Applied GPS Spoofing Detection on Drones

TU Eindhoven

June 21, 2024

Overview

UAVs/Drones & The Threats of GPS Spoofing Attacks

- Unmanned Aerial Vehicle (UAV) or "Drones"
- The Threats of GPS Spoofing Attacks
- The importance of anti-attack spoofing detection methods

Figure: Various kinds of UAVs

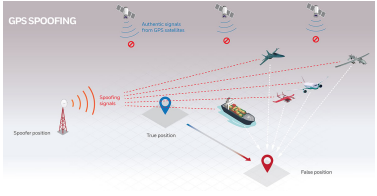


Figure: GPS spoofing

Our Contributions

- Literature review on GPS spoofing and spoofing detection methods.
- Selecting models for this comparative study.
- Carrying out experiments for selected methods on a single dataset.
  - This is a nontrivial task, given the varying scenarios and model inputs across the academic papers in this field.
- Reflecting on the pros and cons of the select methods.

UAVs and GPS

- Global navigation satellite system (GNSS) supports the navigation of a UAV.
- GNSS
  - an umberella term for all satellite-based navigation networks worldwide including the Global Positioning System (GPS).
- GPS Signals
  - Military GPS Signal: Encrypted.
  - Civilian GPS Signal: Not encrypted. Uses publicly available codes.
- Other device on UAVs:
  - GNSS/GPS receiver: Positional data
  - Other sensory devices: IMU, RSS, etc: angular velocity, acceleration, etc.

Figure: GNSS

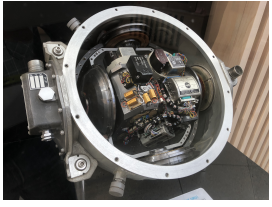


Figure: IMU Sensor

GPS Attacks

Due to the unencrypted nature of civilian GPS signals, the GPS is vulnerable to attacks.

GPS Jamming

Broadcasting radio signals on the same frequency as GPS satellites to overpower or interfere with the relatively weak GPS signals received by a GPS receiver.

GPS Spoofing (Our Focus)

Transmitting fake GPS signals that are structured to mimic legitimate GPS signals.

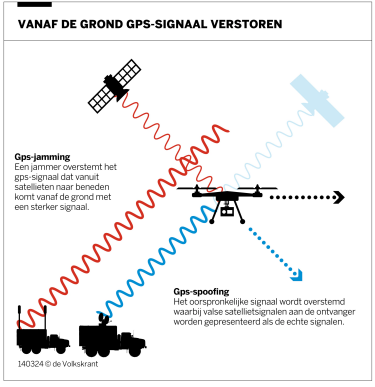


Figure: GPS Jamming and Spoofing

Taxonomy of GPS Spoofing Detection Methods [1]

Signal processing based methods

Detect anomalous jumps in Signal specifications.

- (Oligeri et al., 2019) uses cellular data to estimate UAV locations
- (Elena et al., 2022) proposed a method to normalize UAV signals.

Encryption based methods

Utilize encryption to create unpredictable signal.

Drift based methods

Detect anomalous changes on receiver's position or on clock.

- (Truong et al., 2023) utilize clock bias to detect abnormal signals.
- **Inertial Measurement Unit** Constraining drone's location based on velocity and acceleration.
- (Ian and Ryan, 2020) used two statistical methods to model the signal.
- (Gabriele et al., 2019) used cell sites to estimate the drone's position to detect.
- (Wang et al., 2020) leverages LSTM model to simulate a predicted path to compare the signals received against the predicted path.

[1] Meng, L., Yang, L., Yang, W., & Zhang, L. (2022). A survey of GNSS spoofing and anti-spoofing technology. Remote Sensing, 14(19), 4826.

Taxonomy of GPS Spoofing Detection Methods [2]

Signal/Geographical Location based methods

Monitor the direction of arrival of the signal by considering the received beat carrier phase.

Complementary strategy of multiple detection methods

Mixed detection method combining multiple anti-spoofing strategies rather than a single method.

- (Chafiq and Farid, 2024) used Bayesian methods of multiple varieties.
- (Yalun et al., 2024) physically simulated the drones and usd Kalman filtering to detect attacks.
- (Michieletto et al., 2022) proposed a 3-step mixed modelings, together with different sensory data, to detect attacks on UAV formations and decide corresponding navigation model.

[2] Meng, L., Yang, L., Yang, W., Zhang, L. (2022). A survey of GNSS spoofing and anti-spoofing technology. Remote Sensing, 14(19), 4826.

Selection of Methods

Dataset Coverage

- Able to be conducted on a common, all-inclusive dataset.
- Existing datasets do not satisfy the requirements of different complex methods.
- Limited course timespan.

Reproducibility

- Better with disclosed code examples.
- No transcendal mathematical modeling techniques that we are not able to taclke in the duration of the course.

Choices:

GSM (base model)

GSM-Based Approach (Oligeri et al. 2019).

Machine Learning (Apply the model to GSM dataset)

Novelty-Based Approach: PCA + One-Class Classifiers (Whelan et al. 2020).

Cumulation of Errors (Apply the model to GSM dataset)

Model-based Method (Garrett & Gerdes, 2020).

Problem Formulation

Notations and Symbols

P_{GPS} The received GPS location
P_est The estimated location by GSM signal
x_i The received signal strength by i_th base station
t The time(ms) on each data point
err The distance between p_{GPS} and P_{est}
th The threshold for err
$\hat{t}$ The duration threshold
v_t The drone velocity at time t
a_t The drone accelerate at time t
GSM signal attack detection: $1\{t_j - t_i > \hat{t} | \bigwedge_{x=i}^{j} err_x > th\}$

"Drive Me Not" Dataset

Selected Dataset Oligeri, G., Sciancalepore, S., Ibrahim, O. A., & Di Pietro, R. (2019, May). Drive me not: GPS spoofing detection via cellular network: (architectures, models, and experiments). In Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks (pp. 12-22).

Scenario and Attacker Behavior

- **Fixed Route**
  - Common for UAVs carrying out patrol missions and transporting missions
- **Regularity**
  - The flight path and parameters in a normal flight condition have certain regularity / learnable patterns.
- **Attacker**
  - **Attacker Knowledge** The attacker can know the starting point and return point of the target UAV, and can track the current speed, position and other information of the UAV.
  - **Objective of the Attacker** spoof the UAV to the route set by the attacker.

data > drive-me-not > trace1.csv > data

You, last month | 1 author (You)

```
1  GPS_lat,GPS_long,Network_lat,Network_long,Time,Anchor_Number,Type,Registered,CID,LAC,MCC,MNC,dBm,level
2  25.32834666666667,51.42506,0.0,0.0,1561278474631,0,GSM,true,23733,9301,427,1,-45,4
3  25.32834666666667,51.42506,0.0,0.0,1561278474631,1,GSM,false,29583,9301,427,1,-31,4
4  25.32834666666667,51.42506,0.0,0.0,1561278474631,2,GSM,false,24882,9301,427,1,-31,4
5  25.32834666666667,51.42506,0.0,0.0,1561278474631,3,GSM,false,22223,9301,427,1,-33,4
6  25.32834666666667,51.42506,0.0,0.0,1561278474631,4,GSM,false,22222,9301,427,1,-43,4
7  25.32834666666667,51.42506,0.0,0.0,1561278474631,5,GSM,false,22231,9301,427,1,-63,4
8  25.32834666666667,51.42506,0.0,0.0,1561278474631,6,GSM,false,21253,9301,427,1,-65,4
9  25.32834666666667,51.42506,0.0,0.0,1561278474631,7,GSM,true,10011,150,427,2,-29,4
10 25.32834666666667,51.42506,0.0,0.0,1561278474631,8,GSM,false,10511,150,427,2,-17,4
11 25.32834666666667,51.42506,0.0,0.0,1561278474631,9,GSM,false,10513,150,427,2,-31,4
12 25.32834666666667,51.42506,0.0,0.0,1561278474631,10,GSM,false,10903,181,427,2,-39,4
13 25.32834666666667,51.42506,0.0,0.0,1561278474631,11,GSM,false,16362,150,427,2,-45,4
14 25.32834666666667,51.42506,0.0,0.0,1561278474631,12,GSM,false,10901,181,427,2,-49,4
15 25.32834666666667,51.42506,0.0,0.0,1561278474631,13,GSM,false,10283,150,427,2,-51,4
16 25.32834666666667,51.42506,0.0,0.0,1561278474768,0,GSM,true,23733,9301,427,1,-45,4
17 25.32834666666667,51.42506,0.0,0.0,1561278474768,1,GSM,false,29583,9301,427,1,-31,4
```

Figure: Example of original trace data

Generation of Datasets - Sensor Data

Generation of IMU sensor data

- For (Whelan et al. 2020): Lacking 85 Sensory Data columns.
- To generate velocity and acceleration data from the traces.
- Cons: Choppy and zigzaggy data, not smooth.
- Differentiation? Moving Average? Outlier Handling?

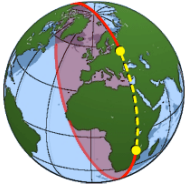


Figure: Haversine distance is needed for calculating distance from coordinates

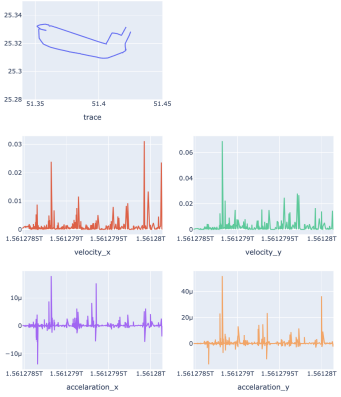


Figure: Example: Generated Velocity / Acceleration Data w.r.t the Original Trace of Trace 1

Generation of Datasets - Spoofed Traces

- Randomly selecting a direction.
- Simulating the speed in Normal Distribution among every data point.
- Spoofing starting from the middle of the benign trace.

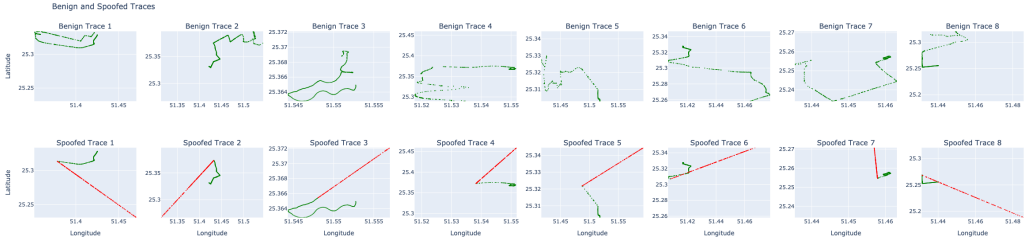


Figure: Spoofed Traces Used

GSM (Oligeri et al. 2019)

Cellular Network Position Estimation

GSM + UMTS

Received Signal Strength Modelling

Using Exponential Distribution to calculate the weight of base stations.

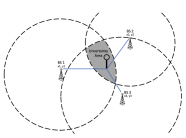


Figure: Position relation between base station and drone

$$y = 1 - f(x, \mu) = 1 - \frac{1}{\mu} e^{-\frac{x}{\mu}},$$

$$w_i = \frac{y_i}{\sum_{i=1}^{N} y_i} \tag{1}$$

Model Parameters

Model Parameters

- μ **in Exponential Distribution:** Minimizing the distance between GPS location and Estimated GSM location.
- **Anomaly threshold:** The difference between 2 locations beyond the threshold as anomaly.
- **Anomaly sequence duration threshold:** The anomaly sequence duration beyond the threshold as attack.

Effectiveness

- A lot of base stations around every location.

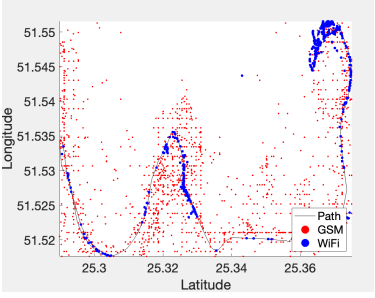


Figure: One UAV trace with base station location

- UAV receives 11-15 Base station signals in most of positions.

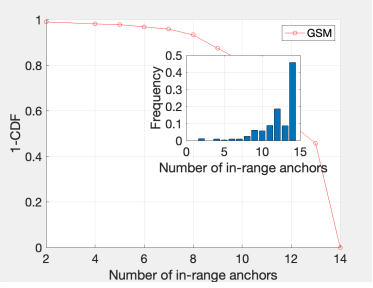
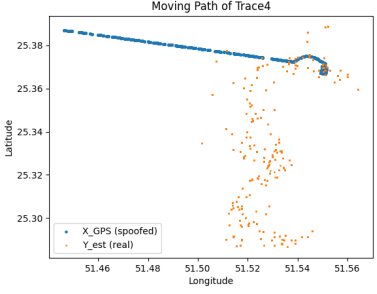


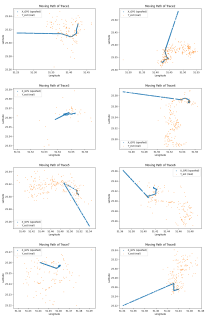
Figure: The distribution of received base station signals

Threshold Estimation

Datasets

- 8 traces collected from Doha, Qatar, driving more than 150km for about 10 hours (Gabriele et al. 2019).
- Trace4 as an example:
  Y_est: position estimated by mobile cellular network.
  X_GPS: position received by the GPS (real signal and our generated spoof signal).





- **Train and Test dataset**
  Train set: Trace 5-8
  Test set: Trace 1-4
- **Training policy**
  Minimizing the duration and distance before spoofed trace being detected

Experiment Results

- **Determine the ideal threshold using data quantiles.**

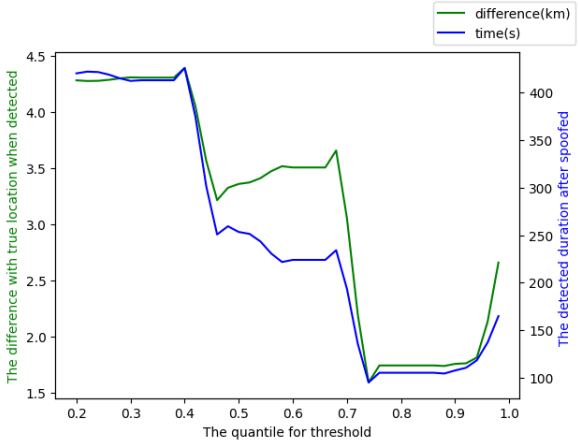


Figure: Detection duration and distance relative to the quantile of the training dataset

Experiment Results

Duration of anomaly sequence

Guarantee 0 false positive rate in training dataset and minimalize the detection duration and distance.

Datasets	False Positive	Duration	Distance
trace1	0.17	100.4	2.23
trace2	0.11	93.3	1.33
trace3	0	93.6	0.94
trace4	0.05	151.0	1.34
trace5	0	173.9	1.83
trace6	0	96.4	1.57
trace7	0	155.1	1.74
trace8	0	144.0	1.41

Table: Drive me not model

PCA + One-Class Classifiers

Make use of a variety of sensor data.

Principle Component Analysis (PCA)

- Decompose to a set orthogonal components with explainability on most of the variance.
- (Whelan et al. 2020) Reduce 85 features to 3 principal components.
- Fit PCA on the training set to transform both the testing and training set.

Feed the reduced components into the OCC models.

Novelty Detection with One-Class Classifiers

- Does **not** require anomalies to pre-exist in the dataset during training.
- New observations outside of learned distribution are classified as **novelties**.
- Can be pre-trained. Requires only flight logs.

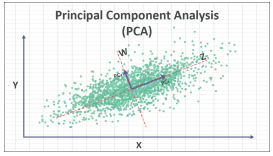


Figure: PCA constructs PCA components with low dimensions

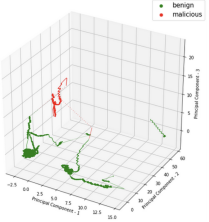


Figure: Reduced components in 3d space (Whelan et al., 2020)

One-Class Classifiers for Novelty Detection

One-Class SVM (OCSVM)

- Unsupervised anomaly detection algorithm.
- Extension of SVM to distinguish the majority of data points from outliers.

Local Outlier Factors (LOF)

- Unsupervised density-based anomaly detection algorithm.
- Assigns an outlier score to each data point based on the local density of its neighborhoods.
- Outliers: points with significantly lower density than their neighbors.

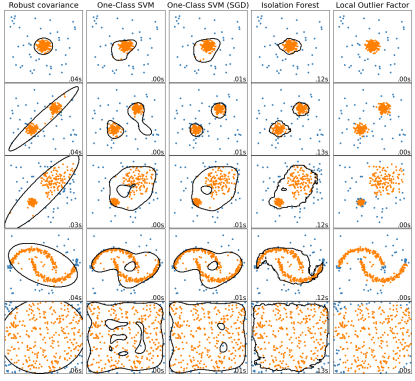


Figure: One-Class Classifier demonstration from sklearn documentation
`https://scikit-learn.org/stable/modules/outlier_detection.html#overview-of-outlier-detection-methods`

One-Class Classifiers for Novelty Detection

Autoencoder

- Hidden encoded layer: compression-decompression reconstruction.
- Only benign data are supposed to reconstruct itself from the model.
- Trained on a benign dataset with MSE loss.
- Compare the MSE difference of the reconstructed input with a Threshold T (how to choose the threshold is not disclosed in (whelan et al. 2020)).

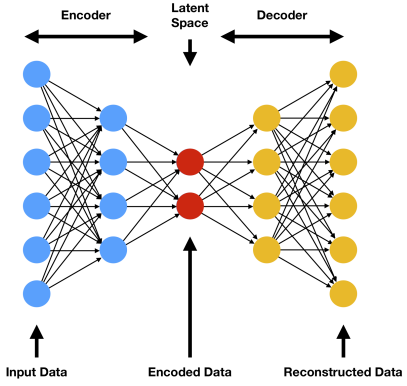


Figure: The Autoencoder reconstructs the input data.

Principle Component Analysis Results

- Inputs: [GPS_lat, GPS_lon, v_lat, v_lon, a_lat, a_lon]
- Num_Components = 3.
- Patterns?

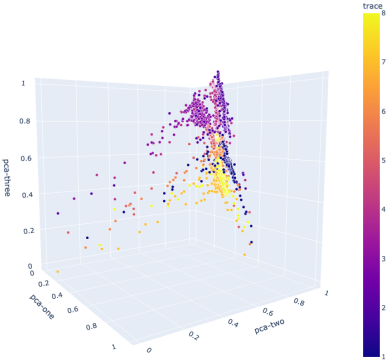


Figure: Visualization of PCA components

One-Class SVM Results

Training on traces, tested on each trace

- Grid search on optimal ν and γ of OCSVM.
- Indiscriminately label all points in the test set as malicious.

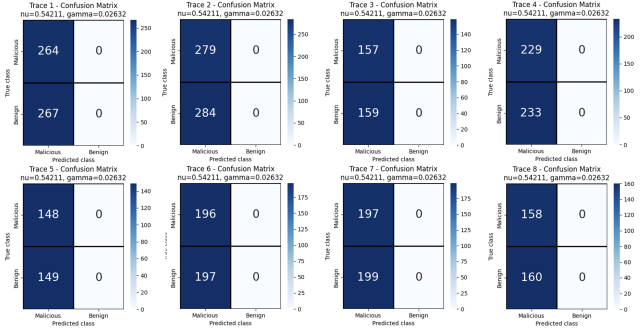


Figure: One-Class SVM results: confusion matrices of each trace

Local Outlier Factors Results

Trained on traces, tested on each trace

- Tuned for the optimal *numneighbors*.
- Was originally the worst method in (Whelan et al. 2020) after all.
- Indiscriminately labels all points in the test set as malicious.

(Whelan et al. 2020) included 85 GNSS-related and sensory data in their work.

OCSVM & LOF do not seem to generalize well on our dataset with limited status data.

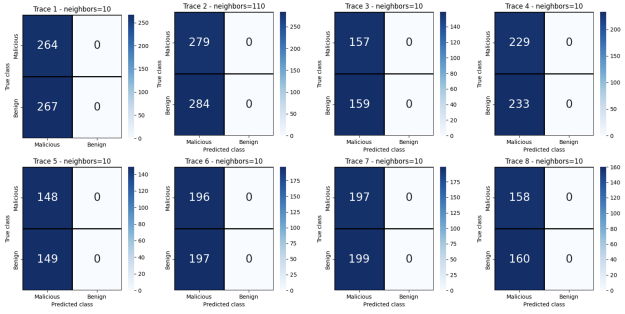


Figure: LOF results: confusion matrices of each trace

Autoencoder Results

Self-defined Autoencoder Model

- model inputs = model ouputs
- Train & validation set: The benign traces
- Test set for detection: The spoofed traces
- Hyperparameter Search
- Spoofed Points: MSE between outputs and inputs $>$ predefined error threshold T

Trial Status

Trial name	status	loc	batch_size	hidden_size	lr	iter	total time (s)	train_loss	val_loss
ray_trainer_03421_00000	TERMINATED	127.0.0.1:90297	32	3	0.001	1	7.72918	0.0431053	0.00729356
ray_trainer_03421_00001	TERMINATED	127.0.0.1:90298	64	3	0.001	1	5.40219	0.0255304	0.00431151
ray_trainer_03421_00002	TERMINATED	127.0.0.1:90299	32	6	0.001	1	7.13482	0.00271494	0.000601879
ray_trainer_03421_00003	TERMINATED	127.0.0.1:90300	64	6	0.001	1	5.54579	0.0288074	0.0050014
ray_trainer_03421_00004	TERMINATED	127.0.0.1:90301	32	3	0.01	1	6.60305	0.151156	0.0251301
ray_trainer_03421_00005	TERMINATED	127.0.0.1:90302	64	3	0.01	1	5.13037	0.0169919	0.00292043
ray_trainer_03421_00006	TERMINATED	127.0.0.1:90303	32	6	0.01	1	7.17763	0.0272919	0.00560031
ray_trainer_03421_00007	TERMINATED	127.0.0.1:90304	64	6	0.01	1	5.64834	0.382157	0.0792823
ray_trainer_03421_00008	TERMINATED	127.0.0.1:90350	32	3	0.1	1	4.58942	0.255833	0.0367685
ray_trainer_03421_00009	TERMINATED	127.0.0.1:90351	64	3	0.1	1	3.30151	0.379265	0.0778871
ray_trainer_03421_00010	TERMINATED	127.0.0.1:90352	32	6	0.1	1	4.5769	0.109397	0.0197297
ray_trainer_03421_00011	TERMINATED	127.0.0.1:90353	64	6	0.1	1	3.32062	0.382139	0.0777353

Figure: Parameter Search on Autoencoder

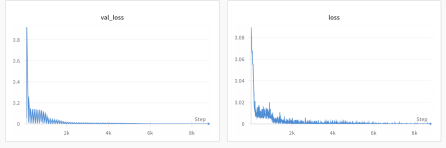


Figure: Model Convergence

Autoencoder Results Cont'd

Threshold T Tuning

- Tuning Objective.
  - $\arg\min_T F_1$ score
- Whelan et al. did not disclose how to choose the Threshold T.
- Choose the T that achieves highest avg F1 score?
- Should we tune on test set?
- Universal Threshold vs. One Threshold for each known trace?

Still not ideal F1 score, compared with what Whelan et al. obtained in their work (94.81% on average).

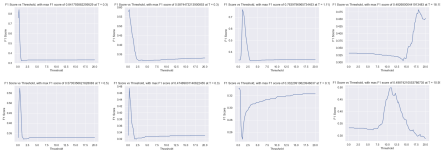


Figure: Tuning Threshold T w.r.t F1 score on each trace

Figure: Average of F1 score on all traces

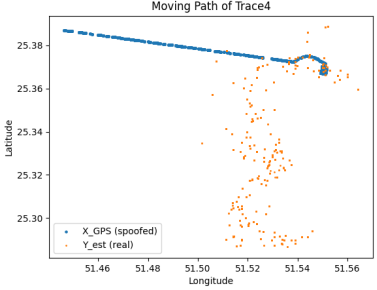
Comparison of Three One-Class Classifiers

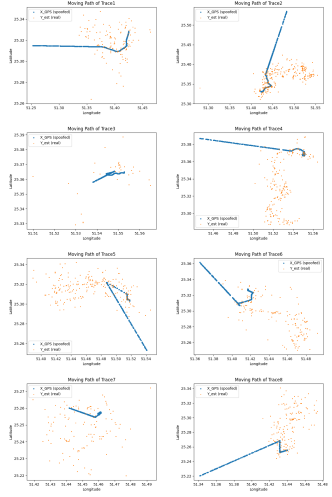
	label	precision	recall	F1 score	Hyperparameters
OC-SVM	benign	n/a	n/a	n/a	nu = 0.5421, gamma = 0.0263
	malicious	0.4968	1.0000	0.6638	
	macro avg	0.2484	0.5000	0.3319	
LOF	benign	n/a	n/a	n/a	numneighbor varies per test trace
	malicious	0.4968	1.0000	0.6638	
	macro avg	0.2484	0.5000	0.3319	
Autoencoder	macro avg	0.6428	0.6050	0.4808	tuned via paramter search

Two Methods from Garrett & Gerdes

Datasets

- The same as that of GSM-based approach.
  Y_est: position estimated by mobile cellular network.
  X_GPS: position received by the GPS (real signal and our generated spoof signal).
- Detect the errors (haversine distance) between the estimated and the actual position in statistic methods.



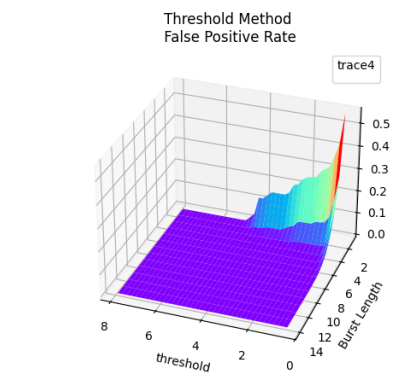


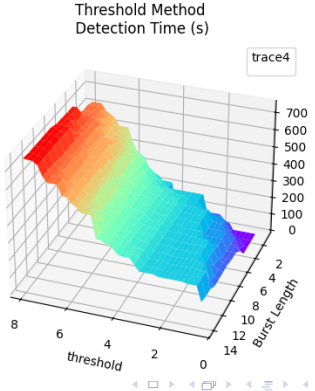
Threshold Method (Garrett & Gerdes, 2020)

- Idea: Alarm when the error exceeds the threshold continuously for more than burst length.
  - r_t: error (haversine distance).
  - τ_t: threshold.
- Tradeoff between FP rate and detection time.

$$A(t) = \begin{cases} 1, & \text{if } |r_t| > \tau_t \\ 0, & \text{if } |r_t| \leq \tau_t \end{cases}$$

Figure: Formula of Threshold Method



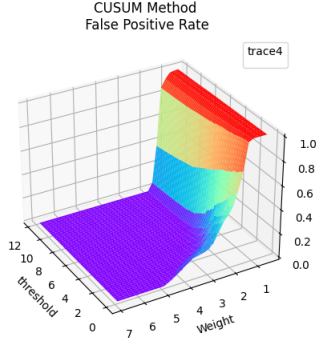


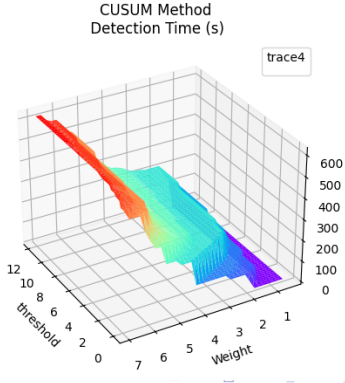
CUSUM Method (Garrett & Gerdes, 2020)

- Idea: Accumulate the error with a weight subtracted in each round. Once the accumulated error exceeds the threshold, trigger an alarm.
  - S_t: accumulated error. τ_t: threshold.
  - r_t: error. b_t: weight subtracted in each round.
- Tradeoff between FP rate and detection time.

$$A(t) = \begin{cases} 1, & \text{if } S_{t-1} > \tau_t \\ S_t = \max(0,\ S_{t-1} + |r_t| - b_t), & \text{if } S_{t-1} \le \tau_t \end{cases}$$

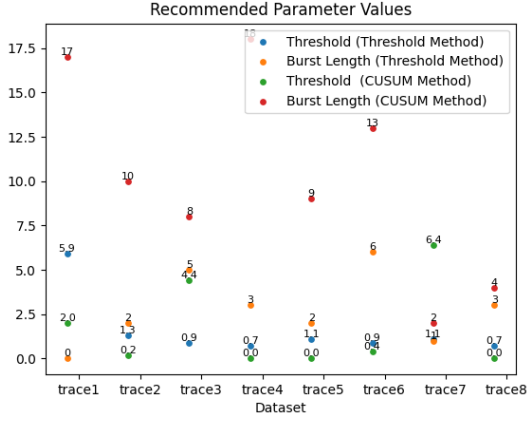
Figure: Formula of CUSUM Method

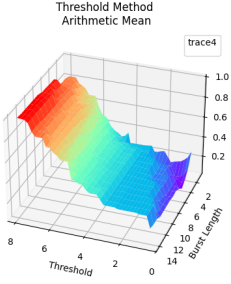


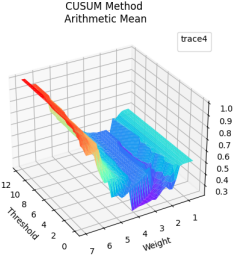


Tradeoff

- Idea: Compute the arithmetic mean of FP rate and (normalized) detection time. Select the minimum point.

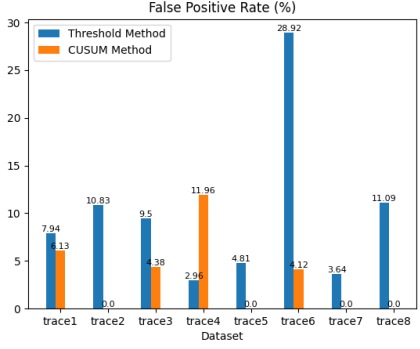


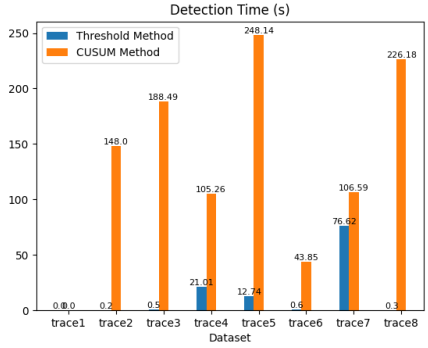




Result and Discussion

- Test other datasets using the recommended parameter values on trace4.
- Tradeoff between FP rate and detection time.
- The pattern may change depending on the choice of parameter values.





Compared with Other ML Methods

- Lightweight, fast, not require many data.
- Suitable for small drones.
- Parameters need to be adjusted according to the actual situation.

Comparison of Model Performance

	GSM	PCA+One-class Classifier	Threshold Method (Garret et al.)	CUSUM (Garret et al.)
Performance Metrics				
F1 Score	N/A benign sequences $\gg$ attack sequence	46% (Autoencoder, average of all traces)	N/A (TP is not applicable)	N/A (TP is not applicable)
False Positive Rate	10.5% (on test set)	34.69% (Autotencoder on trace 4)	9.1% (on trace 4)	0.0% (on trace 4)
Detection Rate	8/8 (on all traces)	8/8 (on all traces)	8/8 (on all traces)	8/8 (on all traces)
Detection Time (Quantitative)	98.330s on train dataset	0.0437s (trace 4)	0.3s (on trace 4)	173.2s (on trace 4)
Detection Time (Qualitative)	Relatively Slow	Very fast	Very fast	Slow

*

Comparison of Model Algorithms

	GSM	PCA+One-class Classifier
Algorithm-related Aspects		
Unit of Model Prediction	Sequence	Single points in one trace
Unit of Detection	Trace, FP rate = P(spoofed)	Single points in one trace
Deviation moment detecting	Can detect	Can detect under more context
Scoring measures	Prediction anomaly sequence in one trace	Predictions of points in one trace
Cross-validation	Applicable	Applicable
Threshold Tuning	Needed for evaluating the anomaly sequence	Needed for Autoencoder Not needed for LOF/OCSVM

	Threshold Method (Garret et al.)	CUSUM (Garret et al.)
Algorithm-related Aspects		
Unit of Model Prediction	Single points in one trace	Single points in one trace
Unit of Detection	Single points in one trace but detection rate is based on trace unit)	Single points in one trace but detection rate is based on trace unit
Deviation moment detecting	Can detect	Can detect
Scoring measures	Predictions of points in one trace Not applicable for confusion matrix	Predictions of points in one trace Not applicable for confusion matrix
Cross-validation	Applicable	Applicable
Threshold Tuning	The drone needs to be tested in advance to obtain suitable parameters.	The drone needs to be tested in advance to obtain suitable parameters.

Comparison of Scenario Settings

	GSM	PCA+One-class Classifier	Threshold Method (Garret et al.)	CUSUM (Garret et al.)
Scenario-related Aspects				
Regularity of Flight path	Not Required	Required	Required	Required
Fixed-route	Not Required	Required	Not Required	Not Required
Pretraining	Required	Required	Required	Required
Detection Effeciency	High	Very high	Very high	High (worse than Garrett's Threshold Method)
Training Time	very fast, O(mn) with n time points, m base stations	Converges very Fast for Autoencoder	Fast	Fast
Comments	Need GSM or Wifi signal together with GPS			

Why are the results not ideal?

The experiment results are not very ideal.
What could possibly have gone wrong?

Reflecting on the Selected Methods

GSM-Based Approach (Oligeri et al. 2019)

- **RSS** The Exponential Distribution of RSS may not be the best model .
- **Duration threshold** The duration threshold used to determine attack may be overfitting.
- **Spoofed trace** The generated spoofed path may not interact with real GPS signal.

Novelty-Based Approach: PCA + One-Class Classifiers (Whelan et al. 2020)

- **Point-wise detection** when do we mark the starting point of spoofing? what if there are mixed ordered occurence of benign and malicoius label predictions?
- **Threshold tuning** Should we use test set to tune T?

Model-Based Method (Garrett & Gerdes, 2020)

Cell sites Cell sites are sparse in rural areas, which can lead to poor estimates of the drone's location.

Tradeoff Hard to tradeoff and set the optimal parameters. A configuration optimal on one path may perform poorly on another path.

Reflecting on the Experiment Dataset

Dataset Availability Issue

Generated sensor data

Lack of necessary IMU sensor data for novelty based methods.

- Drastic data generation and transformation.
- Not good enough sensor data (obtained by differentiation). Not enough variety.

Generated spoofed track

- Identifiable Spoofed Pattern Generated spoofed traces compromise the robustness of our experiments. (Moving in a straight line simply screams "I'm spoofed")
- Simplification of Scenario The attacker will usually monitor our position and speed to launch **a more plausible spoofing attack**.

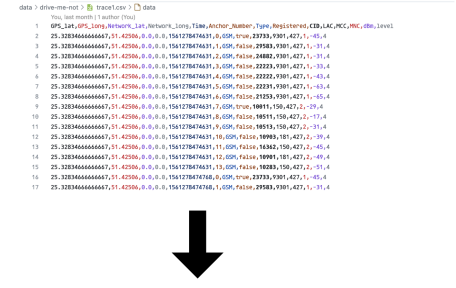


Figure: Drastic data generation and transformation

Future Improvements

- **Comprehensive Dataset for Comparison (Research Question of Group 5)**
  Comprehensive datasets that could be applied to most papers to compare their performance

Q & A