Anomaly Detection in the Open World: Normality Shift Detection, Explanation, and Adaptation

Dongqi Han, Zhiliang Wang, Wenqi Chen, Kai Wang, Rui Yu, Su Wang, Han Zhang, Zhihua Wan, Minghui Jin, Jiahai Yang, Xingang Shi, and Xia Yin
Cyber crimes are becoming more professional and coordinated

- Skilled cyber attackers can **bypass** approximately all the defense systems
Cyber crimes are becoming more professional and coordinated

- Skilled cyber attackers can bypass approximately all the defense systems

Anomaly Detection has been widely used in diverse network security applications

- Learning without knowledge of anomalies
- Ability to detect unforeseen threats
Cyber crimes are becoming more professional and coordinated
  • Skilled cyber attackers can bypass approximately all the defense systems
Anomaly Detection has been widely used in diverse network security applications
  • Learning without knowledge of anomalies
  • Ability to detect unforeseen threats
Deep Learning has shown a great potential to build network security applications
  • Learn better nonlinear and hierarchical features
  • Capture complex and high-dimensional structures
Deep Learning based Anomaly Detection

Zero-positive Learning
(trained with only normal data)
Deep Learning based Anomaly Detection

Reconstruction-based
Deep Learning based Anomaly Detection

Reconstruction-based Training
Deep Learning based Anomaly Detection

Reconstruction-based **Training**

Minimize reconstruction error
Deep Learning based Anomaly Detection

Reconstruction-based Detection
Deep Learning based Anomaly Detection

Reconstruction-based Detection

Anomaly!
Deep Learning based Anomaly Detection

Reconstruction-based

Prediction-based
Deep Learning based Anomaly Detection

Reconstruction-based

Prediction-based Training

Anomaly Detection in the Open World: Normality Shift Detection, Explanation, and Adaptation.

Presenter — Dongqi Han
Deep Learning based Anomaly Detection

- **Prediction-based Training**
  - Maximize predictive probability

- **Reconstruction-based**
Deep Learning based Anomaly Detection

Anomaly Detection in the Open World: Normality Shift Detection, Explanation, and Adaptation.
Deep Learning based Anomaly Detection

Zero-positive Learning
(trained with only normal data)

Reconstruction-based

Prediction-based
Anomaly Detection in Security Applications

Security Applications with Deep Learning based Anomaly Detection:

Network Intrusion Detection (NDSS’18, CCS’23)

Log Anomaly Detection (CCS’17, CCS’19)

Lateral Movement Detection (CCS’19, Security’23)

Host-based Threat Detection (NDSS’20, S&P’23)
Close World vs. Open World

• The great success of machine/deep learning methods are based on the Close-world assumption—testing data must be *similar to the training data* (i.i.d. assumption)
Close World vs. Open World

- The great success of machine/deep learning methods are based on the Close-world assumption—**testing data must be similar to the training data** (i.i.d. assumption)

- However, in Open-world applications, the distribution of testing data can **change over time in unforeseen ways**
  - Concept Drift Problem
  - Example in security: the evolution of malware
Close World vs. Open World

• The great success of machine/deep learning methods are based on the Close-world assumption— **testing data must be similar to the training data** (i.i.d. assumption)

• However, in Open-world applications, the distribution of testing data can **change over time in unforeseen ways**
  • Concept Drift Problem
  • Example in security: the evolution of malware
  • Model performance aging!

![Diagram showing the difference between close-world and open-world data distributions](image)
Concept Drift vs. Normality Shift

- **Concept drift** has been well-studied for supervised classification
  - **Security:** Transcend(*Usenix Sec’19*), CADE(*Usenix Sec’21*), Transcendent(*S&P’22*)
Concept Drift vs. Normality Shift

- **Concept drift** has been well-studied for supervised classification
  - **Security:** Transcend(*Usenix Sec’19*), CADE(*Usenix Sec’21*), Transcendent(*S&P’22*)
  - **Machine Learning:** Out-of-distribution (OOD) detection
Concept Drift vs. Normality Shift

- **Concept drift** has been well-studied for supervised classification
  - **Security:** Transcend(*Usenix Sec’19*), CADE(*Usenix Sec’21*), Transcendent(*S&P’22*)
  - **Machine Learning:** Out-of-distribution (OOD) detection

- **Anomaly detection** models are built upon purely normal data (normality)
  - Immune to the drift of malicious/abnormal behavior
  - More severe impact when the distribution of normality shifts
  - E.g., user behaviors and system themselves (patches, new devices)

---

**Supervised Classification**

- Drifting/OOD Sample
- Class 1
- Class 2

**Anomaly Detection**

- Normality
Concept Drift vs. Normality Shift

- **Concept drift** has been well-studied for supervised classification
  - Security: Transcend(Usenix Sec’19), CADE(Usenix Sec’21), Transcendent(S&P’22)
  - Machine Learning: Out-of-distribution (OOD) detection

- **Anomaly detection** models are built upon purely normal data (normality)
  - Immune to the drift of malicious/abnormal behavior
  - More severe impact when the distribution of normality shifts
  - E.g., user behaviors and system themselves (patches, new devices)
Concept Drift vs. Normality Shift

- **Concept drift** has been well-studied for *supervised classification*
  - Security: Transcend(Usenix Sec’19), CADE(Usenix Sec’21), Transcendent(S&P’22)
  - Machine Learning: Out-of-distribution (OOD) detection

- **Anomaly detection** models are built upon purely normal data (normality)
  - Immune to the drift of malicious/abnormal behavior
  - More severe impact when the distribution of *normality shifts*
  - E.g., user behaviors and system themselves (patches, new devices)

---

**Key Insight 1** – *Without ground-truth label, a normality shift and real anomaly is not distinguishable for anomaly detection!*
Pipelines for Handling Shift

Pipeline 1

- Retraining /Ensemble
- Retraining /Ensemble
- Retraining /Ensemble
- Retraining /Ensemble

Heavy Cost
Lack of Analysis
Delay update
Pipelines for Handling Shift

**Pipeline 1**
- Retraining/Ensemble
- Retraining/Ensemble
- Retraining/Ensemble
- Retraining/Ensemble

**Pipeline 2**
- Detection: Yes
- Detection: No
- Detection: Yes
- Detection: No

**Our Scope**
- Heavy Cost
- Lack of Analysis
- Delay update

---

Presenter — Dongqi Han
Pipelines for Handling Shift

Key Insight 2 – We need to decide whether, when, and how shift occurs before adapting models to the shift!
Detecting Shift in Statistics

Question: How to represent the distribution of normality?

Distribution of feature-space data

Original

Shifted

1D
Detecting Shift in Statistics

**Question:** How to represent the distribution of normality?

<table>
<thead>
<tr>
<th>Distribution of feature-space data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
</tr>
<tr>
<td><img src="image1.png" alt="Histogram" /></td>
</tr>
<tr>
<td><img src="image2.png" alt="Density Plot" /></td>
</tr>
<tr>
<td>Shifted</td>
</tr>
<tr>
<td><img src="image3.png" alt="Histogram" /></td>
</tr>
<tr>
<td><img src="image4.png" alt="Density Plot" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1D</td>
</tr>
<tr>
<td>2D</td>
</tr>
<tr>
<td>...</td>
</tr>
<tr>
<td>1000D?</td>
</tr>
</tbody>
</table>
Detecting Shift in Statistics

**Question:** How to represent the distribution of normality?

Distribution of feature-space data

- **Original**: 1D
- **Shifted**: 2D

Intractable for high-dimensional data!
Detecting Shift in Statistics

**Question: How to represent the distribution of normality?**

**Distribution of feature-space data**

- **Original**
  - 1D
  - 2D
  - Distribution of feature-space data

- **Shifted**
  - 1D
  - 2D
  - Distribution of model outputs

**Our Scope**

Intractable for high-dimensional data!
Detecting Shift in Statistics

Question: How to represent the distribution of normality?

Key Insight 3 – Distribution of normality can be represented by the distribution of model outputs!
OWAD Design

- We present **OWAD (Open World Anomaly Detection) Framework**
  - Detecting, Explaining, and Adapting to normality shift for DL-based anomaly detection.
OWLAD Design

- We present **OWLAD (Open World Anomaly Detection) Framework**
  - Detecting, Explaining, and Adapting to normality shift for DL-based anomaly detection.
We present **OWAD** (Open World Anomaly Detection) Framework

- Detecting, Explaining, and Adapting to normality shift for DL-based anomaly detection.
**OWAD Design**

- We present **OWAD (Open World Anomaly Detection) Framework**
  - Detecting, Explaining, and Adapting to normality shift for DL-based anomaly detection.
• We present **OWAD (Open World Anomaly Detection)** Framework
  • Detecting, Explaining, and Adapting to normality shift for DL-based anomaly detection.
OWAD Design

- We present **OWAD (Open World Anomaly Detection)** Framework
  - Detecting, Explaining, and Adapting to normality shift for DL-based anomaly detection.
OWAD Design

- We present **OWAD** (Open World Anomaly Detection) Framework
  - Detecting, Explaining, and Adapting to normality shift for DL-based anomaly detection.
We present **OWAD (Open World Anomaly Detection)** Framework

- Detecting, Explaining, and Adapting to normality shift for DL-based anomaly detection.
• We present OWAD (Open World Anomaly Detection) Framework
  • Detecting, Explaining, and Adapting to normality shift for DL-based anomaly detection.
• We present **OWAD (Open World Anomaly Detection)** Framework
  • Detecting, Explaining, and Adapting to normality shift for DL-based anomaly detection.
We present OWAD (Open World Anomaly Detection) Framework
- Detecting, Explaining, and Adapting to normality shift for DL-based anomaly detection.
OWAD Design

- We present **OWAD (Open World Anomaly Detection) Framework**
  - Detecting, Explaining, and Adapting to normality shift for DL-based anomaly detection.
We present OWAD (Open World Anomaly Detection) Framework

- Detecting, Explaining, and Adapting to normality shift for DL-based anomaly detection.
We present **OWAD (Open World Anomaly Detection)** Framework
- Detecting, Explaining, and Adapting to normality shift for DL-based anomaly detection.
OWAD Design

• We present **OWAD (Open World Anomaly Detection)** Framework
  • Detecting, Explaining, and Adapting to normality shift for DL-based anomaly detection.
OWAD Design

- We present OWAD (Open World Anomaly Detection) Framework
  - Detecting, Explaining, and Adapting to normality shift for DL-based anomaly detection.
We present **OWAD** (Open World Anomaly Detection) Framework
• Detecting, Explaining, and Adapting to normality shift for DL-based anomaly detection.
We present **OWAD** (**O**pen **W**orld **A**nomaly **D**etection) Framework

- Detecting, Explaining, and Adapting to normality shift for DL-based anomaly detection.

**OWAD Design**

- **Data Collection**
  - Before Shift (Old)
  - After Shift (New)

- **Anomaly Detection**
  - DL Models
  - Output (Outputs)

- **Output Calibration**
  - Calibrator
  - Expected Outputs vs. Actual Outputs

- **Shift Detection**
  - Distribution Test
  - After Shift (New)

- **Shift Explanation**
  - Explainer
  - Mapping

- **Shift Adaptation**
  - Weight Consolidation

**Anomaly Detection in the Open World: Normality Shift Detection, Explanation, and Adaptation.**

Presenter — Dongqi Han
OWAD Design

- We present **OWAD** (Open World Anomaly Detection) Framework
  - Detecting, Explaining, and Adapting to normality shift for DL-based anomaly detection.
We present OWAD (Open World Anomaly Detection) Framework
- Detecting, Explaining, and Adapting to normality shift for DL-based anomaly detection.
We present OWAD (Open World Anomaly Detection) Framework
- Detecting, Explaining, and Adapting to normality shift for DL-based anomaly detection.
Model Calibration for Classification

- Transform classifier scores into class membership probabilities
- E.g., given 100 predictions, each with confidence of 0.8, we expect that 80 should be correctly classified.
Step 1 — Output Calibration

**Model Calibration** for Classification
- Transform classifier scores into class membership probabilities
- E.g., given 100 predictions, each with confidence of 0.8, we expect that 80 should be correctly classified.

Calibration for **Anomaly Detection**
- Expected Meaning: the *percentile* of model outputs (also FPR if threshold is itself)
- E.g., Original: [0.7, 0.8, 0.9, 1.0], Calibrated: [0.25, 0.5, 0.75, 1.0]
**Model Calibration** for Classification

- Transform classifier scores into class membership probabilities
- E.g., given 100 predictions, each with confidence of 0.8, we expect that 80 should be correctly classified.

Calibration for **Anomaly Detection**

- Expected Meaning: the *percentile* of model outputs (also FPR if threshold is itself)
- E.g., Original: [0.7, 0.8, 0.9, 1.0], Calibrated: [0.25, 0.5, 0.75, 1.0]

Calibration Function – **Isotonic Regression**

- **Probabilistic** legality: Convert Anomaly Score into [0,1]
- **Monotonicity**: Without affecting detection performance
- **Non-linear**: Linear transformation of distribution is meaningless
We present **OWAD** (Open World Anomaly Detection) Framework

- Detecting, Explaining, and Adapting to normality shift for DL-based anomaly detection.
Step 2 — Shift Detection

• Hypothesis Test
  • \( H_0 \): Two data follow the same distribution (No drift happen)
  • \( H_1 \): Two data do not follow same distribution (drift happens)

Ref: https://towardsdatascience.com/how-to-use-permutation-tests-bacc79f45749
Step 2 — Shift Detection

• Hypothesis Test
  • H0: Two data follow the same distribution (No drift happen)
  • H1: Two data do not follow same distribution (drift happens)

• Permutation Test
  • Pros: Distribution-free, support any test statistic, and suitable for small set
  • Test Statistic: KL divergence of original and shifted distribution
  • P-value: \( \frac{1+\sum_{\delta}^{N}[(KL(P||Q))<\Delta]}{N+1} < \delta \)

Algorithm 1: Procedure for shift detection

1. Getting original discrete distributions (histograms)
   \( P_{\text{org}} \leftarrow \mathcal{H}_{K}(C(f(x^c))); \quad Q_{\text{org}} \leftarrow \mathcal{H}_{K}(C(f(x^t))); \)
2. Original test statistics
   \( s_{\text{org}} \leftarrow D_{KL}(P_{\text{org}}||Q_{\text{org}}); \)
3. Permutating/Resampling and recomputing two histograms (\( \mathcal{H}_{K} \)) from \( \{C(f(x^c))\} \cup \{C(f(x^t))\} \)
4. \( p \leftarrow \frac{1+\sum_{\delta}^{N}1[s_{\text{org}}\leq D_{KL}(P_{\text{org}}||Q_{\text{org}})]}{N_{p}+1}; \) \( p \)-value of test
5. Return \( p \) confidence of non-shift

Ref: https://towardsdatascience.com/how-to-use-permutation-tests-bacc79f45749
OWAD Design

• We present **OWAD (Open World Anomaly Detection)** Framework
  • Detecting, Explaining, and Adapting to normality shift for DL-based anomaly detection.
Step 3 — Shift Explanation

\[
\min_{m_{c\oplus t} = m_c \odot m_t} \mathcal{L} \{ \mathbb{D}[((1 - m_c) \odot p_c) \oplus (m_t \odot p_t)], \mathbb{D}[p_t] \}
\]

\[
+ \lambda_1 \|m_{c\oplus t}\| - \lambda_2 \mathbb{E}_{m \in m_{c\oplus t}} [m \log m + (1 - m) \log (1 - m)]
\]

(\odot: hadamard product, \oplus: vector concatenation)

\[\mathbb{D}[p_c]\]

\[\mathbb{D}[p_t]\]

\[\mathbb{D}[((1 - m_c) \odot p_c) \oplus (m_t \odot p_t)]\]

Old Distribution

New Distribution

Mixed Distribution
Step 3 — Shift Explanation

\[
\min_{m_{c \oplus t} = m_c \odot m_t} \mathcal{L}\{\mathbb{D}[(1-m_c) \odot p_c] \oplus (m_t \odot p_t), \mathbb{D}[p_t]\} \\
+ \lambda_1 ||m_{c \oplus t}|| - \lambda_2 \mathbb{E}_{m \in m_{c \oplus t}} [m \log m + (1-m) \log (1-m)]
\]

\(\odot: \text{hadamard product, } \oplus: \text{vector concatenation}\)

Mixed samples should accurately reconstruct the new distribution

**Old Distribution**

\[\mathbb{D}[p_c]\]

\[p_c\]

**New Distribution**

\[\mathbb{D}[p_t]\]

\[p_t\]

**Mixed Distribution**

\[\mathbb{D}[(1-m_c) \odot p_c] \oplus (m_t \odot p_t)\]

\[((1-m_c) \odot p_c) \oplus (m_t \odot p_t)\]
Step 3 — Shift Explanation

\[
\min_{m_{c\oplus t} = m_c \odot m_t} \mathcal{L}\{\mathbb{D}[((1-m_c) \odot p_c) \oplus (m_t \odot p_t)], \mathbb{D}[p_t]\} \\
+ \lambda_1 ||m_{c\oplus t}||_1 - \lambda_2 \mathbb{E}_{m \in m_{c\oplus t}} [m \log m + (1-m) \log (1-m)]
\]

**Accuracy Loss**

**Overhead Loss**

Choose as few samples from the new distribution as possible

Mixed samples should accurately reconstruct the new distribution

(\odot: hadamard product, \oplus: vector concatenation)
Step 3 — Shift Explanation

Choose as few samples from the new distribution as possible

\[
\min_{m_c \oplus t = m_c \otimes m_t} \mathcal{L} \left( \mathbb{D} \left[ (1 - m_c) \odot p_c \odot (m_t \odot p_t) \right], \mathbb{D} \left[ p_t \right] \right) \\
+ \lambda_1 ||m_c \oplus t|| - \lambda_2 \mathbb{E} \left[ m \log m + (1 - m) \log (1 - m) \right] _{m \in m_c \oplus t}
\]

Mixed samples should accurately reconstruct the new distribution

Expect \( m_c \) or \( m_t \) to be deterministic (either close to 0 or close to 1)

Old Distribution

\( \mathbb{D} \left[ p_c \right] \)

New Distribution

\( \mathbb{D} \left[ p_t \right] \)

Mixed Distribution

\( \mathbb{D} \left[ (1 - m_c) \odot p_c \odot (m_t \odot p_t) \right] \)
Step 3 — Shift Explanation

Choose as few samples from the new distribution as possible

\[ \min_{m_c \oplus t = m_c \odot m_t} \mathcal{L}\left( \mathbb{D}[((1 - m_c) \odot p_c) \oplus (m_t \odot p_t)], \mathbb{D}[p_t]\right) + \lambda_1 \|m_c \oplus t\| - \lambda_2 \mathbb{E}_{m \in m_c \oplus t} [m \log m + (1 - m) \log (1 - m)] \]

Mixed samples should accurately reconstruct the new distribution

Expect \( m_c \) or \( m_t \) to be deterministic (either close to 0 or close to 1)

Old Distribution

\( \mathbb{D}[p_c] \)

New Distribution

\( \mathbb{D}[p_t] \)

Mixed Distribution

\( \mathbb{D}[(1 - m_c) \odot p_c) \oplus (m_t \odot p_t)] \)
OWAD Design

- We present **OWAD (Open World Anomaly Detection) Framework**
  - Detecting, Explaining, and Adapting to normality shift for DL-based anomaly detection.
Step 4 — Shift Adaptation

\[
\min_{\theta^*} \mathcal{L}\left(\mathcal{D}\left[\left((1 - m_c) \odot p_c(\theta^*)\right) \oplus (m_i \odot p_t(\theta^*))\right], \mathcal{D}\left[p_c(\theta)\right]\right)
\]

\[+ \lambda \sum_{i,j} \Omega_{ij} (\theta_{ij} - \theta^*_{ij})^2 \]

where \( \Omega_{ij} = \sum_{p(x) \sim p_c} \left\| \frac{\partial \ell_2^2(F(x; \theta))}{\partial \theta_{ij}} \right\| \cdot m_c(x) \)

(\(\odot\): hadamard product, \(\oplus\): vector concatenation)
Step 4 — Shift Adaptation

\[ \min_{\theta^*} \mathcal{L} \left[ \left( (1 - m_c) \circ p_c(\theta^*) \right) \oplus \left( m_i \circ p_t(\theta^*) \right) \right], \mathbb{D} \left[ p_c(\theta) \right] \]

\[ + \lambda \sum_{i,j} \Omega_{ij} (\theta_{ij} - \theta_{ij}^*)^2 \]

where \( \Omega_{ij} = \sum_{P(x) \sim p_c} \left\| \frac{\partial}{\partial \theta_{ij}} \left\| \frac{\partial^2 \ell^2(F(x; \theta))}{\partial \theta_{ij}} \right\| \cdot m_c(x) \right\| \cdot m_c(x) \)

(\( \circ \) : hadamard product, \( \oplus \) : vector concatenation)

**Old Distribution**

**New Distribution**

**Mixed Distribution**

**Distributional Shift Adaptation**
Step 4 — Shift Adaptation

$$\min_{\theta^*} \mathcal{L}\{((1 - m_c) \odot p_c(\theta^*)) \oplus (m_t \odot p_t(\theta^*))\}, \mathbb{D}[p_c(\theta)]$$

$$+ \lambda \sum_{i,j} \Omega_{ij} (\theta_{ij} - \theta_{ij}^*)^2$$

where

$$\Omega_{ij} = \sum_{p(x) \sim p_c} \left\| \frac{\partial \ell^2_2(F(x; \theta))}{\partial \theta_{ij}} \right\| \cdot m_c(x)$$

(Elastic Weight Consolidation)

Evaluate the importance of model parameters

$$\mathbb{D}[p_c], \mathbb{D}[p_t], \mathbb{D}[((1 - m_c) \odot p_c) \oplus (m_t \odot p_t)]$$

Old Distribution \quad New Distribution \quad Mixed Distribution
OWAD Design

- We present **OWAD (Open World Anomaly Detection) Framework**
  - Detecting, Explaining, and Adapting to normality shift for DL-based anomaly detection.

Please refer to our paper for more details of OWAD!
Evaluation

Network Intrusion

Log Anomaly

Lateral Movement

- Kitsune [NDSS’18]
- DeepLog [CCS’17]
- GL-GV [RAID’20]
Evaluation

Network Intrusion

Log Anomaly

Lateral Movement

- Kitsune [NDSS’18]
- Anoshift Benchmark [NIPS’22]
- honey pot and campus network traffic

- DeepLog [CCS’17]
- BGL Dataset [DSN’07]
- BlueGene/L supercomputer group Logs

- GL-GV [RAID’20]
- LANL-CMSCSE Dataset
- login events from corporate internal network
Evaluation

Network Intrusion
- Kitsune [NDSS’18]
- Anoshift Benchmark [NIPS’22]
- honey pot and campus network traffic
- 10 years
- detect once a year

Log Anomaly
- DeepLog [CCS’17]
- BGL Dataset [DSN’07]
- BlueGene/L supercomputer group Logs
- 7 months
- detect once a month

Lateral Movement
- GL-GV [RAID’20]
- LANL-CMSCSE Dataset
- login events from corporate internal network
- 58 days
- detect once a week
Normality Shift in Security Applications

Anomaly Detection in the Open World: Normality Shift Detection, Explanation, and Adaptation.

Presenter — Dongqi Han
Normality shift in security domain is quite common and different for each applications (case-by-case)
End-to-end Performance Evaluation

- **Data selection and split**
  - Train anomaly detection model with **Training set** at Time 0
  - Detect shift and update model with **Validation set** at Time 1, 2, 3, ..., N
  - Evaluate the model performance with **Testing set** at Time 0, 1, 2, 3, ..., N
End-to-end Performance Evaluation

- **Data selection and split**
  - Train anomaly detection model with *Training set* at Time 0
  - Detect shift and update model with *Validation set* at Time 1, 2, 3, …, N
  - Evaluate the model performance with *Testing set* at Time 0, 1, 2, 3, …, N

- **Experimental setup**
  - *Single Adaptation*: Update model at Time 1, Test mode at Time 2, 3, …
End-to-end Performance Evaluation

- **Data selection and split**
  - Train anomaly detection model with **Training set** at Time 0
  - Detect shift and update model with **Validation set** at Time 1, 2, 3, ..., N
  - Evaluate the model performance with **Testing set** at Time 0, 1, 2, 3, ..., N

- **Experimental setup**
  - **Single Adaptation**: Update model at Time 1, Test mode at Time 2, 3, ...

![Diagram showing the timeline of data selection and evaluation phases from Time 0 (T0) to Time N (TN).]
End-to-end Performance Evaluation

- **Data selection and split**
  - Train anomaly detection model with **Training set** at Time 0
  - Detect shift and update model with **Validation set** at Time 1, 2, 3, ..., N
  - Evaluate the model performance with **Testing set** at Time 0, 1, 2, 3, ..., N

- **Experimental setup**
  - **Single Adaptation**: Update model at Time 1, Test mode at Time 2, 3, ...
  - **Multiple Adaptations**: Update model at Time 1, 2, 3, ..., Test mode at the same time
Performance of Single Adaptation

![Graph showing performance of single adaptation with different methods against various test times.](image)

- **Adapt @T1**
  - Test @T1
  - Test @T2
  - Test @T3
  - Test @T4

- **Adapt @T2**
  - Test @T2
  - Test @T3
  - Test @T4

The graph compares different methods such as No-Update, Retrain, UNLEARN, CADE, TRANSCENDENT, and OWAD (Ours.).
Performance of Single Adaptation

OWAD outperforms other approaches at the adaptation time, and can also mitigate the performance degradation in subsequent time.
Performance of Single Adaptation

OWAD outperforms other approaches at the adaptation time, and can also mitigate the performance degradation in subsequent time.
Performance of Single Adaptation

OWAD outperforms other approaches at the adaptation time, and can also mitigate the performance degradation \textit{in subsequent time}. 
Performance of Multiple Adaptations

- No-Update
- Retrain
- UNLEARN
- CADE
- TRANSCENDENT
- OWAD (Ours.)

Anomaly Detection in the Open World: Normality Shift Detection, Explanation, and Adaptation.
Presenter — Dongqi Han
Performance of Multiple Adaptations

OWAD can achieve better results with significantly less required labels
# Performance of FP/FNs

<table>
<thead>
<tr>
<th>Methods</th>
<th># FPs (Lower is Better)</th>
<th># FNs (Lower is Better)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>@T1</td>
<td>@T2</td>
</tr>
<tr>
<td>No-Update</td>
<td>2404</td>
<td>903</td>
</tr>
<tr>
<td>Retrain</td>
<td>2238</td>
<td>933</td>
</tr>
<tr>
<td>UNLEARN</td>
<td>3350</td>
<td>1293</td>
</tr>
<tr>
<td>TRANS.</td>
<td>1508</td>
<td>849</td>
</tr>
<tr>
<td>OWAD</td>
<td>1491</td>
<td>701</td>
</tr>
</tbody>
</table>
## Performance of FP/FNs

<table>
<thead>
<tr>
<th>Methods</th>
<th># FPs (Lower is Better)</th>
<th># FNs (Lower is Better)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>@T1</td>
<td>@T2</td>
</tr>
<tr>
<td>No-Update</td>
<td>2404</td>
<td>903</td>
</tr>
<tr>
<td>Retrain</td>
<td>2238</td>
<td>933</td>
</tr>
<tr>
<td>UNLEARN</td>
<td>3350</td>
<td>1293</td>
</tr>
<tr>
<td>TRANS.</td>
<td>1508</td>
<td>849</td>
</tr>
<tr>
<td>OWAD</td>
<td>1491</td>
<td>701</td>
</tr>
</tbody>
</table>

**OWAD is the only approach that can reduce both FPs and FNs**
Real-world Deployment

• **Background**
  • SCADA in State Grid Shanghai Electric Power Company
  • Security Monitoring System (device logs and events)
  • LSTM-based Log Anomaly Detection
  • Performance degradation in long-term deployment
  • **Data**: >10M logs from 20 devices in 5 months (2022)

Ref: https://www.sciencedirect.com/science/article/abs/pii/B9780128053430000188
Real-world Deployment

- **Background**
  - SCADA in State Grid Shanghai Electric Power Company
  - Security Monitoring System (device logs and events)
  - LSTM-based Log Anomaly Detection
  - Performance degradation in long-term deployment
  - **Data**: >10M logs from 20 devices in 5 months (2022)

- **OWAD Shift Detection**

![Graphs](https://www.sciencedirect.com/science/article/abs/pii/B9780128053430000188)

Ref: https://www.sciencedirect.com/science/article/abs/pii/B9780128053430000188

**Presenter — Dongqi Han**
Real-world Deployment

- **OWAD Shift Explanation**
  - Identify 2 key logs inducing the normality shift
  - 1) network volume increases for specific devices
    - `SVR 4 4 eth3 0 0 0 eth2 1 29098502414 30822806215 eth0 1 752064 2107538`
  - 2) new service continuously launches
    - `SVR 4 13 tcp 0.0.0.0 36387 0.0.0.0 0 LISTEN 1129 rpc.statd`
  - Find the key reason of shift:
    - FTP service error due to system update & restart (Jan. 2022)

Ref: https://www.sciencedirect.com/science/article/abs/pii/B9780120128012053430000145
Real-world Deployment

• **OWAD Shift Explanation**
  - Identify 2 key logs inducing the normality shift
  - 1) network volume increases for specific devices
    > SVR 4 4 eth3 0 0 0 eth2 I 29098502414 30822806215 eth0 I 752064 2i07538
  - 2) new service continuously launches
    > SVR 4 13 tcp 0.0.0.0 36387 0.0.0.0 0 LISTEN 1129 rpc.statd
  - Find the key reason of shift:
    - FTP service error due to system update & restart (Jan. 2022)

• **OWAD Shift Adaptation**
  - Reduce >90% False Positives

<table>
<thead>
<tr>
<th>Week 1</th>
<th>Week 9 (@T1)</th>
<th>Week 18 (@T2)</th>
<th>Test @T2 (Adapt@T1)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#FP</td>
<td>#FP</td>
<td>P-value</td>
</tr>
<tr>
<td>Device A</td>
<td>14</td>
<td>25</td>
<td>0.999</td>
</tr>
<tr>
<td>Device B</td>
<td>45</td>
<td>1,027</td>
<td>0.000</td>
</tr>
<tr>
<td>Device C</td>
<td>68</td>
<td>3,071</td>
<td>0.000</td>
</tr>
</tbody>
</table>

Ref: https://www.sciencedirect.com/science/article/abs/pii/B9780123430000188
Takeaways

• Normality shift is quite common and complicated in network security domains

• After calibration, model outputs can effectively represent the normality distribution

• Labeling is inevitable for handling normality shift. Nevertheless, OWAD can achieve better performance with lower labels

• OWAD is shown to be able to reduce both False Positives and False Negatives

https://github.com/dongtsi/OWAD
Anomaly Detection in the Open World: Normality Shift Detection, Explanation, and Adaptation

Thank you! Questions?

Presenter: Dongqi Han

https://github.com/dongtsi
handq19@mails.tsinghua.edu.cn
www.dongqihan.top
OWAD Design

- We present **OWAD** (Open World Anomaly Detection) Framework
  - Detecting, Explaining, and Adapting to normality shift for DL-based anomaly detection.