Sensitivity enhanced method for fault detection and prediction of elevator doors using a margin maximized hyperspace

Minjae Kim*, Seho Son, and Ki-Yong Oh
*Presenter
Applied Dynamics and Intelligent Prognosis Laboratory
Department of Mechanical Convergence Engineering
Hanyang University
Motivation & background

Increasing number of elevators in urban area

Apartments in Seoul

Necessity for effective elevator door fault detection and diagnosis (FDD) methods

“8,256 accidents reported in 2019, which is equal to 22 reports per day”

Effective FDD methods for elevator doors are required
Motivation & background

Existing FDD methods for solving elevator door FDD problems

Complex structure of elevator
Liang et al. 18th CASE IEEE (2022)

Physical sensor-based methods

ML methods with operational data
Zhang et al. 27th ICAC IEEE, (2022)

Challenges for solving real-world FDD problems

“Over 1000:1”
Extremely imbalanced between normal and faulty state
Impossible to define RUL of complex system

Focus on ‘Each component’
Need to define ‘Lifetime’

Fast
Needless of additional sensors

Absence
Impossible to detect degrading sign on binary dataset
→ Impossible to detect failure in advance

Effective solution for real-world FDD problem is required
Motivation & background

Motivation

- The number of elevators in urban area containing lots of buildings is increasing fast.
- The elevator faults, especially for doors frequently occur.
- Accurate but robust FDD methods are required.

Challenges

- Extreme imbalance between normal and scarce fault data.
- Impossible to define RUL of the complex system.
- Impossible to detect degrading sign on binary dataset only containing normal and faulty data.

Research goal

- Develop accurate FDD method for solving highly imbalanced real-world dataset.
- Define RUL focusing on each component instead of the entire system.
- Predict faults even in the absence of degrading data.
Methodology

Overall framework

Phase A. Preprocessing
A.1. Data separation
- Method: Using boolean signal
- Process: Divide each data based on their health condition
A.2. Data interpolation
- Method: Linear and nearest interpolation
- Process: Skip data except gathered with 8~10Hz and interpolate into 10Hz
A.3. Statistic feature extraction
- Method: Peak, mean, RMS
- Process: Extract statistic features and use the data instead raw data to decrease learning difficulty

Phase B. Model construction
B.1. High-level feature extraction
- Method: VAE
- Process: Compress input data into 2D latent space using VAE
B.2. Model optimization
- Method: Bayesian optimization
- Process: Converge the VAE model and find the MMH

Phase C. Application
Fault Detection & Prediction
- Method: SVM
- Process: Classify current states and predict RUL based on marginal distance
Methodology

Phase A. Preprocessing

- **Raw data**
- **Original features**
  - Door location
  - Reference speed
  - Feedback speed
  - Reference torque
  - Feedback torque
  - Differential speed
  - Differential torque

**Time series signals**
- Open strokes
  - Manipulated features
- Close strokes

**Data interpolation**
- Linear interpolation
- Unifying frequency into 10Hz

**Statistical feature extraction**
- Mean
- RMS
- Peak

**Total number of features**: 21

**Knowledge-based feature manipulation**

**Normal**
- **Degrading or Faulty**
  - **High load**

**Motor torque**

**Difference → Manipulated into features**

**Door speed**

**Reference torque**

**Feedback torque**

**Reference speed**

**Feedback speed**
Methodology

Phase B. Model construction

- Margin Maximized Hyperspace (MMH)

  - Normal cluster $→ 0$
    - (To satisfy mean)
  - Faulty cluster $→ 1$
    - (To satisfy variance)

  - Separate normal/faulty cluster

- Bayesian Optimization for enhanced stability

Variational Auto Encoder (VAE)

- Regulate latent space

Maximize sensitivity separating normal and faulty clusters with VAE
Methodology

Phase C. Application of MMH

- Fault detection and prediction using MMH

**Margin Maximized Hyperspace**

Fault detection using SVM

- Faulty cluster
  - RUL: 0.0
- Normal cluster
  - RUL: 1.0

Fault prediction using distance

Application of MMH for fault detection and prediction
Experiment

Elevator door operating dataset

- Data acquisition

List of features used for training and testing

<table>
<thead>
<tr>
<th>Feature number</th>
<th>Feature name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-3</td>
<td>Door location (Peak, mean, RMS)</td>
</tr>
<tr>
<td>4-6</td>
<td>Reference speed (Peak, mean, RMS)</td>
</tr>
<tr>
<td>7-9</td>
<td>Feedback speed (Peak, mean, RMS)</td>
</tr>
<tr>
<td>10-12</td>
<td>Feedback torque (Peak, mean, RMS)</td>
</tr>
<tr>
<td>13-15</td>
<td>Reference torque (Peak, mean, RMS)</td>
</tr>
<tr>
<td>16-18</td>
<td>Feedback torque (Peak, mean, RMS)</td>
</tr>
<tr>
<td>19-21</td>
<td>Differential torque (Peak, mean, RMS)</td>
</tr>
</tbody>
</table>

- 21 features in total
### Elevator door operating dataset

#### Time dependency of elevator door motor health state

- **Open strokes**
  - Defined as normal
  - Demonstrates time delay and huge load

- **Close strokes**
  - Defined as normal
  - Demonstrates time delay and huge load

<table>
<thead>
<tr>
<th>Health state</th>
<th>Number of strokes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>22605</td>
</tr>
<tr>
<td>Degradation</td>
<td>32</td>
</tr>
<tr>
<td>Faulty</td>
<td>36</td>
</tr>
<tr>
<td>Total</td>
<td>22673</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Health state</th>
<th>Number of strokes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>22646</td>
</tr>
<tr>
<td>Degradation</td>
<td>0</td>
</tr>
<tr>
<td>Faulty</td>
<td>31</td>
</tr>
<tr>
<td>Total</td>
<td>22677</td>
</tr>
</tbody>
</table>
Experiment

Reason for selecting open strokes for validation

- Clear indicator of door motor failure
  - Safety issue → Purely opened by motor **torque**, mostly closed by **inertia**
  - Health state of door motor → Less clear in close strokes (**Degradation strokes X**)

- Redundancy for discussing both strokes

Only used **open stroke dataset** for validation
Result & Discussion

Effect of the latent space regulation

VAE w/o regulation

- Locational inconstancy
- Low cohesiveness

VAE w. regulation

- Locational constancy
- High cohesiveness

Variance of normal/faulty clusters

<table>
<thead>
<tr>
<th></th>
<th>axis 1</th>
<th>axis 2</th>
<th>Normal cluster</th>
<th>Faulty cluster</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAE w regulation</td>
<td>10.9</td>
<td>35.55</td>
<td>3.68E-24</td>
<td>1.05E-29</td>
</tr>
<tr>
<td>VAE w/o regulation</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Comparison with other types of autoencoders

AE w. regulation

VAE w. regulation

Distance between normal/faulty clusters

<table>
<thead>
<tr>
<th></th>
<th>Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>AE</td>
<td>0.4359</td>
</tr>
<tr>
<td>VAE</td>
<td>1.4142</td>
</tr>
</tbody>
</table>

Demonstrates locational constancy and high cohesiveness
Validation for fault detection

\[
NPV(\text{Negative Predicted Value}) = \frac{TN}{TN + FN}
\]

Result & Discussion

Fault detection

<table>
<thead>
<tr>
<th>Predicted</th>
<th>Actual</th>
<th>Positive</th>
<th>Negative</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Positive</td>
<td>True positive (TP)</td>
<td>False negative (FN)</td>
</tr>
<tr>
<td></td>
<td>Negative</td>
<td>False positive (FP)</td>
<td>True negative (TN)</td>
</tr>
</tbody>
</table>

Demonstrates high accuracy for fault detection
Result & Discussion

Validation for fault prediction

- Fault prediction using degrading strokes

**Result & Discussion**

- Fault prediction using degrading strokes is effective for fault prediction under absence of degrading data.
Conclusion & Future work

Conclusion

- **MMH (Margin-Maximized Hyperspace)** method is effective at detecting and predicting faults in highly-imbalanced dataset
- This method **maximizes sensitivity** separating two imbalanced clusters and shows locational constancy at latent space
- **Knowledge-based feature manipulation** improves accuracy, so that the method is effective at detecting faults
- **Distance-based RUL estimator** effectively detect potential faults and can quantitively predict RULs even without degrading data

Future work

- **Validation** of the method with elevators at other locations
- **Embed** the method for real-time FDD of operating elevators
Thank you