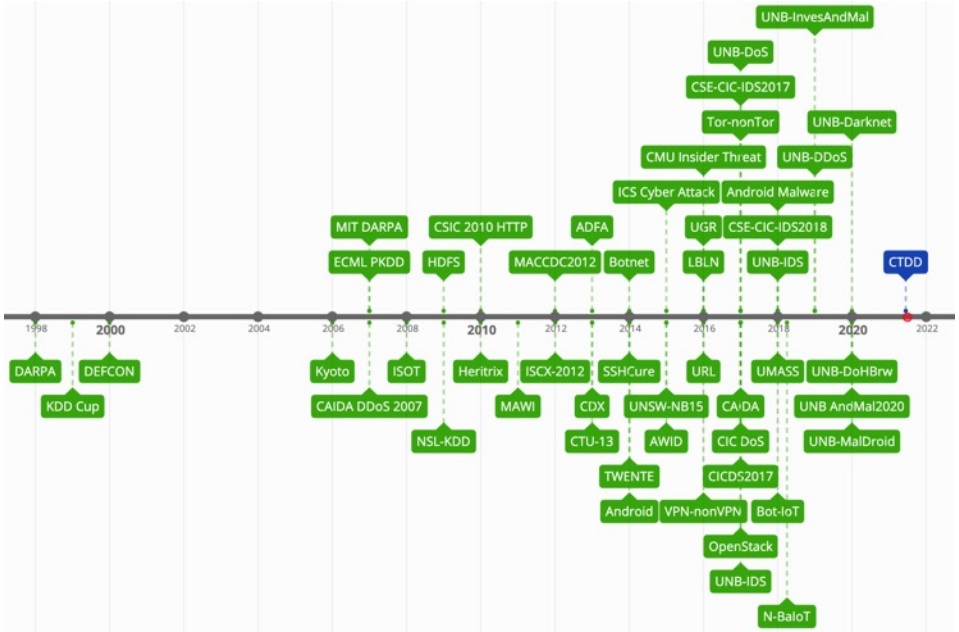
Deep Reasoning Engine to Identify Complex Cyber Threat Thru Human-AI Teaming

Paul Rad, Ph.D.
Associate Professor
Cyber Infrastructure and AI
Sr. Member of the National Academy of Inventors (NAI)
Director Secure AI and Autonomy Lab
Co-Founder Open Cloud Institute (OCI)
The University of Texas at San Antonio
Phone: 210.872.7259

Motivation

- Zero-day attacks observed in edge computing systems and edge devices are used to exploit publicly unknown vulnerabilities often found in new protocols and applications.
- Existing datasets are generated and published with no plans for adding new data representing novel attacks or enabling other research groups to add new data using the similar setups.
- Current solutions often fail to provide the means to integrate existing testbeds and cyber ranges.
- Current solutions often fail to provide modules to enable multiple research groups to join efforts to produce new datasets or extend existing ones.



References: [69-72]

Problem Statement

Detection of system anomalies is a critical task for ensuring the security and reliability of systems. Traditional anomaly detection solutions are signature-based. Although signature-based approach is highly effective for known attacks, these methods are ineffective for custom new attacks and involve significant lead time since attack patterns need to be detected, studied, isolated, reported and disseminated.

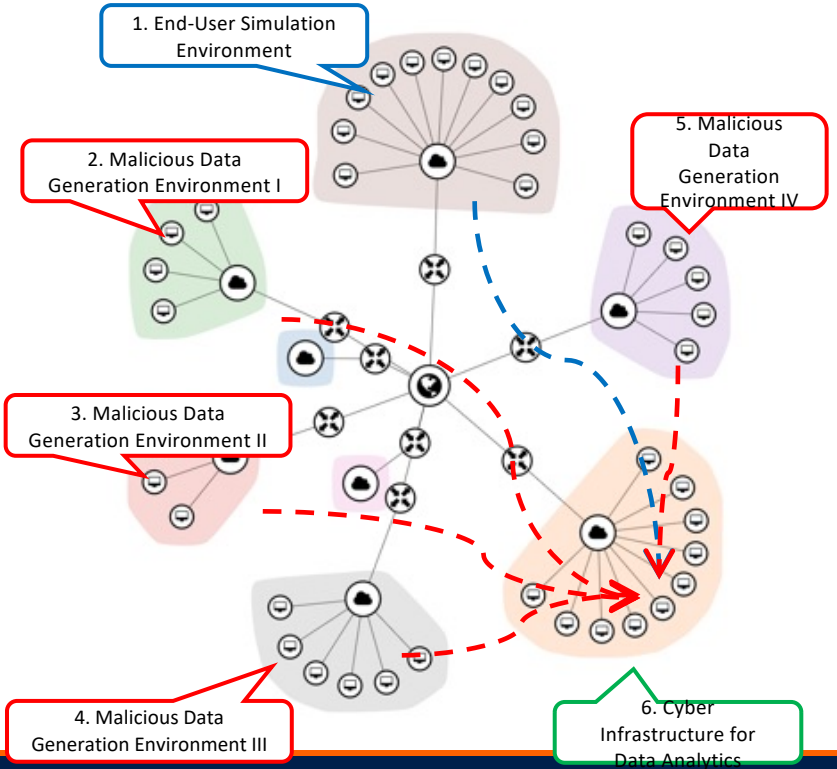
Unrestricted Content
This document does not contain technology or technical data controlled under either the U.S. International Traffic in Arms Regulations or the U.S. Export Administration Regulations.

AIMs

- Deployment of Cloud Cyber Analytics Testbed for Cybersecurity Threat Analysis and Experimentation
- Cybersecurity Infrastructure Attack Simulation and Data Collection for Synthetic Data Generation
- Anomaly Detection using Explainable Deep Model with Learning Models with Compositional Temporal Attention
- Additional Aim recommended by Mr. Lynn Ginader
  - Development of applications for mixed reality glasses designed for immersive situational awareness that combine virtual and physical objects in a real environment where the user can visualize network intrusion detection data in real-time.

Unrestricted Content
This document does not contain technology or technical data controlled under either the U.S. International Traffic in Arms Regulations or the U.S. Export Adm

Training Data Collection Infrastructure



- Cyber Analytics Testbed for Cybersecurity Threat Analysis and Experimentation is composed by multiple environments deployed into educational Cloud.
- End-User Environment (1) hosts software development platform
- Malicious Data Generation Environments (2,3,4,5) hosts frameworks for attack scenario generation
- Cyber Infrastructure for Data Analytics Environment (6) hosts FOSS suite to help a Cybersecurity team search, analyze and visualize data in real time.
- Environments 1-5 host lightweight agents that collect and centralize log, network, metric, and audit data. All data is forwarded to processing pipeline for processing ,enrichment and indexing.
- Indexed data is used by analysts to detect threats, triage events and perform investigation using available SIEM, visualization tools, and L1 Machine Learning models.
- Deep learning developers extract structured and raw unstructured data for generating advanced deep learning models.

Synthetic Data Generation – normal behavior

- Generated 3 end-user environments --namely, IS7033 2020, ITESM 2020, and Practicum 2020 for simulating a non-malicious environment.
- System logs, Zeek Logs, network data, audit data, and system metrics were collected from each environment by the pre-installed and pre-configured light-weight agents.
- A total of 2,199,491,503 data points were collected from the deployed end-user environments consisting of 62 VMs (10 for Practicum 2020, 12 for IS7033 2020, and 40 for ITESM 2020).

Data Collected from End-User Environments

Data Log Source	**# Collected Datapoints**
Audit Logs	339,571,785
MySQL Logs	3,427
System Logs (Unstructured)	60,913,188
System Logs (Structured)	47,383,999
Zeek Logs - ITESM 2020	28,123,364
System Metrics	268,848,320
Network Data	1,454,647,420
Total	**2,199,491,503**

Synthetic Data Generation
Abnormal behavior – attack

- Generated 22 experiments for executing 16 malicious software samples and 6 cyber kill chains.
- A total of 3,758,343 datapoints were collected from experiments generated in the adversarial environments.
- System logs, Zeek Logs, network data, audit data, and system metrics were collected from each environment by the pre-installed and pre-configured light-weight agents.

Data Log Source	# Collected Datapoints
CKC - Zeek	1,417,066
Malicious Software Samples	2,341,277
Total	**3,758,343**

Dat

Attack Case	Description of Scenario	References	# of logs
GonnaCry	GonnaCry is an academic Ransomware program that allows users to infect a client to encrypt files, peripheral devices, and destroy original files. It does not have all the features of WannaCry2.0	S0366	129
ech0raix	A ransomware family that targets QNAP Network Attached Storage (NAS) devices. Devices are compromised by bruteforce attacks or exploiting known vulnerabilities. The ransomware executes a malicious payload that encrypts targeted file extensions on the NAS.	T1486	27
ACK Flood	An attack that exhausts OS finite TCP connections by sending a flood of ACK packets for non existing connections and leveraging the stateful nature of the TCP protocol.	T1499.001	114
NTP DDoS Amplification	A DoS Reflection Amplification Attack. The attack send packets to a third-party server with the spoofed address of a target.	T1498	406
SYN Flood	An attack that exhausts OS finite TCP connections by sending excessive amounts of SYN packets where the 3-way TCP handhake is never completed	T1499.001	132
malaria	An injection attack that injects malicious code via ptrace system calls. The trace system call injection is usually executed by writing arbitrary code into a running process.	T1055-008	182
nemox	A half-virus for infecting any ELF files in a specific directory.	T1027-001	109
nf3ct0r	A virus for infecting ELF files. Also know as an ELF infector.	T1027-001	135
utrojan	Universal trojan for accessing an unauthorized system.	T1036-004	221
Lin Blackhole	A malicious program created using C programming language which on infecting the victim client provides a backdoor to the attacker.	T1587-001 and T1588-001	74
Lin Ovason	A malicious program created using C programming language which on infecting the victim client provides a password protected backdoor to the attacker.	T1587-001 and T1588-001	101
Python Backdoor	A malicious program written using Python programming language infects the victim client to serve a backdoor to the attacker. The attacker can obtain a simple reverse shell using tools like netcat or socat	T1587-001 and T1588-001	179
Binom ASM	A computer virus that is written using assembly language searches for ELF files in the victim client and injects malicious payload. This computer virus requires admin level access for successful infection. It target the files located in the bin directory in the victim's machine	T1027-001	292
Eternity ASM	A computer virus that is written using assembly language that infects a target ELF file in the victim's machine.	T1027-001	152
Dataseg Code Injector	A malware that injects unwanted/malicious code into the data segment of the binaries mainly for defense evasion purposes.	T1027-001	202
Bash Spyware	A simple bash script that uses built-in tools for harvesting internal system data and send it to a compromised mail server	T1119	46

Generated Datasets- Adversarial Environments

Attack Case	Unstructured System Logs	Structured System Logs	Network Data	System Metrics	Audit Data
GonnaCry	2,326	542	1,776	639	5,695
ech0raix	363	323	1,002	385	1,792
ACK Flood	2,925	589	785,920	1,059	2,207
NTP DDoS Amplification	1,555	0	139,913	463	76,451
SYN Flood	3,913	869	1,119,228	868	2,371
malaria	15,858	10,686	2,385	861	6,413
nemox	3,811	907	2,119	450	8,708
nf3ct0r	4,251	1,134	2,776	845	8,489
utrojan	4,499	1,422	3,538	1,410	5,029
Lin Blackhole	479	415	1,786	560	1,968
Lin Ovason	2,676	624	2,162	744	4,614
Python Backdoor	2,930	840	3,520	1,349	2,278
Binom ASM	1,646	1,383	7,117	3,432	16,943
Eternity ASM	2,715	599	2,310	674	6,474
Dataseg Code Injector	2,897	786	4,225	1,642	18,604
Bash Spyware	410	357	980	263	2,110

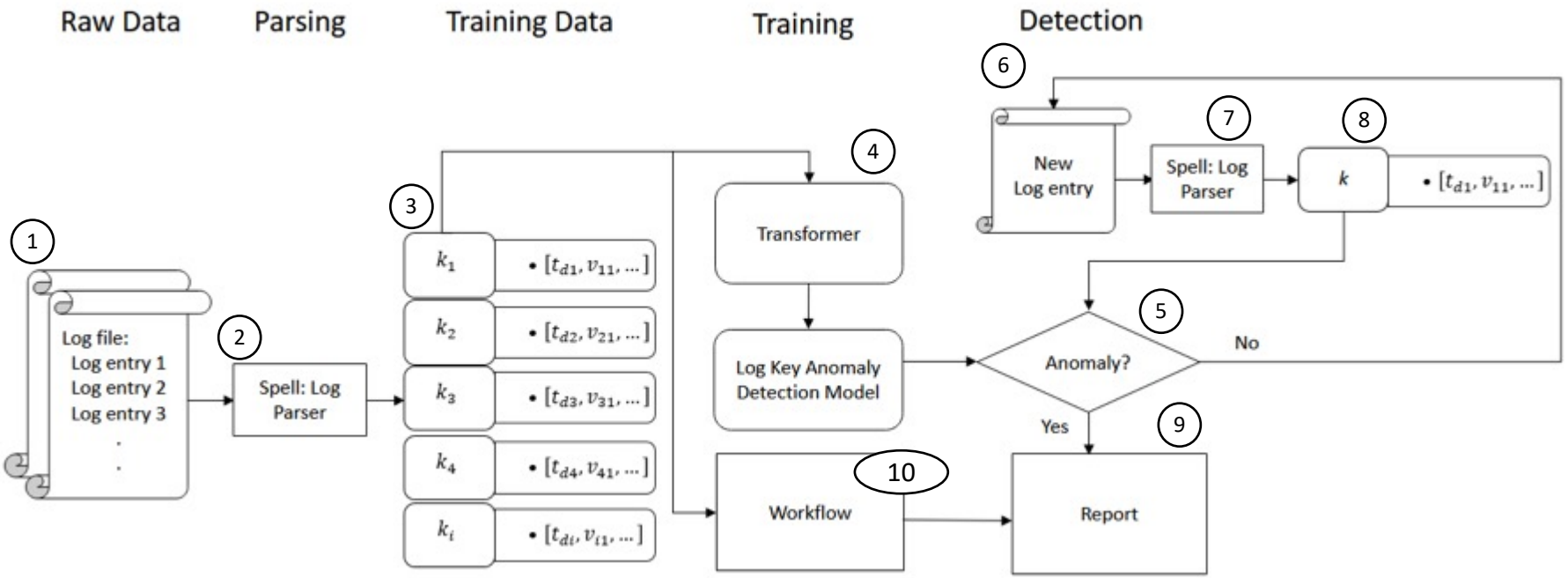
Table 4.16: Malicious software samples of cyber threat detection dataset

Log Type	Cyber Kill Chain Experiments					
	CKC 01	CKC 02	CKC 03	CKC 04	CKC 06	CKC 06
connection	2003	66500	139355	133718	67510	332849
dhcp	3	2	2	1	2	2
dns	109	315	9	94	131	335
dpd	0	0	7	0	0	0
files	61	31635	16555	507	28462	106738
ftp	0	0	0	0	0	1
http	68986	31544	8780	411	28350	106757
kerberos	0	0	0	2	0	0
snmp	0	1	0	0	0	0
ssh	1	2	9	0	9	0
ssl	14	2	0	51	20	0
weird	18	30	45	23	81	54
x509	0	2	0	100	40	0

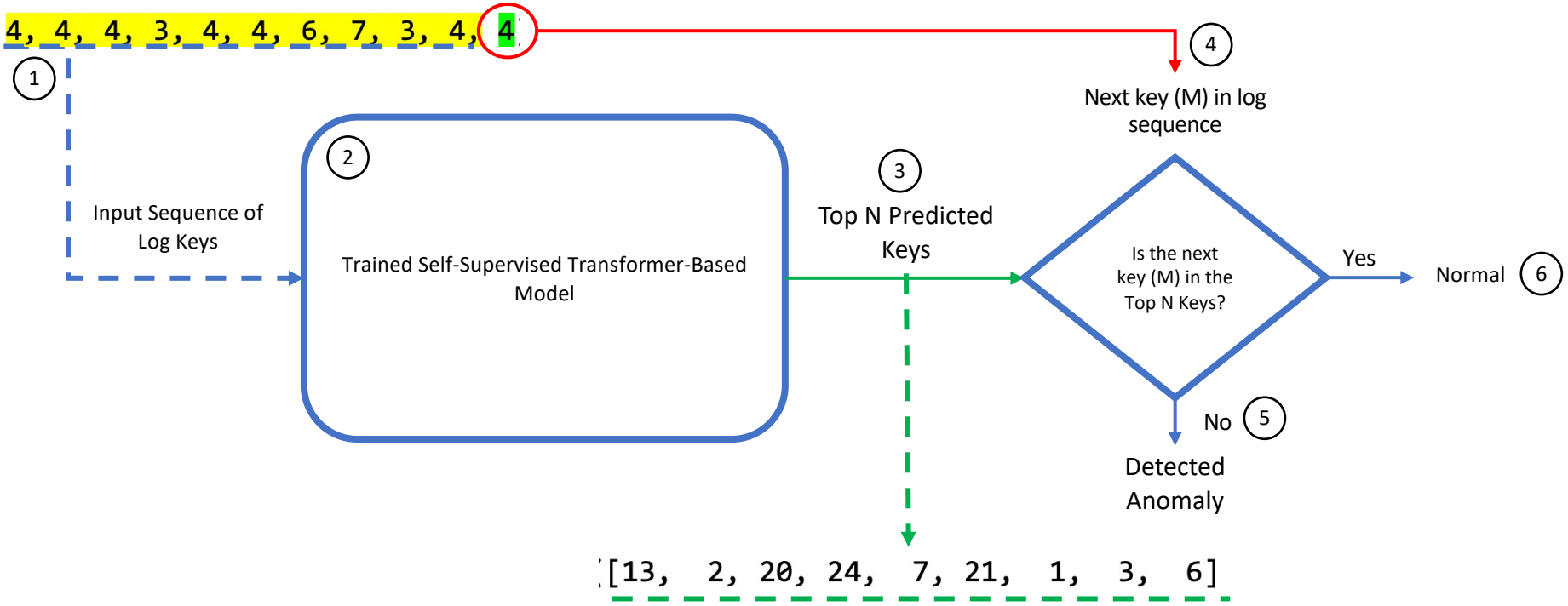
Table 4.17: Structured Zeek logs by log type collected by Filebeat

Unrestricted Content
This document does not contain technology or technical data controlled under either the U.S. International Traffic in Arms Regulations or the U.S. Export Adm

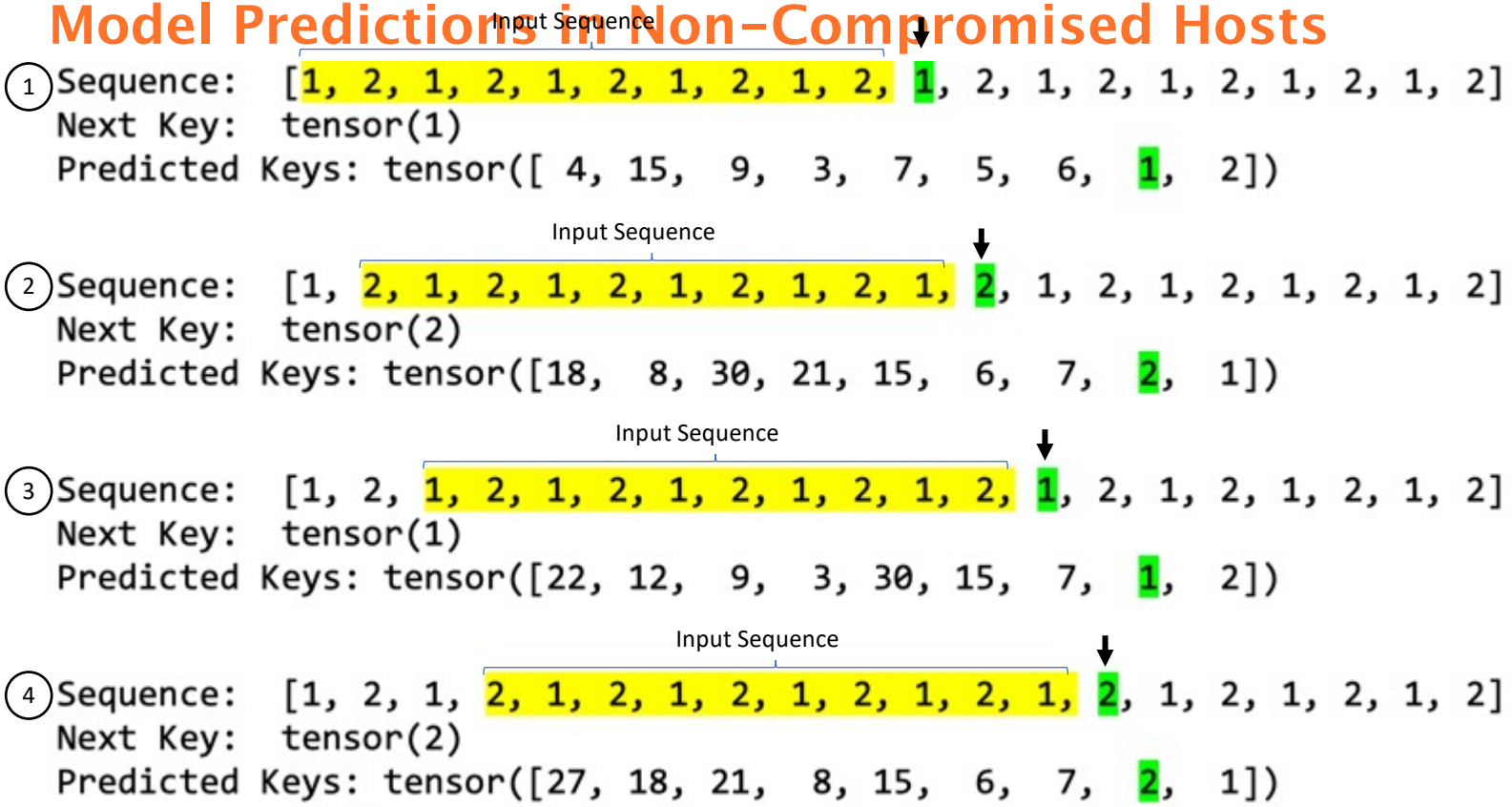
Threat and Anomaly Detection Model Architecture



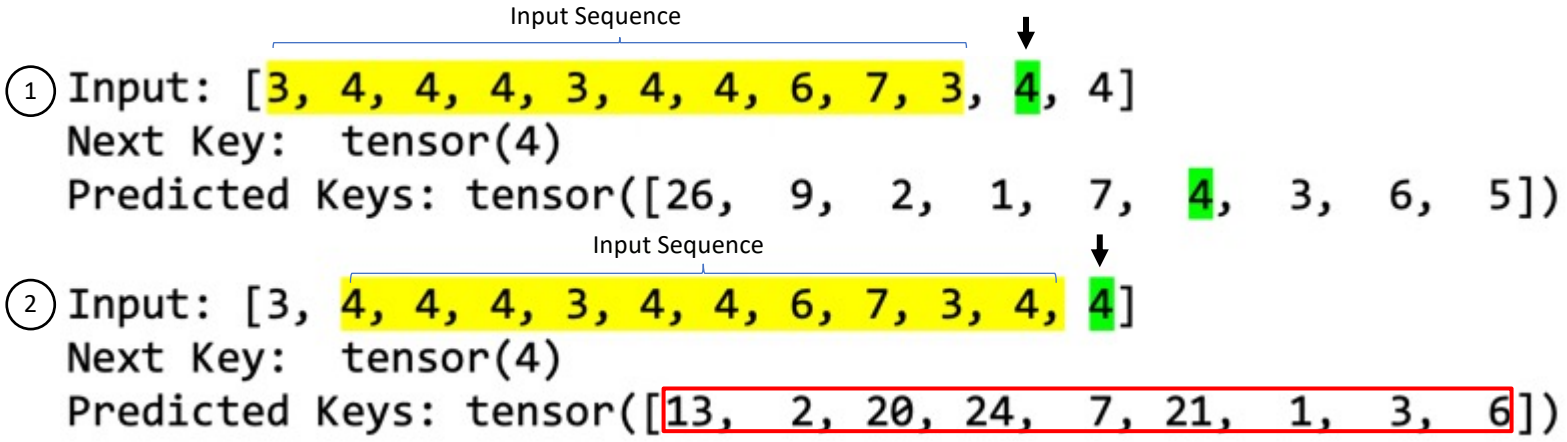
Inference


4, 4, 4, 3, 4, 4, 6, 7, 3, 4, 4
1
Input Sequence of Log Keys
2
Trained Self-Supervised Transformer-Based Model
3
Top N Predicted Keys
[13, 2, 20, 24, 7, 21, 1, 3, 6]
4
Next key (M) in log sequence
Is the next key (M) in the Top N Keys?
Yes
Normal
6
No
5
Detected Anomaly

Model Predictions in Non-Compromised Hosts



Anomaly Detection Based on Model Predictions

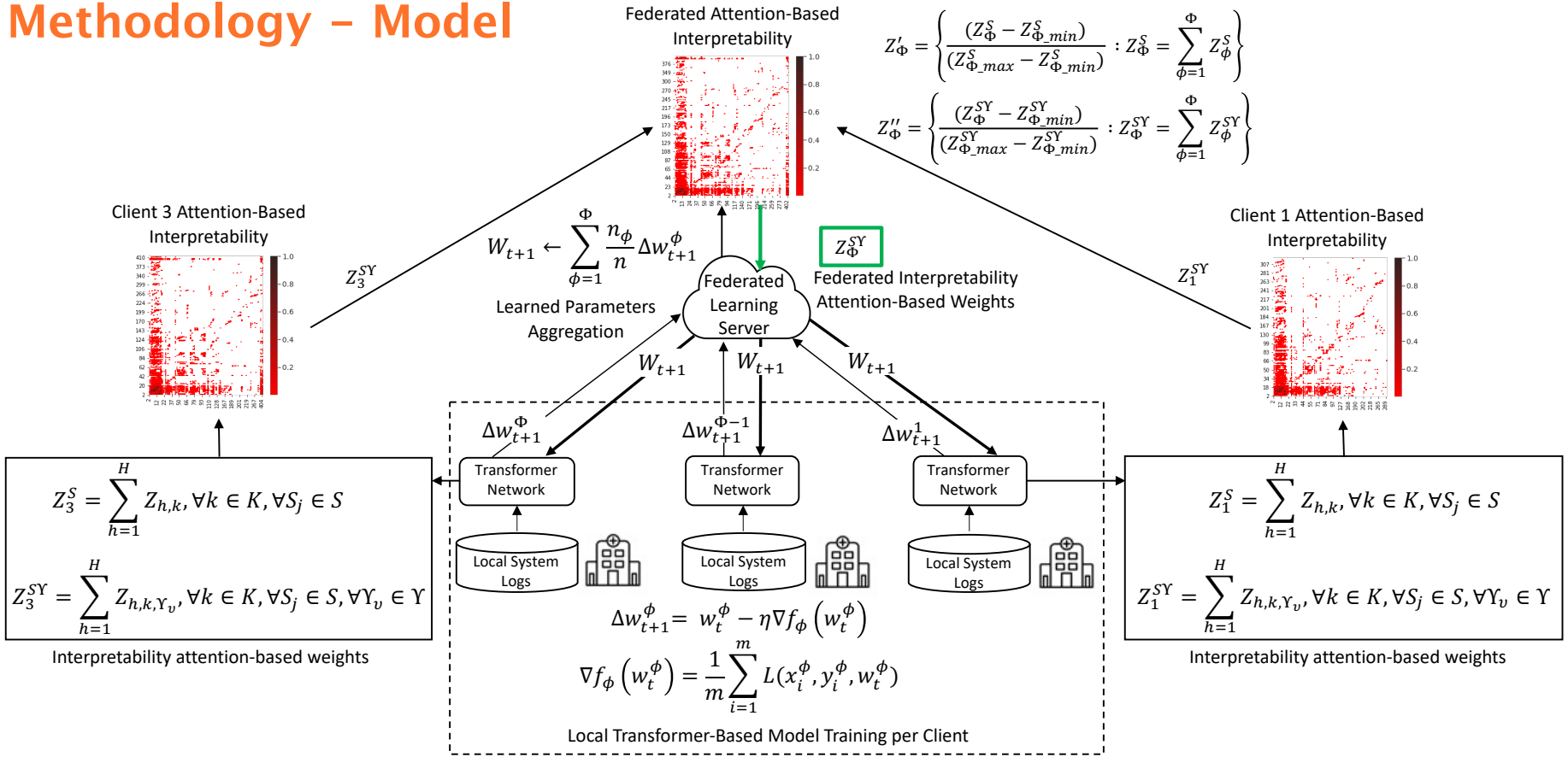


For the second input sequence that is fed into our model, we can observe that the next log key (represented by the number 4) is not within the list of predicted keys. The predicted keys represent the top 10 keys that have the highest probability to follow the given input sequence.

Experimental (results)

Φ	N	H	HDFS				CTDD			
			Accuracy	Precision	Recall	F-score	Accuracy	Precision	Recall	F-score
1	1	1	83.986	92.524	70.240	79.857	**79.394**	**77.333**	**77.333**	**77.333**
2			90.508	84.219	97.211	90.250	73.939	74.242	65.333	69.504
4			89.642	83.063	96.823	89.417	78.788	77.027	76.000	76.510
8			83.054	87.229	73.223	79.615	80.606	77.922	80.000	78.947
10			85.487	76.612	97.720	85.888	80.606	76.543	82.667	79.487
1		2	**94.432**	**93.755**	**93.936**	**93.845**	79.394	78.873	74.667	76.712
2			83.986	92.524	70.240	79.857	78.788	77.778	74.667	76.190
4			89.685	83.938	95.440	89.320	79.394	77.333	77.333	77.333
8			91.242	89.143	91.802	90.453	81.212	77.500	82.667	80.000
10			92.634	88.285	96.507	92.213	80.606	77.215	81.333	79.221
1	4	1	89.883	85.492	93.476	89.306	78.788	72.000	65.333	75.524
2			93.051	90.788	94.179	92.452	78.182	76.712	74.667	75.676
4			88.096	80.063	98.084	88.162	78.788	77.027	76.000	76.510
8			93.072	88.711	97.017	92.678	80.606	78.667	78.667	78.667
10			**93.072**	**88.677**	**97.065**	**92.682**	78.788	77.027	76.000	76.510
1		2	92.820	89.249	95.634	92.331	78.182	78.261	72.000	75.000
2			91.209	85.504	96.992	90.886	74.545	76.190	64.000	69.565
4			89.762	82.534	98.108	89.650	79.394	78.873	74.667	76.712
8			92.776	88.285	96.871	92.379	80.000	79.167	76.000	77.551
10			92.842	88.300	97.017	92.453	**81.818**	**79.221**	**81.333**	**80.263**
1	6	1	92.152	92.344	90.104	91.210	78.788	79.412	72.000	75.52
2			89.291	84.480	93.476	88.751	78.182	78.261	72.000	75.000
4			88.600	84.339	91.826	87.924	78.182	76.712	74.667	75.676
8			90.606	86.017	94.591	90.100	79.394	76.623	78.667	77.632
10			91.560	86.296	96.677	91.192	81.212	78.205	81.333	79.739
1		2	92.568	91.697	91.875	91.786	78.788	78.571	73.333	75.862
2			88.677	81.531	96.895	88.551	77.576	77.143	72.000	74.483
4			82.670	89.570	69.779	78.446	77.576	77.941	70.667	74.126
8			85.630	79.813	91.293	85.168	80.606	78.667	78.667	78.667
10			79.830	76.660	90.420	82.974	**81.818**	**79.221**	**81.333**	**80.263**

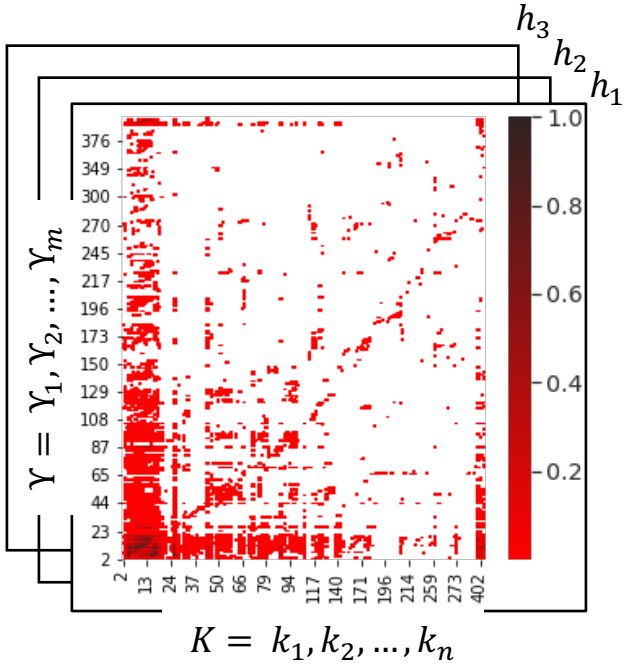
Methodology – Model



Methodology – Interpretability

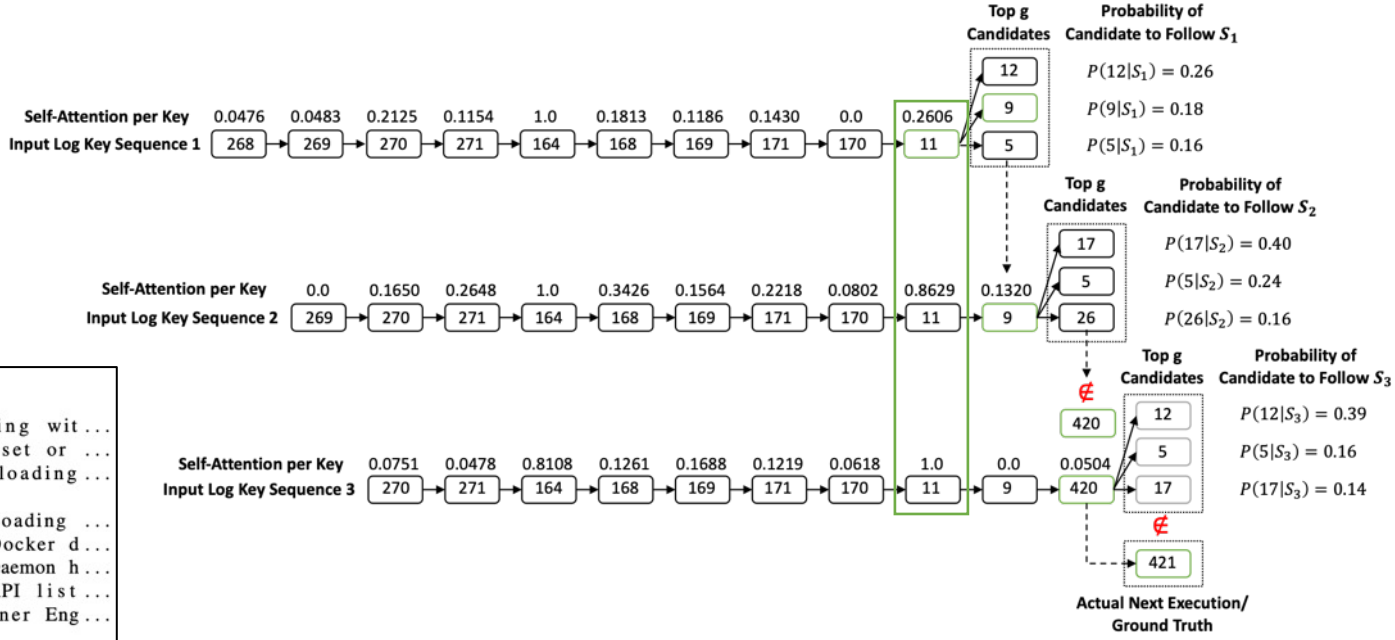
- The interpretability module makes use of the attention calculated based on the input data.
- Using attention we compute the attention-based weights. (1)

(1) $$Z_{\phi}^{S\Upsilon} = \sum_{h=1}^{H} Z_{h,\kappa,\Upsilon_v}, \forall \kappa \in K, \forall S_j \in S, \forall \Upsilon_v \in \Upsilon$$



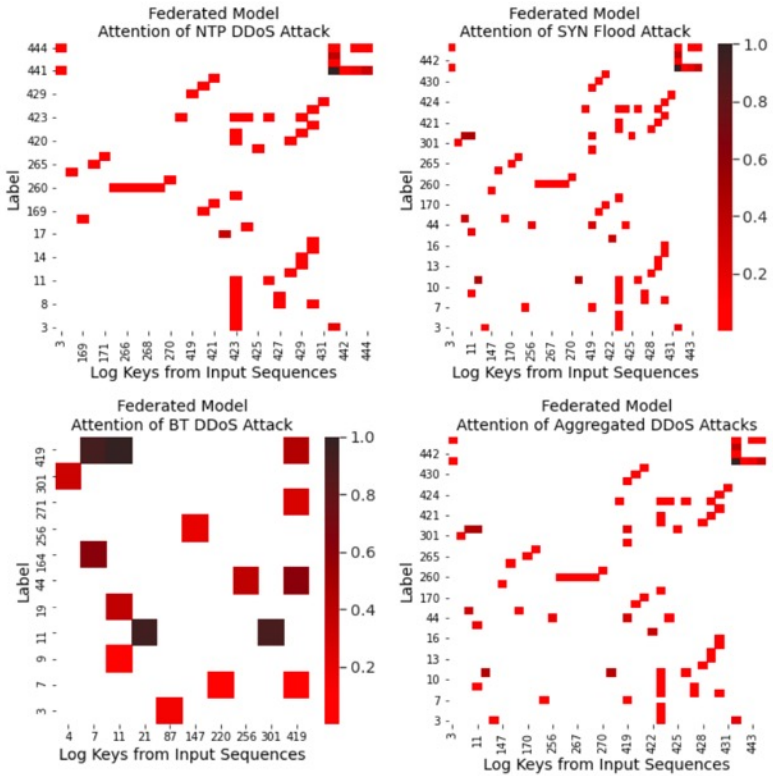
Interpretability – Example of the Model's Inference

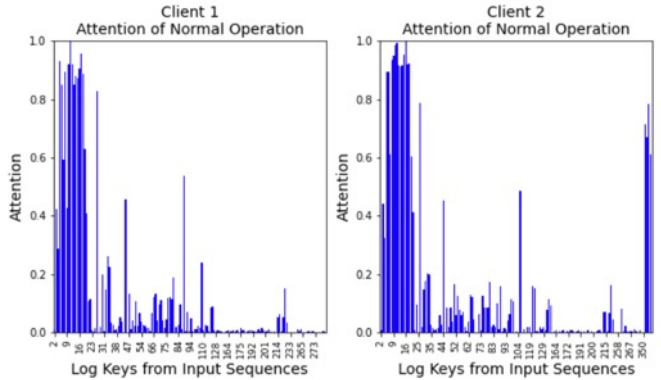
- Evaluating a sequence of log keys produced from a subset of a log file.



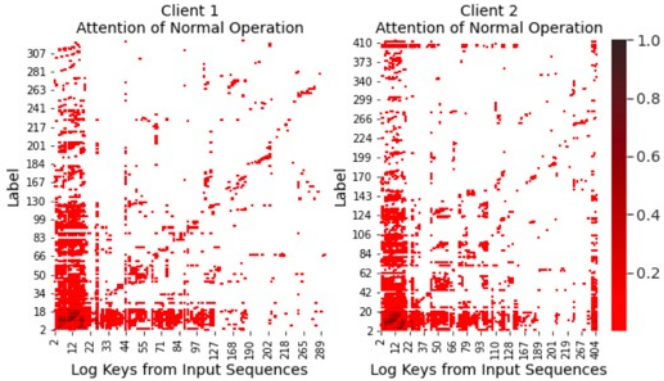
Log Key	Log Template
268	<*> <*> ctnetlink v0.<*> registering wit...
269	link_config autonegotiation is unset or ...
270	WARNING Unknown index <*> seen reloading...
271	<*> Link UP
164	time <*> <*>"" level info msg ""Loading ...
168	time <*> <*>"" level info msg ""Docker d...
169	time <*> <*>"" level info msg ""Daemon h...
171	time <*> <*>"" level info msg ""API list...
170	Started Docker Application Container Eng...
11	ens3 Configured
9	Started ntp-systemd-netif.service.
420	Stopping Network Time Service...
421	ntpd exiting on signal <*> Terminated

Interpretability



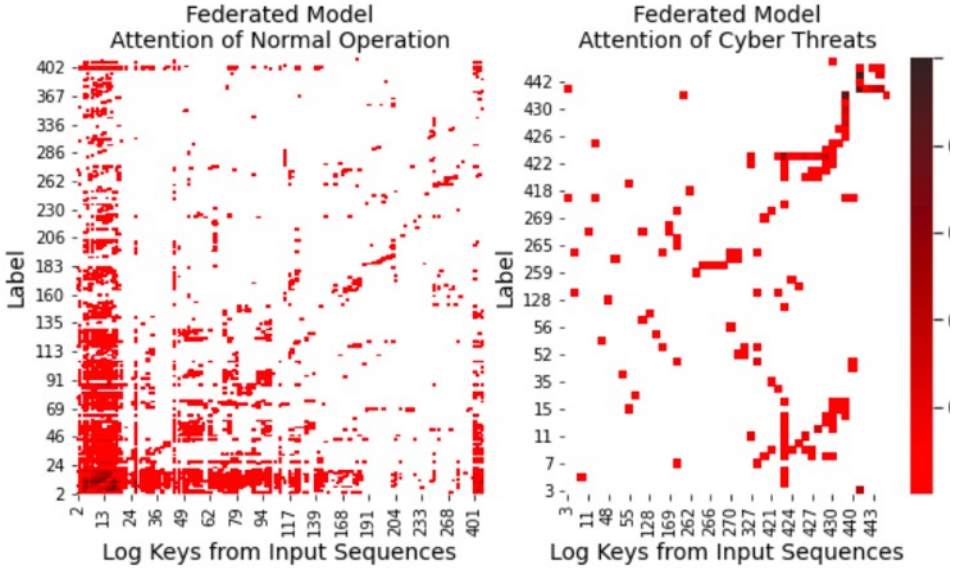


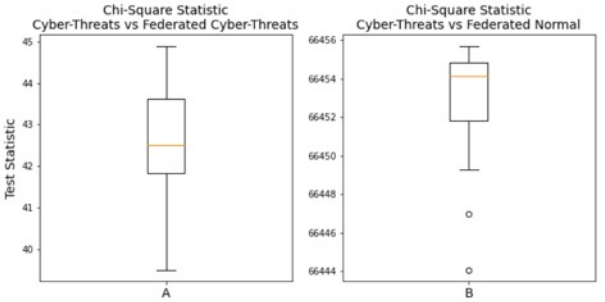
(a) Interpretability attention-based weights Z_ϕ^S for Client 1 (Left) and Client 2 (Right). The attention for each layer in the encoder is aggregated by unique key $\kappa \in K$ for all $S_j \in S$.



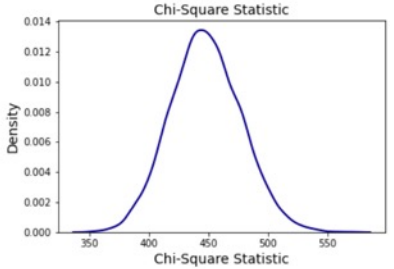
(b) Interpretability attention-based weights $Z_\phi^{S\Upsilon}$ for Client 1 (Left) and Client 2 (Right). The attention of each layer in the encoder is aggregated by each $\upsilon \in$ (y-axis) and each unique key $\kappa \in K$ (x-axis) for all $S_j \in S$.

Interpretability – Goodness of Fit Test





(a) Box plot of Chi-Square statistic computed for Z_{ϕ}^{ST} of each cyberattack in the CTDD dataset with respect to the federatedZ_{Φ}^{ST} of all cyber threats (left) and Z_{Φ}^{ST} of all uncompromsed systems.



(b) Chi-Square distribution curve with 448 degrees of freedom.

Figure 6.7: Goodness of fit test for the multinomial interpretability attention-based weights of each attack with respect to the federated interpretability attention-based weights of compromised and uncompromised systems.

Methodology – Interpretability

- The interpretability module makes use of the attention calculated based on the input data.
- Using attention we compute the attention-based weights. (1)
- Karl Pearson's chi-square statistic is computed to conduct a goodness of fit test and identify if a particular sample follows a specific distribution. (2)
- The interpretability module uses the p-value (4) to fail to reject the hypothesis H0 or reject the hypothesis H1 (3) that the distribution of the interpretability attention-based weights for the given sample $Z\phi$ shares the same distribution as $Z\Phi$.

$$(1)\, Z_{\phi}^{S\Upsilon} = \sum_{h=1}^{H} Z_{h,\kappa,\Upsilon_v}, \forall \kappa \in K, \forall S_j \in S, \forall \Upsilon_v \in \Upsilon$$

$$(2)\, D_i = \sum_{i=1}^{v} \sum_{k=1}^{K} \frac{\left(Z_{\phi}^{S\Upsilon_i} - Z_{\Phi}^{S\Upsilon_i}\right)^2}{Z_{\Phi}^{S\Upsilon_i}}, \forall v \in \Upsilon, \forall \kappa \in K$$

$$(3)\, H_0 : Z_{\phi}^{S\Upsilon} = Z_{\phi}^{S\Upsilon_1}, \ldots, Z_{\phi}^{S\Upsilon_v} = Z_{\Phi}^{S\Upsilon} \quad VS$$

$$H_1 : Z_{\phi}^{S\Upsilon} \neq Z_{\Phi}^{S\Upsilon}$$

$$(4)\, p - value = P\left(\chi^2(v-1) > \sum_{i=1}^{v} \sum_{k=1}^{K} \frac{\left(Z_{\phi}^{S\Upsilon_i} - Z_{\Phi}^{S\Upsilon_i}\right)^2}{Z_{\Phi}^{S\Upsilon_i}}\right)$$

Thank You

Unrestricted Content
This document does not contain technology or technical data controlled under either the U.S. International Traffic in Arms Regulations or the U.S. Export Adm

References

[1] Kruegel, Christopher, and Thomas Toth. "Using decision trees to improve signature-based intrusion detection." International Workshop on Recent Advances in Intrusion Detection. Springer, Berlin, Heidelberg, 2003.

[2] Anjum, Farooq, Dhanant Subhadrabandhu, and Saswati Sarkar. "Signature based intrusion detection for wireless ad-hoc networks: A comparative study of various routing protocols." *2003 IEEE 58th Vehicular Technology Conference. VTC 2003-Fall (IEEE Cat. No. 03CH37484)*. Vol. 3. IEEE, 2003.

[3] Kumar, Vinod, and Om Prakash Sangwan. "Signature based intrusion detection system using SNORT." *International Journal of Computer Applications & Information Technology* 1.3 (2012): 35-41.

[4] Holm, Hannes. "Signature based intrusion detection for zero-day attacks:(not) a closed chapter?." 2014 47th Hawaii international conference on system sciences. IEEE, 2014.

[5] Zhaojun, G., Chao, W., 2010. Statistic and analysis for host-based syslog, in:2010 Second International Workshop on Education Technology and Com-puter Science, IEEE. pp. 277-280.

[6] Wang, Z., Zhu, Y., 2017. A centralized hids framework for private cloud,in: 2017 18th IEEE/ACIS International Conference on Software Engineer-ing, Artificial Intelligence, Networking and Parallel/Distributed Computing(SNPD), IEEE. pp. 115-120.

[7] Deshpande, P., Sharma, S.C., Peddoju, S.K., Junaid, S., 2018. Hids: A hostbased intrusion detection system for cloud computing environment. Interna-tional Journal of System Assurance Engineering and Management 9, 567-576.

[8] Moustafa, N., Slay, J., 2017. A hybrid feature selection for network intrusiondetection systems: Central points. arXiv preprint arXiv:1707.05505 .

[9] Ashfaq, R.A.R., Wang, X.Z., Huang, J.Z., Abbas, H., He, Y.L., 2017. Fuzzinessbased semi-supervised learning approach for intrusion detection system. In-formation Sciences 378, 484-497.

[10] Yin, C., Zhu, Y., Fei, J., He, X., 2017. A deep learning approach for intrusiondetection using recurrent neural networks. IEEE Access 5, 21954-21961.