NRC·CNRC

NRC.CANADA.CA

Provenance Graph-Based Cyber Threat Detection

Shahrear Iqbal

Research Officer, Cyber Security, DT

National Research Council Canada

National Research Council Canada
Conseil national de recherches Canada

Canada

National Research Council

- **The National Research Council of Canada (NRC) is Canada's largest federal research and development organization.**

Our vision

A better Canada and world through excellence in research and innovation.

Our mission

To have an impact by advancing knowledge, applying leading-edge technologies, and working with other innovators to find creative, relevant and sustainable solutions to Canada's current and future economic, social and environmental challenges.

Brief Intro about Myself

- **Finished my Ph.D. in Smart Devices Operating System Security**
- **Worked in a private security company in Toronto**
  - Automatic security compliance – autonomous vehicles
- **Joined NRC (2019)**

Notable Projects

- UAV system security (with Polytechnique Montreal and UWaterloo)
- Unsupervised attack detection in large-scale network logs (with CSE)
- Smart-home based aging care system security (Queen’s)
- Robust and resilient machine learning algorithms for digital health (UBC)
- Advanced persistent threat (APT) detection in Industrial IoT systems (UNB)

How it Began

- **Unsupervised attack detection in large-scale network logs**
  - Task from Communications Security Establishment (CSE), Federal Government of Canada
  - Perform anomaly detection based on host-based telemetry coming from thousands of hosts of a government agency

The Task

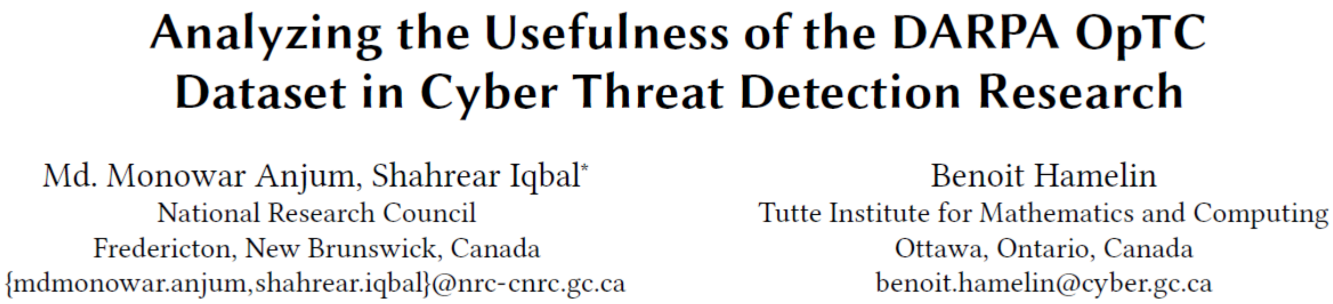
- Perform anomaly detection based on host-based telemetry coming from thousands of hosts of a government agency

- Requirements
  - Research should focus on learning representations that lend themselves to anomaly detection
  - The approach should take into account the temporal aspect of telemetry

The Dataset: DARPA OpTC

- **Recommended by CSE**
- **Published in 2020**
- **Largest Dataset Till Date by DARPA**
- **Have 17 billion+ events captured over 7 days**
  - Contains event logs from windows systems only
- **Red team APT activities include malicious power shell shellcode injection, supply chain attack and malicious updates.**

We analyzed the DARPA OpTC dataset and published a paper in ACM SACMAT 2021



Analyzing the Usefulness of the DARPA OpTC Dataset in Cyber Threat Detection Research

Md. Monowar Anjum, Shahrear Iqbal*
National Research Council
Fredericton, New Brunswick, Canada
{mdmonowar.anjum,shahrear.iqbal}@nrc-cnrc.gc.ca

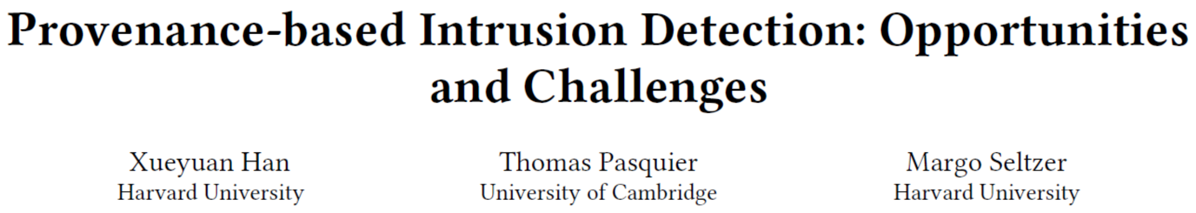
Benoit Hamelin
Tutte Institute for Mathematics and Computing
Ottawa, Ontario, Canada
benoit.hamelin@cyber.gc.ca

The Requirement

- Requirements
  - Research should focus on learning representations of heterogeneous telemetry that lend themselves to anomaly detection
  - The approach should take into account the temporal aspect of telemetry

Provenance Graphs

- Recent works suggested that system provenance graphs are an effective data source for anomaly detection.



Provenance-based Intrusion Detection: Opportunities and Challenges

Xueyuan Han
Harvard University

Thomas Pasquier
University of Cambridge

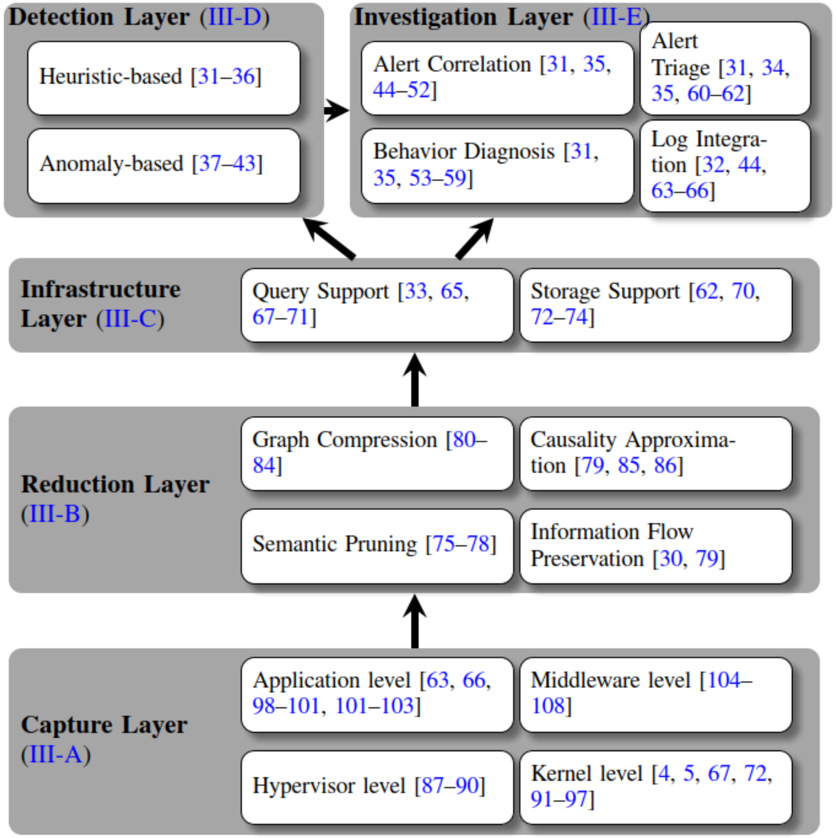
Margo Seltzer
Harvard University

- A system provenance graph is a directed acyclic graph (DAG) that represents causal relationships between running processes and objects (e.g., files, network flow, threads) in a system.

- It can connect events that are temporally distant but causally related.

- Provenance graphs provide rich contextual information regarding an event's neighborhood and it's parent events.

Provenance-based System Pipeline



Detection Layer (III-D)

- Heuristic-based [31–36]
- Anomaly-based [37–43]

Investigation Layer (III-E)

- Alert Correlation [31, 35, 44–52]
- Alert Triage [31, 34, 35, 60–62]
- Behavior Diagnosis [31, 35, 53–59]
- Log Integration [32, 44, 63–66]

Infrastructure Layer (III-C)

- Query Support [33, 65, 67–71]
- Storage Support [62, 70, 72–74]

Reduction Layer (III-B)

- Graph Compression [80–84]
- Causality Approximation [79, 85, 86]
- Semantic Pruning [75–78]
- Information Flow Preservation [30, 79]

Capture Layer (III-A)

- Application level [63, 66, 98–101, 101–103]
- Middleware level [104–108]
- Hypervisor level [87–90]
- Kernel level [4, 5, 67, 72, 91–97]

Fig. 3: We systematize provenance-based system auditing literature based on a taxonomy of the log capture and analysis pipeline.

Inam, M. A., Chen, Y., Goyal, A., Liu, J., Mink, J., Michael, N., ... & Hassan, W. U. (2022, October). SoK: History is a Vast Early Warning System: Auditing the Provenance of System Intrusions. In *2023 IEEE Symposium on Security and Privacy (SP)* (pp. 307-325). IEEE Computer Society.

Issues to Solve

1. **Provenance aware log generation**
2. **Standardization of logs**
3. **Scalability**
4. **Reducing false positives**
5. **Alert investigation**
6. **Actionable report generation**
7. **Explainability of AI decision**

ANUBIS

A Supervised Approach for Advanced Persistent Threat Detection that can Explain Its Decision

Student: Md Monowar Anjum, Master's Student, University of Manitoba

Now: PhD student, University of British Columbia (UBC)

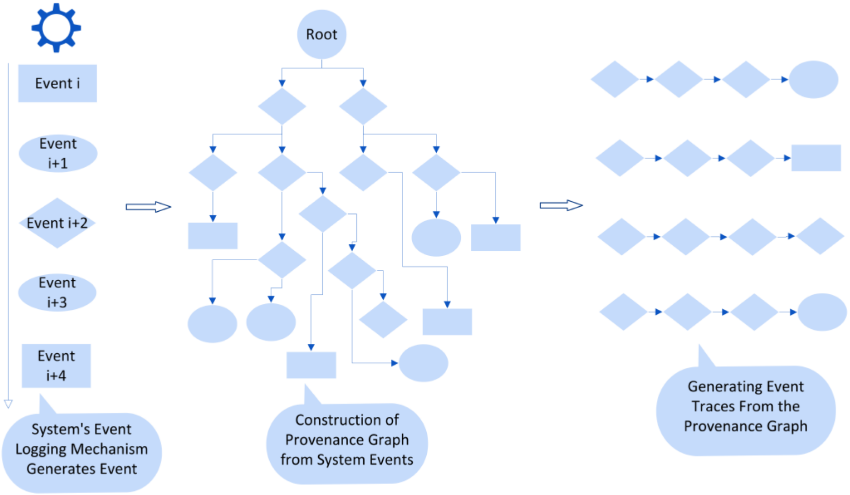
Threat Model / Assumptions

- There is a secure and trusted system level event logging mechanism
- There are multiple hosts in the system which can be connected to a heterogenous network of hosts.
- An outsider adversary attempts to gain access to the hosts via stealthy apt attack.
- No matter how stealthy the APT attack is, the behavior is sufficiently different from the normal program behavior so that statistical models can distinguish between APT and normal program.

ANUBIS Design

1. Detect Advanced Persistent Threat activities from the system state analysis.
2. Reconstruct the attack story and provide actionable intelligence to the analyst for triage.

- We used provenance graphs to represent the system state
- We generated system event traces by performing random walks on the graph


Event i
Event i+1
Event i+2
Event i+3
Event i+4
System's Event Logging Mechanism Generates Event
Root
Construction of Provenance Graph from System Events
Generating Event Traces From the Provenance Graph

Design Questions

- 1. How do we encode the traces numerically?

- 2. How to use the encoded information for the downstream algorithm?

1. Their Contextual Information (i.e., name, privilege level, access pattern, access location etc.)
2. Their Causality Information (i.e., identity of parent process or grand-parent process etc.)
3. The temporal difference between events
4. All those information has to be encoded in a fixed length vector.

Design Question 1: What to Encode?

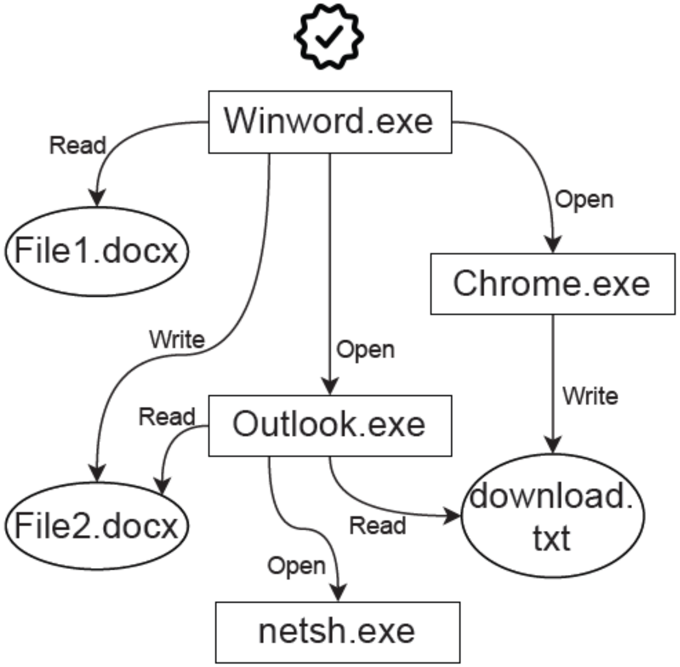
1. **Everything represented in “character strings” and not “transient” (i.e., stays same across the system execution period) are given unique indices and then normalized.**

2. **Example: {0: “netsh.exe”, 1:”chrome.exe”, 2: “windowservice.exe”}**

- After the data is encoded in floating point vectors, we designed a feed forward neural network and trained using the OpTC dataset
- The result was very bad, we could hardly reach 80% accuracy

Design Question 1: What to Encode?

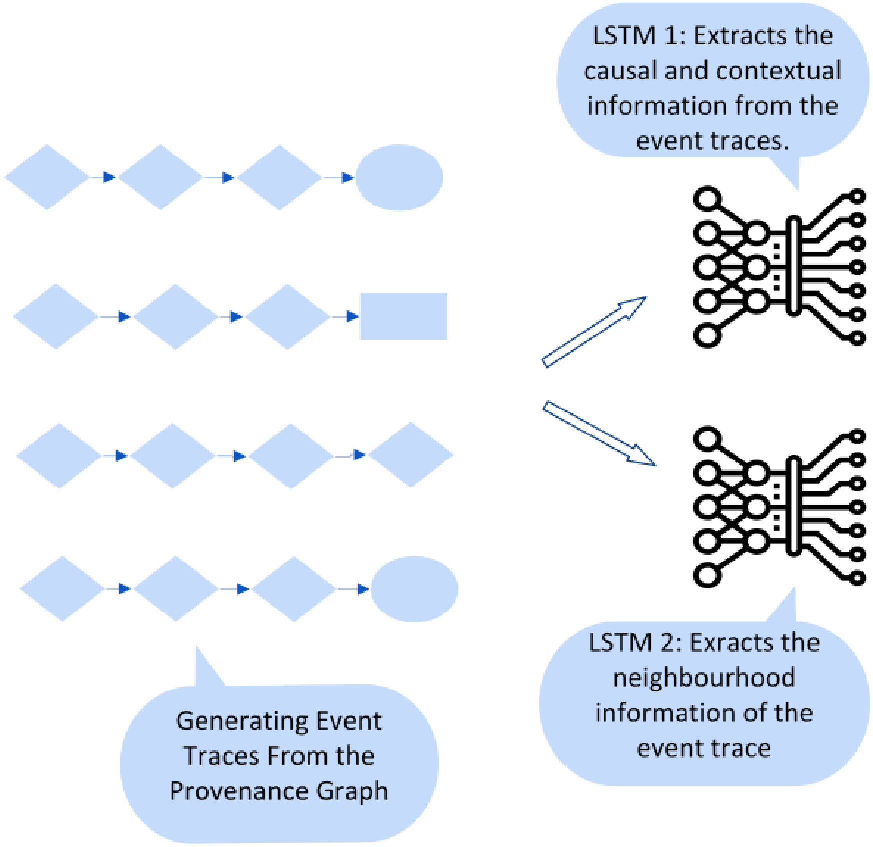
1. We introduce a concept called neighborhood
2. Information about previous neighborhood events (time delta) and possible future events in that neighborhood is encoded using Poisson distribution
3. The fixed floating point vectors are probability values derived from the Poisson distribution



Design Question 2: How to use encoded data?

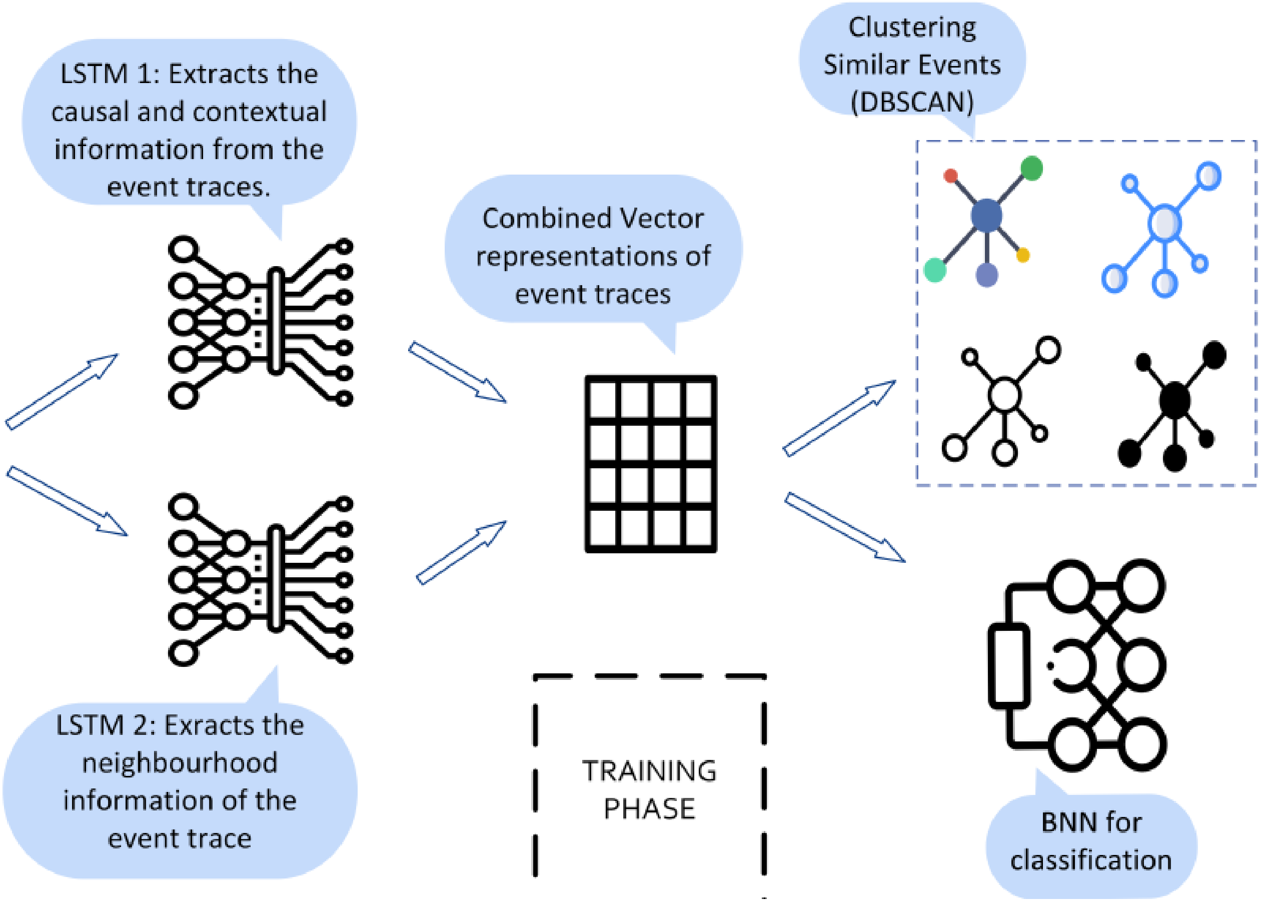
After the data is encoded in floating point vectors, we need to extract the causal relationships from them and use them in a classification model. We do this in a two tier approach:

1. Extracting the causal relationships by using RNN/LSTM layers.


LSTM 1: Extracts the causal and contextual information from the event traces.
Generating Event Traces From the Provenance Graph
LSTM 2: Exracts the neighbourhood information of the event trace

Design Question 2: How to use encoded data?

1. Concatenate the output from LSTM's and feed them in Bayesian Neural Network for classification.



LSTM 1: Extracts the causal and contextual information from the event traces.

LSTM 2: Exracts the neighbourhood information of the event trace

Combined Vector representations of event traces

Clustering Similar Events (DBSCAN)

BNN for classification

TRAINING PHASE

Design Question 2: How to use encoded data?

Why Bayesian Neural Network?:

1. Standard Neural Networks can not quantify uncertainty in the prediction. Bayesian Neural Nets makes a prediction and also we can score the confidence in the prediction. This is useful in unknown APT detection scenario.
2. This is a novel approach. Probabilistic Neural Nets have not been used in APT detection before.

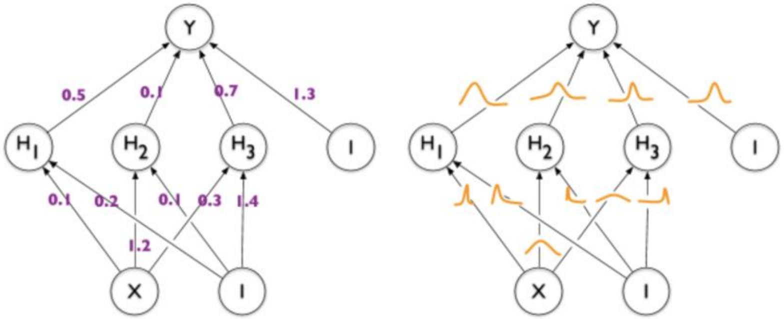


Figure : Comparison between Classical Neural Network and Bayesian Neural Network. Classical NN learns fixed set of weights while BNN learns a distribution of weights for the task at hand.

Design Question 2: How to use encoded data?

How to use uncertainty in prediction?:

1. If a prediction is highly certain then we find out the most similar training example and report that to the cyber-analyst.

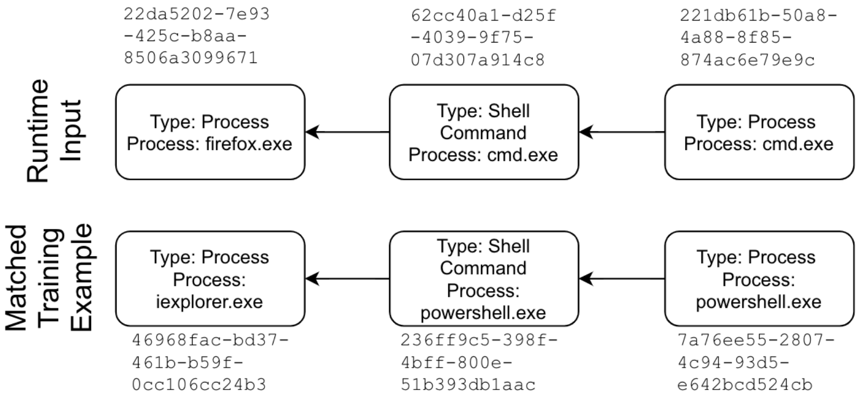


Figure : Explaining prediction with low Uncertainty

Design Question 2: How to use encoded data?

How to use uncertainty in prediction (Continued)?:

1. If a prediction is reported as less certain then we find out the most similar cluster of training examples and report that to the cyber-analyst.

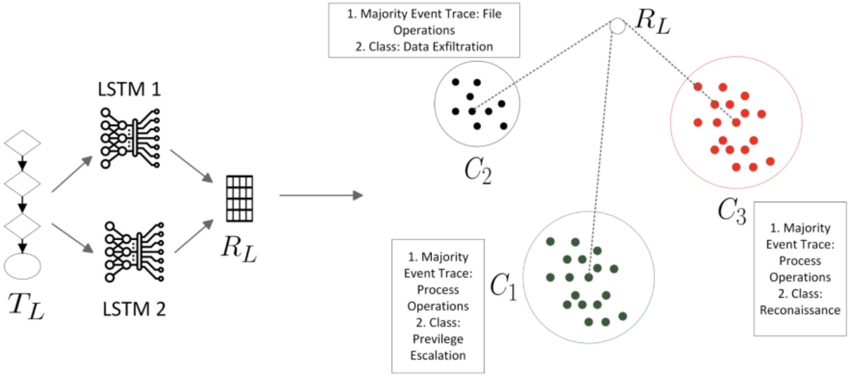
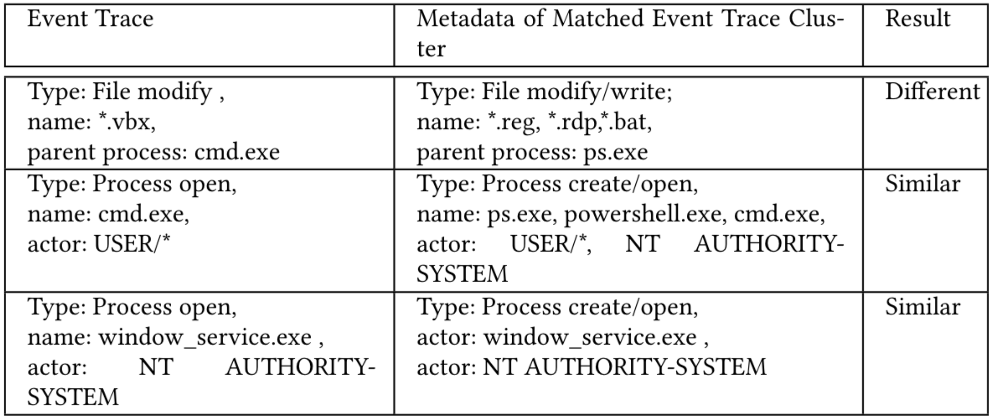

LSTM 1
LSTM 2
T_L
R_L
R_L
1. Majority Event Trace: File Operations
2. Class: Data Exfiltration
C_2
C_3
C_1
1. Majority Event Trace: Process Operations
2. Class: Previlege Escalation
1. Majority Event Trace: Process Operations
2. Class: Reconaissance

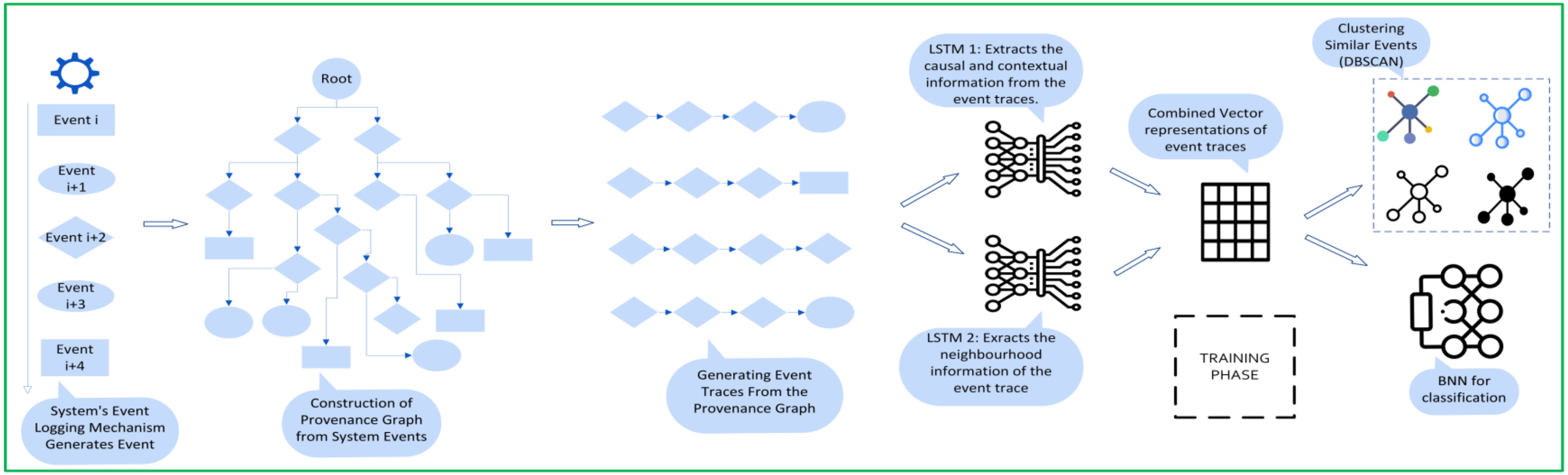
Figure : Process followed by ANUBIS to explain low certainty malicious prediction. Distance between R_L and C_2 is the least. Therefore, the metadata of C_2 is presented in the prediction report of T_L.

Table : Explaining prediction with high uncertainty



Event Trace	Metadata of Matched Event Trace Cluster	Result
Type: File modify , name: *.vbx, parent process: cmd.exe	Type: File modify/write; name: *.reg, *.rdp,*.bat, parent process: ps.exe	Different
Type: Process open, name: cmd.exe, actor: USER/*	Type: Process create/open, name: ps.exe, powershell.exe, cmd.exe, actor: USER/*, NT AUTHORITY-SYSTEM	Similar
Type: Process open, name: window_service.exe , actor: NT AUTHORITY-SYSTEM	Type: Process create/open, actor: window_service.exe , actor: NT AUTHORITY-SYSTEM	Similar

Proposed Solution: ANUBIS DESIGN



Event i

Event i+1

Event i+2

Event i+3

Event i+4

System's Event Logging Mechanism Generates Event

Root

Construction of Provenance Graph from System Events

Generating Event Traces From the Provenance Graph

LSTM 1: Extracts the causal and contextual information from the event traces.

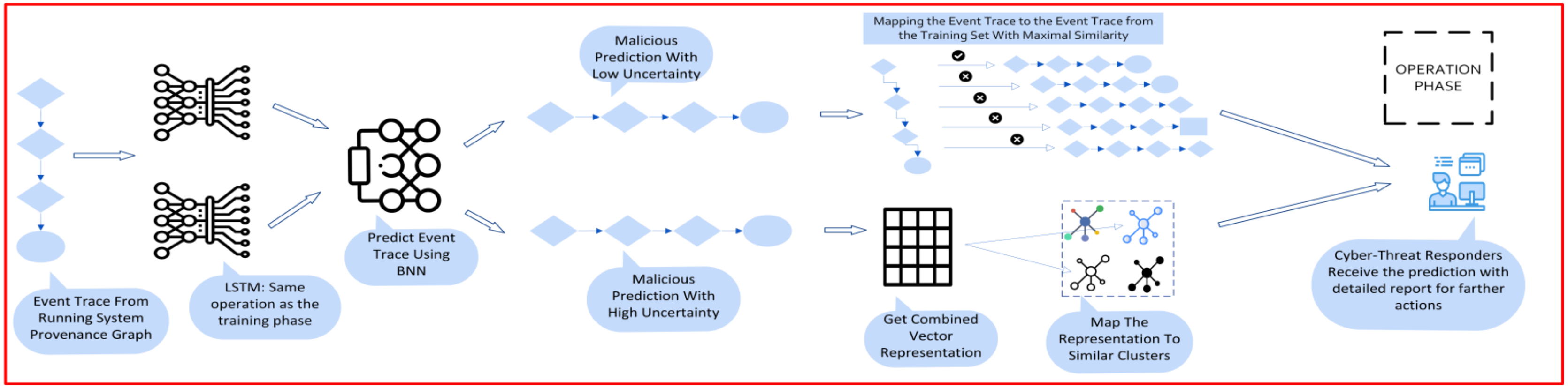
LSTM 2: Exracts the neighbourhood information of the event trace

Combined Vector representations of event traces

TRAINING PHASE

Clustering Similar Events (DBSCAN)

BNN for classification



Event Trace From Running System Provenance Graph

LSTM: Same operation as the training phase

Predict Event Trace Using BNN

Malicious Prediction With Low Uncertainty

Malicious Prediction With High Uncertainty

Mapping the Event Trace to the Event Trace from the Training Set With Maximal Similarity

Get Combined Vector Representation

Map The Representation To Similar Clusters

OPERATION PHASE

Cyber-Threat Responders Receive the prediction with detailed report for farther actions

Results

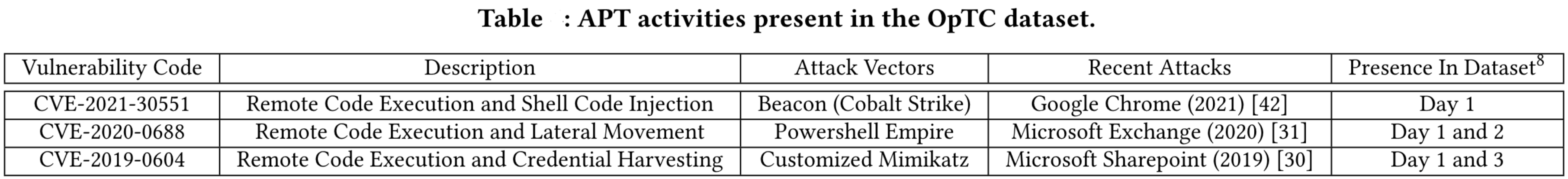


Table : APT activities present in the OpTC dataset.

Vulnerability Code	Description	Attack Vectors	Recent Attacks	Presence In Dataset[8]
CVE-2021-30551	Remote Code Execution and Shell Code Injection	Beacon (Cobalt Strike)	Google Chrome (2021) [42]	Day 1
CVE-2020-0688	Remote Code Execution and Lateral Movement	Powershell Empire	Microsoft Exchange (2020) [31]	Day 1 and 2
CVE-2019-0604	Remote Code Execution and Credential Harvesting	Customized Mimikatz	Microsoft Sharepoint (2019) [30]	Day 1 and 3

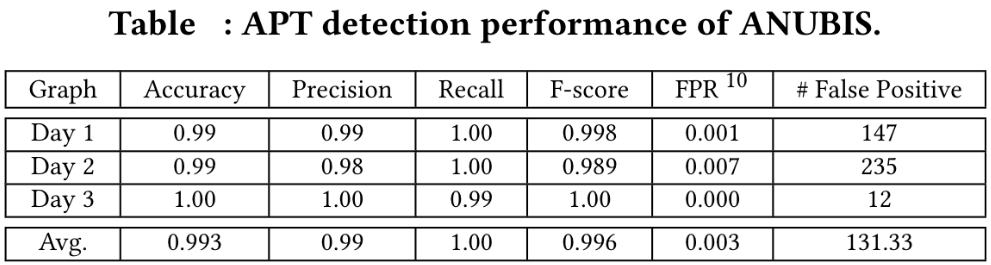


Table : APT detection performance of ANUBIS.

Graph	Accuracy	Precision	Recall	F-score	FPR [10]	# False Positive
Day 1	0.99	0.99	1.00	0.998	0.001	147
Day 2	0.99	0.98	1.00	0.989	0.007	235
Day 3	1.00	1.00	0.99	1.00	0.000	12
Avg.	0.993	0.99	1.00	0.996	0.003	131.33

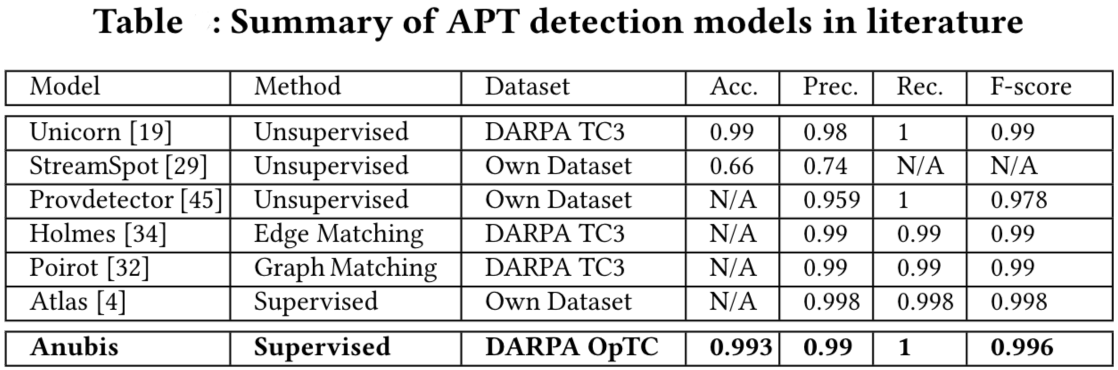


Table : Summary of APT detection models in literature

Model	Method	Dataset	Acc.	Prec.	Rec.	F-score
Unicorn [19]	Unsupervised	DARPA TC3	0.99	0.98	1	0.99
StreamSpot [29]	Unsupervised	Own Dataset	0.66	0.74	N/A	N/A
Provdetector [45]	Unsupervised	Own Dataset	N/A	0.959	1	0.978
Holmes [34]	Edge Matching	DARPA TC3	N/A	0.99	0.99	0.99
Poirot [32]	Graph Matching	DARPA TC3	N/A	0.99	0.99	0.99
Atlas [4]	Supervised	Own Dataset	N/A	0.998	0.998	0.998
Anubis	**Supervised**	**DARPA OpTC**	**0.993**	**0.99**	**1**	**0.996**

The Task

- Perform anomaly detection based on host-based telemetry coming from thousands of hosts of a government agency

- Requirements
  - Research on anomaly detection should be performed against labeled datasets
  - **Research should focus on learning representations that lend themselves to anomaly detection**
  - The approach should take into account the temporal aspect of telemetry

A MEMORY-EFFICIENT APT HUNTING SYSTEM BASED ON ATTACK REPRESENTATION LEARNING

Ahmed Aly

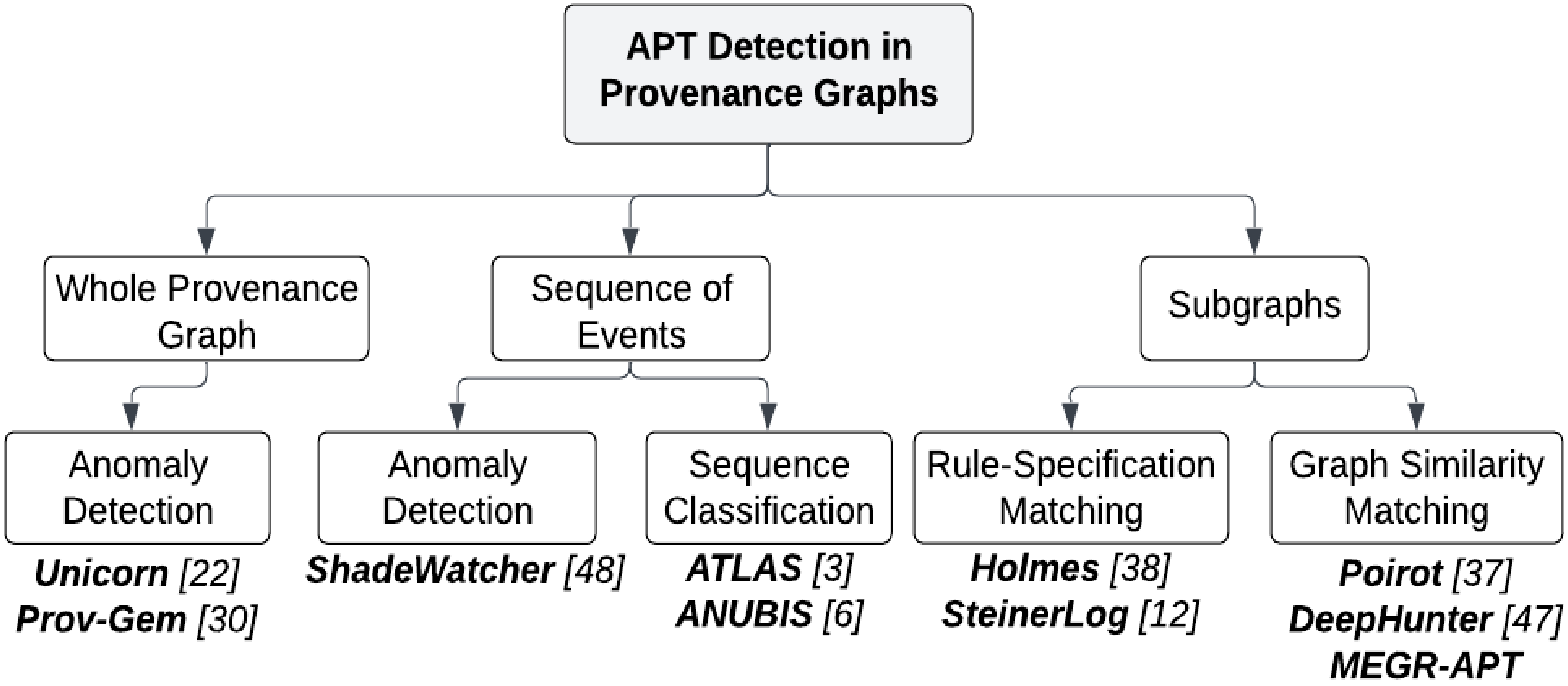
PhD Student, Concordia University

Hunting APT

- **Cyber threat hunting is the process of identifying threats and ongoing attacks by proactively searching for indicators of compromise undetected in the system.**
    - It aims to uncover hidden traces to limit the harm and spread of a specific attack scenario.
- **Once a new attack is discovered, security experts identify the attack's main characteristics and release the attack scenario in Cyber Threat Intelligence (CTI) reports.**
    - Each attack scenario shows **Indicators of Compromises** (IOCs) and strategies related to the attack.
    - An attacker could mutate individual IOCs, but it is harder to mutate the overall attack scenario, including all its IOCs with each new victim.
- **The threat-hunting task becomes more critical when searching for sophisticated, widespread attacks such as Advanced Persistent Threats (APT).**
    - In some cases, APT attacks use a **"low and slow"** approach to stay **undetected for months** or even years.

Related Works

Here we categorize the related works based on different provenance graph granularity and APT detection method.



Examples of Query Graphs and Detected Subgraphs

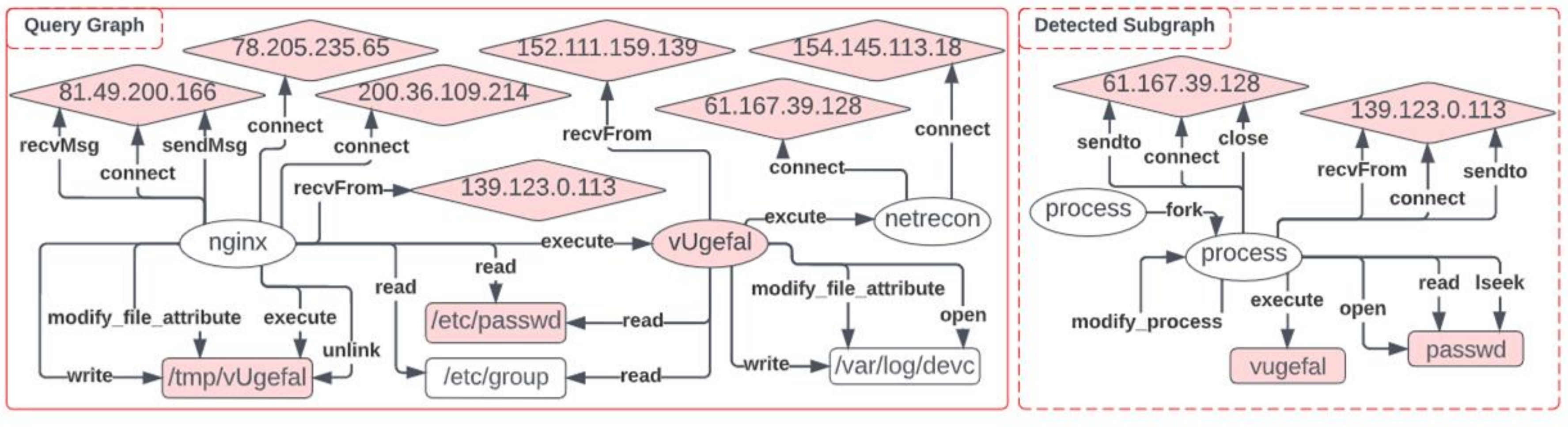


Figure 7: Query graph of the TC3 BSD 1 scenario (on the left) and its detected subgraph (on the right).

Research Direction

- **This research addresses the problem of discovering at scale suspicious subgraphs matching an attack scenario (query graph) recently published in CTI reports.**
- **There is a need for efficient APT hunting systems that scale to large PGs while using limited memory and uncover attacks in a few minutes.**
- **The goal is to hunt APTs in a twofold process:**
    - memory-efficient suspicious **subgraphs extraction**, and
    - fast **subgraph matching** based on graph representation learning.
- **The twofold process balances the trade-off between time and memory efficiency.**

Questions??

Shahrear Iqbal • Research Officer • Shahrear.Iqbal@nrc-cnrc.gc.ca