Evaluating Data Manipulation Threats in WiFi-based People Counting: Risks and Defenses

- **Anthony B.(UW)**
- **Ishraq H. C.(Uniba)**
- **Ibrahim K.(Uniba)**

OFFENSIVE

OFFENSIVE

"Privacy and Security Go Hand in Hand"

OFFENSIVE - REASONS

Simulating Potential Attacks

To see how devices behave

Risk Assessment

Safety Measures

OFFENSIVE - REASONS

Simulating Potential Attacks

To see how devices behave

Risk Assessment

Safety Measures

GDPR
(Compliance and Regulations)

OFFENSIVE – SIMULATING POTENTIAL ATTACKS

To identify vulnerabilities and weaknesses

OFFENSIVE – TO SEE HOW DEVICES BEHAVE

In Real World

OFFENSIVE – RISK ASSESSMENT

Getting information to make informed decision

Mitigating Risks

OFFENSIVE – SAFETY MEASURES

What are and What could?

Improvement of safety measures

OFFENSIVE – Compliance and Regulations

Ensures Compliance

Helps in avoiding potential legal and regulatory issues, (z. B. Data breach, Location Breach)

OFFENSIVE – Compliance and Regulations

Art. 83(4)

"Less severe infringements can result in a fine of €10 million or 2% of a firm's annual revenue from the preceding financial year"

https://gdpr-info.eu/issues/fines-penalties/

https://www.eqs.com/compliance-blog/biggest-gdpr-fines/#:~:text=Less%20severe%20infringements%20can%20result,depending%20on%20what%20is%20higher.

OFFENSIVE – ZERO DAY

OFFENSIVE

WIRED SUBSCRIBE

BRIAN BARRETT SECURITY FEB 3, 2020 1:44 PM

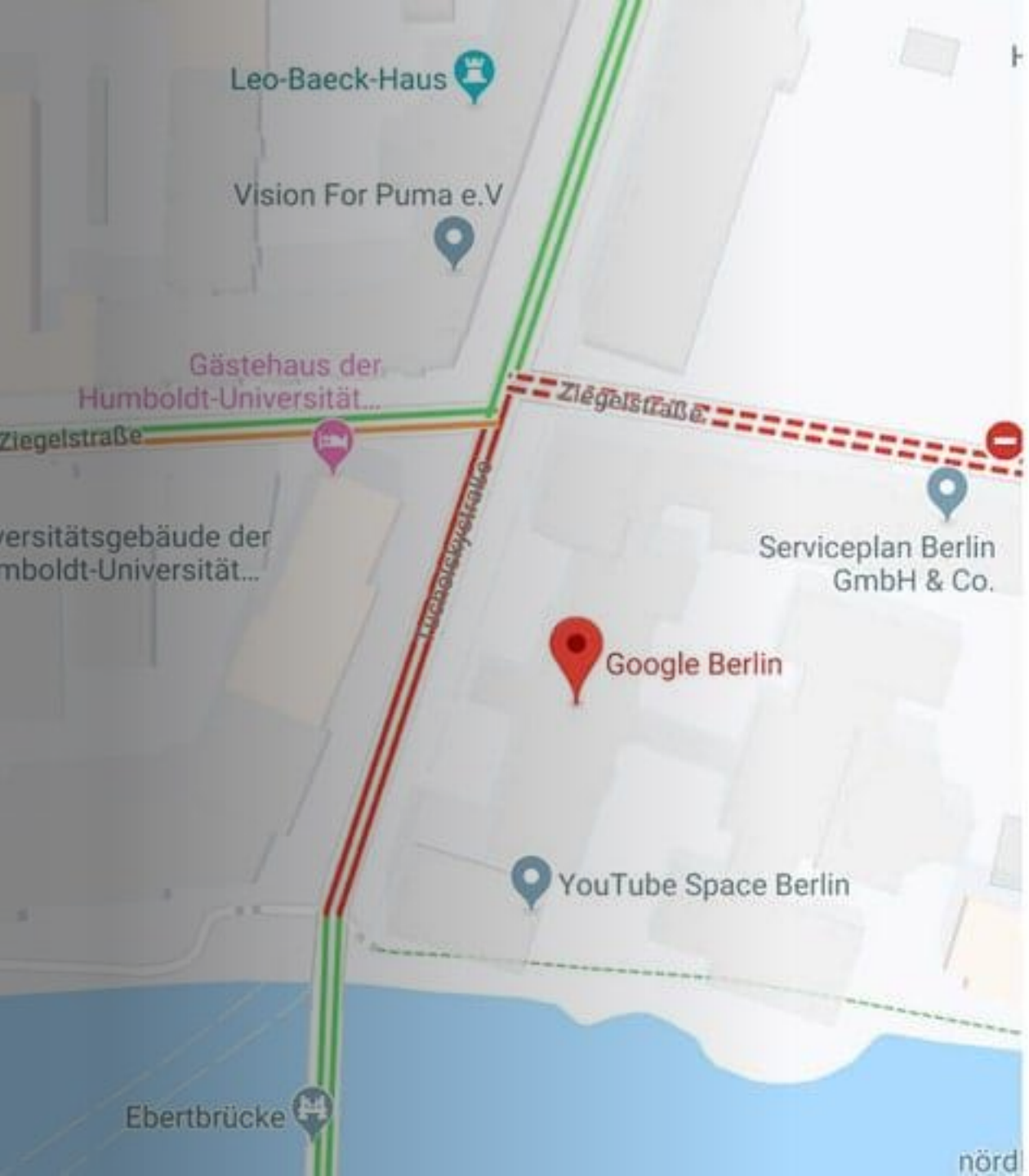
An Artist Used 99 Phones to Fake a Google Maps Traffic Jam

With his "Google Maps Hack," artist Simon Weckert draws attention to the systems we take for granted—and how we let them shape us.

PHOTOGRAPH: SIMON WECKERT

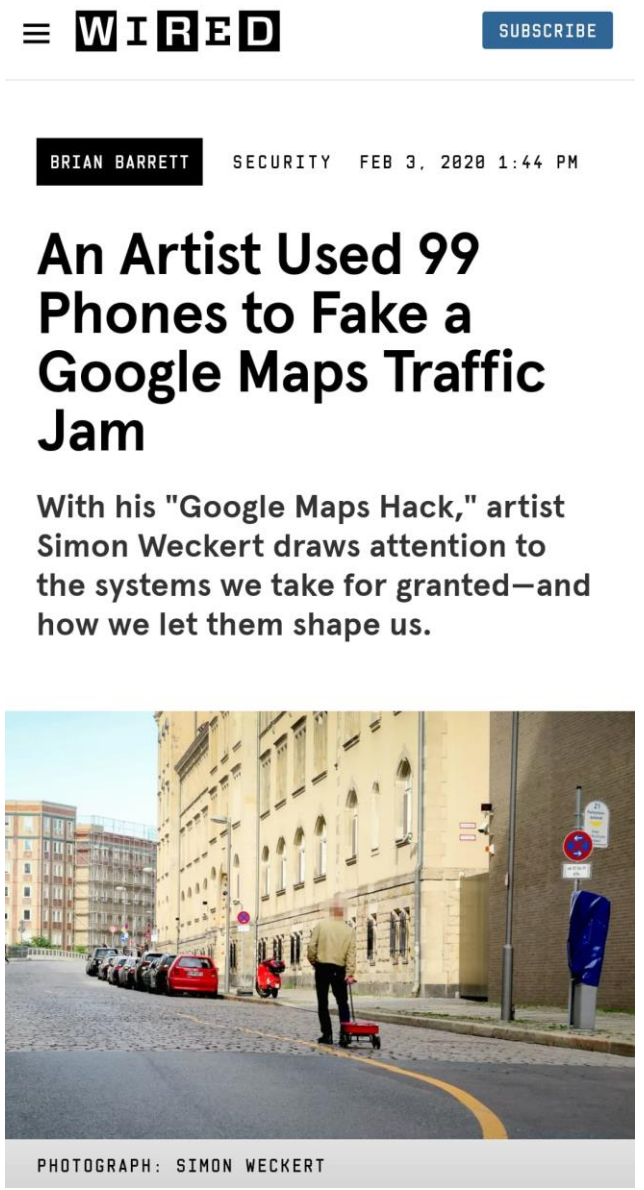
FIG. WIRED ARTICLE

https://www.wired.com/story/99-phones-fake-google-maps-traffic-jam/


Leo-Baeck-Haus
Vision For Puma e.V
Gästehaus der
Humboldt-Universität...
Ziegelstraße
Ziegelstraße
versitätsgebäude der
mboldt-Universität...
Serviceplan Berlin
GmbH & Co.
Google Berlin
YouTube Space Berlin
Ebertbrücke
nördl

OFFENSIVE

Weckert says. “I don’t need the people. I just need their smartphones.”



WIRED

SUBSCRIBE

BRIAN BARRETT SECURITY FEB 3, 2020 1:44 PM

An Artist Used 99 Phones to Fake a Google Maps Traffic Jam

With his "Google Maps Hack," artist Simon Weckert draws attention to the systems we take for granted—and how we let them shape us.

PHOTOGRAPH: SIMON WECKERT

FIG. WIRED ARTICLE

https://www.wired.com/story/99-phones-fake-google-maps-traffic-jam/

OFFENSIVE

Weckert says. "I don't need the people. I just need their smartphones."

"We don't need the people. We just need to inject packets."

WIRED SUBSCRIBE

BRIAN BARRETT SECURITY FEB 3, 2020 1:44 PM

An Artist Used 99 Phones to Fake a Google Maps Traffic Jam

With his "Google Maps Hack," artist Simon Weckert draws attention to the systems we take for granted—and how we let them shape us.

PHOTOGRAPH: SIMON WECKERT

FIG. WIRED ARTICLE

https://www.wired.com/story/99-phones-fake-google-maps-traffic-jam/

OFFENSIVE – ATTACK MODELS

MAC Spoofing

Probe Request Flooding

Relay Attacks/Replay Attacks

DoS (Future Aspect)

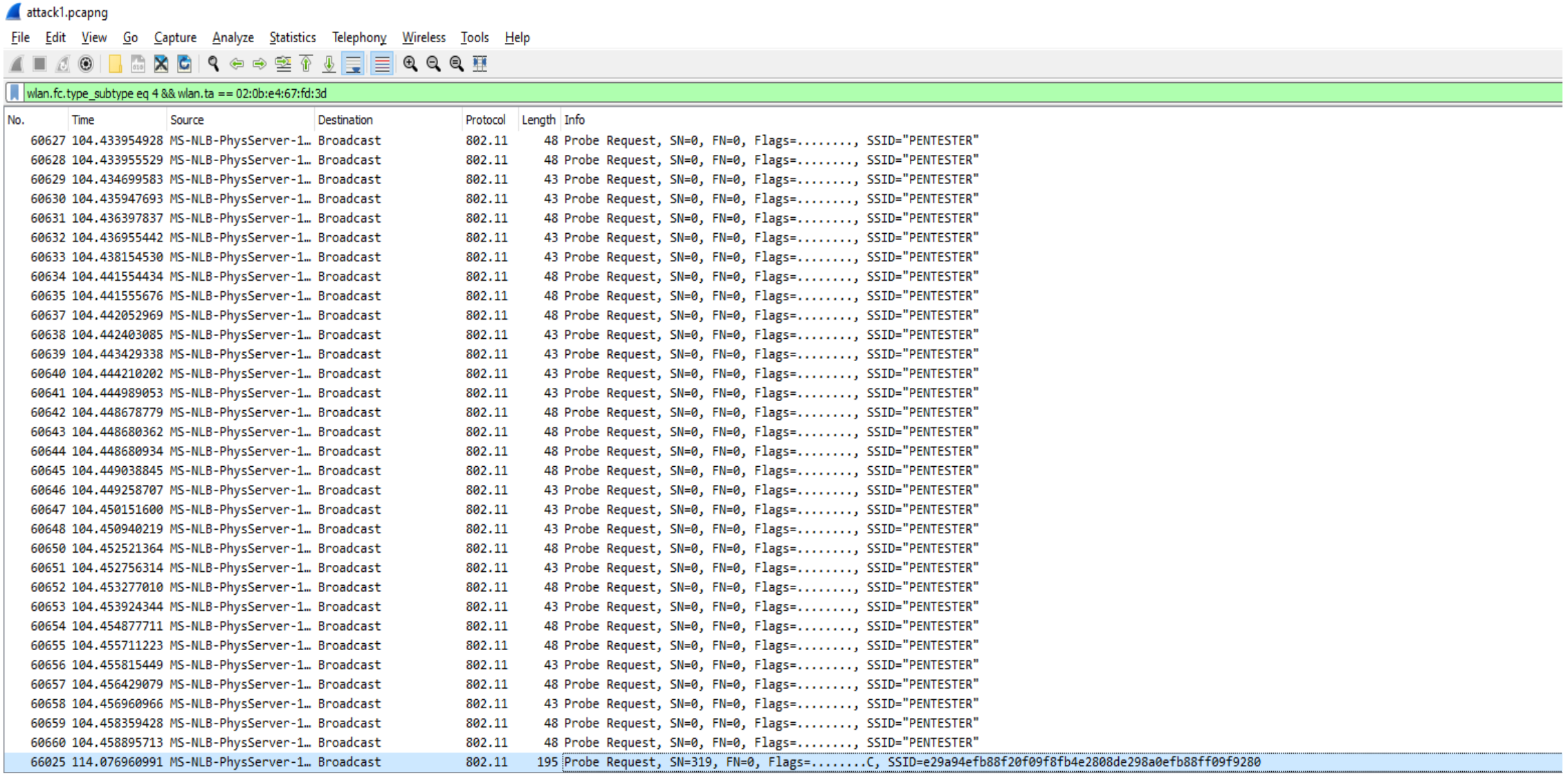


FIG. Probe Burst

	mac_address name	eventtype name	epocutc timestamp without time zone	zone name	rssi integer	techtype integer	salt name
243	148bebd99f4d19bb77956f4d7d84e8917690c3a5d428c5aeb82069...	status	2023-07-27 14:02:05	bz2454	-41	2	
244	accddb183a16ee49a4541b5309ab3d6d6b9034e3920b9febdff4efce	status	2023-07-27 14:02:06	bz2454	-41	2	
245	66ebc9bfb6ce95cf3cca782fd856957a6614c0b25aa0858e9aea97ed	status	2023-07-27 14:02:09	bz2454	-41	2	
246	1d033e8c163c8e1aafa8cd3666e4775890cfecb4f2fc07bdd7233597	status	2023-07-27 14:02:14	bz2454	-41	2	
247	1e8c923adb53bc4f8bf72b5eaaa761860f7572c232e8ff070939c0f5	status	2023-07-27 14:02:14	bz2454	-41	2	
248	44b0494a74e3391544fc6ca50813e95ba9a15a53d1b15761c378b6...	status	2023-07-27 14:02:14	bz2454	-45	2	
249	9783db88efab15a7b6a12847b02373a5101932ab3b4801a973c53...	status	2023-07-27 14:02:15	bz2454	-41	2	
250	a50d7b391653d164304514a416fcd697b79ba8a670f163f77e8247...	status	2023-07-27 14:02:17	bz2454	-41	2	
251	e15841622ed87850a3ac8ba636555b74dbd5331e9c777314c1abc...	status	2023-07-27 14:02:19	bz2454	-41	2	
252	845d51fdffcc2c518b4bbbe45402a35e975ccf3c0680f3784ce3d0d6	status	2023-07-27 14:02:21	bz2454	-43	2	
253	a994cbccb885a4c6c09d8154e7be7a34f879b56e8c27831f8bd87a...	status	2023-07-27 14:02:23	bz2454	-41	2	
254	a994cbccb885a4c6c09d8154e7be7a34f879b56e8c27831f8bd87a...	status	2023-07-27 14:03:20	bz2454	-41	2	
255	a994cbccb885a4c6c09d8154e7be7a34f879b56e8c27831f8bd87a...	status	2023-07-27 14:06:42	bz2454	-41	2	

FIG. Injected Data in the Database

OFFENSIVE – FUTURE APPROACH

Attacking all sensor's at once

Passing them through defense framework

Visualizing the data of *"attack event's"*

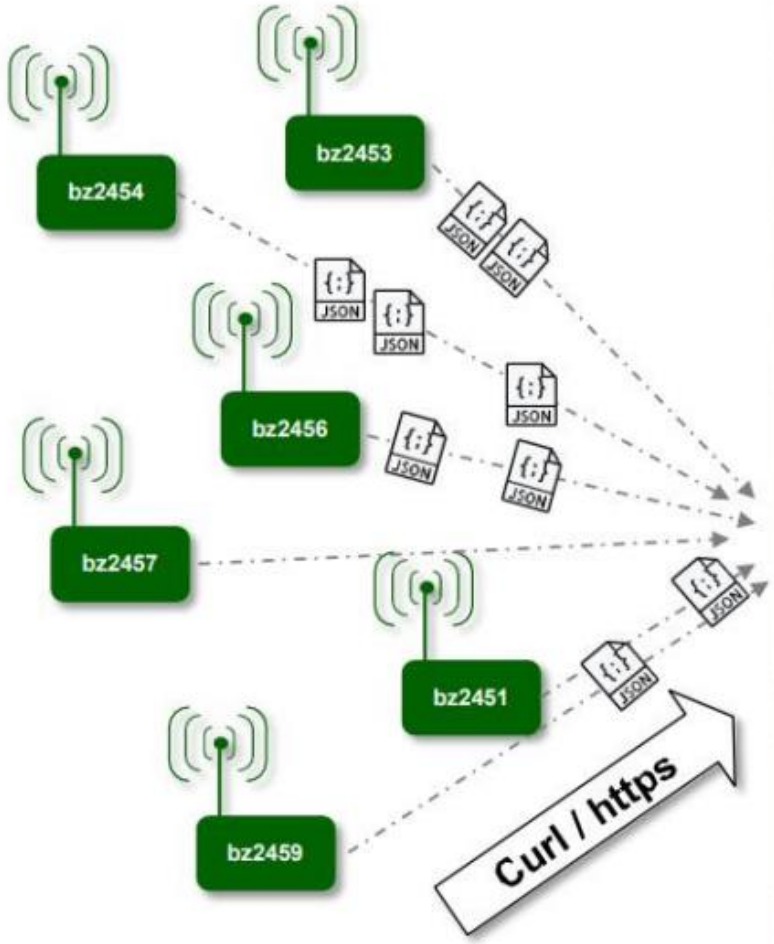
Future Research to mitigate these attacks

(*effective response)*

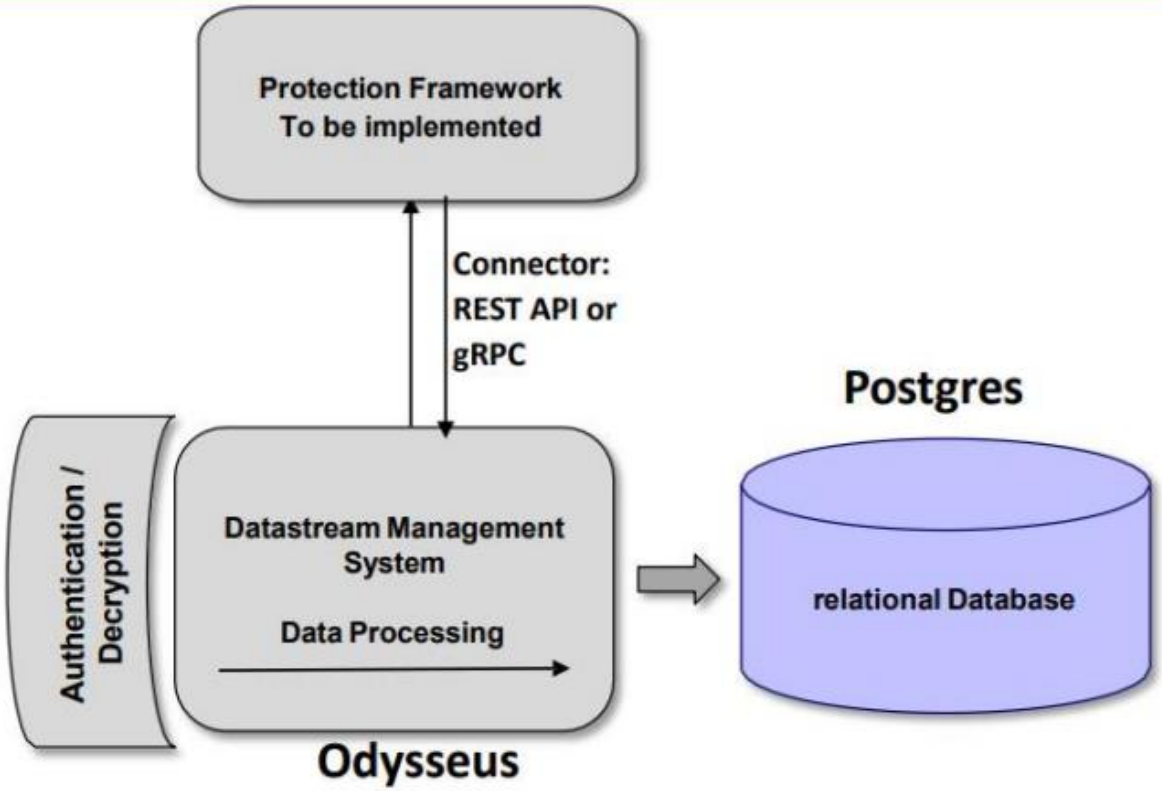
Current Work For Defense Mechanism:

- Generation of dummy dataset
- Creation of a protection component/framework with protection mechanism
- Protection mechanisms:
  - 1. Input validation
  - 2. Data sanitization
- These two functions could also be implemented in Osysseus.
- Anomaly protection mechanism against poisoning with unsupervised learning – Isolation Forest.
  - Isolation Forest is an unsupervised machine learning model for anomaly detection, the model is predicting every anomaly that we have tested so far, with an accuracy of 1.0.
- DoS protection mechanism with online-learning or batch learning.

Data processing



Bamberg City



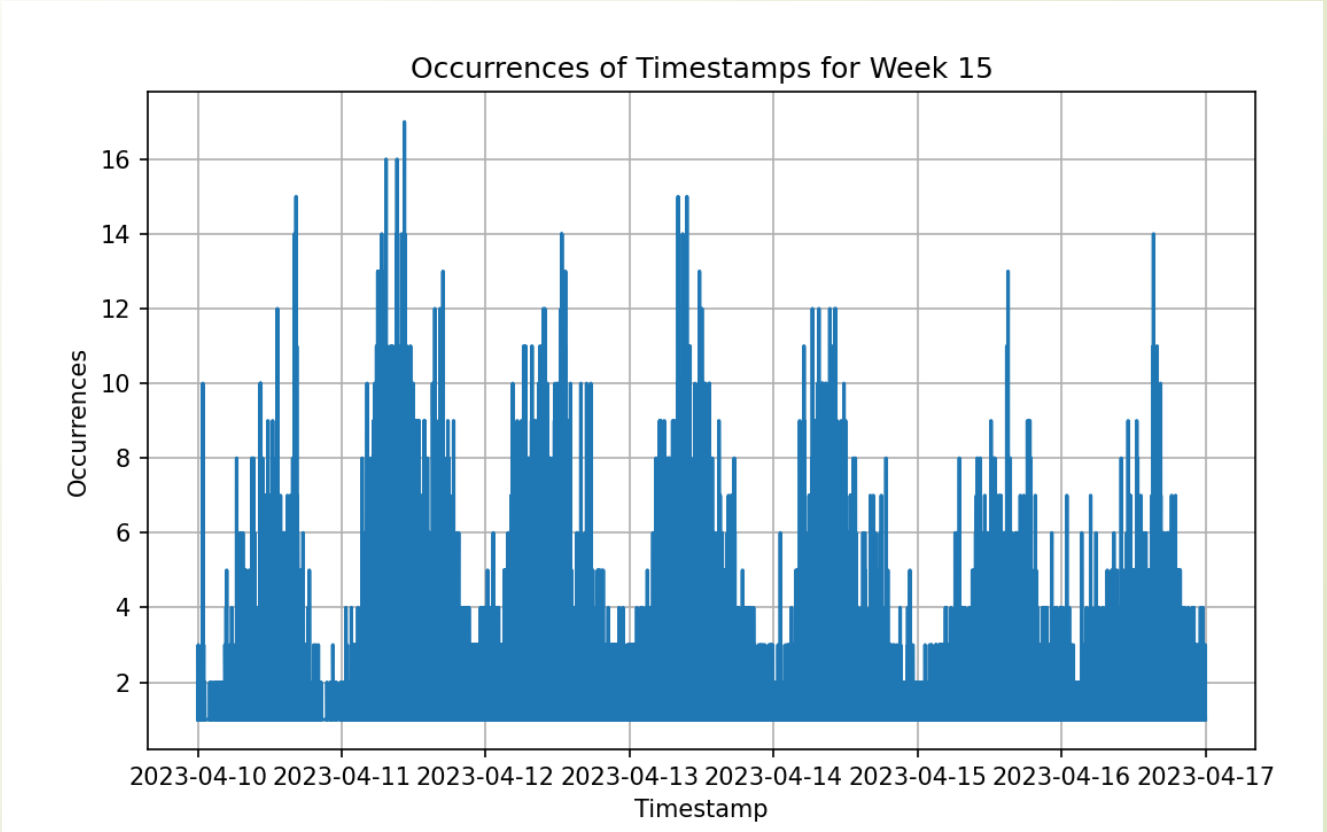
Uni Bamberg, Mobi Chair Datacenter

Input Validation

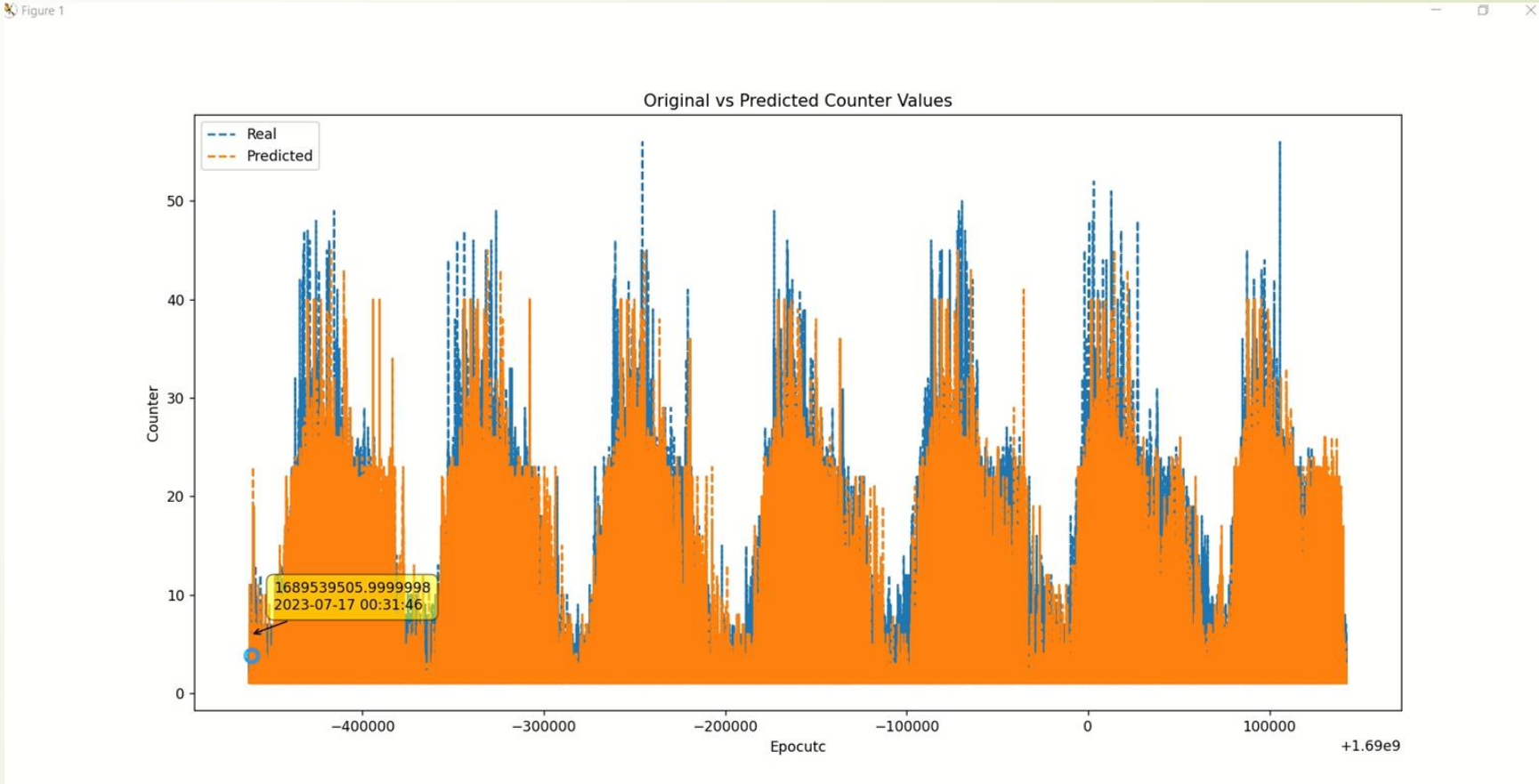
- First step in the process
- Ensure all incoming data is in proper format
  - eventtype: status/leave
  - epocutc: DD/MM/YYYY 00:00:00
  - zone: bz….
- Log invalid inputs
- This is a function we are coding, but it could be implemented also in Odysseus
- Besides, we have an anomaly detection algorithm that can distinguish these patterns without through condition statements.

4. DoS protection mechanism with online-learning:

Currently we are still developing this part of the algorithm. We are trying to predict the flow patterns in time. Our model would have the possibility to know the number of probe requests coming from the sensors, the number of diferente MACs per frame of time(5 mins) to expect,and other statistical predictions, and produce alerts/take actions accordingly. Besides, the model would be able to learn new patterns in the future if changes in the flow of traffic happen.



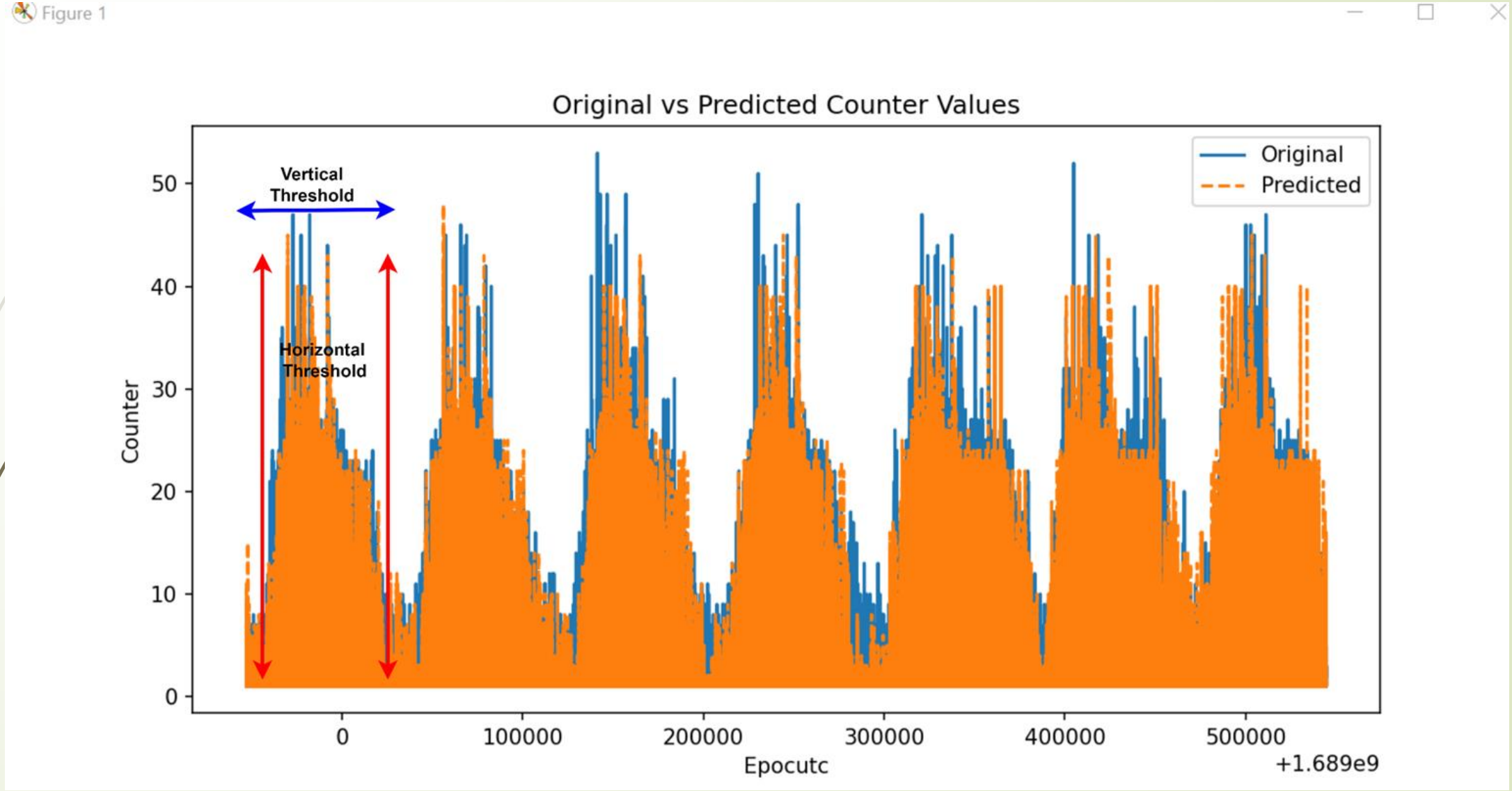
Prediction results for 2 weeks showing good alignment between real and predicted traffic.



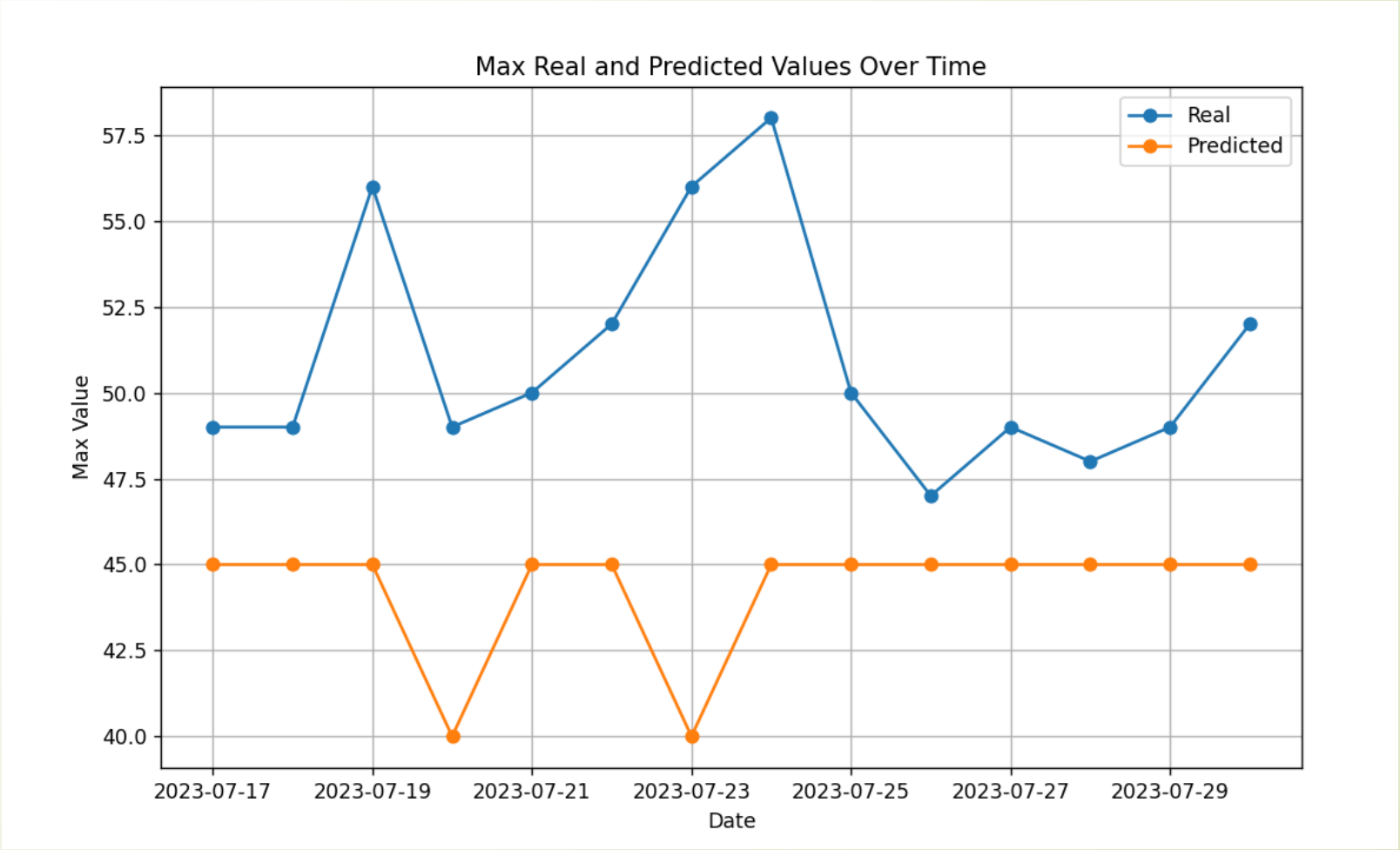
Techniques that we are using:

Techniques	Anomaly detected?
Vertical rate limit(count of probe request coming in an interval of 5 min), applied to all the traffic coming from all the sensors.	1
Horizontal rate limit(pattern recognition in an interval of 5 min), applied to all the traffic coming from all the sensors.	0
Vertical rate limit(count of probe request coming in an interval of 5 min), applied to each sensor.	0
Horizontal rate limit(pattern recognition in an interval of 5 min), applied to each sensor.	0
Counting of unique mac-addresses and setting up a threshold every 5 mins, and holding the mac addresses with more activity in memory.	0

Rate limiting/Threshold technique



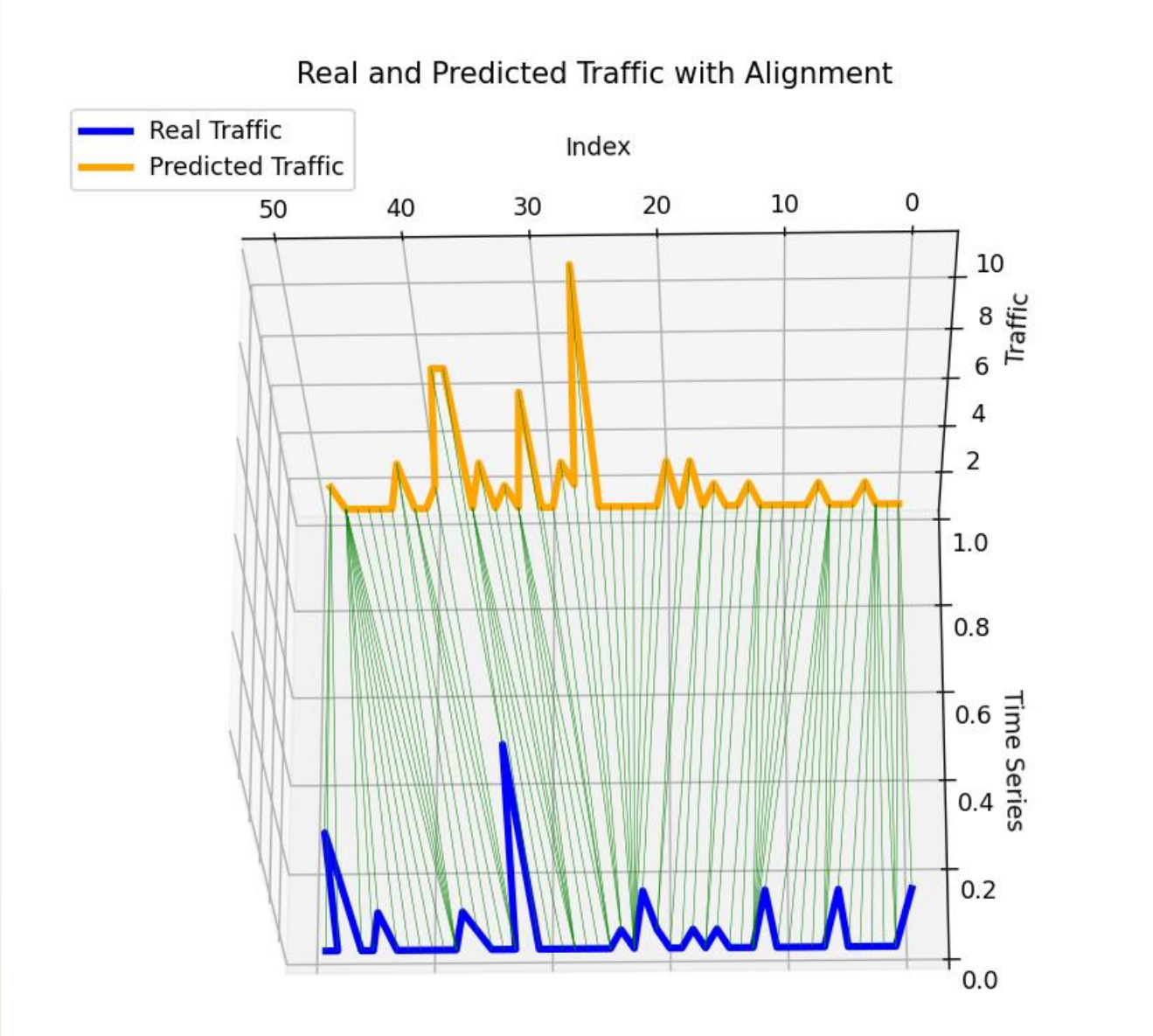
Data to set up a rate limit(Threshold),proposition: predicted plus 25%



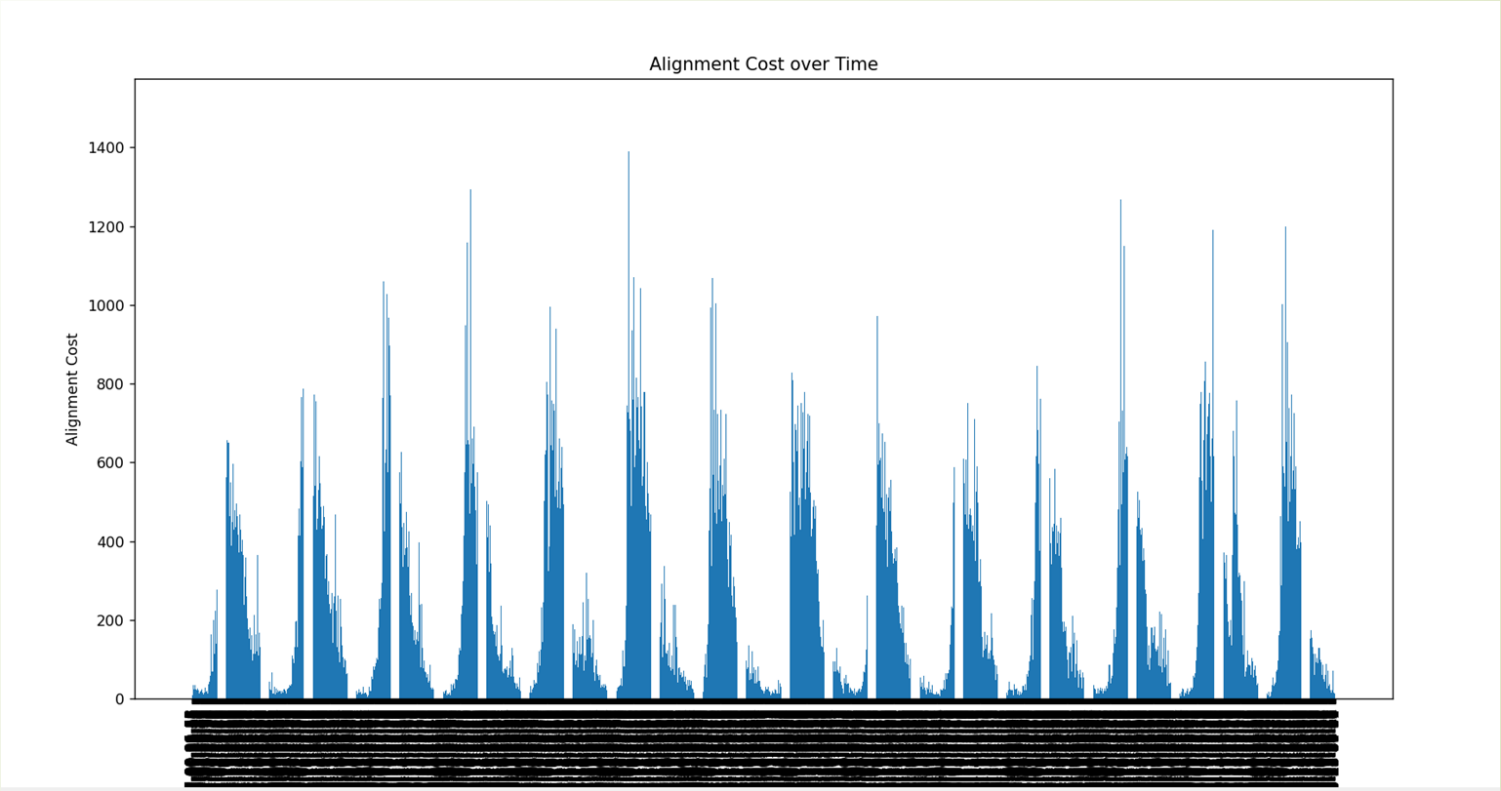
Max Values

Date	Real value	Predicted value
7/17/2023	49	45
7/18/2023	49	45
7/19/2023	56	45
7/20/2023	49	40
7/21/2023	50	45
7/22/2023	52	45
7/23/2023	56	40
7/24/2023	58	45
7/25/2023	50	45
7/26/2023	47	45
7/27/2023	49	45
7/28/2023	48	45
7/29/2023	49	45
7/30/2023	52	45

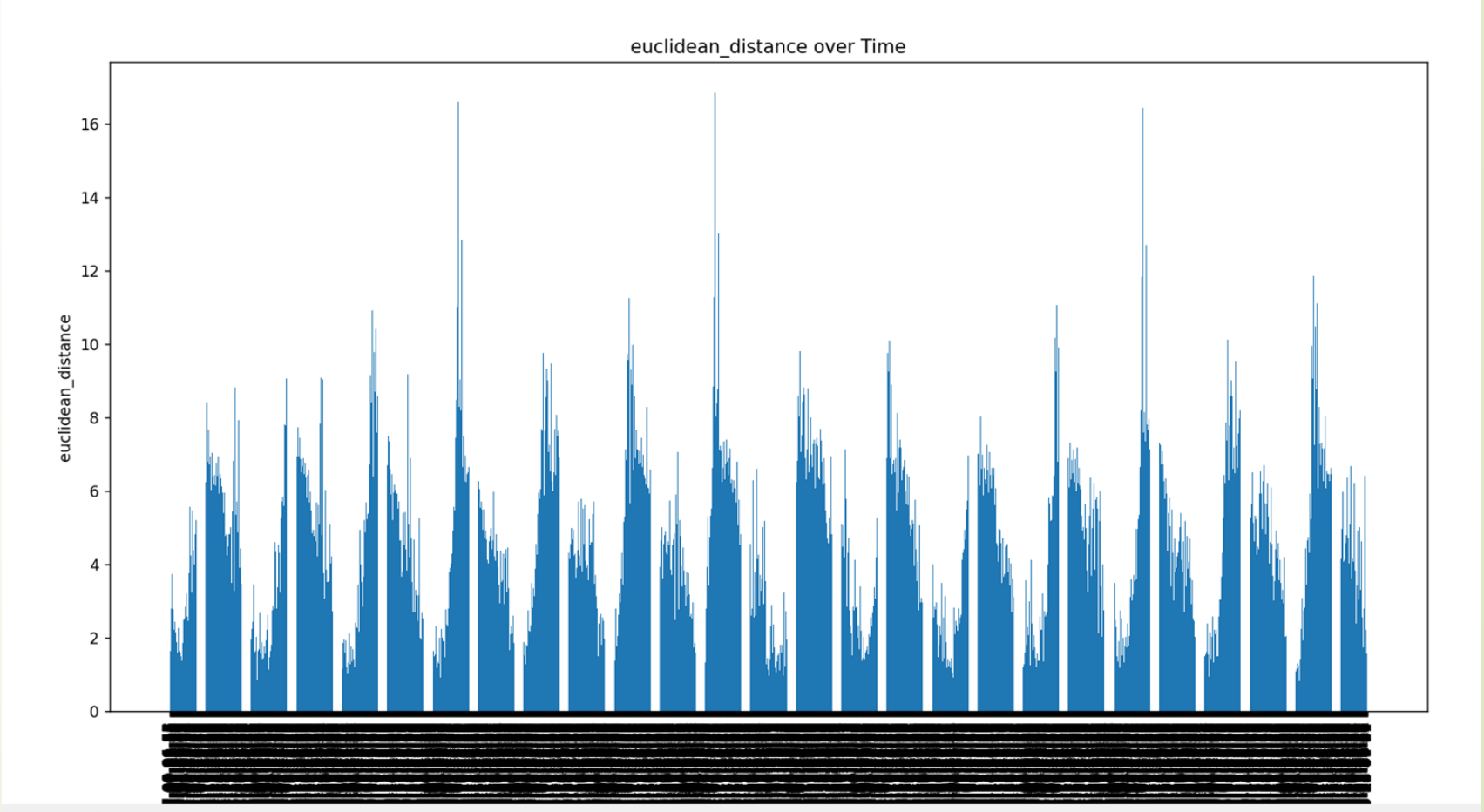
Pattern recognition:



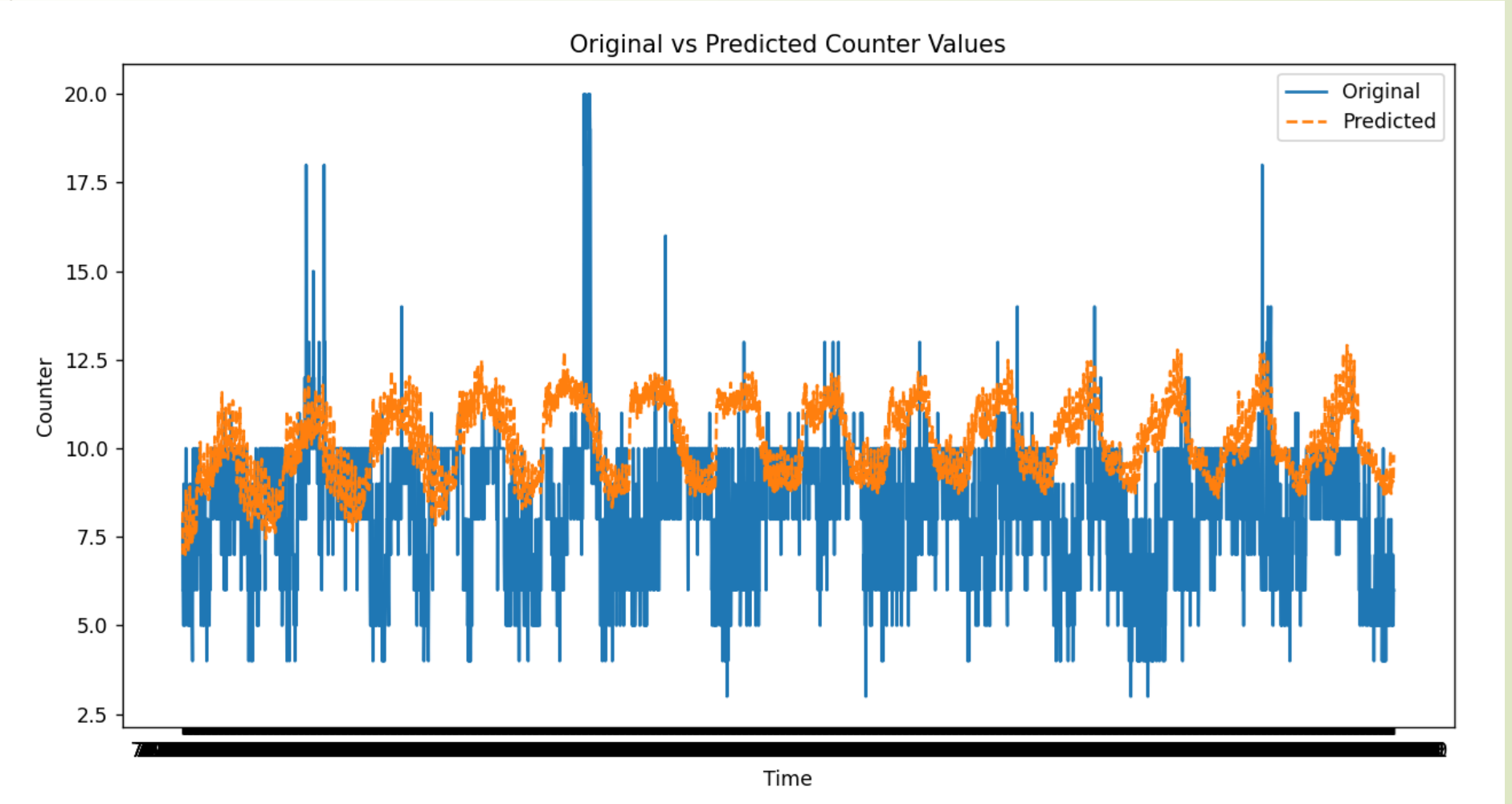
Alignment Cost



Euclidean Distance



Max of unique MACs per 5min, thinking to set up a threshold over the real value(25%).



Logging

anomaly_type	date/time	zone	hashmac	description

- anomaly type: invalid input, abnormal inflow, benign/malign data
- associated zone and hashmac
- data can be used my visualization team

Data Generation

- Data Generated with the Python library Faker
- Data replicates the original data from the data stream
- Data are used to train the models to detect anomalies

Injecting Anomaly Detection Functions with Odysseus Scripts

- Initial attempt to create Odysseus functions but not possible with limited knowledge
- Converting the code we have to PQL and injecting it to the sample of the original Odysseus script.

```
SELECT * FROM testschemadb.flowtrack_raw OFFSET {counter} LIMIT 100
MATCHING ALL (
    eventtype IN ['status', 'leave'],
    zone IN ['bz2452', 'bz2453', 'bz2454', 'bz2457', 'bz2458'],
    techtype = '2',
    mac_address.matches('^[A-Fa-f0-9]{56}$'),
    rssi >= -256 AND rssi <= 0,
    epocutc.matches('^\d{4}-\d{2}-\d{2} \d{2}:\d{2}:\d{2}$')
)
```

QUESTIONS