Anomaly Detection Against GPS Spoofing Attacks on Connected and Autonomous Vehicles Using Learning from Demonstration

Zhen Yang, Jun Ying, Junjie Shen, Yiheng Feng, Qi Alfred Chen, Z. Morley Mao, and Henry X. Liu
1. Dept. of Civil & Environmental Engineering, University of Michigan; 2. Lyles School of Civil Engineering, Purdue University
3. Department of Computer Science, University of California, Irvine 4. Department of Electrical Engineering and Computer Science, University of Michigan

Introduction

- In both autonomous Vehicles (AVs) and connected vehicles (CVs), the localization module, which provides accurate local and global positions, plays a critical role in vehicle navigation and information sharing.
- GPS spoofing attacks pose great challenges to safety applications of connected vehicles (CVs) and localization of autonomous vehicles (AVs).
- This study proposes a generic detection framework to detect anomalies in the localization module of CV/AV using learning from demonstration.

Anomaly Detection Concept

Anomaly Detection Framework

- The anomaly detection framework consists of two steps: offline learning and online detection.
- Learning from demonstration is applied to learn the normal driving policy via maximum entropy inverse reinforcement learning using historical trajectories.
- An anomaly classifier (i.e., a decision tree) is trained with both historical trajectories and known attack trajectories.
- Observed trajectories are compared with predicted optimal trajectories from the learned driving policy to detect anomaly.

Learn from Demonstration

\[
\begin{align*}
\text{minimize} & \quad \theta^T f(s, u) \\
\text{s.t.} & \quad \text{Vehicle dynamic constraints}
\end{align*}
\]

- Feature \( f \) includes different driving objectives.
- Vehicle dynamic constraints represent the kinematics of vehicle motion, assuming the vehicle follows the bicycle model.
- The weight vector \( \theta \) is learned via inverse reinforcement learning.

Maximum entropy inverse reinforcement learning algorithm:

- Compute the empirical feature vector over all demonstrations \( \bar{f}_0 = \frac{1}{m} \sum_{j \in D} f(s_j, u_j) \).
- Normalize the feature, denoted as \( \bar{f} \).
- Initialize every entry of the weight vector \( \theta \) with 1.
- While \( \frac{1}{m} \sum_{j=1}^{m} f(s_j^\theta, u_j) - \bar{f} > \text{threshold} \) do
  - For each demonstrated trajectory collected in the dataset
    - Fix the initial condition and the environment states and optimize the trajectory. The optimized trajectories are denoted as \( \{s_1^\theta, \ldots, s_m^\theta\} \).

  The gradient can be calculated as \( \nabla G_\theta(\theta) = \frac{1}{m} \sum_{j=1}^{m} f(s_j^\theta, u_j) - \bar{f} \). Update the parameter vector: \( \theta(k + 1) = \theta(k) + \gamma \nabla G_\theta(\theta) \), in which \( \gamma \) is the learning rate.

Decision Tree Classifier

- Objective ratio, normality score and average displacement error are taken as classification features.
- Objective ratio: \( OR = \max \frac{OR_t}{\sum_{t=1}^{T} \text{objective mean}_{t} \cdot \text{objective std}_{t}} \)
- Normality score: \( NS = \max \frac{NS_t}{\sum_{t=1}^{T} \text{normality score}_{t} \cdot \text{normality std}_{t}} \)
- Displacement error: \( ED = \max \frac{ED_t}{\sum_{t=1}^{T} \text{displacement error}_{t} \cdot \text{displacement std}_{t}} \)

Experiment Results

- The anomaly detection algorithm is validated on a Multi-Sensor Fusion attack with the KAIST dataset and Forward Collision Warning (FCW) attack on NGSIM dataset.

Performance of online detection on AV/CV threat model

- AV threat model: 94% (47/50) trajectories can be identified no later than the success time of the attack.
- CV threat model: 96% (81/84) trajectories can be identified no later than the success time of the attack.