ITU AI/ML 5G Challenge

Network Failure Prediction

(Problem Statement by KDDI)

Team Name: The Big Fools Team

Affiliation: The University of Malawi

Member: Emmanuel Basikolo

ML5G-PS-005: Network failure prediction on CNFs 5GC with Linux eBPF

Agenda

01 Introduction

- ✓ Introduction

02 Problem statement and Data Extraction

- ✓ Problem statement
- ✓ Description of task 1 and 2
- ✓ Feature extraction for task 2

03 Proposed ML Models

- ✓ Proposed models for used in this project
- ✓ Improvements to models

04 Results

- ✓ Summary of all models' performance
- ✓ Summary of best model's outputs

05 Conclusion

Part 1:
Introduction

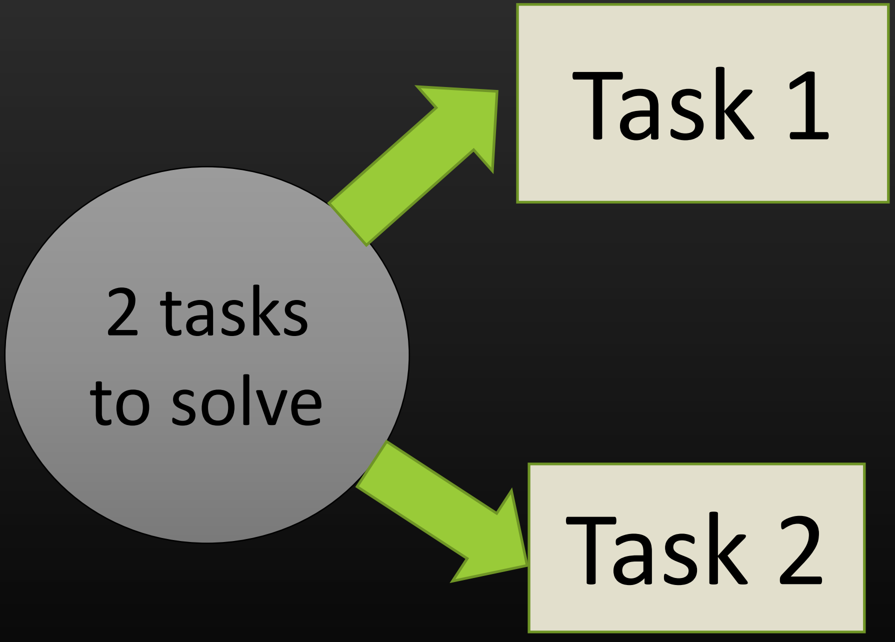
Introduction

- ✓ Commercial 5G network deployments
- ✓ Need proactive operation to minimize 5G service outages due to network failures
- ✓ Problems associated with network outage
- ✓ How machine learning solves these problems
- ✓ Proposed ML solution to predict future network failures on the 5GC

Part 2:
Problem statement and Data Extraction

Problem statement

- ➢ to predict future network failures on the 5GC using AI/ML
- ➢ how early and accurately the future network failures can be predicted



Task 1:

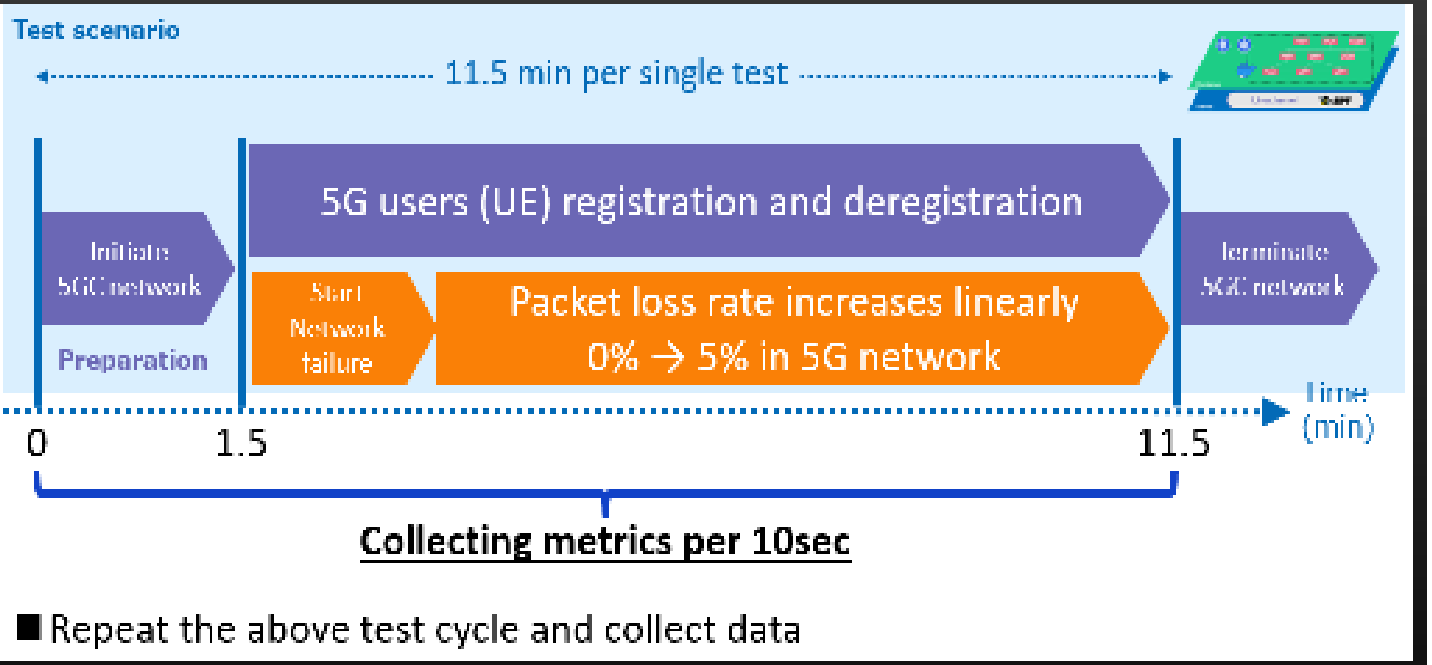
- ❖ use every metrics to create AI/ML models for predicting a network failure which will occur in future
- ❖ target value = # of UE registration failures at 10min (600sec).
- ❖ compete for the detection time t when the models can achieve over 0.9 of *F1-score*

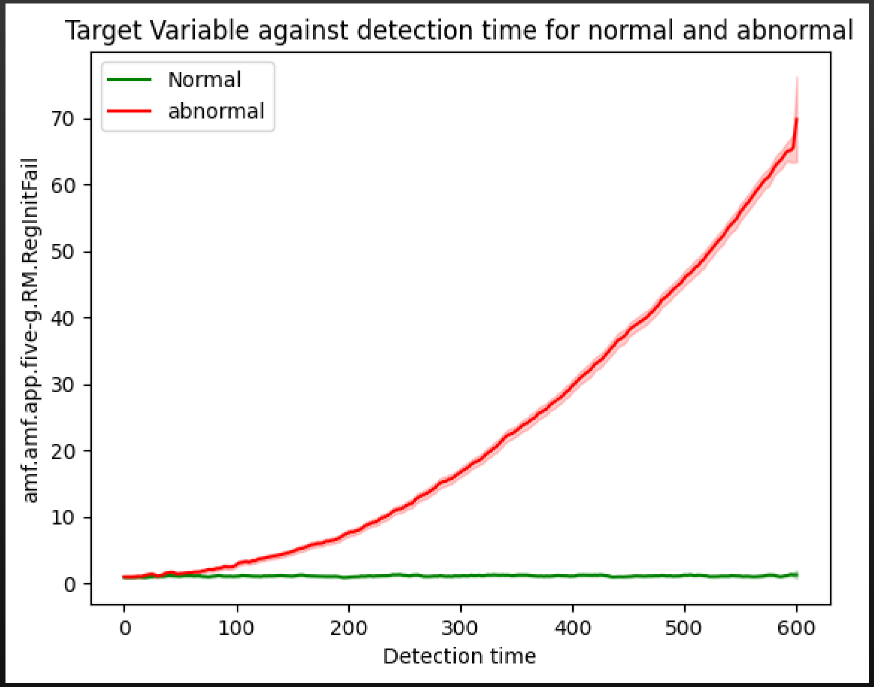
Task 2:

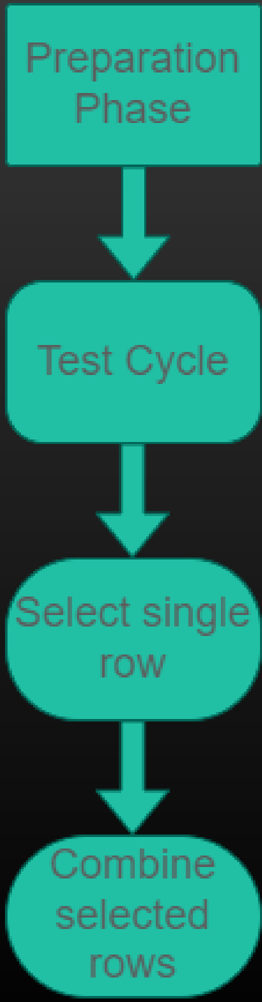
- ➢ select a part of metrics from every metrics to develop models that can predict failures at the detection time t
- ➢ compete for the detection time t and the # of metrics used

Data Collection

- ✓ Preparation phase and registration and deregistration phase
- ✓ Test cycle





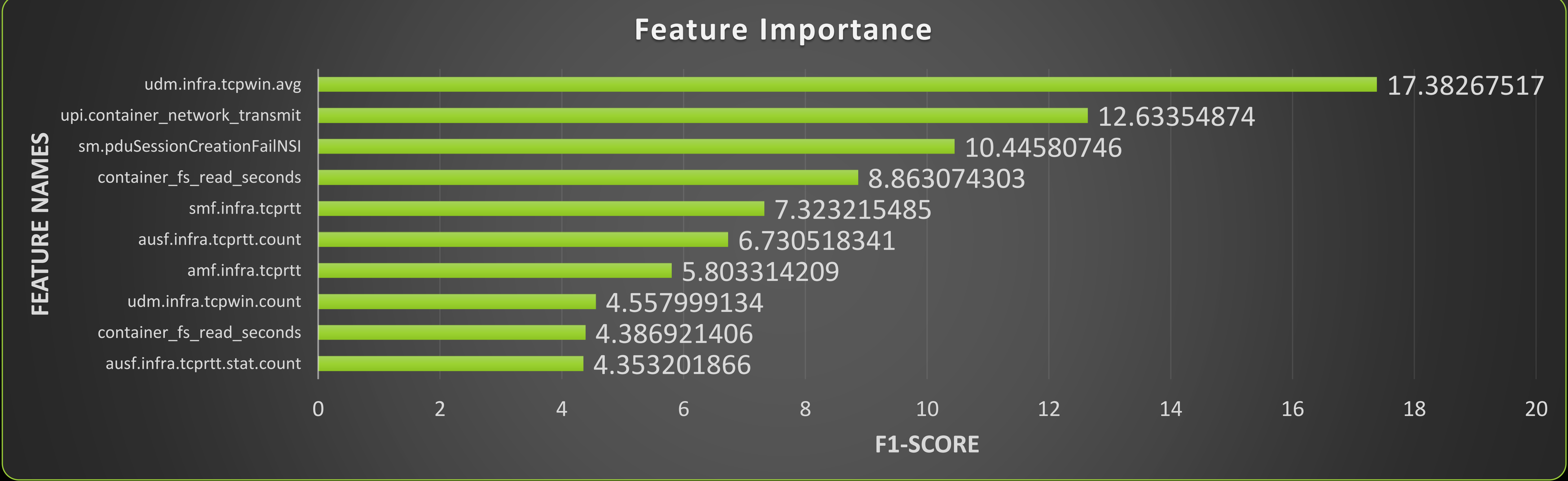


	Total examples	Test cycle size	Selected examples
Training set	42,000 examples	70 examples	600 examples
Test set	21,000 examples	70 examples	300 examples

Dataset For Task 2

- Selected metrics (columns) from selected rows are used
- Metrics are selected based on their importance
- The target variable is *UE registration failures*
- If target variable is greater than certain value, then network failure occurs at 600 seconds

Top 10 Of Most Important Features



Part 3:
Proposed ML Models

Models Proposed

- The problem is treated as Supervised machine learning type

    - Regression models and Decision tree models

Following Models' performance was Compared

- ✓ Support Vector Regression (SVR)
- ✓ Bayesian Ridge
- ✓ Cat Boost Regressor
- ✓ Extreme Gradient Boosting Regressor(XGBR)
- ✓ Random Forest Regressor

Improvements to Conventional Models

- In order to improve model performance, the following was proposed in this work to improve the model performance

  - ✓ SVR Kernel changed (from rbf to sigmoid)

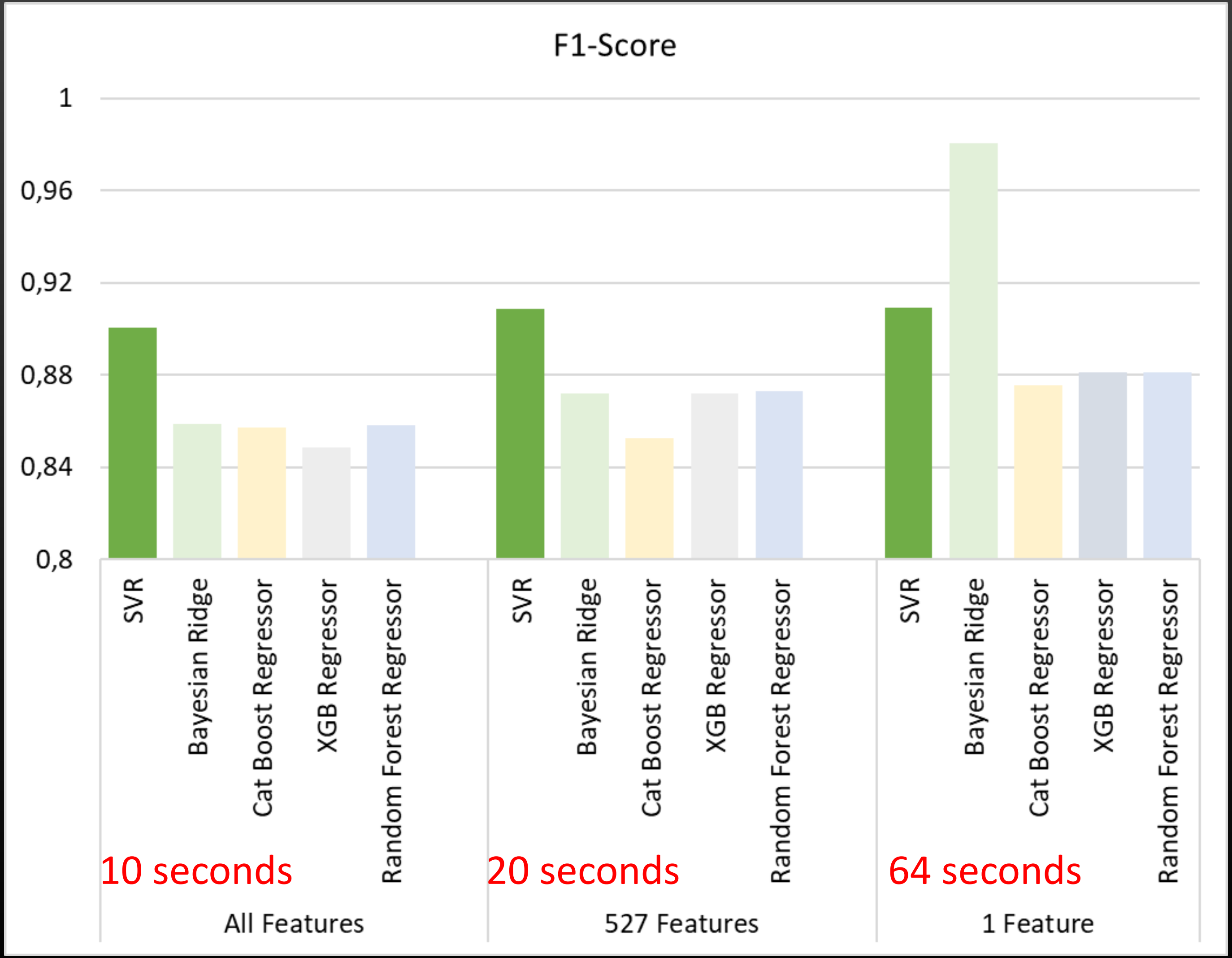
  - ✓ SVR Regulation parameter C, was changed from 1 to 0.095.

Part 4:
ML Model
Performance Results

Analysis of the best model's output

Performance For Different Models

Graph of F1-Score against number of features for different models



Model	Detection Time	# features	F1-Score
SVR	10 seconds	All	0.902
Bayesian Ridge	10 seconds	All	0.859
Cat Boost Regressor	10 seconds	All	0.857
XGB Regressor	10 seconds	All	0.848
RF Regressor	10 seconds	All	0.858

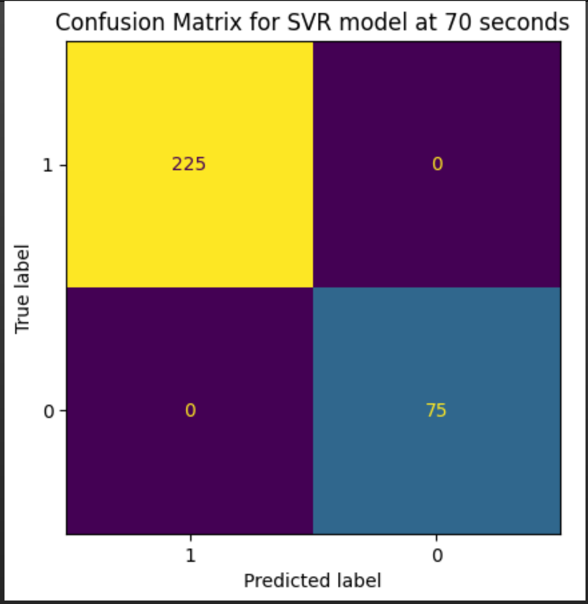
These 2 models achieves better F1 score

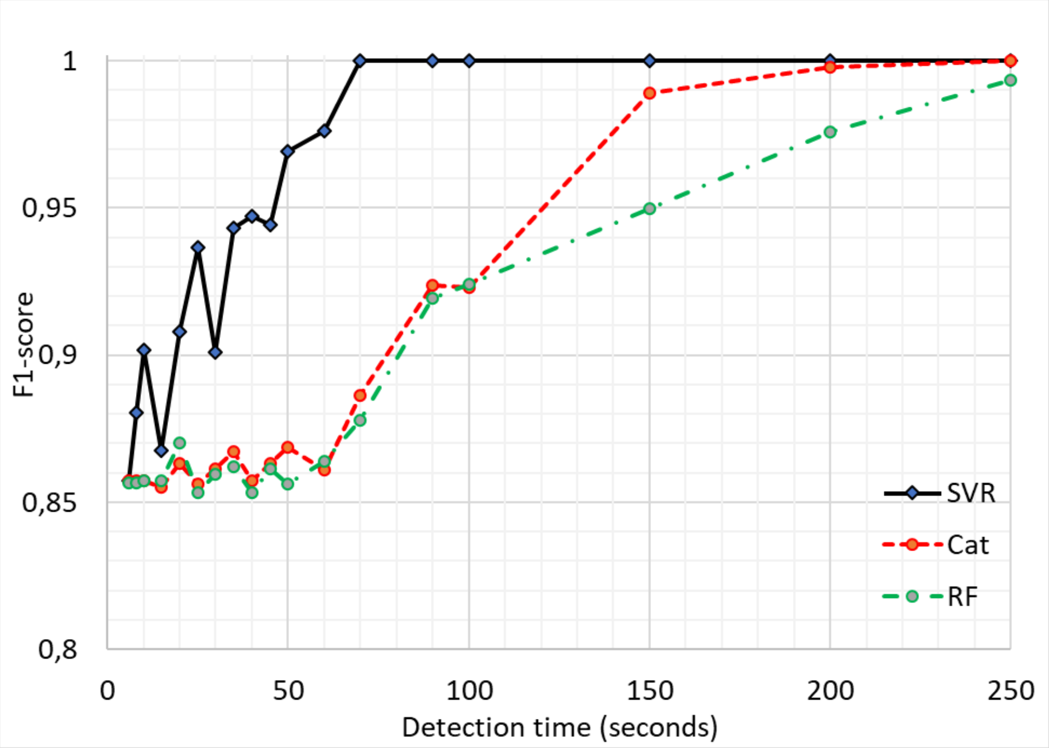
- Support Vector Regression (SVR)
- Bayesian Ridge

Task 1 Results

SVR Model Performance

Detection time	Precision	Recall	F1-Score	Threshold
10 seconds	0.8283582	0.9866667	0,902	0.40
20 seconds	0.841	0.987	0,908	0.48
25 seconds	0.902	0.982	0.940	0.65
70 seconds	1.0	1.0	1.0	0,68
100 seconds	1.0	1.0	1.0	1.45





Actual (True) labels

	1	0
1	222	3
0	46	29

Predicted labels

Confusion matrix for task 1 at 10 seconds using SVR model

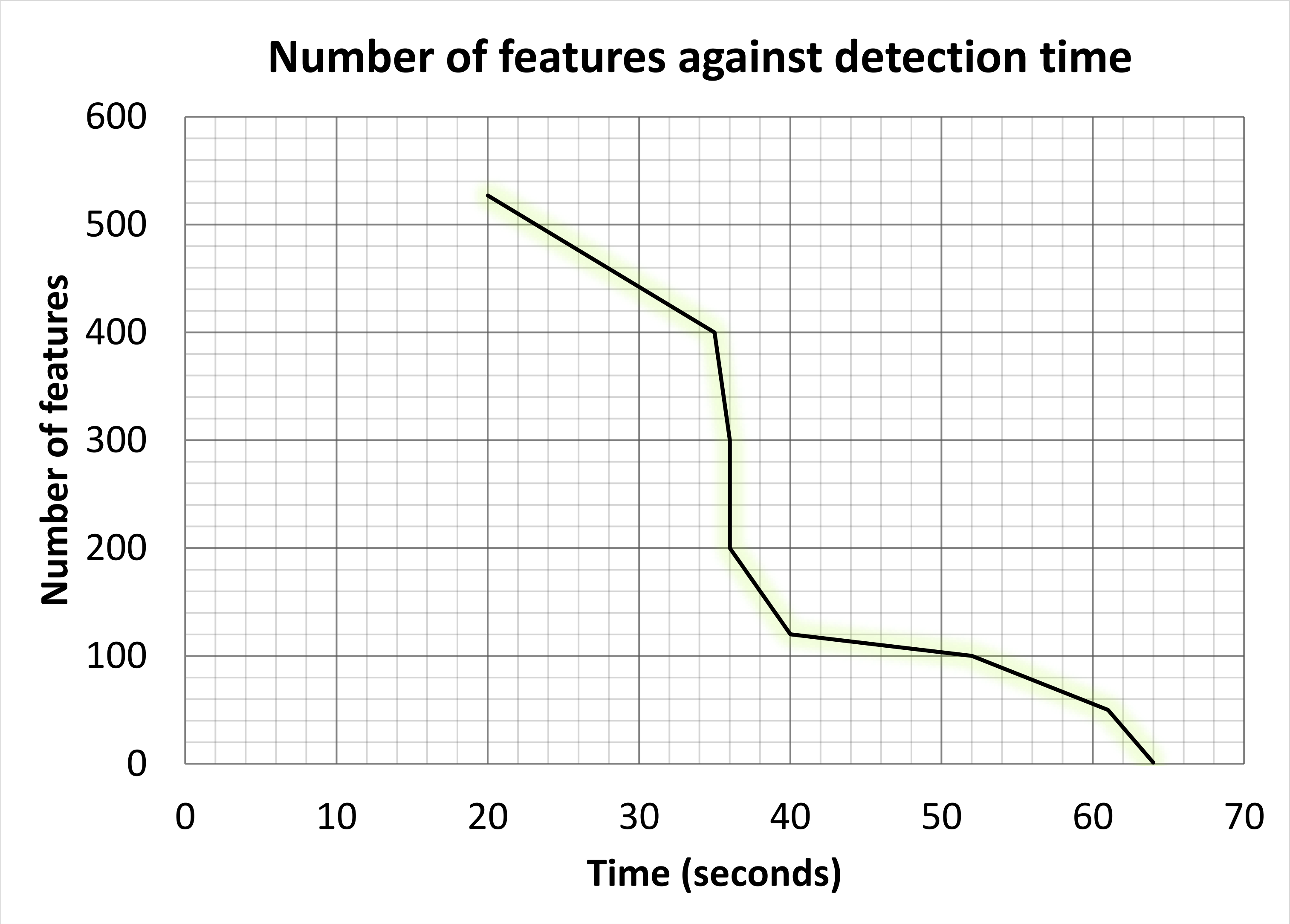
- The model can detect network failure in 10 seconds (100 seconds if preparation phase is included)

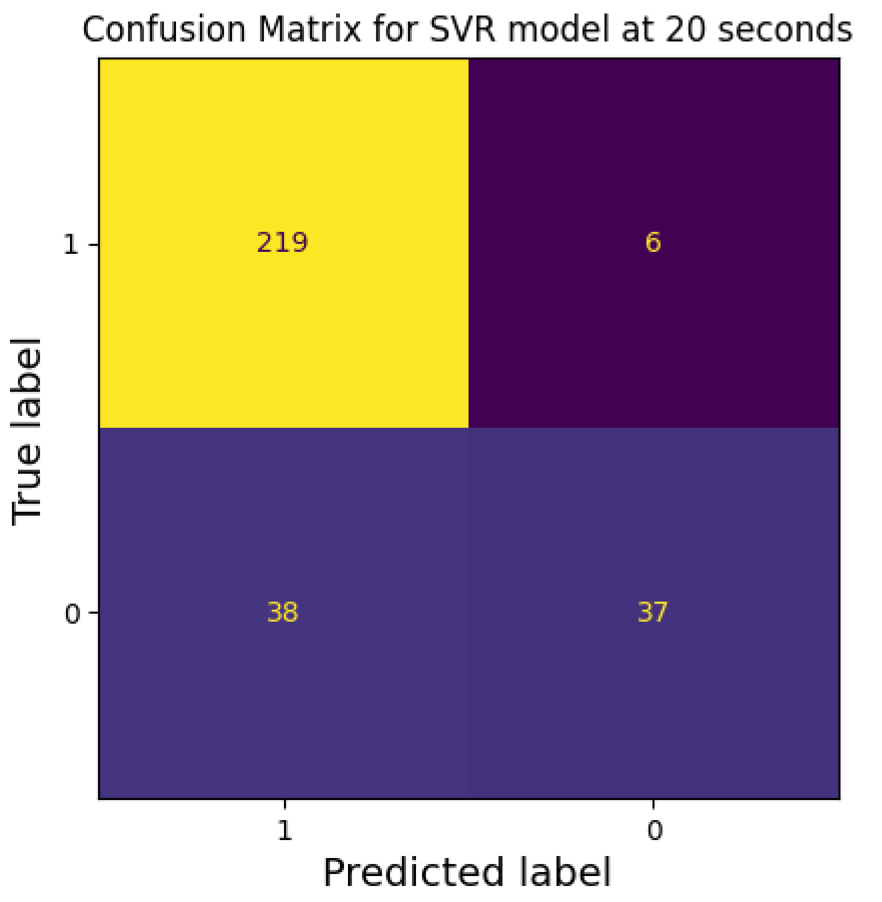
Task 2 Results

# of Features	Time (seconds)	F1-Score	Threshold	Model
1	64	0.980392	1.32	Bayesian Ridge
50	61	0.901010	1.74	SVR
100	52	0.914286	1.14	SVR
120	40	0.902287	1.30	SVR
200	36	0.907975	1.11	SVR
300	36	0.917197	0.62	SVR
400	35	0.905579	0.55	SVR
527	20	0.910603	0.57	SVR

- Using 527 features, the model can detect network failure in 20 seconds (110 seconds if preparation phase is included) which is double the detection time when all features are used as in task 1
- Other features are more important than others and that’s why removing them affects the graph more than other features
- This can be seen from the graph as it was supposed to be linear

Task 2 Results

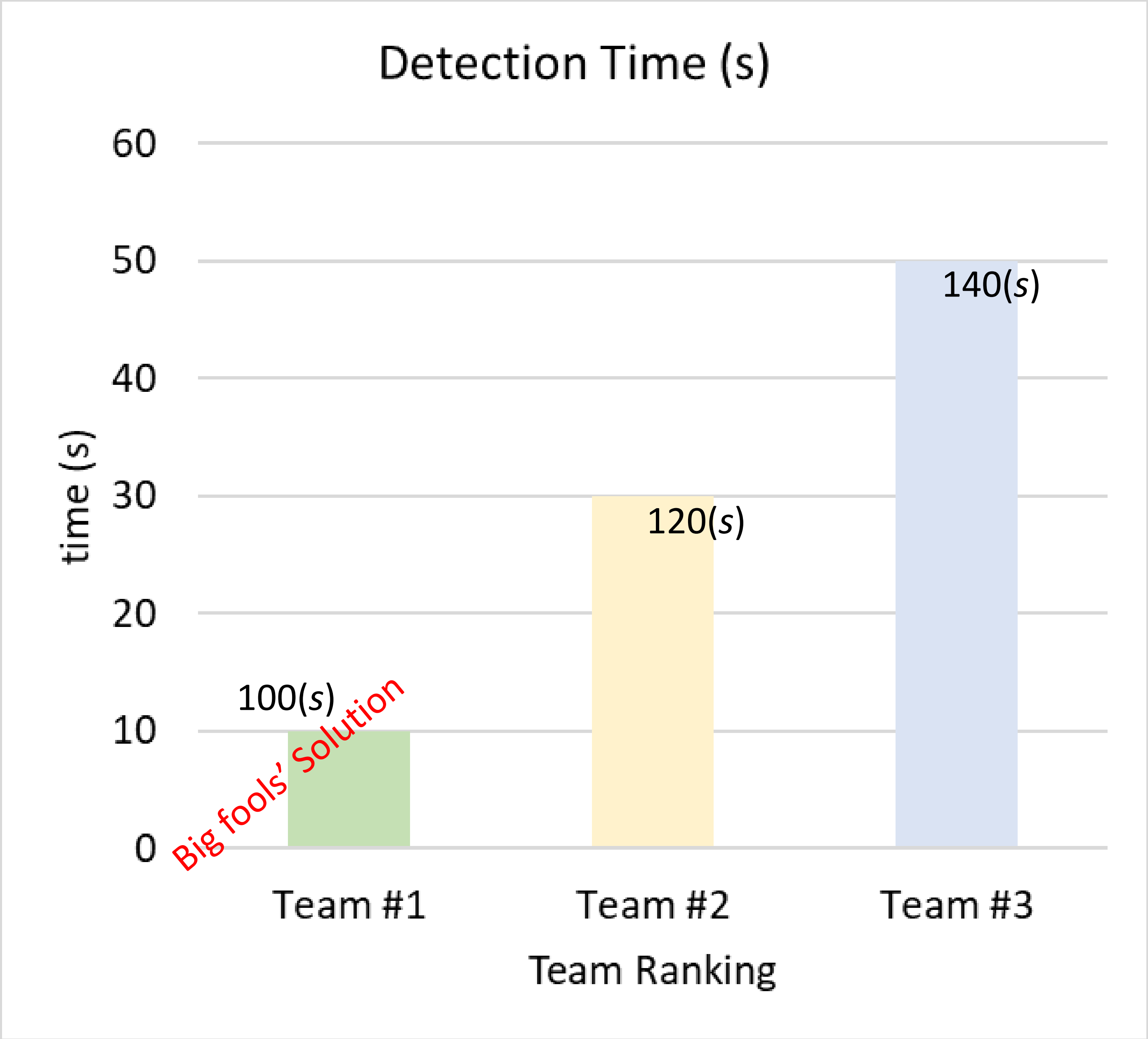




Confusion matrix for task 2 at 10 seconds using 527 features and SVR model

Relationship between number of features and detection time

Performance comparison with other Teams



Conclusion

- ✓ Network failure can be detected with f1-score greater than 0.9
- ✓ Reducing number of features generally increases detection time
- ✓ SVR, XGR and Bayesian Ridge were some of models used.

THANK YOU

The Big Fools (Emmanuel Basikolo)

Supplementary Data
Performance For Different Models

Model Name	Detection Time(seconds)	Number of features	F1-Score
SVR	10	All	0.900609
Bayesian Ridge	10	All	0.858779
Cat Boost Regressor	10	All	0.857143
XGB Regressor	10	All	0.848369
Random Forest Regressor	10	All	0.858238
SVR	20	527	0.908714
Bayesian Ridge	20	527	0.871795
Cat Boost Regressor	20	527	0.852772
XGB Regressor	20	527	0.872211
Random Forest Regressor	20	527	0.872798
SVR	64	1	0.909091
Bayesian Ridge	64	1	0.980392
Cat Boost Regressor	64	1	0.875486
XGB Regressor	64	1	0.880952
Random Forest Regressor	64	1	0.880952

Supplementary Data –Task 1 Results

Time (seconds)	SVR-RMSE
50	0.21602
60	0.191485
70	0
80	0.08165
90	0
100	0
150	0
200	0
250	0
300	0
350	0
400	0.0816
450	0
500	0
550	0.0577
600	0.4714