Large Language Models for Network Security

12/07/2023

Network Security Tasks

- Intrusion Detection
- Log Anomaly Detection
- Network Traffic Classification
- Detect BGP Hijacking Attacks
- Etc

Why LLMs?

- Network packets: the language between machines?
- Logs: the language between software?

Why LLMs?

- Network packets: the language between machines?
- Logs: the language between software?
- Very few labeled samples for attacks and anomaly
- Advantages of building on a “foundation model”?
  - Learn common “knowledge”?
  - Domain adaptation?

ET-BERT: A Contextualized Datagram Representation with Pre-training Transformers for Encrypted Traffic Classification

Lin et al., WWW'22

Traffic Encryption

- Tor, TLS, VPN, etc.
  - Protect privacy and anonymity for users
  - Cybercriminals evade surveillance

Encrypted Traffic Classification

- Detect traffic from malware
  - Mobile phone, desktop, websites, …
- Apply security policy in Enterprise settings
  - Bring your own device
- Censorship

Four Paradigms

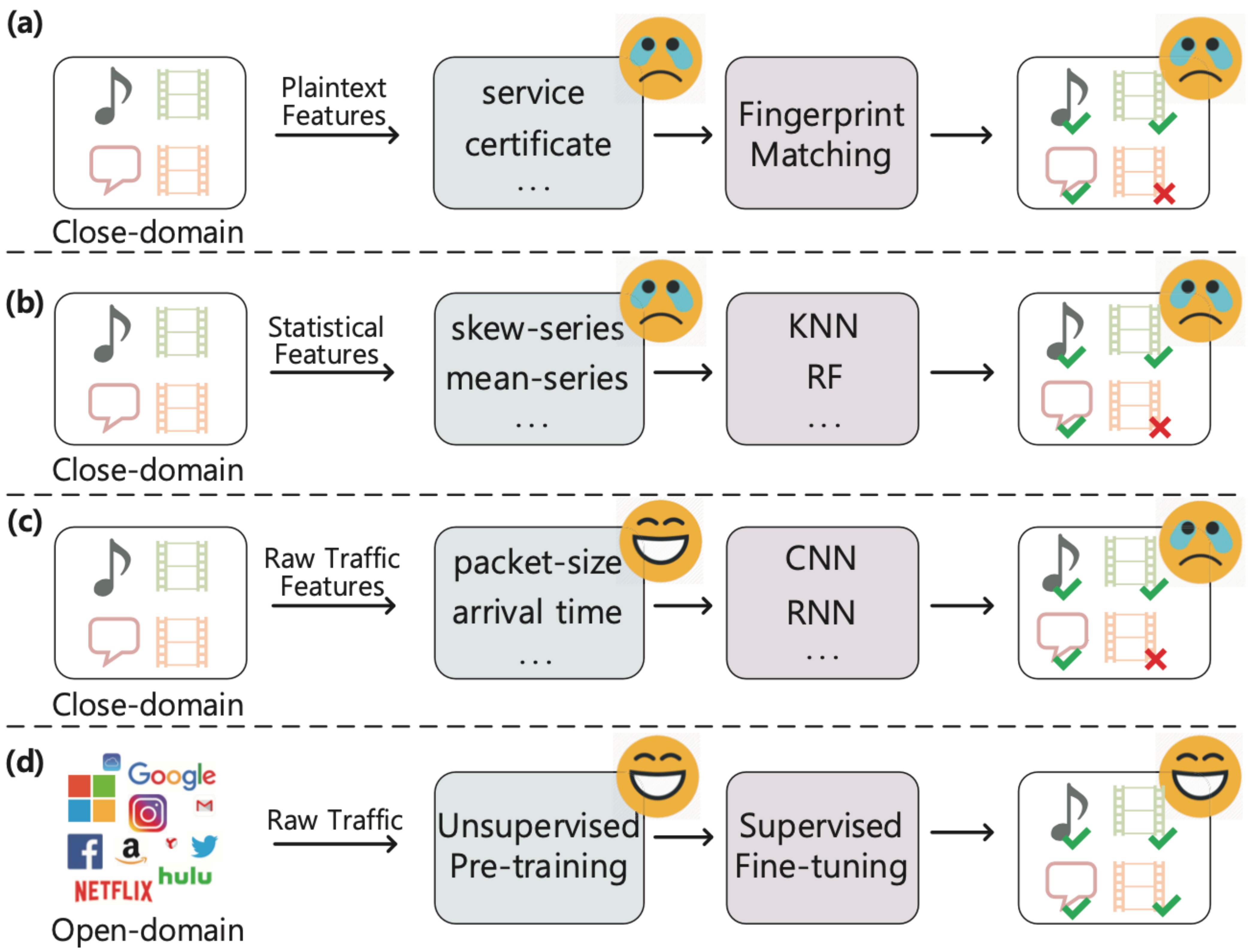


Figure 1: Four main kinds of Encrypted Traffic Classification Methods: (a) Plaintext feature based fingerprint matching. (b) Statistical feature based machine learning. (c) Raw traffic feature based ML. (d) Raw traffic based pre-training.

This Paper: Two New Pre-training Tasks

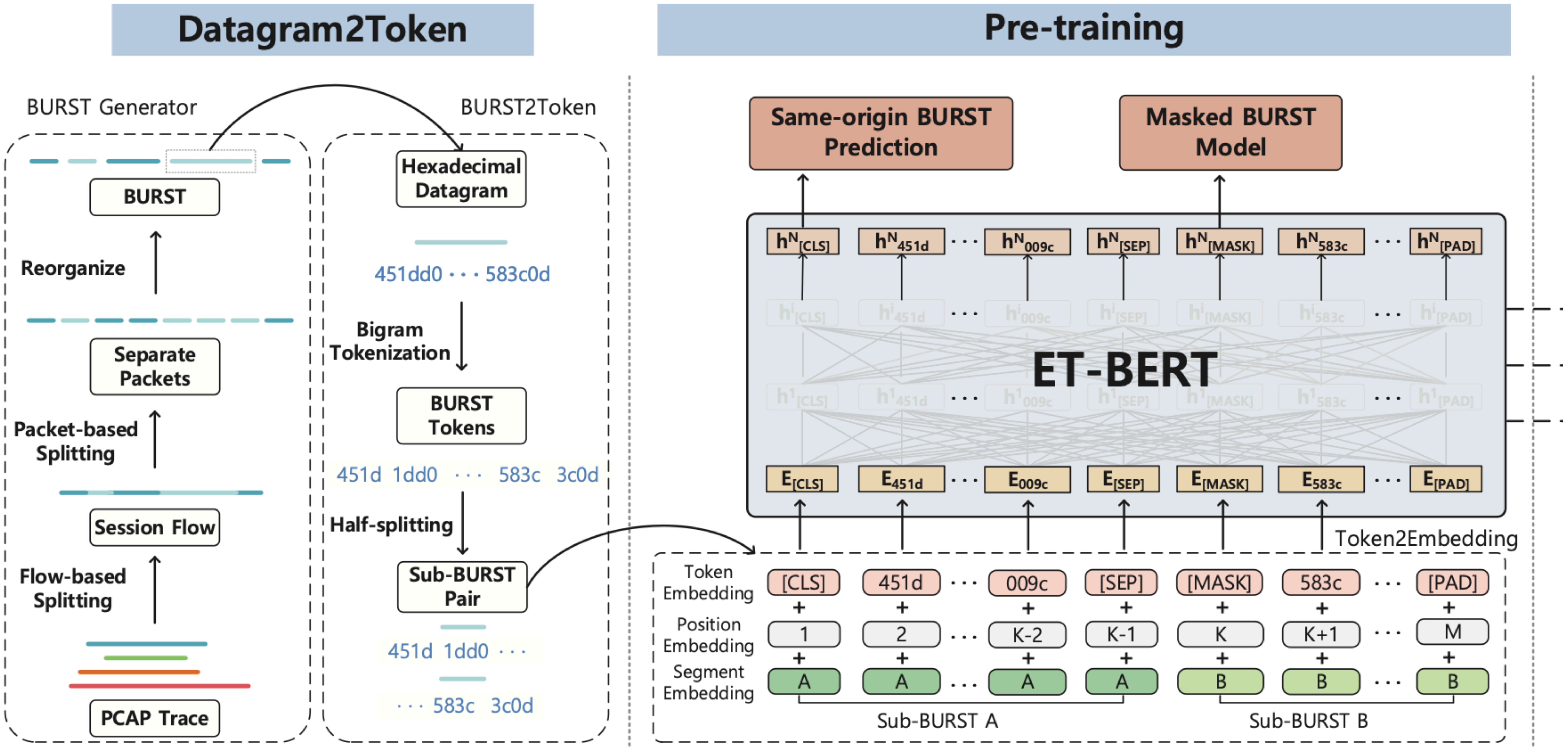
- A new notion of BURST
- ~~Masked Language Model~~ => Masked BURST Model
- Same-origin BURST Prediction

BURST

- Flow: packets p identified by (IPsrc:PORTsrc, IPdst:PORTdst, Protocol)

$$BURST = \begin{cases} B^{src} = \{p_m^{src}, m \in \mathbb{N}^+\} \\ B^{dst} = \{p_n^{dst}, n \in \mathbb{N}^+\} \end{cases}$$

Overview



Datagram2Token
Pre-training
BURST Generator
BURST2Token
BURST
Reorganize
Separate Packets
Packet-based Splitting
Session Flow
Flow-based Splitting
PCAP Trace
Hexadecimal Datagram
451dd0 · · · 583c0d
Bigram Tokenization
BURST Tokens
451d 1dd0 · · · 583c 3c0d
Half-splitting
Sub-BURST Pair
451d 1dd0 · · ·
· · · 583c 3c0d
Same-origin BURST Prediction
Masked BURST Model
ET-BERT
Token2Embedding
Token Embedding
Position Embedding
Segment Embedding
[CLS] 451d 009c [SEP] [MASK] 583c [PAD]
1 2 K-2 K-1 K K+1 M
A A A A B B B
Sub-BURST A
Sub-BURST B

Pre-Training: Masked BURST Model

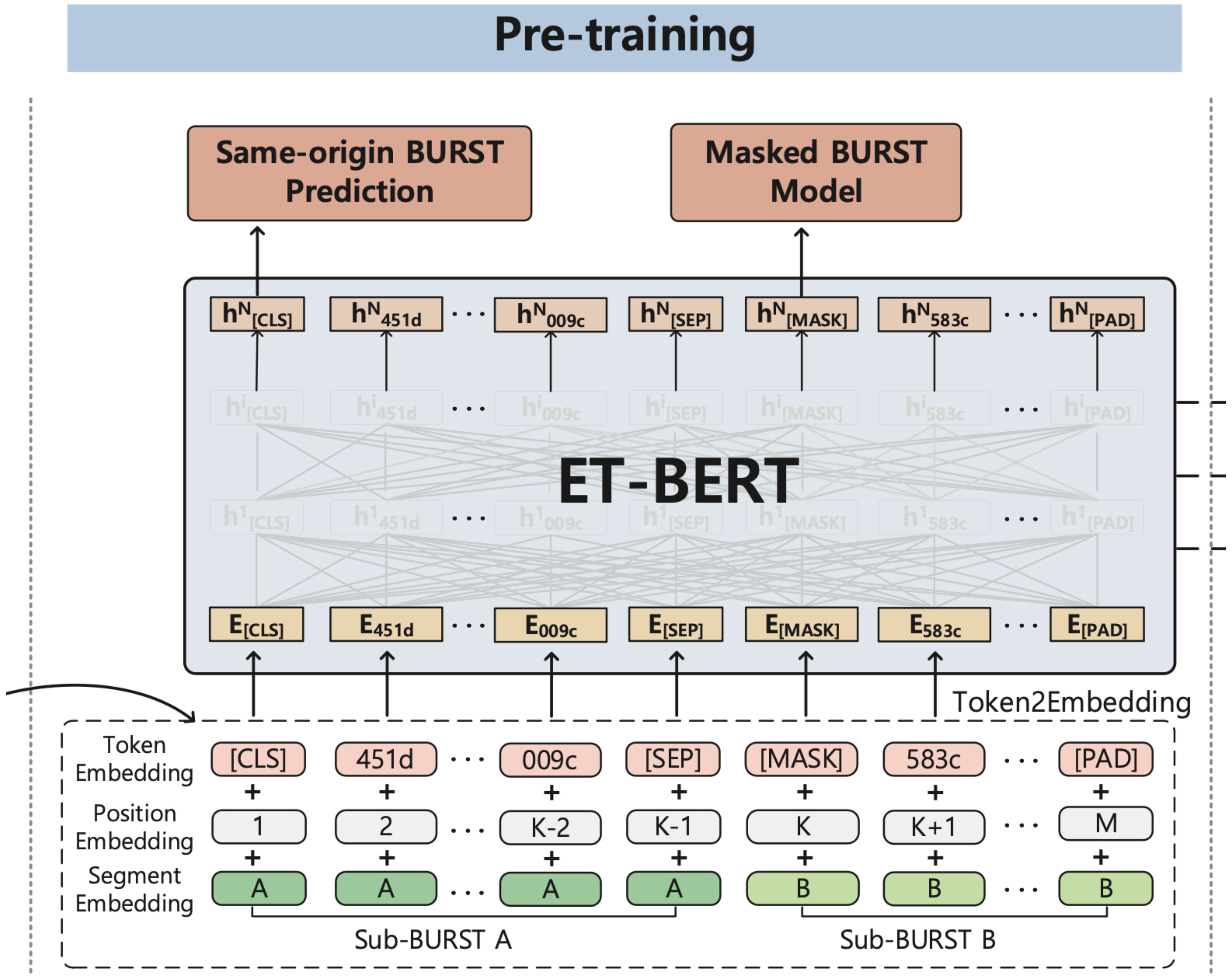
- Masked BURST Model
  - For each token, mask with 15% probability
    - If chosen, replace it with [MASK] with 80% probability
    - Choose a random token to replace it with 10% probability
    - Leave it unchanged at 10% probability
- Predict the masked tokens, minimize negative log likelihood
- Standard Masked Language Model, just the token computation is different

Pre-Training: Same-origin BURST Prediction

Different websites load packets differently,

e.g., the order of objects to load, different categories of the content to load, etc.

Pre-Training: Same-origin BURST Prediction



- 50% of times, Sub-BURST A and Sub-BURST B come from the same origin
- 50% of times, different origins

Pre-Training

- Sum of the two pre-training losses
- 30GB of unlabeled traffic data:
  - (1) ~15GB traffic from the public datasets [9, 30] (VPN Traffic, Network Intrusion Detection Dataset)
  - (2) ~15GB traffic from our passively collected traffic under their own network
- Rich common network protocols: a new encryption protocol based on UDP transport QUIC, Transport Layer Security, File Transfer Protocol, Hyper Text Transfer Protocol, Secure Shell, etc.

Fine Tuning

- Packet level, and Flow level inputs
  - Differences are not very clear to me

Task	Dataset	#Flow	#Packet	#Label
GEAC	Cross-Platform(iOS) [35]	20,858	707,717	196
	Cross-Platform(Android) [35]	27,846	656,044	215
EMC	USTC-TFC [39]	9,853	97,115	20
ETCV	ISCX-VPN-Service [9]	3,694	60,000	12
	ISCX-VPN-App [9]	2,329	77,163	17
EACT	ISCX-Tor [10]	3,021	80,000	16
EAC-1.3	CSTNET-TLS 1.3 (Ours)	46,372	581,709	120

Highlight Results

encrypted traffic classification tasks, remarkably pushing the F1 of ISCX-VPN-Service to 98.9% (5.2%↑), Cross-Platform (Android) to 92.5% (5.4%↑), CSTNET-TLS 1.3 to 97.4% (10.0%↑). Notably, we pro-

- In other datasets, the improvements are small
- In most cases, not a big difference between packet-fine-tuning vs flow-fine-tuning

Interpretation

- Different cipher implementations have varying degrees of randomness
- Some datasets use encryption algorithms with weaker randomness, so ET-BERT does better in these cases

Discussions

Can Language Models Help in System Security? Investigating Log Anomaly Detection using BERT

Almodovar et al., ACL'22

What are Log Anomalies?

- Public datasets:
  - HDFS logs: generated in a private cloud environment using benchmark workloads.
  - BGL is an open dataset of logs collected from a BlueGene/L supercomputer system at Lawrence Livermore National Labs (LLNL) in Livermore, California.
  - Thunderbird is an open dataset of logs collected from a Thunderbird supercomputer system at Sandia National Labs (SNL) in Albuquerque.
  - See examples
- Potential applications:
  - SSH logs, attacker brute force your login system

Input Differences from Previous Works

- Previous works treat each log sentence as an categorical input / one input token
- LogFiT treats logs are literally texts spoken by these systems

Main Idea

Start from BERT that learned information from language language

Do transfer learning on system log data

Anomaly Detection Paradigms

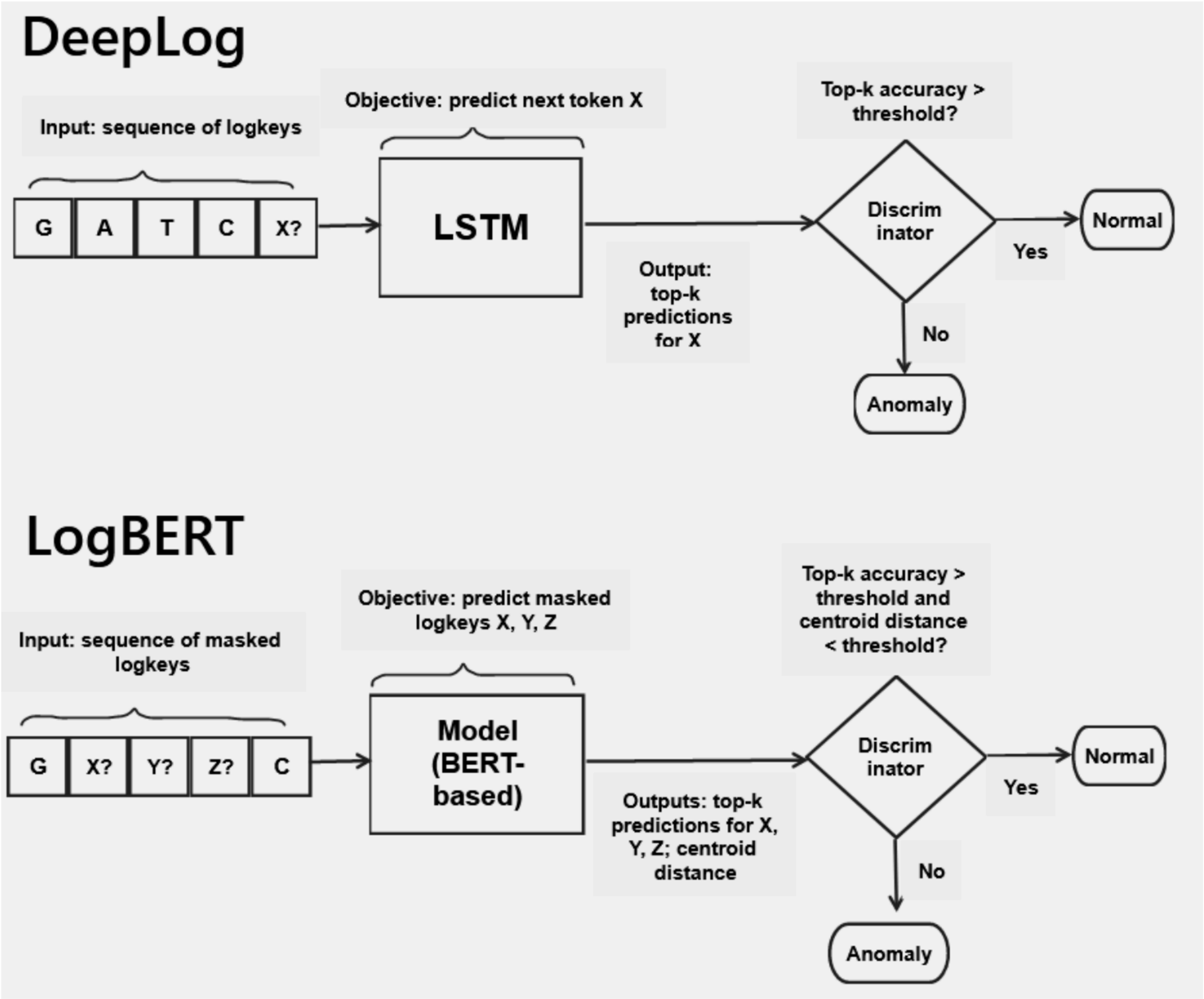


Figure 2: The DeepLog and LogBERT log anomaly detection approaches.

Fine Tuning

- Masked Language Model
- Minimize distance to some centroid

$$Loss_{cdist} = \frac{1}{b}\sum_{j=1}^{b}(CV_j - centroid)^2.$$

Results

	HDFS				BGL				Thunderbird			
Method	P	R	F1	S	P	R	F1	S	P	R	F1	S
DeepLog	100.0	60.90	75.70	100.0	90.2	70.68	79.25	98.32	65.05	99.4	78.64	89.30
LogBERT	24.02	82.80	37.24	47.62	88.92	88.35	88.63	97.59	91.75	95.7	93.69	98.28
LogFiT (ours)	99.78	90.60	94.97	99.96	98.83	84.70	91.22	99.00	89.90	98.80	94.14	97.78

Table 2: Comparison of anomaly detection effectiveness of different methods in terms of Precision (P), Recall (R), F1 score (F) and Specificity (S) on three log datasets (HDFS, BGL, Thunderbird).

- Lower S => Higher FPR

Discussions

- Unclear how the threshold is chosen
  - e.g., maintain a low FPR? High Specificity?
- ?

Why LLMs?

- Network packets: the language between machines?
- Logs: the language between software?
- Very few labeled samples for attacks and anomaly
- Advantages of building on a “foundation model”?
    - Learn common “knowledge”?
    - Domain adaptation?

Discussions

- Other Network Security Tasks?

Final Project Report

- Problem Statement
- Related Work
- Method
- Results
- Takeaway and Lessons Learned