Automobile Engine Fault Diagnosis Using Machine Learning Method

School of Mechanical & Control Engineering

Handong Global Univ.

Oct.22.2024

Industrial AI & Automation
Project 1
Prof. Young-Keun Kim

21900416 An Gyeonheal

21900727 Jin Garam

CONTENTS

1. Proposal Review

1. Proposal Review

- Objective -

By **Developing a Model** that Outperforms those Implemented in **Existing Journals**,
Accurately **Identify** types of **Engine faults**, Providing Greater Speed and Precision in Engine Repairs.

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 10 October 2023, accepted 2 November 2023, date of publication 8 November 2023, date of current version 14 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3331[illegible]6

RESEARCH ARTICLE

EngineFaultDB: A Novel Dataset for Automotive Engine Fault Classification and Baseline Results

MARY VERGARA[1], LEO RAMOS[2,3], (Student Member, IEEE), NÉSTOR DIEGO RIVERA-CAMPOVERDE[4], (Member, IEEE), AND FRANCKLIN RIVAS-ECHEVERRÍA[3,5], (Senior Member, IEEE)
[1]Higher School of Engineering, Science and Technology, Valencian International University, 46002 Valencia, Spain
[2]Computer Vision Center, Universitat Autònoma de Barcelona, Bellaterra, 08193 Barcelona, Spain
[3]Kauel Inc., Houston, TX 77027, USA
[4]Grupo de Investigación en Ingeniería del Transporte (GIIT), Universidad Politécnica Salesiana, Cuenca 010105, Ecuador
[5]Escuela de Ingeniería, Pontificia Universidad Católica del Ecuador Sede Ibarra, Ibarra 100112, Ecuador

Corresponding authors: Mary Vergara (maryjosefina.vergara@professoruniversidadviu.com) and Leo Ramos (leo.ramos@kauel.com)

This work was supported by the Grupo de Investigación en Ingeniería del Transporte (GIIT) at the Universidad Politécnica Salesiana, Ecuador.

- Goal -

A machine learning model that diagnoses 4 types of defects
using 14 input variables.
Over the Journal F-1 Score (≥ 75.1%)

- Expected Effective -

By building a model capable of early **diagnosis** of **automotive defects**, expect positive effects on **time**, **cost**, and **safety**.

By Applying **Machine Learning**, Classification to Identify **Types of Engine Faults**.

Quickly Diagnose and resolve Engine Faults, Enabling Fast, and Efficient Repairs.

[1] Vergara, M., Ramos, L., Rivera-Campoverde, N. D., & Rivas-Echeverría, F. (2023). Enginefaultdb: a novel dataset for automotive engine fault classification and baseline results. *IEEE Access*, *11*, 126155-126171.

1. Proposal Review

- Data Sets Name-

EngineFaultDB (Supervised Learning)

<Engine Data Collection / Gas Analyzer Device>

- Acquisition -

DAQ, Gas Analyzer

- Input -

MAP, TPS, Force, Power, RPM,

Consumption(L/H),

Consumption(L/100), Speed, CO,

HC, CO2, O2, Lambda, AFR

- Output -

Fault type 0: Normal (16,000 entries)

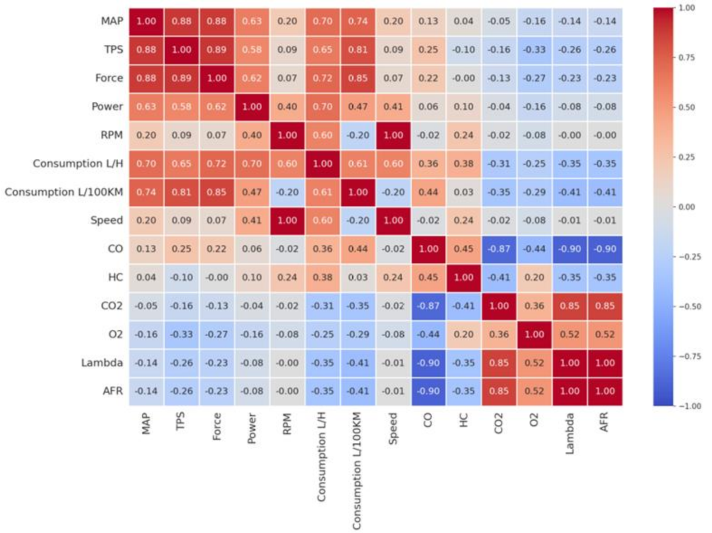
Fault type 1: Rich mixture - High Pressure, Incorrect Sensor, etc. (10,988 entries)

Fault type 2: Lean mixture – Low Pressure, Incorrect Sensor, etc. (15,000 entries)

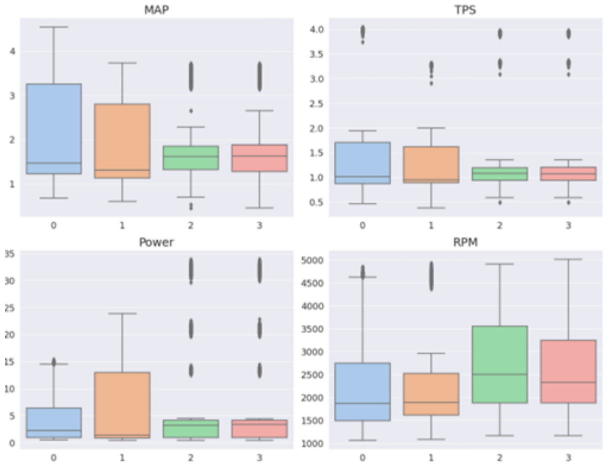
Fault type 3: Low Voltage – Worn Spark, Defective Coil, etc. (14,001 entries)

1. Proposal Review

- How Baseline Journal did-



1. Correlation Analyze



2. Tendency & Outlier

< Journal Performance >

Classifier	Accuracy	Precision	Recall	F1-score
LR	0.576	0.574	0.576	0.574
DT	0.750	0.750	0.750	0.750
RF	0.748	0.748	0.748	0.748
SVC	0.747	**0.768**	0.747	0.715
KNN	**0.751**	0.751	**0.751**	**0.751**
NB	0.394	0.370	0.394	0.353
Neural Net.	0.749	0.748	0.749	0.748

Best Model Performance of Baseline

→ 75.1%

$$X_{\text{scaled}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}}$$

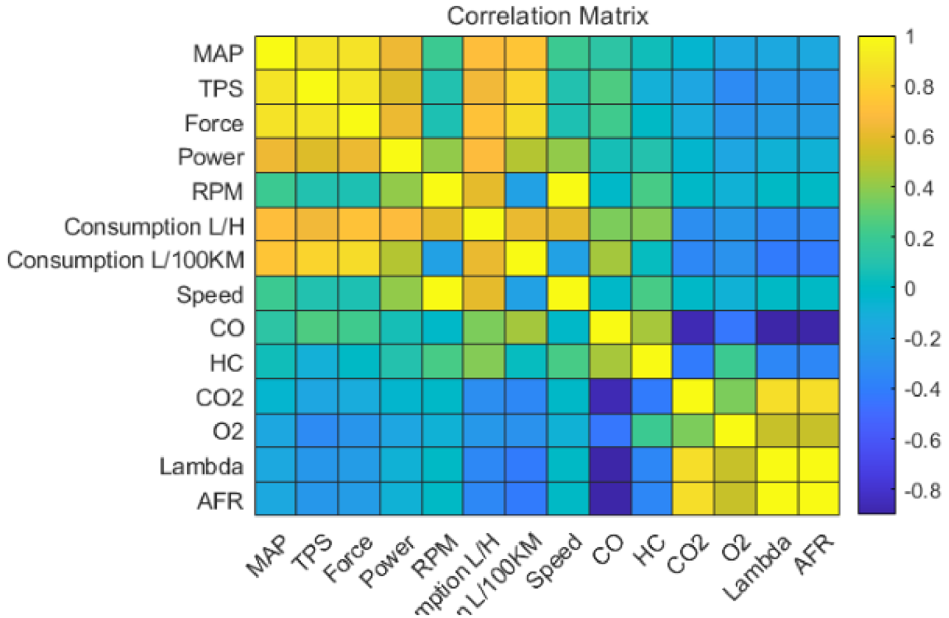
3. Min-Max Scaling

HANDONG GLOBAL UNIVERSITY

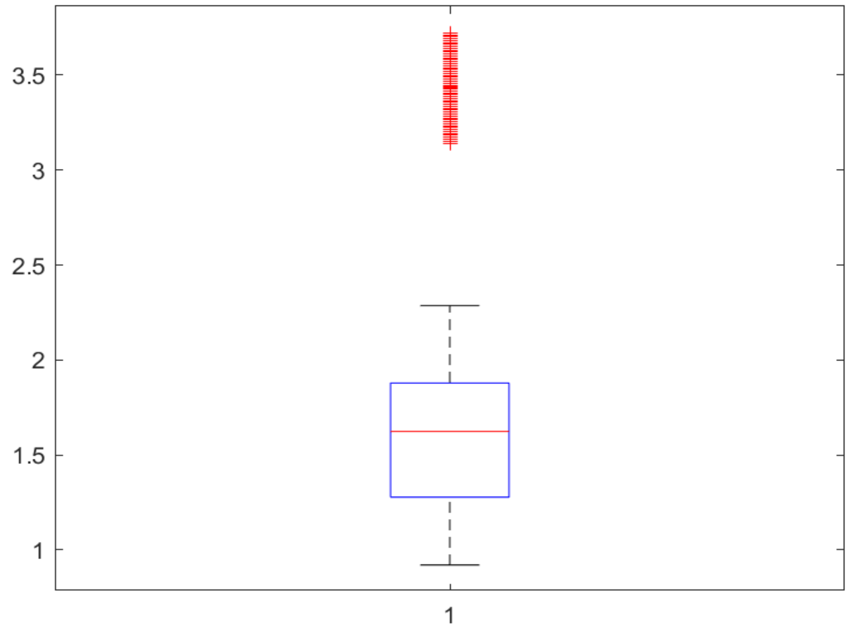
2. Baseline Journal Implementation

2. Baseline Journal Implementation

Following Journal’s Method in MATLAB



< Correlation Analysis >



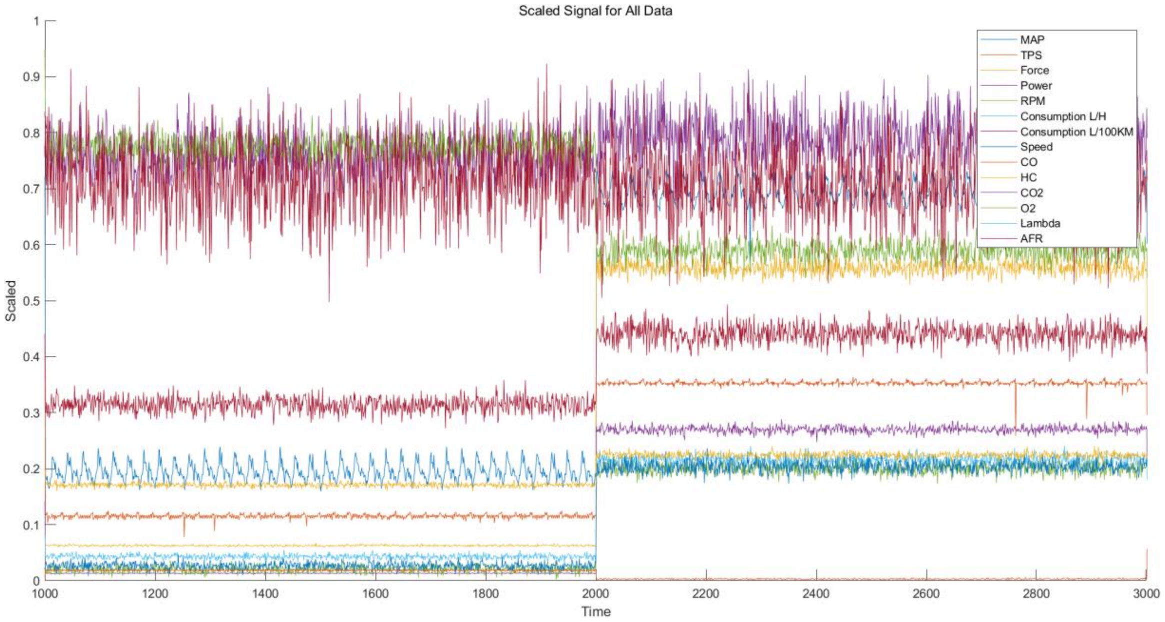
< Box Plot Analysis>

Identified trends using a **correlation table**.

A box plot examines the distribution of **outliers**.

2. Baseline Journal Implementation

Following Journal’s Method in MATLAB



< Min-Max Scaling >

	SVM	TREE	KNN	LR
Accuracy	0.59	0.74	0.75	0.48
Precision	0.58	0.75	0.76	0.48
Recall	0.59	0.75	0.76	0.50
F1 Score	0.57	0.75	0.76	0.48

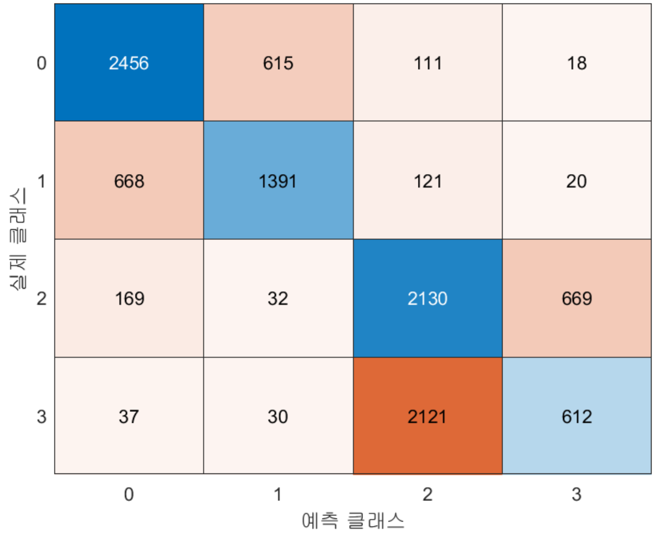
< Performance >

Applied **min-max scaling** to ensure effective training.

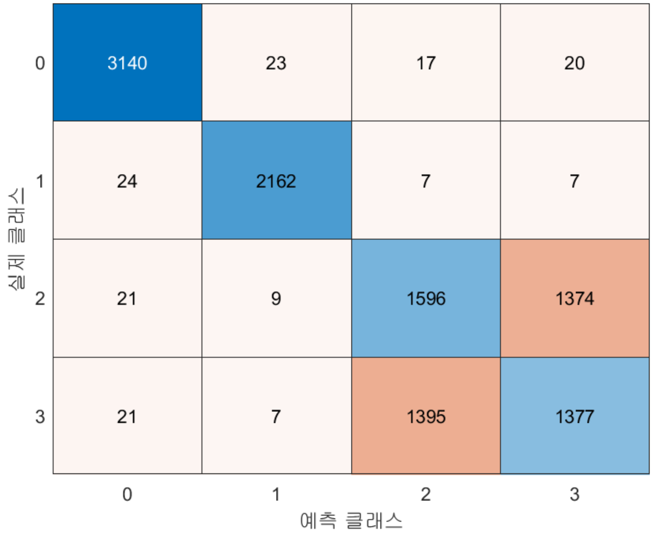
Achieved **similar outcomes** to those reported in the paper.

2. Baseline Journal Implementation

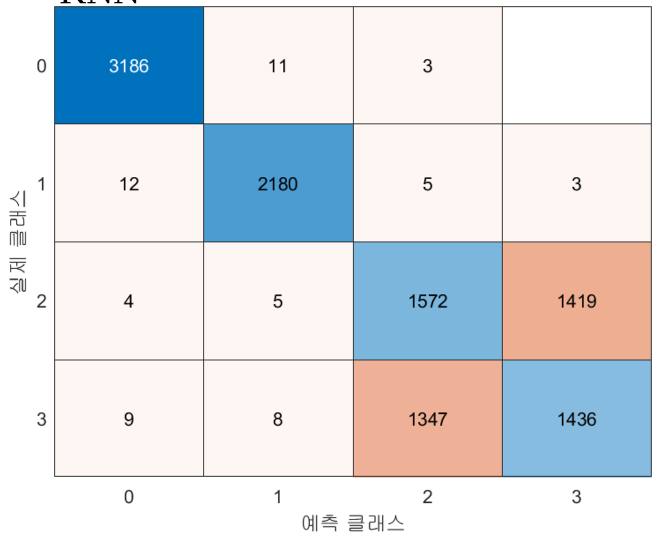
SVM



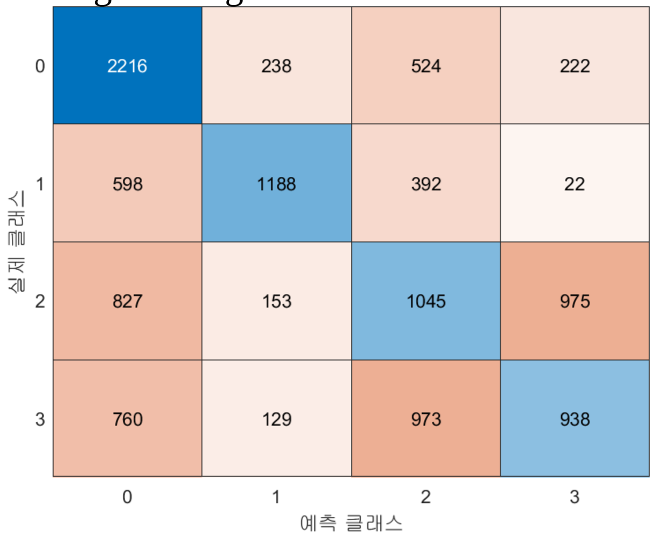
TREE



KNN



Logistic Regression



< Follow Journal Performance >

< Journal Performance >

Classifier	Accuracy	Precision	Recall	F1-score
LR	0.576	0.574	0.576	0.574
DT	0.750	0.750	0.750	0.750
RF	0.748	0.748	0.748	0.748
SVC	0.747	**0.768**	0.747	0.715
KNN	**0.751**	0.751	**0.751**	**0.751**
NB	0.394	0.370	0.394	0.353
Neural Net.	0.749	0.748	0.749	0.748

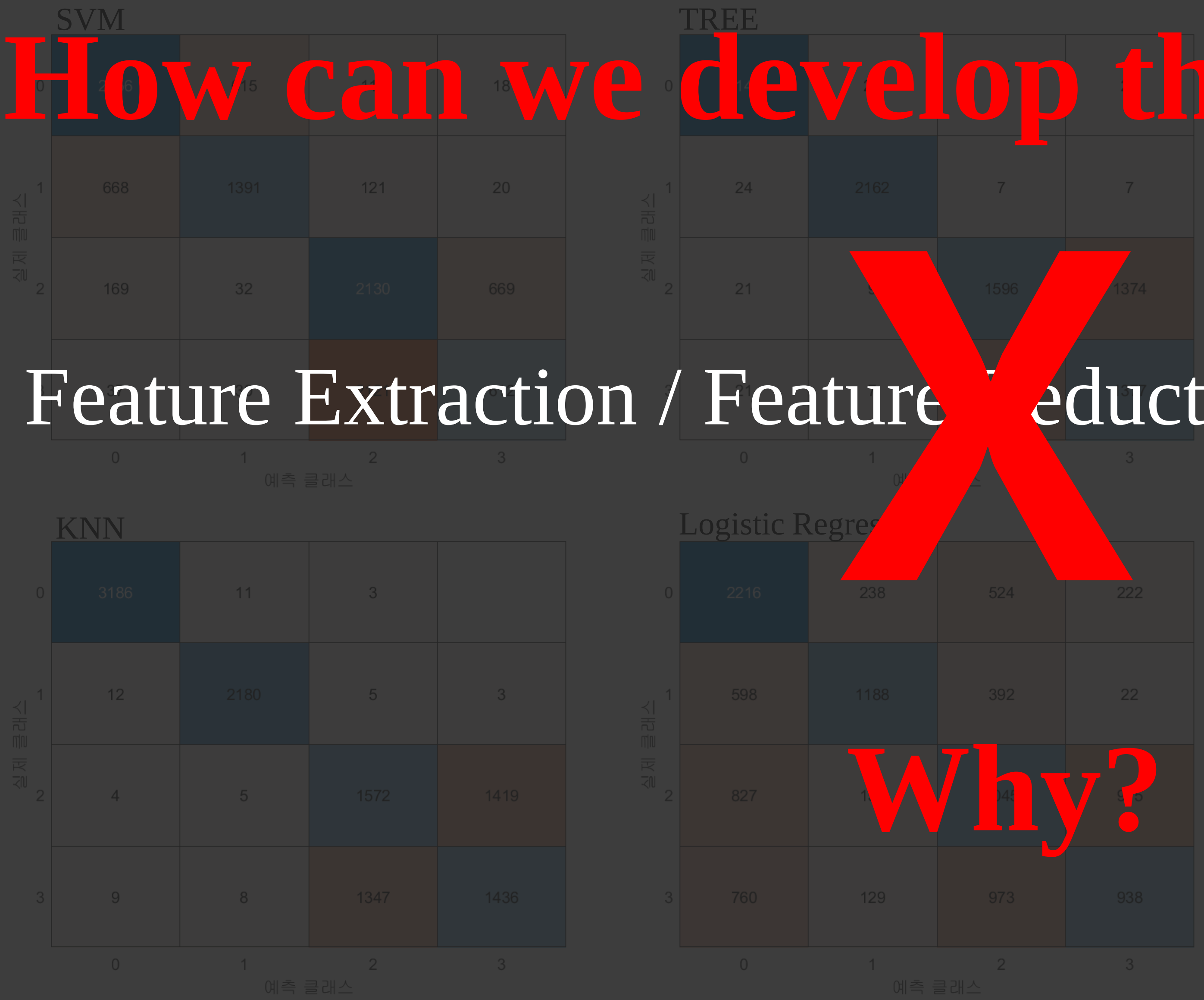
Almost Same!!

< Following Journal Performance >

	SVM	TREE	KNN	LR
Accuracy	0.59	0.74	0.75	0.48
Precision	0.58	0.75	0.76	0.48
Recall	0.59	0.75	0.76	0.50
F1 Score	0.57	0.75	0.76	0.48

2. Baseline Journal Implementation

How can we develop the performance?



Classifier	Accuracy	Precision	Recall	F1-score
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
RF	0.748	0.748	0.748	0.748
SVC	0.747	**0.768**	0.747	0.715
KNN	**0.751**	0.751	**0.751**	**0.751**
NB	0.394	0.370	0.394	0.353
Neural Net.	0.749	0.748	0.749	0.748

< Performance >

Feature Extraction / Feature Reduction / Selecting Features

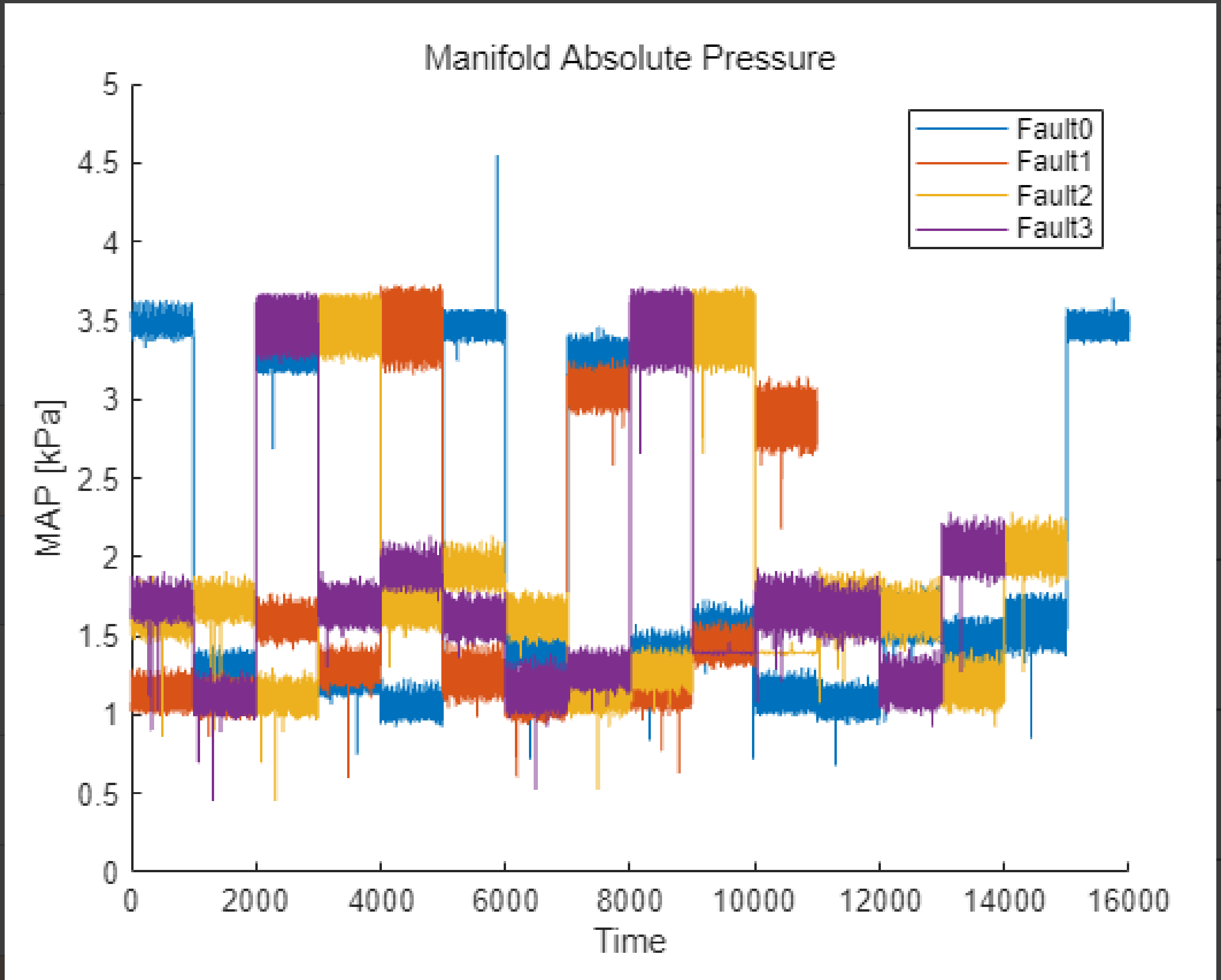
X

	SVM	TREE	KNN	LR
Accuracy	0.59	0.74	0.75	0.48
Precision	0.58	0.75	0.76	0.48
Recall	0.59	0.75	0.76	0.50
F1 Score	0.57	0.75	0.76	0.48

Why?

HANDONG GLOBAL UNIVERSITY

2. Baseline Journa



It’s Important to Focus on Data Distribution

→ Data Preprocessing is Necessary

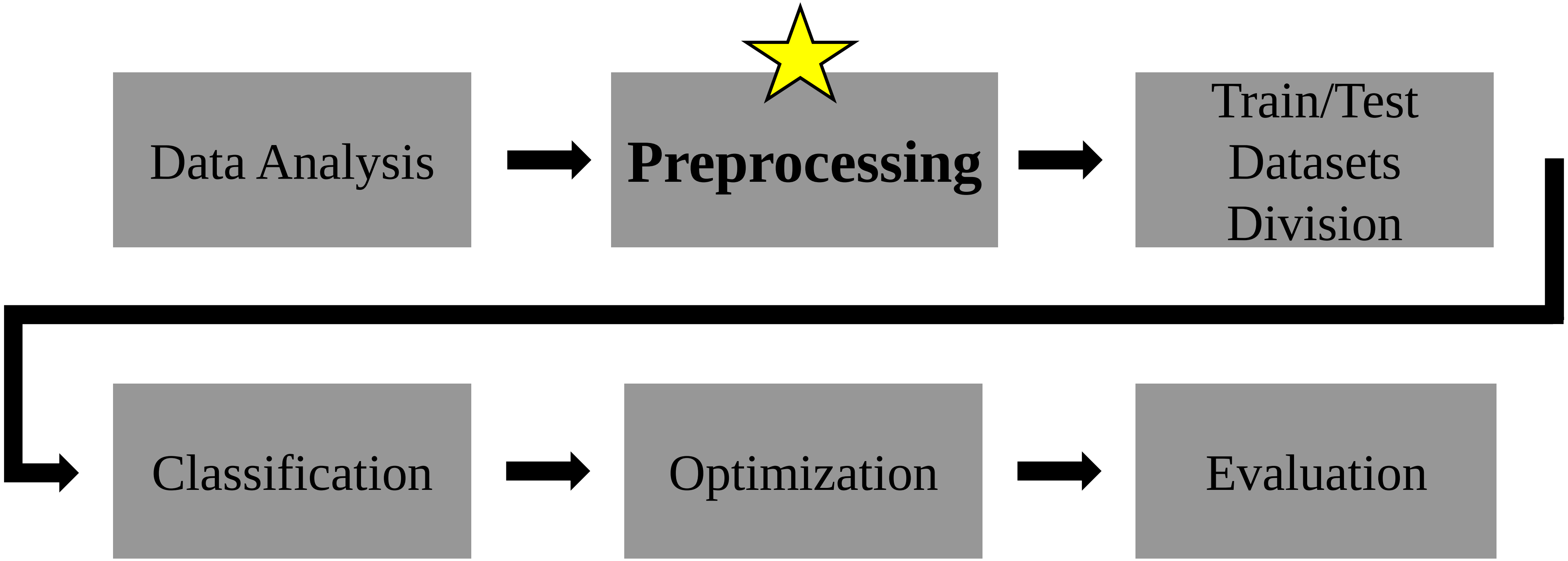
3. Data Preprocessing

3. Data Preprocessing

Proposal Objective	New Objective
Preprocessing	Preprocessing (Scaling, Train & Test Data)
Feature Extraction Feature Reduction Selecting Features	?
Classification (SVM, Tree, KNN etc)	Classification (SVM, Tree, KNN etc)
Evaluation	Evaluation

HANDONG GLOBAL UNIVERSITY

3. Data Preprocessing



3. Data Preprocessing

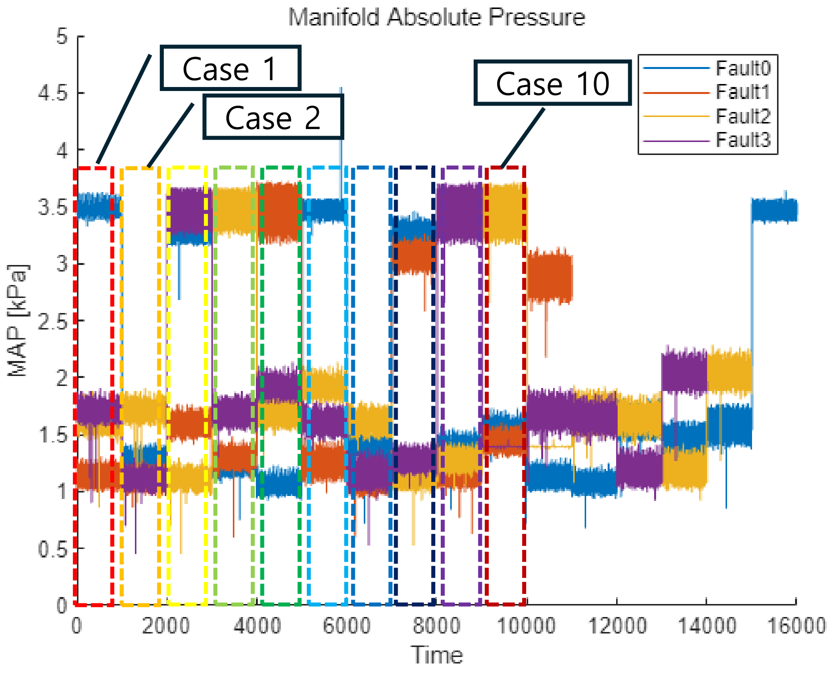
1. Selection

Fault type 0: Normal (16,000 entries) → (10,000 entries)

Fault type 1: Rich mixture - High Pressure, Incorrect Sensor, etc. (10,988 entries) → (10,000 entries)

Fault type 2: Lean mixture – Low Pressure, Incorrect Sensor, etc. (15,000 entries) → (10,000 entries)

Fault type 3: Low Voltage – Worn Spark, Defective Coil, etc. (14,001 entries) → (10,000 entries)



✓ **Equalize the number of normal and defective data**

✓ **10,000 = 1,000 × 10 Cases**

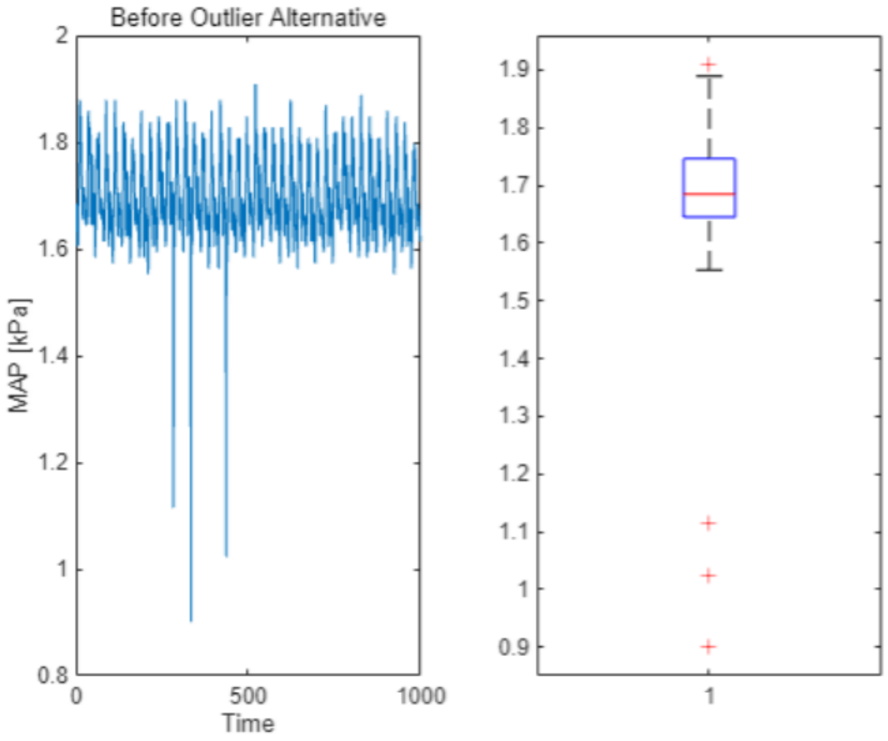
✓ **All the Preprocessing Method Should Apply to each Cases**

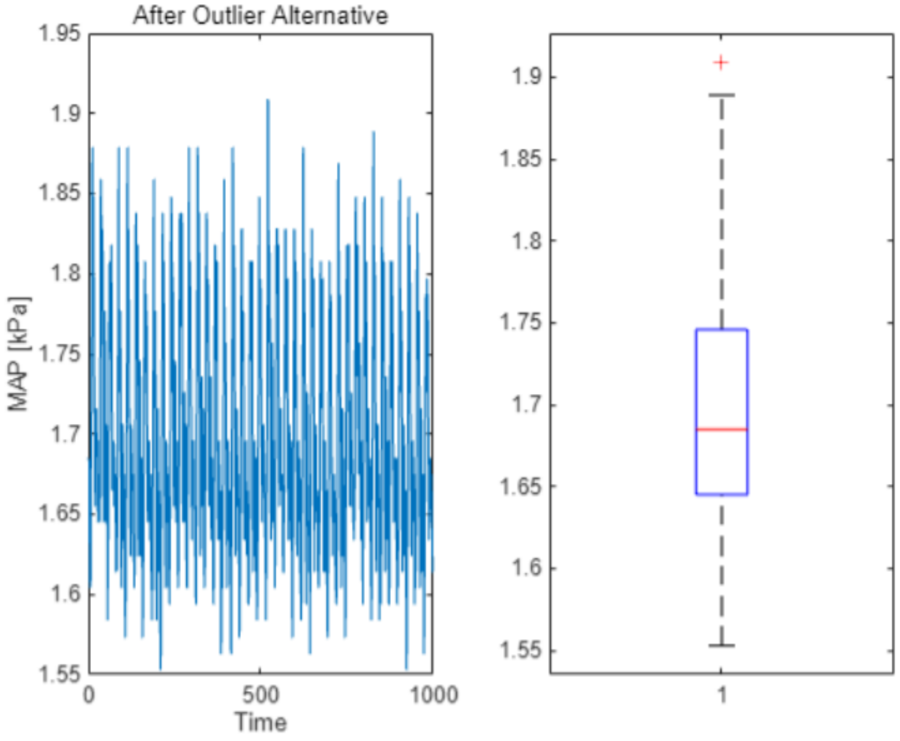
- Outlier Replacement
- Min-Max Scaling
- Filtering

HANDONG UNIVERSITY

3. Data Preprocessing

2. Replacement Outlier

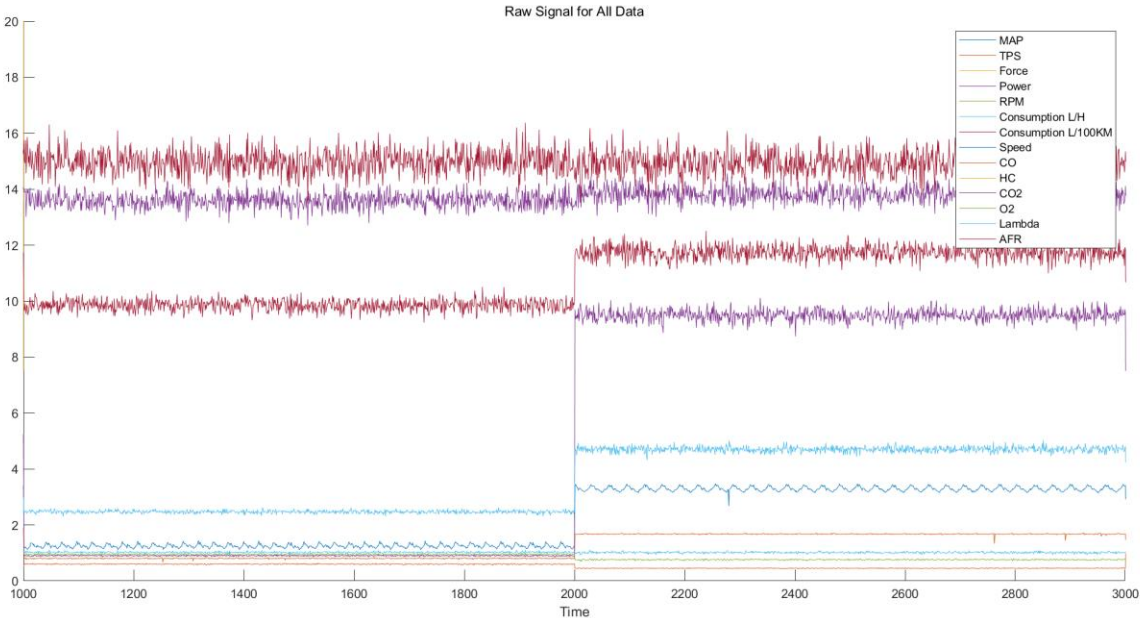


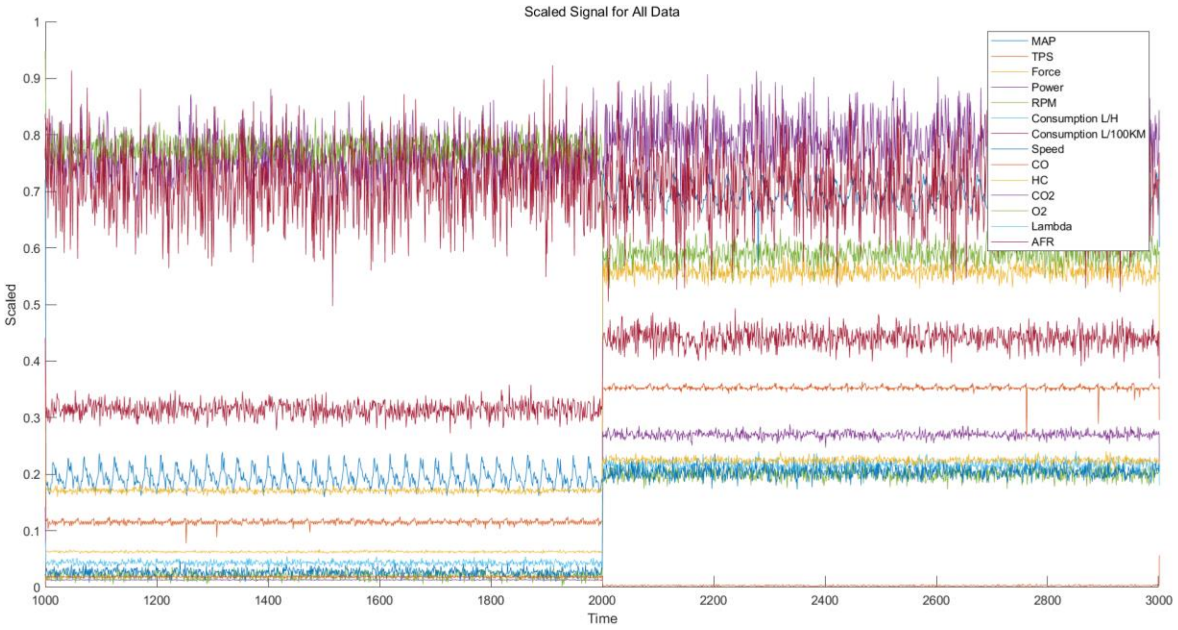


Replace the Outliers: Used **Linear Interpolation**

3. Data Preprocessing

3. Min-Max scaling





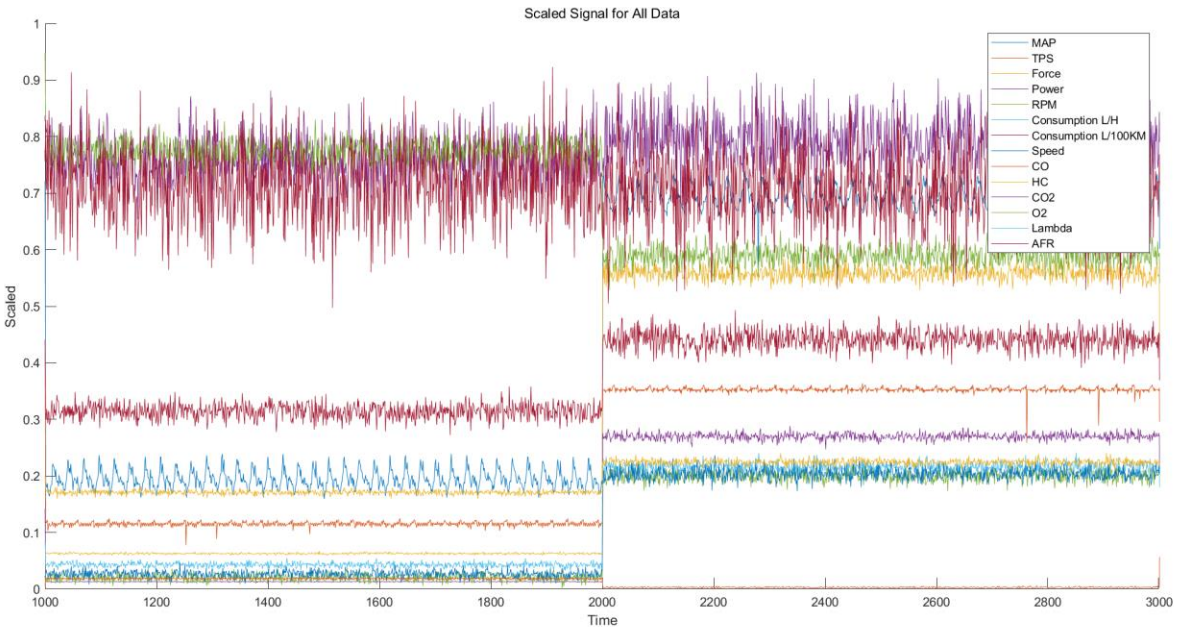
14 Variables have **different magnitude ranges** (ex. %, rpm, Pa etc.).

→ Min-Max Scaling [0, 1].

HANDONG GLOBAL UNIVERSITY

3. Data Preprocessing

4. Butterworth Low Pass Filter



Don’t Focus on Changes Over Time
Focus on the **Distribution** of Variables

Noise → Distributed with a **Large Standard Deviation**

HANDONG GLOBAL UNIVERSITY

4. Classification Model Improvement

3. Data Preprocessing

Test / Train Datasets Split

StratifiedKFold: The Kfold is designed for label datasets with **imbalanced distributions**.

Fault type 0: Normal (10,000 entries)

Fault type 1: Rich mixture - High Pressure, Incorrect Sensor, etc. (10,000 entries)

Fault type 2: Lean mixture – Low Pressure, Incorrect Sensor, etc. (10,000 entries)

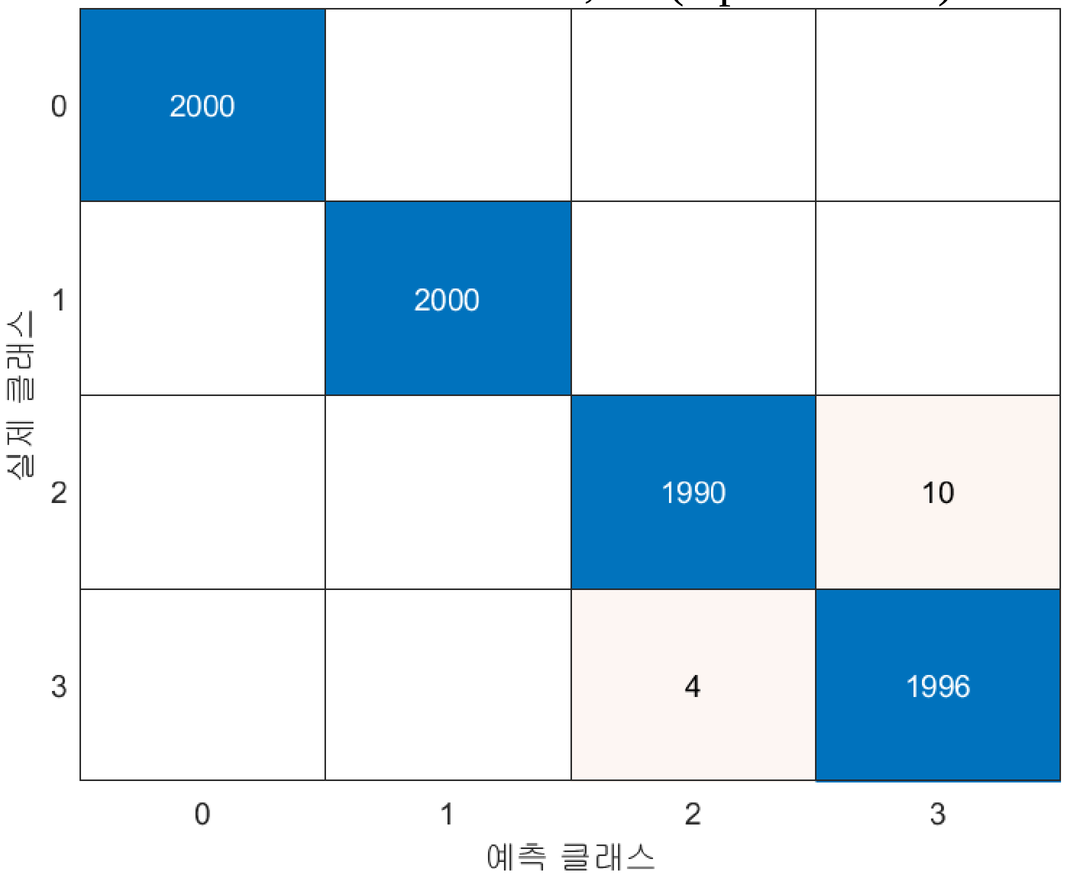
Fault type 3: Low Voltage – Worn Spark, Defective Coil, etc. (10,000 entries)

Choose: (Train: 32,000, Test: 8,000)

The Same Number of Train & Test → **Fairness Train**

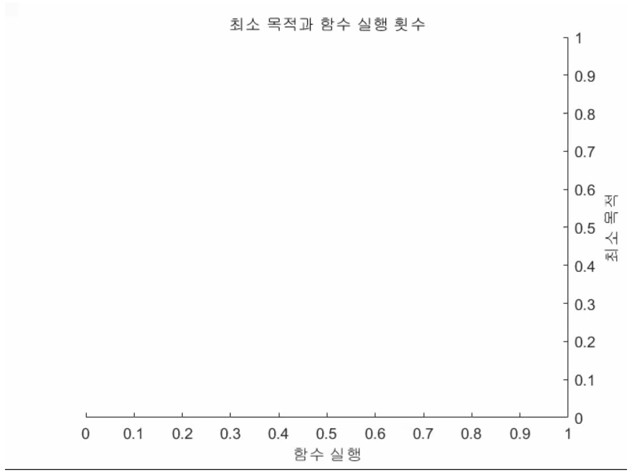
4. Classification Model Improvement

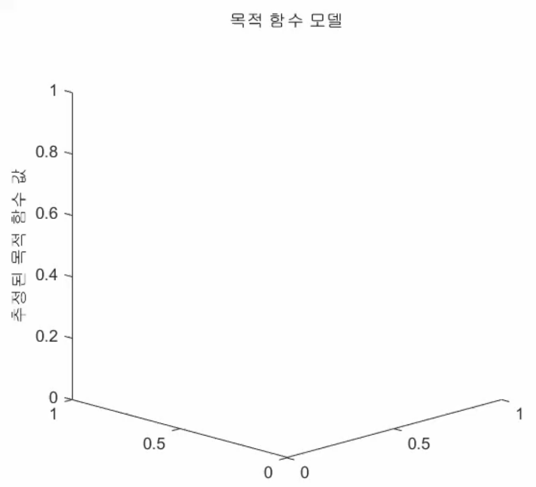
KNN-Number of data: 40,000(Optimization)



Accuracy: 0.99
Recall: 0.99
Precision: 0.99
F1-Score: 0.99

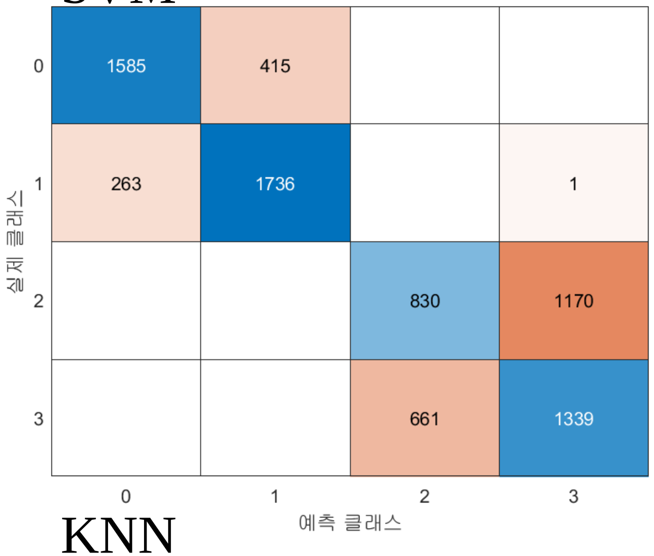
Iter	Eval result	Objective	Objective runtime	BestSoFar (observed)	BestSoFar (estim.)	NumNeighbors	Distance
1	Best	0.16416	3.4096	0.16416	0.16416	103	seuclidean
2	Best	0.064031	2.2297	0.064031	0.068012	16	cityblock
3	Accept	0.75	11.597	0.064031	0.080004	4717	hamming
4	Accept	0.62934	41.8	0.064031	0.087232	11982	cityblock
5	Accept	0.082187	3.5239	0.064031	0.064168	22	cityblock
6	Best	0.004625	3.1968	0.004625	0.0046084	1	cityblock
7	Accept	0.70094	60.35	0.004625	0.004651	15779	seuclidean
8	Accept	0.013875	4.6199	0.004625	0.0046277	1	seuclidean
9	Accept	0.01825	3.9159	0.004625	0.0046241	1	chebychev
10	Accept	0.19809	6.1951	0.004625	0.0046263	338	chebychev
11	Best	0.0036562	3.1542	0.0036562	0.0036813	1	correlation
12	Accept	0.18462	4.8023	0.0036562	0.0036877	240	correlation
13	Accept	0.0037187	2.8983	0.0036562	0.0036849	1	cosine
14	Accept	0.18262	4.9677	0.0036562	0.0036849	224	cosine



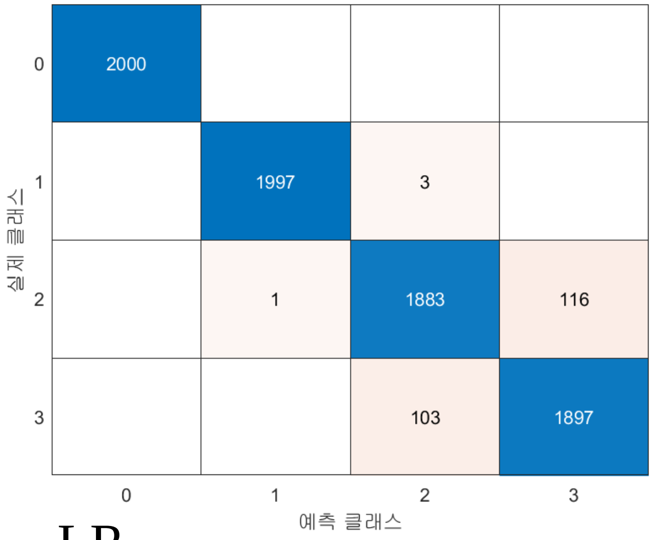


5. Result & Discussion

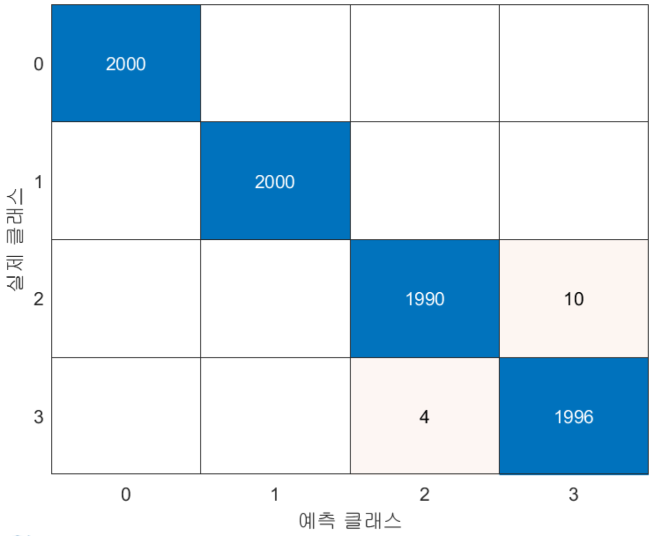
SVM



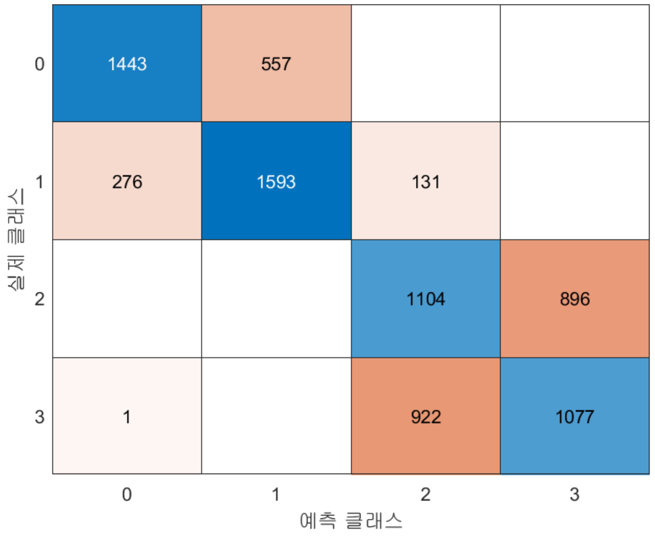
TREE



KNN



LR



< Final Performance Each Model>

	SVM	TREE	KNN	LR
Accuracy	0.69	0.97	0.99	0.65
Precision	0.69	0.97	0.99	0.65
Recall	0.69	0.97	0.99	0.66
F1 Score	0.68	0.97	0.99	0.65

HANDONG GLOBAL UNIVERSITY

5. Result & Discussion

5. Result & Discussion

< Following Journal Performance >

	SVM	TREE	KNN	LR
Accuracy	0.59	0.74	0.75	0.48
Precision	0.58	0.75	0.76	0.48
Recall	0.59	0.75	0.76	0.50
F1 Score	0.57	0.75	0.76	0.48

→

< Final Performance Each Model>

	SVM	TREE	KNN	LR
Accuracy	0.69	0.97	0.99	0.65
Precision	0.69	0.97	0.99	0.65
Recall	0.69	0.97	0.99	0.66
F1 Score	0.68	0.97	0.99	0.65

✓ **76% → 99% (+23%)**

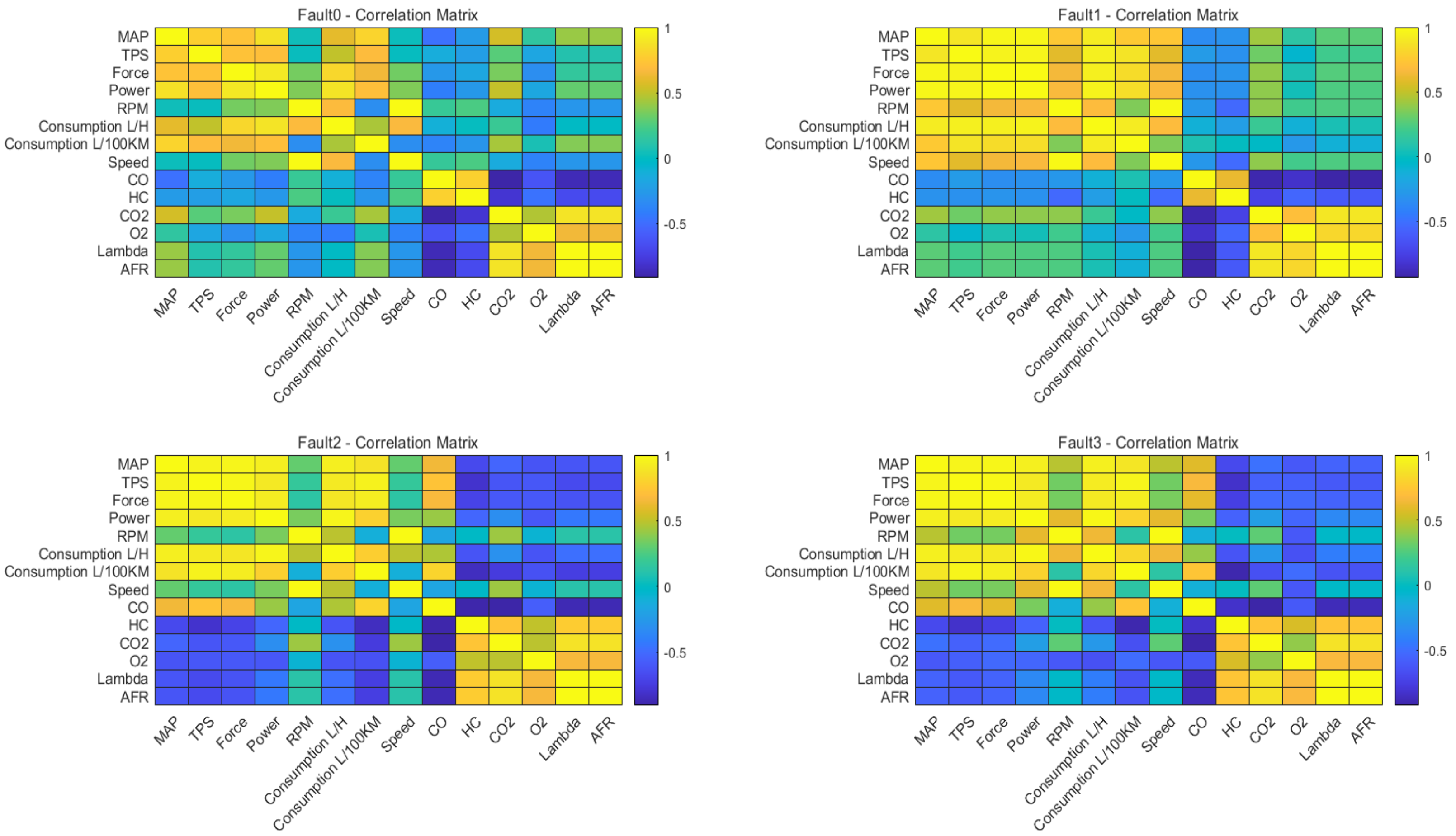
Why KNN & Tree Model has Better Performance than others?

✓ KNN has benefited from having variables with **similar patterns**

✓ Decision Tree(clear classification rules) has advantage in **multiple input**

❖ KNN & Tree is vulnerable to **noise → Butterworth Lowpass Filter**

5. Result & Discussion



- ✓ **Fault2 & Fault3** Show **Similar Tendency.**→ **Difficult to Classification**
- ✓ **AFR, CO, CO2, O2** Correlation: **Exhaust, Combustion Process**.(Fault3)
- ✓ **MAP, TPS, Power, Force** Correlation: **Engine Intake System**. (Fault3)

HANDONG GLOBAL UNIVERSITY

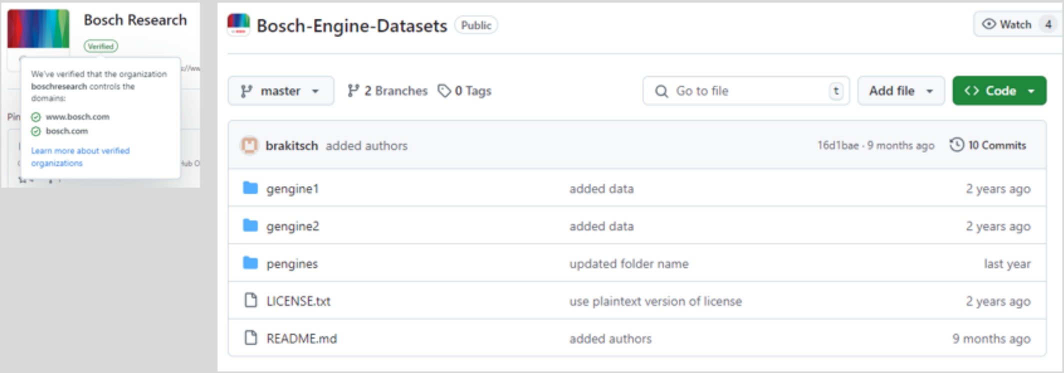
6. Additional Research
- BOSCH Datasets

5. Additional Research

Further Study

Apply the Model to the Open BOSCH Engine Dataset Available on BOSCH's Official GitHub.

Datasets Information



*Unlabeled about Engine Fault

Datasets #

#1. gengine 1
- Speed, Load, Lambda, Ignition Angle, Fuel cutoff, CO, CO2, HC, NOx, O2, Temperature(Manifold), Temperature (Catalyst)

#2. gengine 2
- Speed, Load, Lambda, Ignition Angle, HC, NOx, O2, Temperature(Manifold), Temperature (Catalyst)

#3. pengines
- engine speed, engine load, intake valve opening, air fuel ratio, specific fuel consumption, temperature exhaust manifold, temperature (Catalyst), cylinder pressure, HC, NOx

[5] Bosch Research. (n.d.). Bosch Engine Datasets. GitHub. https://github.com/boschresearch/Bosch-Engine-Datasets

Strategy:

① Using **regression techniques**, **output variables** can be predicted from **input variables**.

② In other words, **physical data that is easily obtainable**, such as engine speed, load, fuel cutoff, etc., **can be used to predict** CO, CO2, HC, etc., as well as temperatures within the engine manifold — data that typically requires a gas analyzer.

③ This allows for **identifying types of engine faults** using only easily accessible data and the predicted output variables.

5. Additional Research

Fuel Cutoff Data Preprocessing

Box Plot & Outlier Replacement

Noise Reduction (Butterworth Low pass filter)

Min-Max Scaling

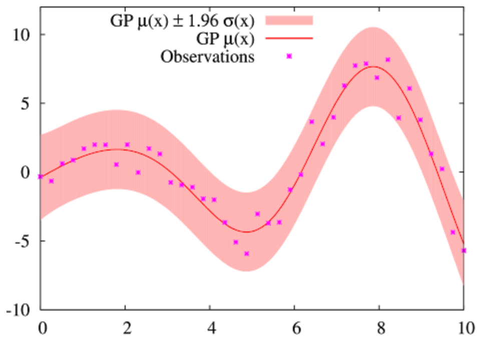
Add Fuel Cutoff Data

Down Sampling

Train/Test Data Split

GPR | Linear Regression | Random Forest | SVR | MOGP

*Python



*** <Gaussian Process Regression>**

Non-parametric, probabilistic regression method that learns the **distribution** of functions.

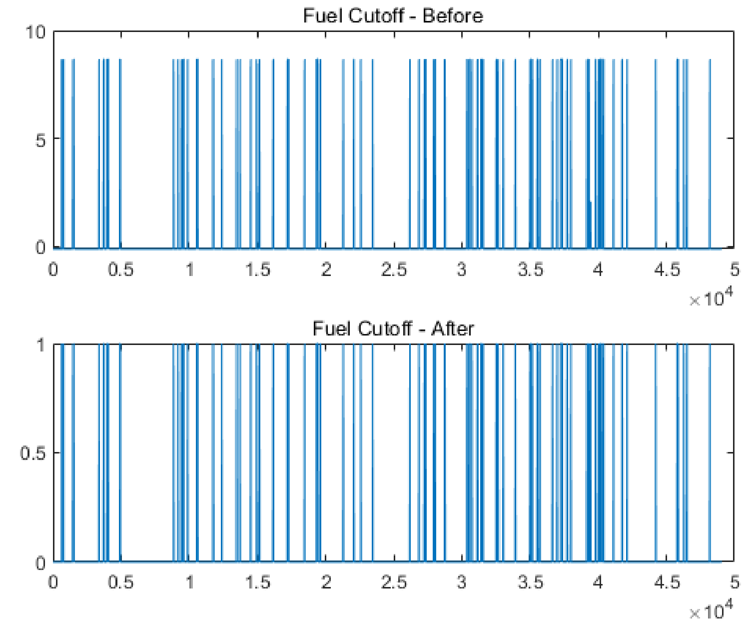
Measures the **correlation** (similarity) between data points.

Data points that are **closer** to each other have **higher similarity**.

Provides **uncertainty estimates** for the data, allowing evaluation of predictions along with **confidence intervals**.

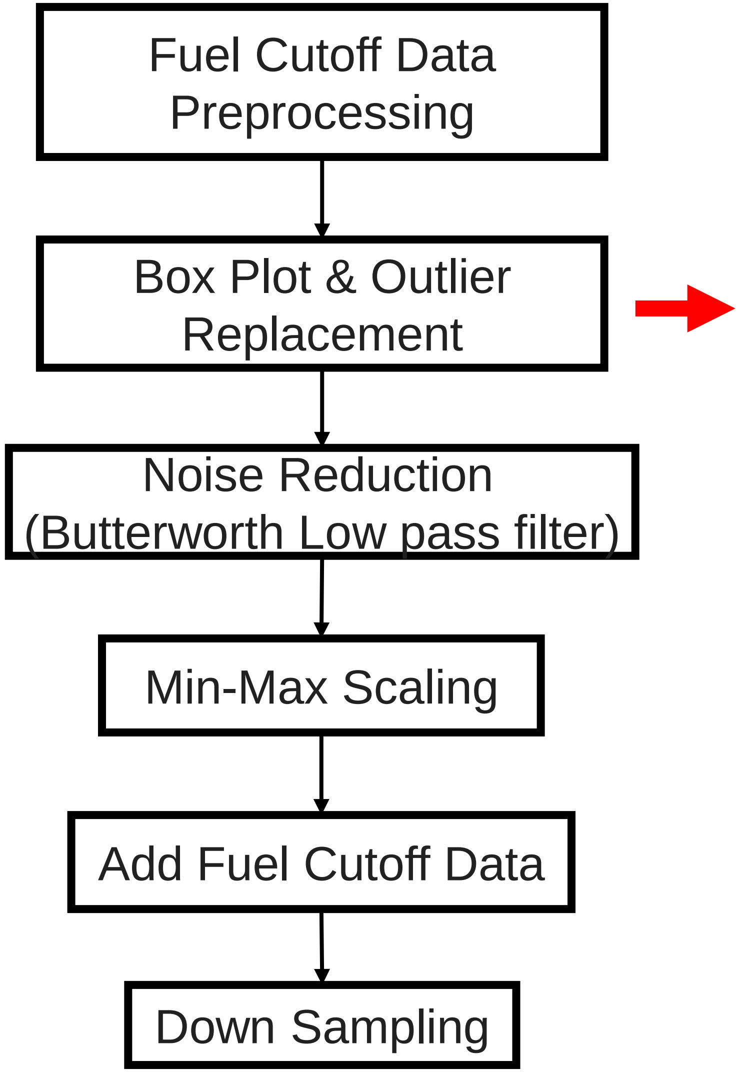
5. Additional Research

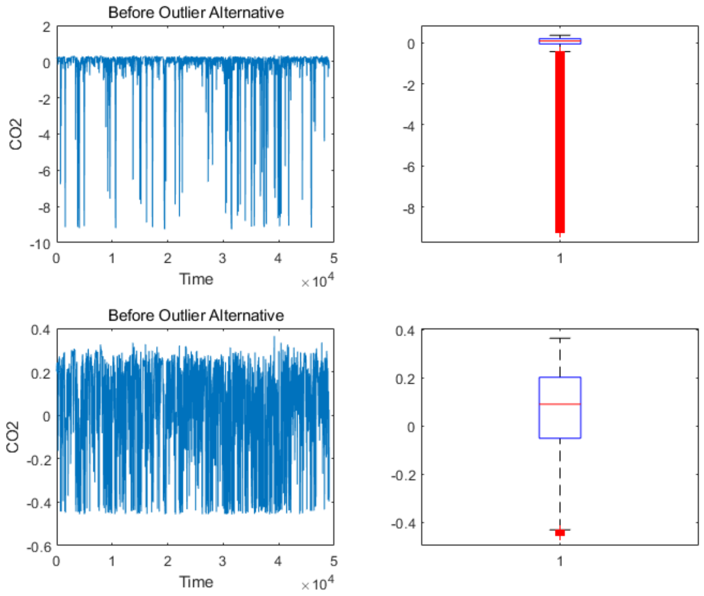
Fuel Cutoff Data Preprocessing
↓
Box Plot & Outlier Replacement
↓
Noise Reduction (Butterworth Low pass filter)
↓
Min-Max Scaling
↓
Add Fuel Cutoff Data
↓
Down Sampling



- ✓ *Fuel Cutoff* data represents the state or timing when the fuel supply to the engine is cut off.
- ✓ *Fuel Cutoff* data is binary, with values of 0 or 1
- ✓ Excluded from preprocessing
- ✓ $\begin{cases} Fuel\ Cutoff < 0 : Fuel\ Cutoff = 0 \\ Fuel\ Cutoff \geq 0 : Fuel\ Cutoff = 1 \end{cases}$

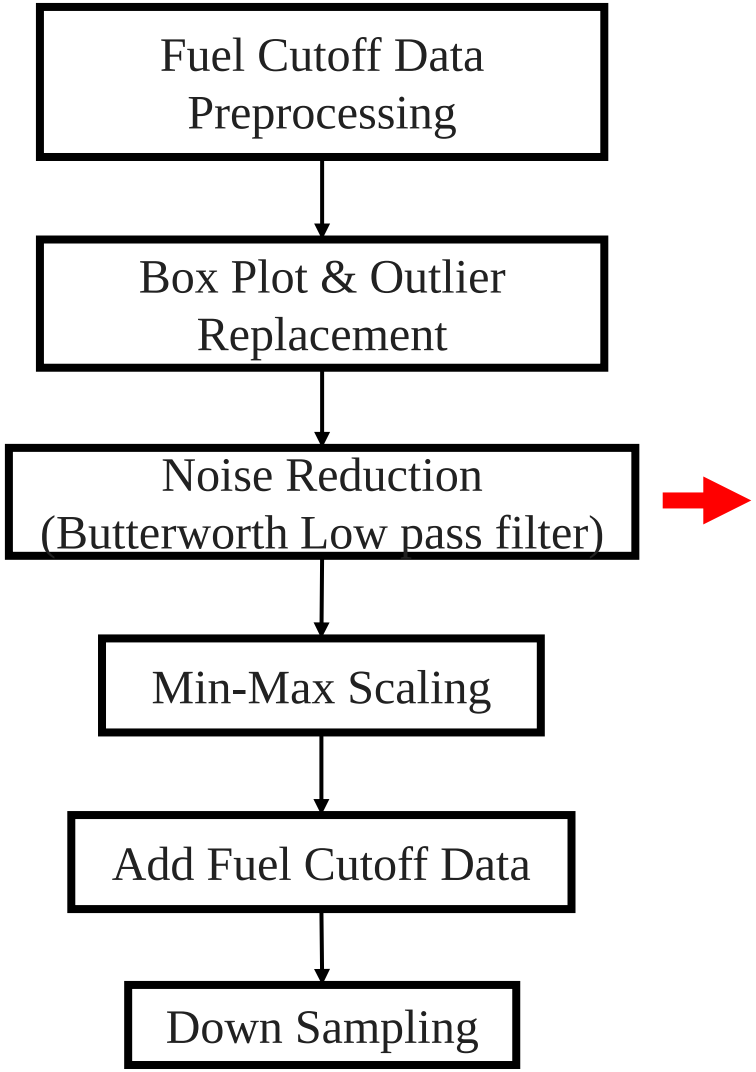
5. Additional Research

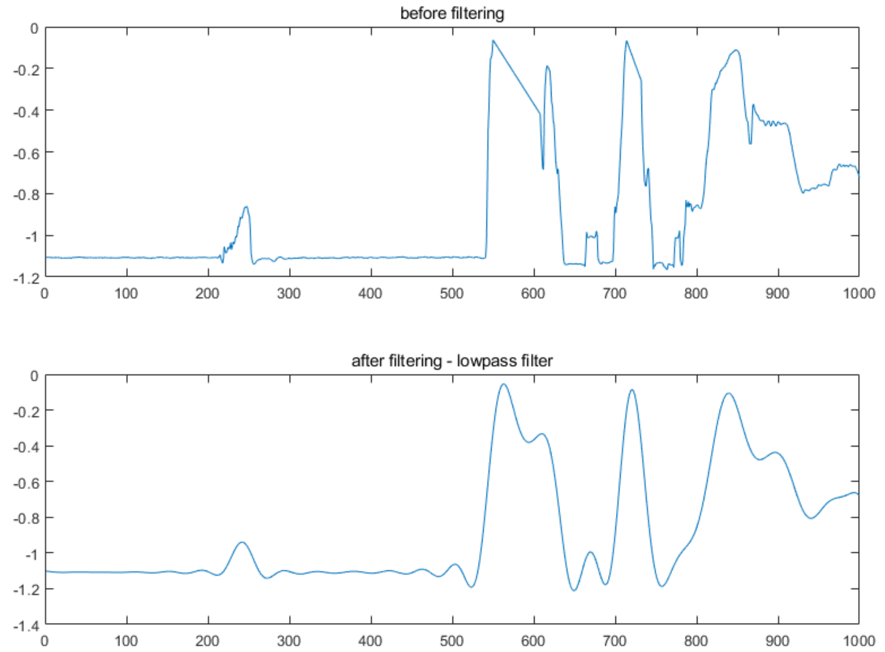




- ✓ Linear interpolation was used to replace outliers.
- ✓ A box plot was utilized to assess the improvement in outlier reduction.

5. Additional Research

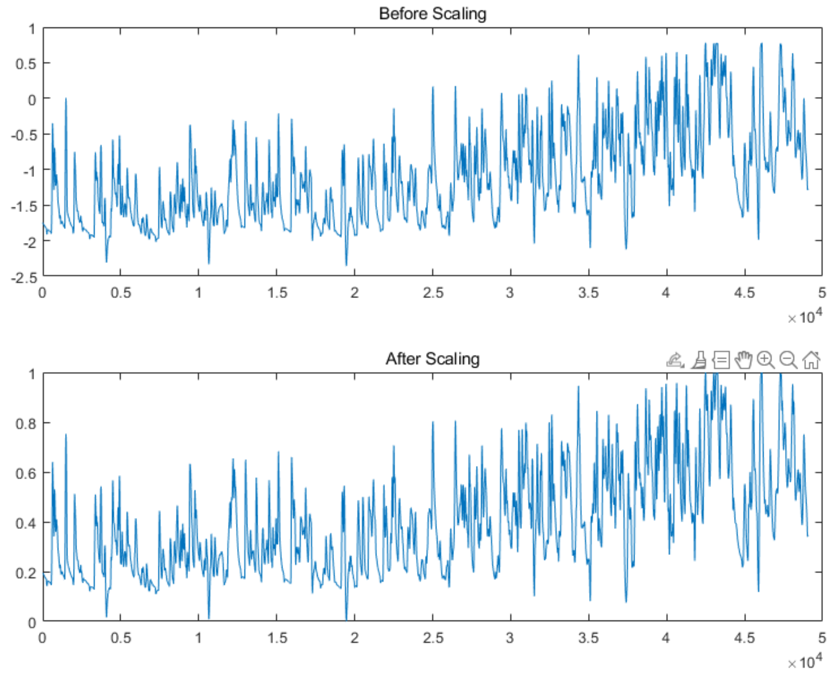




- ✓ A Butterworth low-pass filter was used to manage noise introduced during data acquisition.

5. Additional Research

1. Fuel Cutoff Data Preprocessing
2. Box Plot & Outlier Replacement
3. Noise Reduction (Butterworth Low pass filter)
4. Min-Max Scaling
5. Add Fuel Cutoff Data
6. Down Sampling



- ✓ Scaling was applied due to the differing ranges across variables.
- ✓ Min-Max Scaling was used to normalize the data within a [0, 1] range.

$$X_{\text{scaled}} = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}}$$

5. Additional Research

Fuel Cutoff Data Preprocessing

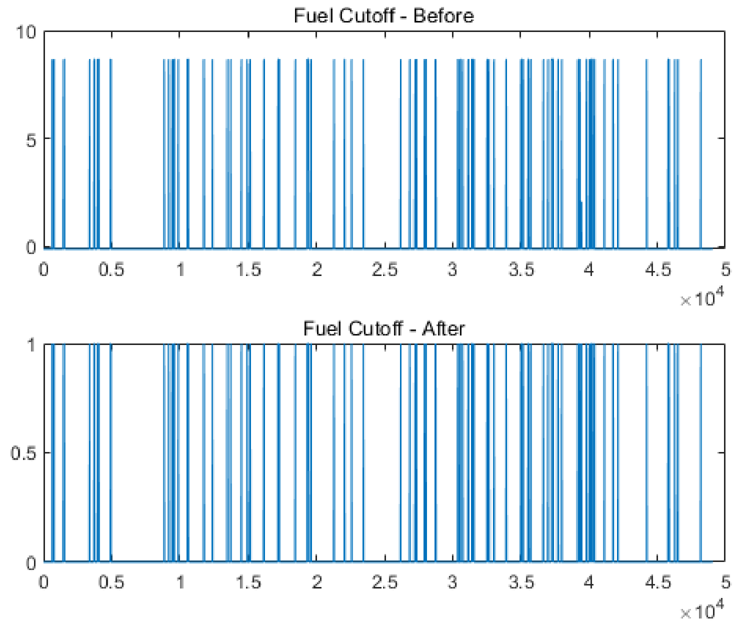
Box Plot & Outlier Replacement

Noise Reduction (Butterworth Low pass filter)

Min-Max Scaling

Add Fuel Cutoff Data

Down Sampling



- ✓ Re-importing Preprocessed Cutoff Data from Step 1
- ✓ The binary data for Fuel Cutoff was not subjected to additional preprocessing.

5. Additional Research

Fuel Cutoff Data Preprocessing

↓

Box Plot & Outlier Replacement

↓

Noise Reduction (Butterworth Low pass filter)

↓

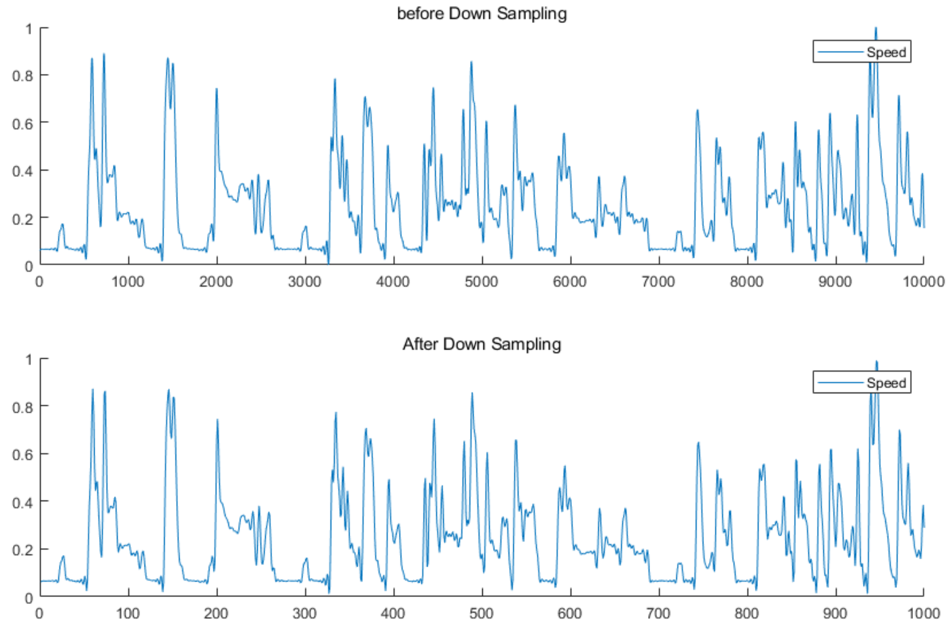
Min-Max Scaling

↓

Add Fuel Cutoff Data

↓

Down Sampling →



- ✓ The computational load of Gaussian Process Regression increases exponentially with the number of input data points.
- ✓ Down-sampling was applied to reduce the data size while preserving trends.
- ✓ Data Entries: 49007 → 4907

5. Additional Research

Train/Test Data Split

GPR

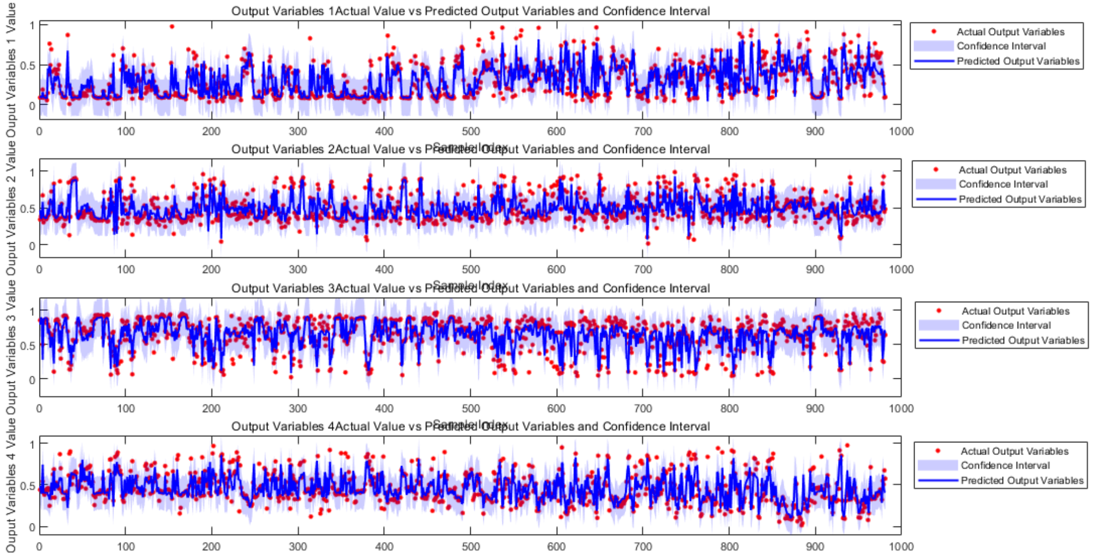
Linear Regression

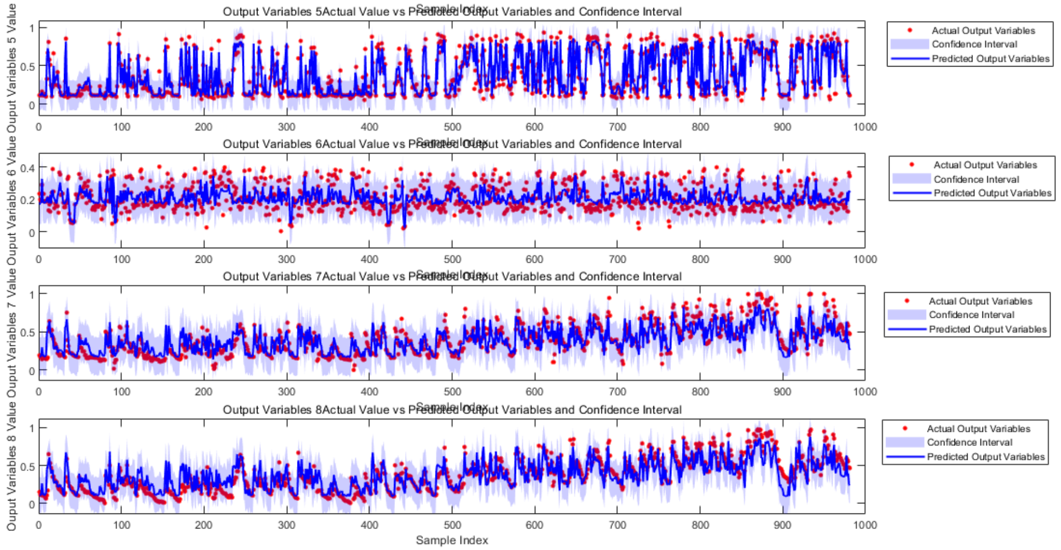
Random Forest

SVR

MOGP

*Python

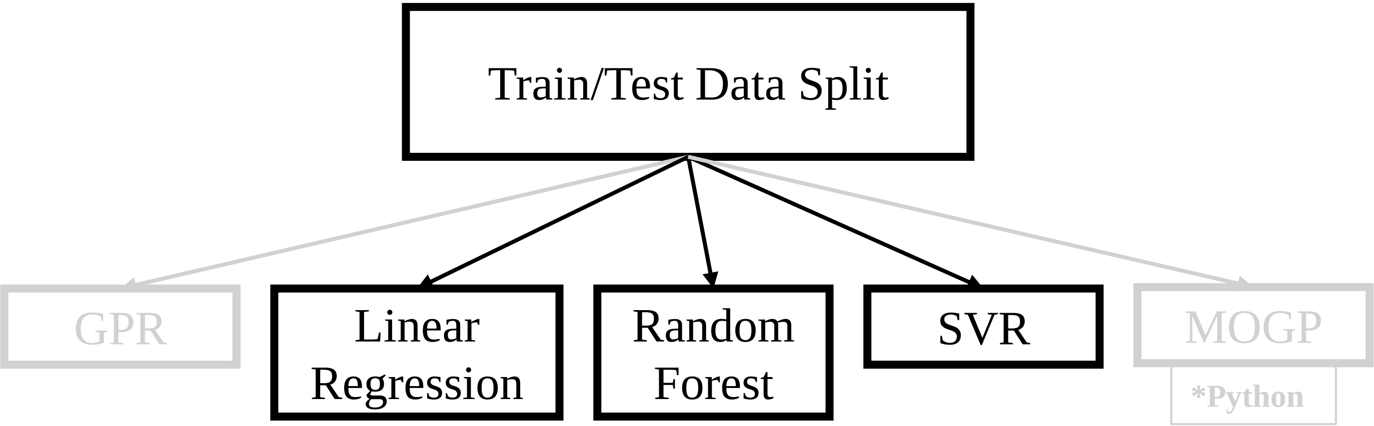




- ✓ Since MATLAB lacks Multi-Output Gaussian Process functions, separate GPR models were trained for each output variable.
- ✓ Confidence intervals and predicted values can be compared to actual values.

✓ GPR Average RMSE = 0.1174

5. Additional Research




```
LinearRegression Model - Output Variables 1 - RMSE: 0.1373
LinearRegression Model - Output Variables 2 - RMSE: 0.1526
LinearRegression Model - Output Variables 3 - RMSE: 0.1944
LinearRegression Model - Output Variables 4 - RMSE: 0.1411
LinearRegression Model - Output Variables 5 - RMSE: 0.1296
LinearRegression Model - Output Variables 6 - RMSE: 0.0781
LinearRegression Model - Output Variables 7 - RMSE: 0.1513
LinearRegression Model - Output Variables 8 - RMSE: 0.1438
```

```
RandomForest Model - Output Variables 1 - RMSE: 0.1180
RandomForest Model - Output Variables 2 - RMSE: 0.1269
RandomForest Model - Output Variables 3 - RMSE: 0.1529
RandomForest Model - Output Variables 4 - RMSE: 0.1149
RandomForest Model - Output Variables 5 - RMSE: 0.1023
RandomForest Model - Output Variables 6 - RMSE: 0.0656
RandomForest Model - Output Variables 7 - RMSE: 0.1272
RandomForest Model - Output Variables 8 - RMSE: 0.1195
```

```
SVM Model - Output Variables 1 - RMSE: 0.1378
SVM Model - Output Variables 2 - RMSE: 0.1564
SVM Model - Output Variables 3 - RMSE: 0.2031
SVM Model - Output Variables 4 - RMSE: 0.1456
SVM Model - Output Variables 5 - RMSE: 0.1312
SVM Model - Output Variables 6 - RMSE: 0.0810
SVM Model - Output Variables 7 - RMSE: 0.1530
SVM Model - Output Variables 8 - RMSE: 0.1444
```

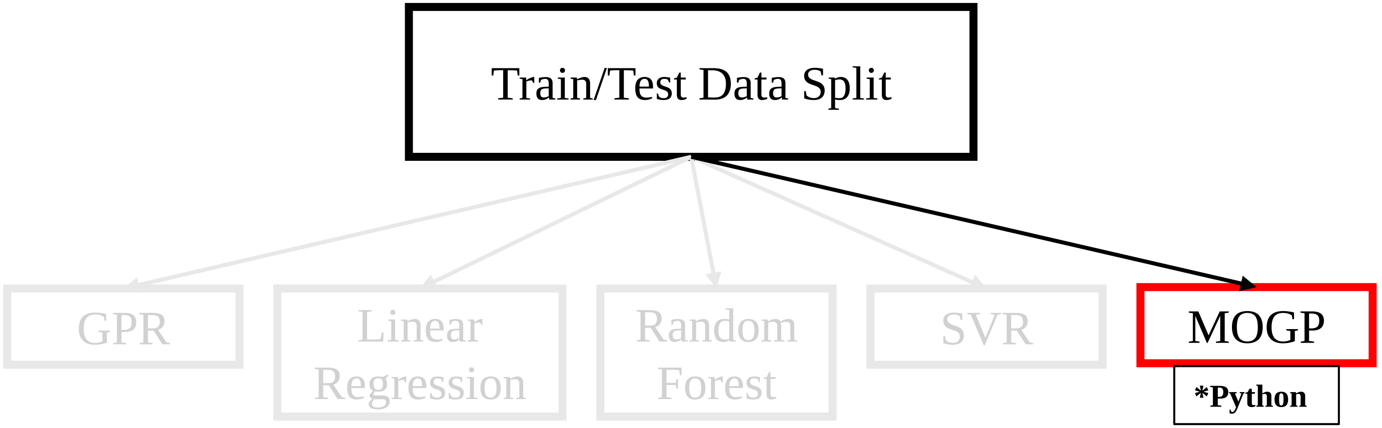
- ✓ Three additional regression models were trained to compare performance with Gaussian Process Regression (GPR).

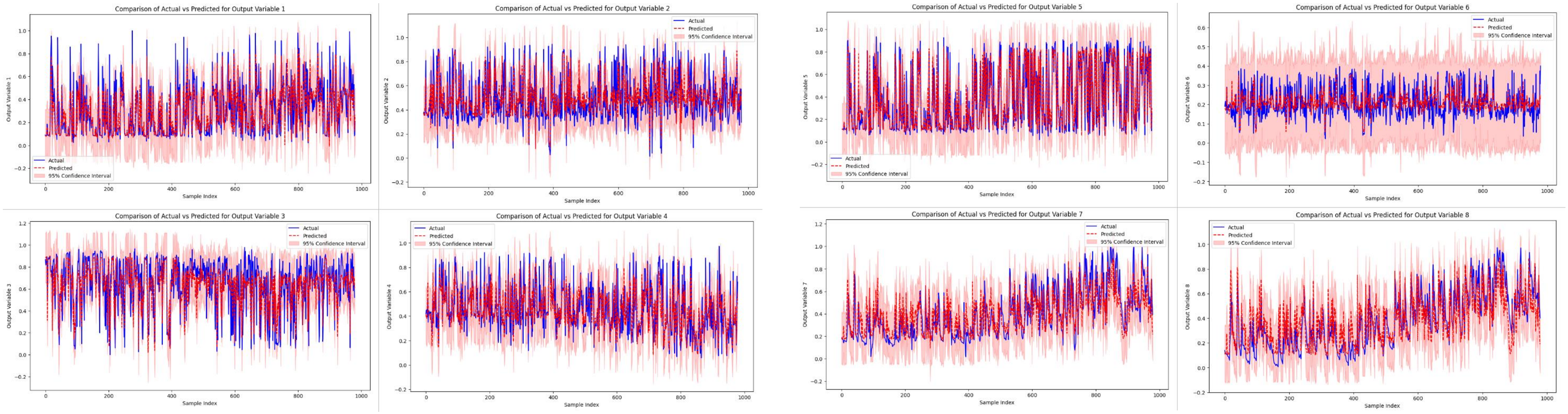
✓ Linear Regression Average RMSE = 0.1410

✓ Random Forest Regression Average RMSE = 0.1159

✓ SVR Average RMSE = 0.1441

5. Additional Research





- ✓ MATLAB does not support Multi-Output Gaussian Process (MOGP), but Python does through the GPy module.
- ✓ MOGP Average RMSE = 0.1188

5. Additional Research – Result & Discussion

	GPR	Linear Regression	Random Forest	SVR	MOGP
RMSE	0.1174	0.1405	0.1180	0.1424	0.1188

✓ **Advantages of GPR/MOGP Over Other Regression Models**

① **Interdependency Modeling:** MOGP captures relationships between multiple output variables, improving accuracy.
② **Uncertainty Estimation:** Provides confidence intervals, enhancing reliability in applications like fault detection.
③ **Non-Linear Capability:** GPR's kernel-based approach effectively models complex, non-linear relationships.

✓ **Computational Considerations of GPR/MOGP**

① **High Cost:** GPR/MOGP is computationally intensive due to the large kernel matrix.
② **Benefit of Large Data:** More data improves accuracy and reliability but increases computational load.

✓ **Conclusion**

GPR/MOGP's advantages in accuracy and uncertainty estimation make it highly **suitable** for **complex, interdependent datasets** like BOSCH Engine Datasets.

✓ **Adaption**

Output Variables must be measured with **expensive equipment** such as a Gas Analyzer, but if Output Variables can be accurately **estimated with Input Variables**, the status or defect of the **engine can be identified using the data**.

Reference

[1] Vergara, M., Ramos, L., Rivera-Campoverde, N. D., & Rivas-Echeverría, F. (2023). Enginefaultdb: a novel dataset for automotive engine fault classification and baseline results. *IEEE Access*, *11*, 126155-126171.

[2] Li, C. Y., Rakitsch, B., & Zimmer, C. (2022, May). Safe active learning for multi-output gaussian processes. In International Conference on Artificial Intelligence and Statistics (pp. 4512-4551). PMLR.