https://drive.google.com/drive/folders/1TRae9ZtcfVsAWCmctBjk6wpYSgH9lnRD?usp=sharing

Applied Machine Learning for Identifying Malicious Sensor Nodes

633 81555 21 Pawat Songkhopanit

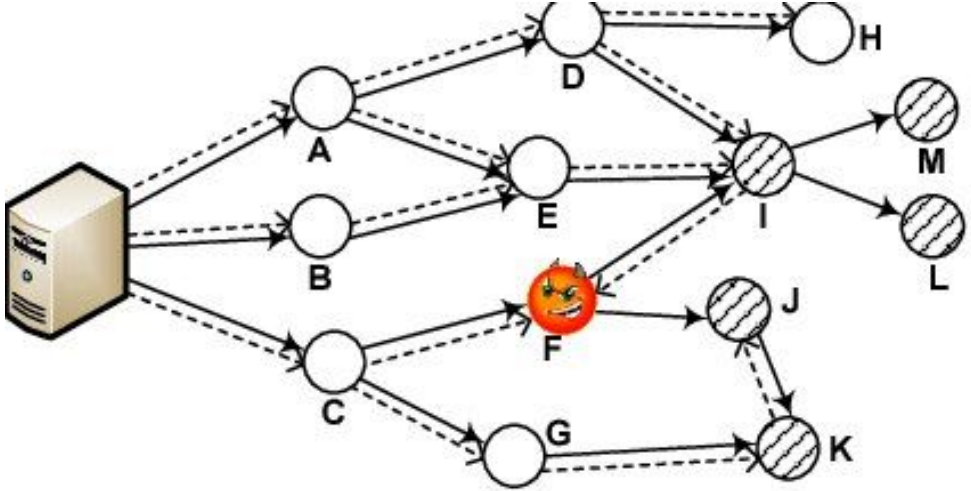
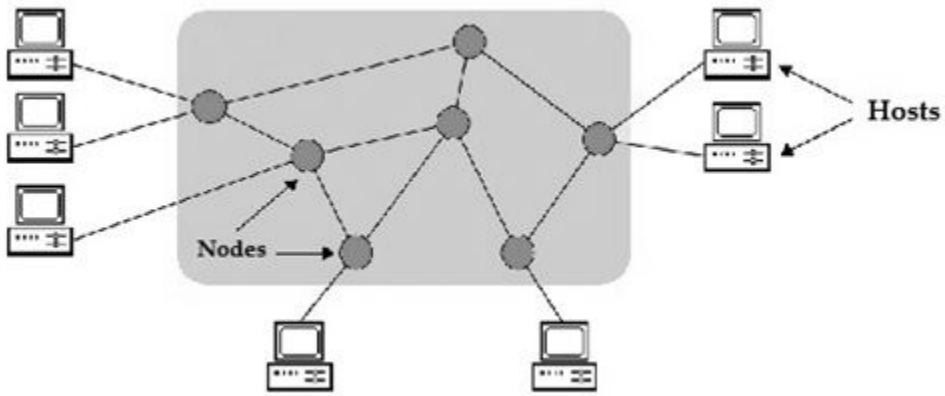


Table of Content

Network nodes

- Any point or device that can send, receive, or forward data packets



Malicious Node

- Device that intentionally acting harmful or disruptive in the systems
- Signs of the attack in the system
  - Lower Network Performance
  - Higher Packet Error Rate
  - Higher Packet Request and Send rate
  - Higher Power Usage
  - Etc.
- How can we detect malicious node?

Traditional Malicious Node Detection

- Intrusion Detection System (IDS)
  - monitors network traffic in real-time
- Intrusion Prevention System (IPS)
  - actively blocking or preventing detected threats in real-time
- Firewall
  - monitors and controls incoming and outgoing network traffic based on predetermined security rules

Problems with traditional methods

- Lots of false negatives and false positives
- Evolving threats
  - Rule-based Systems
  - People can bypass the rule!
- How can we use AI to help with these traditional system?

Big data and AI in Network engineering

- Combining IDS/IPS systems with an AI-based endpoint detection and response (EDR) system
- Highly effective approach to detecting and preventing threats.

Methods and Implementation

- Data Consolidation
- Data Loading
- Data Exploring
- Data Preprocessing
- Modeling
  - Linear Regression
  - Isolation Forests
  - DBSCAN (Density-Based Spatial Clustering of Application with Noise)
  - Support Vector Machine (SVM)

Data Consolidation

[SensorNetGuard: A Dataset for Identifying Malicious Sensor Nodes | IEEE DataPort (ieee-dataport.org)](#)
[GitHub - kmkrphd/SensorNetGuard-A-Dataset-for-Identifying-Malicious-Sensor-Nodes](#)

kmkrphd / SensorNetGuard-A-Dataset-for-Identifying-Malicious-Sensor-Nodes Public

Notifications | Fork 1 | Star 0

Code | Issues | Pull requests | Actions | Projects | Security | Insights

Files

main

Go to file

CODE
LICENSE
README.md
SensorNetGuard A Dataset for Id...

SensorNetGuard-A-Dataset-for-Identifying-Malicious-Sensor-Nodes / CODE

kmkrphd CODE — b11e9f4 · 8 months ago — History

Code | Blame | 52 lines (43 loc) · 1.91 KB | Raw

```
import pandas as pd
import numpy as np
from datetime import datetime
import time

# Function to get real-time sensor data from the testbed
def get_sensor_data(node_id):
    # Replace this function with actual code to collect data from the sensor node
    # This could involve querying a database, an API call, or reading directly from hardware
    # For demonstration, we return a dictionary of random values
    return {
        'Packet_Rate': np.random.normal(50, 10),
        'Packet_Drop_Rate': np.random.normal(2, 0.5),
        # Add all the other attributes here
        'Is_Malicious': np.random.choice([0, 1], p=[0.95, 0.05])
    }

# Initialize an empty DataFrame to store the data
columns = ['Node_ID', 'Timestamp', 'IP_Address', 'Packet_Rate', 'Packet_Drop_Rate', # ... add other columns
           'Is_Malicious']
df = pd.DataFrame(columns=columns)
```

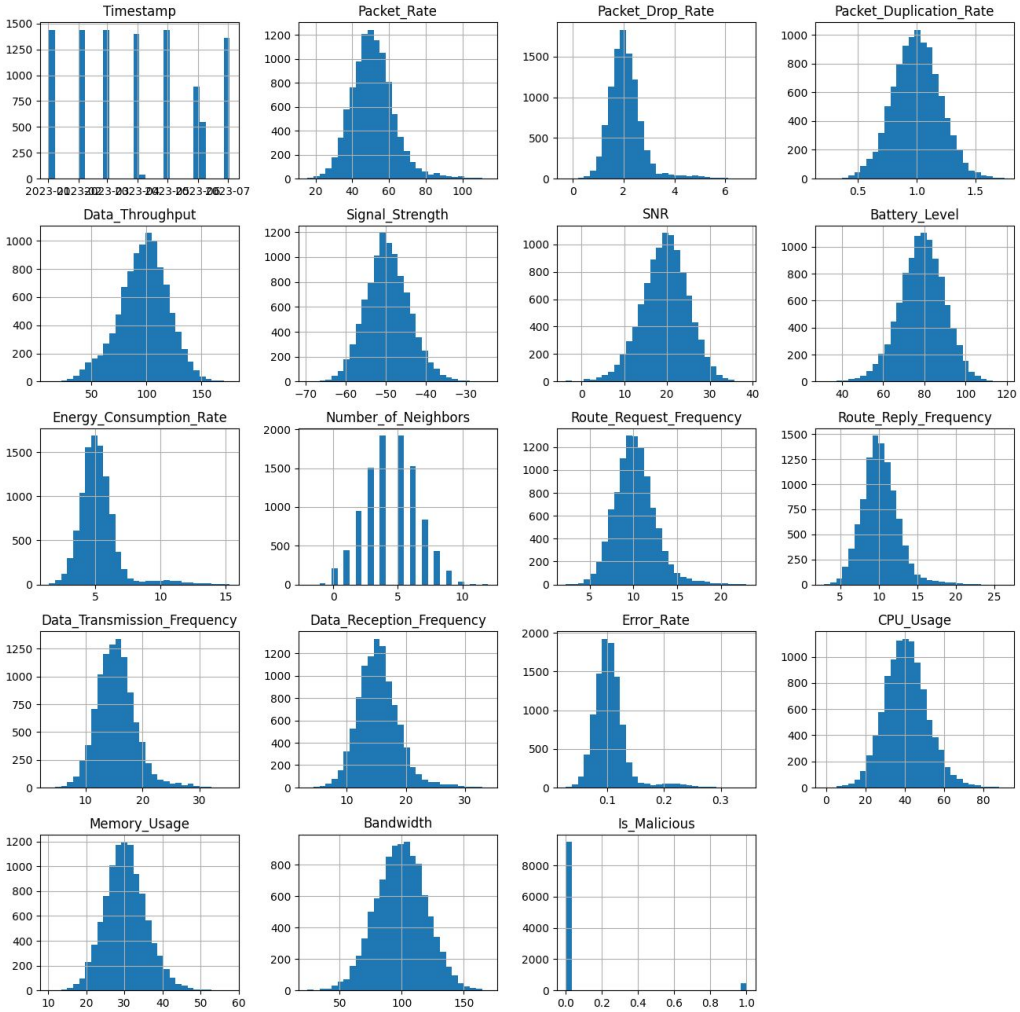
Data Consolidation

Features

- General Metrics: Node ID, Timestamp, IP Address
- Network Traffic Metrics: Packet Rate, Packet Drop Rate, Packet Duplication rate, Data throughput
- Signal Metrics: Signal Strength, Signal-to-Noise Ratio (SNR)
- Power Usage Metrics: Battery Level, Energy Consumption Rate
- Routing Metrics: Number of Neighbors, Route Request Frequency, Route Reply Frequency
- Behavioral Metrics: Data Transmission Frequency, Data Reception Frequency, Error Rate
- Miscellaneous Metrics: CPU Usage, Memory Usage, Bandwidth
- Is_Malicious or not?

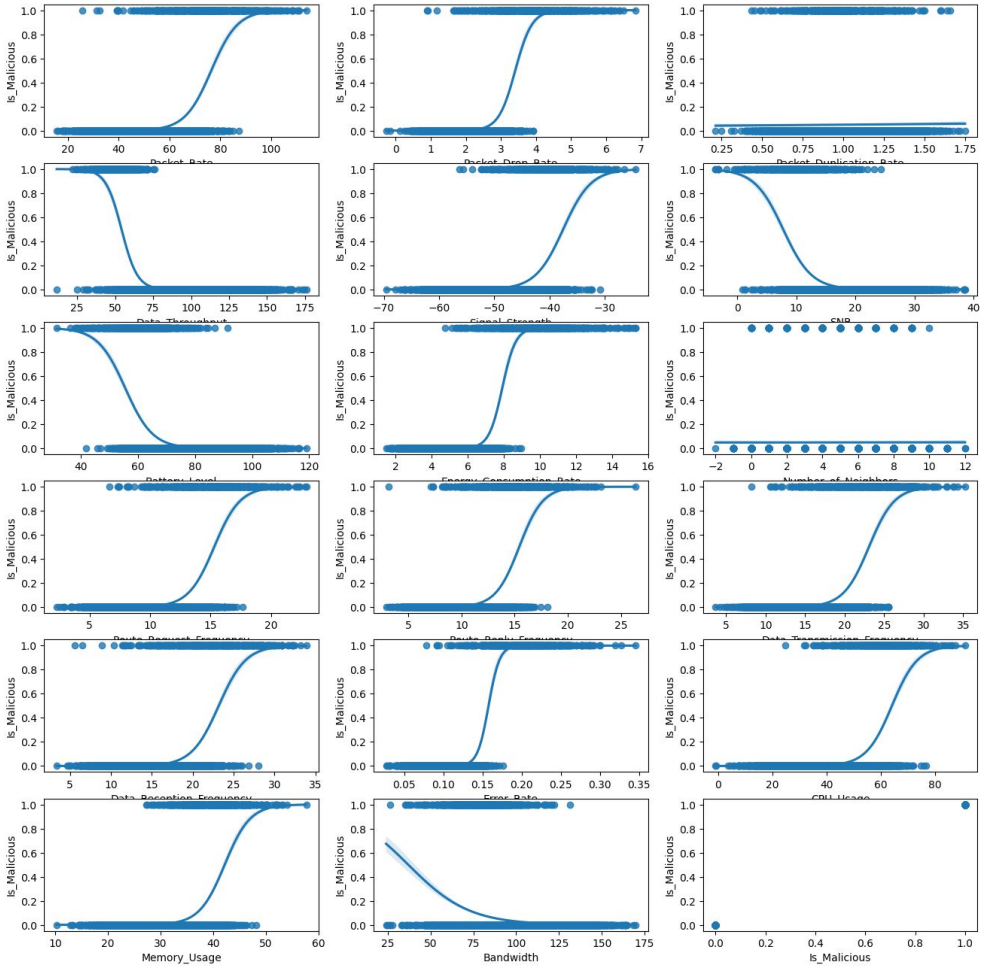
Data Exploring

- Relatively normal distribution
-



Data Exploring

- Clearly a distinction between Malicious and Non-Malicious Nodes

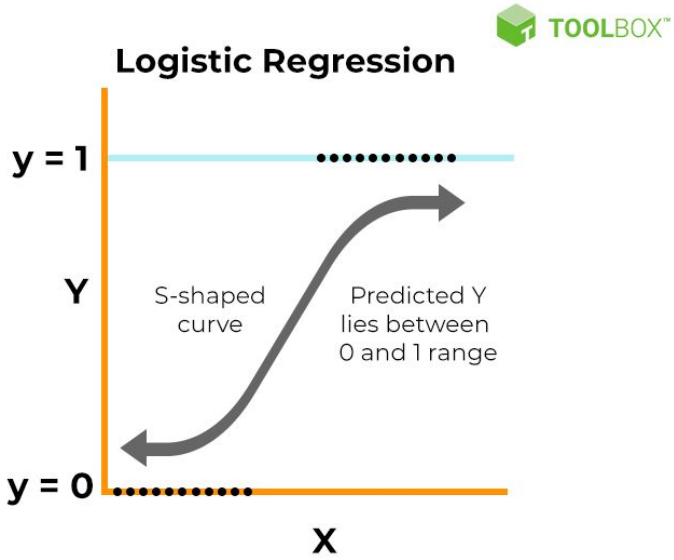


Models

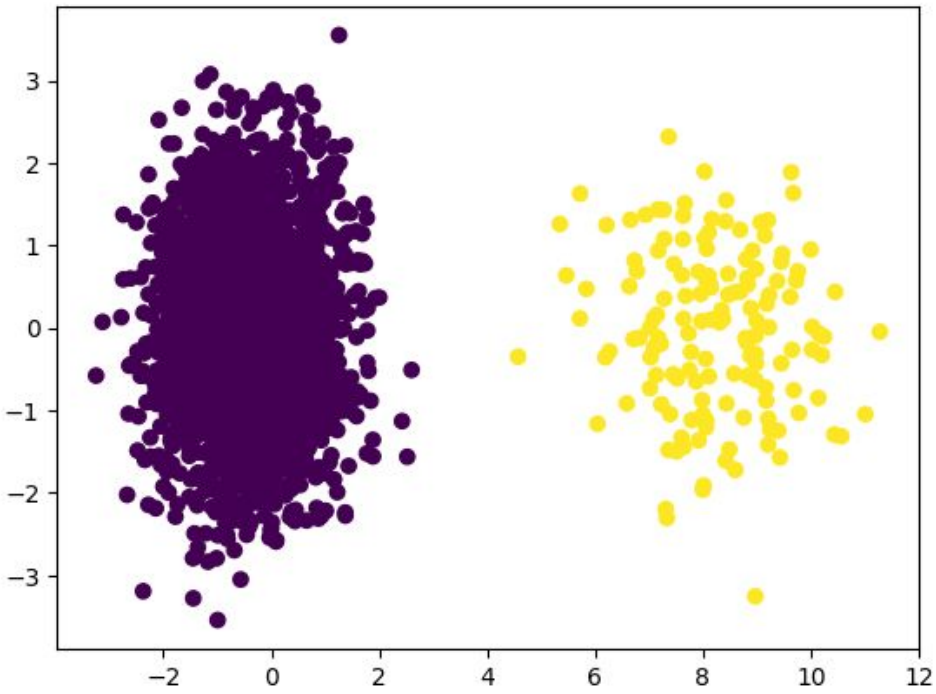
- Logistics Regression
- Isolation Forests
- DBSCAN
- Support Vector Machine (SVM)

Logistics Regression

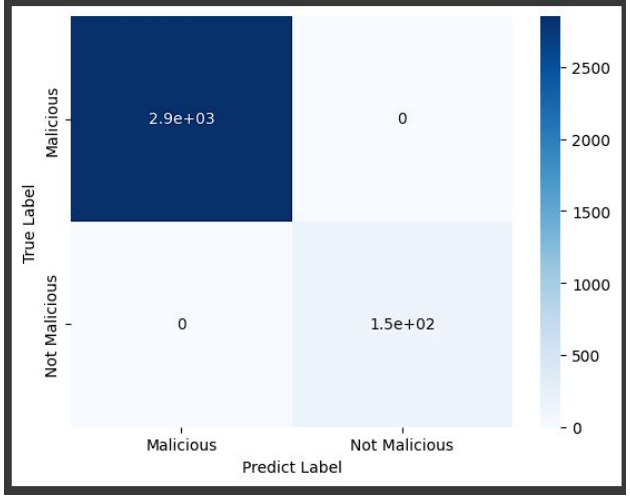
- modeling the probability of a discrete outcome given an input variable
- Malicious or not



Logistic Regression - Results

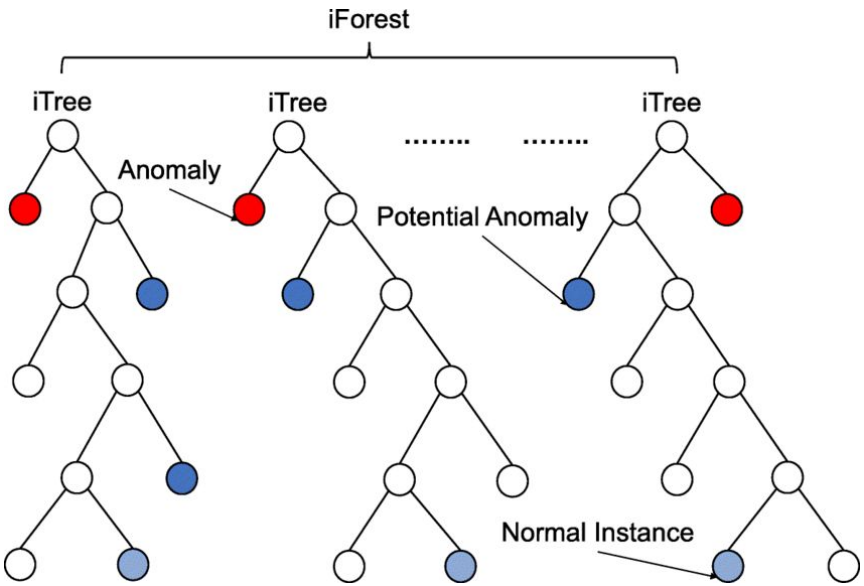



```
Classification Report:
                precision    recall  f1-score   support

    Malicious       1.00      1.00      1.00      2854
Not Malicious       1.00      1.00      1.00       146

     accuracy                           1.00      3000
    macro avg       1.00      1.00      1.00      3000
 weighted avg       1.00      1.00      1.00      3000


Accuracy on train:   1.0

Accuracy on test:  1.0
```

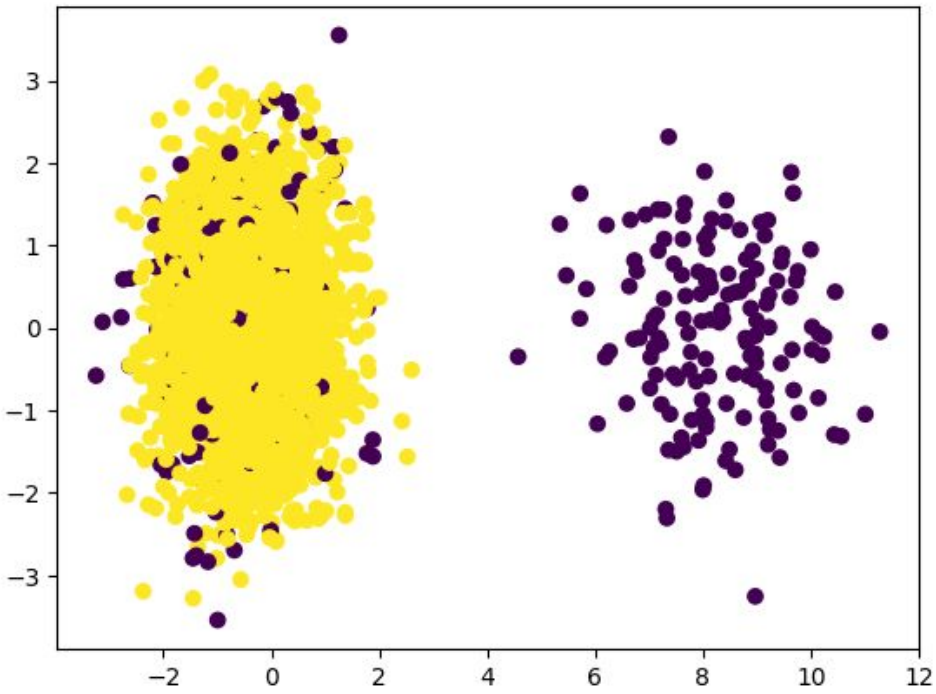


Isolation Forests

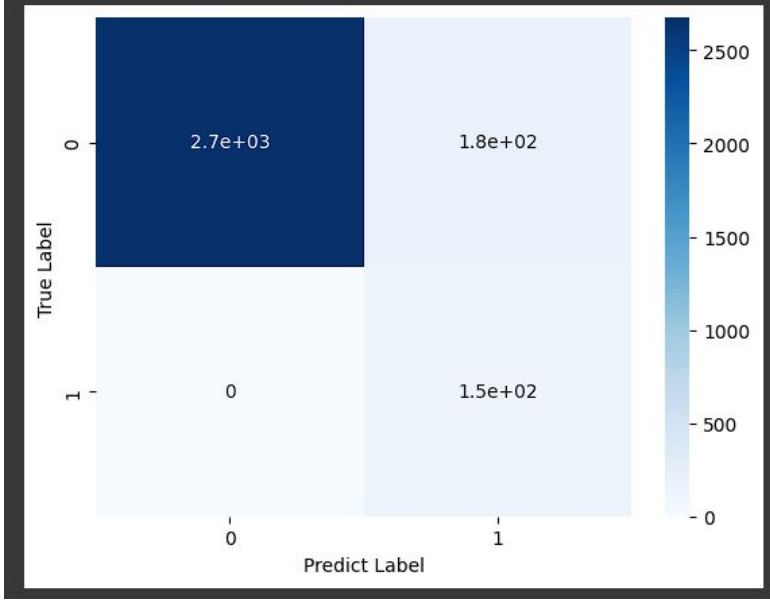
- an algorithm for data anomaly detection
- detects anomalies using binary trees



Isolation Forests - Results

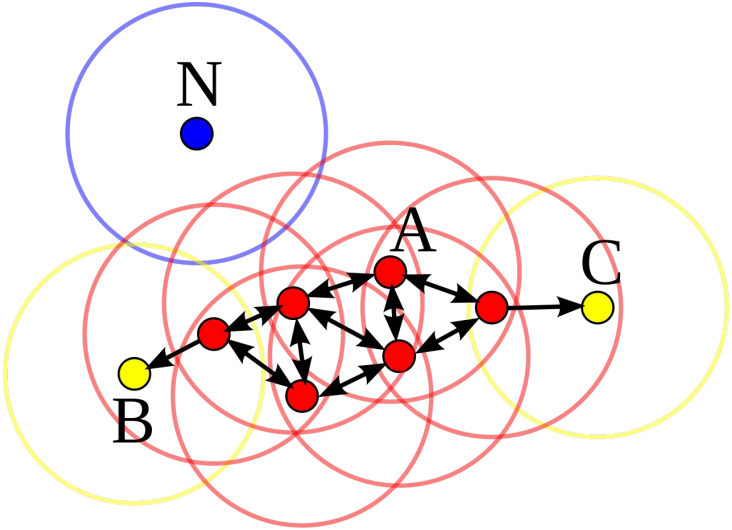



```
Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.94      0.97      2854
           1       0.45      1.00      0.62       146

    accuracy                           0.94      3000
   macro avg       0.73      0.97      0.79      3000
weighted avg       0.97      0.94      0.95      3000
```

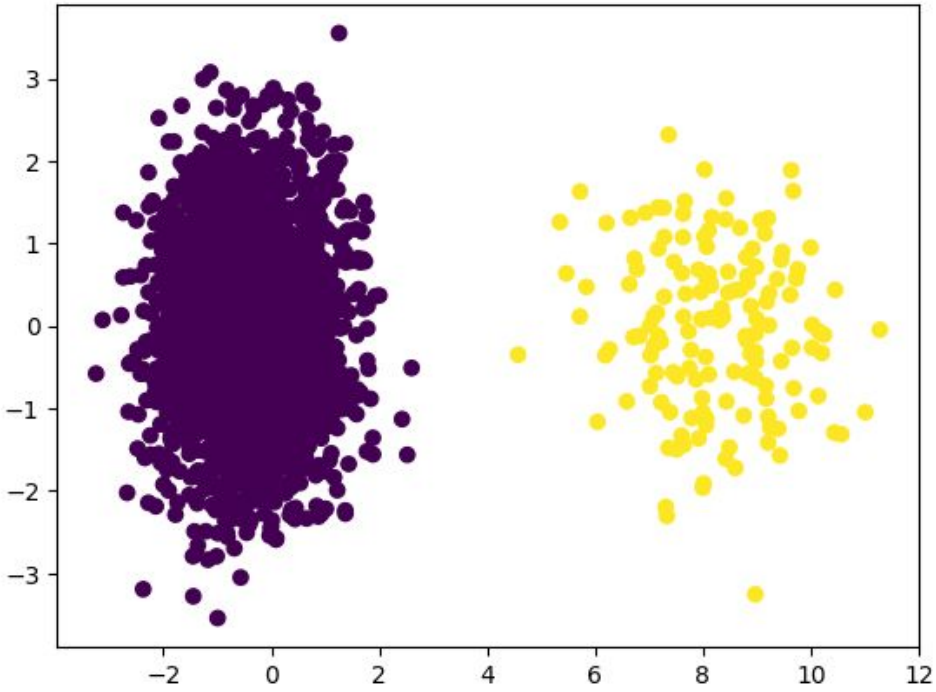


DBSCAN

- Density-based spatial clustering of applications with noise
- marking as outliers points that lie alone in low-density regions



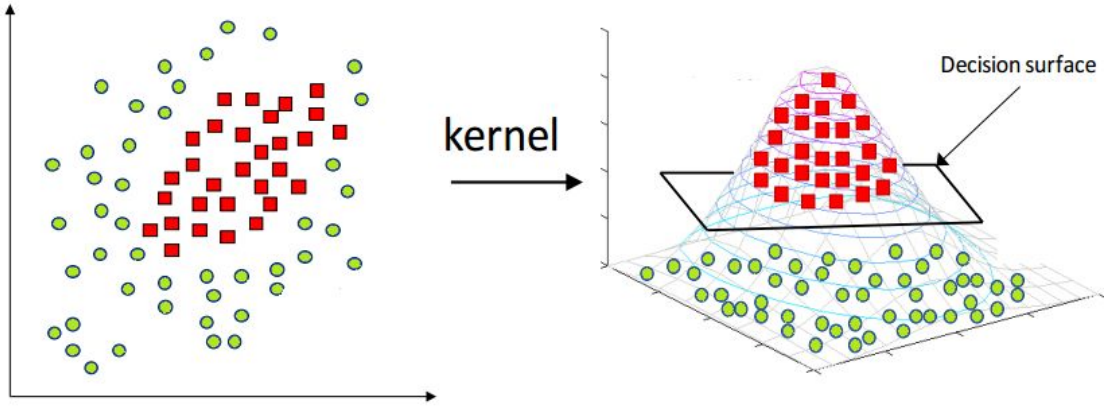
DBSCAN - Results



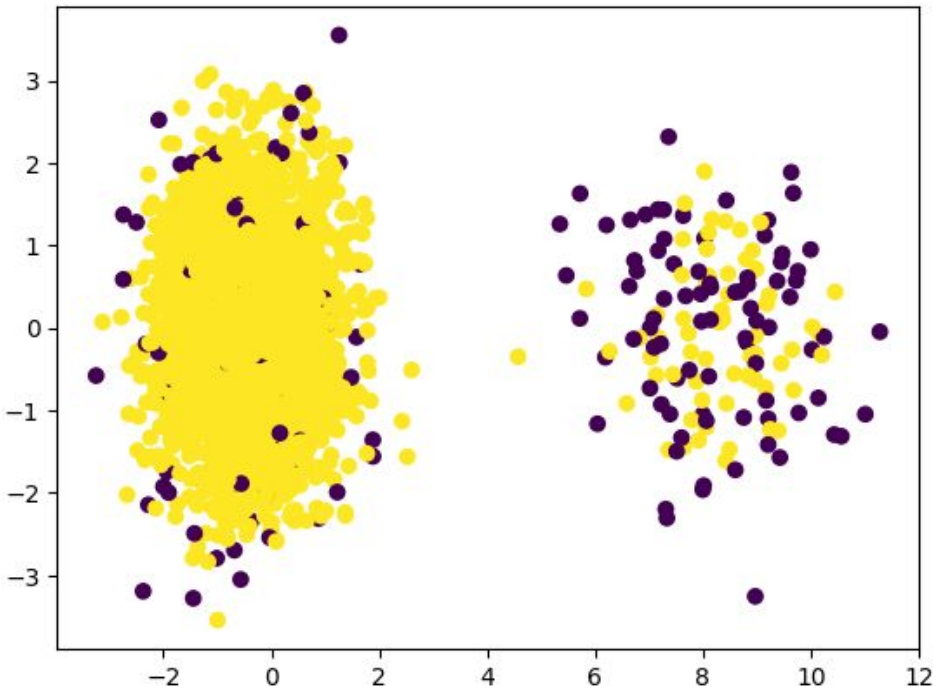

```
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      2854
           1       1.00      1.00      1.00       146

    accuracy                           1.00      3000
   macro avg       1.00      1.00      1.00      3000
weighted avg       1.00      1.00      1.00      3000
```

Support Vector Machine (SVM)

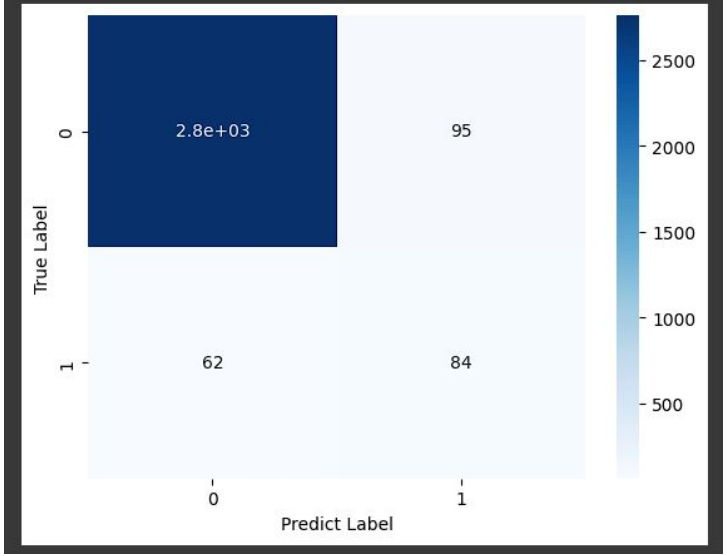
- a supervised machine learning algorithm that classifies data by finding an optimal line or hyperplane



SVM - Results




```
Classification Report:
              precision    recall  f1-score   support

           0       0.98      0.97      0.97      2854
           1       0.47      0.58      0.52       146

    accuracy                           0.95      3000
   macro avg       0.72      0.77      0.74      3000
weighted avg       0.95      0.95      0.95      3000
```



Assumption and Suggestion

- DBSCAN and Logistic Regression have the best results
- DBSCAN might be better than Logistic Regression in the real world scenario
  - Logistic Regression might not detect multiple attackers
  - DBSCAN are better at detecting malicious node
- SVM might also be yield better results with more datasets

Conclusion

- Integrating Big Data and AI with Telecommunication field
- Using DBSCAN and SVM to help with malicious attacker detection and prevention
  - Use AI model to detect a suspicious activity
  - Use AI model to prevent a suspicious activity based on the network tuning parameter