DRIFT: Drift-Resilient Invariant-Feature Transformer for DGA Detection

The 56th Annual IEEE/IFIP International Conference on Dependable Systems and Networks

Chaeyoung Lee*, **Chaeri Jung***, **Seonghoon Jeong**

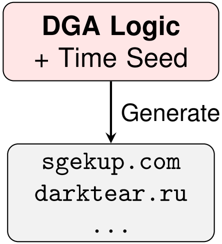
Sookmyung Women's University, Republic of Korea

SOOKMYUNG WOMEN'S UNIVERSITY

Background: DGA and Deep Learning-based Defense

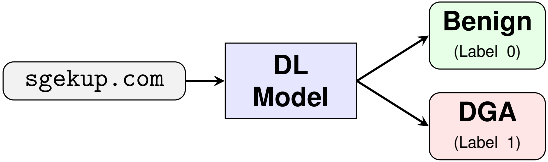
Domain Generation Algorithms (DGA)

- Core evasion mechanism for botnet C&C.
- Generates massive pseudo-random domains.
- Evades traditional **static blacklists**.



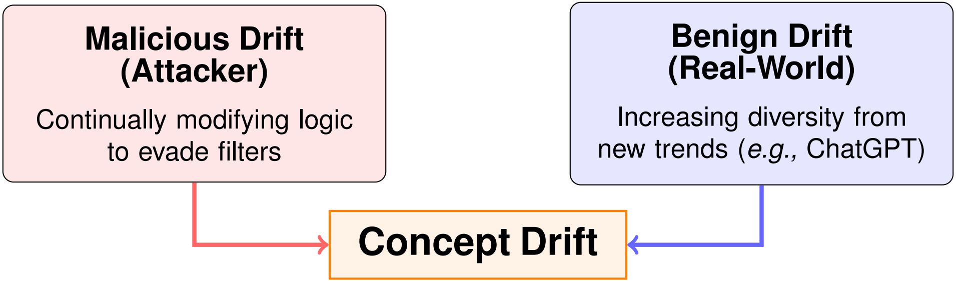
Deep Learning (DL)-based Defense

- The *de facto* standard for DGA detection.
- Operates on raw domain strings (character or subword).
- High accuracy under **static conditions**.



The Challenge: Concept Drift in Domain Names

- The ecosystem is highly dynamic due to two opposing forces:



- Static DL models overfit to temporary features and **rapidly degrade over time.**

Contributions

1. **9-Year Longitudinal Study**
2. **Dual-Branch Architecture (Character + Subword)**
3. **Multi-Task Self-Supervised Pre-Training**

Longitudinal Dataset Construction (2017–2025)

- **Benign Domains:** Alexa & Tranco Top 1M (captures real-world naming trends).
- **Malicious Domains (DGA):** DGArchive (provides precise temporal metadata)

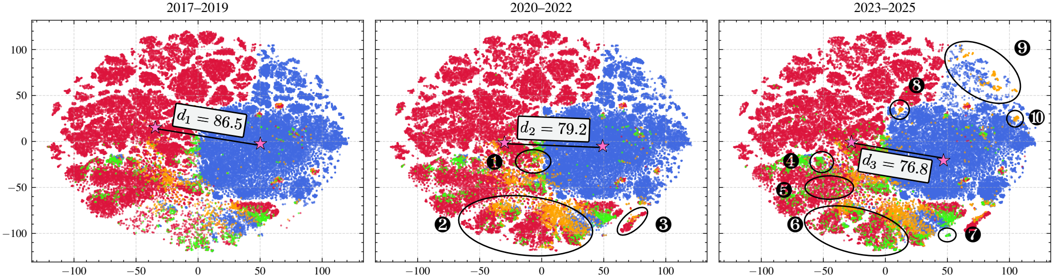
Cumulative Period	Unique Benign	Unique DGA	DGA Families
2017–2017	1.9M	14.8M	58
2017–2019	29.3M	44.9M	65
2017–2021	41.8M	79.7M	77
2017–2023	48.1M	114.1M	81
2017–2025 (Total)	**49.4M**	**149.4M**	**148**

- **High Benign Volatility**
  Over 49M unique benign domains appear across 9 years, showing that a single-year snapshot is insufficient.

- **Massive DGA Diversity**
  149M DGA domains from 148 families provide a comprehensive ground truth for temporal evaluation.

Evidence of Concept Drift: 9-Year Longitudinal Study

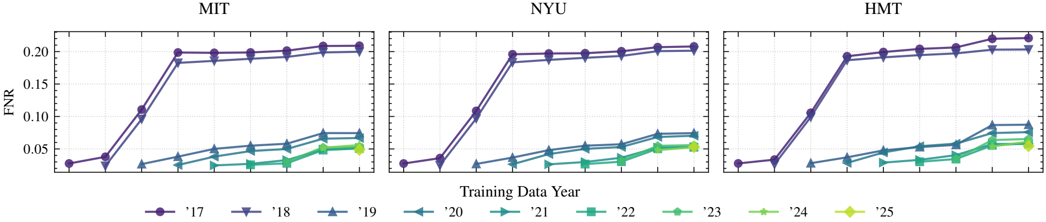
- **Dataset:** 9-year real-world domains (2017–2025, Alexa/Tranco vs. DGArchive)
- **Observation:** The latent spaces of Benign and DGA are **gradually converging** ($d_1 = 86.5 > d_2 = 79.2 > d_3 = 76.8$).



**t-SNE visualization: Margin between DGA (Red) and Benign (Blue) narrows over time.*

Temporal Degradation of Existing Models

- **Evaluation:** Forward-chaining strategy (Train on Y_{train}, Test on $Y_{\mathrm{test}} \geq Y_{\mathrm{train}}$)
- **Critical Failure:** SOTA models suffer from **Rapid Feature Obsolescence** — False Negative Rate (FNR) surges as the temporal gap widens.



- *Note: Retraining on recent data provides only short-lived improvements.*

Proposed Method: DRIFT Architecture

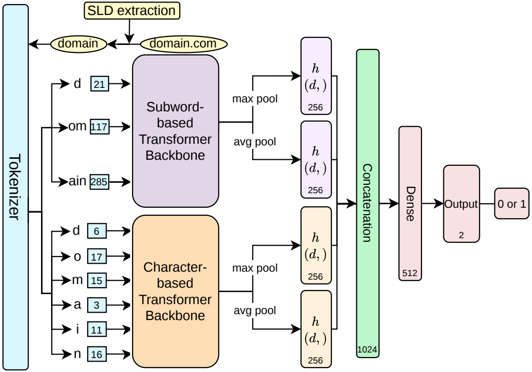
DRIFT: Drift-Resilient Invariant-Feature Transformer

Goal

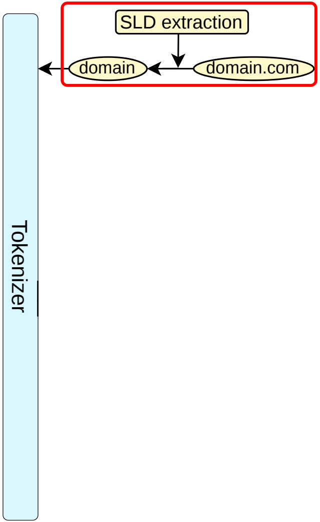
- Learn **invariant structural features** to achieve temporal robustness.

Constraint

- Domain-name-only detection (no OSINT or NXDOMAIN required).



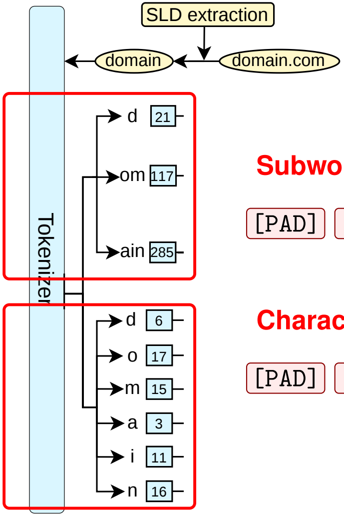
Proposed Method: DRIFT Architecture



Effective SLD extraction

- Precedes tokenization

Proposed Method: DRIFT Architecture



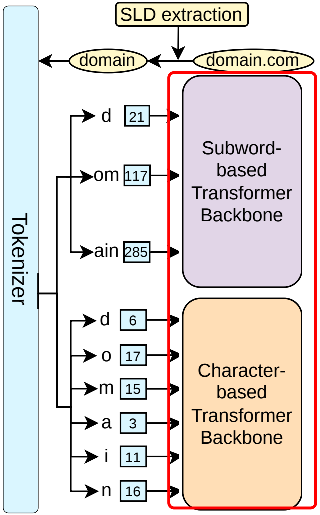
Subword branch

[PAD] [UNK] [CLS] [SEP] [MASK] + subword tokens generated by WordPiece algorithm

Character branch

[PAD] [UNK] [CLS] [SEP] [MASK] + character tokens (a–z, 0–9, -, .)

Proposed Method: DRIFT Architecture



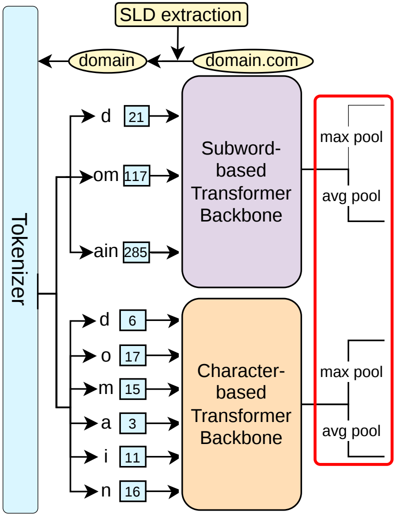
Subword-based Transformer Backbone

- Captures meaningful morphemes in word-based DGAs.
- *e.g.*, `darktear.ru`

Character-based Transformer Backbone

- Captures fine-grained, stochastic lexical patterns.
- *e.g.*, `sgekup.com`

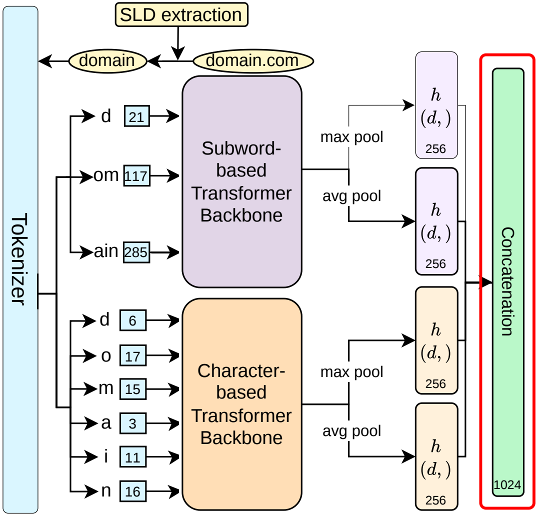
Proposed Method: DRIFT Architecture



Max and Mean Pooling

- Max and mean pooling over the sequence axis
- Hidden vectors h (dim 256) for each pooling output

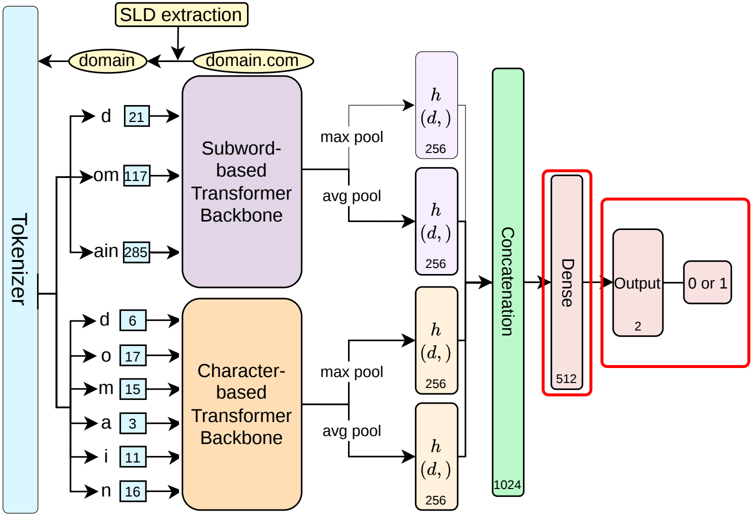
Proposed Method: DRIFT Architecture



Concatenation (dim 1024)

- Concat four h (dim 256)
- 1024-dimensional fused latent vector

Proposed Method: DRIFT Architecture



Dense

- concatenated representation is passed through a dense layer

Output (0 or 1)

- Input: output of dense layer
- passed through a classification head
- produce a probability p

Proposed Method: Data Preprocessing

- **Objective:** Isolate meaningful generative patterns and remove noise.

1. Effective SLD Extraction

- Removed Top-Level Domains (*e.g.,* `.com`, `.xyz`) and ccTLDs (*e.g.,* `.co.kr`).
- **Why?**
  - TLDs = small, **hard-coded** fixed set
  - Model exploits TLD bias *(e.g., attackers overusing cheap TLDs)* $\rightarrow$ **shortcut learning**

2. Normalization & Filtering

- Lowercased all domains (*e.g.,* IEEE.org $\rightarrow$ ieee.org).
- Filtered out non-standard characters (*e.g.,* underscores) following IETF RFC 1035.

3. Deduplication

- Retained only *unique* effective SLDs to prevent over-representing popular domains.

Proposed Method: Self-Supervised Pre-Training

- **Objective:** Learn inherent structural regularities without relying on short-lived features.

1. MTP (Masked Token Prediction)

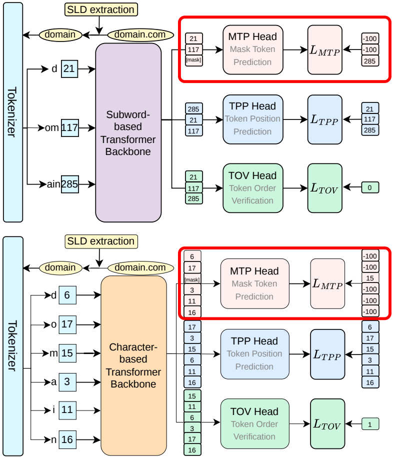
- Predicts hidden tokens based on context.
- Learns *local semantic dependencies*.

Subword Branch Example:

`[CLS] wiki [MASK] di a [SEP]` → `pe`

Character Branch Example:

`[CLS] w i [MASK] i p e d i a [SEP]` → `k`



Proposed Method: Self-Supervised Pre-Training

- **Objective:** Learn inherent structural regularities without relying on short-lived features.

2. TPP (Token Position Prediction)

- Reconstructs original order from scrambled tokens.
- Learns *canonical global structure*.

Subword Branch Example:

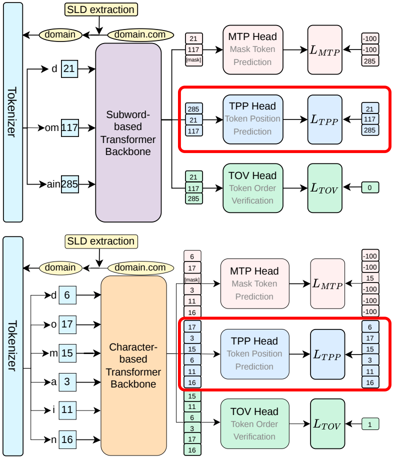
[CLS] di wiki a pe [SEP]

→ [CLS] wiki pe di a [SEP]

Character Branch Example:

[CLS] a i p k w e d i i [SEP]

→ [CLS] w i k i p e d i a [SEP]



Proposed Method: Self-Supervised Pre-Training

- **Objective:** Learn inherent structural regularities without relying on short-lived features.

3. TOV (Token Order Verification)

- Sequence-level binary task (Coherent vs. Scrambled).
- Captures *high-level semantic coherence*.

Subword Branch Example:

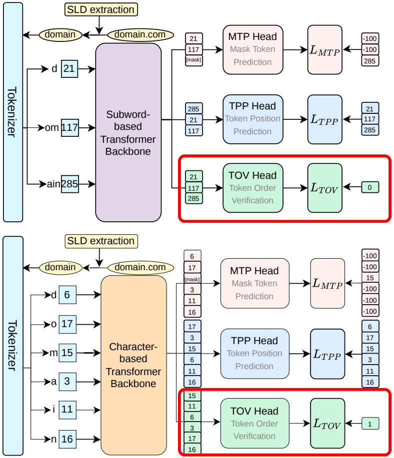
Normal (0): [CLS] wiki pe di a [SEP]

Scrambled (1): [CLS] di wiki a pe [SEP]

Character Branch Example:

Normal (0): [CLS] w i k i p e d i a [SEP]

Scrambled (1): [CLS] a i p k w e d i i [SEP]



Proposed Method: Feature Fusion & Fine-Tuning

1. Feature Fusion

- Extracted hidden states from the last Transformer layer of both branches.
- Applied both **Max pooling** and **Mean pooling** to aggregate sequence information (not just the `[CLS]` token).
- Concatenated into a single robust vector:

$$v_{\text{fusion}} = \left[v_{\text{subword}}; v_{\text{char}}\right]$$

2. Two-Stage Training Strategy

- **Stage 1: Linear Probing (Warm-up)**
  Freeze both pre-trained encoders.
- **Stage 2: End-to-End Fine-Tuning**
  Unfreeze encoders.
- **Benefit:** Preserves invariant structural features from pre-training while adapting to DGA detection.

Experimental Setup

1. Forward-Chaining Protocol

- **Training Period (2017–2019):** Used for pre-training and fine-tuning.
- **Testing Period (2020–2025):** Evaluated sequentially year-by-year.

2. Baselines (10 SOTA Methods)

- **Classical Deep Learning**
  - **Endgame** (LSTM-based)
  - **MIT** (CNN-LSTM hybrid)
  - **NYU** (Character CNN)
- **Residual Networks**
  - **B-ResNet** / **M-ResNet**
- **Statistical Embedding**
  - **Dom2Vec** (Domain word2vec)
- **Advanced Hybrid & Transformers**
  - **HMT** (Hybrid Modified Transformer)
  - **HDDN** (DefenseNet variant)
- **Pre-trained Language Models**
  - **BERT** (Fine-tuned encoder)
  - **SFT-Llama3-8B**

Overall Performance

1. Overall Metrics (Micro-Averaged)

Method	Accuracy	Precision	Recall	F1-score
Endgame (2016)	0.9439	0.9770	0.9397	0.9579
MIT (2016)	0.9306	**0.9867**	0.9101	0.9468
NYU (2016)	0.9320	0.9858	0.9131	0.9481
B-ResNet (2020)	0.9399	0.9797	0.9308	0.9546
M-ResNet+B-cos (2022)	0.9488	0.9775	0.9465	0.9617
Dom2Vec (2023)	0.9279	0.9686	0.9237	0.9457
HMT (2023)	0.9318	0.9865	0.9121	0.9478
SFT-Llama3-8B (2024)	0.8052	0.8951	0.8081	0.8493
HDDN (2025)	0.9392	0.9781	0.9313	0.9541
Fine-tuned BERT	0.9383	0.9837	0.9244	0.9531
DRIFT(Supervised)	0.9420	0.9822	0.9315	0.9562
DRIFT(Ours)	**0.9528**	0.9818	**0.9481**	**0.9646**

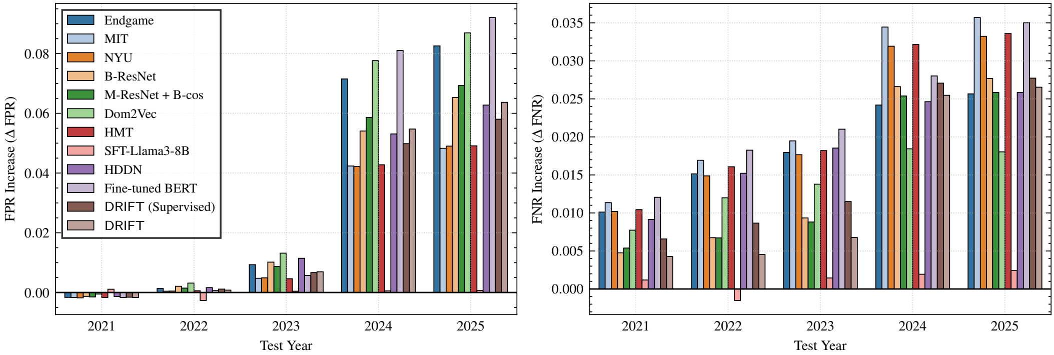
Why Pre-Training Mitigates Drift?

- **Structural Invariance**
  - **Surface vocab:** Shifts annually
  - **Syntax/Structure:** Remains stable (token order, local consistency)
- **Pre-training Effect**
  - Learn structure before labels $\rightarrow$ Invariant representation space
  - Proven by empirical evaluation

Overall Performance

2. Temporal FNR Stability

- Maintains the lowest FNR $\rightarrow$ Highly robust against Concept Drift.



Performance Against Future Unseen DGAs

- **DGA Families:** 65 families in the training set, 83 unseen families in the test set.

Unseen-Families FNR

Model	FNR
MIT (2016)	0.2790
NYU (2016)	0.2551
HMT (2023)	0.2596
DRIFT (Ours)	**0.1439**

Key Takeaways

- **Zero-Day Generalization**
  - DRIFT achieves the lowest FNR (**14.39%**)
  - Outperforms SOTA baselines by **11%p**
- **Invariant Representation**
  - Proves that DRIFT captures stable, language-independent structures
  - Avoids overfitting to training-family lexicons

Ablation Studies: Base Configuration

	Backbone	Pre-train Data	V_{sub}	Unfrozen	F1-score
Base	**H**	**Benign & DGA**	**30,522**	✓	**0.9646**
(a)	H	Benign only	†30,522	×	0.9539
(b)	H	Benign only	500	×	0.9564
(c)	H	Benign only	1,000	×	0.9538
(d)	H	Benign only	30,522	×	0.9533
(e)	S	Benign & DGA	30,522	✓	0.9511
(f)	C	Benign & DGA	–	✓	0.9387
(g)	H	Benign & DGA	30,522	×	0.9585
(h)	H	Benign only	30,522	✓	0.9584

**H: Hybrid, S: Subword only, C: Character only* — † *Off-the-shelf BERT-base-uncased tokenizer.*

- **1. Tokenizer (a–d):**
  The custom domain tokenizer is better.
- **2. Dual vs. Single (e–f):**
  Hybrid architecture strongly outperforms.
- **3. Fine-Tuning Strategy (g):**
  End-to-end fine-tuning yields higher F1.
- **4. Pre-training Data (h):**
  Including DGA samples improves detection.

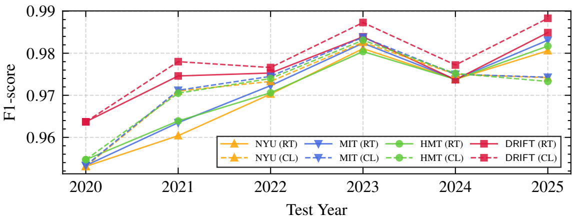
Ablation Studies: Pre-training Subtasks

Case	MTP	TPP	TOV	Accuracy	Precision	Recall	F1-score
1	✓			0.9436	0.9835	0.9385	0.9603
2		✓		0.9442	0.9814	0.9402	0.9603
3			✓	0.9473	0.9802	0.9458	0.9626
4	✓	✓		0.9501	0.9844	0.9460	0.9647
5	✓		✓	0.9518	**0.9857**	0.9473	**0.9660**
6		✓	✓	0.9493	0.9840	0.9451	0.9641
7 (Base)	✓	✓	✓	**0.9528**	0.9818	**0.9481**	0.9646

- **Single Task Analysis (1–3):**
  - TOV alone (Case 3) yields the best single-task performance.
  - Sequence-level verification captures strong global cues.
- **Why use all 3 tasks? (Case 7):**
  - Case 5 has slightly higher F1.
  - **Case 7 maximizes Recall** (Lowest false negatives).
  - **Security tradeoff:** Missing a DGA poses a far greater risk than false alarms.

Performance under Adaptive Drift-Mitigation

- **Setup:** Evaluated adaptability against continuous concept drift using Periodic Retraining (RT) and Continuous Learning (CL).



- **Continuous Learning (CL) in DRIFT**
  - Backbones are **frozen**.
  - Only the classification head is updated.
  - Maximizes efficiency & prevents catastrophic forgetting.
- **Key Results**
  - **CL-DRIFT** achieves the highest F1 every year.
  - Proves pre-trained features are highly **invariant** and adaptable.

Conclusion & Future Work

Conclusion

- **Identified the Problem:**
  Revealed the severity of Concept Drift via a 9-year longitudinal study.
- **Proposed DRIFT:**
  A drift-resilient framework using Hybrid Tokenization and Self-Supervised Pre-training.
- **Dependable Defense:**
  Substantially reduced and stabilized the FNR over time without relying on OSINT or NXDOMAIN.

Future Work

- **Remaining Challenge:**
  Benign Concept Drift. Like all baselines, DRIFT experiences a rise in False Positive Rates (FPRs) as legitimate naming trends evolve.
- **Future Directions:**
  Explore calibrated uncertainty or adaptive thresholding to reduce FPR drift while preserving the low FNR.

Thank You.

Q & A

Contact: chaerry502@sookmyung.ac.kr

Dataset

https://doi.org/10.21227/za2s-9e09

Source code

https://github.com/snsec-net/2026-DSN-DRIFT