methods for Metrics Evaluation 31](#_Toc192164949)

[6.6.1 Micro Averaging 31](#_Toc192164950)

[6.6.2 Macro Averaging 32](#_Toc192164951)

[6.6.3 Weighted Averaging 33](#_Toc192164952)

[6.6.4 Evaluation of Averaging Methods 33](#_Toc192164953)

[7 Design/Framework/Testbed 34](#_Toc192164954)

[8 Results and Discussion 36](#_Toc192164955)

[8.1 Overview of Result Metrics 36](#_Toc192164956)

[8.1.1 Key Performance Metrics 38](#_Toc192164957)

[8.1.2 Performance Trends 38](#_Toc192164958)

[8.1.3 Comparison by Model Type, Ranking Method, and Feature Selection 39](#_Toc192164959)

[8.2 Ranking Results 40](#_Toc192164960)

[8.2.1 Statistical Properties Ranking Results 40](#_Toc192164961)

[8.2.2 Weighted feature selection ranking results using RandomForest Classifier 42](#_Toc192164962)

[8.2.3 Weighted feature selection ranking results using XGBoost Classifier 44](#_Toc192164963)

[8.3 Summary of Best-Performing Models 46](#_Toc192164964)

[8.3.1 Best F1 Scores 46](#_Toc192164965)

[8.3.2 Best accuracy 47](#_Toc192164966)

[8.3.3 Performance of Models with Limited Features 51](#_Toc192164967)

[8.3.4 Performance on time taken 54](#_Toc192164968)

[8.4 Summary of Findings 56](#_Toc192164969)

[8.4.1 Key Observations 58](#_Toc192164970)

[8.4.2 Summary of Best-Performing Configurations 58](#_Toc192164971)

[9 Conclusion 60](#_Toc192164972)

[10 Scope of Future Researche 62](#_Toc192164973)

[11 References 62](#_Toc192164974)

[Appendix A: xxx 63](#_Toc192164975)

List of Figures

[Figure 1 Label Distribution in Function Dataset 19](#_Toc191292396)

[Figure 2 Flowchart of methodology 20](#_Toc191292397)

[Figure 3 Flowchart of RFE 29](#_Toc191292398)

[Figure 4 Flowchart of model evaluation process 36](#_Toc191292399)

[Figure 5 Impact of Feature Reduction on Model Performance 37](#_Toc191292400)

[Figure 6 Content Guidelines 75](#_Toc191292401)

List of Tables

[Table 1 Description of features from New Gas Pipeline dataset 15](#_Toc191292384)

[Table 2 Example Feature Ranking based on metrics from the Function dataset 23](#_Toc191292385)

[Table 3 Feature importance ranking using Random Forest. 24](#_Toc191292386)

[Table 4 Feature importance ranking using XGBoost. 25](#_Toc191292387)

[Table 5 Ranking features in function dataset based on statistical properties 40](#_Toc191292388)

[Table 6 Feature importance rankings for the function dataset using the RandomForest classifier 42](#_Toc191292389)

[Table 7 Feature importance rankings for the function dataset using the XGBoost classifier 44](#_Toc191292390)

[Table 8 Top 30 best preforming combinations based on greatest F1 score 46](#_Toc191292391)

[Table 9 Best preforming combinations when reducing the number of features to 10 or less 49](#_Toc191292392)

[Table 10 XGBoost best performance when less than 10 features are selected 51](#_Toc191292393)

[Table 11 Graph showing all combinations best F1-Score 56](#_Toc191292394)

[Table 12: Formatting Guideline 74](#_Toc191292395)

# Introduction

Supervisory Control and Data Acquisition (SCADA) systems are integral to the operation of critical infrastructure, including utilities like water, electricity, gas, and transportation systems. These systems provide real-time monitoring and control of industrial processes, ensuring efficient and safe operations. SCADA systems gather data from sensors and devices in the field, process this data, and present it to human operators in a comprehensible format, enabling them to make informed decisions and respond swiftly to any issues.

SCADA systems play a crucial role in maintaining the stability and reliability of critical infrastructure. For instance, in the energy sector, SCADA systems help manage the distribution of electricity by monitoring the grid's performance and detecting faults before they escalate into major outages. In the water supply industry, SCADA ensures the quality and availability of water by controlling pumps, valves, and chemical treatment processes. The transportation sector relies on SCADA for the smooth operation of traffic lights, railway signals, and other essential systems.

Given the vital functions that SCADA systems perform, any disruption can have significant repercussions. Ensuring the security and integrity of SCADA systems is paramount to safeguarding the continuous and reliable operation of critical infrastructure.

Vulnerabilities of SCADA Systems

Despite their importance, SCADA systems are inherently vulnerable to cyber threats [1]-[11]. Originally designed for isolated and controlled environments, many SCADA systems were not built with robust security features to counter modern cyber threats. The increasing interconnectivity and integration of SCADA with corporate IT networks and the internet have exposed these systems to a broader range of attack vectors.

One notable example of a SCADA-related cyber incident is the attack on the Colonial Pipeline in May 2021 [6]. The ransomware attack forced the pipeline operator to shut down its operations, leading to significant fuel supply disruptions across the southeastern United States. This incident highlighted the susceptibility of SCADA systems to cyberattacks and the potential for such breaches to cause widespread economic and social disruption.

Other examples of SCADA system breaches include the 2015 cyberattack on Ukraine's power grid [7,8], which resulted in widespread power outages, and the 2010 Stuxnet worm, which targeted Iran's nuclear facilities. These incidents underscore the critical need for enhanced security measures to protect SCADA systems from malicious actors.

Machine Learning and Deep Learning for Anomaly and Attack Detection

The increasing complexity and frequency of cyber threats necessitate advanced solutions for detecting and mitigating attacks on SCADA systems. Machine learning (ML) and deep learning (DL) offer promising approaches to enhance the security of these systems by identifying anomalies and potential attacks in real-time [12-17].

Machine learning algorithms can analyze vast amounts of data generated by SCADA systems to establish baseline behaviors and detect deviations that may indicate cyber threats. These algorithms can be trained to recognize patterns associated with normal operations and flag any unusual activities for further investigation. For example, ML-based systems can detect abnormal network traffic patterns, unauthorized access attempts, or unexpected changes in system configurations.

Deep learning, a subset of machine learning, leverages artificial neural networks to model complex relationships in data. Deep learning techniques can improve the accuracy and speed of anomaly detection by processing high-dimensional data and identifying subtle indicators of attacks that traditional methods might overlook. Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) are particularly effective in analyzing time-series data from SCADA systems, enabling the detection of sophisticated and stealthy attacks.

By integrating ML and DL-based anomaly detection systems, SCADA operators can achieve more robust and proactive security postures. These technologies can provide continuous monitoring, real-time alerts, and automated responses to potential threats, thereby reducing the risk of successful cyberattacks and minimizing their impact on critical infrastructure.

# Research Objectives

The main objective of this research study is to investigate different machine learning classification techniques, data normalization, automated feature selection, and data balancing techniques to explore and achieve robust Machine Learning based models for Intrusion Detection System (IDS) of gas pipeline SCADA system.

In this work, we focus on understanding the data and their processing techniques before modeling through various experiments. This allowed us to select appropriate data processing and feature selection methodologies as well as enable us to develop a new and efficient systematic strategy for obtaining high-performance M/DL model. We proposed in this work a Robust Strategy for Best Machine Learning Model Selection (RS-BMLMS) and validated it on Gas Pipeline dataset as provided by Mississippi State University and Oak-Ridge National Laboratory for industrial control system security research [17]. In this work, we focus on detecting two specific attacks namely Naïve Malicious Response Injection (NMRI) and Complex Malicious Response Injection (CMRI) that are related to response messages/data.

# Uncertainties and Research Challenges

The major research challenge of this work is to design and develop robust machine learning based anomaly detection and intrusion detection models that can be used for intrusion detection in gas pipeline SCADA system.  The developed models will optimize the trade-off between multiple objectives such as reduce the training time, generalize the model to detect new attacks, improve the model accuracy, and minimize the false alarms. At present most of the intrusion detection systems used in practice are signature-based. A signature-based intrusion detection system detects attacks by looking for specific patterns, such as byte sequences in SCADA application, or known malicious instruction sequences used by malware. However, it is difficult to detect new attacks, for which no pattern is available. Machine learning-based anomaly detection can be used for SCADA systems, and it is expected that it will work better than signature-based algorithms for zero-day (new) attacks.  However, the high false alarm rate often limits practitioners to use machine learning based system for intrusion detection. Also, detecting new attack requires the model to be generalized which in turn requires the model to be trained with huge volume of data. Although these models often show excellent performance on the data population they are trained with, these models may not work well with new data.

The uncertainty remains whether the model training can be done efficiently, and model can be generalized to work on new data with high accuracy and low false alarm rate.

Towards overcoming the above uncertainties and challenges, different approaches can be taken such as using ensemble models, automated feature selection techniques and optimizing hyper-parameters. It is expected that these models will be suitable for off-line periodic training and on-line intrusion detection in gas pipeline SCADA system.

The data preprocessing and feature selection impact the model performance significantly. Therefore, in this study we focus on good model development and best model selection strategies by performing a series of experiments as will be discussed in details in subsequent sections.

# SCADA Architecture

A typical SCADA system consists of SCADA Master Terminal Unit (MTU) that is connected to either a remote location unit or connected locally to a local location unit that contains controllers, remote terminal units (RTU) or Programmable Logic Controllers (PLC) at one end and other end is connected to the control system consist of Human Machine Interface (HMI), data historian & workstation. Figure 1 below shows a simple SCADA system architecture [9]. In the following we briefly describe the main components of the SCADA system.

A screenshot of a cell phone

Description automatically generated

Figure 1 SCADA System Architecture [13]

## Sensor/Actuator:

Sensors within industrial field level check whether parts are present, size of the part, color of the part, and whether the product is full or empty, good or bad. Sensors are also used to ensure safety of the equipment, product and operators. A common classification of sensors is contact and non-contact. Another way to classify sensors is by analog and digital.

## PLC/RTU/IED:

SCADA system RTUs are microprocessor-based devices designed to monitor and collect data as well as perform some control functions. These functions depend upon the requirements of the specific industry and the technology used by the RTU manufacturer. The industry has developed smart RTUs with the capability to execute logic and PID closed loop control. The SCADA market has moved away from the traditional proprietary RTUs and toward multi-purpose devices, such as programmable logic controllers (PLCs) and intelligent electronic device (IEDs). However, traditional, purpose-built RTUs are still likely to be used in environmentally challenging applications, such as where extremes of moisture, temperature, or humidity are present, or low power consumption is required [14].

IEC 61131-3, international standard is used to define PLC/RTU programming languages and concepts. Programming languages are divided into two main sections to represent the five main programming languages. Generally, ladder logic and block diagram are the two common methods for the PLC programming.

## SCADA Gateways:

Different devices at field site, uses different types of protocols (e.g. Modbus, serial, fieldbus, dnp3 etc.) to communicate amongst each-other. SCADA gateways are used to connect different protocols to an IP network via ethernet, GSM, Wi-Fi, Zigbee, LoRa, GPRS etc. These devices might not be air-gapped, as they are having antennas, and they are interconnected by different wireless protocols [10]. Generally, these devices are soft target devices to intercept and get into the network by firmware analysis and reverse engineering methodology.

## SCADA Server - MTU

Master Terminal Unit (MTU) is located at the control center. MTU acts as heart of the SCADA system. It issues the commands to the field devices like PLCs/RTUs which are located at remote places from the control center to gather the required data from the plant floor. Furthermore, it processes the information, stores the important status information at data historian and display the information in the form of graphs, curves and tables on HMI to help in taking control decisions. Moreover, the communication between MTU and field control devices are only initiated by the program resides in the MTU. Hence, MTU act as a Master and RTU is the slave, however the level of communication is considered as peer-to-peer communication.

## Data Historian:

A Data Historian (also known as a Process Historian or Operational Historian) is a software program that records and retrieves production and process data by time; it stores the information in a time series database that can efficiently store data with minimal disk space and fast retrieval [16]. Time series information is often displayed in a trend or as tabular data over a time range (ex. the last day, last 8 hours, last year).

## HMI:

An HMI is a software application that presents information to the users about the state of the process and to accept the user and control instructions. HMI provides a graphical representation of the control system process and provides real time data acquisition. It monitors data and makes it available to users, as well as provides an interface for inputs from the users. One of the advantages that HMI provides is that they provide an interface between the user and the machine within the factory floor, a centralized control unit and a local interface of the process for quick monitoring and adjusting.

# Gas Pipeline SCADA Testbed and Dataset

Mississippi State University’s in-house SCADA lab has a gas pipeline system to collect data for cyber attack research as shown in Figure 2. The system consists of three major components: sensors and actuators, a communication network, and supervisory control as shown in the picture below [17].   This section summarizes the testbed components and the methods by which the dataset was collected using this testbed.  Detailed information can be found in [17].

The gas pipeline has two actuators namely a pump and a solenoid as well as a pressure sensor. The actuators are used to maintain the pressure. The pressure is set by the supervisory control system. The gas pipeline has three main system modes: automatic, manual, off. In automated mode, there are two schemes – the pump mode, which turns the pump on and off to keep the pressure in the pipe at the set point and the solenoid mode, in which a relief valve controlled by a solenoid is opened and closed to regulate pressure. Both the pump and solenoid modes used a Proportional-Integral-Derivative (PID) control scheme. In manual mode, the operator must manually control the pump and solenoid.

The next component is the communication network in which the protocol used is serial Modbus RTU. Modbus packets include a header and a payload. For Modbus over a Serial Line, a packet includes a device address, function code, payload, and a cyclic 21 redundancy code (CRC) or linear redundancy code (LRC).

A collage of pipes and a pressure gauge

Description automatically generated

Figure 2 Gas pipeline SCADA testbed [17]

The gas pipeline database of cyber attacks was originally created by Tommy Morris and Wei Gao in 2012 [18].  However, study on the original dataset shows some major flaws in the dataset and it was declared unsuitable for machine learning research [17]. Later Ian Turnipseed of Mississippi State University collected more realistic cyber attack gas pipeline dataset known as “new gas pipeline dataset” and made it publicly available for research community.  In our study the new gas pipeline dataset is used.

The “new gas pipeline” dataset is available in two formats. The first form is a comma separated value (CSV) text file, and the second form is an Attribute Relationship File Format (ARFF). The ARFF dataset was created to make it compatible with Waikato Environment for Knowledge Analysis (WEKA), a tool that has a comprehensive list of machine learning algorithms and has been used by many researchers world-wide for testing the performance of specific machine learning algorithms [14]. Although we are not using WEKA, we preprocessed the ARFF dataset to make it ready for training different classification models.  Each record or instance in the dataset represents one packet being delivered to either the MTU or to the RTU. Each instance in the dataset contains SCADA application information along with payload information.   SCADA systems have fixed network topologies and the transactions between the components are repetitive and regular as opposed to IT network whereas the SCADA application data is very dynamic and irregular in nature. The second category of features is the payload information. The payload information provides information about the gas pipeline’s state, settings, and parameters. These values are critical to detect if any anomaly is found in the system which could be due to malfunction of the system or malicious activities by the cyber attackers. There are total of 274,628 instances or rows in the dataset and each row contains twenty columns. The columns are commonly referred to as features. These features are summarized with a brief description in Table 1.

Table 1 Description of features from New Gas Pipeline dataset

|  |  |  |  |
| --- | --- | --- | --- |
| Feature Id | Feature Symbol | Type | Description |
| 1 | address | real | The address of the slave device. Each slave device in the Modbus is assigned a 8-bit address to identify the slave device the master is communicating to and from |
| 2 | function | real | The function codes are primarily used in the gas pipeline to indicate a read (0x03) and write commands (0x16). But there are possibilities of total 256 such commands. A denial of service attack can be launched by setting a function code of 0x08 which corresponds to diagnostic mode where the device would be always in listening mode. |
| 3 | length | real | Length of the Modbus frame. This is fixed for each command and response frame. Frames that are not of specific length can be easily detected as attack |
| 4 | setpoint | real | This value controls the pressure in the gas pipeline |
| 5 | gain | real | Gain parameter of the PID controller |
| 6 | reset rate | real | Reset rate parameter of PID controller |
| 7 | deadband | real | Deadband parameter of PID controller |
| 8 | cycle time | real | Cycle time parameter of PID controller |
| 9 | rate | real | Rate parameter of PID controller |
| 10 | system mode | {0,1,2} | Controls the duty cycle of the system. The following modes are valid  0 – Off  1 – Manual  2 - Automatic |
| 11 | control scheme | {0,1} | The control scheme in the gas pipeline determines whether the system will be controlled by the pump or by the solenoid. There are two schemes:  0 – Pump  1 - Solenoid |
| 12 | pump | {0,1} | This is the state of the pump when system mode is set to manual. There are two possible values:  0 – Off  1 - On |
| 13 | solenoid | {0,1} | This represents the state of the solenoid valve. There are two possible values:  0 – Closed  1 – Open |
| 14 | pressure | real | The current pressure measurement from the gas pipeline |
| 15 | crc rate | real | The Cyclic Redundancy Check (CRC) allows system to check error within a Modbus frame |
| 16 | command response | {0,1} | This value allows the IDS to learn about command and response frame. Two possible values:  0 – Response  1 – Command |
| 17 | time | real | Timestamp of the instance |
| 18 | binary result | {0,1} | Labels to indicate either attack (1) or normal (0) instance |
| 19 | categorized result | Range(0:7) | Labels to indicate the category of attack |
| 20 | specific result | Range(0:35) | Labels to indicate the specific type of attack |

### ****Data Preprocessing and Exploratory Data Analysis (EDA) on Gas Pipeline SCADA System Dataset****

In this subsection, we discuss the preprocessing and feature extraction steps applied to the gas pipeline dataset, as introduced in Section 5. For the simulation study reported in this section, the original gas pipeline dataset was preprocessed with necessary feature extraction into the following categories:

* **Dataset 1:** The entire gas pipeline dataset was preprocessed to remove data instances with a majority of missing feature values, along with features that are not relevant to this study. This resulted in a reduced dataset containing **64,100 rows (data instances) and 12 columns (11 features plus output).**
* **Dataset 2:** The full dataset was further divided into three subsets based on dataset types: **Command (C), Function (F), and Response (R).** After removing instances with missing features, the resulting subsets had sizes of **64,100 × 13 (Command), 68,848 × 4 (Function), and 141,680 × 7 (Response),** respectively.
* **Dataset 3:** Inspired by the approach described in [18,19], new features were introduced in the **Command and Response subsets** of Dataset 2 by computing delta values of the original feature values and centroids of **N past delta values (user-defined).**
* **Dataset 4:** Inspired by the method described in [20], **special imputation techniques** and **one-hot encoding** were used to preprocess the dataset into three subsets based on dataset types: **Command (C), Function (F), and Response (R).**

For this study, we specifically focus on the **Function MinMax dataset (Dataset 2 - Function Subset),** which contains **68,848 rows and 4 features.** The Function Dataset was derived by excluding data where the function code equals 16 and length equals 90, and where the function code equals 3 and length equals 46 (i.e. excluding the command and response data). Initially, this dataset contained 141,680 rows and 20 columns. After further processing, the dataset size remained 141,680 rows but reduced to 16 columns. The input features for this dataset included address, function, length, setpoint, gain, reset\_rate, deadband, cycle\_time, rate, system\_mode, control\_scheme, pump, solenoid, pressure\_measurement, crc\_rate. The output label for the dataset is binary\_result, and the cleaned data was saved as NewGasFilteredFunction.csv. The distribution of Attack categories for the Function dataset is shown below in figure 1:

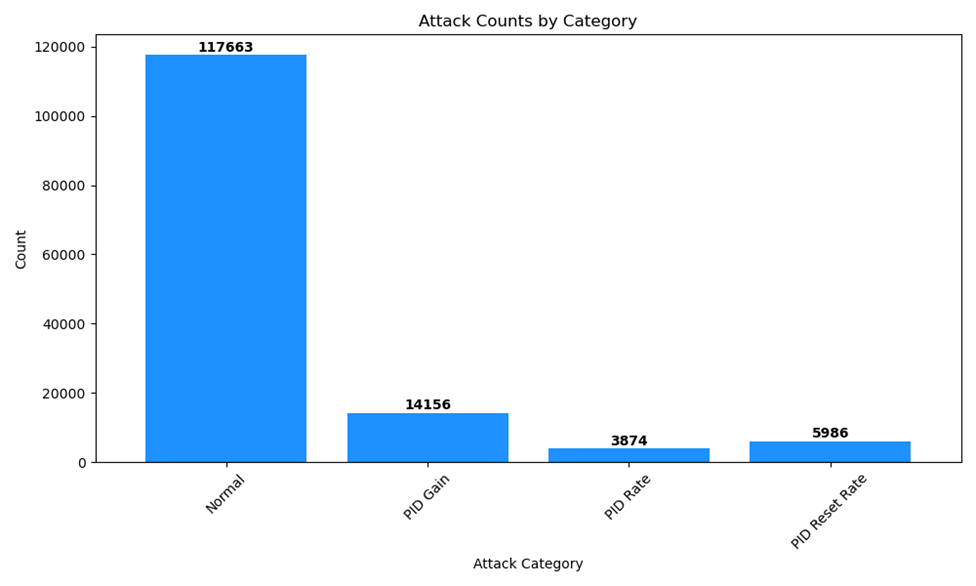


Figure 3 Label Distribution in Function Dataset

The bar chart in Figure 3 illustrates the distribution of different Attack categories within the function dataset after preprocessing. As shown, the dataset is highly imbalanced, with a significantly larger number of normal instances compared to the various attack categories.

To provide a clearer understanding of the attack categories represented in the dataset, Table 2 below outlines the corresponding label numbers assigned to each cyberattack type.

*Table 2 Specific categories of cyber attacks in a SCADA system*

|  |  |  |  |
| --- | --- | --- | --- |
| Attack Number Category Description | | | |
| Setpoint | 1–2 | MPCI | Changes the pressure set point outside and inside of the range of normal operation |
| PID gain | 3–4 | MPCI | Changes the gain outside and inside of the range of normal operation |
| PID reset rate | 5–6 | MPCI | Changes the reset rate outside and inside of the range of normal operation |
| PID rate | 7–8 | MPCI | Changes the rate outside and inside of the range of normal operation |

The dataset's severe class imbalance must be considered when training and evaluating machine learning models. The high prevalence of normal instances compared to attack categories could impact predictive performance, necessitating strategies such as weighted loss functions or data resampling techniques to improve detection capabilities.

# Methodology/Approach

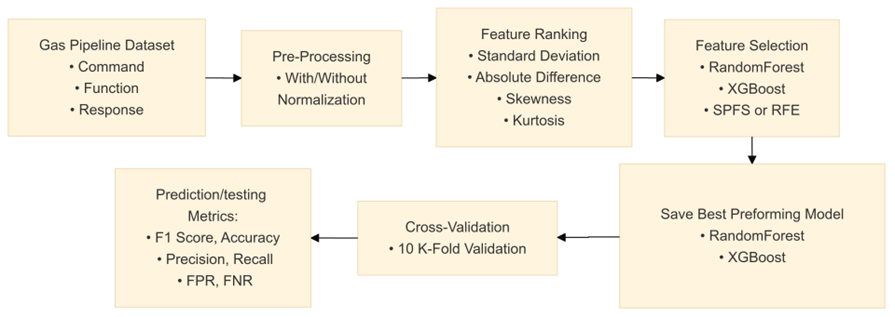


Figure 4 Flowchart of methodology

The methodology begins with data ingestion and preprocessing, where the script reads a CSV file containing gas pipeline data (e.g., commands, responses, and functions), initializes paths for storing results, metrics, and models, and optionally normalizes (e.g., Min–Max scaling) and labels datasets if needed. Feature ranking is performed using statistical measures such as standard deviation, absolute difference (mean vs. median), skewness, and kurtosis to assign importance scores to features, with an overall rank determined by aggregating these metrics. Feature selection is conducted using methods like Recursive Feature Elimination (RFE), which iteratively removes less important features, or Sequential Progressive Feature Selection (SPFS), which incrementally builds a feature set by retaining only features that improve model performance. Models, such as Random Forest or XGBoost, are trained during this process, with performance evaluated using metrics like accuracy, precision, recall, F1 score, false positive rate (FPR), and false negative rate (FNR), while training time is also recorded for comparison. Although the current code does not explicitly include cross-validation or prediction, k-fold cross-validation can be applied after feature selection to validate model performance, and the final trained model is saved as a serialized file (e.g., .pkl) for future prediction tasks. Figure 5 illustrate proposed methodology to classify the multple attack scenario and normal events using gas pipeline dataset.

## Feature Ranking

To preprocess the dataset and identify the most significant features, we applied multiple feature ranking techniques. These methods aim to quantify each feature’s importance in relation to the target variable, ensuring a refined dataset for improved model performance.

Our approach combining statistical analysis and machine learning-based importance to achieve a comprehensive ranking. First, we analyzed the dataset’s statistical properties (SP), evaluating metrics such as standard deviation, skewness, and kurtosis to identify key patterns and variations.

To further refine feature ranking, we employed Weighted Feature Importance (WFI) using two machine learning models: Random Forest and XGBoost. These models assessed feature significance based on their contribution to predictive accuracy, providing a more data-driven selection process. The integration of statistical and model-based ranking ensured that only the most relevant features were retained for subsequent analysis.

### Statistical Properties of Dataset (SP)

Before applying advanced feature selection methods, we first analyzed the statistical properties of the dataset to identify features with high variability, skewed distributions, or extreme outliers that could influence model performance. Understanding these statistical properties allowed us to make informed decisions on which features to prioritize or remove before proceeding with more sophisticated ranking techniques.

To assess the characteristics of each feature, we computed several key statistical metrics. Standard deviation was used to measure variability, as features with higher variance are typically more informative for classification. The absolute difference between the mean and median helped highlight skewed distributions, which often indicate the presence of outliers or unique patterns in the data. Skewness was calculated to quantify data asymmetry, as highly skewed features may introduce biases into the model. Finally, kurtosis was measured to assess the "tailedness" of a distribution, as extreme values could indicate whether a feature had heavy-tailed or light-tailed characteristics, potentially affecting predictive power.

The ranking process was implemented using Python, where each statistical metric was computed for all features. Features were then ranked individually based on their values for each metric. To create a comprehensive ranking system, a total rank was assigned to each feature by summing its rankings across all four metrics, with lower total ranks indicating higher feature importance. This ranking provided a structured approach to selecting the most relevant features for subsequent modeling steps.

The final output of this process was a detailed ranking table containing individual metric values, rankings for each metric, and a consolidated total rank. This table served as a foundation for further feature selection methods, ensuring that the most statistically significant features were retained for model training and evaluation.

One of the outputs from the statistical properties ranking method is shown in Table 2:

Table 2 Example Feature Ranking based on metrics from the Function dataset

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Std Dev | Abs Diff | Skew | Kurtosis | Std Dev Rank | Abs Diff Rank | Skew Rank | Kurtosis Rank | Total Rank |
| address | 0.02 | 0.00 | 21.10 | 695.97 | 1 | 1 | 1 | 16 | 19 |
| pressure | 0.04 | 0.00 | 20.20 | 419.63 | 2 | 2 | 2 | 15 | 21 |
| reset\_rate | 0.08 | 0.00 | 3.01 | 20.31 | 5 | 3 | 5 | 10 | 23 |
| setpoint | 0.14 | 0.00 | 0.96 | 2.90 | 8 | 4 | 9 | 7 | 28 |
| rate | 0.09 | 0.01 | 8.09 | 69.06 | 6 | 6 | 3 | 14 | 29 |
| cycle\_time | 0.08 | 0.03 | 2.49 | 21.71 | 4 | 8 | 6 | 11 | 29 |
| deadband | 0.17 | 0.02 | 0.17 | -0.97 | 9 | 7 | 12 | 5 | 33 |
| function | 0.14 | 0.03 | 5.12 | 27.34 | 7 | 9 | 4 | 13 | 33 |
| system\_mode | 0.40 | 0.25 | 1.16 | -0.42 | 11 | 11 | 8 | 6 | 36 |
| gain | 0.07 | 0.01 | -4.83 | 26.24 | 3 | 5 | 16 | 12 | 36 |
| crc\_rate | 0.42 | 0.28 | 0.09 | -1.97 | 12 | 12 | 13 | 1 | 38 |
| pump | 0.47 | 0.32 | 0.77 | -1.41 | 13 | 13 | 10 | 4 | 40 |
| solenoid | 0.50 | 0.45 | 0.19 | -1.96 | 15 | 15 | 11 | 2 | 43 |
| length | 0.19 | 0.04 | -4.47 | 18.80 | 10 | 10 | 15 | 9 | 44 |
| control\_scheme | 0.47 | 0.34 | -0.67 | -1.54 | 14 | 14 | 14 | 3 | 45 |
| Label | 1.81 | 0.78 | 2.12 | 3.18 | 16 | 16 | 7 | 8 | 47 |

### WFI Using Random Forest

Random Forest assigns importance scores using the Gini impurity criterion, which measures the reduction in node impurity caused by each feature.

**Implementation:**

1. The dataset was split into features (X) and the target variable (y).
2. A RandomForestClassifier was trained on the full dataset.
3. Feature importance scores were extracted from the model’s feature\_importances\_ attribute.
4. Features were ranked based on their importance, with higher values indicating greater predictive power.

**Outputs:**

Tabel 3 shows an example of feature ranking based on Random Forest WFI.

Table 3 Feature importance ranking using Random Forest.

|  |  |
| --- | --- |
| Name | Random Forest Feature Importance |
| function | 0.170343 |
| crc\_rate | 0.14766 |
| length | 0.135827 |
| cycle\_time | 0.094819 |
| reset\_rate | 0.093911 |
| setpoint | 0.088682 |
| gain | 0.074429 |
| deadband | 0.071171 |
| rate | 0.047493 |
| pump | 0.027476 |
| system\_mode | 0.025344 |
| control\_scheme | 0.008638 |
| solenoid | 0.006637 |
| address | 0.006363 |
| pressure\_measurement | 0.001208 |

### WFI Using XGboost

We also applied Weighted Feature Importance (WFI) using XGBoost, a gradient boosting method known for its robustness in feature selection. Unlike Random Forest, XGBoost provides gain-based importance scores, which indicate how much each feature contributes to reducing the model's overall loss function.

**Implementation:**

1. The dataset was split into input features (X) and target variable (y).
2. The class labels were encoded using LabelEncoder to ensure sequential class representation.
3. An XGBoost classifier was trained using the encoded labels.
4. The feature\_importances\_ attribute of the trained model was used to compute importance scores.
5. The features were ranked based on their importance, with higher values indicating greater influence on predictions.

**Outputs:**

The WFI process produced a ranked list of features based on their importance in the XGBoost model. The results were stored in a table, as shown in Table 4:

Table 4 Feature importance ranking using XGBoost.

|  |  |
| --- | --- |
| Name | XGBoost Feature Importance |
| length | 0.349779 |
| function | 0.183473 |
| crc\_rate | 0.175295 |
| rate | 0.154045 |
| pump | 0.032862 |
| gain | 0.021015 |
| cycle\_time | 0.018802 |
| system\_mode | 0.013917 |
| reset\_rate | 0.011271 |
| setpoint | 0.011252 |
| deadband | 0.009006 |
| control\_scheme | 0.008318 |
| solenoid | 0.0062 |
| pressure\_measurement | 0.004765 |
| address | 0 |

These ranked features were used as input for subsequent feature selection methods, ensuring the most relevant attributes were retained.

### Feature Reduction Analysis

After ranking the features using SP and WFI methods, we conducted an in-depth analysis to examine how reducing the number of features affected model performance. The primary objective of this analysis was to determine the smallest subset of features that could maintain high predictive accuracy while minimizing computational complexity.

To achieve this, we first ranked the features using two distinct approaches: the SP (Statistical Properties) method and the WFI method, which utilizes Random Forest and XGBoost for feature importance evaluation. The ranked feature list generated from these methods served as the foundation for selecting different feature subsets for further testing.

Following this, we implemented an iterative feature removal and selection process. We systematically tested various feature set sizes by progressively eliminating the least important features, removing the top one to five features from the ranked list. Additionally, we explored the performance of minimal feature sets by selecting only the top one to five most important features to assess whether a significantly reduced subset could still yield satisfactory predictive results.

For each of these feature subsets, we trained machine learning models using Random Forest and XGBoost. These models were then cross validated using multiple averaging strategies, including micro, and weighted F1-Scores, to ensure a comprehensive assessment of their predictive performance across different evaluation metrics.

The results of this analysis provided valuable insights into the relationship between feature count and classification performance. We identified the optimal number of features necessary to sustain high predictive accuracy and observed how the removal of low-importance features influenced model effectiveness. Furthermore, this process allowed us to quantify the trade-off between feature reduction and computational efficiency, ensuring that the final model maintained strong performance without unnecessary complexity. By systematically refining the feature selection process, we successfully determined the smallest subset of features that could be used without compromising classification accuracy.

## Selective Progressive Feature Selection (SPFS)

The SPFS algorithm was applied as a dynamic feature selection technique, iteratively adding features and evaluating their contributions to model performance, specifically focusing on the F1-Score. The process began with an empty feature set, progressively incorporating features based on their ranking from the previous step. At each iteration, the selected features were used to train either a Random Forest or XGBoost classifier, after which the model was evaluated on a validation set using the F1-Score. If adding a feature improved performance, it was retained; otherwise, it was discarded. This iterative refinement resulted in a final subset of selected features that maximized predictive performance. The SPFS process also generated a comprehensive metrics table detailing F1-Score, precision, recall, false positive rate (FPR), and false negative rate (FNR) at each step.

Below is a flow chart that shows the SPFS process:

A diagram of a performance measurement

AI-generated content may be incorrect.

Figure 6 Flowchart of SPFS process

## Recursive Feature Elimination (RFE)

Recursive Feature Elimination (RFE) is a backward feature elimination algorithm designed to iteratively remove the least significant features until a specified number of features remains. This method is particularly useful for identifying and eliminating features that negatively impact model performance, thereby enhancing predictive accuracy and computational efficiency.

The implementation of RFE began with an initial setup where all features were included in the model. From there, an elimination process was carried out, in which features were removed one at a time, starting with the least important feature as determined by the model’s internal feature importance metric. After each removal, the model was retrained, and its F1-Score was recalculated to evaluate its performance. This iterative process continued until the optimal subset of features was identified based on the highest recorded F1-Score.

The RFE process produced several key outputs. First, it generated a ranking of features based on their contribution to the model’s overall performance. Additionally, it identified a subset of features that yielded the best results, similar in function to the SPFS method.

Figure 3 presents the flow chart of RFE process:

A diagram of a process flow

AI-generated content may be incorrect.

Figure 7 Flowchart of RFE

## Cross-Validation

To ensure the reliability and robustness of the model, we employed k-fold cross-validation using the selected features identified during the feature selection process. This method divides the training dataset into k equally sized folds (with k=10 in this implementation), where the model is trained on k-1 folds and validated on the remaining fold. The process is repeated k times, with each fold used as the validation set exactly once, ensuring that every data point in the training set is used for both training and validation.

The training dataset was generated through an 80/20 split of the original data, where 80% was allocated for training and 20% reserved as a separate test set for final evaluation. During cross-validation, only the training data is used, and the results of all folds are aggregated to evaluate the model's overall performance.

To determine the best-performing model, we selected the model that achieved the highest F1 score across the 10-folds. This model was then tested on the remaining 20% of the data to assess its final performance. After evaluation, the selected model was saved as a .pkl file for future prediction tasks.

## Metrics Evaluation

After cross-validation, the best model's performance was evaluated using a variety of metrics to assess its predictive accuracy and ability to generalize to unseen data. The following metrics were calculated for each fold during the cross-validation process:

* **Accuracy**: The proportion of correctly predicted instances out of the total instances. It provides a general overview of the model's performance.
* **Precision**: The ratio of true positive predictions to all positive predictions. It evaluates the model's ability to avoid false positives.
* **Recall**: The ratio of true positive predictions to all actual positive instances. It measures the model's ability to identify all relevant instances.
* **F1 Score**: The harmonic mean of precision and recall, balancing the trade-off between the two metrics.
* **False Positive Rate (FPR)**: The proportion of negative instances incorrectly classified as positive.
* **False Negative Rate (FNR)**: The proportion of positive instances incorrectly classified as negative.

These metrics were saved to a CSV file for documentation and analysis.

## Averaging Methods for Metrics Evaluation

In multi-class classification tasks, evaluation metrics such as Precision, Recall, and F1-Score are commonly averaged using different strategies to provide a holistic assessment of the model's performance across all classes. The three primary averaging methods used in this study are Micro Averaging, Macro Averaging, and Weighted Averaging. Each method offers a unique perspective on performance, making them essential for analyzing model effectiveness in different contexts.

### Micro Averaging

Micro averaging computes the metrics by aggregating the contributions of all classes before performing the final calculation. It treats every individual instance equally, regardless of class distribution. This is especially useful when class imbalance is present.

The micro-average calculations are given by:

Precision = Recall = F1 =

where:

* TP (True Positives) is the number of correctly classified instances for each class.
* FP (False Positives) is the number of misclassified instances.
* FN (False Negatives) is the number of missed instances.

Micro averaging gives an overall measure of the model's performance and is particularly useful in highly imbalanced datasets, where larger classes would otherwise dominate the results.

### Weighted Averaging

Weighted averaging is similar to macro averaging, but it weighs each class's contribution to the final score based on the number of true instances in that class.

Precision = Recall = F1 =

where:

is the proportion of total samples belonging to class , computed as

Weighted averaging provides a balanced measure that accounts for both class-wise performance and class distribution. It is particularly useful when comparing models on datasets with highly imbalanced class distributions.

### Evaluation of Averaging Methods

For each iteration of model evaluation, we tested both micro and weighted averaging to determine which provided the best performance. Initially, we also considered macro averaging, but its scores were significantly lower than the other two methods, making it unsuitable for this analysis.

During the experiments, micro averaging generally performed well in cases where class imbalance was significant, as it emphasized overall correctness rather than per-class performance. Weighted averaging provided a balanced perspective by incorporating class distributions, making it particularly useful for understanding real-world classification performance

The results of these comparisons are further analyzed in Section 8, where we discuss the impact of each averaging method on final model performance.

# Design/Framework/Testbed

To systematically evaluate different feature selection and ranking techniques, we implemented a structured framework that iterated through various combinations of ranking methods, feature selection strategies, model types, and averaging methods. The objective of this framework was to identify the most effective approach for feature selection and model training, ultimately improving classification performance while maintaining computational efficiency.

The process began by looping through each ranking method, including SP (Statistical Properties) and WFI (Weight-based Feature Importance), to establish an initial ordering of features based on their statistical significance or model-derived importance scores. Following this, we applied different feature selection techniques, such as Recursive Feature Elimination (RFE), Statistical Property-based Feature Selection (SPFS), and manual removal or retention of top-ranked features, to determine the optimal feature subset for training.

For each selected subset of features, two machine learning models—Random Forest and XGBoost—were trained and evaluated using multiple averaging strategies (micro, macro, and weighted F1-Scores) to ensure a robust comparison across different evaluation metrics. Each experiment involved training a model on the selected features and measuring its performance across accuracy, precision, recall, F1-Score, false positive rate (FPR), and false negative rate (FNR).

In addition to testing all possible feature selection techniques, we performed an iterative feature reduction analysis, systematically removing the least important features (from 1 to X) to analyze their impact on model performance. The goal was to determine the minimum number of features required to maintain high classification accuracy while improving computational efficiency. By comparing the performance of models trained on progressively smaller feature subsets, we identified potential trade-offs between feature count and predictive power.

This comprehensive approach allowed us to evaluate the effects of different feature selection and ranking methods, ultimately leading to the identification of the most effective strategy for feature reduction in gas pipeline anomaly detection. The results from these experiments provided insights into which feature selection and ranking combinations yielded the highest F1-Scores while maintaining low false positive and false negative rates.

The diagram below illustrates the automated pipeline used for feature selection and model evaluation in our study. The process begins with loading the dataset, followed by ranking the features using Statistical Properties (SP) or Weighted Feature Importance (WFI) methods, which include Random Forest and XGBoost. After ranking, different feature selection techniques such as Recursive Feature Elimination (RFE) and Statistical Properties Feature Selection (SPFS) are applied to refine the feature subset. Additionally, an alternative approach is employed where the least important *X* features are systematically removed. The selected feature subset is then used to train a machine learning model, after which the model's performance is evaluated using various metrics, including F1-Score, Accuracy, Precision, Recall, False Positive Rate (FPR), False Negative Rate (FNR), and computation time.

To comprehensively analyze the impact of feature selection on model performance, a script was developed to systematically loop through all possible combinations of ranking methods, feature selection techniques, model types (Random Forest and XGBoost), and averaging strategies (micro, macro, weighted). This ensured a thorough evaluation of different configurations, allowing for the identification of the optimal approach that maximizes predictive accuracy while minimizing feature redundancy.

A diagram of a structure

AI-generated content may be incorrect.

Figure 8 Flowchart of model evaluation process

# Results and Discussion

## Overview of Result Metrics

The performance of the models was evaluated using standard classification metrics, including Accuracy, Precision, Recall, F1 Score, False Positive Rate (FPR), and False Negative Rate (FNR). The results were analyzed across different feature selection methods and varying numbers of selected features to assess the trade-off between model complexity and predictive performance.

One of the key findings from this analysis is the impact of feature reduction on model performance. The F1 Score, a balanced measure of precision and recall, was used as the primary performance metric for evaluation. The figure below illustrates how the F1 Score changes as the number of selected features decreases.

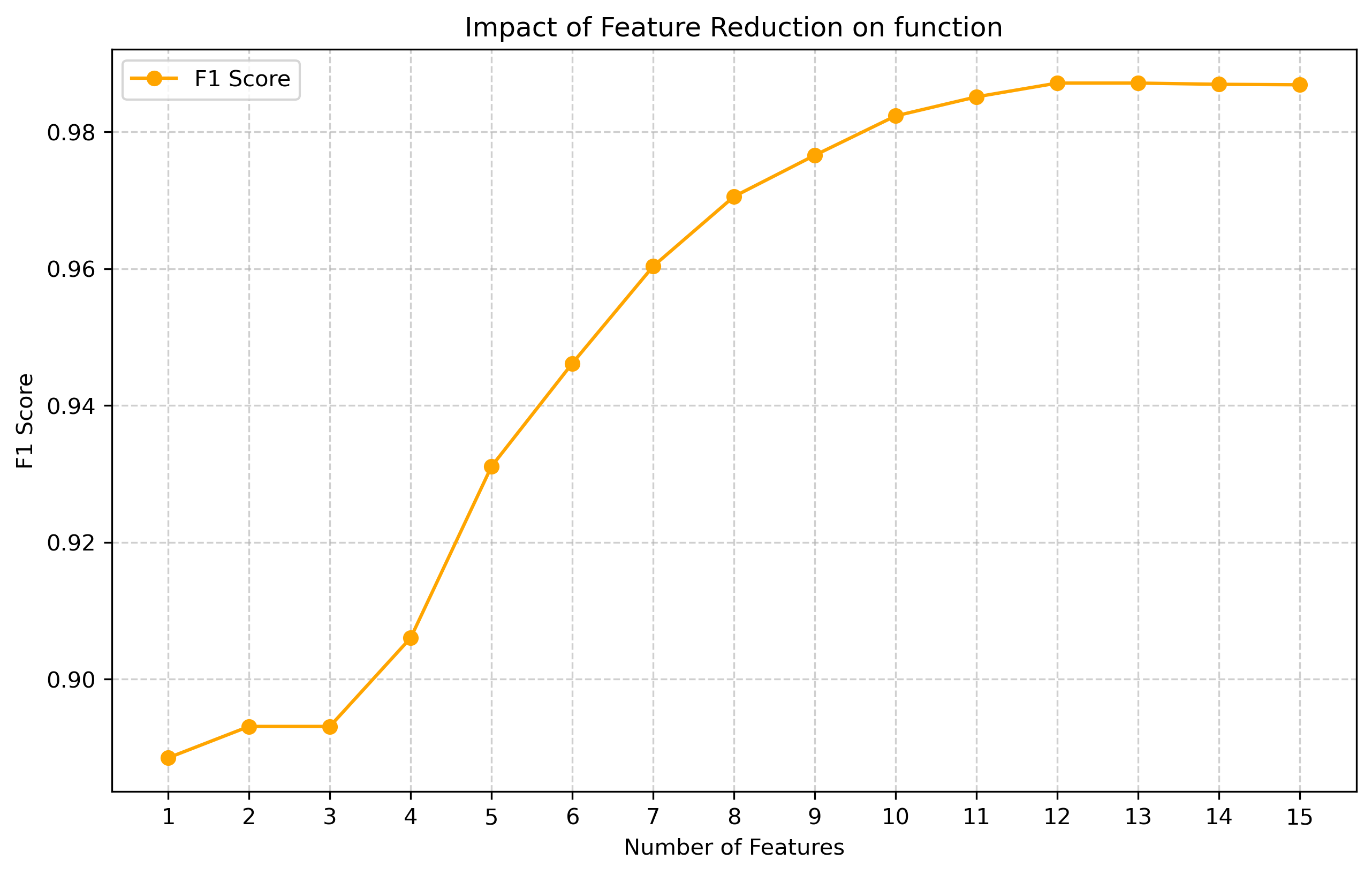


Figure 9 Impact of Feature Reduction on Model Performance (using SP and Micro Averaging)

From the graph, we can observe that the F1 Score remains consistently high when using 10 or more features. As the number of selected features decreases from 10 to 6, there is a gradual decline in performance, which suggests that some features, while not critical, still contribute to overall predictive accuracy. However, when fewer than 6 features are used, there is a significant drop in F1 Score, indicating that certain features are essential for maintaining classification performance. This trend highlights the importance of selecting an optimal number of features to balance predictive accuracy with computational efficiency. By carefully applying feature selection techniques, we can ensure minimal loss in classification performance while improving efficiency in model training and inference.

### Key Performance Metrics

The following metrics were used to evaluate model performance:

* **Accuracy**: The percentage of correctly classified instances.
* **Precision**: The ratio of correctly predicted positive observations to total predicted positives.
* **Recall**: The ratio of correctly predicted positive observations to actual positives.
* **F1 Score**: The harmonic mean of precision and recall, providing a balanced assessment of model performance.
* **False Positive Rate (FPR)**: The proportion of false positives among all negative cases.
* **False Negative Rate (FNR)**: The proportion of false negatives among all positive cases.
* **Time Taken**: The computational time required to train and evaluate the model.

### Performance Trends

A deeper analysis of performance trends revealed several key insights. The most noticeable decline in model performance occurred when the number of selected features fell below seven. This suggests that removing redundant features can enhance efficiency, eliminating too many features negatively impacts classification accuracy. Models trained with 11 to 15 features consistently achieved the highest performance, reinforcing the importance of feature selection.

A comparison between Random Forest (RF) and XGBoost (XGB) models showed that both performed robustly across different feature sets. However, XGBoost models slightly outperformed Random Forest, particularly when a larger number of features were included. When fewer than seven features were selected, the performance of XGBoost models deteriorated more rapidly than Random Forest, indicating that Random Forest models are more resilient to feature reduction.

Feature ranking methods also played a crucial role in determining the most informative feature subsets. Ranking techniques such as Statistical Properties (SP) and Weighted Feature Importance (WFI) provided valuable insights into which features contributed most to model performance. Sequential Progressive Feature Selection (SPFS) and Recursive Feature Elimination (RFE) were effective in selecting feature subsets that retained high predictive accuracy. Interestingly, models that used removal-based methods (rem\_1 to rem\_5) exhibited only marginal decreases in performance, suggesting that a small number of features could be safely discarded. However, removing more than six features resulted in a significant decline in accuracy, confirming the necessity of certain features for maintaining model success.

### Comparison by Model Type, Ranking Method, and Feature Selection

To determine the most effective feature selection and ranking methods, we analyzed the highest-performing models across different configurations. The findings indicate that models with 11 to 15 selected features consistently achieved the highest F1 scores, with only minor differences between different feature selection strategies. The highest recorded F1 score of 0.991805 was achieved using the Recursive Feature Elimination (RFE) method with the Statistical Properties (SP) ranking approach in a Random Forest (RF) model.

Both SPFS and RFE demonstrated strong performance in feature selection, with RFE slightly outperforming SPFS in some cases. Removing low-importance features (e.g., rem\_1 to rem\_5) resulted in models that retained high F1 scores, indicating that unnecessary features could be discarded with minimal performance loss. However, removing more than six features led to a sharp decline in accuracy and F1 Score, emphasizing the importance of key predictive features.

A comparison between Random Forest and XGBoost models revealed that both classifiers performed well across different feature sets, though Random Forest was more resilient to feature reduction. When fewer features were used, XGBoost models experienced more pronounced performance degradation, whereas Random Forest maintained stable predictive accuracy.

Ranking methods also had a notable impact on model performance. The Statistical Properties (SP) and Weighted Feature Importance (WFI) ranking methods provided meaningful insights into feature importance, enabling more effective feature selection. Retaining only top-ranked features often led to significant performance degradation, highlighting the importance of including certain lower-ranked features.

## Ranking Results

### Statistical Properties Ranking Results

Below are the results of the ranking using the SP method, which evaluates features based on statistical properties such as Standard Deviation, Absolute Difference, Skewness, and Kurtosis. The Total Rank is derived by summing the individual rankings, with a lower total rank indicating higher feature importance.

Table 5 Ranking features in function dataset based on statistical properties

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Std. Dev | Abs. Diff. | Skew. | Kurtosis | Std. Dev. Rank | Abs. Diff. Rank | Skew. Rank | Kurtosis Rank | Total Rank |
| address | 0.020107 | 0.000383 | 21.10314 | 695.9677 | 1 | 1 | 1 | 15 | 18 |
| pressure\_measurement | 0.040258 | 0.002154 | 20.20244 | 419.6274 | 2 | 2 | 2 | 14 | 20 |
| reset\_rate | 0.080048 | 0.003066 | 3.008437 | 20.3115 | 5 | 3 | 5 | 9 | 22 |
| setpoint | 0.137378 | 0.004575 | 0.959121 | 2.900798 | 8 | 4 | 8 | 7 | 27 |
| cycle\_time | 0.078341 | 0.026718 | 2.4926 | 21.71307 | 4 | 8 | 6 | 10 | 28 |
| rate | 0.085772 | 0.011896 | 8.086138 | 69.05565 | 6 | 6 | 3 | 13 | 28 |
| function | 0.136961 | 0.030515 | 5.122153 | 27.34285 | 7 | 9 | 4 | 12 | 32 |
| deadband | 0.172885 | 0.018316 | 0.171086 | -0.96751 | 9 | 7 | 11 | 5 | 32 |
| gain | 0.072631 | 0.008429 | -4.83451 | 26.2416 | 3 | 5 | 15 | 11 | 34 |
| system\_mode | 0.397895 | 0.246882 | 1.161342 | -0.41861 | 11 | 11 | 7 | 6 | 35 |
| crc\_rate | 0.424578 | 0.27986 | 0.091263 | -1.97471 | 12 | 12 | 12 | 1 | 37 |
| pump | 0.466624 | 0.320379 | 0.769877 | -1.40729 | 13 | 13 | 9 | 4 | 39 |
| length | 0.186097 | 0.041966 | -4.47194 | 18.80183 | 10 | 10 | 14 | 8 | 42 |
| solenoid | 0.497716 | 0.452241 | 0.191915 | -1.96317 | 15 | 15 | 10 | 2 | 42 |
| control\_scheme | 0.473766 | 0.340163 | -0.67475 | -1.54471 | 14 | 14 | 13 | 3 | 44 |

The most important features identified in this ranking analysis are *Address*, *Pressure Measurement*, and *Reset Rate*. *Address* emerged as the most significant feature, achieving the lowest total rank of 18 by ranking first across all individual statistical metrics. *Pressure Measurement* followed closely with a total rank of 20, and *Reset Rate* ranked 22, indicating that these features strongly contribute to the dataset’s variability and predictive power. Their consistent high rankings suggest they should be prioritized in feature selection and model development.

In contrast, moderately important features such as *Setpoint*, *Cycle Time*, and *Rate* demonstrated mid-range rankings, with total ranks between 27 and 28. While these features contribute to the dataset’s structure, their influence appears to be less significant than the top-ranked features. They may still hold some value in predictive modeling but might be considered for removal if computational efficiency or model simplicity is a concern.

The least important features included *Control Scheme*, *Solenoid*, and *Length*, each receiving the highest total ranks (42-44). These features exhibited minimal impact across all statistical metrics, suggesting they may be redundant. Given their low importance, they could potentially be removed during feature selection without significantly affecting model performance.

These findings have direct implications for feature selection. *Address*, *Pressure Measurement*, and *Reset Rate* should be retained across all models due to their strong contribution to predictive accuracy. On the other hand, features such as *Control Scheme*, *Solenoid*, and *Length* can be considered for removal when reducing dimensionality, as they exhibit the least impact. This ranking provides valuable insights into how different features influence the dataset, offering a data-driven approach to optimizing feature selection and improving model efficiency.

### Weighted feature selection ranking results using RandomForest Classifier

This section presents the feature importance results obtained using a RandomForest classifier. Unlike the SP method—which relies on statistical properties—the RandomForest ranking quantifies the importance of each feature based on how much it reduces uncertainty (impurity) in the decision trees. The importance scores are derived from the contribution of each feature to the model's overall predictive performance. A higher feature importance score indicates that the feature plays a more critical role in making accurate predictions. The following table (Table 6) shows the ranked features for the function dataset using the RandomForest classifier.

Table 6 Feature importance rankings for the function dataset using the RandomForest classifier

|  |  |
| --- | --- |
| Name | Random Forest Feature Importance |
| function | 0.170343 |
| crc\_rate | 0.14766 |
| length | 0.135827 |
| cycle\_time | 0.094819 |
| reset\_rate | 0.093911 |
| setpoint | 0.088682 |
| gain | 0.074429 |
| deadband | 0.071171 |
| rate | 0.047493 |
| pump | 0.027476 |
| system\_mode | 0.025344 |
| control\_scheme | 0.008638 |
| solenoid | 0.006637 |
| address | 0.006363 |
| pressure\_measurement | 0.001208 |

The feature-importance results obtained from the Random Forest model provide a different perspective on the significance of various features compared to the SP ranking method. According to this analysis, the most influential features were Function (Feature Importance = 0.1703), CRC Rate (0.1477), and Length (0.1358). These features were assigned the highest importance scores by the Random Forest model, indicating that they played a crucial role in the predictive performance. Their strong feature importance suggests that they should be retained when using feature selection methods based on Random Forest rankings.

A set of moderately important features followed, including Cycle Time (0.0948), Reset Rate (0.0939), Setpoint (0.0887), and Gain (0.0744). These features contributed to model performance but to a lesser extent than the top-ranked ones. While they may still provide valuable information, their importance may vary depending on the specific model and dataset characteristics.

On the other hand, the least important features were Address (0.0064) and Pressure Measurement (0.0012), along with Solenoid (0.0066), Control Scheme (0.0086), and System Mode (0.0253). These features had significantly lower importance scores, indicating that they had minimal impact on the model’s decision-making process. Given their low importance, these features could be considered for removal during feature selection, as they may not contribute significantly to model performance when using Random Forest as the ranking method.

These findings highlight how different feature ranking techniques can yield varying results. Unlike the SP ranking method, which identified Address and Pressure Measurement as the most important features, the Random Forest model ranked these features as the least significant. This contrast underscores the importance of selecting a ranking method that aligns with the specific modeling approach being used. For Random Forest-based feature selection, Function, CRC Rate, and Length should be prioritized, whereas Address and Pressure Measurement may be less relevant.

### Weighted feature selection ranking results using XGBoost Classifier

This section details the feature importance rankings as determined by an XGBoost classifier. XGBoost is a gradient boosting algorithm that evaluates the contribution of each feature based on the improvement it brings to the model’s loss function. The resulting importance scores reflect the degree to which each feature influences predictive accuracy. In this context, higher scores indicate a more substantial impact on model performance. Table 7 summarizes the feature importance rankings for the function dataset using the XGBoost classifier.

Table 7 Feature importance rankings for the function dataset using the XGBoost classifier

|  |  |
| --- | --- |
| Name | XGBoost Feature Importance |
| length | 0.349779 |
| function | 0.183473 |
| crc\_rate | 0.175295 |
| rate | 0.154045 |
| pump | 0.032862 |
| gain | 0.021015 |
| cycle\_time | 0.018802 |
| system\_mode | 0.013917 |
| reset\_rate | 0.011271 |
| setpoint | 0.011252 |
| deadband | 0.009006 |
| control\_scheme | 0.008318 |
| solenoid | 0.0062 |
| pressure\_measurement | 0.004765 |
| address | 0 |

The feature importance results obtained from the XGBoost model reveal a distinct prioritization of features compared to both the SP ranking method and the Random Forest model. According to XGBoost, the most influential feature was Length (Feature Importance = 0.3498), followed by Function (0.1835) and CRC Rate (0.1753). These three features received the highest importance scores, suggesting they play a crucial role in the model’s predictive performance. Given their strong influence, these features should be prioritized in feature selection when using XGBoost-based ranking methods.

A set of moderately important features followed, including Rate (0.1540), Pump (0.0329), Gain (0.0210), and Cycle Time (0.0188). While these features were not as dominant as the top three, they still contributed meaningfully to the model’s decision-making. Retaining these features may improve performance, though their impact might be more context-dependent.

The least important features in the XGBoost analysis were Solenoid (0.0062), Pressure Measurement (0.0048), and Address (0.0000). Address, which was ranked highly in the SP ranking method, was found to have no importance in the XGBoost model. Similarly, Pressure Measurement, which was among the most significant features in SP but least important in Random Forest, was again found to be one of the least valuable features under XGBoost. This suggests that these features may not contribute meaningful information when using XGBoost as a model, reinforcing the need to tailor feature selection based on the specific ranking methodology being used.

Overall, the XGBoost model places strong importance on Length, Function, and CRC Rate, while deeming Address and Pressure Measurement as nearly irrelevant. This contrasts sharply with both the SP ranking method, where Address was the top feature, and the Random Forest ranking, which found Function and CRC Rate important but ranked Address and Pressure Measurement among the lowest. These variations highlight the importance of selecting a ranking method that aligns with the underlying model being used for classification.

## ****Summary of Best-Performing Models****

### Best F1 Scores

To identify the most effective model configurations, we analyzed various feature selection methods, classifiers, and ranking approaches. Table 8 presents the **top 30 models ranked by F1 score**, highlighting optimal configurations for intrusion detection.

Table 8 Top 30 best preforming combinations based on greatest F1 score

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Rank Method | Feature Selection | Model Type | Average Type | Number of Features | Removed Features | F1 Score | Ranking Time | Feature Selection Time | Eval. Time | Total Time |
| SP | RFE | RF | micro | 13 | crc\_rate, pressure\_measurement | 0.9871 | 0.27 | 140.38 | 67.12 | 207.78 |
| WFI-XGB | RFE | RF | micro | 12 | function, pressure\_measurement, address | 0.9871 | 3.54 | 118.17 | 67.25 | 188.97 |
| SP | None | RF | micro | 15 |  | 0.987 | 0.27 | 8.35 | 72.78 | 81.40 |
| WFI-RF | None | RF | micro | 15 |  | 0.987 | 7.61 | 8.11 | 73.41 | 89.13 |
| WFI-XGB | None | RF | micro | 15 |  | 0.987 | 3.54 | 8.26 | 74.10 | 85.91 |
| SP | RFE | RF | weighted | 13 | crc\_rate, pressure\_measurement | 0.9869 | 0.27 | 140.38 | 66.52 | 207.18 |
| WFI-XGB | RFE | RF | weighted | 12 | function, pressure\_measurement, address | 0.9869 | 3.54 | 118.17 | 68.03 | 189.75 |
| WFI-RF | rem\_1 | RF | micro | 14 | pressure\_measurement | 0.9869 | 7.61 | 8.11 | 62.77 | 78.50 |
| WFI-XGB | rem\_1 | RF | micro | 14 | address | 0.9869 | 3.54 | 8.26 | 72.81 | 84.61 |
| WFI-XGB | SPFS | RF | micro | 15 |  | 0.9869 | 3.54 | 70.69 | 74.46 | 148.69 |
| WFI-XGB | rem\_2 | RF | micro | 13 | pressure\_measurement, address | 0.9868 | 3.54 | 8.26 | 65.85 | 77.65 |
| WFI-RF | None | RF | weighted | 15 |  | 0.9867 | 7.61 | 8.11 | 74.23 | 89.96 |
| SP | None | RF | weighted | 15 |  | 0.9867 | 0.27 | 8.35 | 74.61 | 83.23 |
| WFI-XGB | None | RF | weighted | 15 |  | 0.9867 | 3.54 | 8.26 | 72.91 | 84.72 |
| WFI-RF | rem\_1 | RF | weighted | 14 | pressure\_measurement | 0.9867 | 7.61 | 8.11 | 61.37 | 77.09 |
| WFI-XGB | rem\_1 | RF | weighted | 14 | address | 0.9867 | 3.54 | 8.26 | 77.16 | 88.96 |
| WFI-RF | rem\_2 | RF | micro | 13 | pressure\_measurement, address | 0.9867 | 7.61 | 8.11 | 64.98 | 80.70 |
| WFI-RF | RFE | RF | micro | 12 | crc\_rate, pressure\_measurement, address | 0.9867 | 7.61 | 119.02 | 66.42 | 193.04 |
| WFI-XGB | SPFS | RF | weighted | 15 |  | 0.9866 | 3.54 | 70.69 | 74.01 | 148.24 |
| WFI-RF | SPFS | RF | micro | 15 |  | 0.9866 | 7.61 | 89.74 | 74.15 | 171.50 |
| SP | SPFS | RF | micro | 15 |  | 0.9866 | 0.27 | 139.50 | 75.16 | 214.93 |
| SP | rem\_1 | RF | micro | 14 | control\_scheme | 0.9866 | 0.27 | 8.35 | 76.69 | 85.31 |
| WFI-XGB | rem\_2 | RF | weighted | 13 | pressure\_measurement, address | 0.9865 | 3.54 | 8.26 | 64.52 | 76.33 |
| WFI-RF | RFE | RF | weighted | 12 | crc\_rate, pressure\_measurement, address | 0.9865 | 7.61 | 119.02 | 65.53 | 192.16 |
| WFI-RF | rem\_2 | RF | weighted | 13 | pressure\_measurement, address | 0.9865 | 7.61 | 8.11 | 65.10 | 80.83 |
| SP | SPFS | RF | weighted | 15 |  | 0.9863 | 0.27 | 139.50 | 72.74 | 212.51 |
| WFI-RF | SPFS | RF | weighted | 15 |  | 0.9863 | 7.61 | 89.74 | 72.23 | 169.58 |
| SP | rem\_1 | RF | weighted | 14 | control\_scheme | 0.9862 | 0.27 | 8.35 | 75.18 | 83.80 |
| WFI-RF | rem\_3 | RF | micro | 12 | solenoid, pressure\_measurement, address | 0.9855 | 7.61 | 8.11 | 64.55 | 80.28 |
| WFI-XGB | rem\_3 | RF | micro | 12 | solenoid, pressure\_measurement, address | 0.9854 | 3.54 | 8.26 | 64.85 | 76.65 |

Based on the best preforming models, Random Forest (RF) models consistently outperformed XGBoost (XGB), achieving a maximum F1 score of 0.9871, compared to 0.9853 for XGB. However, RF models required three times longer to evaluate and train than XGB, adding approximately 100 extra seconds to the total process.

Feature selection methods significantly influenced performance and runtime. Recursive Feature Elimination (RFE) yielded the highest F1 scores but was computationally expensive, often taking over 140 seconds for feature selection alone. In contrast, directly removing the least important features (e.g., crc\_rate, pressure\_measurement, function, and address) achieved similar performance with much lower computational cost.

When calculating the F1 score during cross-validation, the micro-average method slightly outperformed the weighted approach across top models. The best-performing configurations used 12 to 15 features, with models retaining 13 or 14 features yielding optimal results while maintaining reasonable evaluation times.

Overall, RF models provided superior accuracy, but XGB offered a significantly faster alternative.

### Best accuracy

Tabel 9 ranks the top 30 model configurations based on accuracy, highlighting the trade-offs between feature selection methods, classifier choice, and computational efficiency.

Table 9 Best 30 accuracy scores (Micro Averaging)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Rank Method | Feature Selection | Model Type | Number of Features | Removed Features | Accuracy | F1 Score | Ranking Time | Feature Selection Time | Evaluation Time | Total Time |
| SP | RFE | RF | 13 | crc\_rate, pressure\_measurement | 0.9871 | 0.9871 | 0.27 | 140.38 | 67.12 | 207.78 |
| WFI-XGB | RFE | RF | 12 | function, pressure\_measurement, address | 0.9871 | 0.9871 | 3.54 | 118.17 | 67.25 | 188.97 |
| SP | None | RF | 15 |  | 0.987 | 0.987 | 0.27 | 8.35 | 72.78 | 81.40 |
| WFI-RF | None | RF | 15 |  | 0.987 | 0.987 | 7.61 | 8.11 | 73.41 | 89.13 |
| WFI-XGB | None | RF | 15 |  | 0.987 | 0.987 | 3.54 | 8.26 | 74.10 | 85.91 |
| WFI-RF | rem\_1 | RF | 14 | pressure\_measurement | 0.9869 | 0.9869 | 7.61 | 8.11 | 62.77 | 78.50 |
| WFI-XGB | rem\_1 | RF | 14 | address | 0.9869 | 0.9869 | 3.54 | 8.26 | 72.81 | 84.61 |
| WFI-XGB | SPFS | RF | 15 |  | 0.9869 | 0.9869 | 3.54 | 70.69 | 74.46 | 148.69 |
| WFI-XGB | rem\_2 | RF | 13 | pressure\_measurement, address | 0.9868 | 0.9868 | 3.54 | 8.26 | 65.85 | 77.65 |
| WFI-RF | rem\_2 | RF | 13 | pressure\_measurement, address | 0.9867 | 0.9867 | 7.61 | 8.11 | 64.98 | 80.70 |
| WFI-RF | RFE | RF | 12 | crc\_rate, pressure\_measurement, address | 0.9867 | 0.9867 | 7.61 | 119.02 | 66.42 | 193.04 |
| WFI-RF | SPFS | RF | 15 |  | 0.9866 | 0.9866 | 7.61 | 89.74 | 74.15 | 171.50 |
| SP | SPFS | RF | 15 |  | 0.9866 | 0.9866 | 0.27 | 139.50 | 75.16 | 214.93 |
| SP | rem\_1 | RF | 14 | control\_scheme | 0.9866 | 0.9866 | 0.27 | 8.35 | 76.69 | 85.31 |
| WFI-RF | rem\_3 | RF | 12 | solenoid, pressure\_measurement, address | 0.9855 | 0.9855 | 7.61 | 8.11 | 64.55 | 80.28 |
| WFI-XGB | rem\_3 | RF | 12 | solenoid, pressure\_measurement, address | 0.9854 | 0.9854 | 3.54 | 8.26 | 64.85 | 76.65 |
| SP | None | XGB | 15 |  | 0.9853 | 0.9853 | 0.27 | 2.65 | 21.84 | 24.76 |
| WFI-RF | None | XGB | 15 |  | 0.9853 | 0.9853 | 7.61 | 2.73 | 23.53 | 33.87 |
| WFI-XGB | None | XGB | 15 |  | 0.9853 | 0.9853 | 3.54 | 2.86 | 24.29 | 30.69 |
| SP | SPFS | XGB | 15 |  | 0.9852 | 0.9852 | 0.27 | 30.72 | 21.60 | 52.60 |
| WFI-RF | SPFS | XGB | 15 |  | 0.9852 | 0.9852 | 7.61 | 30.97 | 23.81 | 62.40 |
| WFI-XGB | rem\_4 | RF | 11 | solenoid, address, pressure\_measurement, control\_scheme | 0.9851 | 0.9851 | 3.54 | 8.26 | 68.31 | 80.12 |
| WFI-XGB | rem\_1 | XGB | 14 | address | 0.985 | 0.985 | 3.54 | 2.86 | 21.52 | 27.92 |
| WFI-XGB | SPFS | XGB | 15 |  | 0.985 | 0.985 | 3.54 | 34.77 | 25.16 | 63.47 |
| WFI-XGB | RFE | XGB | 14 | address | 0.985 | 0.985 | 3.54 | 46.49 | 26.57 | 76.60 |
| WFI-RF | rem\_4 | RF | 11 | solenoid, address, pressure\_measurement, control\_scheme | 0.985 | 0.985 | 7.61 | 8.11 | 67.49 | 83.22 |
| SP | rem\_2 | RF | 13 | solenoid, control\_scheme | 0.9849 | 0.9849 | 0.27 | 8.35 | 81.36 | 89.99 |
| SP | rem\_1 | XGB | 14 | control\_scheme | 0.9847 | 0.9847 | 0.27 | 2.65 | 21.57 | 24.49 |
| SP | RFE | XGB | 13 | function, address | 0.9846 | 0.9846 | 0.27 | 36.71 | 21.50 | 58.48 |
| SP | rem\_3 | RF | 12 | solenoid, length, control\_scheme | 0.9846 | 0.9846 | 0.27 | 8.35 | 81.72 | 90.34 |

Similar to the F1 score rankings, Random Forest (RF) models outperformed XGBoost (XGB) in terms of accuracy, with RF achieving a maximum accuracy of 0.9871, compared to 0.9853 for XGB. However, RF models took three times longer to evaluate and required about 100 extra seconds to complete the entire process compared to XGB.

To improve clarity, the table has been reduced to only display micro-averaged results, as micro and weighted averaging performed very similarly across all models.

Recursive Feature Elimination (RFE) consistently achieved the highest accuracy but at a significant computational cost, often exceeding 140 seconds for feature selection alone when using a Random Forest model. Selective Progressive Feature Selection (SPFS) also performed well, but, like RFE, required substantial computation time. However, when using XGBoost, the feature selection time significantly decreased due to its faster model training speed. For example, RFE with Random Forest took approximately 119 seconds, while RFE with XGBoost only took around 46 seconds, as shown below. This demonstrates that XGBoost trains nearly three times faster than Random Forest (RF), leading to a substantial reduction in feature selection time. This trend is also evident in evaluation time, where XGBoost consistently requires less time to evaluate models compared to RF, further reinforcing its efficiency advantage. This will be explored further in section 8.3.4.

### Performance of Models with Limited Features

To analyze how well models perform when using a reduced number of features, we filtered the dataset to include only models that retained fewer than 11 features. The goal of this analysis was to determine which feature selection methods and ranking techniques maintained high performance despite significant feature reduction. The results were sorted by F1 Score in descending order to identify the best-performing configurations.

The results indicate that Random Forest (RF) consistently outperformed XGBoost (XGB) when using a reduced feature set. Across all top-performing models with fewer than 11 features, RF was the preferred model type, achieving F1 scores above 0.97 in most cases. Additionally, SP and WFI-RF ranking methods performed well, with SP-based feature selection leading to the highest F1 score when 11 features were used.

Table 10 presents the best-performing models under feature-limited conditions:

Table 10 Best preforming combinations when reducing the number of features to 11 or less (micro averaging)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Rank Method | Model Type | Number of Features | Removed Features | F1 Score | Ranking Time | Feature Selection Time | Evaluation Time | Total Time |
| WFI-XGB | RF | 11 | solenoid, address, pressure\_measurement, control\_scheme | 0.9851 | 3.54 | 8.26 | 68.31 | 80.12 |
| WFI-RF | RF | 11 | solenoid, address, pressure\_measurement, control\_scheme | 0.9850 | 7.61 | 8.11 | 67.49 | 83.22 |
| SP | RF | 11 | pump, solenoid, length, control\_scheme | 0.9823 | 0.27 | 8.35 | 86.61 | 95.23 |
| SP | RF | 10 | pump, control\_scheme, solenoid, crc\_rate, length | 0.9823 | 0.27 | 8.35 | 96.71 | 105.34 |
| WFI-RF | RF | 10 | system\_mode, control\_scheme, solenoid, pressure\_measurement, address | 0.9816 | 7.61 | 8.11 | 67.97 | 83.70 |
| WFI-XGB | XGB | 11 | solenoid, address, pressure\_measurement, control\_scheme | 0.9809 | 3.54 | 2.86 | 19.77 | 26.17 |
| WFI-RF | XGB | 11 | solenoid, address, pressure\_measurement, control\_scheme | 0.9809 | 7.61 | 2.73 | 21.70 | 32.04 |
| SP | XGB | 11 | pump, solenoid, length, control\_scheme | 0.9803 | 0.27 | 2.65 | 20.53 | 23.45 |
| SP | XGB | 10 | pump, control\_scheme, solenoid, crc\_rate, length | 0.9802 | 0.27 | 2.65 | 20.21 | 23.13 |
| WFI-XGB | RF | 10 | deadband, control\_scheme, solenoid, pressure\_measurement, address | 0.9775 | 3.54 | 8.26 | 60.55 | 72.36 |
| WFI-RF | XGB | 10 | system\_mode, control\_scheme, solenoid, pressure\_measurement, address | 0.9767 | 7.61 | 2.73 | 21.15 | 31.49 |
| WFI-RF | RF | 9 | system\_mode, pump, control\_scheme, solenoid, pressure\_measurement, address | 0.9766 | 7.61 | 8.11 | 72.10 | 87.83 |
| SP | RF | 9 | system\_mode, pump, control\_scheme, solenoid, crc\_rate, length | 0.9764 | 0.27 | 8.35 | 102.21 | 110.83 |
| SP | XGB | 9 | system\_mode, pump, control\_scheme, solenoid, crc\_rate, length | 0.9751 | 0.27 | 2.65 | 19.24 | 22.16 |
| WFI-XGB | XGB | 10 | deadband, control\_scheme, solenoid, pressure\_measurement, address | 0.974 | 3.54 | 2.86 | 20.26 | 26.66 |
| WFI-RF | RF | 8 | system\_mode, pump, control\_scheme, solenoid, pressure\_measurement, rate, address | 0.9705 | 7.61 | 8.11 | 59.29 | 75.02 |
| WFI-RF | XGB | 9 | system\_mode, pump, control\_scheme, solenoid, pressure\_measurement, address | 0.9703 | 7.61 | 2.73 | 21.22 | 31.55 |
| WFI-XGB | RF | 9 | deadband, setpoint, control\_scheme, solenoid, pressure\_measurement, address | 0.9696 | 3.54 | 8.26 | 50.15 | 61.95 |
| WFI-XGB | XGB | 9 | deadband, setpoint, control\_scheme, solenoid, pressure\_measurement, address | 0.9651 | 3.54 | 2.86 | 19.48 | 25.88 |
| SP | RF | 8 | system\_mode, pump, control\_scheme, solenoid, crc\_rate, gain, length | 0.965 | 0.27 | 8.35 | 84.30 | 92.92 |
| SP | XGB | 8 | system\_mode, pump, control\_scheme, solenoid, crc\_rate, gain, length | 0.9639 | 0.27 | 2.65 | 19.31 | 22.23 |
| WFI-RF | XGB | 8 | system\_mode, pump, control\_scheme, solenoid, pressure\_measurement, rate, address | 0.9628 | 7.61 | 2.73 | 21.55 | 31.89 |
| WFI-RF | RF | 7 | system\_mode, pump, deadband, control\_scheme, solenoid, pressure\_measurement, rate, address | 0.9603 | 7.61 | 8.11 | 55.47 | 71.19 |
| SP | XGB | 7 | system\_mode, pump, deadband, control\_scheme, solenoid, crc\_rate, gain, length | 0.9561 | 0.27 | 2.65 | 18.89 | 21.81 |
| WFI-RF | XGB | 7 | system\_mode, pump, deadband, control\_scheme, solenoid, pressure\_measurement, rate, address | 0.9534 | 7.61 | 2.73 | 20.80 | 31.14 |
| SP | RF | 7 | system\_mode, pump, deadband, control\_scheme, solenoid, crc\_rate, gain, length | 0.952 | 0.27 | 8.35 | 76.31 | 84.94 |
| WFI-XGB | RF | 8 | deadband, setpoint, control\_scheme, solenoid, pressure\_measurement, reset\_rate, address | 0.9472 | 3.54 | 8.26 | 32.85 | 44.66 |
| WFI-RF | RF | 6 | system\_mode, pump, deadband, control\_scheme, solenoid, pressure\_measurement, rate, gain, address | 0.9461 | 7.61 | 8.11 | 57.43 | 73.15 |
| WFI-XGB | XGB | 8 | deadband, setpoint, control\_scheme, solenoid, pressure\_measurement, reset\_rate, address | 0.9456 | 3.54 | 2.86 | 18.13 | 24.53 |
| WFI-RF | XGB | 6 | system\_mode, pump, deadband, control\_scheme, solenoid, pressure\_measurement, rate, gain, address | 0.9399 | 7.61 | 2.73 | 19.15 | 29.48 |

The results show that Random Forest (RF) consistently outperformed XGBoost (XGB) when using fewer than 11 features, demonstrating greater resilience to feature reduction. RF maintained an F1 score above 0.97, even when the number of features was reduced to seven, highlighting its ability to handle reduced-dimensional datasets effectively.

Among the ranking methods, WFI-XGB and WFI-RF produced the highest F1 scores. The best-performing model, WFI-XGB with RF and 11 features, achieved an F1 score of 0.9851, closely followed by WFI-RF with RF at 0.9850, confirming that Weighted Feature Importance (WFI) is highly effective in reduced feature scenarios, especially with RF.

Feature elimination trends indicate that removing up to 6 features had minimal impact, keeping F1 scores near 0.98. However, performance declined more noticeably beyond 7-9 removed features, suggesting a threshold where further reduction harms classification accuracy. The most frequently removed features in high-performing models included Control Scheme, Solenoid, and Pump, indicating their lower contribution to predictive accuracy.

Since all feature selection methods involved removing a set number of least important features, the feature selection time column in Table 10 essentially represents model creation time. This means the time listed under "Feature Selection Time" accounts for both ranking and model creation.

XGBoost struggled slightly as the number of features dropped below 11, with gradual declines in F1 score compared to RF, though it still achieved F1 scores above 0.96 in most cases. Additionally, XGBoost training times varied between 2.65 and 8+ seconds, highlighting inconsistencies in handling different feature subsets. In contrast, RF remained stable above 0.97 with 7-11 features, confirming its superior resilience to feature elimination.

Overall, RF is the better choice for models with fewer features, as it maintains strong predictive performance even with aggressive feature reduction. However, XGBoost remains competitive, though it benefits more from a larger feature set.

### Performance on time taken

To evaluate the efficiency of different model configurations, we analyzed the time required for ranking, feature selection, and evaluation. Initially, models were ranked by F1 score, and the top 20 were selected. These models were then re-ranked by total time taken, ensuring that we prioritize models that achieve both high classification performance and computational efficiency, which is essential for real-time applications.

Table 11 Ranking of Model Configurations by Total Computational Time (Best 20 F1 Scores)

(micro averaging)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Rank Method | Feature Selection | Model Type | Number of Features | F1 Score | Ranking Time | Feature Selection Time | Evaluation Time | Total Time |
| SP | None | XGB | 15 | 0.9853 | 0.2713 | 2.6469 | 21.8385 | 24.7566 |
| WFI-XGB | None | XGB | 15 | 0.9853 | 3.5413 | 2.8595 | 24.2904 | 30.6913 |
| WFI-RF | None | XGB | 15 | 0.9853 | 7.6107 | 2.7278 | 23.5335 | 33.8719 |
| SP | SPFS | XGB | 15 | 0.9852 | 0.2713 | 30.7238 | 21.6013 | 52.5964 |
| WFI-XGB | rem\_3 | RF | 12 | 0.9854 | 3.5413 | 8.2624 | 64.8469 | 76.6506 |
| WFI-XGB | rem\_2 | RF | 13 | 0.9868 | 3.5413 | 8.2624 | 65.8471 | 77.6508 |
| WFI-RF | rem\_1 | RF | 14 | 0.9869 | 7.6107 | 8.1132 | 62.7743 | 78.4982 |
| WFI-RF | rem\_3 | RF | 12 | 0.9855 | 7.6107 | 8.1132 | 64.5516 | 80.2754 |
| WFI-RF | rem\_2 | RF | 13 | 0.9867 | 7.6107 | 8.1132 | 64.9786 | 80.7024 |
| SP | None | RF | 15 | 0.987 | 0.2713 | 8.3521 | 72.7778 | 81.4012 |
| WFI-XGB | rem\_1 | RF | 14 | 0.9869 | 3.5413 | 8.2624 | 72.8105 | 84.6142 |
| SP | rem\_1 | RF | 14 | 0.9866 | 0.2713 | 8.3521 | 76.689 | 85.3124 |
| WFI-XGB | None | RF | 15 | 0.987 | 3.5413 | 8.2624 | 74.1028 | 85.9065 |
| WFI-RF | None | RF | 15 | 0.987 | 7.6107 | 8.1132 | 73.407 | 89.1308 |
| WFI-XGB | SPFS | RF | 15 | 0.9869 | 3.5413 | 70.686 | 74.4607 | 148.6881 |
| WFI-RF | SPFS | RF | 15 | 0.9866 | 7.6107 | 89.7379 | 74.1535 | 171.502 |
| WFI-XGB | RFE | RF | 12 | 0.9871 | 3.5413 | 118.1741 | 67.2535 | 188.969 |
| WFI-RF | RFE | RF | 12 | 0.9867 | 7.6107 | 119.0163 | 66.4172 | 193.0441 |
| SP | RFE | RF | 13 | 0.9871 | 0.2713 | 140.3831 | 67.1213 | 207.7756 |
| SP | SPFS | RF | 15 | 0.9866 | 0.2713 | 139.4981 | 75.1577 | 214.9271 |

The findings reveal that XGBoost (XGB) models tend to have lower total processing times compared to Random Forest (RF) when no feature selection is applied. For example, the fastest model (SP with no feature selection on XGB) required only 24.76 seconds, making it the most efficient high-performing model. However, when feature selection is introduced, RF becomes the preferred model, maintaining high F1 scores with moderate processing times. The best RF model (WFI-XGB with RFE at 12 features) achieved an F1 score of 0.9871 in 188.97 seconds, demonstrating that while RF takes longer for feature selection, it performs strongly with fewer features.

A clear trade-off exists between feature selection and processing time. Models without feature selection (SP, WFI-XGB, and WFI-RF on XGB) required the least processing time, showing that skipping feature selection significantly improves computational efficiency. However, feature selection increases processing time, with methods like SPFS and RFE extending total times beyond 140 seconds, while models without feature selection completed in under 35 seconds.

If computational efficiency is the priority, models without feature selection (SP, WFI-XGB, and WFI-RF with XGB) are the fastest, taking under 35 seconds while achieving strong F1 scores. On the other hand, if feature reduction is required, RF models offer the best balance, maintaining high F1 scores (0.9871) with reasonable processing times (~188-214 seconds in RFE configurations). However, complex feature selection methods (SPFS, RFE) significantly increase processing time, making them less suitable for real-time applications where quick responses are necessary.

### Model Comparison

A direct comparison between XGBoost and Random Forest is shown in Table 12, which highlights the significant difference in computational time. Even though RF models achieved a higher accuracy, their total computation time was drastically higher than XGBoost models. For instance, an RF model using RFE with 9 features required 148.69 seconds, compared to a similar XGB model requiring only 63.47 seconds.

Table Time comparison between RF and XGBoost

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Rank Method | Feature Selection | Model Type | Average Type | Number of Features | Accuracy | F1 Score | Ranking Time | Feature Selection Time | Evaluation Time | Total Time |
| WFI-XGB | RFE | RF | micro | 12 | 0.9871 | 0.9871 | 3.5413 | 118.1741 | 67.2535 | 188.9690 |
| WFI-XGB | SPFS | RF | micro | 15 | 0.9869 | 0.9869 | 3.5413 | 70.6860 | 74.4607 | 148.6881 |
| WFI-XGB | RFE | XGB | micro | 14 | 0.9850 | 0.9850 | 3.5413 | 46.4898 | 26.5656 | 76.5967 |
| WFI-XGB | SPFS | XGB | micro | 15 | 0.9850 | 0.9850 | 3.5413 | 34.7657 | 25.1620 | 63.4690 |

In summary, XGBoost is the best choice for fast, high-performing models without feature selection, whereas Random Forest excels in feature-limited scenarios but requires more processing time. The optimal model choice depends on the trade-off between speed and feature selection complexity.

## Summary of Findings

This study highlights the critical roles of feature selection, classifier type, and computational efficiency in model performance. Optimal model performance consistently occurred with 11 to 15 retained features, with notable declines in accuracy when fewer than 7 features were used. Thus, moderate feature elimination maintains accuracy, but excessive removal significantly reduces performance.

Recursive Feature Elimination (RFE) and Sequential Progressive Feature Selection (SPFS) effectively selected relevant features, with RFE achieving top F1 scores but at the expense of longer processing times, making it less practical for real-time applications. Direct removal of low-impact features such as address, pressure measurement, and crc\_rate provided a computationally efficient alternative while maintaining competitive accuracy.

Comparing classifiers, Random Forest (RF) consistently outperformed XGBoost (XGB) in scenarios with fewer than 11 features, maintaining high F1 scores (above 0.97) even when reduced to just 7 features. Conversely, XGB exhibited a greater dependency on larger feature sets, experiencing notable performance declines with fewer features.

Feature ranking methods significantly influenced outcomes, with Statistical Properties (SP) and Weighted Feature Importance (WFI) consistently identifying critical features. Simpler removal methods (rem\_1, rem\_2) offered comparable accuracy to RFE and SPFS at substantially lower computational costs.

XGB models provided the fastest processing times when no feature selection was used, ideal for time-sensitive applications. However, when feature selection was introduced, RF models offered the best balance of accuracy and efficiency, making them preferable for scenarios with limited features.

### Key Observations

The highest F1 scores were achieved by models retaining between 11 and 15 features, emphasizing the importance of careful feature selection, as removing more than six features significantly decreased performance. Random Forest (RF) models showed greater resilience to feature reduction compared to XGBoost (XGB), maintaining high accuracy even with fewer features. Feature ranking techniques like Statistical Properties (SP) and Weighted Feature Importance (WFI) consistently identified the most essential features. Direct removal of low-impact features (e.g., address, pressure measurement, control scheme) had minimal impact on accuracy but greatly enhanced computational efficiency. XGB models without feature selection delivered the fastest processing times, typically under 35 seconds, whereas RF models provided a superior balance of accuracy and efficiency when feature selection was applied. While Recursive Feature Elimination (RFE) improved accuracy, it significantly increased computational costs, often exceeding 140 seconds, making simpler removal methods more practical for efficiency-focused scenarios.

### Summary of Best-Performing Configurations

Optimal configurations utilized RF models combined with RFE or SPFS, with the top-performing model (WFI-XGB with RF and 12 features) achieving an F1 score of 0.9871. Eliminating features like address, pressure measurement, and crc\_rate improved computational efficiency without sacrificing accuracy.

XGB performed best in scenarios without feature selection, combining strong accuracy with rapid execution (24.76 seconds). Models retaining between 13 and 14 features offered an ideal balance of accuracy and efficiency.

Overall, XGB is ideal for fast classification without feature selection, while RF provides superior predictive accuracy when feature selection is involved.

The visualization below illustrates the best-performing configurations, highlighting their F1 scores. Models utilizing RFE and SPFS with RF demonstrated the highest classification performance, while XGBoost models without feature selection achieved the shortest processing times.

A graph with numbers and lines

AI-generated content may be incorrect.

Figure Graph showing all combinations best F1-Score

# Conclusion

The implementation of the RS-BMLMS (Robust Strategy for Best Machine Learning Model Selection) methodology has proven to be a systematic and effective approach for enhancing intrusion detection in gas pipeline SCADA systems. This study highlights the critical role of feature selection, ranking techniques, and machine learning model evaluation in optimizing predictive performance while ensuring computational efficiency. By exploring multiple feature selection methods, ranking strategies, and classification models, we have developed a framework that is both repeatable and reusable, ensuring robust anomaly detection in critical infrastructure.

One of the key contributions of this work is the thorough evaluation of feature ranking methods, particularly Statistical Properties (SP) and Weighted Feature Importance (WFI) using Random Forest and XGBoost. These ranking strategies were essential in identifying the most influential features, ensuring that selected models focused on the most relevant attributes while eliminating redundant or non-contributory data. The use of Recursive Feature Elimination (RFE) and Sequential Progressive Feature Selection (SPFS) further refined the feature selection process, enabling the development of high-performance models with a minimal yet highly informative feature set.

A key aspect of this study was the iterative feature reduction analysis, which systematically removed the least important features and assessed their impact on model performance. The results showed that models with 11 to 15 features consistently achieved the best F1 Scores. Notably, there was only a marginal performance drop when using rem\_x feature selection methods compared to RFE or SPFS. However, removing more than six features resulted in a sharp decline in accuracy, highlighting the necessity of retaining certain features to maintain classification effectiveness. This analysis reinforced the importance of thoughtful feature selection to optimize both performance and computational efficiency.

In terms of model performance, Random Forest (RF) and XGBoost (XGB) classifiers were extensively tested, with Random Forest showing greater resilience to feature reduction. Both models performed well with larger feature sets, but XGBoost models experienced a noticeable accuracy decline when fewer than 11 features were selected. This suggests that Random Forest is more adaptable to feature selection strategies, making it a preferred choice for SCADA intrusion detection applications, particularly in environments where feature reduction is necessary.

The study also revealed a clear trade-off between model accuracy and computational efficiency. XGBoost models achieved significantly lower total processing times compared to Random Forest when no feature selection was applied, with the fastest model (SP with no feature selection on XGB) completing in just 24.76 seconds. However, when feature selection was introduced, RF models provided a better balance between accuracy and computational cost, particularly in feature-limited scenarios. The best RF model (WFI-XGB with RFE, 12 features) reached an F1 score of 0.9871, though at the expense of a total processing time of 188.97 seconds.

Overall, XGBoost is the best choice for high-speed classification tasks without feature selection, while Random Forest excels when feature selection is necessary, offering higher accuracy at a computational cost. The findings from this study emphasize that choosing the right combination of model, ranking method, and feature selection strategy is crucial for optimizing both accuracy and efficiency in intrusion detection for SCADA systems. The robust framework developed here will serve as a valuable tool for real-time anomaly detection, ensuring the continued reliability and security of gas pipeline SCADA systems.

# Scope of Future Researche

<Discuss the scope of future research and identify how this particular research work can be extended in future>

# References

The following is an example of References for this document – replace with the appropriate list of References that have been referred to in this document.

1. Akyildiz, I. F., W. Su, Y. Sankarasubramaniam and E. Cayirci. “A survey on sensor networks”. *IEEE Commun. Mag*., 40 (8), 2002, pp. 102-114.

2. Bartolini, N. T. et al. “Snap and spread: a self-deployment algorithm for sensor networks”. *Lecture Notes on Comput. Sci.,* 5067, 2008, pp. 451-456.

3. Fletcher, G., “Placement and relocation of wireless sensor nodes by a team of robots”, Masters Thesis, University of Ottawa, 2010.

4. Ma, M. and Yang, Y. “Adaptive triangular deployment algorithm for unattended mobile sensor networks”. *IEEE Trans. Comput.,* 56 (7), 2007, pp. 946-958.

# Appendix A: xxx

Link to GitHub: <https://github.com/DanielGoel/FeatureSelectionML.git>

main.py: Loops through all possible combinations of Ranking, Feature selection and model testing using ranking.py, feature\_selection.py and model\_testing.py.

|  |
| --- |
| import os  import pandas as pd  from ranking import DataAnalysisRanker  from feature\_selection import FeatureSelector  from model\_testing import ModelEvaluator  # List of CSV files to process  csv\_files = {      "function": "GaspipelineDatasets/NewGasFilteredFunctionMinMax.csv",      "command": "GaspipelineDatasets/NewGasFilteredCommandMinMax.csv",      "all": "GaspipelineDatasets/NewGasFilteredAllMinMax.csv",      "response": "GaspipelineDatasets/NewGasFilteredResponseNNNoOHEMulti.csv"  }  model\_dir = 'results/models/'  ranking\_methods = ["SP", "WFI-RF", "WFI-XGB"]  # Added WFI-RF and WFI-XGB  use\_std\_dev = use\_abs\_diff = use\_skewness = use\_kurtosis = True  for dataset\_name, input\_file in csv\_files.items():      for ranking\_method in ranking\_methods:          ranking\_file = f'results/Rankings/Ranking\_{ranking\_method}\_{dataset\_name}.csv'            # Step 1: Rank features          ranker = DataAnalysisRanker(ranking\_file, input\_file, ranking\_file)          ranker.analyze(ranking\_method, use\_std\_dev, use\_abs\_diff, use\_skewness, use\_kurtosis)          ranker.save\_results()            # Load ranked features          rankings\_df = pd.read\_csv(ranking\_file)          ranked\_features = rankings\_df.iloc[:, 0].tolist()          total\_features = len(ranked\_features)          print(f"Processing dataset: {dataset\_name} with {total\_features} features using {ranking\_method}.")            for model\_type in ["RF", "XGB"]:              for feature\_selection\_method in ["RFE", "SPFS", "None"]:                  # Step 2: Select Features                  selector = FeatureSelector(dataset\_name, ranking\_file, input\_file)                  selector.perform\_feature\_selection(model\_type, feature\_selection\_method)                    model\_filename = f"{dataset\_name}\_{model\_type}\_{feature\_selection\_method}.pkl"                  model\_path = os.path.join(model\_dir, model\_filename)                    for average\_type in ["micro", "macro", "weighted"]:                      evaluator = ModelEvaluator(input\_file, average\_type, selector.selected\_features, model\_type, feature\_selection\_method, model\_path, ranking\_method)                      evaluator.train\_and\_evaluate()                # Step 3: Remove 1 to (total\_features - 1) features iteratively              for num\_remove in range(1, total\_features):                  selected\_features = ranked\_features[:-num\_remove]                  feature\_selection\_method = f"rem\_{num\_remove}"                  model\_filename = f"{model\_type}\_{feature\_selection\_method}.pkl"                  model\_path = os.path.join(model\_dir, model\_filename)                    selector.selected\_features = selected\_features                  selector.train\_and\_save\_model(selected\_features, model\_type, model\_path)                    for average\_type in ["micro", "macro", "weighted"]:                      evaluator = ModelEvaluator(input\_file, average\_type, selected\_features, model\_type, feature\_selection\_method, model\_path, ranking\_method)                      evaluator.train\_and\_evaluate()        print(f"Completed processing for {dataset\_name} dataset.\n") |

ranking.py: Ranks feature importance using weighted feature importance and statistical properties

|  |
| --- |
| import pandas as pd  import numpy as np  from scipy.stats import skew, kurtosis  from xgboost import XGBClassifier  from sklearn.ensemble import RandomForestClassifier  from sklearn.preprocessing import LabelEncoder  class DataAnalysisRanker:      def \_\_init\_\_(self, ranking\_file, input\_file, output\_file):          self.ranking\_file = ranking\_file          self.input\_file = input\_file          self.output\_file = output\_file          self.df = pd.read\_csv(input\_file)          self.results\_df = None      def compute\_standard\_deviation(self):          return self.df.std()      def compute\_abs\_diff\_mean\_median(self):          return np.abs(self.df.mean() - self.df.median())      def compute\_skewness(self):          return self.df.apply(skew)      def compute\_kurtosis(self):          return self.df.apply(kurtosis)      def analyze(self, rank\_method, use\_std\_dev, use\_abs\_diff, use\_skewness, use\_kurtosis):          metrics = {}          ranks = {}          if rank\_method == 'WFI-RF':              feature\_importances\_df = self.wfi\_RandomForest()              feature\_importances\_df = feature\_importances\_df.set\_index('Name')              self.df = self.df[feature\_importances\_df.index]            elif rank\_method == 'WFI-XGB':              feature\_importances\_df = self.wfi\_XGBoost()              feature\_importances\_df = feature\_importances\_df.set\_index('Name')              self.df = self.df[feature\_importances\_df.index]            else:              if 'Label' in self.df.columns:                  self.df = self.df.drop(columns=['Label'])              if use\_std\_dev:                  metrics['Standard Deviation'] = self.compute\_standard\_deviation()                  ranks['Standard Deviation Rank'] = metrics['Standard Deviation'].rank(ascending=True)              if use\_abs\_diff:                  metrics['Absolute Difference'] = self.compute\_abs\_diff\_mean\_median()                  ranks['Absolute Difference Rank'] = metrics['Absolute Difference'].rank(ascending=True)              if use\_skewness:                  metrics['Skewness'] = self.compute\_skewness()                  ranks['Skewness Rank'] = metrics['Skewness'].rank(ascending=False)              if use\_kurtosis:                  metrics['Kurtosis'] = self.compute\_kurtosis()                  ranks['Kurtosis Rank'] = metrics['Kurtosis'].rank(ascending=True)              # Combine all metrics and ranks into a single DataFrame              OutputData = {\*\*metrics, \*\*ranks}              self.results\_df = pd.DataFrame({                  'Name': self.df.columns,                  \*\*OutputData              })              if ranks:                  total\_rank = sum(ranks.values())                  self.results\_df['Total Rank'] = total\_rank                  self.feature\_importances\_df = self.results\_df.sort\_values('Total Rank', ascending=True)          self.feature\_importances\_df = self.feature\_importances\_df.reset\_index()          #print(feature\_importances\_df)          self.feature\_importances\_df = self.feature\_importances\_df.set\_index('Name')          self.save\_results()        def wfi\_RandomForest(self):          target\_column = 'Label'          y = self.df[target\_column]          X = self.df.drop(columns=[target\_column])          print(X.columns)          model = RandomForestClassifier(random\_state=42)          model.fit(X, y)          self.feature\_importances = model.feature\_importances\_          self.feature\_importances\_df = pd.DataFrame({              'Name': X.columns,              'Random Forest Feature Importance': self.feature\_importances          }).sort\_values(by='Random Forest Feature Importance', ascending=False)  # Sorting by importance          return self.feature\_importances\_df        def wfi\_XGBoost(self):          target\_column = 'Label'          y = self.df[target\_column]          X = self.df.drop(columns=[target\_column])            # Convert class labels to a sequential range          label\_encoder = LabelEncoder()          y\_encoded = label\_encoder.fit\_transform(y)  # Converts [0, 3, 4, 5, 6, 7] → [0, 1, 2, 3, 4, 5]          model = XGBClassifier(use\_label\_encoder=False, eval\_metric='logloss', random\_state=42)          model.fit(X, y\_encoded)  # Use encoded labels          self.feature\_importances = model.feature\_importances\_          self.feature\_importances\_df = pd.DataFrame({              'Name': X.columns,              'XGBoost Feature Importance': self.feature\_importances          }).sort\_values(by='XGBoost Feature Importance', ascending=False)          return self.feature\_importances\_df.sort\_values(by='XGBoost Feature Importance', ascending=False)      def save\_results(self):          df\_to\_save = self.feature\_importances\_df          df\_to\_save.to\_csv(self.ranking\_file, index=True)          print(f"Results saved to {self.ranking\_file}") |

feature\_selection.py: Selects important features using RFE and SPFS

|  |
| --- |
| import pandas as pd  import os  import xgboost as xgb  from sklearn.ensemble import RandomForestClassifier  from sklearn.model\_selection import train\_test\_split  import time  from sklearn.metrics import f1\_score  import pickle  class FeatureSelector:      def \_\_init\_\_(self, dataset\_name, ranking\_file, input\_file):          self.ranking\_file = ranking\_file          self.input\_file = input\_file          self.dataset\_name = dataset\_name          self.df = pd.read\_csv(input\_file)          self.selected\_features = []          self.metrics\_dir = "results/metrics/"          self.model\_dir = "results/models/"          os.makedirs(self.metrics\_dir, exist\_ok=True)          os.makedirs(self.model\_dir, exist\_ok=True)          # Load feature ranking order          self.ranked\_features = self.load\_ranked\_features()      def load\_ranked\_features(self):          rankings\_df = pd.read\_csv(self.ranking\_file)          ranked\_features = rankings\_df.iloc[:, 0].tolist()  # First column contains feature names          return [feature for feature in ranked\_features if feature in self.df.columns]  # Keep only valid features      def perform\_feature\_selection(self, model\_type, method):          if method == "RFE":              self.perform\_rfe(model\_type)          elif method == "SPFS":              self.perform\_spfs(model\_type)          else:              print("Skipped Feature Selection. Using all features.")              self.perform\_baseline(model\_type)      def perform\_rfe(self, model\_type):          target\_column = 'Label'            if target\_column not in self.df.columns:              raise KeyError(f"Column '{target\_column}' not found in dataset!")          y = self.df[target\_column]          X = self.df.drop(columns=[target\_column])  # Use ranked feature order          # Select Model Type          if model\_type == 'XGB':              y = y.astype('category').cat.codes              model = xgb.XGBClassifier(seed=42)          else:              model = RandomForestClassifier(random\_state=42)          selected\_features = self.ranked\_features.copy()  # Start with all ranked features          best\_f1\_score = 0          print("Starting Recursive Feature Elimination...")          # Evaluate the model with all features first          X\_train, X\_test, y\_train, y\_test = train\_test\_split(X[selected\_features], y, test\_size=0.2, random\_state=42)          model.fit(X\_train, y\_train)          y\_pred = model.predict(X\_test)          best\_f1\_score = self.get\_f1\_score(y\_test, y\_pred)['F1 Score']          print(f"Initial F1 Score with all features: {best\_f1\_score:.4f}")          for feature in reversed(self.ranked\_features):              if feature in selected\_features:                  temp\_features = selected\_features.copy()                  temp\_features.remove(feature)                  print(f"Testing without feature: {feature}")                  # Train with updated feature set                  X\_train, X\_test, y\_train, y\_test = train\_test\_split(X[temp\_features], y, test\_size=0.2, random\_state=42)                  model.fit(X\_train, y\_train)                  y\_pred = model.predict(X\_test)                  current\_f1\_score = self.get\_f1\_score(y\_test, y\_pred)['F1 Score']                  if current\_f1\_score >= best\_f1\_score:                      best\_f1\_score = current\_f1\_score                      selected\_features = temp\_features                      print(f"Removing {feature} improved or maintained F1 Score to {current\_f1\_score:.4f}")                  else:                      print(f"Keeping {feature} as removing it decreased F1 Score to {current\_f1\_score:.4f}")          # Save final RFE model          self.selected\_features = selected\_features          self.save\_final\_model(model, model\_type, "RFE")          print(f"Final selected features: {self.selected\_features}")          print(f"Final F1 Score: {best\_f1\_score:.4f}")      def perform\_spfs(self, model\_type):          self.selected\_features = []          target\_column = 'Label'            if target\_column not in self.df.columns:              raise KeyError(f"Column '{target\_column}' not found in dataset!")          y = self.df[target\_column]          X = self.df.drop(columns=[target\_column])          metrics\_data = []          print("Starting SPFS...")          self.ranked\_features = [feature for feature in self.ranked\_features if feature != target\_column]          for feature in self.ranked\_features:              self.selected\_features.append(feature)              X\_selected = X[self.selected\_features]              X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_selected, y, test\_size=0.2, random\_state=42)              if model\_type == 'XGB':                  y\_train = y\_train.astype('category').cat.codes                  y\_test = y\_test.astype('category').cat.codes                  model = xgb.XGBClassifier(seed=42)              else:                  model = RandomForestClassifier(random\_state=42)              model.fit(X\_train, y\_train)              y\_pred = model.predict(X\_test)              f1\_score = self.get\_f1\_score(y\_test, y\_pred)['F1 Score']              if f1\_score > max([m['F1 Score'] for m in metrics\_data] or [0]):                  print(f"Feature {feature} kept. New best F1 score = {f1\_score:.4f}\n")              else:                  self.selected\_features.remove(feature)                  print(f"Feature {feature} removed. F1 score did not improve.")          # Save final SPFS model          self.save\_final\_model(model, model\_type, "SPFS")          print(f"Final selected features: {self.selected\_features}")        def perform\_baseline(self, model\_type):          # Use all features except 'Label'          self.selected\_features = self.df.columns.tolist()          self.selected\_features.remove('Label')          # Extract features and target          X = self.df[self.selected\_features]          y = self.df['Label']          # Encode y if using XGBoost          if model\_type.lower() == "xgb":              y = y.astype('category').cat.codes  # Convert labels to numerical codes              model = xgb.XGBClassifier(seed=42)          else:              model = RandomForestClassifier(random\_state=42)          # Split data (80% train, 20% test)          X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)          # Train the model          model.fit(X\_train, y\_train)          # Save the trained model          model\_path = f"results/models/{self.dataset\_name}\_{model\_type}\_None.pkl"          os.makedirs(os.path.dirname(model\_path), exist\_ok=True)            with open(model\_path, 'wb') as model\_file:              pickle.dump(model, model\_file)          print(f"✅ Baseline Model ({model\_type}) saved to {model\_path}")        def get\_f1\_score(self, y\_true, y\_pred):          f1 = f1\_score(y\_true, y\_pred, average='weighted')          return {'F1 Score': f1}      def save\_final\_model(self, model, model\_type, method):          model\_filename = f"{self.dataset\_name}\_{model\_type}\_{method}.pkl"          model\_path = os.path.join(self.model\_dir, model\_filename)          with open(model\_path, 'wb') as f:              pickle.dump(model, f)          print(f"Model saved to {model\_path}")      def train\_and\_save\_model(self, selected\_features, model\_type, model\_path):          print(f"Training model using selected features: {selected\_features}")          target\_column = 'Label'            y = self.df[target\_column]          X = self.df[selected\_features]  # Use selected features!            # Select model type          if model\_type.lower() == "xgb":              y = y.astype('category').cat.codes              model = xgb.XGBClassifier(seed=42)          else:              model = RandomForestClassifier(random\_state=42)            # Split dataset          X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)            # Train model          model.fit(X\_train, y\_train)            # Save model          os.makedirs(os.path.dirname(model\_path), exist\_ok=True)          with open(model\_path, 'wb') as f:              pickle.dump(model, f)            print(f"✅ Model saved to {model\_path}") |

model\_testing.py: Cross-validation to evaluate the model efficiency

|  |
| --- |
| import pandas as pd  import os  import pickle  import time  from sklearn.model\_selection import train\_test\_split, KFold  from sklearn.metrics import f1\_score, precision\_score, recall\_score, accuracy\_score, confusion\_matrix  class ModelEvaluator:      def \_\_init\_\_(self, input\_file, average\_type, selected\_features, model\_type, feature\_selection\_method, model\_path, rank\_method):          self.rank\_method = rank\_method          self.input\_file = input\_file          self.feature\_selection\_method = feature\_selection\_method          self.selected\_features = selected\_features          self.average\_type = average\_type          self.model\_type = model\_type          self.model\_path = model\_path          self.df = pd.read\_csv(input\_file)          self.target\_column = "Label"          self.metrics\_log\_file = "results/metrics/evaluation\_log.csv"          self.removed\_features = list(set(self.df.columns) - set(self.selected\_features)- {self.target\_column})        def load\_model(self):          """Load pre-trained model from a file"""          if not os.path.exists(self.model\_path):              raise FileNotFoundError(f"Model file not found at {self.model\_path}")          with open(self.model\_path, 'rb') as model\_file:              model = pickle.load(model\_file)          return model      def evaluate\_model(self, y\_true, y\_pred):          accuracy = accuracy\_score(y\_true, y\_pred)          precision = precision\_score(y\_true, y\_pred, average=self.average\_type, zero\_division=0)          recall = recall\_score(y\_true, y\_pred, average=self.average\_type, zero\_division=0)          f1 = f1\_score(y\_true, y\_pred, average=self.average\_type)          cm = confusion\_matrix(y\_true, y\_pred)          fpr = cm[0][1] / cm[0].sum() if cm[0].sum() > 0 else 0          fnr = cm[1][0] / cm[1].sum() if cm[1].sum() > 0 else 0          return {              'Accuracy': accuracy,              'Precision': precision,              'Recall': recall,              'F1 Score': f1,              'FPR': fpr,              'FNR': fnr          }      def train\_and\_evaluate(self):          """Perform evaluation using 10-Fold Cross-Validation"""          start\_time = time.time()            if self.feature\_selection\_method == "none":              self.selected\_features = self.df.columns.tolist()              self.selected\_features.remove(self.target\_column)  # Ensure target is removed            X = self.df[self.selected\_features]          y = self.df[self.target\_column]          # Ensure y is encoded correctly for XGBoost          if "xgboost" in self.model\_type.lower():              y = y.astype('category').cat.codes  # Convert labels to numerical format          kf = KFold(n\_splits=10, shuffle=True, random\_state=42)          model = self.load\_model()          metrics\_data = []          print("Performing 10-Fold Cross-Validation...")            for fold, (train\_index, val\_index) in enumerate(kf.split(X, y), start=1):              X\_train, X\_val = X.iloc[train\_index], X.iloc[val\_index]              y\_train, y\_val = y.iloc[train\_index], y.iloc[val\_index]                # Ensure feature names match the ones used during training              if hasattr(model, "feature\_names\_in\_"):                  X\_val = X\_val[model.feature\_names\_in\_]              # Use pre-trained model for prediction              y\_pred = model.predict(X\_val)              metrics = self.evaluate\_model(y\_val, y\_pred)              metrics\_data.append(metrics)              print(f"Fold {fold}: F1 Score = {metrics['F1 Score']:.4f}")          # Compute average metrics          avg\_metrics = pd.DataFrame(metrics\_data).mean(numeric\_only=True)          total\_time\_taken = time.time() - start\_time          # Create a log entry          log\_entry = pd.DataFrame({              'Input File': [os.path.basename(self.input\_file)],              'Rank Method': [self.rank\_method],              'Feature Selection': [self.feature\_selection\_method],              'Model Type': [self.model\_type],              'Average Type': [self.average\_type],              'Number of Features': [len(self.selected\_features)],              'Removed Features': [", ".join(self.removed\_features)],              'Accuracy': [avg\_metrics['Accuracy']],              'Precision': [avg\_metrics['Precision']],              'Recall': [avg\_metrics['Recall']],              'F1 Score': [avg\_metrics['F1 Score']],              'FPR': [avg\_metrics['FPR']],              'FNR': [avg\_metrics['FNR']],              'Time Taken': [total\_time\_taken]          })          # Append results to the log file          os.makedirs(os.path.dirname(self.metrics\_log\_file), exist\_ok=True)          if not os.path.exists(self.metrics\_log\_file):              log\_entry.to\_csv(self.metrics\_log\_file, index=False)          else:              log\_entry.to\_csv(self.metrics\_log\_file, mode='a', header=False, index=False)          print(f"✅ Results appended to {self.metrics\_log\_file}") |

Full evaluation\_log.csv Results:

*Table 13 Full Evaluation Log csv file*

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Rank Method | Feature Selection | Model Type | Average Type | Number of Features | Removed Features | Accuracy | Precision | Recall | F1 Score | FPR | FNR | Time Taken |
| WFI-RF | SPFS | rf | micro | 15 |  | 0.9918 | 0.9918 | 0.9918 | 0.9918 | 0.0008 | 0.094 | 1.5151 |
| WFI-RF | SPFS | xgb | micro | 15 |  | 0.9918 | 0.9918 | 0.9918 | 0.9918 | 0.0008 | 0.094 | 1.5044 |
| WFI-XGB | rem\_2 | rf | micro | 13 | address, pressure\_measurement | 0.9918 | 0.9918 | 0.9918 | 0.9918 | 0.0009 | 0.0932 | 1.4249 |
| WFI-XGB | rem\_1 | rf | micro | 14 | address | 0.9918 | 0.9918 | 0.9918 | 0.9918 | 0.0009 | 0.0919 | 1.4563 |
| SP | SPFS | rf | micro | 15 |  | 0.9918 | 0.9918 | 0.9918 | 0.9918 | 0.0009 | 0.0921 | 1.4238 |
| SP | SPFS | xgb | micro | 15 |  | 0.9918 | 0.9918 | 0.9918 | 0.9918 | 0.0009 | 0.0921 | 1.4958 |
| SP | RFE | rf | micro | 13 | pressure\_measurement, crc\_rate | 0.9918 | 0.9918 | 0.9918 | 0.9918 | 0.0009 | 0.0932 | 1.5962 |
| SP | RFE | xgb | micro | 13 | pressure\_measurement, crc\_rate | 0.9918 | 0.9918 | 0.9918 | 0.9918 | 0.0009 | 0.0932 | 1.3897 |
| WFI-RF | rem\_2 | rf | micro | 13 | address, pressure\_measurement | 0.9918 | 0.9918 | 0.9918 | 0.9918 | 0.0009 | 0.0929 | 1.4085 |
| WFI-RF | rem\_1 | rf | micro | 14 | pressure\_measurement | 0.9918 | 0.9918 | 0.9918 | 0.9918 | 0.0009 | 0.0929 | 1.4456 |
| WFI-XGB | SPFS | rf | micro | 15 |  | 0.9918 | 0.9918 | 0.9918 | 0.9918 | 0.0009 | 0.0924 | 1.4906 |
| WFI-XGB | SPFS | xgb | micro | 15 |  | 0.9918 | 0.9918 | 0.9918 | 0.9918 | 0.0009 | 0.0924 | 1.5239 |
| SP | None | rf | micro | 15 |  | 0.9918 | 0.9918 | 0.9918 | 0.9918 | 0.0009 | 0.0921 | 1.4436 |
| WFI-RF | None | rf | micro | 15 |  | 0.9918 | 0.9918 | 0.9918 | 0.9918 | 0.0009 | 0.0921 | 1.5647 |
| WFI-XGB | None | rf | micro | 15 |  | 0.9918 | 0.9918 | 0.9918 | 0.9918 | 0.0009 | 0.0921 | 1.5377 |
| WFI-XGB | rem\_2 | rf | weighted | 13 | address, pressure\_measurement | 0.9918 | 0.9915 | 0.9918 | 0.9915 | 0.0009 | 0.0932 | 1.3906 |
| WFI-XGB | rem\_1 | rf | weighted | 14 | address | 0.9918 | 0.9915 | 0.9918 | 0.9915 | 0.0009 | 0.0919 | 1.3902 |
| WFI-RF | SPFS | rf | weighted | 15 |  | 0.9918 | 0.9915 | 0.9918 | 0.9915 | 0.0008 | 0.094 | 1.4143 |
| WFI-RF | SPFS | xgb | weighted | 15 |  | 0.9918 | 0.9915 | 0.9918 | 0.9915 | 0.0008 | 0.094 | 1.4239 |
| SP | SPFS | rf | weighted | 15 |  | 0.9918 | 0.9915 | 0.9918 | 0.9915 | 0.0009 | 0.0921 | 1.369 |
| SP | SPFS | xgb | weighted | 15 |  | 0.9918 | 0.9915 | 0.9918 | 0.9915 | 0.0009 | 0.0921 | 1.4305 |
| SP | RFE | rf | weighted | 13 | pressure\_measurement, crc\_rate | 0.9918 | 0.9915 | 0.9918 | 0.9915 | 0.0009 | 0.0932 | 1.5099 |
| SP | RFE | xgb | weighted | 13 | pressure\_measurement, crc\_rate | 0.9918 | 0.9915 | 0.9918 | 0.9915 | 0.0009 | 0.0932 | 1.3492 |
| WFI-RF | rem\_2 | rf | weighted | 13 | address, pressure\_measurement | 0.9918 | 0.9915 | 0.9918 | 0.9915 | 0.0009 | 0.0929 | 1.3571 |
| WFI-RF | rem\_1 | rf | weighted | 14 | pressure\_measurement | 0.9918 | 0.9915 | 0.9918 | 0.9915 | 0.0009 | 0.0929 | 1.3741 |
| SP | None | rf | weighted | 15 |  | 0.9918 | 0.9915 | 0.9918 | 0.9915 | 0.0009 | 0.0921 | 1.3929 |
| WFI-RF | None | rf | weighted | 15 |  | 0.9918 | 0.9915 | 0.9918 | 0.9915 | 0.0009 | 0.0921 | 1.4254 |
| WFI-XGB | None | rf | weighted | 15 |  | 0.9918 | 0.9915 | 0.9918 | 0.9915 | 0.0009 | 0.0921 | 1.4228 |
| WFI-XGB | SPFS | rf | weighted | 15 |  | 0.9918 | 0.9915 | 0.9918 | 0.9915 | 0.0009 | 0.0924 | 1.4495 |
| WFI-XGB | SPFS | xgb | weighted | 15 |  | 0.9918 | 0.9915 | 0.9918 | 0.9915 | 0.0009 | 0.0924 | 1.4468 |
| SP | RFE | rf | macro | 13 | pressure\_measurement, crc\_rate | 0.9918 | 0.9624 | 0.9224 | 0.9405 | 0.0009 | 0.0932 | 1.7345 |
| SP | RFE | xgb | macro | 13 | pressure\_measurement, crc\_rate | 0.9918 | 0.9624 | 0.9224 | 0.9405 | 0.0009 | 0.0932 | 1.453 |
| WFI-XGB | rem\_2 | rf | macro | 13 | address, pressure\_measurement | 0.9918 | 0.9623 | 0.9223 | 0.9404 | 0.0009 | 0.0932 | 1.3971 |
| WFI-RF | rem\_1 | rf | macro | 14 | pressure\_measurement | 0.9918 | 0.9623 | 0.9223 | 0.9404 | 0.0009 | 0.0929 | 1.3969 |
| WFI-RF | rem\_2 | rf | macro | 13 | address, pressure\_measurement | 0.9918 | 0.9622 | 0.9223 | 0.9403 | 0.0009 | 0.0929 | 1.3769 |
| SP | SPFS | rf | macro | 15 |  | 0.9918 | 0.9626 | 0.9218 | 0.9402 | 0.0009 | 0.0921 | 1.3729 |
| SP | SPFS | xgb | macro | 15 |  | 0.9918 | 0.9626 | 0.9218 | 0.9402 | 0.0009 | 0.0921 | 1.4418 |
| WFI-XGB | rem\_1 | rf | macro | 14 | address | 0.9918 | 0.9624 | 0.9218 | 0.9402 | 0.0009 | 0.0919 | 1.4115 |
| WFI-RF | SPFS | rf | macro | 15 |  | 0.9918 | 0.9628 | 0.9215 | 0.9402 | 0.0008 | 0.094 | 1.4406 |
| WFI-RF | SPFS | xgb | macro | 15 |  | 0.9918 | 0.9628 | 0.9215 | 0.9402 | 0.0008 | 0.094 | 1.4309 |
| SP | None | rf | macro | 15 |  | 0.9918 | 0.9624 | 0.9217 | 0.94 | 0.0009 | 0.0921 | 1.3722 |
| WFI-RF | None | rf | macro | 15 |  | 0.9918 | 0.9624 | 0.9217 | 0.94 | 0.0009 | 0.0921 | 1.4367 |
| WFI-XGB | None | rf | macro | 15 |  | 0.9918 | 0.9624 | 0.9217 | 0.94 | 0.0009 | 0.0921 | 1.4381 |
| WFI-XGB | SPFS | rf | macro | 15 |  | 0.9918 | 0.9624 | 0.9216 | 0.94 | 0.0009 | 0.0924 | 1.4582 |
| WFI-XGB | SPFS | xgb | macro | 15 |  | 0.9918 | 0.9624 | 0.9216 | 0.94 | 0.0009 | 0.0924 | 1.4202 |
| WFI-XGB | RFE | rf | micro | 12 | address, function, pressure\_measurement | 0.9917 | 0.9917 | 0.9917 | 0.9917 | 0.0009 | 0.0932 | 1.5157 |
| WFI-XGB | RFE | xgb | micro | 12 | address, function, pressure\_measurement | 0.9917 | 0.9917 | 0.9917 | 0.9917 | 0.0009 | 0.0932 | 1.5112 |
| WFI-XGB | RFE | rf | weighted | 12 | address, function, pressure\_measurement | 0.9917 | 0.9914 | 0.9917 | 0.9914 | 0.0009 | 0.0932 | 1.4392 |
| WFI-XGB | RFE | xgb | weighted | 12 | address, function, pressure\_measurement | 0.9917 | 0.9914 | 0.9917 | 0.9914 | 0.0009 | 0.0932 | 1.4613 |
| WFI-XGB | RFE | rf | macro | 12 | address, function, pressure\_measurement | 0.9917 | 0.9624 | 0.9222 | 0.9404 | 0.0009 | 0.0932 | 1.4294 |
| WFI-XGB | RFE | xgb | macro | 12 | address, function, pressure\_measurement | 0.9917 | 0.9624 | 0.9222 | 0.9404 | 0.0009 | 0.0932 | 1.4196 |
| SP | rem\_1 | rf | micro | 14 | control\_scheme | 0.9912 | 0.9912 | 0.9912 | 0.9912 | 0.0011 | 0.0993 | 1.3912 |
| SP | rem\_1 | rf | weighted | 14 | control\_scheme | 0.9912 | 0.9909 | 0.9912 | 0.9908 | 0.0011 | 0.0993 | 1.3577 |
| SP | rem\_1 | rf | macro | 14 | control\_scheme | 0.9912 | 0.9632 | 0.9133 | 0.935 | 0.0011 | 0.0993 | 1.3621 |
| WFI-RF | rem\_3 | rf | micro | 12 | solenoid, address, pressure\_measurement | 0.9896 | 0.9896 | 0.9896 | 0.9896 | 0.0021 | 0.1208 | 1.4036 |
| WFI-XGB | rem\_3 | rf | micro | 12 | solenoid, address, pressure\_measurement | 0.9896 | 0.9896 | 0.9896 | 0.9896 | 0.0021 | 0.1208 | 1.4032 |
| WFI-RF | rem\_3 | rf | weighted | 12 | solenoid, address, pressure\_measurement | 0.9896 | 0.9892 | 0.9896 | 0.9892 | 0.0021 | 0.1208 | 1.3615 |
| WFI-XGB | rem\_3 | rf | weighted | 12 | solenoid, address, pressure\_measurement | 0.9896 | 0.9891 | 0.9896 | 0.9892 | 0.0021 | 0.1208 | 1.3678 |
| WFI-RF | rem\_3 | rf | macro | 12 | solenoid, address, pressure\_measurement | 0.9896 | 0.9521 | 0.9097 | 0.9284 | 0.0021 | 0.1208 | 1.3478 |
| WFI-XGB | rem\_3 | rf | macro | 12 | solenoid, address, pressure\_measurement | 0.9896 | 0.9521 | 0.9096 | 0.9283 | 0.0021 | 0.1208 | 1.3508 |
| SP | rem\_2 | rf | micro | 13 | solenoid, control\_scheme | 0.989 | 0.989 | 0.989 | 0.989 | 0.0023 | 0.125 | 1.372 |
| WFI-RF | rem\_4 | rf | micro | 11 | solenoid, address, pressure\_measurement, control\_scheme | 0.989 | 0.989 | 0.989 | 0.989 | 0.0023 | 0.1258 | 1.3815 |
| WFI-XGB | rem\_4 | rf | micro | 11 | solenoid, address, pressure\_measurement, control\_scheme | 0.989 | 0.989 | 0.989 | 0.989 | 0.0023 | 0.126 | 1.3526 |
| SP | rem\_3 | rf | micro | 12 | solenoid, control\_scheme, length | 0.989 | 0.989 | 0.989 | 0.989 | 0.0024 | 0.1253 | 1.3749 |
| SP | rem\_2 | rf | weighted | 13 | solenoid, control\_scheme | 0.989 | 0.9885 | 0.989 | 0.9886 | 0.0023 | 0.125 | 1.3445 |
| WFI-RF | rem\_4 | rf | weighted | 11 | solenoid, address, pressure\_measurement, control\_scheme | 0.989 | 0.9885 | 0.989 | 0.9886 | 0.0023 | 0.1258 | 1.3639 |
| WFI-XGB | rem\_4 | rf | weighted | 11 | solenoid, address, pressure\_measurement, control\_scheme | 0.989 | 0.9885 | 0.989 | 0.9885 | 0.0023 | 0.126 | 1.3251 |
| SP | rem\_3 | rf | weighted | 12 | solenoid, control\_scheme, length | 0.989 | 0.9885 | 0.989 | 0.9885 | 0.0024 | 0.1253 | 1.3221 |
| WFI-XGB | rem\_4 | rf | macro | 11 | solenoid, address, pressure\_measurement, control\_scheme | 0.989 | 0.9504 | 0.9022 | 0.9228 | 0.0023 | 0.126 | 1.3273 |
| SP | rem\_2 | rf | macro | 13 | solenoid, control\_scheme | 0.989 | 0.9507 | 0.9021 | 0.9228 | 0.0023 | 0.125 | 1.3456 |
| WFI-RF | rem\_4 | rf | macro | 11 | solenoid, address, pressure\_measurement, control\_scheme | 0.989 | 0.9502 | 0.9023 | 0.9228 | 0.0023 | 0.1258 | 1.3536 |
| SP | rem\_3 | rf | macro | 12 | solenoid, control\_scheme, length | 0.989 | 0.9504 | 0.902 | 0.9226 | 0.0024 | 0.1253 | 1.3352 |
| SP | rem\_4 | rf | micro | 11 | solenoid, control\_scheme, length, pump | 0.9866 | 0.9866 | 0.9866 | 0.9866 | 0.0035 | 0.172 | 1.3729 |
| SP | rem\_4 | rf | weighted | 11 | solenoid, control\_scheme, length, pump | 0.9866 | 0.9859 | 0.9866 | 0.986 | 0.0035 | 0.172 | 1.3361 |
| SP | rem\_4 | rf | macro | 11 | solenoid, control\_scheme, length, pump | 0.9866 | 0.9434 | 0.8897 | 0.9121 | 0.0035 | 0.172 | 1.3109 |
| SP | rem\_5 | rf | micro | 10 | length, crc\_rate, control\_scheme, solenoid, pump | 0.9865 | 0.9865 | 0.9865 | 0.9865 | 0.0035 | 0.1712 | 1.3893 |
| SP | rem\_5 | rf | weighted | 10 | length, crc\_rate, control\_scheme, solenoid, pump | 0.9865 | 0.9859 | 0.9865 | 0.9859 | 0.0035 | 0.1712 | 1.4755 |
| SP | rem\_5 | rf | macro | 10 | length, crc\_rate, control\_scheme, solenoid, pump | 0.9865 | 0.943 | 0.8895 | 0.9119 | 0.0035 | 0.1712 | 1.3869 |
| WFI-RF | rem\_5 | rf | micro | 10 | address, pressure\_measurement, control\_scheme, system\_mode, solenoid | 0.9861 | 0.9861 | 0.9861 | 0.9861 | 0.0029 | 0.2071 | 1.3853 |
| WFI-RF | rem\_5 | rf | weighted | 10 | address, pressure\_measurement, control\_scheme, system\_mode, solenoid | 0.9861 | 0.9854 | 0.9861 | 0.9854 | 0.0029 | 0.2071 | 1.3213 |
| WFI-RF | rem\_5 | rf | macro | 10 | address, pressure\_measurement, control\_scheme, system\_mode, solenoid | 0.9861 | 0.9463 | 0.8852 | 0.911 | 0.0029 | 0.2071 | 1.3254 |
| WFI-XGB | rem\_5 | rf | micro | 10 | address, pressure\_measurement, control\_scheme, solenoid, deadband | 0.981 | 0.981 | 0.981 | 0.981 | 0.0021 | 0.1733 | 1.2935 |
| WFI-XGB | rem\_5 | rf | weighted | 10 | address, pressure\_measurement, control\_scheme, solenoid, deadband | 0.981 | 0.9802 | 0.981 | 0.9799 | 0.0021 | 0.1733 | 1.2895 |
| WFI-XGB | rem\_5 | rf | macro | 10 | address, pressure\_measurement, control\_scheme, solenoid, deadband | 0.981 | 0.9455 | 0.8597 | 0.8933 | 0.0021 | 0.1733 | 1.3348 |
| WFI-RF | rem\_6 | rf | micro | 9 | address, pressure\_measurement, control\_scheme, system\_mode, solenoid, pump | 0.9808 | 0.9808 | 0.9808 | 0.9808 | 0.0052 | 0.3209 | 1.3437 |
| SP | rem\_6 | rf | micro | 9 | length, crc\_rate, control\_scheme, system\_mode, solenoid, pump | 0.9808 | 0.9808 | 0.9808 | 0.9808 | 0.0053 | 0.3176 | 1.4859 |
| WFI-RF | rem\_6 | rf | weighted | 9 | address, pressure\_measurement, control\_scheme, system\_mode, solenoid, pump | 0.9808 | 0.9796 | 0.9808 | 0.9798 | 0.0052 | 0.3209 | 1.279 |
| SP | rem\_6 | rf | weighted | 9 | length, crc\_rate, control\_scheme, system\_mode, solenoid, pump | 0.9808 | 0.9795 | 0.9808 | 0.9797 | 0.0053 | 0.3176 | 1.3226 |
| WFI-RF | rem\_6 | rf | macro | 9 | address, pressure\_measurement, control\_scheme, system\_mode, solenoid, pump | 0.9808 | 0.9274 | 0.8631 | 0.8901 | 0.0052 | 0.3209 | 1.323 |
| SP | rem\_6 | rf | macro | 9 | length, crc\_rate, control\_scheme, system\_mode, solenoid, pump | 0.9808 | 0.9297 | 0.8603 | 0.8888 | 0.0053 | 0.3176 | 1.3407 |
| WFI-RF | rem\_7 | rf | micro | 8 | address, pressure\_measurement, control\_scheme, system\_mode, solenoid, rate, pump | 0.9745 | 0.9745 | 0.9745 | 0.9745 | 0.0052 | 0.317 | 1.3377 |
| WFI-RF | rem\_7 | rf | weighted | 8 | address, pressure\_measurement, control\_scheme, system\_mode, solenoid, rate, pump | 0.9745 | 0.9733 | 0.9745 | 0.9735 | 0.0052 | 0.317 | 1.3343 |
| WFI-RF | rem\_7 | rf | macro | 8 | address, pressure\_measurement, control\_scheme, system\_mode, solenoid, rate, pump | 0.9745 | 0.92 | 0.8531 | 0.8814 | 0.0052 | 0.317 | 1.3632 |
| WFI-XGB | rem\_6 | rf | micro | 9 | address, pressure\_measurement, setpoint, control\_scheme, solenoid, deadband | 0.9717 | 0.9717 | 0.9717 | 0.9717 | 0.0018 | 0.2325 | 1.3613 |
| WFI-XGB | rem\_6 | rf | weighted | 9 | address, pressure\_measurement, setpoint, control\_scheme, solenoid, deadband | 0.9717 | 0.9706 | 0.9717 | 0.9698 | 0.0018 | 0.2325 | 1.2894 |
| WFI-XGB | rem\_6 | rf | macro | 9 | address, pressure\_measurement, setpoint, control\_scheme, solenoid, deadband | 0.9717 | 0.9425 | 0.8221 | 0.8655 | 0.0018 | 0.2325 | 1.2444 |
| SP | rem\_7 | rf | micro | 8 | length, crc\_rate, control\_scheme, gain, system\_mode, solenoid, pump | 0.9699 | 0.9699 | 0.9699 | 0.9699 | 0.0049 | 0.3952 | 1.4467 |
| SP | rem\_7 | rf | weighted | 8 | length, crc\_rate, control\_scheme, gain, system\_mode, solenoid, pump | 0.9699 | 0.968 | 0.9699 | 0.9679 | 0.0049 | 0.3952 | 1.4029 |
| SP | rem\_7 | rf | macro | 8 | length, crc\_rate, control\_scheme, gain, system\_mode, solenoid, pump | 0.9699 | 0.9195 | 0.8127 | 0.8536 | 0.0049 | 0.3952 | 1.3937 |
| WFI-RF | rem\_8 | rf | micro | 7 | address, pressure\_measurement, deadband, control\_scheme, system\_mode, solenoid, rate, pump | 0.9634 | 0.9634 | 0.9634 | 0.9634 | 0.0049 | 0.4219 | 1.2602 |
| WFI-RF | rem\_8 | rf | weighted | 7 | address, pressure\_measurement, deadband, control\_scheme, system\_mode, solenoid, rate, pump | 0.9634 | 0.9611 | 0.9634 | 0.9609 | 0.0049 | 0.4219 | 1.2708 |
| WFI-RF | rem\_8 | rf | macro | 7 | address, pressure\_measurement, deadband, control\_scheme, system\_mode, solenoid, rate, pump | 0.9634 | 0.9106 | 0.7948 | 0.8387 | 0.0049 | 0.4219 | 1.2872 |
| SP | rem\_8 | rf | micro | 7 | length, crc\_rate, deadband, control\_scheme, gain, system\_mode, solenoid, pump | 0.9564 | 0.9564 | 0.9564 | 0.9564 | 0.0038 | 0.5529 | 1.3134 |
| SP | rem\_8 | rf | weighted | 7 | length, crc\_rate, deadband, control\_scheme, gain, system\_mode, solenoid, pump | 0.9564 | 0.9536 | 0.9564 | 0.9515 | 0.0038 | 0.5529 | 1.2621 |
| SP | rem\_8 | rf | macro | 7 | length, crc\_rate, deadband, control\_scheme, gain, system\_mode, solenoid, pump | 0.9564 | 0.9154 | 0.7426 | 0.7972 | 0.0038 | 0.5529 | 1.2567 |
| WFI-RF | rem\_9 | rf | micro | 6 | address, pressure\_measurement, deadband, control\_scheme, gain, system\_mode, solenoid, rate, pump | 0.9489 | 0.9489 | 0.9489 | 0.9489 | 0.0042 | 0.5332 | 1.3321 |
| WFI-RF | rem\_9 | rf | weighted | 6 | address, pressure\_measurement, deadband, control\_scheme, gain, system\_mode, solenoid, rate, pump | 0.9489 | 0.9452 | 0.9489 | 0.9437 | 0.0042 | 0.5332 | 1.2868 |
| WFI-RF | rem\_9 | rf | macro | 6 | address, pressure\_measurement, deadband, control\_scheme, gain, system\_mode, solenoid, rate, pump | 0.9489 | 0.8971 | 0.7345 | 0.7886 | 0.0042 | 0.5332 | 1.2179 |
| WFI-XGB | rem\_7 | rf | micro | 8 | address, reset\_rate, pressure\_measurement, setpoint, control\_scheme, solenoid, deadband | 0.9487 | 0.9487 | 0.9487 | 0.9487 | 0.0018 | 0.4537 | 1.1745 |
| WFI-XGB | rem\_7 | rf | weighted | 8 | address, reset\_rate, pressure\_measurement, setpoint, control\_scheme, solenoid, deadband | 0.9487 | 0.9472 | 0.9487 | 0.9417 | 0.0018 | 0.4537 | 1.1083 |
| WFI-XGB | rem\_7 | rf | macro | 8 | address, reset\_rate, pressure\_measurement, setpoint, control\_scheme, solenoid, deadband | 0.9487 | 0.9219 | 0.7067 | 0.7612 | 0.0018 | 0.4537 | 1.1251 |
| WFI-XGB | rem\_8 | rf | micro | 7 | address, reset\_rate, pressure\_measurement, setpoint, control\_scheme, system\_mode, solenoid, deadband | 0.9378 | 0.9378 | 0.9378 | 0.9378 | 0.0026 | 0.7868 | 1.0696 |
| WFI-XGB | rem\_8 | rf | weighted | 7 | address, reset\_rate, pressure\_measurement, setpoint, control\_scheme, system\_mode, solenoid, deadband | 0.9378 | 0.9325 | 0.9378 | 0.9258 | 0.0026 | 0.7868 | 1.0889 |
| WFI-XGB | rem\_8 | rf | macro | 7 | address, reset\_rate, pressure\_measurement, setpoint, control\_scheme, system\_mode, solenoid, deadband | 0.9378 | 0.8789 | 0.6476 | 0.6981 | 0.0026 | 0.7868 | 1.0627 |
| WFI-RF | rem\_10 | rf | micro | 5 | address, pressure\_measurement, setpoint, deadband, control\_scheme, gain, system\_mode, solenoid, rate, pump | 0.9322 | 0.9322 | 0.9322 | 0.9322 | 0.0031 | 0.7088 | 1.1777 |
| WFI-RF | rem\_10 | rf | weighted | 5 | address, pressure\_measurement, setpoint, deadband, control\_scheme, gain, system\_mode, solenoid, rate, pump | 0.9322 | 0.9257 | 0.9322 | 0.9211 | 0.0031 | 0.7088 | 1.1598 |
| WFI-RF | rem\_10 | rf | macro | 5 | address, pressure\_measurement, setpoint, deadband, control\_scheme, gain, system\_mode, solenoid, rate, pump | 0.9322 | 0.8684 | 0.6623 | 0.719 | 0.0031 | 0.7088 | 1.1897 |
| WFI-XGB | rem\_9 | rf | micro | 6 | address, reset\_rate, pressure\_measurement, setpoint, control\_scheme, system\_mode, solenoid, deadband, cycle\_time | 0.9161 | 0.9161 | 0.9161 | 0.9161 | 0 | 1 | 0.7758 |
| WFI-XGB | rem\_9 | rf | weighted | 6 | address, reset\_rate, pressure\_measurement, setpoint, control\_scheme, system\_mode, solenoid, deadband, cycle\_time | 0.9161 | 0.8882 | 0.9161 | 0.8886 | 0 | 1 | 0.7685 |
| WFI-XGB | rem\_9 | rf | macro | 6 | address, reset\_rate, pressure\_measurement, setpoint, control\_scheme, system\_mode, solenoid, deadband, cycle\_time | 0.9161 | 0.6491 | 0.552 | 0.573 | 0 | 1 | 0.7748 |
| WFI-RF | rem\_11 | rf | micro | 4 | address, reset\_rate, pressure\_measurement, setpoint, deadband, control\_scheme, gain, system\_mode, solenoid, rate, pump | 0.9057 | 0.9057 | 0.9057 | 0.9057 | 0 | 0.9995 | 0.8976 |
| WFI-RF | rem\_11 | rf | weighted | 4 | address, reset\_rate, pressure\_measurement, setpoint, deadband, control\_scheme, gain, system\_mode, solenoid, rate, pump | 0.9057 | 0.8841 | 0.9057 | 0.8711 | 0 | 0.9995 | 0.8409 |
| WFI-RF | rem\_11 | rf | macro | 4 | address, reset\_rate, pressure\_measurement, setpoint, deadband, control\_scheme, gain, system\_mode, solenoid, rate, pump | 0.9057 | 0.6773 | 0.5287 | 0.5428 | 0 | 0.9995 | 0.8851 |
| WFI-XGB | rem\_10 | rf | micro | 5 | address, reset\_rate, pressure\_measurement, setpoint, control\_scheme, gain, system\_mode, solenoid, deadband, cycle\_time | 0.9033 | 0.9033 | 0.9033 | 0.9033 | 0 | 1 | 0.6134 |
| WFI-XGB | rem\_11 | rf | micro | 4 | address, reset\_rate, pressure\_measurement, setpoint, deadband, control\_scheme, gain, system\_mode, solenoid, pump, cycle\_time | 0.9033 | 0.9033 | 0.9033 | 0.9033 | 0 | 1 | 0.532 |
| WFI-XGB | rem\_10 | rf | weighted | 5 | address, reset\_rate, pressure\_measurement, setpoint, control\_scheme, gain, system\_mode, solenoid, deadband, cycle\_time | 0.9033 | 0.8778 | 0.9033 | 0.8664 | 0 | 1 | 0.5665 |
| WFI-XGB | rem\_11 | rf | weighted | 4 | address, reset\_rate, pressure\_measurement, setpoint, deadband, control\_scheme, gain, system\_mode, solenoid, pump, cycle\_time | 0.9033 | 0.8778 | 0.9033 | 0.8664 | 0 | 1 | 0.5155 |
| WFI-XGB | rem\_10 | rf | macro | 5 | address, reset\_rate, pressure\_measurement, setpoint, control\_scheme, gain, system\_mode, solenoid, deadband, cycle\_time | 0.9033 | 0.647 | 0.5223 | 0.5337 | 0 | 1 | 0.5634 |
| WFI-XGB | rem\_11 | rf | macro | 4 | address, reset\_rate, pressure\_measurement, setpoint, deadband, control\_scheme, gain, system\_mode, solenoid, pump, cycle\_time | 0.9033 | 0.647 | 0.5223 | 0.5337 | 0 | 1 | 0.5063 |
| WFI-RF | rem\_12 | rf | micro | 3 | address, pump, reset\_rate, pressure\_measurement, setpoint, control\_scheme, gain, system\_mode, solenoid, rate, deadband, cycle\_time | 0.8916 | 0.8916 | 0.8916 | 0.8916 | 0 | 1 | 0.5335 |
| WFI-RF | rem\_13 | rf | micro | 2 | address, pump, length, reset\_rate, pressure\_measurement, setpoint, control\_scheme, gain, system\_mode, solenoid, rate, deadband, cycle\_time | 0.8916 | 0.8916 | 0.8916 | 0.8916 | 0 | 1 | 0.4988 |
| WFI-XGB | rem\_12 | rf | micro | 3 | address, pump, reset\_rate, pressure\_measurement, setpoint, control\_scheme, gain, system\_mode, solenoid, rate, deadband, cycle\_time | 0.8916 | 0.8916 | 0.8916 | 0.8916 | 0 | 1 | 0.4792 |
| WFI-XGB | rem\_13 | rf | micro | 2 | address, pump, reset\_rate, pressure\_measurement, setpoint, crc\_rate, control\_scheme, gain, system\_mode, solenoid, rate, deadband, cycle\_time | 0.8916 | 0.8916 | 0.8916 | 0.8916 | 0 | 1 | 0.4601 |
| WFI-RF | rem\_12 | rf | weighted | 3 | address, pump, reset\_rate, pressure\_measurement, setpoint, control\_scheme, gain, system\_mode, solenoid, rate, deadband, cycle\_time | 0.8916 | 0.7964 | 0.8916 | 0.8411 | 0 | 1 | 0.4944 |
| WFI-RF | rem\_13 | rf | weighted | 2 | address, pump, length, reset\_rate, pressure\_measurement, setpoint, control\_scheme, gain, system\_mode, solenoid, rate, deadband, cycle\_time | 0.8916 | 0.7964 | 0.8916 | 0.8411 | 0 | 1 | 0.4654 |
| WFI-XGB | rem\_12 | rf | weighted | 3 | address, pump, reset\_rate, pressure\_measurement, setpoint, control\_scheme, gain, system\_mode, solenoid, rate, deadband, cycle\_time | 0.8916 | 0.7964 | 0.8916 | 0.8411 | 0 | 1 | 0.5091 |
| WFI-XGB | rem\_13 | rf | weighted | 2 | address, pump, reset\_rate, pressure\_measurement, setpoint, crc\_rate, control\_scheme, gain, system\_mode, solenoid, rate, deadband, cycle\_time | 0.8916 | 0.7964 | 0.8916 | 0.8411 | 0 | 1 | 0.4839 |
| WFI-RF | rem\_12 | rf | macro | 3 | address, pump, reset\_rate, pressure\_measurement, setpoint, control\_scheme, gain, system\_mode, solenoid, rate, deadband, cycle\_time | 0.8916 | 0.4784 | 0.4952 | 0.4862 | 0 | 1 | 0.4931 |
| WFI-RF | rem\_13 | rf | macro | 2 | address, pump, length, reset\_rate, pressure\_measurement, setpoint, control\_scheme, gain, system\_mode, solenoid, rate, deadband, cycle\_time | 0.8916 | 0.4784 | 0.4952 | 0.4862 | 0 | 1 | 0.45 |
| WFI-XGB | rem\_12 | rf | macro | 3 | address, pump, reset\_rate, pressure\_measurement, setpoint, control\_scheme, gain, system\_mode, solenoid, rate, deadband, cycle\_time | 0.8916 | 0.4784 | 0.4952 | 0.4862 | 0 | 1 | 0.4759 |
| WFI-XGB | rem\_13 | rf | macro | 2 | address, pump, reset\_rate, pressure\_measurement, setpoint, crc\_rate, control\_scheme, gain, system\_mode, solenoid, rate, deadband, cycle\_time | 0.8916 | 0.4784 | 0.4952 | 0.4862 | 0 | 1 | 0.4583 |
| WFI-RF | rem\_14 | rf | micro | 1 | address, pump, length, reset\_rate, pressure\_measurement, setpoint, crc\_rate, control\_scheme, gain, system\_mode, solenoid, rate, deadband, cycle\_time | 0.8869 | 0.8869 | 0.8869 | 0.8869 | 0 | 1 | 0.5122 |
| WFI-RF | rem\_14 | rf | weighted | 1 | address, pump, length, reset\_rate, pressure\_measurement, setpoint, crc\_rate, control\_scheme, gain, system\_mode, solenoid, rate, deadband, cycle\_time | 0.8869 | 0.7924 | 0.8869 | 0.8362 | 0 | 1 | 0.4819 |
| WFI-RF | rem\_14 | rf | macro | 1 | address, pump, length, reset\_rate, pressure\_measurement, setpoint, crc\_rate, control\_scheme, gain, system\_mode, solenoid, rate, deadband, cycle\_time | 0.8869 | 0.4588 | 0.4661 | 0.4592 | 0 | 1 | 0.4739 |
| WFI-RF | RFE | rf | micro | 12 | address, pressure\_measurement, crc\_rate | 0.8826 | 0.8826 | 0.8826 | 0.8826 | 0.0139 | 0.5862 | 1.7775 |
| WFI-RF | RFE | xgb | micro | 12 | address, pressure\_measurement, crc\_rate | 0.8826 | 0.8826 | 0.8826 | 0.8826 | 0.0139 | 0.5862 | 1.7493 |
| WFI-RF | RFE | rf | weighted | 12 | address, pressure\_measurement, crc\_rate | 0.8826 | 0.8786 | 0.8826 | 0.8795 | 0.0139 | 0.5862 | 1.6412 |
| WFI-RF | RFE | xgb | weighted | 12 | address, pressure\_measurement, crc\_rate | 0.8826 | 0.8786 | 0.8826 | 0.8795 | 0.0139 | 0.5862 | 1.6431 |
| WFI-RF | RFE | rf | macro | 12 | address, pressure\_measurement, crc\_rate | 0.8826 | 0.7225 | 0.6273 | 0.6621 | 0.0139 | 0.5862 | 1.6762 |
| WFI-RF | RFE | xgb | macro | 12 | address, pressure\_measurement, crc\_rate | 0.8826 | 0.7225 | 0.6273 | 0.6621 | 0.0139 | 0.5862 | 1.6754 |
| WFI-XGB | rem\_14 | rf | micro | 1 | address, pump, reset\_rate, pressure\_measurement, setpoint, crc\_rate, control\_scheme, function, gain, system\_mode, solenoid, rate, deadband, cycle\_time | 0.8729 | 0.8729 | 0.8729 | 0.8729 | 0 | 1 | 0.3335 |
| WFI-XGB | rem\_14 | rf | weighted | 1 | address, pump, reset\_rate, pressure\_measurement, setpoint, crc\_rate, control\_scheme, function, gain, system\_mode, solenoid, rate, deadband, cycle\_time | 0.8729 | 0.7821 | 0.8729 | 0.8181 | 0 | 1 | 0.3097 |
| WFI-XGB | rem\_14 | rf | macro | 1 | address, pump, reset\_rate, pressure\_measurement, setpoint, crc\_rate, control\_scheme, function, gain, system\_mode, solenoid, rate, deadband, cycle\_time | 0.8729 | 0.4405 | 0.3808 | 0.3741 | 0 | 1 | 0.3187 |
| SP | rem\_9 | rf | micro | 6 | length, crc\_rate, deadband, control\_scheme, function, gain, system\_mode, solenoid, pump | 0.8394 | 0.8394 | 0.8394 | 0.8394 | 0.0024 | 0.9253 | 1.5429 |
| SP | rem\_9 | rf | weighted | 6 | length, crc\_rate, deadband, control\_scheme, function, gain, system\_mode, solenoid, pump | 0.8394 | 0.8127 | 0.8394 | 0.8146 | 0.0024 | 0.9253 | 1.8426 |
| SP | rem\_9 | rf | macro | 6 | length, crc\_rate, deadband, control\_scheme, function, gain, system\_mode, solenoid, pump | 0.8394 | 0.6015 | 0.3472 | 0.3998 | 0.0024 | 0.9253 | 1.7387 |
| SP | rem\_10 | rf | micro | 5 | length, crc\_rate, deadband, control\_scheme, function, gain, system\_mode, solenoid, rate, pump | 0.8393 | 0.8393 | 0.8393 | 0.8393 | 0.0031 | 0.906 | 1.8538 |
| SP | rem\_10 | rf | weighted | 5 | length, crc\_rate, deadband, control\_scheme, function, gain, system\_mode, solenoid, rate, pump | 0.8393 | 0.8103 | 0.8393 | 0.8129 | 0.0031 | 0.906 | 1.505 |
| SP | rem\_10 | rf | macro | 5 | length, crc\_rate, deadband, control\_scheme, function, gain, system\_mode, solenoid, rate, pump | 0.8393 | 0.5991 | 0.3445 | 0.3996 | 0.0031 | 0.906 | 1.5499 |
| SP | rem\_11 | rf | micro | 4 | length, crc\_rate, deadband, control\_scheme, function, gain, system\_mode, solenoid, rate, pump, cycle\_time | 0.8384 | 0.8384 | 0.8384 | 0.8384 | 0.0018 | 0.9432 | 1.4413 |
| SP | rem\_11 | rf | weighted | 4 | length, crc\_rate, deadband, control\_scheme, function, gain, system\_mode, solenoid, rate, pump, cycle\_time | 0.8384 | 0.8036 | 0.8384 | 0.8006 | 0.0018 | 0.9432 | 1.4293 |
| SP | rem\_11 | rf | macro | 4 | length, crc\_rate, deadband, control\_scheme, function, gain, system\_mode, solenoid, rate, pump, cycle\_time | 0.8384 | 0.6089 | 0.2923 | 0.3431 | 0.0018 | 0.9432 | 1.4418 |
| SP | rem\_12 | rf | micro | 3 | pump, length, setpoint, crc\_rate, control\_scheme, function, gain, system\_mode, solenoid, rate, deadband, cycle\_time | 0.8353 | 0.8353 | 0.8353 | 0.8353 | 0.0001 | 0.9985 | 1.0093 |
| SP | rem\_12 | rf | weighted | 3 | pump, length, setpoint, crc\_rate, control\_scheme, function, gain, system\_mode, solenoid, rate, deadband, cycle\_time | 0.8353 | 0.7981 | 0.8353 | 0.773 | 0.0001 | 0.9985 | 1.3909 |
| SP | rem\_12 | rf | macro | 3 | pump, length, setpoint, crc\_rate, control\_scheme, function, gain, system\_mode, solenoid, rate, deadband, cycle\_time | 0.8353 | 0.6375 | 0.2093 | 0.228 | 0.0001 | 0.9985 | 1.0849 |
| SP | rem\_13 | rf | micro | 2 | pump, length, reset\_rate, setpoint, crc\_rate, control\_scheme, function, gain, system\_mode, solenoid, rate, deadband, cycle\_time | 0.8347 | 0.8347 | 0.8347 | 0.8347 | 0 | 1 | 0.4807 |
| SP | rem\_14 | rf | micro | 1 | pump, length, reset\_rate, pressure\_measurement, setpoint, crc\_rate, control\_scheme, function, gain, system\_mode, solenoid, rate, deadband, cycle\_time | 0.8347 | 0.8347 | 0.8347 | 0.8347 | 0 | 1 | 0.3333 |
| SP | rem\_13 | rf | weighted | 2 | pump, length, reset\_rate, setpoint, crc\_rate, control\_scheme, function, gain, system\_mode, solenoid, rate, deadband, cycle\_time | 0.8347 | 0.7277 | 0.8347 | 0.7627 | 0 | 1 | 0.4646 |
| SP | rem\_14 | rf | weighted | 1 | pump, length, reset\_rate, pressure\_measurement, setpoint, crc\_rate, control\_scheme, function, gain, system\_mode, solenoid, rate, deadband, cycle\_time | 0.8347 | 0.72 | 0.8347 | 0.7627 | 0 | 1 | 0.3552 |
| SP | rem\_13 | rf | macro | 2 | pump, length, reset\_rate, setpoint, crc\_rate, control\_scheme, function, gain, system\_mode, solenoid, rate, deadband, cycle\_time | 0.8347 | 0.3304 | 0.1929 | 0.1969 | 0 | 1 | 0.4674 |
| SP | rem\_14 | rf | macro | 1 | pump, length, reset\_rate, pressure\_measurement, setpoint, crc\_rate, control\_scheme, function, gain, system\_mode, solenoid, rate, deadband, cycle\_time | 0.8347 | 0.3057 | 0.1926 | 0.1964 | 0 | 1 | 0.3164 |
| SP | rem\_13 | xgb | micro | 2 | pump, length, reset\_rate, setpoint, crc\_rate, control\_scheme, function, gain, system\_mode, solenoid, rate, deadband, cycle\_time | 0.8305 | 0.8305 | 0.8305 | 0.8305 | 0 | 1 | 1.1697 |
| SP | rem\_14 | xgb | micro | 1 | pump, length, reset\_rate, pressure\_measurement, setpoint, crc\_rate, control\_scheme, function, gain, system\_mode, solenoid, rate, deadband, cycle\_time | 0.8305 | 0.8305 | 0.8305 | 0.8305 | 0 | 1 | 0.8048 |
| WFI-RF | rem\_12 | xgb | micro | 3 | address, pump, reset\_rate, pressure\_measurement, setpoint, control\_scheme, gain, system\_mode, solenoid, rate, deadband, cycle\_time | 0.8305 | 0.8305 | 0.8305 | 0.8305 | 0 | 1 | 1.2921 |
| WFI-RF | rem\_13 | xgb | micro | 2 | address, pump, length, reset\_rate, pressure\_measurement, setpoint, control\_scheme, gain, system\_mode, solenoid, rate, deadband, cycle\_time | 0.8305 | 0.8305 | 0.8305 | 0.8305 | 0 | 1 | 0.7544 |
| WFI-RF | rem\_14 | xgb | micro | 1 | address, pump, length, reset\_rate, pressure\_measurement, setpoint, crc\_rate, control\_scheme, gain, system\_mode, solenoid, rate, deadband, cycle\_time | 0.8305 | 0.8305 | 0.8305 | 0.8305 | 0 | 1 | 0.8048 |
| WFI-XGB | rem\_12 | xgb | micro | 3 | address, pump, reset\_rate, pressure\_measurement, setpoint, control\_scheme, gain, system\_mode, solenoid, rate, deadband, cycle\_time | 0.8305 | 0.8305 | 0.8305 | 0.8305 | 0 | 1 | 0.8888 |
| WFI-XGB | rem\_13 | xgb | micro | 2 | address, pump, reset\_rate, pressure\_measurement, setpoint, crc\_rate, control\_scheme, gain, system\_mode, solenoid, rate, deadband, cycle\_time | 0.8305 | 0.8305 | 0.8305 | 0.8305 | 0 | 1 | 0.8024 |
| WFI-XGB | rem\_14 | xgb | micro | 1 | address, pump, reset\_rate, pressure\_measurement, setpoint, crc\_rate, control\_scheme, function, gain, system\_mode, solenoid, rate, deadband, cycle\_time | 0.8305 | 0.8305 | 0.8305 | 0.8305 | 0 | 1 | 0.871 |
| WFI-XGB | rem\_9 | xgb | micro | 6 | address, reset\_rate, pressure\_measurement, setpoint, control\_scheme, system\_mode, solenoid, deadband, cycle\_time | 0.8305 | 0.8305 | 0.8305 | 0.8305 | 0 | 0 | 1.064 |
| WFI-XGB | rem\_10 | xgb | micro | 5 | address, reset\_rate, pressure\_measurement, setpoint, control\_scheme, gain, system\_mode, solenoid, deadband, cycle\_time | 0.8305 | 0.8305 | 0.8305 | 0.8305 | 0 | 0 | 0.9589 |
| WFI-XGB | rem\_11 | xgb | micro | 4 | address, reset\_rate, pressure\_measurement, setpoint, deadband, control\_scheme, gain, system\_mode, solenoid, pump, cycle\_time | 0.8305 | 0.8305 | 0.8305 | 0.8305 | 0 | 0 | 0.868 |
| WFI-XGB | rem\_9 | xgb | weighted | 6 | address, reset\_rate, pressure\_measurement, setpoint, control\_scheme, system\_mode, solenoid, deadband, cycle\_time | 0.8305 | 0.7547 | 0.8305 | 0.7908 | 0 | 0 | 0.5932 |
| WFI-XGB | rem\_10 | xgb | weighted | 5 | address, reset\_rate, pressure\_measurement, setpoint, control\_scheme, gain, system\_mode, solenoid, deadband, cycle\_time | 0.8305 | 0.7442 | 0.8305 | 0.785 | 0 | 0 | 0.4929 |
| WFI-XGB | rem\_11 | xgb | weighted | 4 | address, reset\_rate, pressure\_measurement, setpoint, deadband, control\_scheme, gain, system\_mode, solenoid, pump, cycle\_time | 0.8305 | 0.7442 | 0.8305 | 0.785 | 0 | 0 | 0.512 |
| WFI-RF | rem\_12 | xgb | weighted | 3 | address, pump, reset\_rate, pressure\_measurement, setpoint, control\_scheme, gain, system\_mode, solenoid, rate, deadband, cycle\_time | 0.8305 | 0.735 | 0.8305 | 0.7798 | 0 | 1 | 0.4346 |
| WFI-RF | rem\_13 | xgb | weighted | 2 | address, pump, length, reset\_rate, pressure\_measurement, setpoint, control\_scheme, gain, system\_mode, solenoid, rate, deadband, cycle\_time | 0.8305 | 0.735 | 0.8305 | 0.7798 | 0 | 1 | 0.4469 |
| WFI-RF | rem\_14 | xgb | weighted | 1 | address, pump, length, reset\_rate, pressure\_measurement, setpoint, crc\_rate, control\_scheme, gain, system\_mode, solenoid, rate, deadband, cycle\_time | 0.8305 | 0.735 | 0.8305 | 0.7798 | 0 | 1 | 0.4225 |
| WFI-XGB | rem\_12 | xgb | weighted | 3 | address, pump, reset\_rate, pressure\_measurement, setpoint, control\_scheme, gain, system\_mode, solenoid, rate, deadband, cycle\_time | 0.8305 | 0.735 | 0.8305 | 0.7798 | 0 | 1 | 0.4424 |
| WFI-XGB | rem\_13 | xgb | weighted | 2 | address, pump, reset\_rate, pressure\_measurement, setpoint, crc\_rate, control\_scheme, gain, system\_mode, solenoid, rate, deadband, cycle\_time | 0.8305 | 0.735 | 0.8305 | 0.7798 | 0 | 1 | 0.4161 |
| WFI-XGB | rem\_14 | xgb | weighted | 1 | address, pump, reset\_rate, pressure\_measurement, setpoint, crc\_rate, control\_scheme, function, gain, system\_mode, solenoid, rate, deadband, cycle\_time | 0.8305 | 0.7283 | 0.8305 | 0.776 | 0 | 1 | 0.3983 |
| SP | rem\_13 | xgb | weighted | 2 | pump, length, reset\_rate, setpoint, crc\_rate, control\_scheme, function, gain, system\_mode, solenoid, rate, deadband, cycle\_time | 0.8305 | 0.6927 | 0.8305 | 0.7553 | 0 | 1 | 0.6827 |
| SP | rem\_14 | xgb | weighted | 1 | pump, length, reset\_rate, pressure\_measurement, setpoint, crc\_rate, control\_scheme, function, gain, system\_mode, solenoid, rate, deadband, cycle\_time | 0.8305 | 0.6927 | 0.8305 | 0.7553 | 0 | 1 | 0.3677 |
| WFI-RF | rem\_12 | xgb | macro | 3 | address, pump, reset\_rate, pressure\_measurement, setpoint, control\_scheme, gain, system\_mode, solenoid, rate, deadband, cycle\_time | 0.8305 | 0.1475 | 0.1667 | 0.1565 | 0 | 1 | 0.4223 |
| WFI-RF | rem\_13 | xgb | macro | 2 | address, pump, length, reset\_rate, pressure\_measurement, setpoint, control\_scheme, gain, system\_mode, solenoid, rate, deadband, cycle\_time | 0.8305 | 0.1475 | 0.1667 | 0.1565 | 0 | 1 | 0.4353 |
| WFI-RF | rem\_14 | xgb | macro | 1 | address, pump, length, reset\_rate, pressure\_measurement, setpoint, crc\_rate, control\_scheme, gain, system\_mode, solenoid, rate, deadband, cycle\_time | 0.8305 | 0.1475 | 0.1667 | 0.1565 | 0 | 1 | 0.4855 |
| WFI-XGB | rem\_12 | xgb | macro | 3 | address, pump, reset\_rate, pressure\_measurement, setpoint, control\_scheme, gain, system\_mode, solenoid, rate, deadband, cycle\_time | 0.8305 | 0.1475 | 0.1667 | 0.1565 | 0 | 1 | 0.4366 |
| WFI-XGB | rem\_13 | xgb | macro | 2 | address, pump, reset\_rate, pressure\_measurement, setpoint, crc\_rate, control\_scheme, gain, system\_mode, solenoid, rate, deadband, cycle\_time | 0.8305 | 0.1475 | 0.1667 | 0.1565 | 0 | 1 | 0.4432 |
| WFI-XGB | rem\_14 | xgb | macro | 1 | address, pump, reset\_rate, pressure\_measurement, setpoint, crc\_rate, control\_scheme, function, gain, system\_mode, solenoid, rate, deadband, cycle\_time | 0.8305 | 0.1462 | 0.1667 | 0.1557 | 0 | 1 | 0.4463 |
| SP | rem\_13 | xgb | macro | 2 | pump, length, reset\_rate, setpoint, crc\_rate, control\_scheme, function, gain, system\_mode, solenoid, rate, deadband, cycle\_time | 0.8305 | 0.139 | 0.1667 | 0.1516 | 0 | 1 | 0.6525 |
| SP | rem\_14 | xgb | macro | 1 | pump, length, reset\_rate, pressure\_measurement, setpoint, crc\_rate, control\_scheme, function, gain, system\_mode, solenoid, rate, deadband, cycle\_time | 0.8305 | 0.139 | 0.1667 | 0.1516 | 0 | 1 | 0.3983 |
| WFI-XGB | rem\_9 | xgb | macro | 6 | address, reset\_rate, pressure\_measurement, setpoint, control\_scheme, system\_mode, solenoid, deadband, cycle\_time | 0.8305 | 0.1298 | 0.1429 | 0.136 | 0 | 0 | 0.5896 |
| WFI-XGB | rem\_10 | xgb | macro | 5 | address, reset\_rate, pressure\_measurement, setpoint, control\_scheme, gain, system\_mode, solenoid, deadband, cycle\_time | 0.8305 | 0.128 | 0.1429 | 0.135 | 0 | 0 | 0.4979 |
| WFI-XGB | rem\_11 | xgb | macro | 4 | address, reset\_rate, pressure\_measurement, setpoint, deadband, control\_scheme, gain, system\_mode, solenoid, pump, cycle\_time | 0.8305 | 0.128 | 0.1429 | 0.135 | 0 | 0 | 0.5092 |
| WFI-RF | rem\_11 | xgb | micro | 4 | address, reset\_rate, pressure\_measurement, setpoint, deadband, control\_scheme, gain, system\_mode, solenoid, rate, pump | 0.8303 | 0.8303 | 0.8303 | 0.8303 | 0.0002 | 0 | 0.9865 |
| WFI-RF | rem\_11 | xgb | weighted | 4 | address, reset\_rate, pressure\_measurement, setpoint, deadband, control\_scheme, gain, system\_mode, solenoid, rate, pump | 0.8303 | 0.7464 | 0.8303 | 0.7861 | 0.0002 | 0 | 0.6024 |
| WFI-RF | rem\_11 | xgb | macro | 4 | address, reset\_rate, pressure\_measurement, setpoint, deadband, control\_scheme, gain, system\_mode, solenoid, rate, pump | 0.8303 | 0.1284 | 0.1428 | 0.1352 | 0.0002 | 0 | 0.6126 |
| SP | rem\_1 | xgb | micro | 14 | control\_scheme | 0.8294 | 0.8294 | 0.8294 | 0.8294 | 0.0005 | 0 | 1.2028 |
| SP | rem\_1 | xgb | weighted | 14 | control\_scheme | 0.8294 | 0.8205 | 0.8294 | 0.825 | 0.0005 | 0 | 0.6677 |
| SP | rem\_1 | xgb | macro | 14 | control\_scheme | 0.8294 | 0.1235 | 0.1248 | 0.1242 | 0.0005 | 0 | 0.6458 |
| SP | None | xgb | micro | 15 |  | 0.8292 | 0.8292 | 0.8292 | 0.8292 | 0.0004 | 0 | 1.2081 |
| WFI-RF | None | xgb | micro | 15 |  | 0.8292 | 0.8292 | 0.8292 | 0.8292 | 0.0004 | 0 | 0.6696 |
| WFI-XGB | None | xgb | micro | 15 |  | 0.8292 | 0.8292 | 0.8292 | 0.8292 | 0.0004 | 0 | 0.6607 |
| WFI-XGB | rem\_8 | xgb | micro | 7 | address, reset\_rate, pressure\_measurement, setpoint, control\_scheme, system\_mode, solenoid, deadband | 0.8292 | 0.8292 | 0.8292 | 0.8292 | 0.0009 | 0 | 10.7052 |
| SP | None | xgb | weighted | 15 |  | 0.8292 | 0.8209 | 0.8292 | 0.825 | 0.0004 | 0 | 0.6618 |
| WFI-RF | None | xgb | weighted | 15 |  | 0.8292 | 0.8209 | 0.8292 | 0.825 | 0.0004 | 0 | 0.641 |
| WFI-XGB | None | xgb | weighted | 15 |  | 0.8292 | 0.8209 | 0.8292 | 0.825 | 0.0004 | 0 | 0.6519 |
| WFI-XGB | rem\_8 | xgb | weighted | 7 | address, reset\_rate, pressure\_measurement, setpoint, control\_scheme, system\_mode, solenoid, deadband | 0.8292 | 0.7821 | 0.8292 | 0.8003 | 0.0009 | 0 | 0.5507 |
| SP | None | xgb | macro | 15 |  | 0.8292 | 0.1237 | 0.1248 | 0.1242 | 0.0004 | 0 | 0.6493 |
| WFI-RF | None | xgb | macro | 15 |  | 0.8292 | 0.1237 | 0.1248 | 0.1242 | 0.0004 | 0 | 0.641 |
| WFI-XGB | None | xgb | macro | 15 |  | 0.8292 | 0.1237 | 0.1248 | 0.1242 | 0.0004 | 0 | 0.6498 |
| WFI-XGB | rem\_8 | xgb | macro | 7 | address, reset\_rate, pressure\_measurement, setpoint, control\_scheme, system\_mode, solenoid, deadband | 0.8292 | 0.1323 | 0.125 | 0.1209 | 0.0009 | 0 | 0.5779 |
| WFI-RF | rem\_1 | xgb | micro | 14 | pressure\_measurement | 0.8291 | 0.8291 | 0.8291 | 0.8291 | 0.0006 | 0 | 1.1478 |
| WFI-RF | rem\_2 | xgb | micro | 13 | address, pressure\_measurement | 0.8291 | 0.8291 | 0.8291 | 0.8291 | 0.0006 | 0 | 1.1923 |
| WFI-XGB | rem\_2 | xgb | micro | 13 | address, pressure\_measurement | 0.8291 | 0.8291 | 0.8291 | 0.8291 | 0.0006 | 0 | 1.2239 |
| WFI-XGB | rem\_1 | xgb | micro | 14 | address | 0.8291 | 0.8291 | 0.8291 | 0.8291 | 0.0004 | 0 | 1.2006 |
| WFI-XGB | rem\_1 | xgb | weighted | 14 | address | 0.8291 | 0.8215 | 0.8291 | 0.8253 | 0.0004 | 0 | 0.6686 |
| WFI-RF | rem\_1 | xgb | weighted | 14 | pressure\_measurement | 0.8291 | 0.8182 | 0.8291 | 0.8236 | 0.0006 | 0 | 0.6597 |
| WFI-RF | rem\_2 | xgb | weighted | 13 | address, pressure\_measurement | 0.8291 | 0.8182 | 0.8291 | 0.8236 | 0.0006 | 0 | 0.6869 |
| WFI-XGB | rem\_2 | xgb | weighted | 13 | address, pressure\_measurement | 0.8291 | 0.8182 | 0.8291 | 0.8236 | 0.0006 | 0 | 0.623 |
| WFI-XGB | rem\_1 | xgb | macro | 14 | address | 0.8291 | 0.1237 | 0.1248 | 0.1242 | 0.0004 | 0 | 0.6542 |
| WFI-RF | rem\_1 | xgb | macro | 14 | pressure\_measurement | 0.8291 | 0.1231 | 0.1248 | 0.124 | 0.0006 | 0 | 0.6422 |
| WFI-RF | rem\_2 | xgb | macro | 13 | address, pressure\_measurement | 0.8291 | 0.1231 | 0.1248 | 0.124 | 0.0006 | 0 | 0.6392 |
| WFI-XGB | rem\_2 | xgb | macro | 13 | address, pressure\_measurement | 0.8291 | 0.1231 | 0.1248 | 0.124 | 0.0006 | 0 | 0.641 |
| WFI-XGB | rem\_6 | xgb | micro | 9 | address, pressure\_measurement, setpoint, control\_scheme, solenoid, deadband | 0.829 | 0.829 | 0.829 | 0.829 | 0.0009 | 0 | 1.0062 |
| WFI-XGB | rem\_6 | xgb | weighted | 9 | address, pressure\_measurement, setpoint, control\_scheme, solenoid, deadband | 0.829 | 0.8023 | 0.829 | 0.8142 | 0.0009 | 0 | 0.6173 |
| WFI-XGB | rem\_6 | xgb | macro | 9 | address, pressure\_measurement, setpoint, control\_scheme, solenoid, deadband | 0.829 | 0.1247 | 0.1249 | 0.1228 | 0.0009 | 0 | 0.5877 |
| SP | rem\_3 | xgb | micro | 12 | solenoid, control\_scheme, length | 0.8289 | 0.8289 | 0.8289 | 0.8289 | 0.0007 | 0 | 1.0679 |
| SP | rem\_3 | xgb | weighted | 12 | solenoid, control\_scheme, length | 0.8289 | 0.8201 | 0.8289 | 0.8245 | 0.0007 | 0 | 0.6312 |
| SP | rem\_3 | xgb | macro | 12 | solenoid, control\_scheme, length | 0.8289 | 0.1234 | 0.1248 | 0.1241 | 0.0007 | 0 | 0.6357 |
| WFI-RF | rem\_3 | xgb | micro | 12 | solenoid, address, pressure\_measurement | 0.8288 | 0.8288 | 0.8288 | 0.8288 | 0.0011 | 0 | 1.0904 |
| WFI-XGB | rem\_3 | xgb | micro | 12 | solenoid, address, pressure\_measurement | 0.8288 | 0.8288 | 0.8288 | 0.8288 | 0.0011 | 0 | 1.1266 |
| SP | rem\_2 | xgb | micro | 13 | solenoid, control\_scheme | 0.8288 | 0.8288 | 0.8288 | 0.8288 | 0.0007 | 0 | 1.0341 |
| SP | rem\_2 | xgb | weighted | 13 | solenoid, control\_scheme | 0.8288 | 0.8194 | 0.8288 | 0.8241 | 0.0007 | 0 | 0.6407 |
| WFI-RF | rem\_3 | xgb | weighted | 12 | solenoid, address, pressure\_measurement | 0.8288 | 0.8161 | 0.8288 | 0.8224 | 0.0011 | 0 | 0.6342 |
| WFI-XGB | rem\_3 | xgb | weighted | 12 | solenoid, address, pressure\_measurement | 0.8288 | 0.8161 | 0.8288 | 0.8224 | 0.0011 | 0 | 0.6225 |
| SP | rem\_2 | xgb | macro | 13 | solenoid, control\_scheme | 0.8288 | 0.1233 | 0.1247 | 0.124 | 0.0007 | 0 | 0.6708 |
| WFI-RF | rem\_3 | xgb | macro | 12 | solenoid, address, pressure\_measurement | 0.8288 | 0.1228 | 0.1247 | 0.1238 | 0.0011 | 0 | 0.6375 |
| WFI-XGB | rem\_3 | xgb | macro | 12 | solenoid, address, pressure\_measurement | 0.8288 | 0.1228 | 0.1247 | 0.1238 | 0.0011 | 0 | 0.633 |
| WFI-XGB | rem\_7 | xgb | micro | 8 | address, reset\_rate, pressure\_measurement, setpoint, control\_scheme, solenoid, deadband | 0.8287 | 0.8287 | 0.8287 | 0.8287 | 0.0014 | 0 | 0.9961 |
| SP | rem\_5 | xgb | micro | 10 | length, crc\_rate, control\_scheme, solenoid, pump | 0.8287 | 0.8287 | 0.8287 | 0.8287 | 0.0012 | 0 | 1.1699 |
| SP | rem\_5 | xgb | weighted | 10 | length, crc\_rate, control\_scheme, solenoid, pump | 0.8287 | 0.8162 | 0.8287 | 0.8224 | 0.0012 | 0 | 0.6266 |
| WFI-XGB | rem\_7 | xgb | weighted | 8 | address, reset\_rate, pressure\_measurement, setpoint, control\_scheme, solenoid, deadband | 0.8287 | 0.7925 | 0.8287 | 0.8053 | 0.0014 | 0 | 0.5934 |
| SP | rem\_5 | xgb | macro | 10 | length, crc\_rate, control\_scheme, solenoid, pump | 0.8287 | 0.1229 | 0.1247 | 0.1238 | 0.0012 | 0 | 0.6362 |
| WFI-XGB | rem\_7 | xgb | macro | 8 | address, reset\_rate, pressure\_measurement, setpoint, control\_scheme, solenoid, deadband | 0.8287 | 0.1347 | 0.125 | 0.1217 | 0.0014 | 0 | 0.5831 |
| WFI-XGB | rem\_5 | xgb | micro | 10 | address, pressure\_measurement, control\_scheme, solenoid, deadband | 0.8286 | 0.8286 | 0.8286 | 0.8286 | 0.0007 | 0 | 1.4411 |
| WFI-RF | rem\_4 | xgb | micro | 11 | solenoid, address, pressure\_measurement, control\_scheme | 0.8286 | 0.8286 | 0.8286 | 0.8286 | 0.001 | 0 | 1.0233 |
| WFI-XGB | rem\_4 | xgb | micro | 11 | solenoid, address, pressure\_measurement, control\_scheme | 0.8286 | 0.8286 | 0.8286 | 0.8286 | 0.001 | 0 | 1.0105 |
| WFI-RF | rem\_4 | xgb | weighted | 11 | solenoid, address, pressure\_measurement, control\_scheme | 0.8286 | 0.8164 | 0.8286 | 0.8224 | 0.001 | 0 | 0.6138 |
| WFI-XGB | rem\_4 | xgb | weighted | 11 | solenoid, address, pressure\_measurement, control\_scheme | 0.8286 | 0.8164 | 0.8286 | 0.8224 | 0.001 | 0 | 0.6273 |
| WFI-XGB | rem\_5 | xgb | weighted | 10 | address, pressure\_measurement, control\_scheme, solenoid, deadband | 0.8286 | 0.8098 | 0.8286 | 0.8189 | 0.0007 | 0 | 0.601 |
| WFI-RF | rem\_4 | xgb | macro | 11 | solenoid, address, pressure\_measurement, control\_scheme | 0.8286 | 0.1229 | 0.1247 | 0.1238 | 0.001 | 0 | 0.6184 |
| WFI-XGB | rem\_4 | xgb | macro | 11 | solenoid, address, pressure\_measurement, control\_scheme | 0.8286 | 0.1229 | 0.1247 | 0.1238 | 0.001 | 0 | 0.6432 |
| WFI-XGB | rem\_5 | xgb | macro | 10 | address, pressure\_measurement, control\_scheme, solenoid, deadband | 0.8286 | 0.1227 | 0.1248 | 0.1233 | 0.0007 | 0 | 0.6388 |
| SP | rem\_4 | xgb | micro | 11 | solenoid, control\_scheme, length, pump | 0.8285 | 0.8285 | 0.8285 | 0.8285 | 0.0014 | 0 | 0.9888 |
| SP | rem\_4 | xgb | weighted | 11 | solenoid, control\_scheme, length, pump | 0.8285 | 0.8172 | 0.8285 | 0.8228 | 0.0014 | 0 | 0.6248 |
| SP | rem\_4 | xgb | macro | 11 | solenoid, control\_scheme, length, pump | 0.8285 | 0.123 | 0.1247 | 0.1238 | 0.0014 | 0 | 0.6623 |
| SP | rem\_7 | xgb | micro | 8 | length, crc\_rate, control\_scheme, gain, system\_mode, solenoid, pump | 0.8284 | 0.8284 | 0.8284 | 0.8284 | 0.0015 | 0 | 0.9746 |
| SP | rem\_7 | xgb | weighted | 8 | length, crc\_rate, control\_scheme, gain, system\_mode, solenoid, pump | 0.8284 | 0.8041 | 0.8284 | 0.8153 | 0.0015 | 0 | 0.5888 |
| SP | rem\_7 | xgb | macro | 8 | length, crc\_rate, control\_scheme, gain, system\_mode, solenoid, pump | 0.8284 | 0.124 | 0.1249 | 0.1231 | 0.0015 | 0 | 0.6207 |
| WFI-RF | rem\_10 | xgb | micro | 5 | address, pressure\_measurement, setpoint, deadband, control\_scheme, gain, system\_mode, solenoid, rate, pump | 0.8283 | 0.8283 | 0.8283 | 0.8283 | 0.0011 | 0 | 0.9331 |
| SP | rem\_6 | xgb | micro | 9 | length, crc\_rate, control\_scheme, system\_mode, solenoid, pump | 0.8283 | 0.8283 | 0.8283 | 0.8283 | 0.0017 | 0 | 0.9652 |
| SP | rem\_6 | xgb | weighted | 9 | length, crc\_rate, control\_scheme, system\_mode, solenoid, pump | 0.8283 | 0.813 | 0.8283 | 0.8206 | 0.0017 | 0 | 0.6375 |
| WFI-RF | rem\_10 | xgb | weighted | 5 | address, pressure\_measurement, setpoint, deadband, control\_scheme, gain, system\_mode, solenoid, rate, pump | 0.8283 | 0.7678 | 0.8283 | 0.795 | 0.0011 | 0 | 0.5568 |
| SP | rem\_6 | xgb | macro | 9 | length, crc\_rate, control\_scheme, system\_mode, solenoid, pump | 0.8283 | 0.1224 | 0.1247 | 0.1235 | 0.0017 | 0 | 0.6198 |
| WFI-RF | rem\_10 | xgb | macro | 5 | address, pressure\_measurement, setpoint, deadband, control\_scheme, gain, system\_mode, solenoid, rate, pump | 0.8283 | 0.1217 | 0.1248 | 0.1199 | 0.0011 | 0 | 0.5413 |
| WFI-RF | rem\_5 | xgb | micro | 10 | address, pressure\_measurement, control\_scheme, system\_mode, solenoid | 0.8282 | 0.8282 | 0.8282 | 0.8282 | 0.0018 | 0 | 0.9904 |
| SP | rem\_8 | xgb | micro | 7 | length, crc\_rate, deadband, control\_scheme, gain, system\_mode, solenoid, pump | 0.8282 | 0.8282 | 0.8282 | 0.8282 | 0.0015 | 0 | 1.0686 |
| WFI-RF | rem\_5 | xgb | weighted | 10 | address, pressure\_measurement, control\_scheme, system\_mode, solenoid | 0.8282 | 0.8131 | 0.8282 | 0.8206 | 0.0018 | 0 | 0.6055 |
| SP | rem\_8 | xgb | weighted | 7 | length, crc\_rate, deadband, control\_scheme, gain, system\_mode, solenoid, pump | 0.8282 | 0.7981 | 0.8282 | 0.8119 | 0.0015 | 0 | 0.6148 |
| WFI-RF | rem\_5 | xgb | macro | 10 | address, pressure\_measurement, control\_scheme, system\_mode, solenoid | 0.8282 | 0.1224 | 0.1247 | 0.1235 | 0.0018 | 0 | 0.6239 |
| SP | rem\_8 | xgb | macro | 7 | length, crc\_rate, deadband, control\_scheme, gain, system\_mode, solenoid, pump | 0.8282 | 0.1237 | 0.1249 | 0.1227 | 0.0015 | 0 | 0.6081 |
| WFI-RF | rem\_6 | xgb | micro | 9 | address, pressure\_measurement, control\_scheme, system\_mode, solenoid, pump | 0.8273 | 0.8273 | 0.8273 | 0.8273 | 0.003 | 0 | 0.9745 |
| WFI-RF | rem\_6 | xgb | weighted | 9 | address, pressure\_measurement, control\_scheme, system\_mode, solenoid, pump | 0.8273 | 0.8083 | 0.8273 | 0.8176 | 0.003 | 0 | 0.6009 |
| WFI-RF | rem\_6 | xgb | macro | 9 | address, pressure\_measurement, control\_scheme, system\_mode, solenoid, pump | 0.8273 | 0.1218 | 0.1245 | 0.1231 | 0.003 | 0 | 0.5888 |
| WFI-RF | rem\_8 | xgb | micro | 7 | address, pressure\_measurement, deadband, control\_scheme, system\_mode, solenoid, rate, pump | 0.8268 | 0.8268 | 0.8268 | 0.8268 | 0.0025 | 0 | 1.2284 |
| WFI-RF | rem\_9 | xgb | micro | 6 | address, pressure\_measurement, deadband, control\_scheme, gain, system\_mode, solenoid, rate, pump | 0.8268 | 0.8268 | 0.8268 | 0.8268 | 0.0021 | 0 | 0.9531 |
| WFI-RF | rem\_8 | xgb | weighted | 7 | address, pressure\_measurement, deadband, control\_scheme, system\_mode, solenoid, rate, pump | 0.8268 | 0.7923 | 0.8268 | 0.8089 | 0.0025 | 0 | 0.5794 |
| WFI-RF | rem\_9 | xgb | weighted | 6 | address, pressure\_measurement, deadband, control\_scheme, gain, system\_mode, solenoid, rate, pump | 0.8268 | 0.7819 | 0.8268 | 0.8026 | 0.0021 | 0 | 0.578 |
| WFI-RF | rem\_8 | xgb | macro | 7 | address, pressure\_measurement, deadband, control\_scheme, system\_mode, solenoid, rate, pump | 0.8268 | 0.1202 | 0.1245 | 0.1218 | 0.0025 | 0 | 0.5753 |
| WFI-RF | rem\_9 | xgb | macro | 6 | address, pressure\_measurement, deadband, control\_scheme, gain, system\_mode, solenoid, rate, pump | 0.8268 | 0.1215 | 0.1246 | 0.1211 | 0.0021 | 0 | 0.5634 |
| WFI-RF | rem\_7 | xgb | micro | 8 | address, pressure\_measurement, control\_scheme, system\_mode, solenoid, rate, pump | 0.8263 | 0.8263 | 0.8263 | 0.8263 | 0.0032 | 0 | 1.0184 |
| WFI-RF | rem\_7 | xgb | weighted | 8 | address, pressure\_measurement, control\_scheme, system\_mode, solenoid, rate, pump | 0.8263 | 0.8013 | 0.8263 | 0.8136 | 0.0032 | 0 | 0.5901 |
| WFI-RF | rem\_7 | xgb | macro | 8 | address, pressure\_measurement, control\_scheme, system\_mode, solenoid, rate, pump | 0.8263 | 0.1206 | 0.1244 | 0.1225 | 0.0032 | 0 | 0.5771 |
| SP | rem\_12 | xgb | micro | 3 | pump, length, setpoint, crc\_rate, control\_scheme, function, gain, system\_mode, solenoid, rate, deadband, cycle\_time | 0.8204 | 0.8204 | 0.8204 | 0.8204 | 0.0006 | 0 | 1.0851 |
| SP | rem\_12 | xgb | weighted | 3 | pump, length, setpoint, crc\_rate, control\_scheme, function, gain, system\_mode, solenoid, rate, deadband, cycle\_time | 0.8204 | 0.7016 | 0.8204 | 0.7564 | 0.0006 | 0 | 0.5929 |
| SP | rem\_12 | xgb | macro | 3 | pump, length, setpoint, crc\_rate, control\_scheme, function, gain, system\_mode, solenoid, rate, deadband, cycle\_time | 0.8204 | 0.1074 | 0.1253 | 0.1156 | 0.0006 | 0 | 0.6198 |
| SP | rem\_11 | xgb | micro | 4 | length, crc\_rate, deadband, control\_scheme, function, gain, system\_mode, solenoid, rate, pump, cycle\_time | 0.8097 | 0.8097 | 0.8097 | 0.8097 | 0.0012 | 0 | 1.0197 |
| SP | rem\_11 | xgb | weighted | 4 | length, crc\_rate, deadband, control\_scheme, function, gain, system\_mode, solenoid, rate, pump, cycle\_time | 0.8097 | 0.7113 | 0.8097 | 0.7573 | 0.0012 | 0 | 0.5873 |
| SP | rem\_11 | xgb | macro | 4 | length, crc\_rate, deadband, control\_scheme, function, gain, system\_mode, solenoid, rate, pump, cycle\_time | 0.8097 | 0.1072 | 0.122 | 0.1141 | 0.0012 | 0 | 0.614 |
| SP | rem\_10 | xgb | micro | 5 | length, crc\_rate, deadband, control\_scheme, function, gain, system\_mode, solenoid, rate, pump | 0.8018 | 0.8018 | 0.8018 | 0.8018 | 0.001 | 0 | 1.0692 |
| SP | rem\_10 | xgb | weighted | 5 | length, crc\_rate, deadband, control\_scheme, function, gain, system\_mode, solenoid, rate, pump | 0.8018 | 0.7176 | 0.8018 | 0.7573 | 0.001 | 0 | 0.6167 |
| SP | rem\_10 | xgb | macro | 5 | length, crc\_rate, deadband, control\_scheme, function, gain, system\_mode, solenoid, rate, pump | 0.8018 | 0.1083 | 0.121 | 0.1143 | 0.001 | 0 | 0.6272 |
| SP | rem\_9 | xgb | micro | 6 | length, crc\_rate, deadband, control\_scheme, function, gain, system\_mode, solenoid, pump | 0.7967 | 0.7967 | 0.7967 | 0.7967 | 0.001 | 0 | 1.035 |
| SP | rem\_9 | xgb | weighted | 6 | length, crc\_rate, deadband, control\_scheme, function, gain, system\_mode, solenoid, pump | 0.7967 | 0.7211 | 0.7967 | 0.7566 | 0.001 | 0 | 0.6015 |
| SP | rem\_9 | xgb | macro | 6 | length, crc\_rate, deadband, control\_scheme, function, gain, system\_mode, solenoid, pump | 0.7967 | 0.11 | 0.1202 | 0.1142 | 0.001 | 0 | 0.6189 |

**Formatting Guideline for Sanstream Technical Reports**

**(Delete this section before submission)**

Follow guidelines as described in Table 1 consistently throughout the report.

Table 14: Formatting Guideline

|  |  |  |
| --- | --- | --- |
| **MS Word Feature** | **Where to set** | **Expected Value** |
| Style |  | **Header 1-4** for headings as appropriate.  **Normal** for the body |
| Font |  | **Arial 12 pt** for the body of the report  **Arial 11 pt and Italics** for Figure and Table captions  Example:  *Figure 1: Caption of figure 1* |
| Alignment |  | **Justify** for the body of the report  **Left** for text in Table  **Centre** for figure and table captions |
| Spacing |  | **1.5** for the body of the report |
| Captions |  | All figures and tables must have captions.  **Table** captions should be on **top** of the table  **Figure** captions should be at the **bottom** of the figure |

**Content Guidelines**

|  |
| --- |
| Header Types:  See the superscript on the Section Headers  a - Mandatory for all reports  b - Mandatory for survey reports  c - Mandatory for documenting a new methodology/approach proposed by you  d - Mandatory if describing a design of a software system, testbed, framework, etc.  e - Mandatory for reports on experimental work  *1. You are encouraged to use more sections (Heading 1) than listed here and subsections (Heading 2,3,4) as deemed appropriate for your report.*  *2. You can also slightly modify the title of the Section especially for the Design section to be more specific*  *3. Remove the superscripts from the headers and the “Formatting Guideline for Cistel Technical Report” section once the report is completed and ready for review.*  *4. Perform an Update Field on the Table of Contents, List of Figures and List of Tables.* |

Figure 11 Content Guidelines