Reproducing ATT&CK Techniques and Lifecycles to Train Machine Learning Classifier

Fietyata Yudha, Ying-Dar Lin, Yuan-Cheng Lai, Didik Sudyana, Ren-Hung Hwang

***Abstract*—The MITRE ATT&CK framework categorizes threat actor behaviors into a sequence of techniques called the attack lifecycle. Based on this, Our work introduces a dual-labeled dataset that is accurately labeled with distinct techniques and lifecycles defined by ATT&CK. The dataset offered a more thorough perspective than previous datasets that employed binary or attack classification. It encompassed 17 distinct techniques throughout five lifecycles and was generated through an automated method that guarantees reproducibility and learnability. Reproducibility guarantees the dataset's consistency, whereas learnability signifies its use in training machine learning models. Our analysis produced a positive result. The dataset achieved a Pearson correlation coefficient of 0.7 for reproducibility. Regarding distinguishing classes, it exhibits an average AUC-ROC score of 0.92 for techniques and 0.82 for lifecycles. The model training yielded an average F1 score of 0.95 for technique classification and 0.9 for lifecycle classification, but only for the traffic dataset.**

***Index Terms*—ATT&CK, technique, lifecycle, Intrusion Detection System.**

I. Introduction

MITRE ATT&CK, or ATT&CK, is a fundamental framework in cybersecurity that aids in developing threat models. This framework encompasses an extensive knowledge base derived from real-world observations of adversary tactics and techniques. It includes various cyberattack methodologies, tools, perpetrators, and mitigation strategies [1]. The framework is structured as a matrix, outlining several tactics, techniques, and sub-techniques.

The MITRE ATT&CK v16 framework defines 14 tactics within the enterprise matrix, closely corresponding to traditional hacking stages such as reconnaissance, gaining access, maintaining access, and clearing track. These tactics are sequentially numbered from one to fourteen, reflecting the conventional hacking steps. Tactics serve to categorize various techniques within the ATT&CK framework. The sequences of techniques form an attack's lifecycle [2], effectively illustrating a threat actor's behavior and methodology during the attack.

Employing the ATT&CK framework for intrusion detection provides a significant chance to gather detailed information regarding attacks, encompassing tactics, techniques, and sequences. It offers enhanced insights into the attack's behavior. These insights are crucial for recognizing potential attack variants of existing attacks and for formulating entirely new attacks through the analysis of certain lifecycle segments.

Machine learning has proven effective in detecting privacy risks and identifying security threats, showcasing its potential for application in ATT&CK-based detection. Numerous studies have investigated the ATT&CK framework, mostly concentrating on the transformation of text-based materials into ATT&CK tactics or techniques through Natural Language Processing (NLP), including reports and online resources [3]–[5]. Several studies have concentrated on the technical aspects of attacks while ignoring the entities defined in the ATT&CK framework. For instance, [6] included constructed anomalies into captured packets, while [7]–[10] executed attacks through attack replay. Automated attack generation methods have been employed to emulate attack behaviors and develop datasets, as demonstrated by [11], [12], to enhance manual attack generation. Our prior work introduced CREMEv1, a framework designed for replicating datasets [13] but without taking into account the ATT&CK techniques and lifecycles. Recently, the University of West Florida (UWF) [14], [15] aimed to construct datasets that incorporate ATT&CK entities, concentrating exclusively on tactics without addressing techniques or lifecycles.

Utilizing the ATT&CK framework for dataset labeling offers a systematic and uniform method for classifying attack behaviors, thereby improving both the accuracy of labeling and the clarity of interpretation. The framework's comprehensive technique mappings facilitate the capture of detailed attacker behaviors at each stage, thereby enabling the model to learn patterns specific to the lifecycle. This enhanced training process not only improved detection accuracy for established techniques but also increased the model's adaptability to new or variant attacks by highlighting behavior-based patterns.

Utilizing the ATT&CK framework, we develop a dual-labeled dataset that classifies attacks according to their techniques and lifecycles. This dataset has two columns of multi-class labels: one related to techniques outlined in the ATT&CK framework and the other related to attack lifecycles. Our dual-label system significantly enhances the investigation of the relationship between attack techniques and their lifecycles. This innovation addresses a substantial problem in existing public datasets, which generally employ single-label classifications. By modifying CREMEv1, we aim to ensure the reproducibility. In our context, "reproducibility" indicates the ability of others to replicate the dataset utilizing our approach. On the other hand, the machine learning model's ability to

Fietyata Yudha is with EECS International Graduate Program, National Yang Ming Chiao Tung University, Hsinchu City 300, Taiwan

Ying-Dar Lin is with Department of Computer Science, National Yang Ming Chiao Tung University, Hsinchu City 300, Taiwan

Yuan-Cheng Lain is with the Department of Information Management, National Taiwan University of Science and Technology, Tapei 106, Taiwan

Didik Sudyana is with the Computer and Network Center, National Cheng Kung University, Tainan 701, Taiwan

Ren-Hung Hwang is with the College of Artificial Intelligence, National Yang Ming Chiao Tung University, Tainan 711, Taiwan

Fietyata Yudha is the corresponding author

TABLE I
SUMMARY OF WORK IN ATTACK REPRODUCTION, DATASET GENERATION, AND ATT&CK IMPLEMENTATION

Author	Auto-Reproduction	Self-developed Tool	Attack Behavior[a]	Data Source			ATT&CK Label	
				Traffic	Log	Accounting	Technique	Lifecycle
[14]	×	×	-	×	○	×	×	×
[7]			Pr, Re, Bn	○	×			
[9]			Pr, Re, Eg, Bn					
[10]			Re, Eg, Bn					
[15]			-		○			
[8]		○	-		×			
[6]			Pr, Re					
[11]	○		Re					
[12]			Re, Eg, Bn					
[13]			Re, Em, Bn		○	○		
Ours			Pr, Re, Em, Bn				○	○

[a] Different types of behaviors that were executed during the attack reproduction process; Pr: Probing, Re: Resource exhaustion, Eg: Exploitation general, Em: Exploitation metasploit, Bn: Botnet

understand the dataset must be evaluated to ensure the datasets are suitable for learning. Additionally, we implement specific methods to evaluate reproducibility and the ability of models to identify techniques and lifecycles.

This work includes three key significant contributions. First, we align the ATT&CK framework to replicate attacks and generate datasets. These datasets are precisely labeled with two types of multi-class labels: the specific techniques employed in the attacks and the attack lifecycle. Furthermore, we deliver a novel approach for assessing reproducibility that differs from the conventional metric, which concentrates on the quantity of data for every task. We employ Principal Component Analysis (PCA) and the Pearson Correlation Coefficient (PCC) to analyze similarities among datasets from various replays. This technique offers a more comprehensive and accurate assessment of reproducibility. Last, we assess the dataset's learnability by evaluating the classification performance of several machine-learning models. These models are tasked with classifying techniques and lifecycles. This evaluation employs fundamental metrics such as an Area Under the Curve-Receiver Operating Characteristic (AUC-ROC) and F1-score.

II. RELATED WORKS

Table I compares studies on attack reproduction, dataset creation, and the application of the ATT&CK framework in IDS datasets. These works are classified by several criteria, including method of reproduction, the tools developed, types of attack behavior, data sources, and ATT&CK labels (techniques and lifecycle stages).

Regarding reproduction method, most studies use manual method to generate datasets, such as work by [6]–[10], [14], [15]. In contrast, a limited subset of studies [11]–[13] developed specialized tools for automation. In terms of datasets, attack behaviors are divided into five categories: probing, resource exhaustion, generic exploitation, Metasploit exploitation, and botnets. Notably, resource exhaustion was the most commonly observed behavior across multiple works. The majority of data sources are traffic data, with the exception of [14], which used a Zeek connection log to generate a traffic log file. The use of ATT&CK labels is a key breakthrough that sets our work apart from earlier works. The application of ATT&CK labels represents a significant advancement that distinguishes our work from prior studies. The notable exceptions are the works of [14] and [15], which included terminology for techniques such as discovery and reconnaissance.

III. DESIGN ISSUES

A. *Attack reproduction*

Publicly accessible attack datasets replicate attack data without considering ATT&CK techniques and their lifecycles. This is due to the reason that these datasets typically demonstrate a single phase of the attack, do not consider the attack sequence, and do not employ specific techniques or lifecycles from the ATT&CK framework. For example, the CSE-CIC-IDS2018 dataset [1] runs the attack individually without considering the attack sequence.

To acquire the raw data containing the comprehensive information from ATT&CK, the dataset must be replicated via the ATT&CK technique and lifecycle. Additionally, the common data that is recorded by the public dataset is traffic, which only

[1] https://www.unb.ca/cic/datasets/ids-2018.html

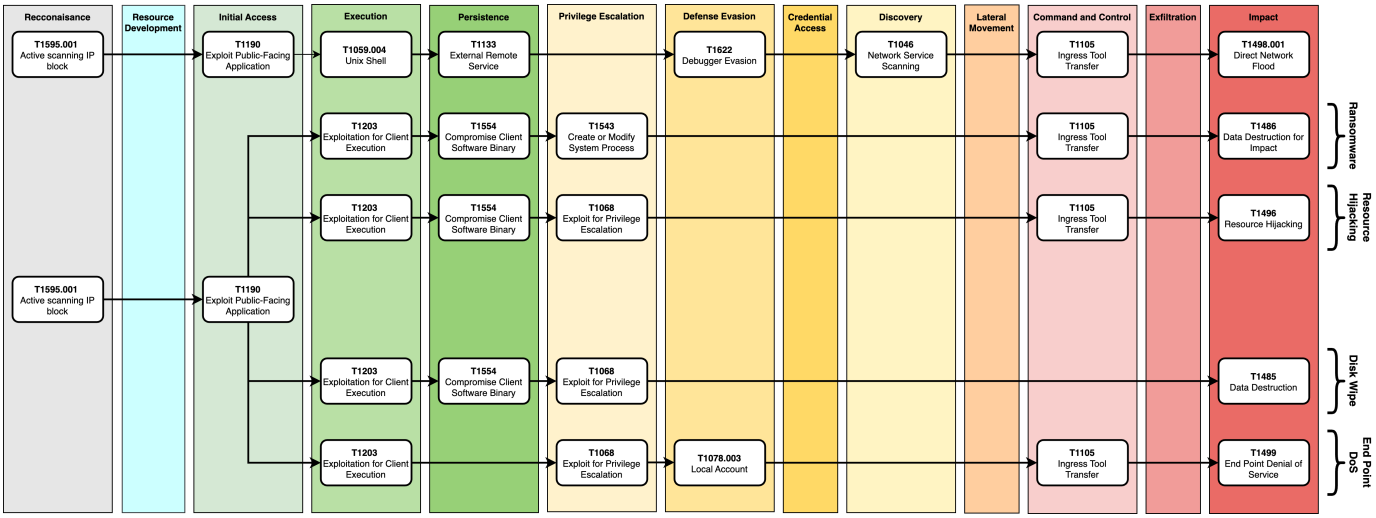


Fig. 1. Selected Attack Lifecycle

shows a single point of view. The needs of multiple points of view are also important in terms of dataset generation.

B. *Data labeling*

Numerous publicly accessible datasets currently employ the single or multi-class label. However, replication of these datasets does not account for labeling utilizing ATT&CK techniques and lifecycles. To annotate a dataset with relevant ATT&CK information, it is essential to accurately recreate the attacks, align the appropriate techniques and lifecycles, examine the raw data, and label it appropriately.

C. *Dateset evaluation*

Reproducing the dataset with the ATT&CK label fulfills two objectives: enhancing the dataset's quality and strengthening the machine learning ability to learn and identify ATT&CK-Related information, including techniques and lifecycles. A reproducibility metric must be established to guarantee that the generated dataset maintains consistent data over multiple attempts. Additionally, a method of evaluation is necessary to assess the learnability of a dataset utilizing machine learning techniques.

IV. Solution Idea

A. *ATT&CK-based attack reproduction*

In order to replicate the attacks, we employ a variety of attack variants, including botnets, disk wipes, ransomware, resource hijacking, and endpoint denial of service. Each attack variant is linked with techniques utilizing ATT&CK Navigator[2] and through a manual comparison of the ATT&CK technique descriptions or contexts with the attack behavior or functionality of the attack tool. Figure 1 illustrates the lifecycle for each attack variant and is derived from the ATT&CK enterprise matrix. Our analysis identified 17 unique techniques, comprising 6 common techniques utilized over many lifecycles. The common techniques have been identified as T1595.001, T1190, T1203, T1554, T1068, and T1105.

Figure 2a depicts the process for replicating an attack variant. Each stage of the attack lifecycle is replicated using preset settings, as shown in Figure 1. In the context of replaying attack, we define several notations:

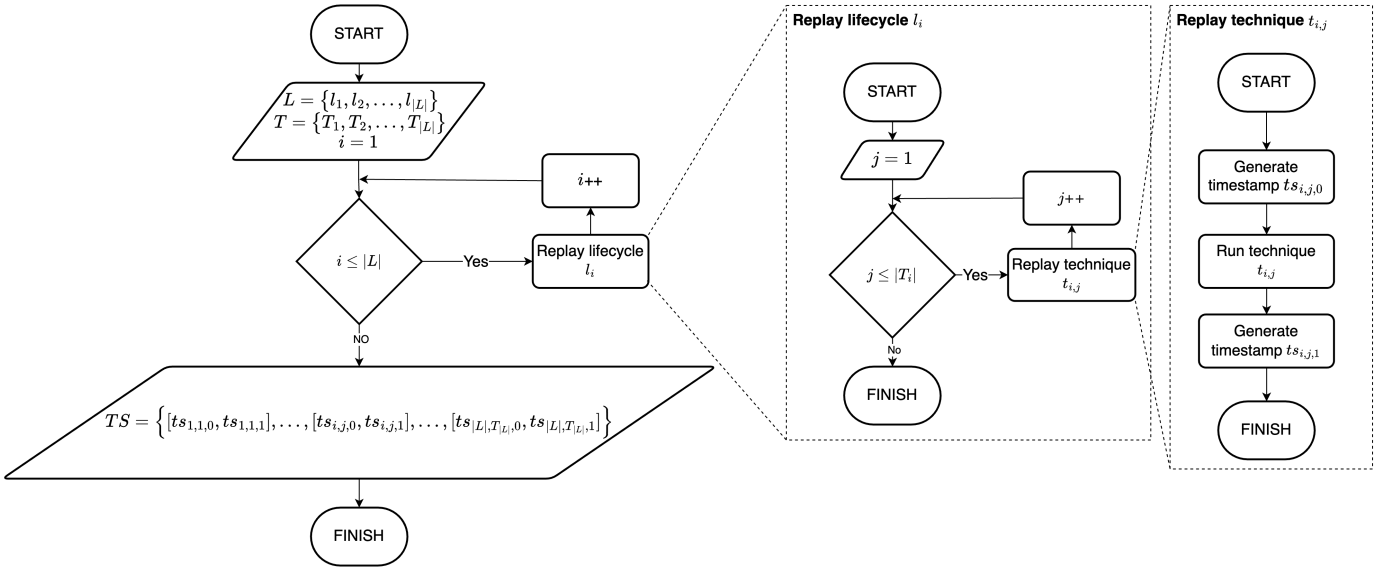
- L: the set of lifecycles l_i, which denotes the i-th lifecycle.
- T_i: the set of techniques $t_{i,j}$, which denotes the j-th technique in the i-th lifecycle.
- $|X|$: the number of elements in set X.
- TS: the set of a list of timestamps that is represented as $[ts_{i,j,0}, ts_{i,j,1}]$, where $ts_{i,j,0}$ and $ts_{i,j,1}$ denote the start and end timestamps of the j-th technique in the i-th lifecycle, respectively.

To start the reproduction process, the number of lifecycles $|L|$ is counted, and a loop is established. The number of techniques $|T_i|$ is calculated for each lifecycle l_i. Subsequently, every technique $t_{i,j}$ is executed, yielding a collection of timestamps TS accompanied by lists of timestamps. During the replay, each technique produced a pair of timestamps ($ts_{i,j,0}$ and $ts_{i,j,1}$) indicating the start and end times of a particular technique, respectively. The timestamp for a lifecycle is aggregated into a set, TS, which constitutes a collection of timestamp lists.

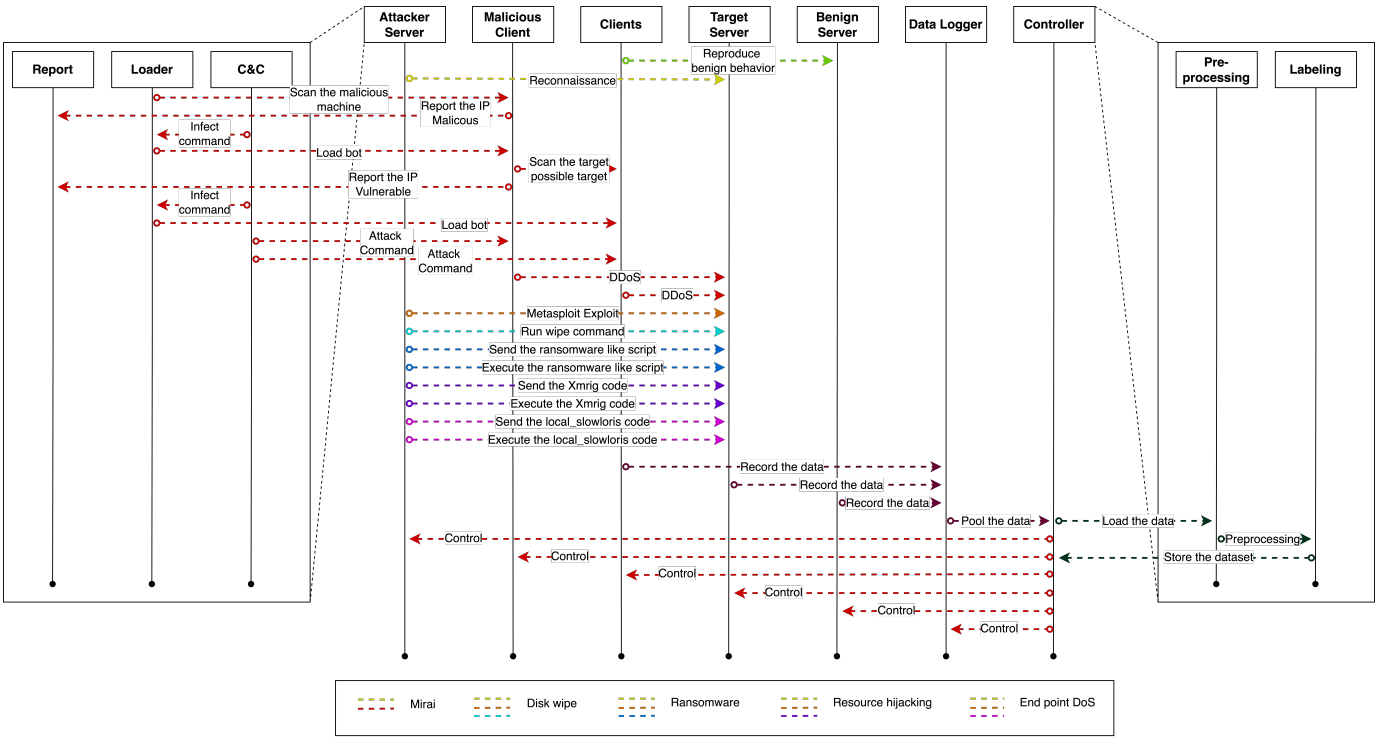
This process produces the raw information from 3 data sources, namely accounting, syslog, and traffic. The raw information from the reproduction process is preprocessed to extract the features before performing the labeling. For accounting data, the raw information is transformed into a dataset using a straightforward approach that retains the original features from the raw data. Syslog data is processed through the event template via the Drain algorithm, which is a log parsing method that efficiently groups raw log data into structured templates to create a dataset. These approaches differ significantly from the traffic dataset, where a dedicated feature extraction framework is employed.

[2]https://mitre-attack.github.io/attack-navigator/

This article has been accepted for publication in IEEE Network. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/MNET.2025.3551333



(a) Attack Reproduction Process Flowchart



(b) Attack Reproduction Process for Each Lifecycle

Fig. 2. Attack Reproduction

B. *Dual-labeled dataset labeling*

The application of dual-labeling, involving both MITRE ATT&CK techniques and lifecycle within the dataset, introduces notable challenges related to complexity and label overlap. Complexity arises when a single attack step, which is a phase involving specific actions by the attacker in an attack lifecycle, corresponds to multiple techniques in the same or even different tactics. To address this, we integrate the attack step's position within the attack lifecycle and its alignment with ATT&CK tactics. For instance, if a step corresponds to the techniques "Active Scanning" and "Network Service Discovery" but occurs in the front part of the attack sequence, we prioritize the technique associated with the earlier tactic—specifically the "Active Scanning" technique.

Furthermore, label overlap between lifecycles emerges when identical or similar techniques are applicable across multiple lifecycles. To mitigate these overlaps, we implement a structured labeling methodology. For technique labeling, two primary constraints are employed: host-specific data (such as IP addresses or hostnames), which refer to the identified information that distinguishes specific devices within the network, and breakpoint data generated during attack replays.

Specifically, we use timestamps $ts_{i,j,0}$ and $ts_{i,j,1}$, as illustrated in Figure 2a within the replay technique section. For lifecycle labeling, we rely exclusively on breakpoint data, specifically the initial breakpoint of the first technique and the final breakpoint of the last technique within a particular lifecycle (e.g., $ts_{1,0,0}$ and $ts_{1,n,1}$, where n denotes the final technique in the sequence).

C. *Data evaluation on reproducibility and learnability*

We utilize PCA and PCC to evaluate the reproducibility of replay attempts by examining structural patterns and linear correlations within the data. PCA reduces the dimensionality of the dataset, enabling focus on the primary sources of variance that capture the fundamental characteristics of the attacks. This is essential for recognizing fundamental similarities in the replayed data, regardless of noise or irrelevant features. PCC quantifies the linear correlation between the reduced datasets, offering a clear metric of similarity based on the alignment of the attack replays. This combination provides an effective approach for comparing replay attempts, prioritizing overall data structure and volume rather than calculation of individual actions.

Alternative dimensionality reduction techniques, such as t-SNE(t-distributed Stochastic Neighbor Embedding) and UMAP(Uniform Manifold Approximation and Projection) can be employed to effectively visualize high-dimensional data and capture non-linear relationships. However, both methods primarily focus on uncovering patterns and grouping similar data points rather than evaluating precise quantitative similarities. This is because they do not inherently preserve distances between data points in the same way that PCA does.

We employ a diverse set of machine learning classifiers to evaluate learnability, emphasizing the importance of leveraging varied models to enhance performance. The evaluation utilized AUC-ROC and F1-score metrics, which provide comprehensive insights into classifier performance. In particular, the AUC-ROC metric is valuable for analyzing true and false positive rates across different thresholds, making it especially effective for handling datasets with class imbalances.

V. Implementation

A. *ATT&CK-based technique and lifecycle attack reproduction*

1) ***CREME modification****:* In our previous work [13], we developed a toolchain that enabled the replication of attacks, the generation of datasets from various sources, the assessment of dataset quality, and the classification of data utilizing a binary classification.

This work introduced major modifications to the original tool and testbed architecture, leading to the creation of CREMEv2[3]. Key enhancements include adding a dedicated router and a host-only network to isolate the virtual environment from the main operating system. At the technical level, the attack code was deconstructed into functions, each representing a technique within the attack lifecycle. Start and end breakpoints were developed for each function to enable precise dataset labeling with appropriate techniques.

2) ***Selected Attack Lifecycle****:* This work reproduced five attack lifecycles, as outlined in Figure 1, with particular processes depicted in Figure 2b. Each lifecycle was precisely chosen to represent a wide range of cyber threats, including botnets, ransomware, resource hijacking, and Denial-of-Service (DoS) attacks. These specific attacks were selected for their distinctive characteristics and varying impacts, demonstrating the wide range of attack methods observed in actual cybersecurity incidents. The selection process was based on the ubiquity and relevance of these threats across several industries, making them highly illustrative of larger attack scenarios. Botnets and Denial of Service attacks were commonly employed for extensive disruptions, while ransomware and resource hijacking have evolved into more sophisticated and destructive threats aimed at both individuals and companies. By covering these variations, we ensured that the work covered the current trend of the cyberattack and covered almost all of the attack lifecycle from initial compromise to exploitation and impact.

In addition to replicating attack scenarios, we incorporated a range of benign behaviors during the dataset generation process. We configured HTTP, FTP, and SMTP services to emulate benign activity that runs on a benign machine. This arrangement was essential for guaranteeing that our dataset accurately represented realistic real-world scenarios.

B. *ATT&CK-based dual-labeled data labeling*

Our dataset integrated data from three distinct sources: accounting, Syslog, and traffic. Accounting data, collected using the Atop tool from five hosts (one benign server, one target, and three clients), was processed into a dataset containing detailed system-level information. This information includes process-specific features such as process ID (PID) and command (CMD), total memory (MEM) and virtual memory size (VSIZE), and page faults (MINFLT, MAJFLT). CPU utilization is represented by features CPU percent (CPU) and CPU core (CPUNR). Disk activity includes read/write bytes (RDDSK, WRDSK) and request size (RSIZE). The dataset also captures scheduling and priority features (PRI, POLI), process states (S), and resource growth (RGROW).

Syslog data, aggregated from the same machines using Rsyslog, was processed with the Drain technique to create a structured dataset. However, due to the textual nature of syslog data, it only consists of two features: content and component.

Traffic data was captured with TCPdump from six machines, including the five hosts plus one additional malicious client. The captured data was transformed into a traffic dataset using Argus. This dataset includes features such as source and destination port numbers (Sport, Dport), statistical properties (Mean, StdDev, Sum), and protocol flags (Flgs_). Other features include the total number of packets and bytes (TotPkts, TotBytes), flow directionality (SrcPkts, DstPkts), protocol states (State_), and protocol types (Proto_*).

In all datasets, common features such as host-specific identity(hostname or IP address) and timestamps were consistently available across data sources.

[3]https://github.com/HighSpeedNetworkLabNYCU/CREMEv2

TABLE II
COMPARISON WITH OTHER DATASETS

Datasets	Datasources[a]			Benign[b]	Attack Diversity						Reproduceable	Complete Kill Chain	Label	
	Tr.	Log	Acc.		Scan	Bruteforce	DoS	DDoS	Botnet	Msf[c]			ATT&CK Technique	ATT&CK Lifecycle
KDD99[5]	○	×	×	○	○	○	○	×	×	×	×	×	×	×
UNSW-NB15[6]	○	×	×	○	○	○	○	×	×	×	×	×	×	×
ISCX2012[7]	○	×	×	○	○	○	○	○	○	×	×	×	×	×
CIC-IDS2017[8]	○	×	×	○	○	○	○	○	○	×	×	×	×	×
DARPA1999[9]	○	○	×	○	○	×	○	×	×	×	×	×	×	×
CSE-CIC-IDS2018[10]	○	○	×	○	○	○	○	○	○	×	×	×	×	×
CREMEv1[11]	○	○	○	○	×	×	○	○	○	○	○	×	×	×
CREMEv2	○	○	○	○	○	○	○	○	○	○	○	○	○	○

[a] Different types of data sources; Tr: Network Traffic, Log: log or similar data sources (e.g., Windows event), Acc: accounting or host statistics related data sources.
[b] benign or normal behavior.
[c] Exploitation Metasploit framework.
[5] KDD Cup 1999 Dataset. "The KDD Cup 1999 Data." Available at: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.
[6] Moustafa, Nour, and Jill Slay. "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)." 2015 Military Communications and Information Systems Conference (MilCIS). IEEE, 2015.
[7] Shiravi, Ahmad, et al. "Toward developing a systematic approach to generate benchmark datasets for intrusion detection." Computers & Security, 2012.
[8] Sharafaldin, Iman, Arash Habibi Lashkari, and Ali A. Ghorbani. "Toward generating a new intrusion detection dataset and intrusion traffic characterization." International Conference on Information Systems Security and Privacy (ICISSP), 2018.
[9] Lippmann, Richard P., et al. "Evaluating intrusion detection systems: The 1998 DARPA off-line intrusion detection evaluation." DARPA Information Survivability Conference and Exposition, 2000.
[10] Lashkari, Arash Habibi, et al. "A detailed analysis of the CICIDS2017 dataset for comprehensive IDS benchmarking." ICISSP 2018. Available at: http://www.unb.ca/cic/datasets/ids-2018.html.
[11] Huu-Khoi Bui, Ying-Dar Lin, et al. "CREME: A toolchain of automatic dataset collection for machine learning in intrusion detection." Journal of Network and Computer Applications, 2021.

During the labeling phase, data was annotated with host-specific identity addresses and associated breakpoints. For the Mirai lifecycle, labels included details regarding the attacker, target, and both vulnerable and non-vulnerable client machines. For other lifecycles, labeling focused on the attacker and target systems only.

C. *Dataset evaluation on reproducibility and learnability*

Our dataset analysis centered on two primary concerns: reproducibility and learnability. To determine reproducibility, we established a ground truth dataset that can be accessed at Kaggle[4] and formulated four different scenarios. The first scenario replicated the attack precisely as it occurred on the ground truth, with additional scenarios extending the attack duration to generate larger datasets. We employed PCA and then PCC to evaluate linear correlations after matching features across different datasets.

We proposed a methodology to evaluate dataset learnability using a diverse set of machine learning classifiers with an 80:20 ratio for training and testing. All models were implemented using their default configurations from the scikit-learn and XGBoost libraries, focusing on AUC-ROC and F1-score metrics to ensure a comprehensive evaluation. XGBoost was selected for AUC-ROC analysis due to its ability to handle complex data and mitigate overfitting effectively. It was applied to all three datasets and evaluated in a One-vs.-All setting to measure classification accuracy and the classifier's capacity to differentiate across classes.

Additionally, standard hyperparameters were used for all models to streamline the evaluation process and focus on the datasets' inherent learnability. This approach ensured a consistent baseline for performance comparison across different models and datasets.

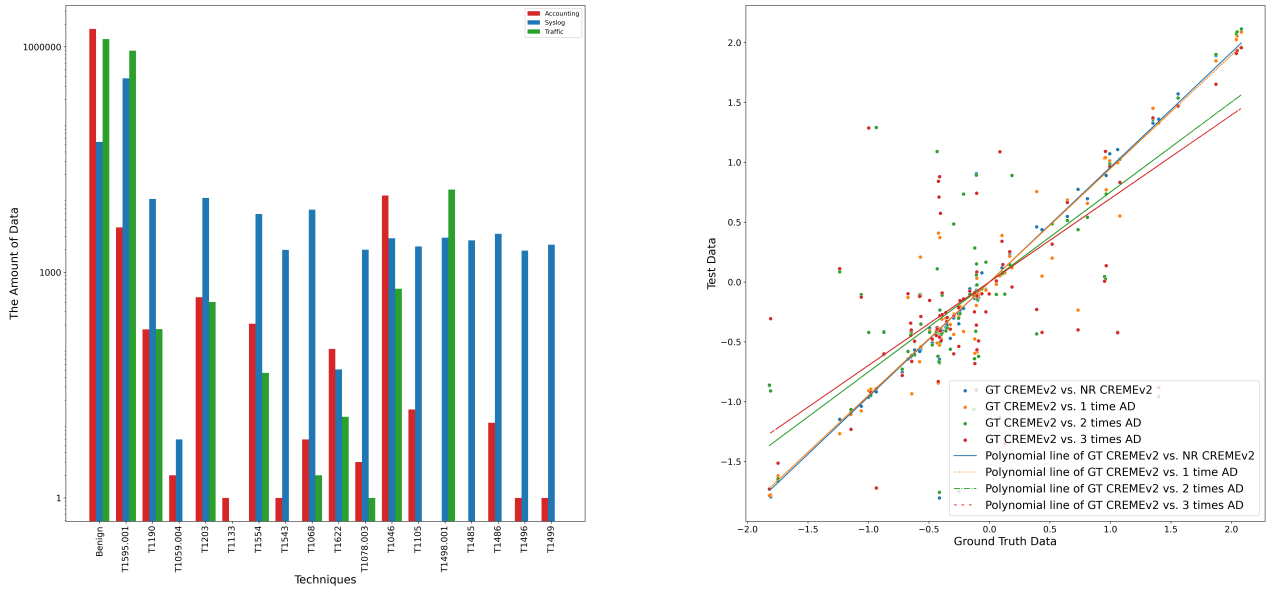
[4]https://www.kaggle.com/datasets/masjohncook/cremev2-datasets

VI. RESULT, DISCUSSION, AND LIMITATION

The comparison of the proposed dataset with other publicly available datasets is presented in Table II. Our proposed dataset demonstrates superior performance compared to others. The contributions of three data sources substantiate the findings, highlighting the integration of botnet and Metasploit frameworks to improve attack diversity, the application of the kill chain in attack replay, the incorporation of dual-labeling with technique and lifecycle, and the proposal of a tool for dataset reproduction.

On the other hand, to guarantee that a dataset is both reproducible and suitable for learning in machine learning applications, we evaluated reproducibility by conducting statistical tests to compare the dataset's consistency across various situations. Classification performance was evaluated by measuring it using AUC-ROC and F1-score.

This article has been accepted for publication in IEEE Network. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/MNET.2025.3551333



(a) Attack Replay Statistic

(b) Pearson Correlation Coefficient of All Scenarios

Fig. 3. Reproducibility Evaluation Result.

A. *Dataset reproducibility*

The resulting dataset comprises 4,583,180 samples (46 features) for accounting, 235,962 (61 features) for syslog, and 2,355,448 (59 features) for traffic, reflecting diverse data volumes across the sources. Figure 3a further illustrates the statistical distribution of attacks, showing the data volume for each technique across the three sources. Syslog contained the highest number of replicated techniques (15), followed by accounting (14) and traffic (11). While most ATT&CK techniques were replicated with substantial data volumes, a few had fewer than 100 instances.

Figure 3b shows the PCC graph comparing test data to ground truth. Lines represented scenarios: GT (ground truth), NR (normal replay), and AD (extended attack durations). The traffic dataset shows strong reproducibility, with data points closely aligned to the polynomial line and Pearson correlation coefficients exceeding 70%.

The decreased PCC score for each replayed scenario is attributable to the extended attack duration, which led to a rise in the quantity of data points. The inclusion of the attack duration led to a 30% amplification in data volume. Nonetheless, the dense data distribution and elevated PCC scores indicated that our system effectively produced data that was close to the original and maintained a degree of reproducibility.

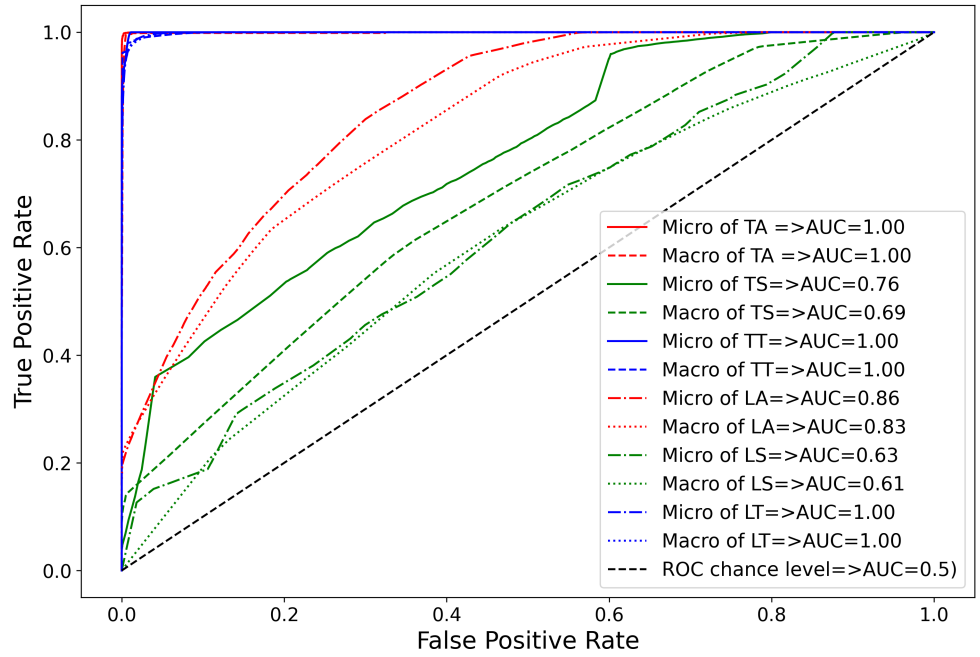
B. *Dataset learnability*

Figure 4a displays the mean AUC-ROC scores for these labels, assessed through both micro and macro averages. The micro average combines all classes' results before calculating the metric, providing an aggregate view and emphasizing larger classes. In contrast, the macro average calculates the metric for each class individually, offering a balanced assessment regardless of class size or distribution. This dual approach yields insights into both overall and individual class performance.

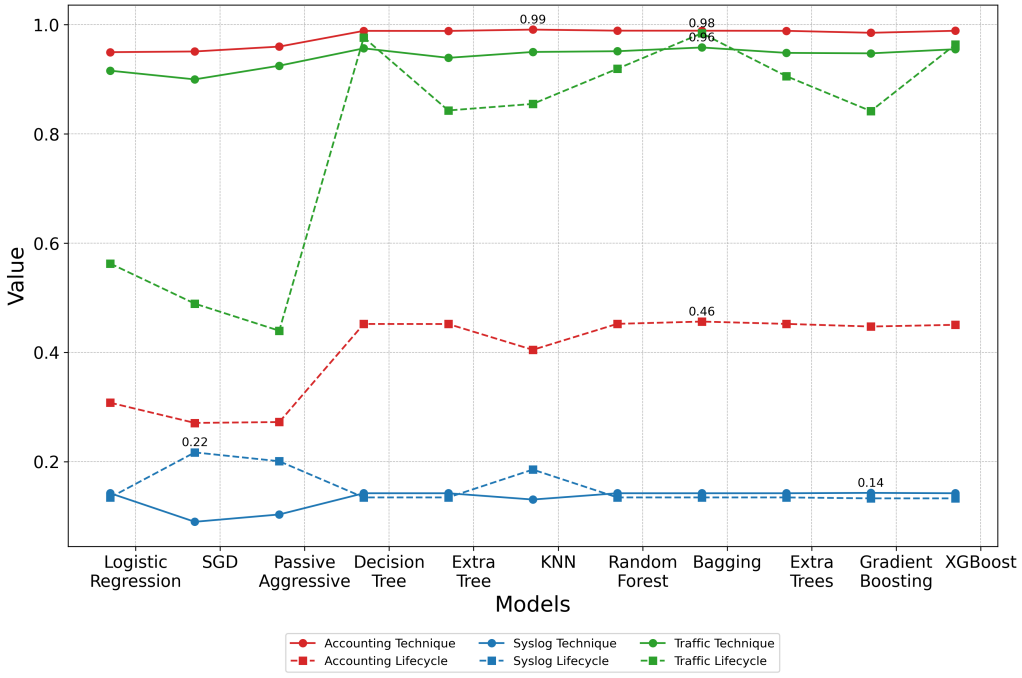
For the traffic and accounting datasets, classifications of technique labels achieved perfect scores on both micro and macro averages, with consistent performance across all classes. The Syslog dataset, however, demonstrated a notable difference: its micro score surpassed the macro score by 0.1 in the AUC-ROC metric. This discrepancy suggested a possible bias toward predominant classes, with limited generalization across diverse or underrepresented classes. In lifecycle label classification, the traffic dataset consistently delivered high AUC-ROC scores. For accounting and Syslog datasets, AUC-ROC scores for lifecycle labels were lower than those for technique labels, but the difference between micro and macro scores remained below 0.05, indicating a more consistent class performance with minimal bias. While AUC-ROC evaluations for technique labels were robust across datasets, lifecycle label classification showed a performance decline. The accounting dataset recorded a 0.14 reduction in micro scores and a 0.17 reduction in macro scores. For the Syslog dataset, micro and macro scores dropped to 0.63 and 0.61, respectively. On average, the models achieved a micro score of 0.92 for technique classification and a macro score of 0.82 for lifecycle classification across datasets. These results suggest that our machine-learning models show significant potential for technique classification within the traffic dataset

The F1-score results indicated significant differences in model performance across datasets and labeling schemes as depicted in figure 4b, highlighting the impact of dataset

This article has been accepted for publication in IEEE Network. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/MNET.2025.3551333



(a) Micro and Macro ROC Curve of All Datasets



(b) F1-Scores Evaluation Metric

Fig. 4. Evaluation Result

structure on classification results. The classification techniques tested for the Traffic and Accounting datasets achieved high F1-scores, reaching maximum values of 99% and 98%, respectively. The strong performance of technique classification is mainly due to clearly defined patterns throughout these datasets. The Traffic dataset had robust performance in lifecycle classification, achieving a maximum F1-score of 0.96. This result indicates that the dataset's configuration or feature extraction may enhance the effective capture of the lifecycles. In contrast, the performance of the Accounting dataset in lifecycle classification was under 50%, probably due to the fewer unique lifecycles presented in the data, which may restrict the

model's capacity to accurately distinguish across lifecycles. The Syslog dataset performs poorly in both technique and lifecycle classifications, with maximum F1-scores of merely 0.14 and 0.22, respectively. This underperformance may be attributed to the inadequate representation of the data. The situation is compounded by the lack of a specific preprocessing framework similar to that utilized for the Traffic data.

Training time for the machine learning models was also measured for both technique and lifecycle labels. The accounting dataset required average training times of 1670.5 seconds for technique labels and 89.5 seconds for lifecycle labels. The Syslog dataset required 737.5 seconds for technique labels and 57.9 seconds for lifecycle labels. The traffic dataset required 736.7 seconds for technique labels and 656.9 seconds for lifecycle labels. These results highlight the computational demands of different labeling approaches and the varying complexities of datasets during model training.

We report F1-scores as the primary metric due to their ability to balance precision and recall, which are critical for understanding model performance in classification tasks. In this work, the accuracy, precision, and recall metrics closely align with the reported F1-scores. Thus, the F1-score effectively represents overall classification performance and provides a comprehensive evaluation of related metrics.

The performance differences among datasets highlight the critical role of a strong, customized feature extraction technique. The Traffic data is improved with a structured framework that improves classification accuracy, whereas the accounting and Syslog datasets are constrained by insufficient data representation. This underscores the importance of specialized feature extraction methods for improving performance in a variety of data sources.

This work represents the first attempt to create a dataset with dual labeling for both MITRE ATT&CK techniques and attack lifecycles. Due to the novelty of this approach, the scope of attack scenarios replayed is limited to a predefined set of lifecycles and techniques.

VII. Conclusion and Future Work

This work introduced a novel approach for generating high-quality, dual-labeled datasets for machine learning, utilizing the ATT&CK. Our primary objective was to address and resolve the reproducibility challenges associated with existing dataset production methods. Our work yielded some significant conclusions:

1) **Automation:** The automated reproduction of attack scenarios indicates a substantial enhancement in efficiency, preserving considerable time and resources relative to manual dataset generation.
2) **Reproducibility:** Our innovative tool that automated the generation of the IDS datasets ensures reproducibility. Additionally, a novel evaluation method employing PCA and PCC was used to measure the reproducibility. This approach allows for consistent dataset production across various replay scenarios.
3) **Learnability:** The effectiveness of our dataset in identifying attack techniques and lifecycles was confirmed through the evaluation of the ML model through AUC-ROC and F1-score.

In future work, we aim to expand our solution by incorporating additional attack techniques and lifecycles. Furthermore, we will examine alternative preprocessing approaches to boost model performance, particularly in the context of accounting and syslog datasets.

References

[1] The MITRE Corporation. (2022, 4) Mitre att&ck®. [Online]. Available: https://attack.mitre.org/

[2] B. E. Strom, A. Applebaum, D. P. Miller, K. C. Nickels, A. G. Pennington, and C. B. Thomas, "Mitre att&ck: Design and philosophy," 3 2020.

[3] O. Grigorescu, A. Nica, M. Dascalu, and R. Rughinis, "Cve2att&ck: Bert-based mapping of cves to mitre att&ck techniques," *Algorithms*, vol. 15, p. 314, 8 2022.

[4] B. Ampel, S. Samtani, S. Ullman, and H. Chen, "Linking common vulnerabilities and exposures to the mitre att&ck framework: A self-distillation approach," *arXiv preprint*, 8 2021. [Online]. Available: http://arxiv.org/abs/2108.01696

[5] S.-X. Lin, Z.-J. Li, T.-Y. Chen, and D.-J. Wu, "Attack tactic labeling for cyber threat hunting," vol. 2022-February. IEEE, 2 2022, pp. 34–39. [Online]. Available: https://ieeexplore.ieee.org/document/9728949/

[6] D. Brauckhoff, A. Wagner, and M. May, "Flame: A flow-level anomaly modeling engine." CSET, 2008.

[7] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," *Computers & Security*, vol. 31, pp. 357–374, 5 2012.

[8] N. Rajasinghe, J. Samarabandu, and X. Wang, "Insecs-dcs: A highly customizable network intrusion dataset creation framework." IEEE, 5 2018, pp. 1–4.

[9] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," vol. 2018-January. SciTePress, 2018, pp. 108–116.

[10] R. Ishibashi, K. Miyamoto, C. Han, T. Ban, T. Takahashi, and J. Takeuchi, "Generating labeled training datasets towards unified network intrusion detection systems," *IEEE Access*, vol. 10, pp. 53 972–53 986, 2022.

[11] E. Vasilomanolakis, C. G. Cordero, N. Milanov, and M. Muhlhauser, "Towards the creation of synthetic, yet realistic, intrusion detection datasets." IEEE, 4 2016, pp. 1209–1214.

[12] C. G. Cordero, E. Vasilomanolakis, A. Wainakh, M. Mühlhäuser, and S. Nadjm-Tehrani, "On generating network traffic datasets with synthetic attacks for intrusion detection," *ACM Transactions on Privacy and Security*, vol. 24, pp. 1–39, 2 2021. [Online]. Available: https://dl.acm.org/doi/10.1145/3424155

[13] H.-K. Bui, Y.-D. Lin, R.-H. Hwang, P.-C. Lin, V.-L. Nguyen, and Y.-C. Lai, "Creme: A toolchain of automatic dataset collection for machine learning in intrusion detection," *Journal of Network and Computer Applications*, vol. 193, p. 103212, 11 2021. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1084804521002137

[14] S. Bagui, D. Mink, S. Bagui, T. Ghosh, T. McElroy, E. Paredes, N. Khasnavis, and R. Plenkers, "Detecting reconnaissance and discovery tactics from the mitre att&ck framework in zeek conn logs using spark's machine learning in the big data framework," *Sensors*, vol. 22, p. 7999, 10 2022.

[15] S. S. Bagui, D. Mink, S. C. Bagui, T. Ghosh, R. Plenkers, T. McElroy, S. Dulaney, and S. Shabanali, "Introducing uwf-zeekdata22: A comprehensive network traffic dataset based on the mitre att&ck framework," *Data*, vol. 8, p. 18, 1 2023.

Fietyata Yudha received an M.S. degree with a digital forensics specialization from Universitas Islam Indonesia (UII) in 2013. He is a lecturer and researcher at Universitas Islam Indonesia and a member of the Center for Digital Forensic Studies. He is currently pursuing a Ph.D. degree at EECS IGP, National Yang Ming Chiao Tung University (NYCU), Taiwan. His research interests include cybersecurity, machine learning, and digital forensics.

Ying-Dar Lin (Fellow, IEEE) is a Chair Professor of computer science at National Yang Ming Chiao Tung University(NYCU), Taiwan. He received his Ph.D. in computer science from the University of California at Los Angeles in 1993. His research interests include network softwarization, cybersecurity, and wireless communications. His work on multihop cellular was the first along this line and has been cited over 1000 times. He is an IEEE Distinguished Lecturer. He has served or is serving on the Editorial Boards of several IEEE journals and magazines and was the Editor-in-Chief of IEEE Communications Surveys & Tutorials during 2016–2020.

Yuan-Cheng Lai received his Ph.D. degree in the Department of Computer and Information Science from National Chiao Tung University in 1997. He joined the faculty of the Department of Information Management at the National Taiwan University of Science and Technology in August 2001 and has been a distinguished professor since June 2012. His research interests include performance analysis, software-defined networking, wireless networks, and IoT security.

Didik Sudyana received his Ph.D. degree in the Department of Electrical Engineering and Computer Science (EECS) from National Yang Ming Chiao Tung University (NYCU) in 2024. He is an Assistant Professor at Computer and Network Center of National Cheng Kung University (NCKU), Taiwan. His research interests include cybersecurity, cyber physical system, and network design and optimization

Ren-Hung Hwang (Senior Member, IEEE) received his Ph.D. degree in computer science from the University of Massachusetts, Amherst, Massachusetts, USA, in 1993. He is the Dean of the College of Artificial Intelligence at National Yang Ming Chiao Tung University (NYCU), Taiwan. Before joining NYCU, he was with National Chung Cheng University, Taiwan, from 1993 to 2022. He is currently on the editorial boards of IEEE Communications Surveys and Tutorials and IEICE Transactions on Communications. His research interests include deep learning, network security, wireless communications, the Internet of Things, and cloud and edge computing.