AI-based Anomaly Detection for Technical Systems

Andrey Morozov and Sheng Ding
ASME TEC Talk, 19.07.2022
Outline

- Part 2: Anomalies and anomaly detection methods.
- Part 3: Example of a DL-based anomaly detector (Kraken).
- Part 4: Challenges and solutions.
Part 1

Faults, errors, failures
Faults, errors, failures
Examples of safety-critical systems

Medical exo-skeleton
Flexible production line
Intelligent transportation
Faults, errors, failures

Networked heterogeneous components

Actuator
Sensor
Controller
Network
Faults, errors, failures
Examples of internal faults

Sensor faults:
- Noise, drift, freeze, offset

Network faults:
- Packet loss, delay, jitter

Computing HW faults:
- Bit-flip, stuck-at

Timing errors

Failure:
- e.g. hazardous actuator command

(a) Sensor faults

- Bias
- Drift
- Freeze
- Noise
Faults, errors, failures
Examples of internal faults

(b) Computing hardware faults

(c) Network faults

Sensor faults:
Noise, drift, freeze, offset

Computing HW faults:
Bit-flip, stuck-at

Timing errors

Network faults:
Packet loss, delay, jitter.

Failure:
e.g. hazardous actuator command

Bit flip

0 1 0 1 1 1 0 1
Faults, errors, failures
Examples of external faults

Sensor faults:
- Noise
- Drift
- Freeze
- Offset

Network faults:
- Packet loss
- Delay
- Jitter

Computing HW faults:
- Bit-flip
- Stuck-at

Timing errors

Failure:
- e.g. hazardous actuator command

(d) Hacker attacks

(e) Environmental conditions

https://innovate.ieee.org/innovation-spotlight/vehicle-detection/
Faults, errors, failures
System states

[Diagram showing system states with nodes labeled S, C, A, and N representing Sensor, Controller, Actuator, and Network, respectively. The diagram illustrates the flow of operation time from nominal to various states and back, with transitions indicated by arrows.]
Faults, errors, failures

System states

Nominal system operation states.
Faults, errors, failures

System errors

Fault — a defect in the system. Fault activation results in error occurrence.

Sensor faults: Noise, drift, freeze, offset

Network faults: Packet loss, delay, jitter.

Computing HW faults: Bit-flip, stuck-at

Timing errors

Error

Nominal

Operation time
Faults, errors, failures

Error propagation

- **Error** – incorrect internal system state. System is still functioning, possibly in degraded mode.
- **Error propagation** – one error leads to another error.

**Sensor faults:** Noise, drift, freeze, offset

**Computing HW faults:** Bit-flip, stuck-at

**Timing errors**

**Network faults:** Packet loss, delay, jitter

**Data errors**

**Timing errors**

**Failure:** e.g. hazardous actuator command
An error can cause a failure. Failure – incorrect delivered service, externally visible deviation from system specification.
Faults, errors, failures

Error detection

- Sensor faults: Noise, drift, freeze, offset
- Network faults: Packet loss, delay, jitter
- Computing HW faults: Bit-flip, stuck-at
- Timing errors

Failure:
- e.g. hazardous actuator command

Error detection, failure prevention, mitigation and recovery.
Part 2
Anomalies
Anomalies

What is an anomaly?

- An anomaly is an observation or a sequence of observations which deviates remarkably from the general distribution of data.
- The set of the anomalies form a very small part of the dataset.
Anomalies

Data types

• **Time series** is a series of data points indexed in time order.

• **Temporal** data include time-series, but also data with timestamps of unequal interval.

• **Univariate** data takes only one dimension, e.g., single sensor readings.

• **Multivariate** data contains multiple dimensions, e.g., images or time-series of several sensors.

• **Labelled dataset**: an annotation exists for each element, which determines if it is a normal or anomalous.

Example of temporal (time-series), multivariate, labelled data:
Anomalies
Anomaly classification

Three different types of anomalies exist.

- **Point anomalies**: If a point deviates significantly from the rest of the data.

- **Collective anomalies**: Individual points are not anomalous, but a sequence of points are labelled as an anomaly.

- **Contextual anomalies**: Some points can be normal in a certain context, while detected as anomaly in another context.

https://arxiv.org/abs/2204.01637
Anomalies
Classification of detection methods

- Expert knowledge
- Monotonicity-check
- Rule-based methods
- Range-check

- Model-based methods
- Parameter estimation-based methods (e.g. Recursive Least Squares)
- State estimation-based (e.g. Kalman filters, Luenberger observers)

- Data-driven methods

Diagram:
- Plant
  - y(t)
  - u(t)
- Observer
  - ŷ(t)
- Error weighting
Anomalies
Classification of detection methods

- Expert knowledge
- Model-based methods
- Data-driven methods
  - Statistical methods
    - Statistical process control
    - Multivariate CUnulated SUMs control chart (MCUSUM)
    - Multivariate Exponential Weighted Moving Average (MEWMA)
  - Prediction methods
    - Vector Autoregressive (VAR) method
    - Autoregressive Moving Average Model (ARIMA)
    - Prediction Confidence Interval (PCI)
    - Simple/Double/ Triple Exponential Smoothing (SES, DES, TES)
    - Principal Component Analysis (PCA)
  - Decomposition methods
    - Singular Spectrum Analysis (SSA)
    - Independent Component Analysis (ICA)
  - Similarity-search (e.g., Matrix-Profile (MP))
    - K-Means Clustering
    - Density-Based Spatial Clustering of Applications with Noise (DBSCAN)
    - Local Outlier Factor (LOF)
    - Isolation Forest (IF)
    - One-Class Support Vector Machines (OC-SVM)
    - Extreme Gradient boosting (XGBoost)

University of Stuttgart
https://commons.wikimedia.org/wiki/File:DBSCAN-Gaussian-data.svg
Anomalies
Classification of detection methods

Expert knowledge

Model-based methods

Anomaly detection methods

Statistical methods

Machine Learning methods

Data-driven methods

Deep learning methods

Multiple Layer Perceptron (MLP)

Long Short Term Memory (LSTM) network

Gated Recurrent Unit (GRU)

Residual Neural Network (ResNet)

WaveNet

Deep Unsupervised Anomaly Detection (DeepAnT)

MLP-UAE (univariate)

MLP-MAE (multivariate)

LSTM-AE

Variational AE (VAE)

Transformer for Anomaly Detection in multivariate time series data (TranAD)

Transformers with Association Discrepancy

Multi-variate anomaly detection with GAN (MAD-GAN)

Time series anomaly detector using GAN (TadGAN)
Anomalies
Approaches to DL-based anomaly detection

1) Classification (MLP, CNN):
   • Supervised learning, good performance.
   • Requires sufficient labeled erroneous instances.

2) Prediction (LSTM):
   • Unsupervised learning, labels are not required.
   • On-line localization, and mitigation.

3) Reconstruction (AE):
   • Based on encoder-decoder architecture.
   • Not so efficient.
## Anomalies

Performance of detection methods

<table>
<thead>
<tr>
<th>Paper</th>
<th>Year</th>
<th>Conclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anomaly Detection in Univariate Time-series: A Survey on the State-of-the-Art</td>
<td>2020</td>
<td>- DL are flawed (statistical methods are better than ML and DL)</td>
</tr>
<tr>
<td>Current time series anomaly detection benchmarks are flawed and are creating the illusion of progress.</td>
<td>2021</td>
<td>- DL are flawed (95% published results can’t be trusted, AD can be solved good enough with older methods)</td>
</tr>
<tr>
<td>An Evaluation of Anomaly Detection and Diagnosis in Multivariate Time Series</td>
<td>2021</td>
<td>- UAE is best (fancy DNN design might not work as they promised, trivial NN might be better than them)</td>
</tr>
<tr>
<td>Do Deep Neural Networks Contribute to Multivariate Time Series Anomaly Detection?</td>
<td>2022</td>
<td>+ No fit for all solution (positive evidence that DL do prove real advantage in some circumstances)</td>
</tr>
<tr>
<td>Anomaly Detection in Time Series: A Comprehensive Evaluation</td>
<td>2022</td>
<td>+ No fit to all solution (there is no clear winner, no one-size-fits-all solution)</td>
</tr>
</tbody>
</table>
Anomalies
Performance of detection methods

Deep Learning: e.g. LSTM, Transformer, Autoencoder.


Statistical Approaches: ARIMA-Model, SES/DES/TES.

Expert knowledge: e.g. rules, range-check.

Part 3

Kraken
Kraken
Deep-learning based anomaly detector

Sensor faults:
- Noise
- Drift
- Freeze
- Offset

Network faults:
- Packet loss
- Delay
- Jitter

Computing HW faults:
- Bit-flip
- Stuck-at

Data errors

Timing errors

Failure:
- e.g. hazardous actuator command

Deep learning based autonomous error detector

Actuator
Sensor
Controller
Network
Access point

Operation time

Failure
Nominal

FAIL
STOP

Access point

Actuator
Sensor
Controller
Network
Access point

University of Stuttgart
Kraken
Deep-learning based anomaly detector

1) Multi-Layer Perceptron (MLP)
   • Low performance for time series.

2) Recurrent Neural Network (RNN)
   • Has memory to process sequences of inputs.
   • Can learn temporal dynamic behavior.
   • Fail to capture the context as time steps increase (vanishing gradient problem).

3) Long Short Term Memory (LSTM)
   • Designed to avoid the vanishing gradient problem.
Kraken
Deep-learning based anomaly detector

Workflow
Kraken
Deep-learning based anomaly detector

Workflow
Kraken
Deep-learning based anomaly detector

Stacked LSTM Architecture:
• Layers: two consecutive hidden LSTM layers fully connected (80 units)
• Look-back: 50 steps.
• Look-ahead: 1 steps.
• Dropout: 0.3

Training-parameters:
• Batch size: 70
• Optimizer: Adam
• Epochs: 35 epochs with early stopping.
Kraken
Deep-learning based anomaly detector

Workflow
Kraken
Deep-learning based anomaly detector

Prediction:

- The LSTM network predicts the next value (lookahead $q = 1$)
- based on the previous 50 time steps (lookback $p = 50$).

![Graph showing prediction and residual values with lookback and lookahead parameters]
Kraken
Deep-learning based anomaly detector

Workflow

Method 1: Gaussian distribution based detection
Method 2: Dynamic threshold based detection
Kraken
Deep-learning based anomaly detector

Gaussian distribution based threshold

Dynamic threshold

University of Stuttgart
Kraken
Deep-learning based anomaly detector

Workflow

LSTM hyperparameters

Error-free signal (Training data)

Preprocessing → Training

Trained LSTM

Actual signal with errors (Test data)

Preprocessing → Prediction

Detection

Anomaly indices and scores

Evaluation

Precision and recall

Error labels
Kraken
Deep-learning based anomaly detector

Evaluation

F1: recall and precision equally important
F2: recall twice as important as precision
F0.5: recall half as important as the precision
# Kraken

## Other use cases

<table>
<thead>
<tr>
<th>Simulink Model Description</th>
<th>Collected Data</th>
<th>Detector</th>
<th>Publications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Emulation of manufacturing process with two manipulators sharing a tool:</td>
<td>Time series data</td>
<td>MLP</td>
<td>- KrakenBox: Deep Learning based Error Detector for Industrial Cyber-Physical Systems (IMECE 2021)</td>
</tr>
<tr>
<td>- One robot takes the tool from tool holder A with randomized waypoints and puts it to tool holder B.</td>
<td>Collected from the sensors of the joints</td>
<td>Auto Encoder</td>
<td>- Anomaly Detection for Cyber Physical Systems using Transformers (IMECE 2021)</td>
</tr>
<tr>
<td>- Another robot takes the tool from tool holder B and put it back to tool holder A</td>
<td>Saved as CSV files</td>
<td>Stacked GRU</td>
<td>- Model-Based Error Detection for Industrial Automation Systems Using LSTM Networks (IMRS'2020)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Stacked LSTM</td>
<td>- Deep Learning-based Fehlerdetektor für industrielle Cyber-Physische Systeme (Industrie 4.0 Management)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Transformer</td>
<td>- On-line error detection and mitigation for time-series data of cyber-physical systems using deep learning based methods (EDCC 2019)</td>
</tr>
</tbody>
</table>

**Robotic Collaborative Manipulators**

- Lower-limb 6-DOF assistive exoskeleton system.
- The high-level controller is realized on a single-board computer connected to the joint controllers via the CAN bus using CANopen protocol.
- The system along with the Simulink model is courtesy of KIT, Dr. Ihab Mameev.

### Robotic Exoskeleton

<table>
<thead>
<tr>
<th>Simulink Model Description</th>
<th>Collected Data</th>
<th>Detector</th>
<th>Publications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our simulation allows users to develop automated driving algorithms and assess their safety and performance. With the help of this, the safety of the implemented component or algorithm can be measured on both the vehicle level and the traffic level. We collected data from a scenario built under a simple scenario:</td>
<td>Time series data</td>
<td>Stacked GRU</td>
<td>- Deep Learning-based Error Mitigation for Assistive Exoskeleton with Computational-Resource-Limited Platform and Edge Tensor Processing Unit (IMECE 2021)</td>
</tr>
<tr>
<td>- A front vehicle capable of sharing its position and speed</td>
<td>Representing the signals collected from the joint</td>
<td>Stacked LSTM</td>
<td>- Modal-based Fault Injection Experiments for the Safety Analysis of Exoskeleton System (IMECE 2020)</td>
</tr>
<tr>
<td>- While another vehicle following it using adaptive cruise control system</td>
<td>Saved as CSV files</td>
<td>Transformer</td>
<td></td>
</tr>
</tbody>
</table>

**Autonomous Vehicle System**

**Unmanned Aerial Vehicle**

- Flight Control System
- Multicopter Model
- Sensor Model
- Environment Model

<table>
<thead>
<tr>
<th>Simulink Model Description</th>
<th>Collected Data</th>
<th>Detector</th>
<th>Publications</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Flight Control System</td>
<td>Representing the speed and acceleration of the vehicles</td>
<td>Gradient Boosting</td>
<td></td>
</tr>
<tr>
<td>- Multicopter Model</td>
<td>Transformed through wavelet filter into figures</td>
<td>CNN</td>
<td></td>
</tr>
<tr>
<td>- Sensor Model</td>
<td>Saved as figures in jpg form</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Environment Model</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Other use cases**

- Random Forest
- Bidirectional LSTM
- CNN-LSTM

- IMU Sensor Faults Detection for UAV using Machine Learning (IJARL 2022)
Kraken

Other use cases

Table 7. Performance of Machine Learning and Deep Learning models based on Test Dataset for Accelerometer

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Test Accuracy</th>
<th>F1 Score</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random Forest w/o Feature Engg. (baseline model)</td>
<td>89.0%</td>
<td>87.0%</td>
<td>88.0%</td>
<td>87.0%</td>
</tr>
<tr>
<td>Random Forest with Feature Engg.</td>
<td>98%</td>
<td>98%</td>
<td>97%</td>
<td>99%</td>
</tr>
<tr>
<td>Hybrid CNN-LSTM w/o Feature Engg.</td>
<td>99.22%</td>
<td>99.0%</td>
<td>99.0%</td>
<td>99.0%</td>
</tr>
<tr>
<td>BiLSTM w/o Feature Engg.</td>
<td>95%</td>
<td>94%</td>
<td>95%</td>
<td>94%</td>
</tr>
</tbody>
</table>

Table 8. Performance of Machine Learning and Deep Learning models based on Test Dataset for Gyroscope

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Test Accuracy</th>
<th>F1 Score</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random Forest w/o Feature Engg. (baseline model)</td>
<td>82.0%</td>
<td>82.0%</td>
<td>82.0%</td>
<td>81.0%</td>
</tr>
<tr>
<td>Random Forest with Feature Engg.</td>
<td>97%</td>
<td>96%</td>
<td>96%</td>
<td>97%</td>
</tr>
<tr>
<td>Hybrid CNN-LSTM w/o Feature Engg.</td>
<td>90.0%</td>
<td>90.0%</td>
<td>91.0%</td>
<td>90.0%</td>
</tr>
<tr>
<td>Hybrid CNN-LSTM with Feature Engg.</td>
<td>93.0%</td>
<td>92.0%</td>
<td>92.0%</td>
<td>93.0%</td>
</tr>
<tr>
<td>BiLSTM w/o Feature Engg.</td>
<td>84.0%</td>
<td>83.0%</td>
<td>82.0%</td>
<td>83.0%</td>
</tr>
</tbody>
</table>

Fig. 12. Confusion Matrix of Hybrid CNN-LSTM model based on Test Dataset for Accelerometer.

Fig. 13. Confusion Matrix of Random Forest model based on Test Dataset for Gyroscope.
Kraken

Time Series Anomaly Detection for Simulink

Features:

- Multiple DNN architectures
- Customizable hyper-parameters
- Several detection approaches
- Several evaluation methods
- Multiple fault types
- Multiple fault injection methods

"Tool Paper: Time Series Anomaly Detection Platform for MATLAB Simulink", Accepted to IMBSA 2022

Open source: https://github.com/mbsa-tud/tsad_platform
Part 4

Challenges
Challenges
How to select a suitable anomaly detector?

Context-aware anomaly detector:

- Search for optimal detection approach, deep learning architecture, hyperparameters;
- Combination (Ensemble) of several detectors;
- Dynamic switch of the detectors.
Challenges
How to select a suitable anomaly detector?

Context-aware anomaly detector:

Simulation → Data → Features → Zoo → Performance → Suitable detector

System analysis → Access point features: Expect fault types, dependencies, ...

Data → Features → Classifier → Suitable detector
Challenges
How to select a suitable access points?

Context-aware anomaly detector:

- System-level control flow, data flow, and error propagation analysis
- Dynamic switch according to the attention mechanism

Formal methods for error propagation analysis

Simulative fault injection experiments
Challenges

How to generate training and testing data?

Fault Injection Tool FIBlock for Simulink

The user can specify:

- Fault type: Stuck-at, Package drop, Bias/Offset, Bit flips, Time delay, Noise.
- Fault event: Failure probability, Mean Time to Failure, Failure Rate Distribution.
- Fault effect: Once, Constant time, Infinite time, Mean Time to Repair.

Augmented data = Normal data (real data) + Fault samples (from a database)
Challenges
How to generate training and testing data?

Reinforcement Learning-based Fault Injection
Challenges
Three levels of anomaly detection

System-of-Systems-Level
- Attention switching
- Scaling
- Edge-Fog-Cloud

System-Level
- System analysis
- Dynamic switching access points
- Multivariate time series

Component-Level
- Selection, combination, tuning of DNNs
- Dynamic switching of DNNs
- Univariate time series
Thank you
Vielen Dank!

Jun.-Prof. Dr.-Ing. Andrey Morozov

e-mail andrey.morozov@ias.uni-stuttgart.de
phone +49 (0) 711 685-67312
www.ias.uni-stuttgart.de/en/institute/team/Morozov/

University of Stuttgart
Institute of Industrial Automation and Software Engineering
Networked Automation Systems