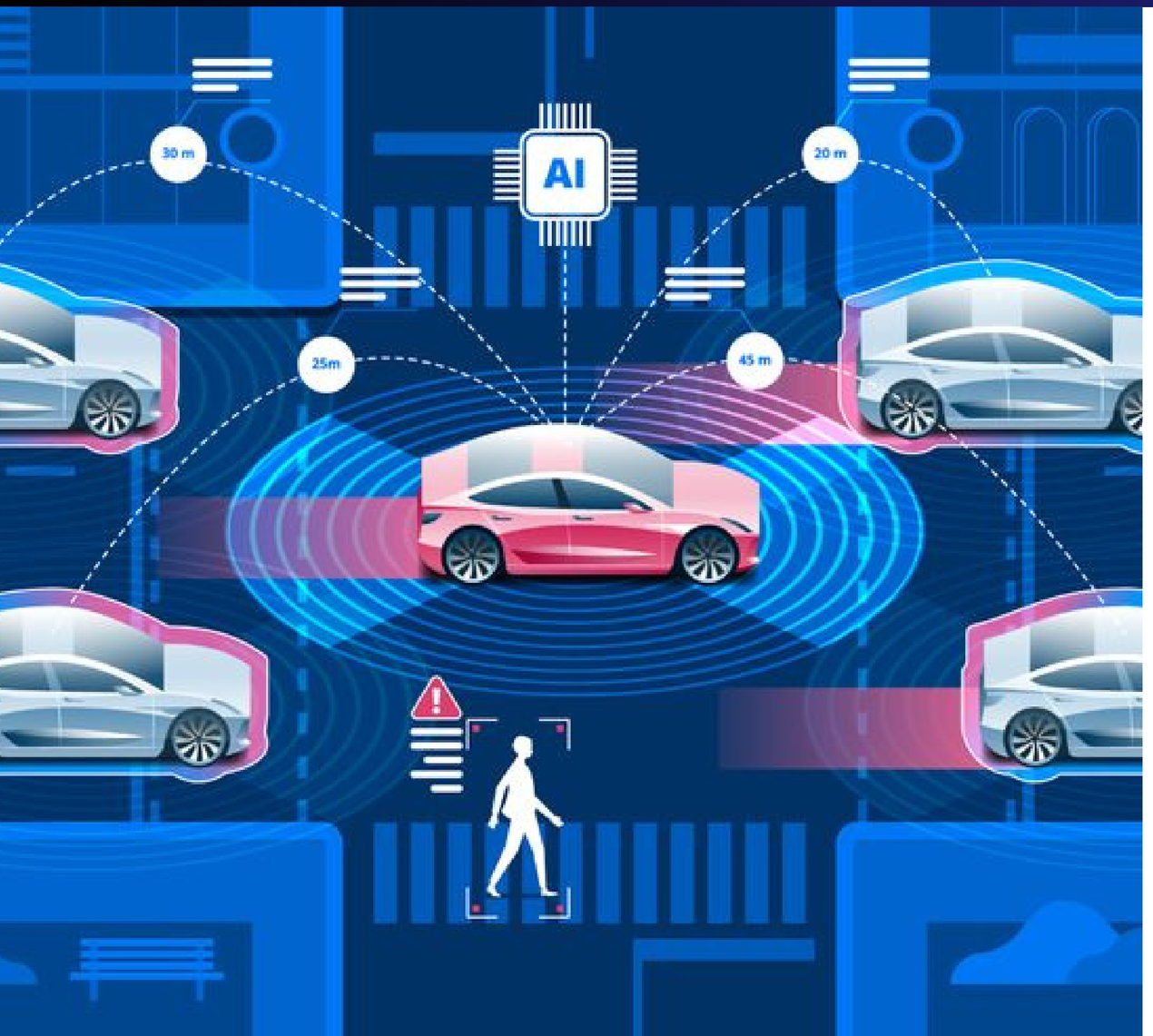
Progress Report

Topic : Intrusion Detection in Internet of Vehicles

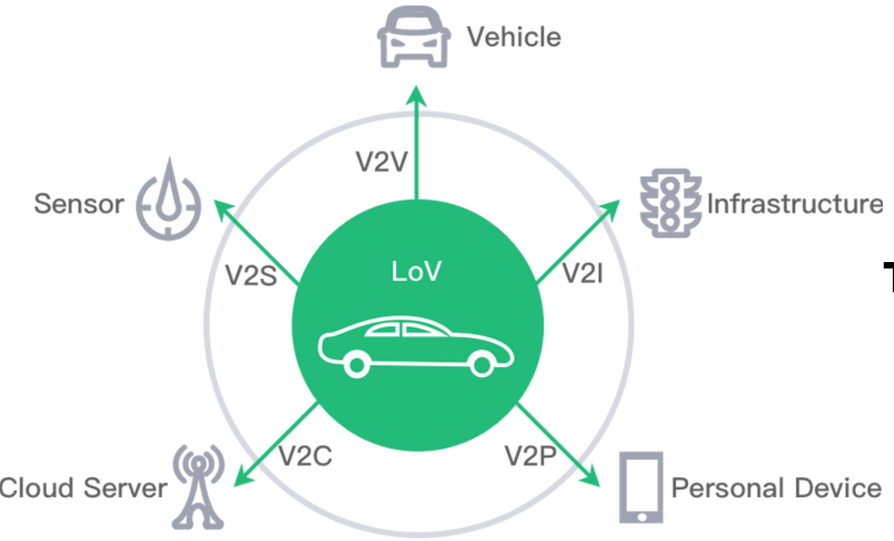
01

What is IoV ?



- The Internet of Vehicles (IoV) seamlessly integrates vehicles with internet connectivity and sophisticated communication technologies, heralding a new era of network-controlled automobiles, including Autonomous Vehicles (AVs) and Connected Vehicles (CVs).
- IoV enables the Intelligent Transportation System (ITS) to communicate with other vehicles, humans, infrastructure, the Internet, and the cloud.
- IoV architecture comprises intra-vehicle networks (IVNs) and external vehicular networks, each playing a pivotal role in enabling advanced functionalities and ensuring seamless communication.

How does IoV communicate ?



Infrastructure:

- IoV communication relies on sophisticated infrastructure including intra-vehicle networks (IVNs) and external vehicular networks.
- Intra-vehicle communication is facilitated by the Controller Area Network (CAN) bus, enabling data exchange among Electronic Control Units (ECUs) within vehicles.

Types of Communication:

- Intra-vehicle communication within IVNs allows for real-time data exchange among ECUs, supporting various vehicle functionalities.
- External vehicular networks utilize Vehicle-To-Everything (V2X) technology, enabling vehicles to communicate with other vehicles, infrastructure, and smart devices.

Technologies:

- The Controller Area Network (CAN) bus serves as the backbone of intra-vehicle communication, facilitating efficient data transfer.
- Vehicle-To-Everything (V2X) communication leverages technologies like Dedicated Short-Range Communication (DSRC) or Cellular Vehicle-to-Everything (C-V2X) to enable connectivity between vehicles and external entities.

Integration:

- The integration of intra-vehicle and external communication networks enables seamless data exchange and communication within the IoV ecosystem.
- This interconnectedness enhances safety, supports advanced features like autonomous driving, and fosters innovation in the automotive industry.

Security Issues in IoV

- **Expanded Attack Surfaces:** Increased connectivity in IoV systems leads to expanded attack surfaces, making them more susceptible to cyber threats. Both intra-vehicle networks (IVNs) and external vehicular networks are vulnerable to attack vectors.
- **Lack of Authentication and Encryption:** The Controller Area Network (CAN) bus in IVNs lacks robust authentication and encryption mechanisms for communication between Electronic Control Units (ECUs).This exposes IVNs to unauthorized access and manipulation of critical vehicle functions.
- **Shortcomings in CAN Protocol:** The CAN protocol's design limitations, such as short message length and lack of security features, make it vulnerable to various attacks. Attackers can inject malicious messages into the CAN bus, potentially compromising or disrupting vehicle systems.
- **External Network Vulnerabilities:** External vehicular networks, utilizing technologies like Vehicle-To-Everything (V2X), are also vulnerable to attacks. Connectivity with external entities introduces risks of data interception, spoofing, and unauthorized access to vehicle systems.
- **Cyber-Physical Attacks:** Exploiting vulnerabilities in IoV communication can lead to cyber-physical attacks, where malicious actions in the digital domain impact physical vehicle operations. Unauthorized remote access to vehicle controls or manipulation of sensor data can lead to accidents or system malfunctions.
- **Emerging Threat Landscape:** The evolving nature of cyber threats presents ongoing challenges for IoV security. Attackers continuously develop new techniques and exploit emerging technologies, necessitating proactive security measures like intrusion detection systems leveraging machine learning algorithms.

Solutions to Security Issues in IoV

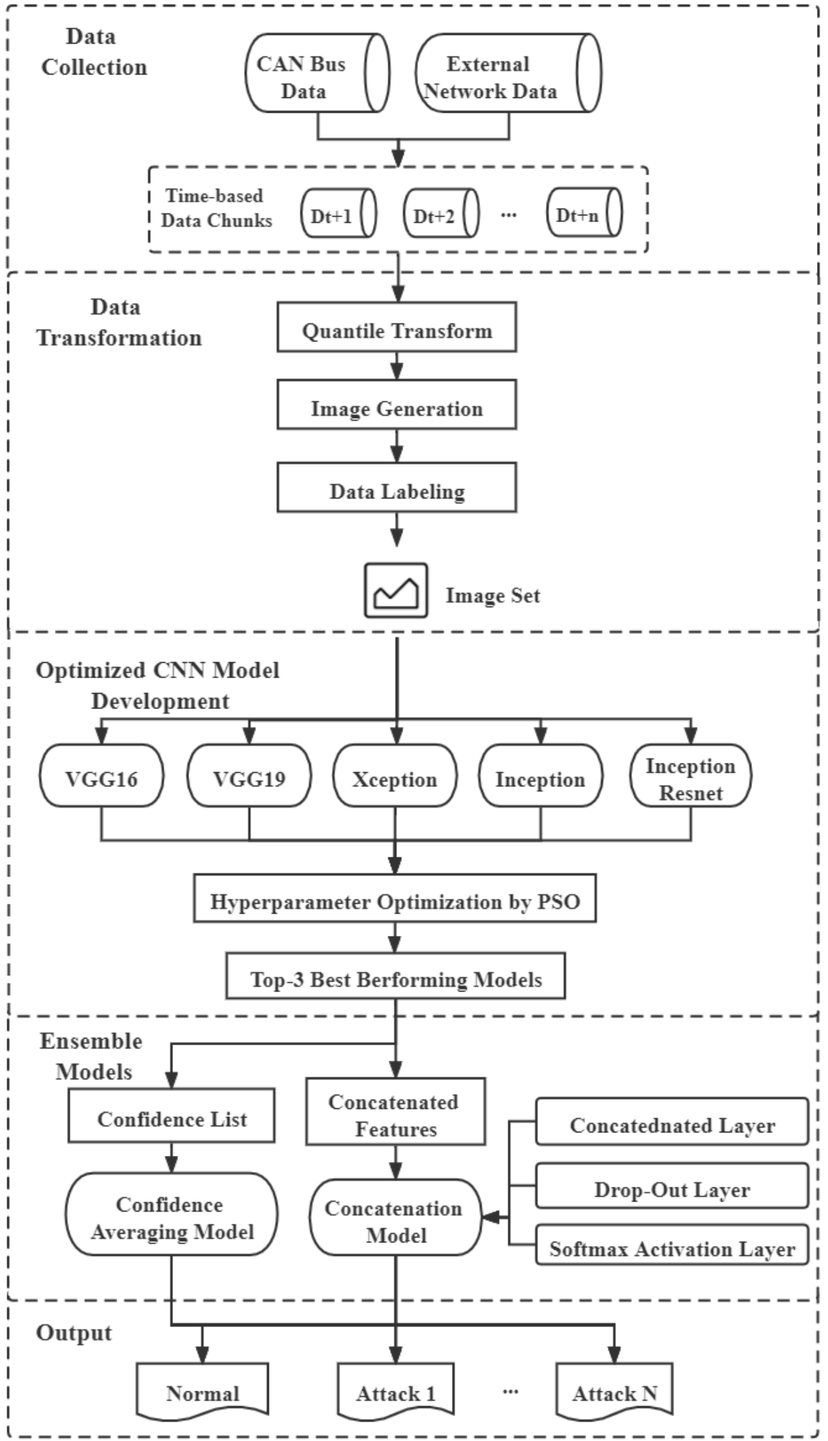
- **Implement Intrusion Detection Systems (IDSs)**: Deploying advanced IDSs like the multi-tiered hybrid IDS (MTH-IDS) facilitates real-time detection and mitigation of cyber threats within Internet of Vehicles (IoV) networks, ensuring proactive defense mechanisms against various types of attacks.
- **Enhance Authentication Mechanisms:** Strengthening authentication protocols within IoV networks enhances security by preventing unauthorized access, thereby safeguarding sensitive data and functionalities of connected vehicles from malicious actors.
- **Secure Communication Protocols:** Utilizing encryption and secure communication protocols ensures the confidentiality and integrity of data transmitted between vehicles and external entities, mitigating the risk of interception, tampering, or unauthorized access by adversaries.
- **Regular Security Audits:** Conducting frequent security audits and updates enables the identification and remediation of vulnerabilities within IoV systems, ensuring that they remain resilient to evolving cyber threats and maintain a high level of security posture.
- **Education and Awareness:** Providing comprehensive education and awareness programs on cybersecurity best practices empowers users and stakeholders within the IoV ecosystem to recognize and mitigate potential security risks, fostering a proactive security culture and reducing the likelihood of successful cyber attacks.

Research Paper (link)

Abstract of The Paper

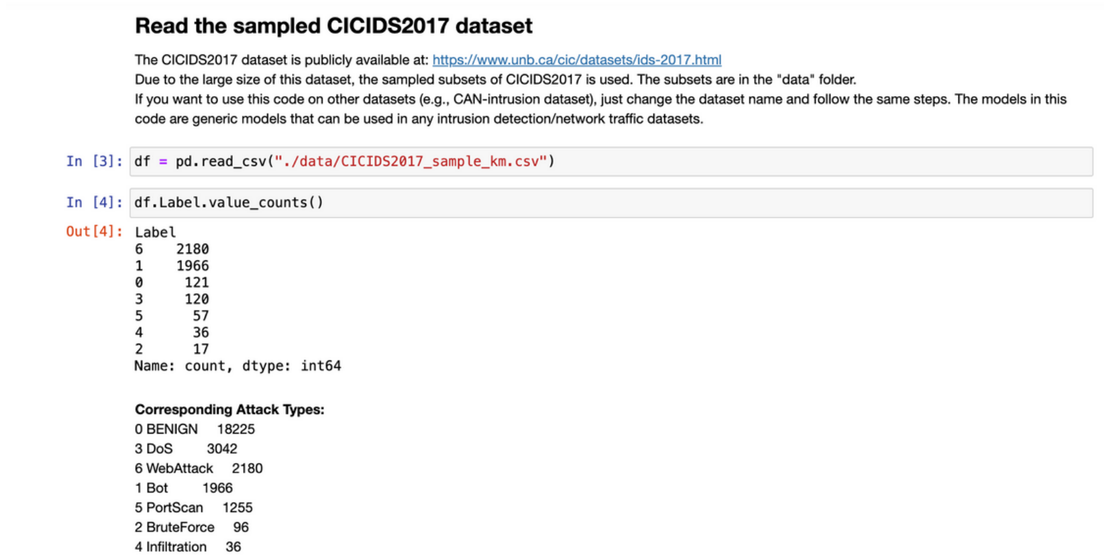
Modern vehicles, including autonomous vehicles and connected vehicles, are increasingly connected to the external world, which enables various functionalities and services. However, the improving connectivity also increases the attack surfaces of the Internet of Vehicles (IoV), causing its vulnerabilities to cyber-threats. Due to the lack of authentication and encryption procedures in vehicular networks, Intrusion Detection Systems (IDSs) are essential approaches to protect modern vehicle systems from network attacks. In this paper, a transfer learning and ensemble learning-based IDS is proposed for IoV systems using convolutional neural networks (CNNs) and hyper-parameter optimization techniques. In the experiments, the proposed IDS has demonstrated over 99.25% detection rates and F1-scores on two well-known public benchmark IoV security datasets: the Car-Hacking dataset and the CICIDS2017 dataset. This shows the effectiveness of the proposed IDS for cyber-attack detection in both intra-vehicle and external vehicular networks.

The Research Paper we have chosen was published in IEEE International Conference on Communications (IEEE ICC) in year 2022.


Data Collection
CAN Bus Data
External Network Data
Time-based Data Chunks
Dt+1
Dt+2
...
Dt+n
Data Transformation
Quantile Transform
Image Generation
Data Labeling
Image Set
Optimized CNN Model Development
VGG16
VGG19
Xception
Inception
Inception Resnet
Hyperparameter Optimization by PSO
Top-3 Best Berforming Models
Ensemble Models
Confidence List
Concatenated Features
Concatednated Layer
Drop-Out Layer
Softmax Activation Layer
Confidence Averaging Model
Concatenation Model
Output
Normal
Attack 1
...
Attack N

What we have done till now ?

1. Implemented Code in Jupyter Notebook: Developed code in a Jupyter Notebook environment for training purposes.
2. Trained Model on CICIDS2017 Dataset: Utilized the CICIDS2017 dataset, a public data benchmark for Intrusion Detection in the Internet of Vehicles (IOV), to train the model.
3. Exploring Simulation Tools: Currently exploring simulation tools for simulating the trained model, with a focus on potentially using OMNET++.
4. Snapshot of Progress: Provided snapshots of the work done thus far, indicating substantial progress in the project.



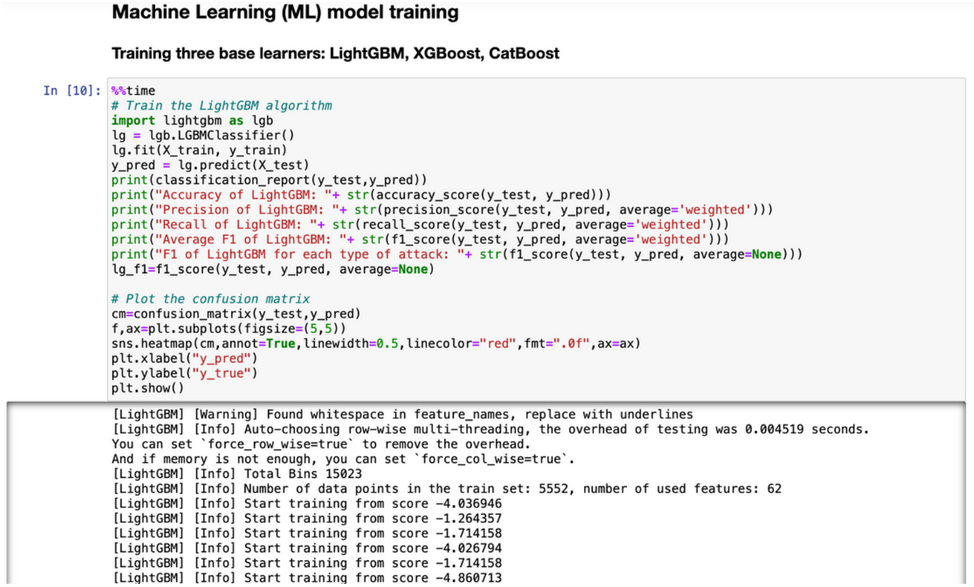
Read the sampled CICIDS2017 dataset

The CICIDS2017 dataset is publicly available at: https://www.unb.ca/cic/datasets/ids-2017.html
Due to the large size of this dataset, the sampled subsets of CICIDS2017 is used. The subsets are in the "data" folder.
If you want to use this code on other datasets (e.g., CAN-intrusion dataset), just change the dataset name and follow the same steps. The models in this code are generic models that can be used in any intrusion detection/network traffic datasets.

```
In [3]: df = pd.read_csv("./data/CICIDS2017_sample_km.csv")

In [4]: df.Label.value_counts()

Out[4]: Label
6    2180
1    1966
0     121
3     120
5      57
4      36
2      17
Name: count, dtype: int64
```

Corresponding Attack Types:

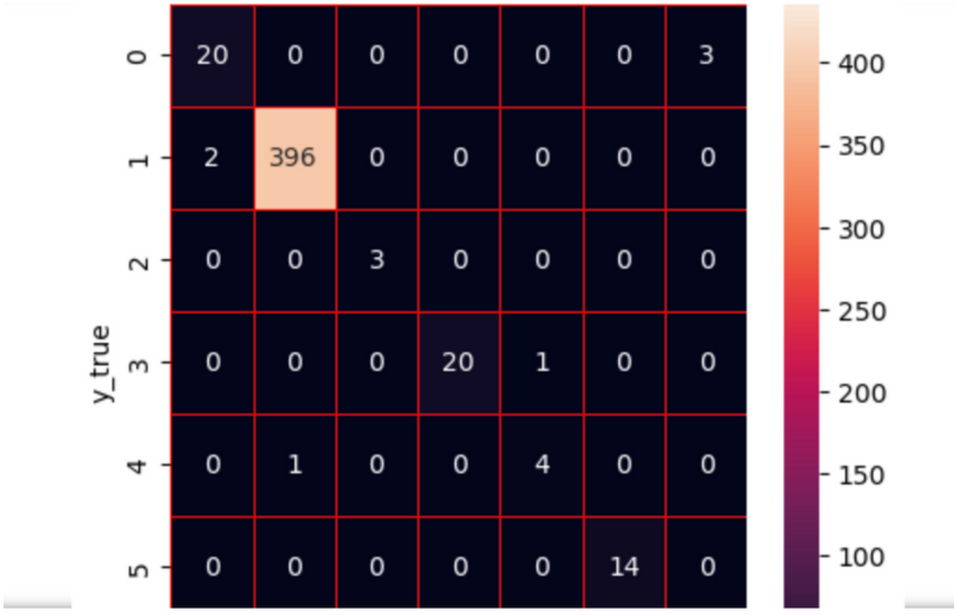
0 BENIGN 18225
3 DoS 3042
6 WebAttack 2180
1 Bot 1966
5 PortScan 1255
2 BruteForce 96
4 Infiltration 36



Machine Learning (ML) model training

Training three base learners: LightGBM, XGBoost, CatBoost

```
In [10]: %%time
# Train the LightGBM algorithm
import lightgbm as lgb
lg = lgb.LGBMClassifier()
lg.fit(X_train, y_train)
y_pred = lg.predict(X_test)
print(classification_report(y_test,y_pred))
print("Accuracy of LightGBM: "+ str(accuracy_score(y_test, y_pred)))
print("Precision of LightGBM: "+ str(precision_score(y_test, y_pred, average='weighted')))
print("Recall of LightGBM: "+ str(recall_score(y_test, y_pred, average='weighted')))
print("Average F1 of LightGBM: "+ str(f1_score(y_test, y_pred, average='weighted')))
print("F1 of LightGBM for each type of attack: "+ str(f1_score(y_test, y_pred, average=None)))
lg_f1=f1_score(y_test, y_pred, average=None)

# Plot the confusion matrix
cm=confusion_matrix(y_test,y_pred)
f,ax=plt.subplots(figsize=(5,5))
sns.heatmap(cm,annot=True,linewidth=0.5,linecolor="red",fmt=".0f",ax=ax)
plt.xlabel("y_pred")
plt.ylabel("y_true")
plt.show()

[LightGBM] [Warning] Found whitespace in feature_names, replace with underlines
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of testing was 0.004519 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 15023
[LightGBM] [Info] Number of data points in the train set: 5552, number of used features: 62
[LightGBM] [Info] Start training from score -4.036946
[LightGBM] [Info] Start training from score -1.264357
[LightGBM] [Info] Start training from score -1.714158
[LightGBM] [Info] Start training from score -4.026794
[LightGBM] [Info] Start training from score -1.714158
[LightGBM] [Info] Start training from score -4.860713
```



In [11]:
```
%%time
# Train the XGBoost algorithm
import xgboost as xgb
xg = xgb.XGBClassifier()

X_train_x = X_train.values
X_test_x = X_test.values

xg.fit(X_train_x, y_train)

y_pred = xg.predict(X_test_x)
print(classification_report(y_test,y_pred))
print("Accuracy of XGBoost: "+ str(accuracy_score(y_test, y_pred)))
print("Precision of XGBoost: "+ str(precision_score(y_test, y_pred, average='weighted')))
print("Recall of XGBoost: "+ str(recall_score(y_test, y_pred, average='weighted')))
print("Average F1 of XGBoost: "+ str(f1_score(y_test, y_pred, average='weighted')))
print("F1 of XGBoost for each type of attack: "+ str(f1_score(y_test, y_pred, average=None)))
xg_f1=f1_score(y_test, y_pred, average=None)

# Plot the confusion matrix
cm=confusion_matrix(y_test,y_pred)
f,ax=plt.subplots(figsize=(5,5))
sns.heatmap(cm,annot=True,linewidth=0.5,linecolor="red",fmt=".0f",ax=ax)
plt.xlabel("y_pred")
plt.ylabel("y_true")
plt.show()
```

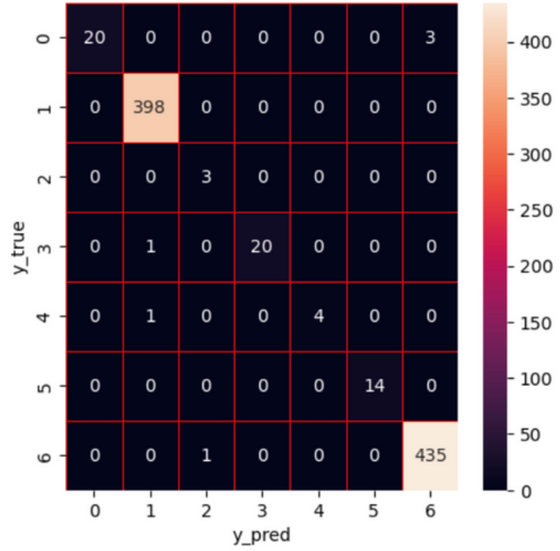
```
              precision    recall  f1-score   support

           0       1.00      0.83      0.90        23
           1       1.00      1.00      1.00       398
           2       0.75      1.00      0.86         3
           3       1.00      0.95      0.98        21
           4       0.80      0.80      0.80         5
           5       1.00      1.00      1.00        14
           6       0.99      1.00      0.99       436

    accuracy                           0.99       900
   macro avg       0.93      0.94      0.93       900
weighted avg       0.99      0.99      0.99       900

Accuracy of XGBoost: 0.9922222222222222
Precision of XGBoost: 0.9925331567973338
Recall of XGBoost: 0.9922222222222222
Average F1 of XGBoost: 0.9920866178223829
F1 of XGBoost for each type of attack: [0.9047619  0.99874529 0.85714286 0.97560976 0.8        1.
 0.99428571]
```

In [17]:
```
# The performance of the proposed lCCDE model
print("Accuracy of LCCDE: "+ str(accuracy_score(yt, yp)))
print("Precision of LCCDE: "+ str(precision_score(yt, yp, average='weighted')))
print("Recall of LCCDE: "+ str(recall_score(yt, yp, average='weighted')))
print("Average F1 of LCCDE: "+ str(f1_score(yt, yp, average='weighted')))
print("F1 of LCCDE for each type of attack: "+ str(f1_score(yt, yp, average=None)))
```

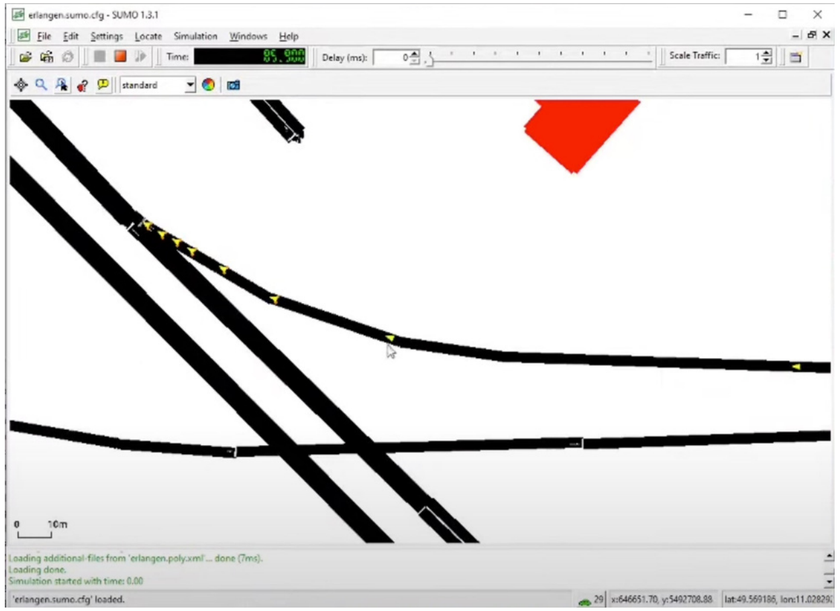
```
Accuracy of LCCDE: 0.9944444444444445
Precision of LCCDE: 0.9944783345988357
Recall of LCCDE: 0.9944444444444445
Average F1 of LCCDE: 0.994261385869455
F1 of LCCDE for each type of attack: [0.93023256 0.99749373 1.         0.97560976 0.88888889 1.
 0.99657143]
```

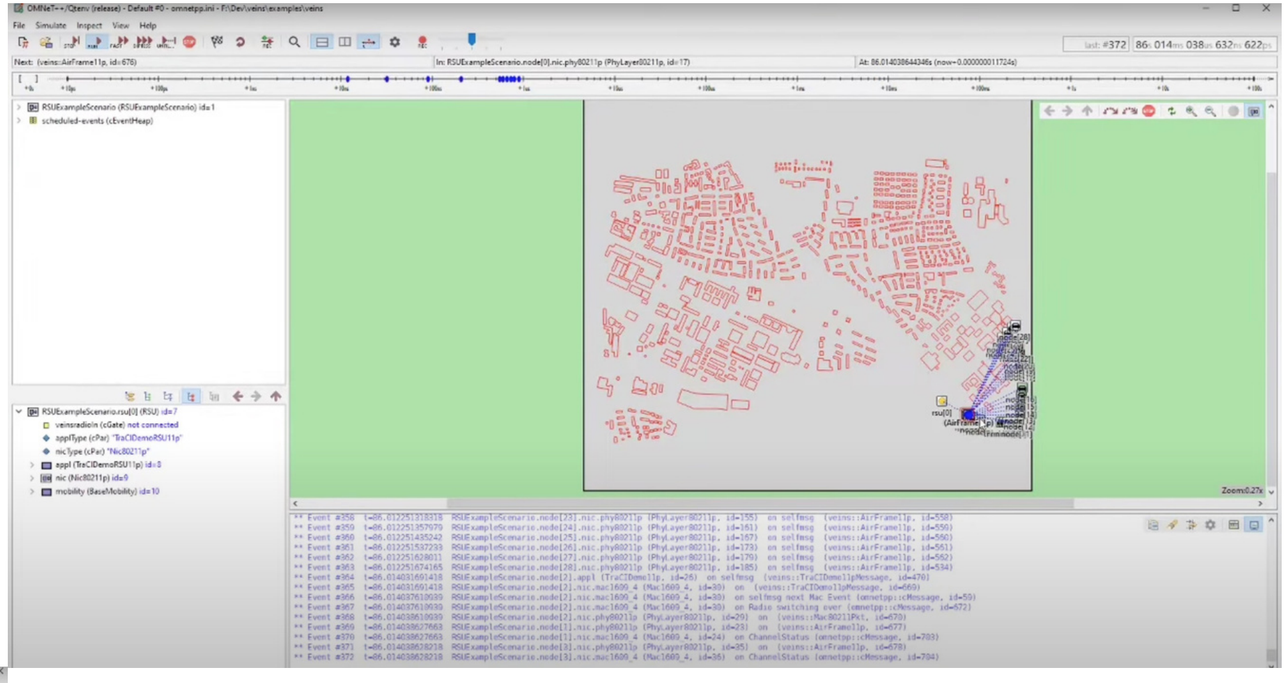
In [18]:
```
# Comparison: The F1-scores for each base model
print("F1 of LightGBM for each type of attack: "+ str(lg_f1))
print("F1 of XGBoost for each type of attack: "+ str(xg_f1))
print("F1 of CatBoost for each type of attack: "+ str(cb_f1))
```

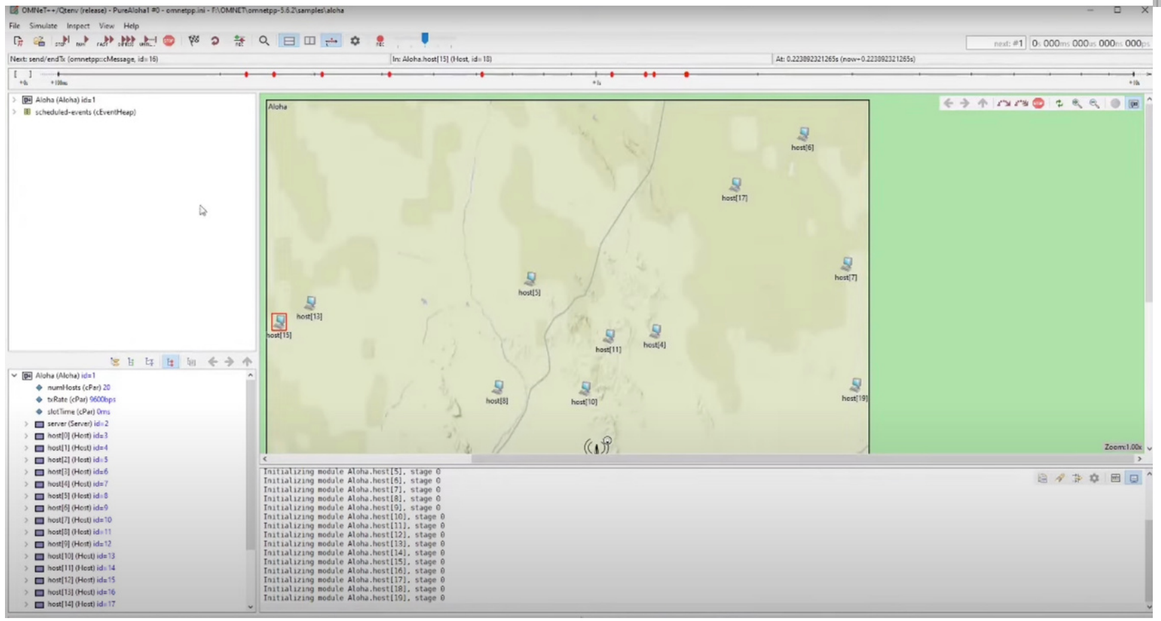
```
F1 of LightGBM for each type of attack: [0.88888889 0.99622642 1.         0.97560976 0.8        1.
 0.99657143]
F1 of XGBoost for each type of attack: [0.9047619  0.99874529 0.85714286 0.97560976 0.8        1.
 0.99428571]
F1 of CatBoost for each type of attack: [0.93023256 0.99749373 0.85714286 0.97560976 0.88888889 1.
 0.99542334]
```

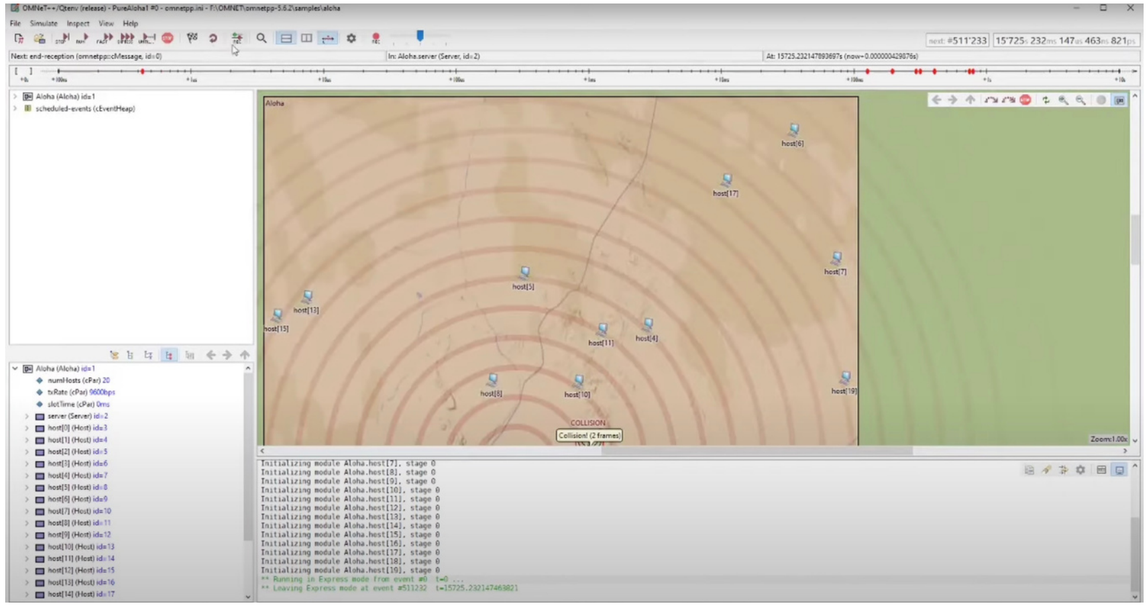



```
CPU times: user 1min 33s, sys: 2.63 s, total: 1min 36s
Wall time: 16.6 s
```


erlangen.sumo.cfg - SUMO 1.3.1
File Edit Settings Locate Simulation Windows Help
Time:
Delay (ms):
Scale Traffic:
standard
0 10m
Loading additional files from 'erlangen.poly.xml'... done (7ms).
Loading done.
Simulation started with time: 0.00
'erlangen.sumo.cfg' loaded.


OMNeT++/Qtenv (release) - Default #0 - omnetpp.ini - F:\Dev\veins\examples\veins
RSUExampleScenario (RSUExampleScenario) id=1
scheduled-events (cEventHeap)
RSUExampleScenario.rsu[0] (RSU) id=7
Zoom:0.27x


OMNeT++/Qtenv (release) - PureAlohaExperiment #0 - omnetpp.ini - F:\OMNET\omnetpp-5.6.2\samples\aloha
Aloha (Aloha) id=1
scheduled-events (cEventHeap)
Aloha
host[0]
host[1]
host[2]
host[3]
host[4]
host[5]
host[6]
host[7]
host[8]
host[9]
host[10]
host[11]
host[12]
host[13]
host[14]
host[15]
Zoom:1.00x


OMNeT++/Qtenv (release) - PureAlohaExperiment #0 - omnetpp.ini - F:\OMNET\omnetpp-5.6.2\samples\aloha
Aloha (Aloha) id=1
scheduled-events (cEventHeap)
Aloha
COLLISION
Collision! (2 frames)
Zoom:1.00x

06

Our Team

Kushaagr Gupta

Branch - Information Technology
Roll No. - 2021UIT3089
Section - 2

Dhruv Mahalwar

Branch - Information Technology
Roll No. - 2021UIT3073
Section - 2


THANK YOU