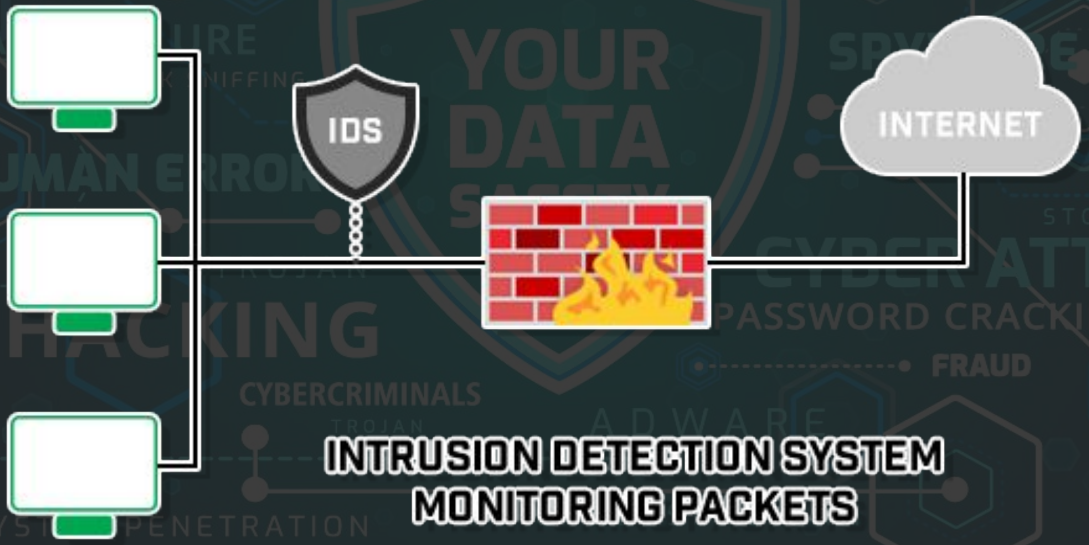
Network Intrusion Detection System

on real time data with Machine Learning

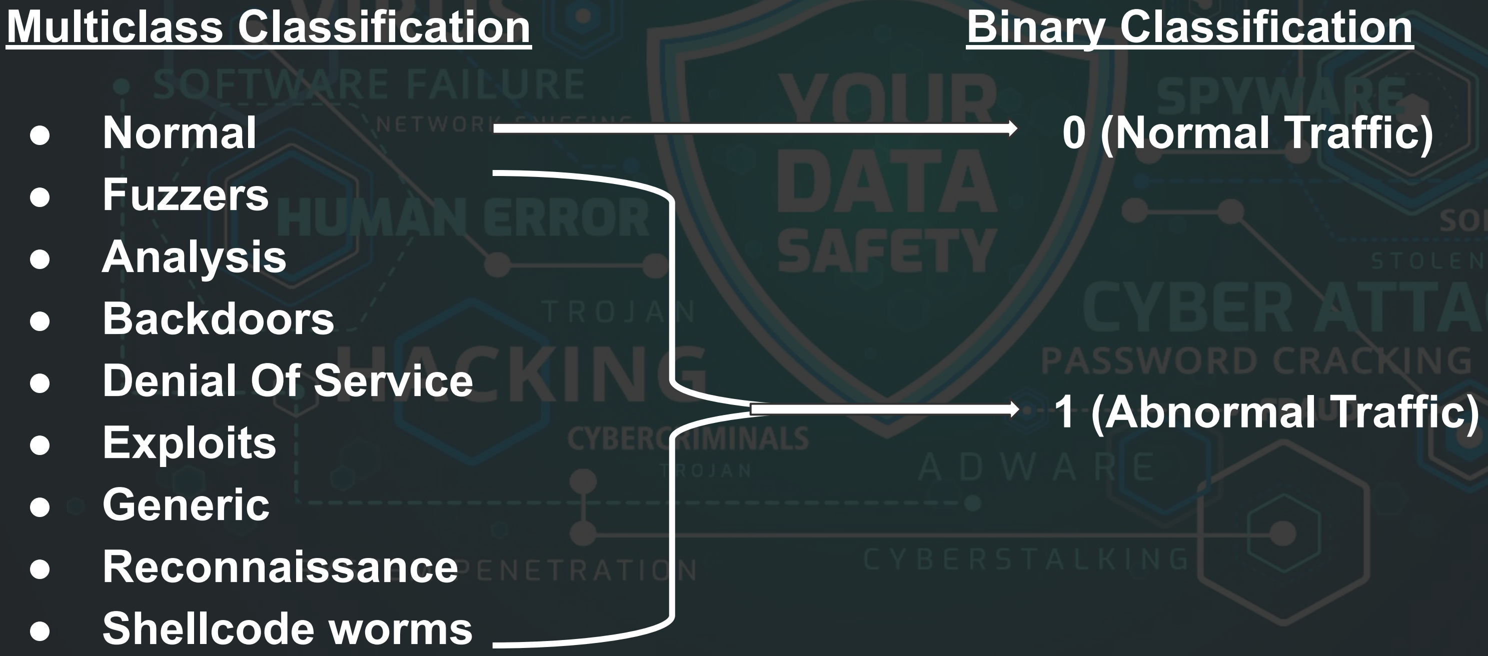
But what is it really a Intrusion Detection System?

An intrusion detection system (IDS) is a device or software application that monitors a network or systems for malicious activity.



Dataset - Problem Goals

The dataset of the problem was the UNSW-NB15 Dataset .



Multiclass Classification

- **Normal**
- **Fuzzers**
- **Analysis**
- **Backdoors**
- **Denial Of Service**
- **Exploits**
- **Generic**
- **Reconnaissance**
- **Shellcode worms**

Binary Classification

0 (Normal Traffic)

1 (Abnormal Traffic)

The basic idea - Implementation

- Create a machine learning model that can make decisions and validate if the current traffic is related to an attack or a normal flow.
- We conducted our own research on papers regarding the machine learning algorithms that are related and finally we used the below:

Tests	Normal/Abnormal Classification	Attack Type Classification
Multilayer Perceptron	83.96%	82.81%
Random Forest	81.33%	68.08%
Support Vector Machines	76.10%	68.08%

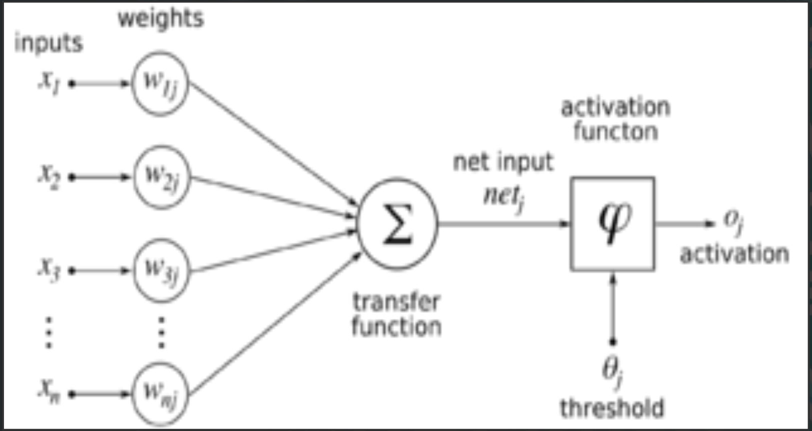
Which model should we choose ?

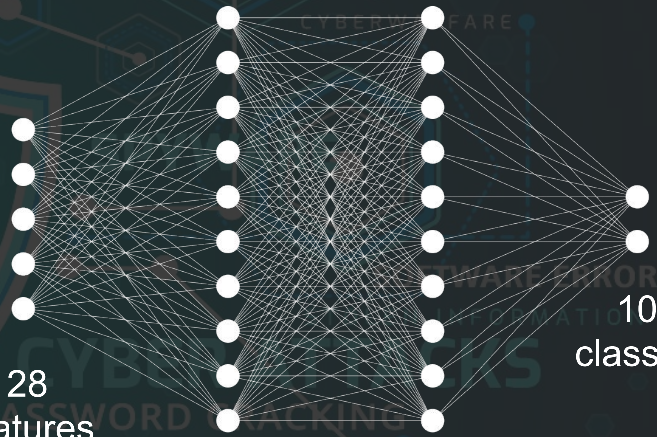
It is obvious that MLP produces the better results of the three.We can use the current model for evaluation at the data feeding phase.

MLP
RF
SVM
MLP

Multilayer Perceptron - Best solution

Perceptron





28
features

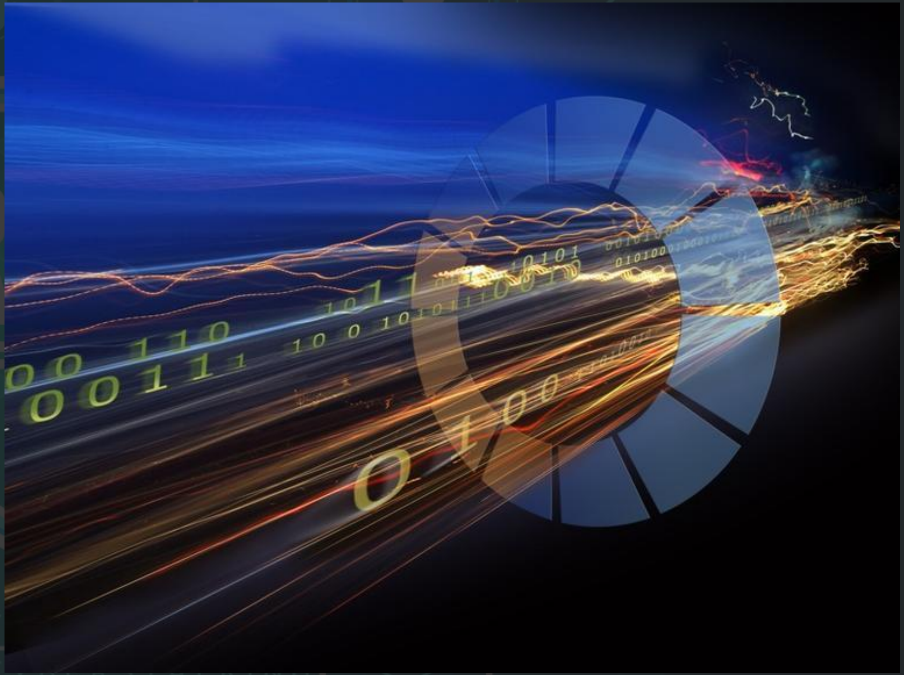
300
layers

150
layers

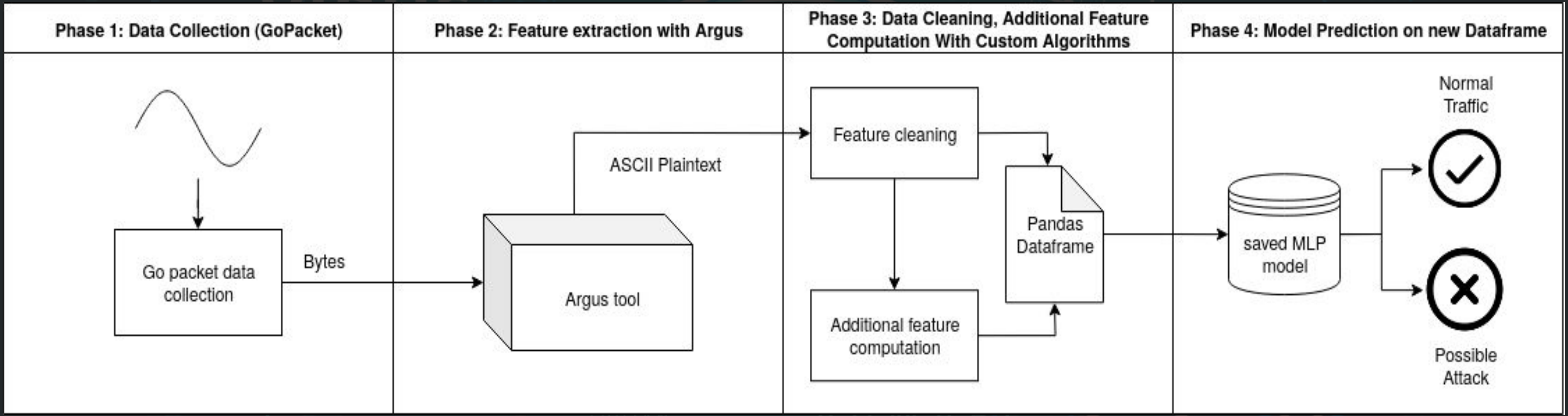
10
classes

Extension: Real time data streaming

Nowadays the data that each network device handles grows exponentially so we must adjust our implementation to real time circumstances.



Data Stream Pipeline - Overview



Phase 1: Data Collection (GoPacket)

Go packet data collection

Bytes

Phase 2: Feature extraction with Argus

Argus tool

ASCII Plaintext

Phase 3: Data Cleaning, Additional Feature Computation With Custom Algorithms

Feature cleaning

Additional feature computation

Pandas Dataframe

Phase 4: Model Prediction on new Dataframe

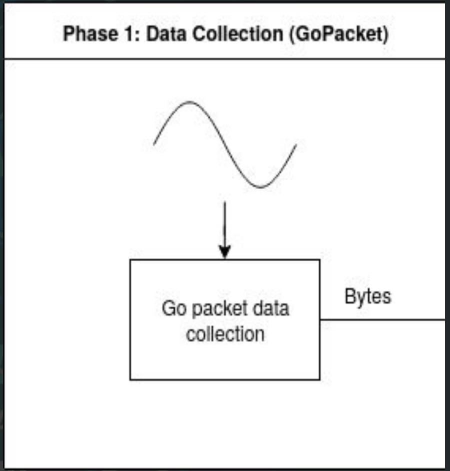
saved MLP model

Normal Traffic

Possible Attack

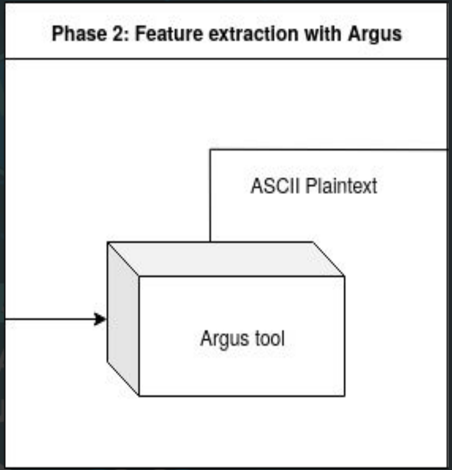
Packet Collection - Steps

- Choose a network interface you want to use.
- Bind to that interface and start listening the traffic.
- Collect each packet.
- Pass the packet byte stream to the next phase of the pipeline.


Phase 1: Data Collection (GoPacket)
Go packet data collection
Bytes

Argus Tool

- Retrieve the byte stream from the previous step.
- Handle the data properly.
- Extract some reports using Argus tool.
- Use ra to read the reports and extract the features we want for our model.
- Pass the above output to the next phase for cleaning and further processing.

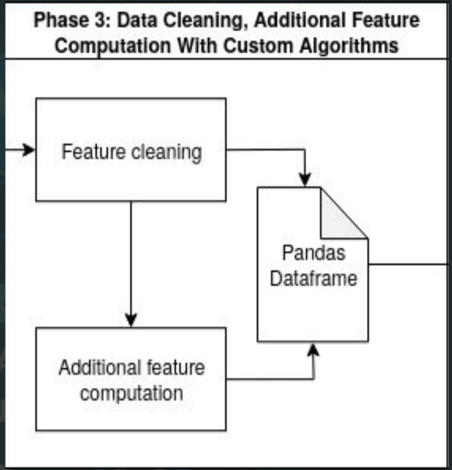


Argus - Features

dur	proto	service	state
spkts	dpkts	rate	sttl
dttl	sload	dload	sinpkt
dinpkt	sjit	djit	swin
stcpb	dtcpb	tcprtt	smeanz
dmeanz	trans_depth	res_body_len	ct_srv_src
ct_state_ttl	ct_dst_ltm	is_ftp_login	ct_flw_http_m

Cleaning - feature extraction

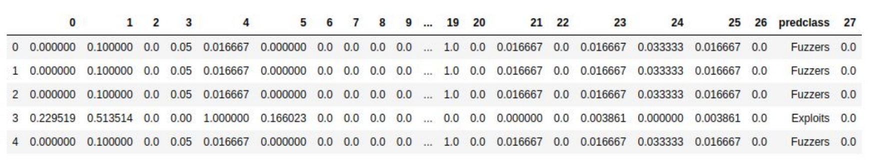
- Retrieve the byte stream from the previous step.
- Generate the “extra” features that are based on some algorithms.
- Generate from the above the dataframe that we will forward to the model.



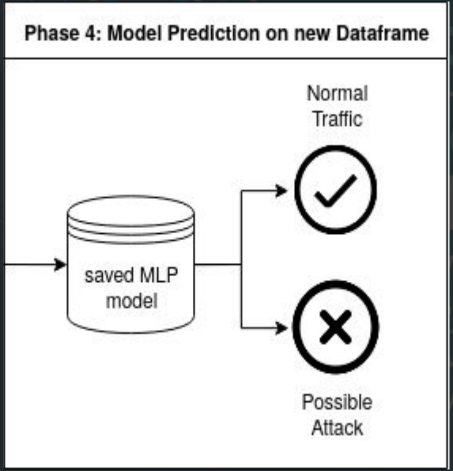
Phase 3: Data Cleaning, Additional Feature Computation With Custom Algorithms

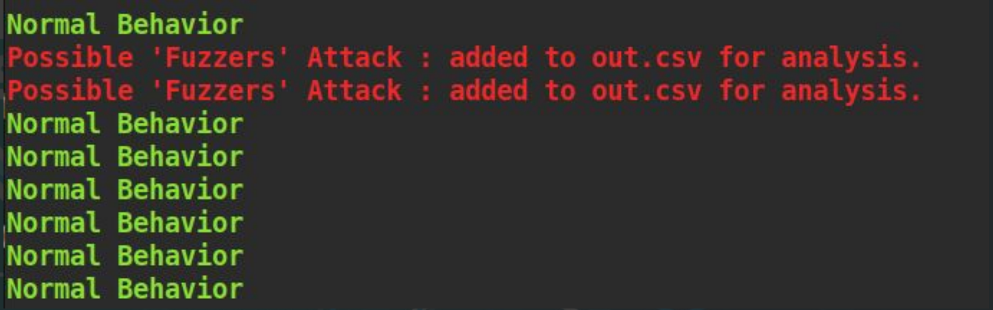
Model Prediction

We feed the generated dataframe to the model and after some processing it produces the result.



	0	1	2	3	4	5	6	7	8	9	...	19	20	21	22	23	24	25	26	predclass	27
0	0.000000	0.100000	0.0	0.05	0.016667	0.000000	0.0	0.0	0.0	0.0	...	1.0	0.0	0.016667	0.0	0.016667	0.033333	0.016667	0.0	Fuzzers	0.0
1	0.000000	0.100000	0.0	0.05	0.016667	0.000000	0.0	0.0	0.0	0.0	...	1.0	0.0	0.016667	0.0	0.016667	0.033333	0.016667	0.0	Fuzzers	0.0
2	0.000000	0.100000	0.0	0.05	0.016667	0.000000	0.0	0.0	0.0	0.0	...	1.0	0.0	0.016667	0.0	0.016667	0.033333	0.016667	0.0	Fuzzers	0.0
3	0.229519	0.513514	0.0	0.00	1.000000	0.166023	0.0	0.0	0.0	0.0	...	0.0	0.0	0.000000	0.0	0.003861	0.000000	0.003861	0.0	Exploits	0.0
4	0.000000	0.100000	0.0	0.05	0.016667	0.000000	0.0	0.0	0.0	0.0	...	1.0	0.0	0.016667	0.0	0.016667	0.033333	0.016667	0.0	Fuzzers	0.0





```
Normal Behavior
Possible 'Fuzzers' Attack : added to out.csv for analysis.
Possible 'Fuzzers' Attack : added to out.csv for analysis.
Normal Behavior
Normal Behavior
Normal Behavior
Normal Behavior
Normal Behavior
Normal Behavior
```

Thank you for your attention!