ADVANCED MACHINE LEARNING
FOR ANOMALY DETECTION AND
JAMMER LOCALIZATION

Professor Laura Ruotsalainen,
Computer Science, University of Helsinki
Finnish Center for Artificial Intelligence FCAI
RESEARCH GOALS

• Resilience and security of geospatial data for critical infrastructures (REASON)
  • Academy of Finland 2020 – 2023, with FGI, VTT
• In REASON UH’s SDA group will develop
  • GNSS Fault Detection and Diagnosis system based on Long-Short Term Memory (LSTM) deep learning models for **anomaly detection**
  • Machine learning model for **localizing jammers**
• Long Short-Term Memory network
  • Recurrent neural network capable of learning long sequence prediction problems

• Autoencoders are neural networks that can compress and reconstruct data
• Reconstruction error can be used to identify anomalies
First unsupervised LSTM based autoencoder for GNSS anomaly detection

First fully complex-valued variant from the detector
RESULTS WITH SIMULATED DATA

Accuracy 75%

@Outi Savolainen
VERIFICATION WITH REAL WORD DATA (JAMMING)

Accuracy 99.8%

Next step: classification of the detected anomalies

@Outi Savolainen
Jammer localization – setup

Measurement
- Carrier-to-noise ratio (C/N0) +
  Automatic gain control (AGC)

Multipath environment
- City model + ray-tracing

Localization method
- Raw classification + fine searching

@Zhe Yan
Multipath Simulation Settings

- An urban area about 0.5 km²
- 9 monitoring nodes, 2 m above the roofs
- $5 \times 9$ blocks with $60 \times 60$ m
- 1500 samples in each block
- 3 GPS satellites’ $C/N0 + 1$ front-end AGC
- 45 blocks $\times$ 1500 samples $\times$ 4 features

Ray tracing

- Maximum reflections: 5
- Maximum relative pass loss with the first path: 40 dB (otherwise discard it)
- Materials of the building and terrain: concrete

Description of the ray-tracing paths between the jammer and monitors in Sello shopping center area, Espoo, Finland.
Localization method (Raw classification)

First step: the raw localization is described as a classification problem

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cubic SVM</td>
<td>71.9%</td>
</tr>
<tr>
<td>Fine Gaussian SVM</td>
<td>70.1%</td>
</tr>
<tr>
<td>Fine KNN</td>
<td>70.2%</td>
</tr>
<tr>
<td>Weighted KNN</td>
<td>70.8%</td>
</tr>
<tr>
<td>Subspace KNN</td>
<td>78.0%</td>
</tr>
<tr>
<td>Wide Neural Network</td>
<td>70.2%</td>
</tr>
<tr>
<td>Bagged Trees</td>
<td>77.1%</td>
</tr>
</tbody>
</table>

Traditional supervised machine learning methods

@Zhe Yan
Localization method (Fine searching)

- Second step: optimization method is used in the finer searching within the block

Objective:
Minimize \[ \sum C/N_0 \text{(Optional jammer location)} - \sum C/N_0 \text{(Real jammer location)} \]

Amount of the stations

Problem: common optimization method cannot be used because the cost function value is given by ray-tracing simulation, but the mathematic expression of the cost function cannot be given.

Solution: Gravitational Search Algorithm (GSA), no cost function expression is needed

@Zhe Yan
Localization method (Fine searching)

Basic idea of Gravitational Search Algorithm (GSA)

- Optional location points are assigned with different mass according to their fitness (value of the cost function)
- By the forces among the optional points, they are attracted to move towards the best solution.

Optional location points are assigned with different mass according to their fitness (value of the cost function).

By the forces among the optional points, they are attracted to move towards the best solution.

Newton’s law on universal gravitation

\[ F_g(t) = G(t) \frac{M_i(t) \times M_j(t)}{R_{ij}(t) + \varepsilon} \]

From the equation on the previous slide

\[ m_i(t) = \frac{fit_i(t) - \text{worst}(t)}{\text{best}(t) - \text{worst}(t)} \]

Localization method (Fine searching)

Examples of the searching process of GSA

Limited accuracy due to closest point not having the lowest C/N0

@Zhe Yan
Localization method (Fine searching)

The other reason that we can only obtain a limited accuracy

While getting close to the jammer, the C/N0 becomes unreliable and the AGC saturates

@Zhe Yan
# Test results

- Randomly generate $5 \times 9 \times 20 = 900$ jamming points

<table>
<thead>
<tr>
<th>Method</th>
<th>Fixed rate</th>
<th>Successful rate (&lt;60m)</th>
<th>Average latitude error</th>
<th>STD of latitude error</th>
<th>Average longitude error</th>
<th>STD of longitude error</th>
<th>Average horizontal error</th>
<th>STD of horizontal error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classification + GSA</td>
<td>100%</td>
<td>78.0%</td>
<td>-0.28 m</td>
<td>22.00 m</td>
<td>0.56 m</td>
<td>18.97 m</td>
<td>24.55 m</td>
<td>15.51 m</td>
</tr>
<tr>
<td>Pathloss model + Least squares</td>
<td>20.7%</td>
<td>3.2%</td>
<td>6.19 m</td>
<td>181.06 m</td>
<td>84.56 m</td>
<td>168.59 m</td>
<td>214.52 m</td>
<td>148.87 m</td>
</tr>
</tbody>
</table>

**Benchmark**

- Effectively jammed station < 3 or cannot converge accurately enough (2D position + 1 public error)

- Average C/N gap: 3.15 dB-Hz
- STD of C/N gap: 2.02 dB-Hz

Break through the limitation of effective jamming zone
THANK YOU!

laura.ruotsalainen@helsinki.fi
www.fcai.fi