



# MALWARE DETECTION

Use for IoT Devices

Second Capstone Presentation

Springboard DSCT Bootcamp

Pablo Ruiz Lopez

# PROBLEM IDENTIFICATION

## STATEMENT

IoT has proven to have a significant impact on human life by the integration of devices in a myriad of industries. There will be around 125 IoT devices connected to the internet by 2030.

Developing deployable technology in the form of algorithms, frameworks or even complete SIEM Systems, could be extremely useful to get a better understanding of the behavior of malware infections where IoT devices are the main target.

## STAKEHOLDERS TO PROVIDE KEY INSIGHTS

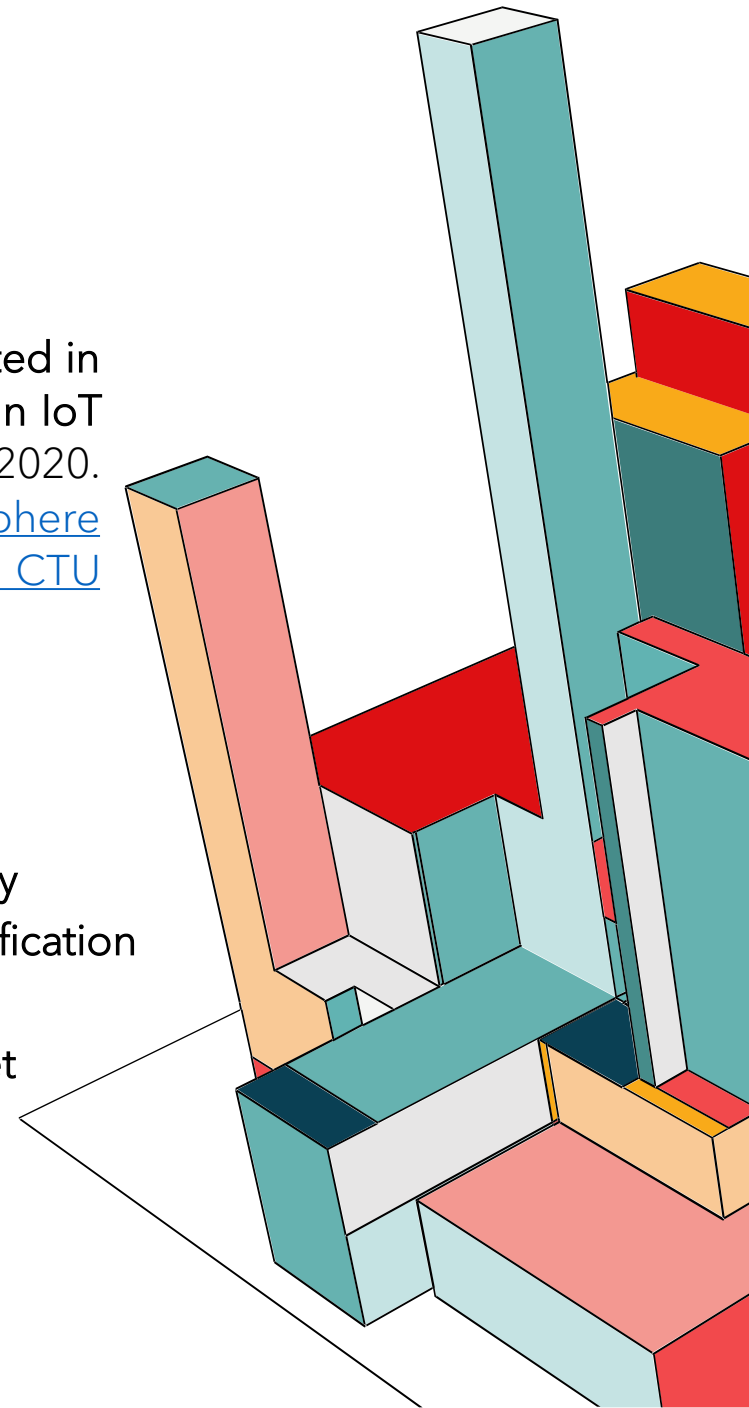
- Cyber Security Teams
- Antivirus Companies
- IoT Companies

## SCOPE OF SOLUTION SPACE

Dataset: 20 malware captures executed in IoT devices, and 3 captures for benign IoT devices traffic. Published in January 2020. These were captured in the [Stratosphere Laboratory, AIC group, FEL, CTU University, Czech Republic](#).

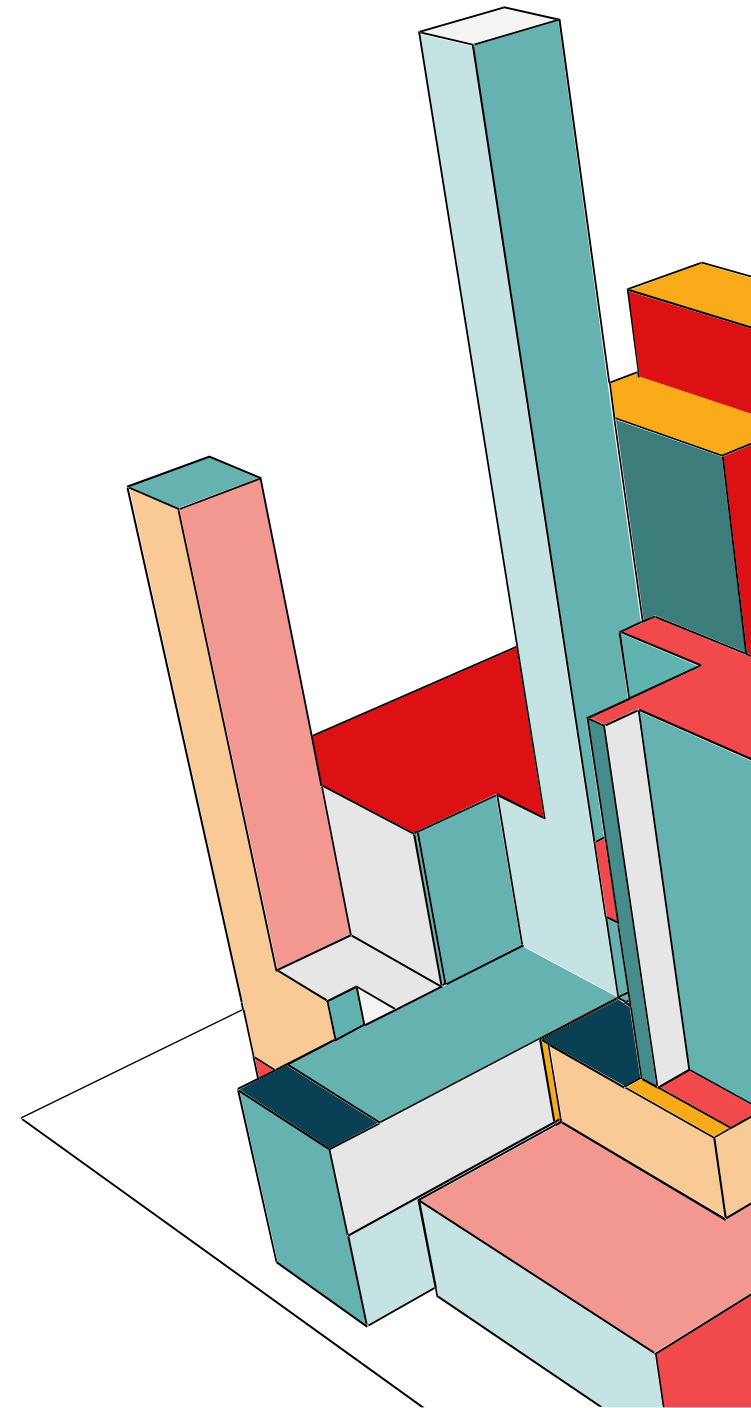
## KEY DATA SOURCES

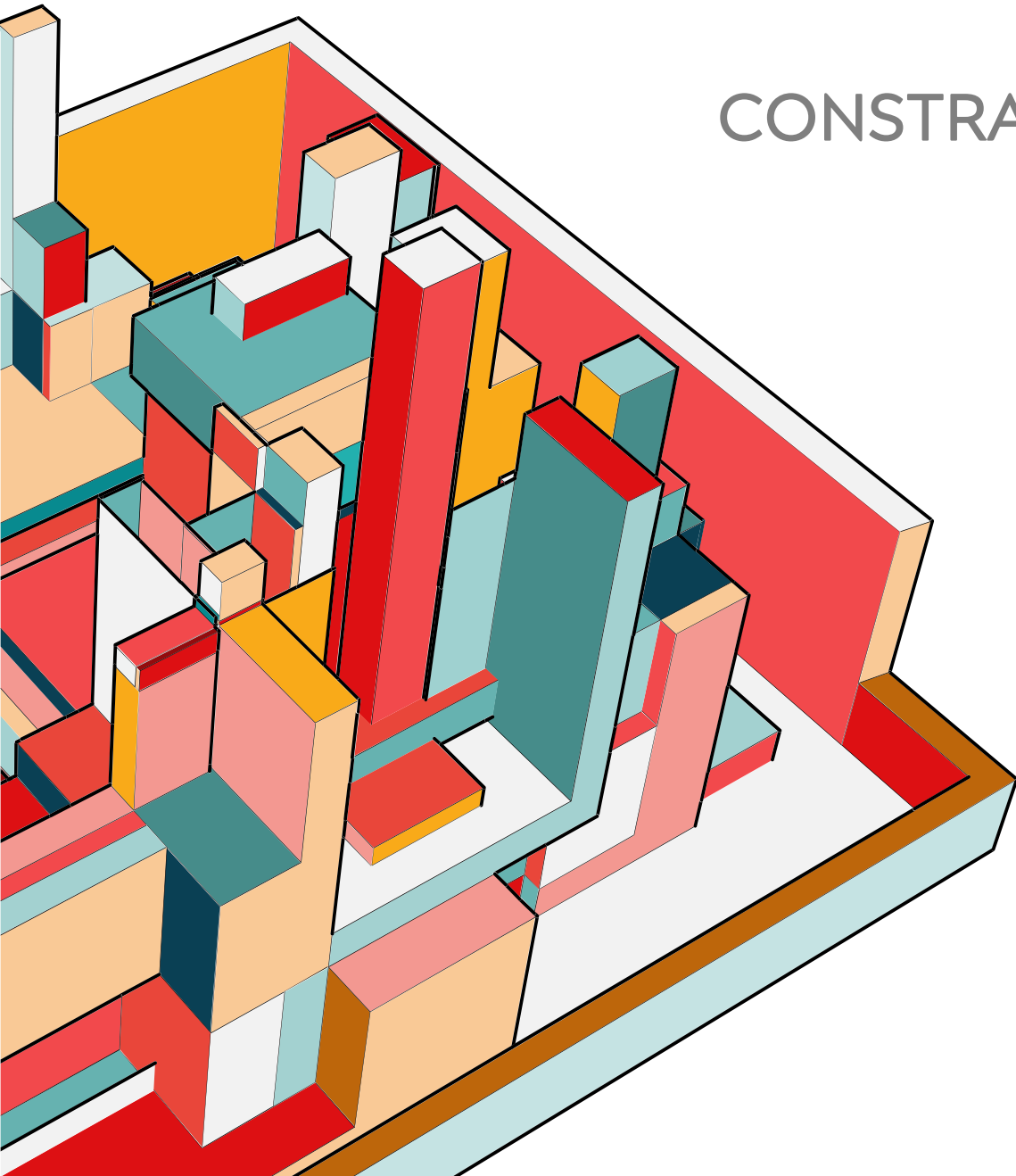
- Log files provided by the laboratory
- Network analyzer used for classification and log file retrieval
- Classification methods spreadsheet



# CRITERIA FOR SUCCESS

- Implement a **Machine Learning algorithm able to detect at least 80%** of malicious network flows
- Implementing a **Malware Type Detector**
- Having a deployable pipeline for **malware detection in real-time** setting





## CONSTRAINS WITHIN SOLUTION SPACE

- Not getting enough **computational power** to analyze all log files.
- Malware information **being outdated** and not representative of how new malware works
- **Class imbalance** for the binary class or multi class project

# KEY FINDINGS AND INSIGHTS

Binary classification project (Class 1-Malicious, Class 0-Benign) with a final dataset  
of:

**380,000 observations**

**67 features**

## Key Features Enough for a One-Feature Model

*Protocol*  
*Origin Port*  
*Port Used for Response*  
*Network History*  
*# of Bytes Sent to Host*  
*# of Response Bytes*  
*# of Response Packets*

## Constraints faced

Severe class imbalance for  
types of malware (8 classes)

Dataset only enough for binary  
classification

RAM power enough for  
analyzing 6 scenarios

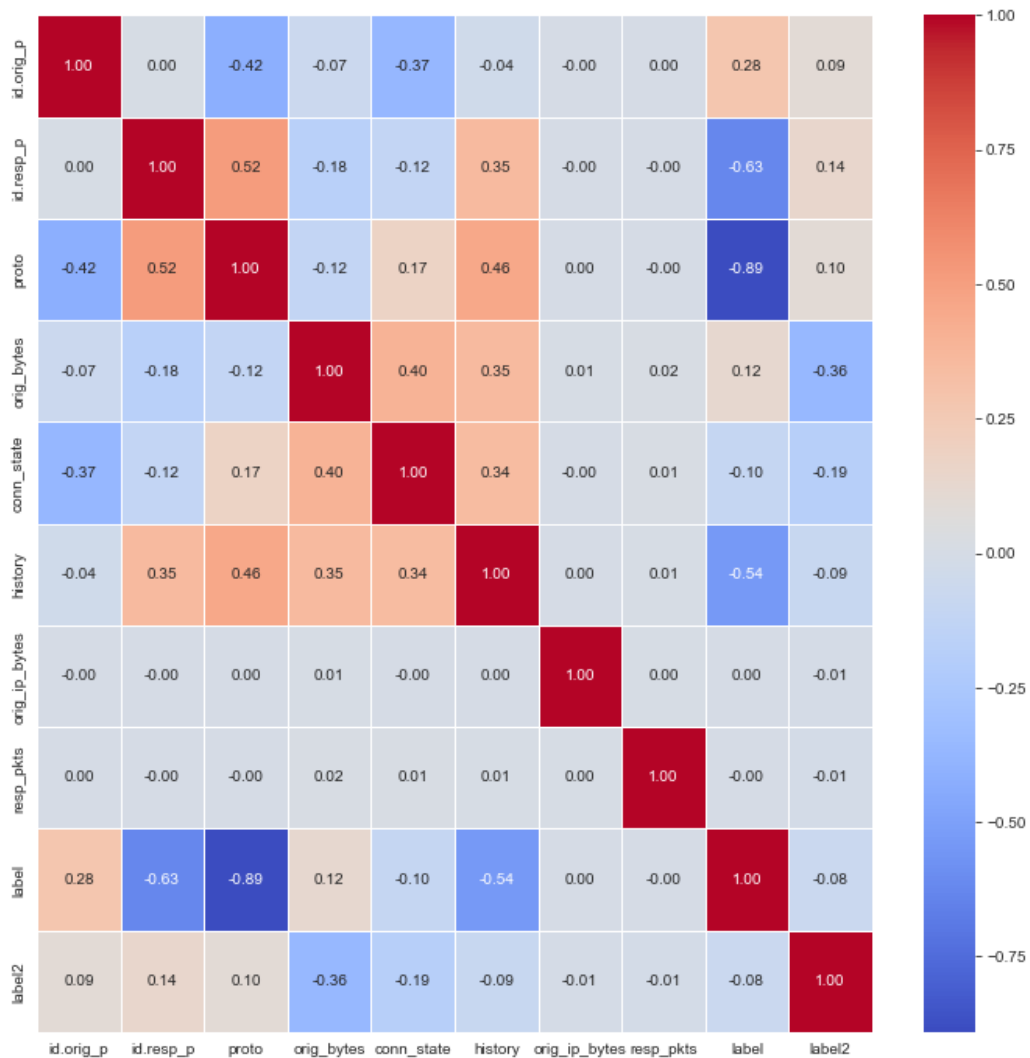
## Insights

Data Distribution between  
classes very marked in  
some features

Time-related Features nor  
dimensionality reduction

Key Features dropped to  
add complexity to the  
modeling

# FEATURE HANDLING



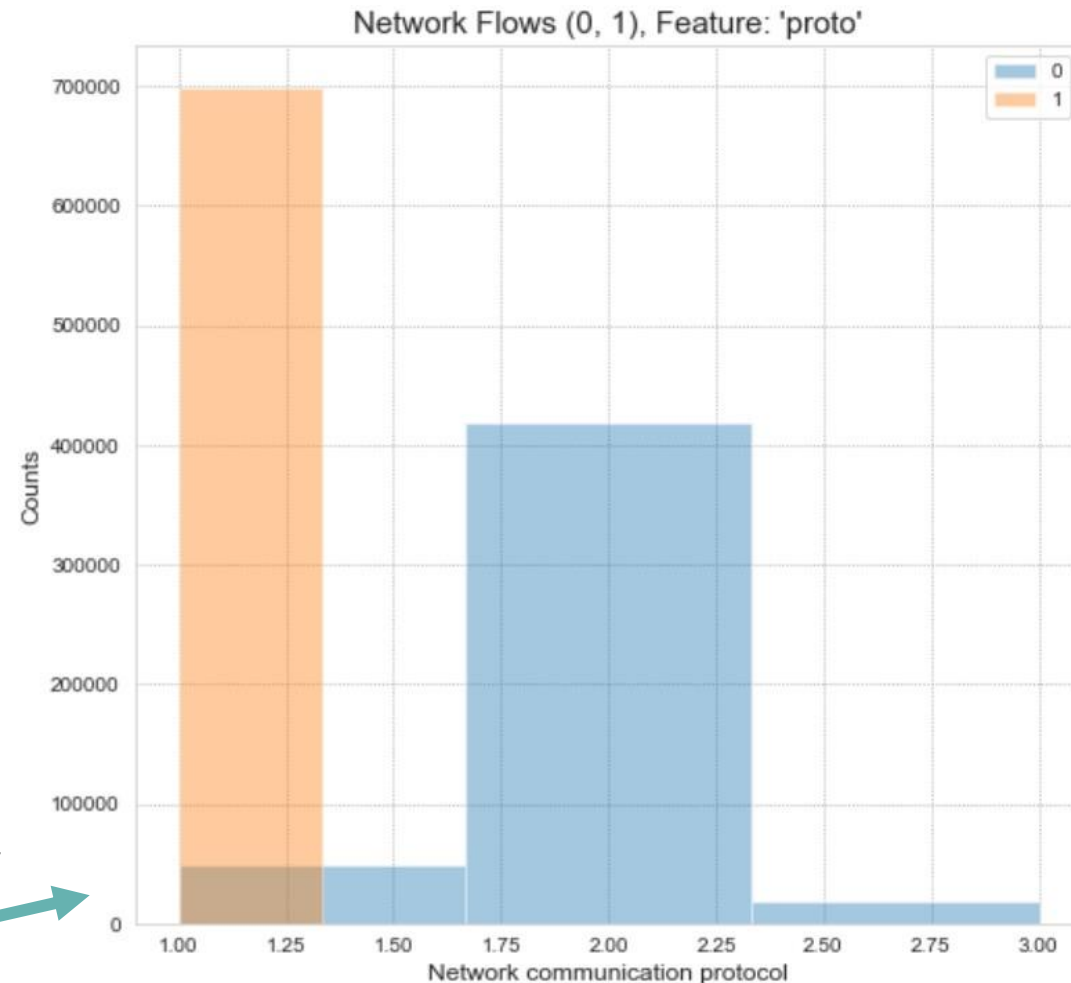
Feature drop was based on:

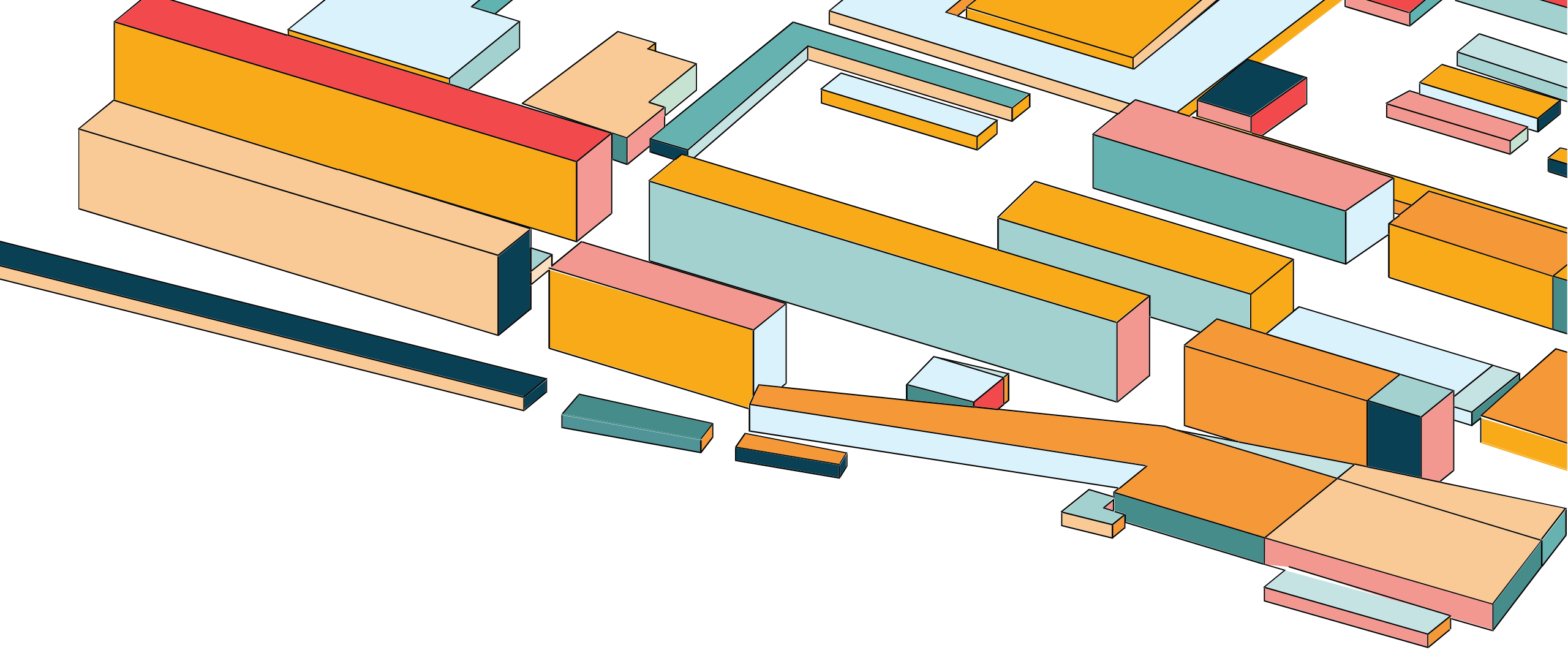
- Collinearity

- Correlation with label 1

- Distribution differences between Classes

(example of Protocol)





# MODEL RESULTS AND ANALYSIS

# “ONE-FEATURE” MODELS

Model	Feature	Precision	Recall
1	proto	0.89	1.0
2	id.orig_p	0.88	0.90
3	id.resp_p	0.90	0.92
4	History	0.90	0.93
5	orig_ip_bytes	0.88	1.0

\* Features eventually dropped

Based on simple assumptions

Features showing high differences  
between classes were used

**Example:**

“All malicious network flows are using TCP  
protocol”



# MODELS AND METRICS USED

METRICS **COMPUTED** PER MODEL

Classification Report  
Precision  
Recall  
Support  
Accuracy  
ROC-AUC

MODEL RESULTS

Model	Precision	Recall
Logistic Regression	0.84	0.90
KNN	0.76	0.87
Decision Tree	0.78	0.80
Random Forest	0.81	0.86
SVM	0.82	0.92
XGBoost	0.81	0.87

METRICS **PRIORITIZED** FOR MODEL SELECTION

Precision (Class 1)  
Recall (Class 1)



# MODEL SELECTION RESULTS

The model selected was:

Support Vector  
Machine

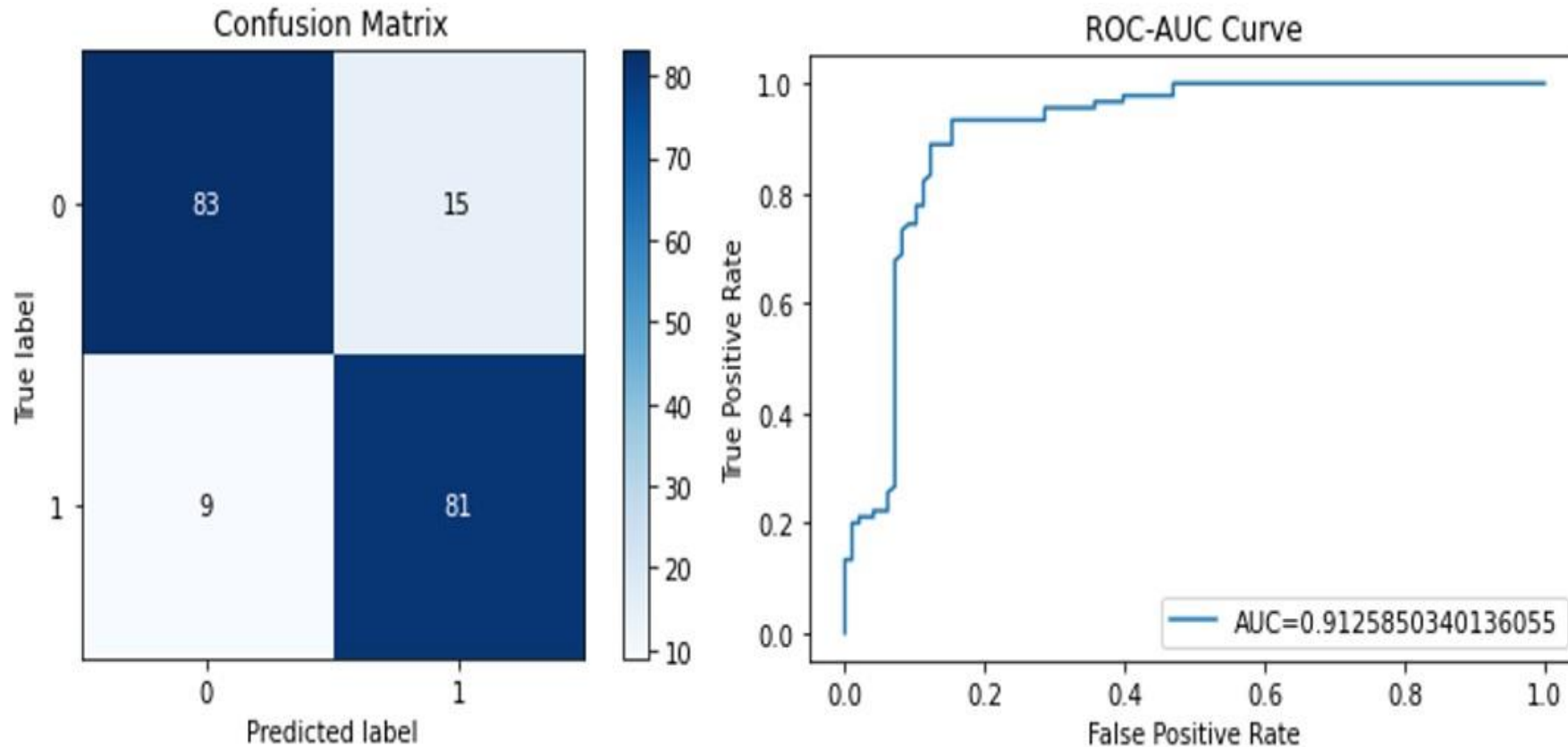
## Considerations:

- Precision and Recall
- Computational time
- Overall performance

## SVM (Tuned) MODEL RESULTS

Precision:	0.84
Recall	0.90
F1 Score	0.87
Support	0.90
Accuracy	0.87
Best AUC Score	0.87

# SVM RESULTS AND PARAMETERS



Parameters tuned and selected  
(GridSearchCV)

- Scaler
- C
- coef0
- decision\_function\_shape
- degree
- gamma
- kernel
- probability

# SUMMARY & FUTURE WORK

## KEY TAKEAWAYS

1. Paramount importance of EDA
2. Features containing HUGE amount of variance
3. Types of Malware used similar "MO's" (e.g., TCP in all of them)

## FUTURE WORK

1. Feature analysis bypassing the network analyzer used
2. Train the model in a Cloud Environment to analyze all available types of bots
3. Addressing issues for a multi-class problem
4. Include and deploy the model in a streaming pipeline for real-time detection

# THANK YOU

Pablo Ruiz Lopez

+52 55-2729-6472

pablowlweb8@gmail.com

<https://github.com/pablo-git8>

