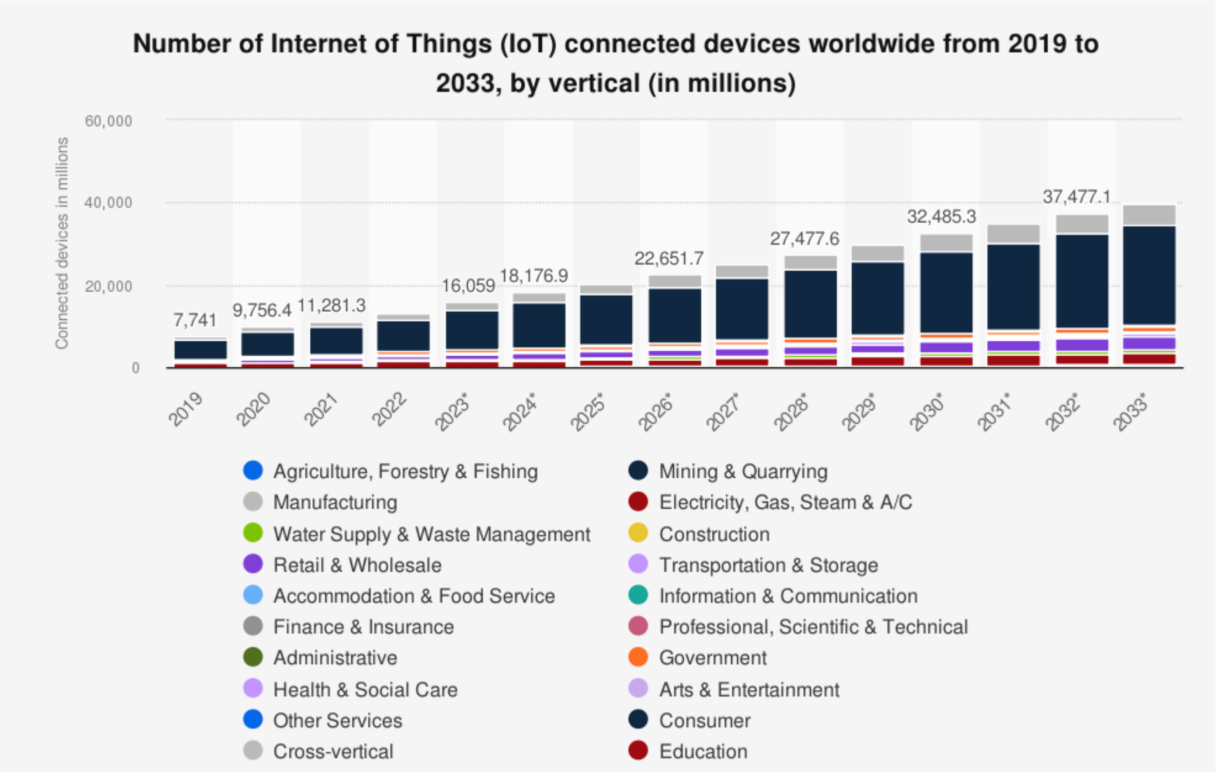
TwinGuard: An Adaptive Digital Twin for Real-Time HTTP(S) Intrusion Detection and Threat Intelligence

Yuanyuan Zhou, Anna Maria Mandalari
Ryu Kuki, Takayuki Sasaki, Katsunari Yoshioka

Motivation

- Modern IoT Challenges Demand New Defences



IoT devices are **widely deployed** across critical infrastructure domains

Traditional IDS struggle with **evolving, obfuscated threats**

Resource constraints on IoT and edge devices limit the feasibility of heavy-weight security solutions

Limited labelled data in real world settings makes **supervised detection** difficult

Real-time, adaptive, and explainable intrusion detection is urgently needed

Figure source: Transforma Insights. "Number of Internet of Things (IoT) Connected Devices Worldwide from 2019 to 2033, by Vertical (in Millions) ." *Statista*, Statista Inc., 10 May 2024, https://www.statista.com/statistics/1194682/iot-connected-devices-vertically/

Previous Work

Focus	Papers	Method	Contribution
Digital twins in cybersecurity	Rajab et al (2024)	data generation based on new attacks	proposed an DT based AutoML pipeline to enhance intrusion detection
	Nintsiou et al(2023)	Honeypot behaviour optimization	combines digital twin technology with honeypots to enhance Honeypot Behaviour

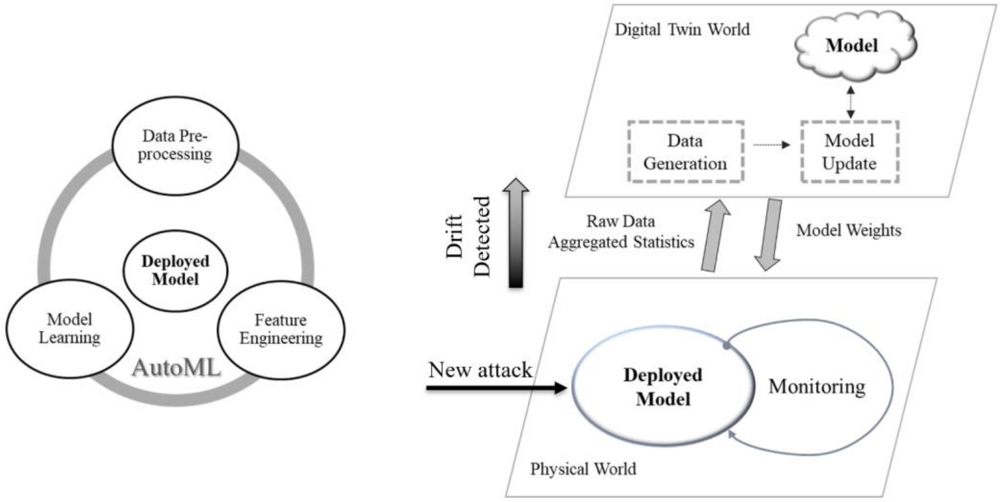


FIGURE 3: Overview of the System Environment

Rajab et al (2024)

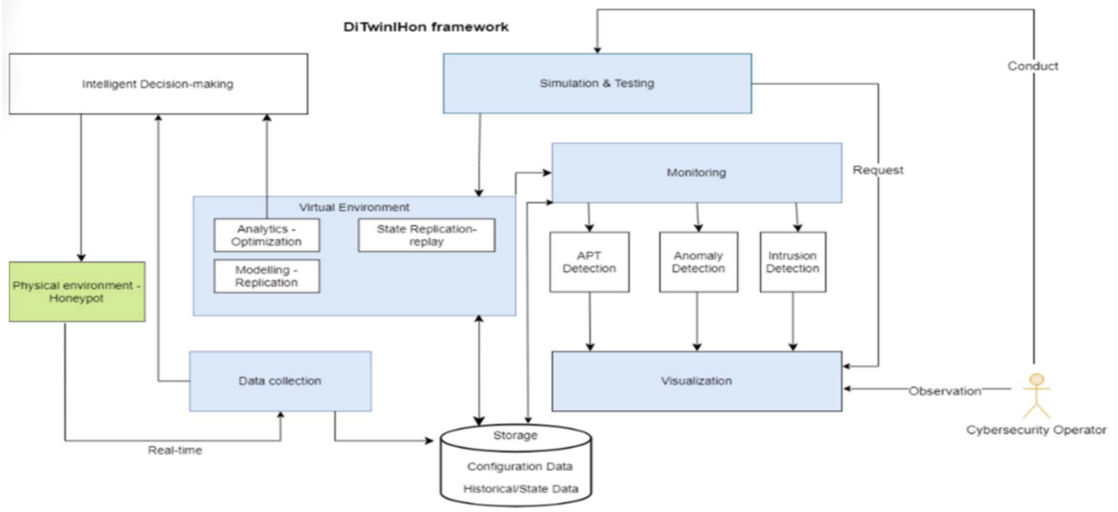


Figure 1. DiTwinHon framework

Nintsiou et al(2023)

- Digital twin concepts are widely applied in **Industrial Control System (ICS) security**, rarely **web-based attacks.**
- Prior work targets **physical systems** or **network-layer threats**, and focus on data generation
- No existing system uses **real-time honeypot data** to detect **application-layer attacks** adaptively.

Previous Work

Focus	Papers	Method	Contribution
Wild Web Attack Analysis	**Canali et al. (2013)**	Real-world honeypot attack sessions with multi-stage workflow analysis	13 post-exploitation types (e.g., web shells, IRC bots, spam)
	Li et al. (2021)	Honeysite-based bot & HTTP threat study	Categorizes traffic (scanning, credential stuffing, exploits); highlights fingerprinting limits of UA strings

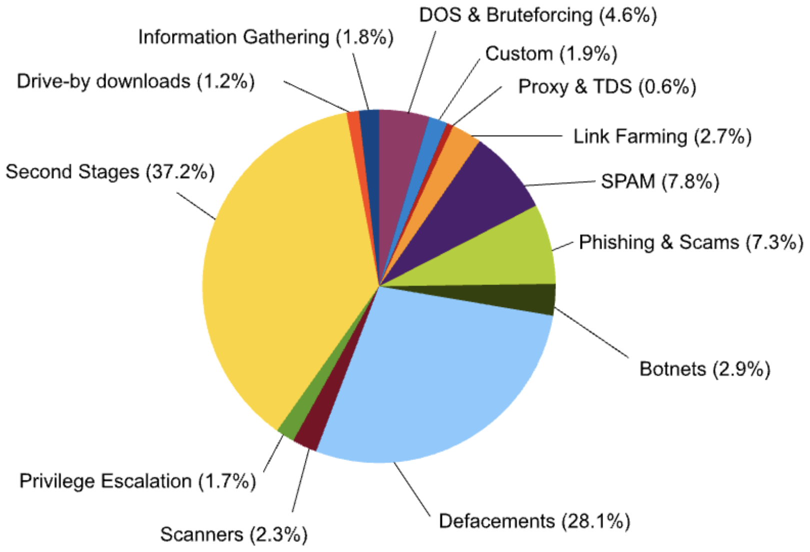


Figure 6. Attack behavior, based on unique files uploaded

TABLE IV: Popular TLS fingerprint distribution. Entries below the line correspond to Chromium-based tools that were not in the top ten, in terms of unique bot IP count.

Tools	Unique FPs	IP Count	Total Requests
Go-http-client	28	15,862	8,708,876
Libwww-perl or wget	17	6,102	120,423
PycURL/curl	26	3,942	80,374
Python-urllib 3	8	2,858	22,885
NetcraftSurveyAgent	2	2,381	14,464
msnbot/bingbot	4	1,995	44,437
Chrome-1(Googlebot)	1	1,836	28,082
Python-requests 2.x	11	1,063	754,711
commix/v2.9-stable	3	1,029	5,738
Java/1.8.0	8	308	1,710
MJ12Bot	2	289	28,065
Chrome-2(Chrome, Opera)	1	490	66,631
Chrome-3(Headless Chrome)	1	80	2,829
Chrome-4(coc_coc_browser)	1	4	101
Total	**113**	38,239	9,879,326

- Existing taxonomies are often limited to **specific attack categories.**
- Prior fingerprinting work mostly focuses on **source identification.**
- We analyze the intrusions from the wild and give the profiling based on **behavioral characteristics** and **taxonomy validation**

Introduction

Digital Twin Framework

- mirrors real attacker behaviour: captured by honeypots
- using a virtual model that learns and adapts over time

Core Mechanisms

structured sequence modelling + ML classification + semantic profiling

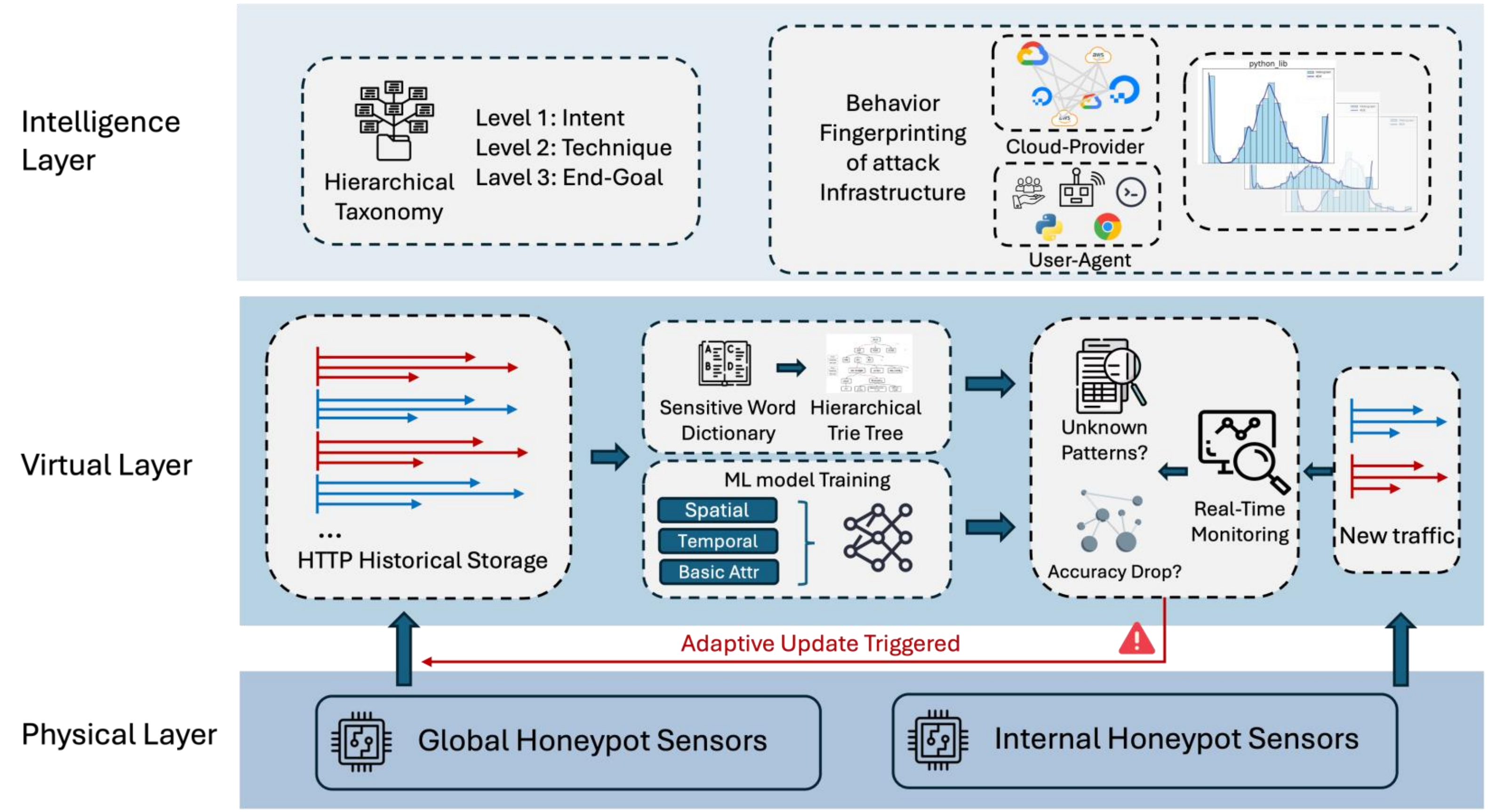
TwinGuard Properties

Modular

Lightweight

Extensible

TwinGuard Design

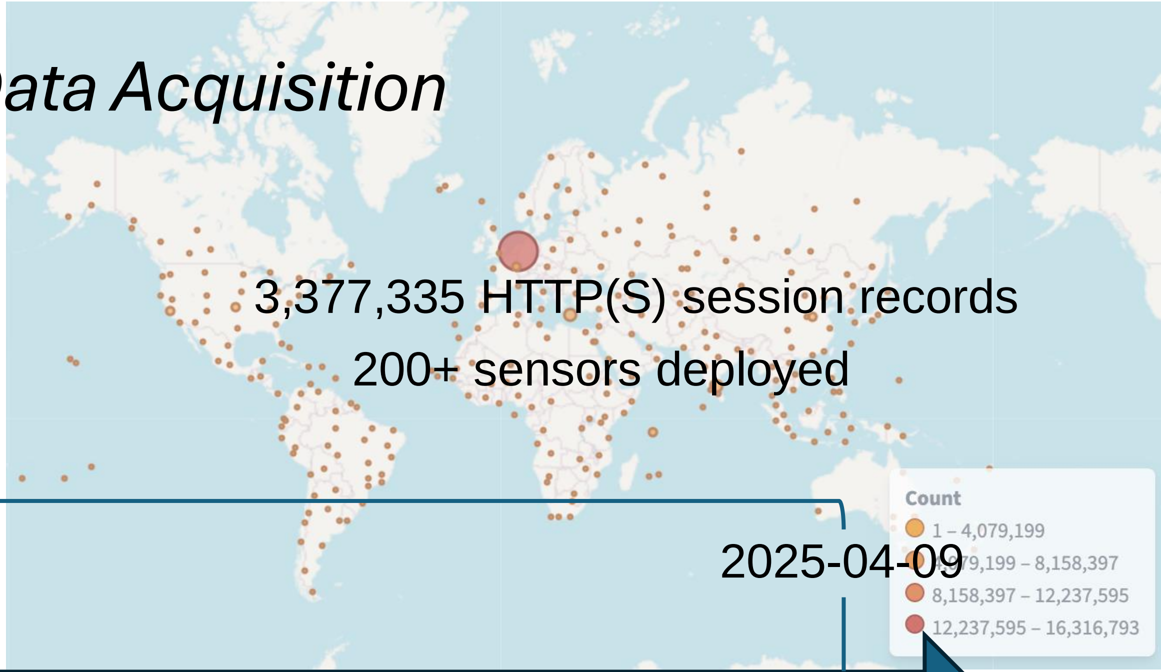


- Hierarchical Labelling
- Attacker Fingerprinting
- Reveals what, where, and how threats evolve

- Trie-Based Path Model matching
- Keyword dictionary for Granularity Reduction
- ML Classifiers for IDS
- Sliding-Window retraining Mechanism

- Capture Real-world HTTP(S) attacks

Physical Layer – *Honeypot Networks and Data Acquisition*



GLOBAL CYBER ALLIANCE

Primary Honeypot Network

ProxyPot

3,377,335 HTTP(S) session records

200+ sensors deployed

2025-03-15

2025-04-09

2025-03-26

2025-03-31

To test generalization under heterogeneous input →

Yokohama National University YNU

Internal Honeypot Network

X-POT

847,869 HTTP requests

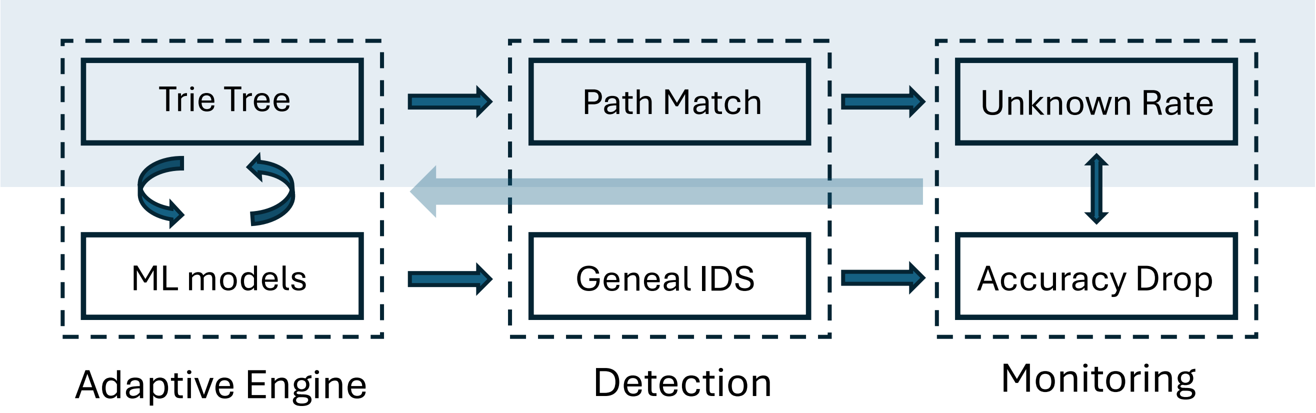
19 sensors deployed

70% of fields align with our primary schema

Virtual Layer – Real-Time Monitoring and Adaptive Detection

Trie Monitoring

interpretable view of structured request paths by aggregating common behaviour patterns



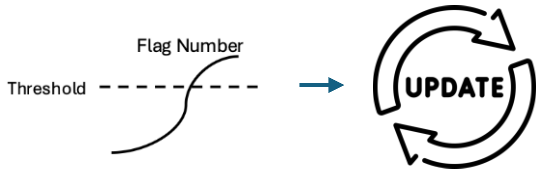
Sensitive Word Extraction → configurable frequency threshold (default= 20)

↓

Structured Path Representation → *(method → status → URI keywords)* { *GET /app/login_panel?user=admin&theme=dark HTTP/1.1* ; *(GET → 200 → login,admin)*

↓

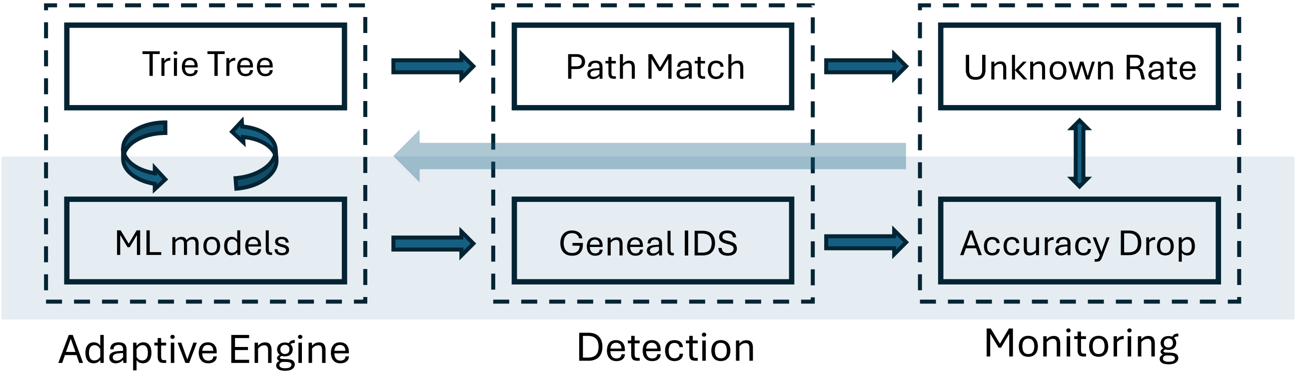
Path Match → Unknown Flag++ →

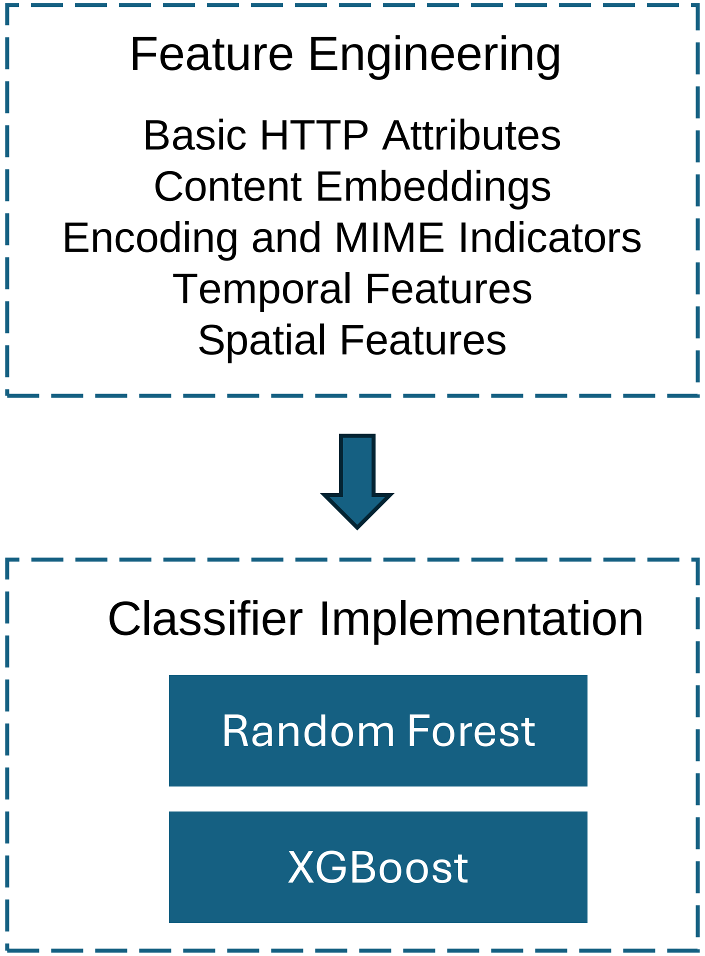


Virtual Layer – Real-Time Monitoring and Adaptive Detection

Machine learning classifiers

general-purpose intrusion detection component

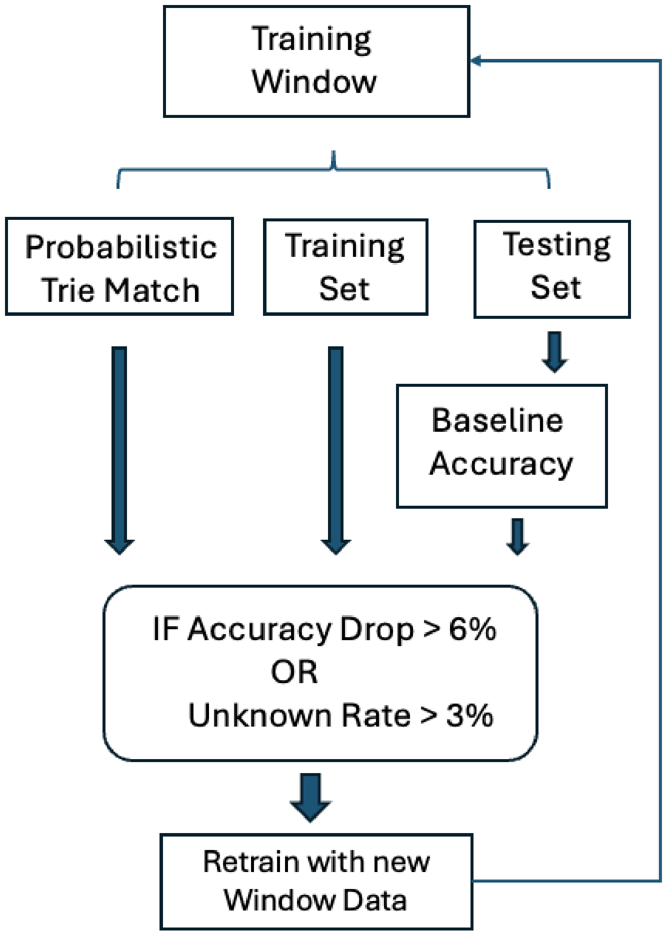




Virtual Layer – Real-Time Monitoring and Adaptive Detection

Sliding Window Mechanism

continuously monitors performance degradation and structural novelty within the HTTP(S) traffic stream



Monitoring module: Adaptive Loop Structure

Classification:

Scan | Attempt | Intrusion-Control

Stable Periods:

- both classifiers drops by less than **6.0%**
- the unknown pattern rate under **3.0%**

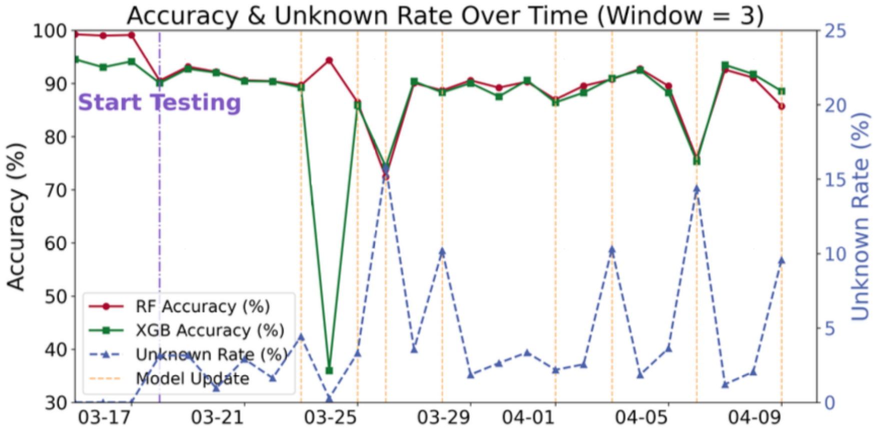
Labeling Criteria:

- Intrusions are labelled using **rule-based matching** of structured request paths, **payload content**, and **endpoint semantics**.

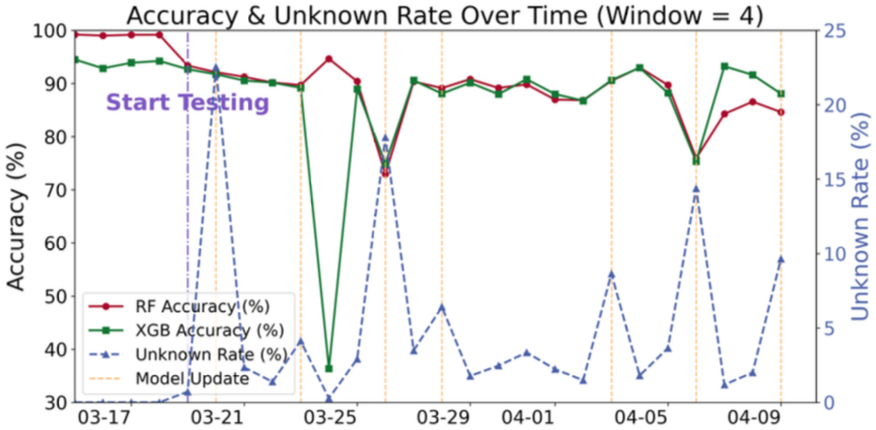
- If a spike in unknown patterns occurs without existing labels, we check if **new labelling is needed** to maintain detection accurate.

Virtual Layer – Real-Time Monitoring and Adaptive Detection

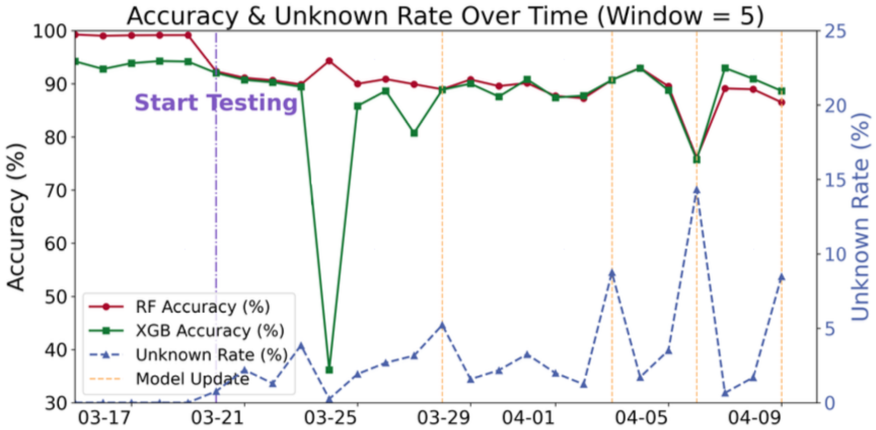
Accuracy and Unknown Rate Dynamics



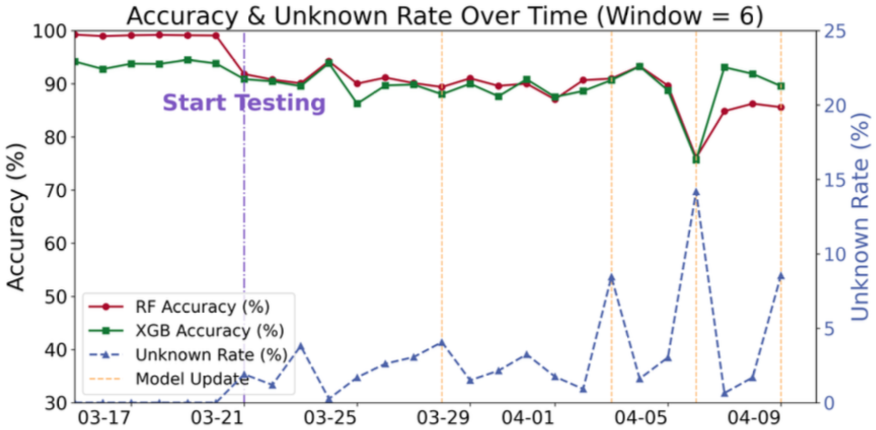
(a) $w = 3$



(b) $w = 4$



(c) $w = 5$



(d) $w = 6$

Smaller Windows

- Fast Reaction
- Frequent Updates
- Higher Volatility

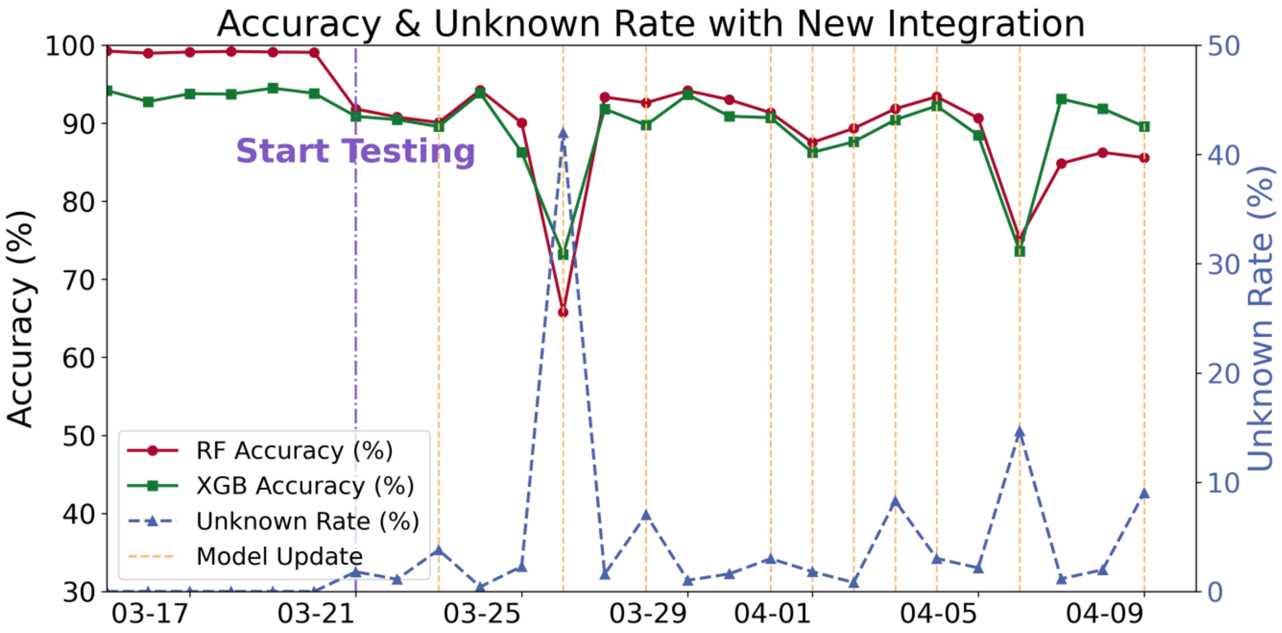
Larger Windows

- Stable Accuracy
- Fewer Updates
- Lower Unknown Rate

w = 6 strikes a balance between the model utility and stable performance

Virtual Layer – Real-Time Monitoring and Adaptive Detection

Adaptive ability with the integration of X-POT



Adaptation to a new honeypot (X-Pot) source under window size w = 6.

A surge in unknown sequences and an accuracy drop is observed upon integration, followed by recovery after retraining.

Intelligence Layer: Intrusion Labelling and Attacker Attribution

Hierarchical Pattern-Based Intrusion Labelling

Intrusion Category	Technique	End Goal
Exploit Attempts	File Inclusion (LFI/RFI)	Code Execution
	Misconfiguration Exploit	Priv. Esc. / Info Leak
	REST/JSON Abuse	Data Leak / Enumeration
	SQL Injection (SQLi)	DB Access / Bypass
	Command Injection	Code Execution
	Denial of Service (DoS)	Resource Exhaustion
Web Shell Upload	Simple Shell Upload	Persistent Access
	Obfuscated Shell Upload	Stealth Backdoor
	Two-Stage Payload	Loader & Dropper
Post-Exploitation Activity	Botnet C2 Callback	Remote Control
	Cronjob Deployment	Persistence
	Spam Mailer Setup	Email Abuse
	Proxy/Relay Deployment	Lateral Movement
Delivery / Downloader	Direct Script Drop	Code Execution
	Drive-by Download / JS	User Exploitation
Obfuscated / Anomalous Behavior	Junk Payload Flood	Resource Exhaustion
	Unknown Pattern	Undiscovered Variant

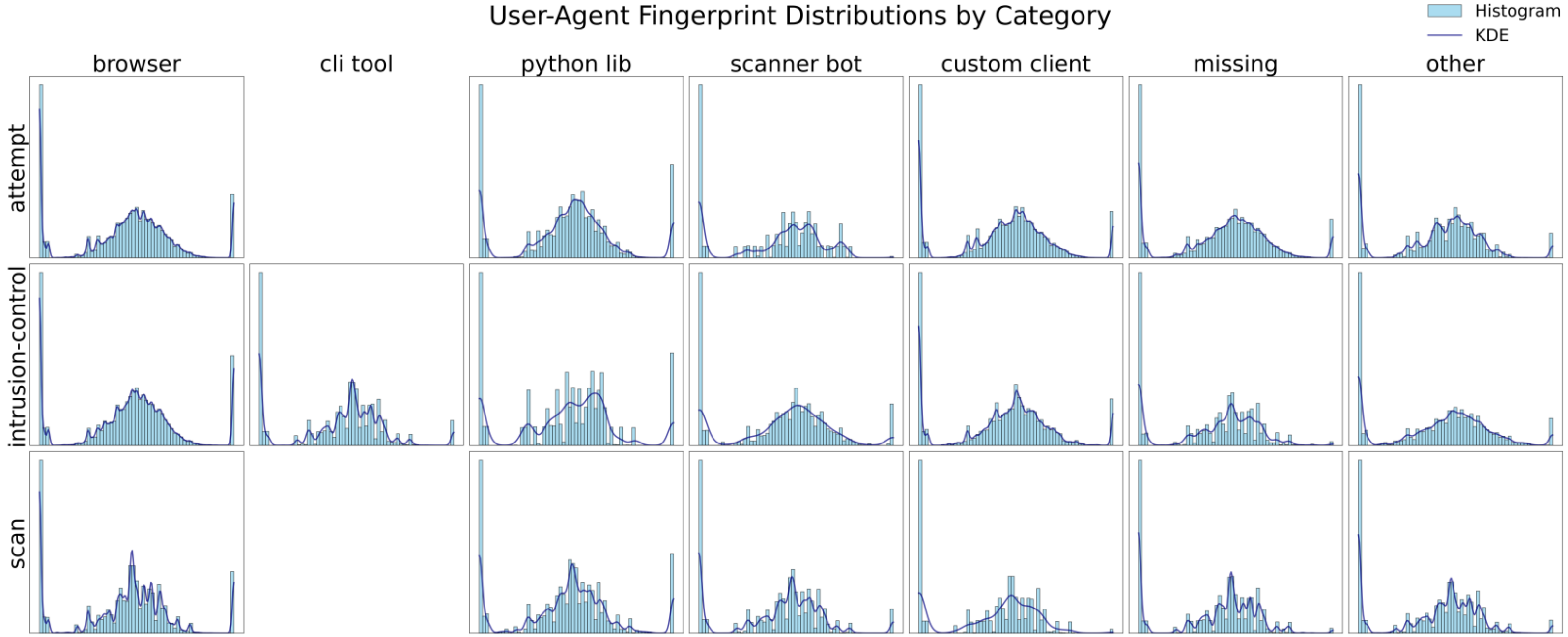
Hierarchical taxonomy structure:

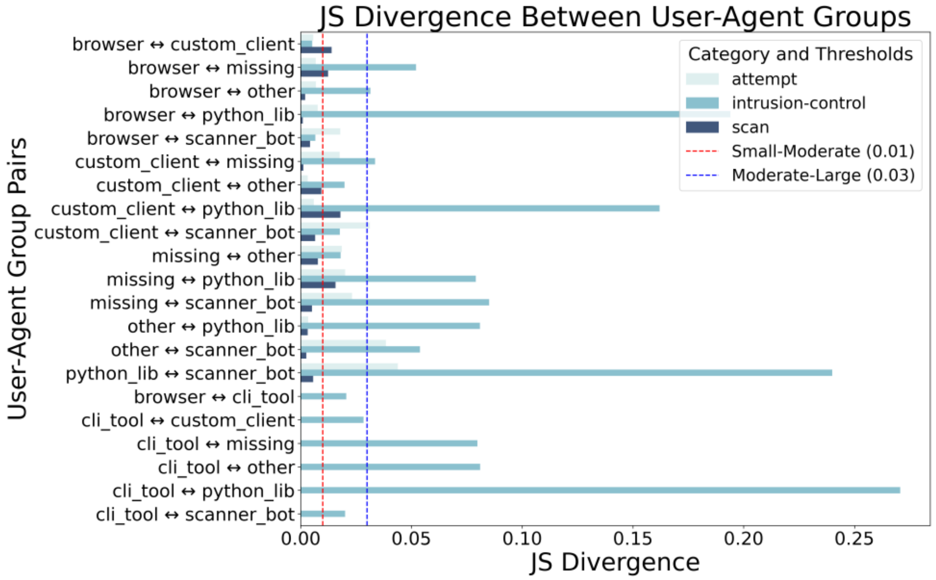
- Level 1: Parent Category (e.g., Exploit, Downloader) *~high-level intent*
- Level 2: Subtypes (e.g., SQLi, Command Injection). *~how it's done*
- Level 3: End Goals (Execution, Leak, etc.). *~why the attacker is doing it*

Intelligence Layer: Intrusion Labelling and Attacker Attribution

Attacker Behavioural Fingerprinting Feature distributions are visualized using histograms and kernel density estimates (KDE)

User-Agent





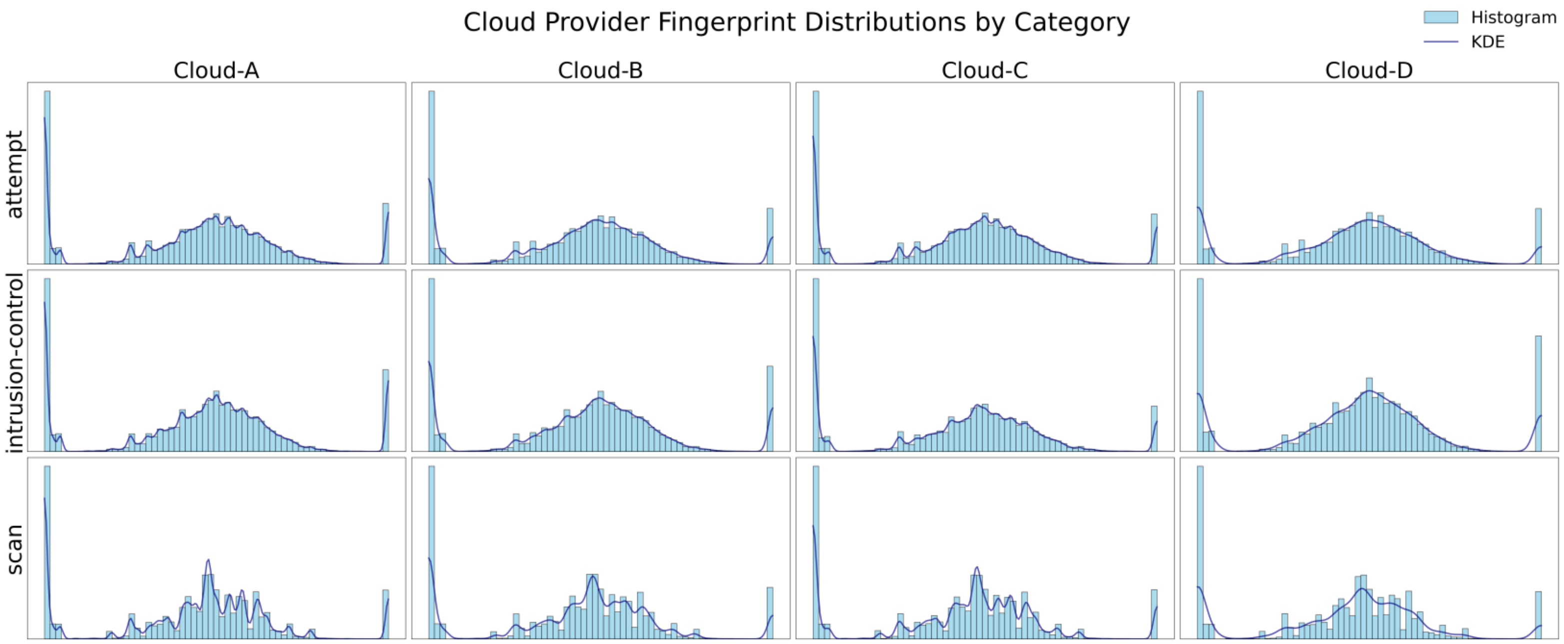
The *x-axis* represents different HTTP session features, and the *y-axis* indicates their normalized values across sessions.

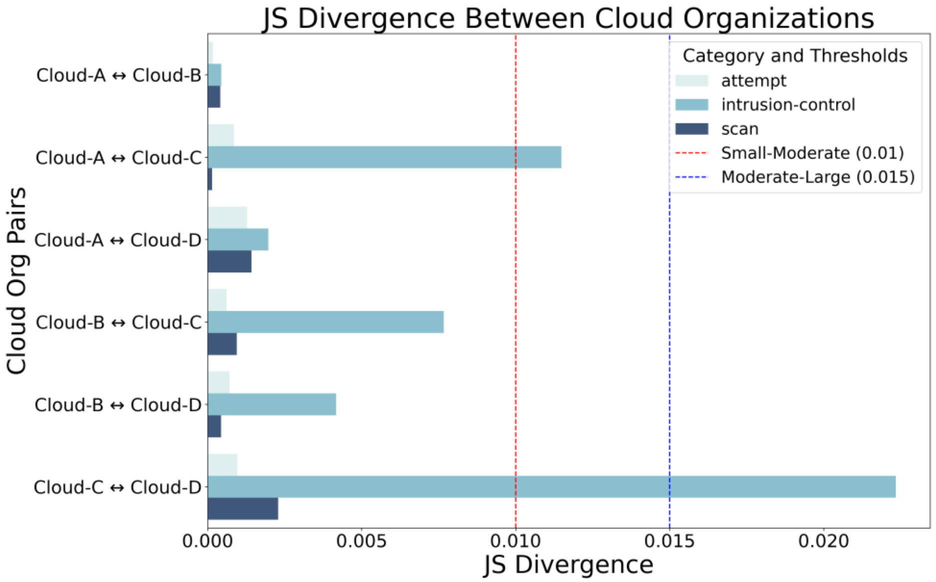
- **Diverse behaviour across UA groups**, especially in intrusion-control.
- **High divergence** observed between *scanner bot*, *python library* , indicates distinct attack behaviours.

Intelligence Layer: Intrusion Labelling and Attacker Attribution

Attacker Behavioural Fingerprinting

Cloud Provider

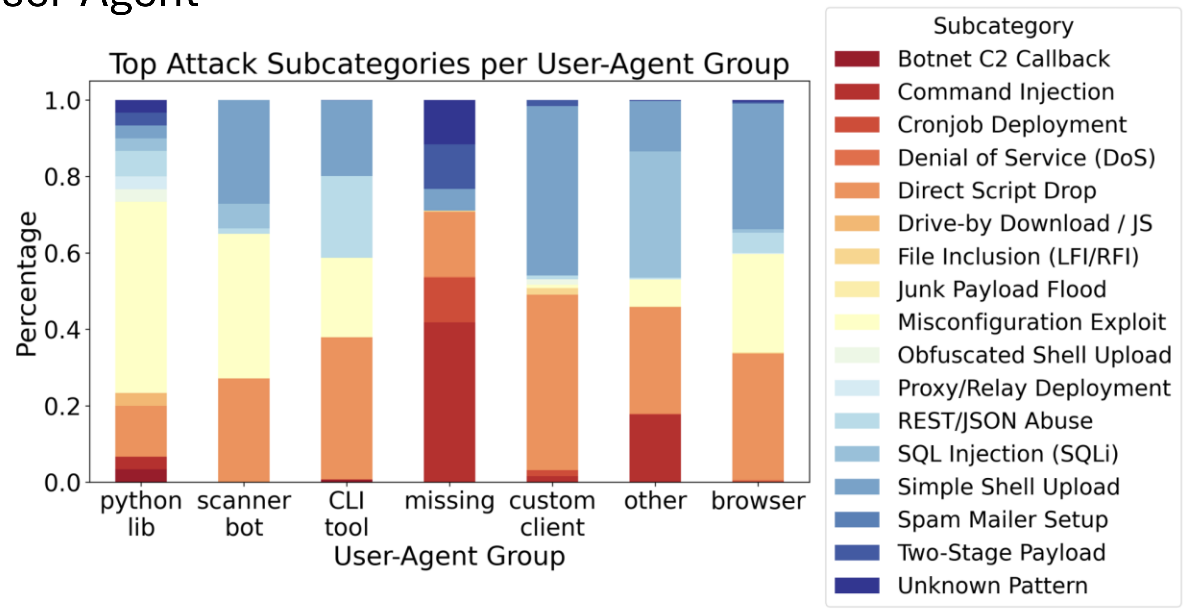




- **Overall low divergence** → attack behaviour is largely consistent across cloud platforms.
- **Cloud C shows slight divergence** in intrusion-control attacks.
- **Impact is minimal** → cloud provider has **limited influence** on attack diversity.

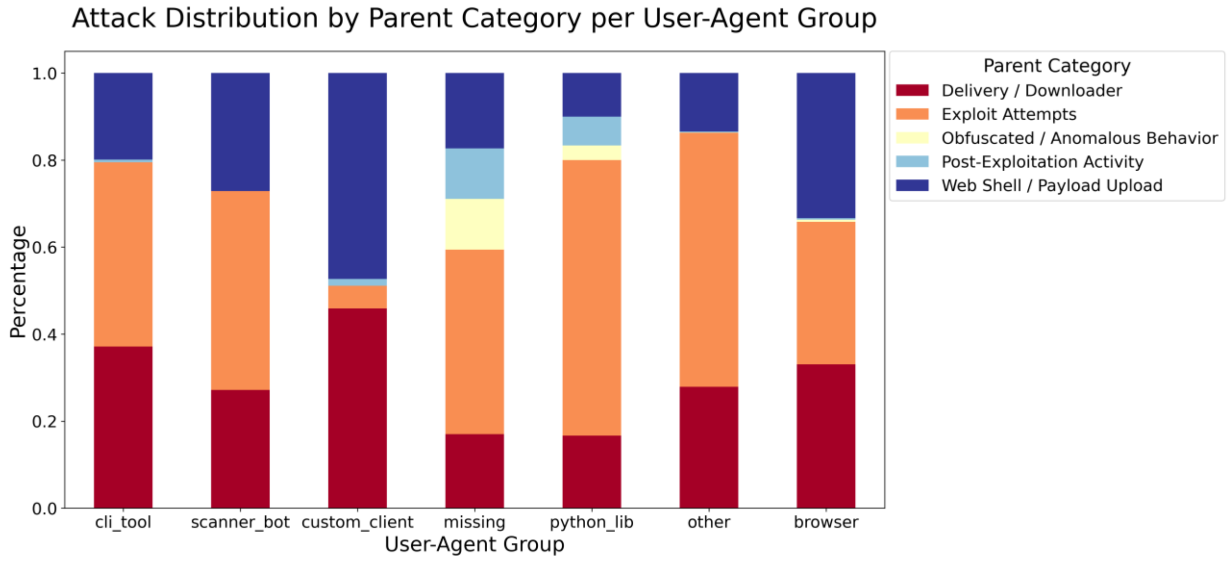
Intelligence Layer: Intrusion Labelling and Attacker Attribution

User-Agent



Browser and CLI tool sessions are concentrated in broad categories like exploit attempts and web shell uploads, reflecting traditional probing behaviour.

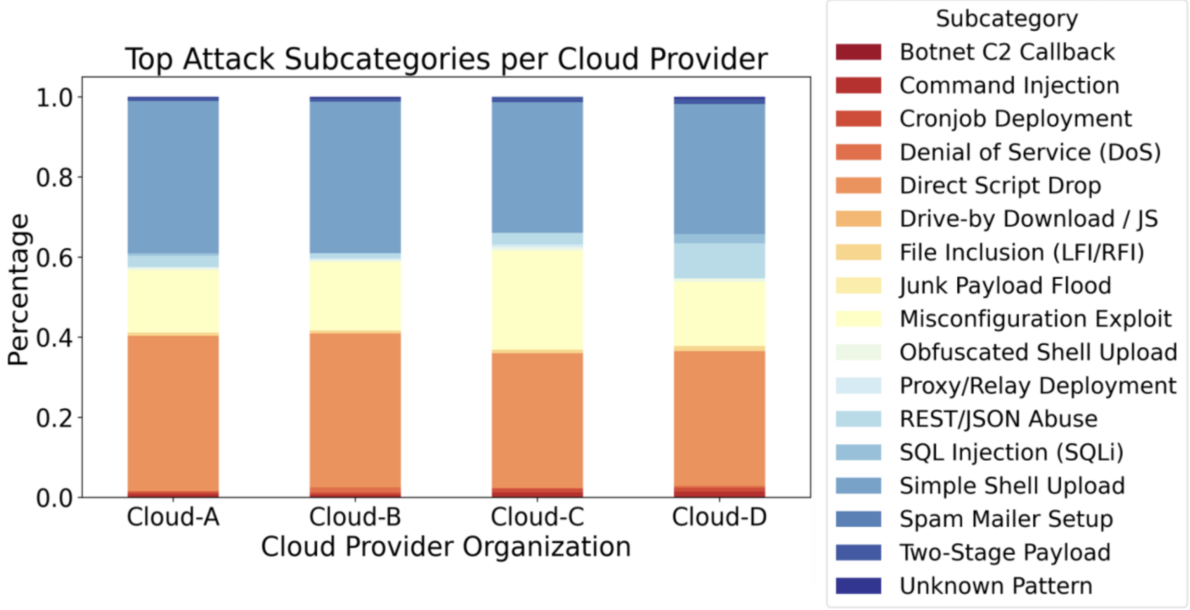
python libraries and scanner bots demonstrate greater technique diversity, especially in misconfiguration exploits and file inclusion (LFI/RFI).

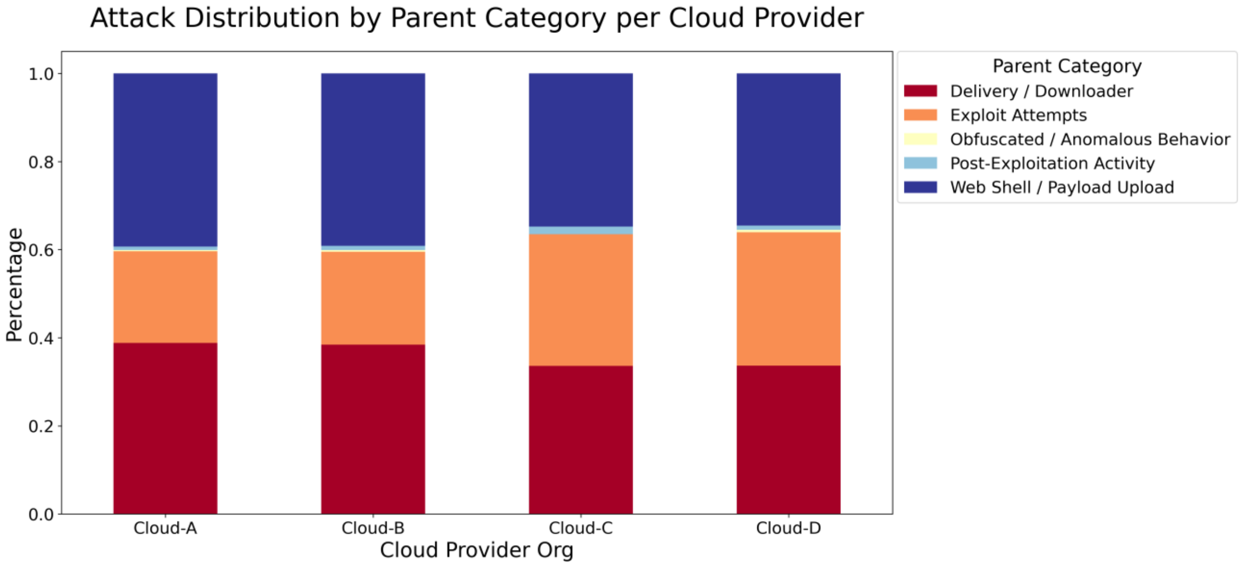


The missing and other categories display highly irregular distributions, suggesting spoofed or unstable automation strategies.

Intelligence Layer: Intrusion Labelling and Attacker Attribution

Cloud Provider





- **Shared Attack Focus**: All cloud providers show similar dominance in script drops & shell uploads, matching low JS divergence.

- **Minor Exploit Variations**: Slight shifts (e.g., more SQLi on Cloud-D, misconfiguration on Cloud-C) don't alter overall behaviour.

- Confirms cloud-based attacks are likely **templated and automated**, regardless of provider.

Conclusion

High Accuracy & Responsiveness

- Maintains **>90% accuracy** during stable periods
- **Dual classifiers** + **sequence monitoring (Trie)** ensure robustness

Adaptive Retraining Triggered by Novelty

- **Strong negative correlation** between unknown rate and accuracy
- **42% spike** in unknowns + **30% accuracy drop** mitigated in **1 update cycle**

Real-World Deployment with Diverse Traffic

- Processes traffic from **heterogeneous honeypot sources**
- Demonstrates **adaptability across environments**

Behavioral Intelligence

- Reveals **diverse attacker behaviour** across user-agent types
- **Cloud-based traffic** shows consistent patterns → shared tooling

Future Work

Real-World Deployment & Evaluation

Transition from honeypot-only testing to real production environments

Expand Protocol Coverage

Move beyond HTTP(S) to include protocols like SSH, FTP, and DNS

Enable Continuous Streaming

Integrate TwinGuard with live traffic pipelines, from time-bounded snapshots to fully real-time monitoring

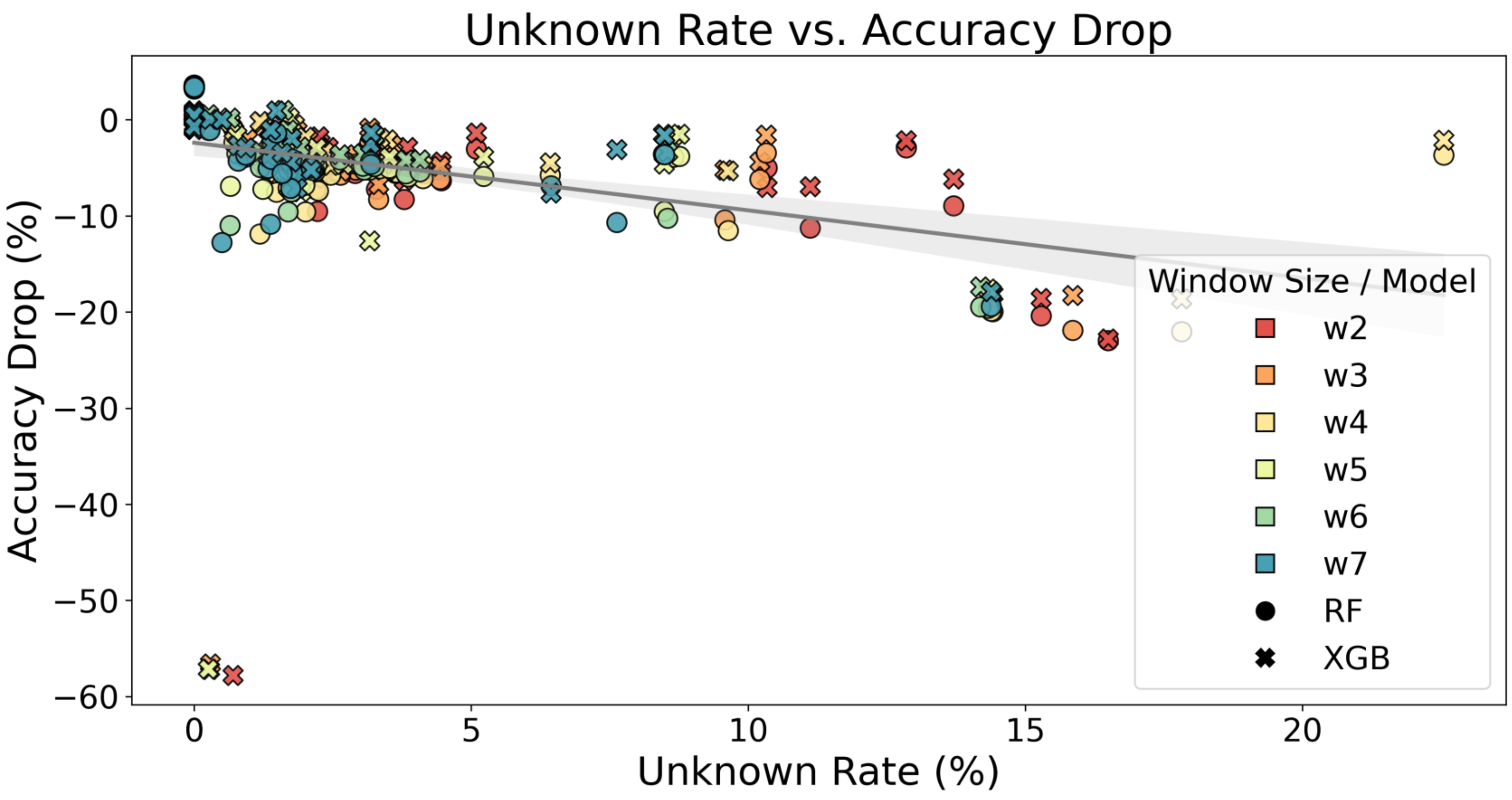
Lightweight IoT Deployment

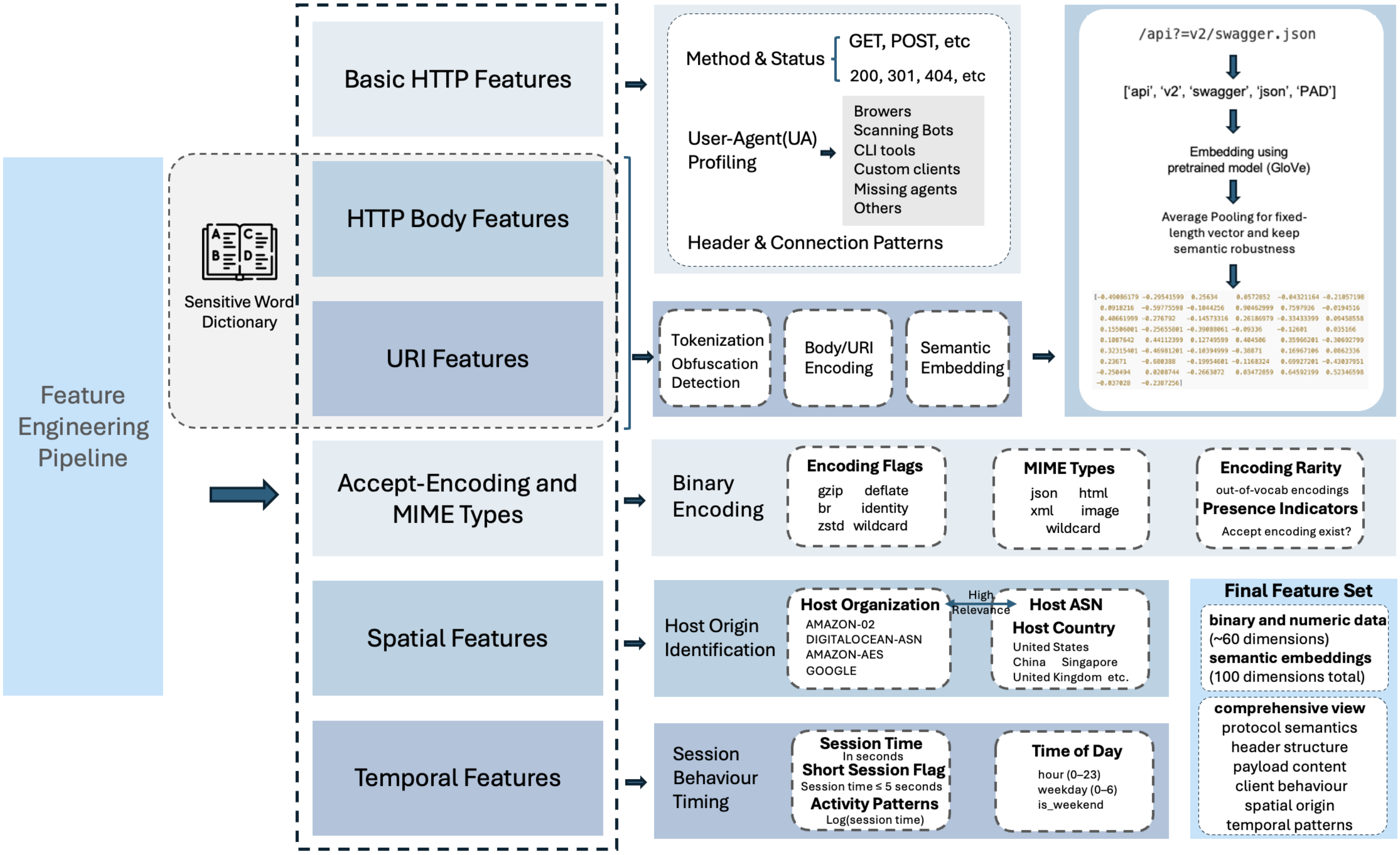
Deploy TwinGuard on IoT gateways and edge devices; Test responsiveness and overhead in resource-constrained settings

Follow us:
https://safenetiot.github.io/
https://www.youtube.com/watch?v=0fg0acuRbUA

Contact:
yuanyuan.zhou.23@ucl.ac.uk

Appendix


Unknown Rate vs. Accuracy Drop
Accuracy Drop (%)
0
−10
−20
−30
−40
−50
−60
Unknown Rate (%)
0
5
10
15
20
Window Size / Model
w2
w3
w4
w5
w6
w7
RF
XGB



Feature Engineering Pipeline
Sensitive Word Dictionary
Basic HTTP Features
HTTP Body Features
URI Features
Accept-Encoding and MIME Types
Spatial Features
Temporal Features
Method & Status
GET, POST, etc
200, 301, 404, etc
User-Agent(UA) Profiling
Browers
Scanning Bots
CLI tools
Custom clients
Missing agents
Others
Header & Connection Patterns
Tokenization Obfuscation Detection
Body/URI Encoding
Semantic Embedding
/api?=v2/swagger.json
['api', 'v2', 'swagger', 'json', 'PAD']
Embedding using pretrained model (GloVe)
Average Pooling for fixed-length vector and keep semantic robustness
Binary Encoding
Encoding Flags
gzip deflate
br identity
zstd wildcard
MIME Types
json html
xml image
wildcard
Encoding Rarity
out-of-vocab encodings
Presence Indicators
Accept encoding exist?
Host Origin Identification
Host Organization
AMAZON-02
DIGITALOCEAN-ASN
AMAZON-AES
GOOGLE
High Relevance
Host ASN
Host Country
United States
China Singapore
United Kingdom etc.
Session Behaviour Timing
Session Time
In seconds
Short Session Flag
Session time ≤ 5 seconds
Activity Patterns
Log(session time)
Time of Day
hour (0–23)
weekday (0–6)
is_weekend
Final Feature Set
binary and numeric data (~60 dimensions)
semantic embeddings (100 dimensions total)
comprehensive view
protocol semantics
header structure
payload content
client behaviour
spatial origin
temporal patterns