## INTRODUCTION

In part 2 of this project, classification algorithms are used to learn how to classify single faults. Eight machine learning algorithms are compared in terms of performance by evaluating metrics like precision, recall, accuracy, f1-scores and area under the (receiver operating) curve. The empirical results are shown in the tables below. Each model is fitted with and without hyperparameter tuning.

## **RESULTS**

#### **GRU**

		Evaluation Metric				
Classification		Precisio		Accurac	F1-	ROC-
Metric	Estimation Type	n	Recall	у	score	AUC
	Without Hyperparameter Tuning	1.67%	9.09%	86.36%	2.82%	66.00%
XG Boost	With Hyperparameter Tuning		27.27			
	with hyperparameter ranning	3.45%	%	81.82%	6.12%	55.00%
	Without Hyperparameter Tuning		63.64			
Logistic regression		2.76%	%	50.40%	5.28%	58.00%
	With Hyperparameter Tuning	0.700/	63.64	50.000/	E 0.40/	<b>57.00</b> 0/
	3, ,	2.79%	%	50.99%	5.34%	57.00%
	Without Hyperparameter Tuning	2.060/	36.36	72 720/	F 400/	62.000/
Adaboost		2.96%	% 45.45	72.73%	5.48%	63.00%
	With Hyperparameter Tuning	3.05%	45.45 %	67.39%	5.71%	57.00%
	Without Hyperparameter Tuning	0.00%	0.00%	85.57%	0.00%	62.00%
Gradient Boosting	J					
	With Hyperparameter Tuning	2.78%	9.09%	91.11%	4.26%	51.00%
Dagisian Trans	Without Hyperparameter Tuning With Hyperparameter Tuning	1.59%	9.09%	85.77%	2.70%	44.00%
Decision Trees		2 200/	27.27 %	72 120/	4 220/	F1 000/
		2.29%	% 45.45	73.12%	4.23%	51.00%
	Without Hyperparameter Tuning	2.33%	43.43 %	51.19%	3.89%	
SVM		2.5570	18.18	31.1370	3.0370	
	With Hyperparameter Tuning	1.37% % 69.76%	2.55%			
	Without Hyperparameter Tuning	0.00%	0.00%	91.50%	0.00%	54.00%
Random Forests	With Hyperparameter Tuning	0.00%	0.00%	91.90%	0.00%	47.00%
	3, ,	0.0070	18.18	51.5070	0.0070	77.0070
KNN	Without Hyperparameter Tuning	2.13%	%	80.04%	3.81%	49.00%
1/1414	With Hyperparameter Tuning	16.95%	9.09%	86.56%	2.86%	49.00%
	Jr. F. S.	1				

For anomalies classified using the best (after hyperparameter tuning) GRU model, the highest accuracy of 91.8% was achieved using random forest model with hyperparameter tuning. The precision, recall, and f1-scores are 0 since the data was very imbalanced (few anomalies overall). This model had an roc-auc score of 0.47 which is below the 0.5 cut off point. A better model is random forests without hyperparameter tuning which achieved an accuracy of 91.5% and an roc-auc score of 0.54.

#### **LTSM**

		Evaluation Metric					
Classification		Precisio		Accurac	F1-	ROC-	
Metric	Estimation Type	n	Recall	у	score	AUC	
	Without Hyperparameter Tuning		50.00				
XG Boost	without Hyperparameter runing	4.98%	%	60.28%	9.05%	61.00%	
AG B003t	With Hyperparameter Tuning		25.00				
	with hyperparameter running	4.81%	%	77.47%	8.06%	52.00%	
	Without Hyperparameter Tuning		30.00				
Logistic regression	Williage Hyperparameter ranning	2.53%	%	51.58%	4.67%	46.00%	
209/34/67/69/633/6/7	With Hyperparameter Tuning		30.00				
	The state of the	2.42%	%	4.94%	4.48%	40.00%	
	Without Hyperparameter Tuning		20.00				
Adaboost	3	3.67%	%	76.09%	6.20%	55.00%	
	With Hyperparameter Tuning	- aaa,	50.00	60 670/	0.400/	E.C. 0.00/	
	3, ,		9.13%	56.00%			
Gradient Boosting	Without Hyperparameter Tuning	2 740/	10.00	00.440/	4.200/	F2 000/	
	3, ,	2.74%	%	82.41%	4.30%	52.00%	
_	With Hyperparameter Tuning	0.000/	10.00	04 070/	0.000/	F2 000/	
		8.00%	% 15.00	91.87%	8.89%	53.00%	
	Without Hyperparameter Tuning	5.36%	15.00	86.17%	7.89%	53.00%	
<b>Decision Trees</b>		3.30%	55.00	00.1770	7.0970	33.00%	
	With Hyperparameter Tuning	5.50%	33.00 %	60.87%	10.00%	58.00%	
		3.5070	35.00	00.0770	10.0070	30.0070	
	Without Hyperparameter Tuning	3.80%	%	62.45%	6.86%		
SVM		3.0070	30.00	02.1570	0.0070		
	With Hyperparameter Tuning	3.73%	%	66.60%	6.63%		
			10.00				
Random Forests	Without Hyperparameter Tuning	8.00%	%	91.90%	8.89%	56.00%	
	, <del>,</del> ,		10.00				
	With Hyperparameter Tuning	7.41%	%	91.50%	8.51%	52.00%	
	Without I hyperparenter Touris		20.00				
KNN	Without Hyperparameter Tuning	3.92%	%	77.47%	6.56%	53.00%	
KIVIV	With Hungrages star Tuning		10.00				
	With Hyperparameter Tuning	4.00%	%	88.74%	6.56%	51.00%	

For anomalies generated using the best LTSM score, the highest accuracy was again achieved using random forests, this time without hyperparameter tuning. This model had an accuracy of 91.9% and roc-auc of 56%. Other well-performing models are random forests with hyperparameter tuning (accuracy=91.5%, roc-auc = 52%) and Gradient Boosting without hyperparameter tuning (accuracy=91.87%, roc-auc = 53%). The precision, recall, and f1-scores are low since the data was very imbalanced (few anomalies overall).

### **RNN**

			Evaluation Metric				
Classification		Precisio		Accurac	F1-	ROC-	
Metric	Estimation Type	n	Recall	y	score	AUC	

	Without Hyperparameter Tuning		33.33			
XG Boost	Thursday types per amount to an integral	1.53%	%	83.40%	16.00%	59.00%
	With Hyperparameter Tuning		16.67			
	With Hyperparameter ranning	5.00%	%	81.03%	7.69%	50.00%
	Without Hyperparameter Tuning		70.83			
Logistic regression	Without Hyperparameter ranning	5.94%	%	45.45%	10.97%	54.00%
Logistic regression	With Hyperparameter Tuning		70.83			
	with Hyperparameter ranning	5.94%	%	45.45%	10.97%	58.00%
	Without Hyperparameter Tuning		25.00			
Adaboost	without Hyperparameter running	5.13%	%	74.51%	8.51%	56.00%
71000000	With Hyperparameter Tuning		45.83			
	with Hyperparameter ranning	7.97%	%	7.23%	13.58%	60.00%
	Without Hyperparameter Tuning		12.50			
Gradient Boosting		4.17%	%	82.21%	6.25%	59.00%
	With Hyperparameter Tuning	6.90%	8.33%	90.32%	7.55%	51.00%
Decision Trees	Without Hyperparameter Tuning		16.67			
Decision rices	without Hyperparameter running	6.67%	%	84.98%	9.52%	55.00%
	With Hyperparameter Tuning		50.00			
		5.85%	%	59.49%	10.48%	55.00%
	Without Hyperparameter Tuning		37.50			
SVM	without Hyperparameter running	4.97%	%	63.04%	8.78%	
37777	With Hyperparameter Tuning		29.17			
	with Hyperparameter ranning	4.00%	%	63.44%	7.04%	
Random Forests	Without Hyperparameter Tuning	0.00%	0.00%	88.74%	0.00%	61.00%
Nulluulli Fulests	With Hyperparameter Tuning	0.00%	0.00%	88.74%	0.00%	53.00%
	Without Hypernarameter Tuning		29.17			
IZATAT	Without Hyperparameter Tuning	7.37%	%	79.25%	11.76%	56.00%
KNN	With the armarameter Timing		16.67			
	With Hyperparameter Tuning	8.89%	%	8.79%	11.59%	54.00%

For anomalies generated using RNN, the highest accuracy was achieved by the Gradient Boosting algorithm with hyperparameter tuning accuracy=90.32%, roc-auc = 51%). The precision, recall, and f1-scores are low since the data was very imbalanced (few anomalies overall).

# **ANN**

		Evaluation Metric				
Classification		Precisio		Accurac	F1-	ROC-
Metric	Estimation Type	n	Recall	у	score	AUC
	Without Hyperparameter Tuning 4.		64.71			
VC Danet		4.76%	%	55.34%	8.87%	63.00%
XG Boost	With Hyperparameter Tuning		11.76			
		2.47%	%	81.42%	4.08%	48.00%
Logistic regression	Mith out II was an arrange atox Tuning		64.71			
	Without Hyperparameter Tuning	4.04%	%	47.23%	7.61%	57.00%
	With Hyperparameter Tuning	3.93%	64.71	45.65%	7.41%	55.00%

			%			
	Without Hyperparameter Tuning		35.29			
		4.11%	%	70.16%	7.36%	63.00%
Adaboost			64.71			
	With Hyperparameter Tuning	5.82%	%	63.64%	10.68%	64.00%
	Market and the second second as Tourism		17.65			
Gradient Boosting	Without Hyperparameter Tuning	3.85%	%	82.41%	6.32%	59.00%
	With Hyperparameter Tuning	2.38%	5.88%	88.74%	3.39%	49.00%
	Without Hyperparameter Tuning	1.69%	5.88%	85.38%	2.63%	46.00%
Decision Trees	With Hyperparameter Tuning		64.71			
		4.64%	%	54.15%	8.66%	59.00%
	Without Hyperparameter Tuning		64.71			
CVAA		3.89%	%	45.06%	7.33%	
SVM	New T		41.18			
	With Hyperparameter Tuning	4.19%	%	66.40%	7.61%	
Days days Fayasta	Without Hyperparameter Tuning	2.94%	5.88%	90.32%	3.92%	54.00%
Random Forests	With Hyperparameter Tuning	2.78%	5.88%	89.92%	3.77%	49.00%
KNN	LACTA A T		17.65			
	Without Hyperparameter Tuning	2.46%	%	73.72%	4.32%	43.00%
	With Hyperparameter Tuning	1.41%	5.88%	83.00%	2.27%	46.00%

For anomalies generated using ANN, the best model was random forests without hyperparameter tuning accuracy=90.32%, roc-auc = 54%). The precision, recall, and f1-scores are low since the data was very imbalanced (few anomalies overall).

The best anomaly classification model appears to be random forests followed by gradient boosting.