A Scalable Multi-Datasource IDS Dataset with Technique and Lifecycle Labels Based on MITRE ATT&CK

Fietyata Yudha*, Ying-Dar Lin†, Yuan-Cheng Lai‡, Ren-Hung Hwang§, Rasul Mankaev*
*EECS International Graduate Program, National Yang Ming Chiao Tung University, Hsinchu, Taiwan
†Department of Computer Science, National Yang Ming Chiao Tung University, Hsinchu, Taiwan
‡Department of Information Management, National Taiwan University of Science and Technology, Taipei, Taiwan
§College of Artificial Intelligence, National Yang Ming Chiao Tung University, Tainan, Taiwan
Emails: fyudha@cs.nycu.edu.tw, ydlin@cs.nctu.edu.tw, laiyc@cs.ntust.edu.tw, rhhwang@nycu.edu.tw, rasul.ee12@nycu.edu.tw

***Abstract*—Machine Learning-based Intrusion Detection Systems (ML-IDS) rely on high-quality datasets with structured labels to effectively identify complex and evolving cyber threats. However, most existing IDS datasets rely on single data sources and coarse labels, which restrict their ability to accurately model multi-stage adversarial behavior. To address these issues, we propose AR-MANO (Attack Reproduction with Management and Orchestration), a modular framework for orchestrating synchronized attack reproduction and data collection from various sources, including accounting, Syslog, and traffic. AR-MANO enables the creation of CREMEv3, a scalable, multi-source IDS dataset labeled with MITRE ATT&CK techniques and the attack lifecycle. We evaluated CREMEv3 using eight machine learning classifiers and obtained average F1-scores of 0.6483 for technique classification and 0.5410 for lifecycle classification without feature selection. After applying feature selection, these scores improved to 0.9572 and 0.8317, respectively. CREMEv3 outperformed CIC-IDS2017, CSE-CIC-IDS2018, and UNSW-NB15, with a class imbalance ratio of about 0.1, class entropy of 0.56, and a Gini coefficient of approximately 1.0. CREMEv3 provides a robust and scalable foundation for the development and evaluation of ML-based IDS.**

***Index Terms*—Intrusion Detection System, Machine Learning, MITRE ATT&CK, Dual-Labeled Dataset, Techniques, Lifecycles, Attack Reproduction**

I. Introduction

Intrusion Detection Systems (IDS) play a critical role in identifying and mitigating cyber threats. Traditional IDSs are mostly rule-based, which makes them challenging to adapt to new or evolving attack behaviors [1]. In contrast, Machine Learning-based IDS (ML-IDS) addresses this limitation by learning attack patterns from data. However, the effectiveness of ML-IDS is heavily reliant on the quality of the training dataset, particularly regarding the richness, structure, and accuracy of its labels [2].

Numerous public IDS datasets have primarily relied on binary or coarse-grained multi-class labels [3]. Notable examples include KDD99 [4] and NSL-KDD [5], which are legacy datasets that utilize network traffic data and only categorize attacks into broad types, such as denial-of-service or brute-force attacks. Subsequent datasets, like UNSW-NB15 [6] and CIC-IDS2017 [7], expanded the range of attacks but continued to focus exclusively on traffic data and generic labels. More recent efforts, such as CSE-CIC-IDS2018 [8], introduced host-level visibility through system logs. However, they still failed to align with standardized models of adversary behavior. Furthermore, these datasets predominantly depend on a single data source, most often network traffic, which limits their capacity to capture system-level or host-specific behaviors. To facilitate more contextual and effective threat detection, there is a growing demand for multi-datasource datasets that integrate traffic, logs, and accounting data into a cohesive and semantically rich representation of attack behavior.

The MITRE ATT&CK framework offers a structured taxonomy of adversary tactics, techniques, and sub-techniques [9]. It also models how attacks progress through sequential stages, often referred to as the adversary lifecycle. Integrating ATT&CK into dataset labeling enhances granularity by providing detailed labels that describe specific adversarial behaviors, rather than broad attack categories [10]. While these advantages make ATT&CK highly valuable for IDS dataset design, its adoption remains limited. For instance, UWF-Zeek [11] attempted to incorporate ATT&CK-based labeling but was restricted to tactic-level, lacking fine-grained technique or lifecycle information. More broadly, most existing IDS datasets rely on isolated labeling and do not include dual labels that capture both the technique used and the stage in which it occurs.

Dual labeling offers two distinct benefits. First, it enables the model to learn what technique is being used. Second, it enables the model to identify where the technique occurs within the lifecycle. This combination supports adversary tracking and sequence learning, which are critical for modeling real-world multi-stage intrusions. One prior dataset, CREMEv2, attempted to incorporate both techniques and lifecycle labels [12]. As the first effort to develop a dual-labeled IDS dataset, it relied on manually crafted lifecycles to demonstrate the feasibility of the approach. At that time, no automation tools were available to convert narrative threat descriptions into

TABLE I
SUMMARY OF WORK IN IDS DATASET GENERATION

Paper	Data [a] Sources	Attack [b] Diversity	Complete Kill Chain	Label		Scaleable	Automation	
				Technique	Lifecycle		Lifecycle Generation	Attack Reproduction
[4]	T	S, Bf, D	×	×	×	×	×	×
[6]	T	S, Bf, D						
[13]	T	S, Bf, D, D2, B						
[7]	T	S, Bf, D, D2, B						
[14]	T, L	S						
[8]	T, L	S, Bf, D, D2, B						
[15]	A, L, T	D, D2, B, E						
[12]	A, L, T	S, Bf, D, D2, B, E	○	○	○			
Ours	A, L, T	S, Bf, D, D2, B, E				○	○	○

[a] Different types of data sources that were collected during the attack reproduction process; T: Network Traffic, L: Log/Syslog, A: Accounting/Host Statistics
[b] Different types of attack that were launched during the attack reproduction process; S: Scanning, Bf: Bruteforce, D: Denial of Services, D2: Distributed Denial of Services, B: Botnet, E: Exploitation

structured execution plans. Consequently, the dataset depended heavily on manual scripting and was limited to five predefined lifecycles, inherently restricting its scalability.

To address the limitations, we adopt a structured and controllable attack reproduction strategy. Executing predefined attack lifecycles in a virtualized testbed enables the consistent and scalable generation of datasets. This approach enables step-by-step simulation of adversary behaviors while collecting synchronized data from multiple sources, including accounting, Syslog, and network traffic. Structured execution ensures that each technique is precisely mapped to a corresponding lifecycle, while event-based labeling aligns each label with its actual execution window. These properties improve label accuracy, facilitate multi-perspective alignment, and support the construction of semantically rich datasets suitable for training ML-based IDS models.

This work addresses these limitations by proposing an automated, scalable dataset generation pipeline tailored for dual labeling using ATT&CK techniques and lifecycle. The core challenges include: (1) reproducing multi-stage attacks in a consistent and extensible manner, (2) collecting synchronized data from accounting, Syslog, and traffic sources, and (3) assigning precise dual labels without manual intervention.

To achieve this goal, we are introducing AR-MANO (Attack Reproduction with Management and Orchestration), a virtualized orchestration framework built on the MANO architecture. AR-MANO executes predefined lifecycles encoded in modular YAML files, coordinates virtual machine (VM) operations, manages benign and adversarial behavior, and gathers synchronized data across multiple processes. It facilitates event-based dual labeling through timestamp-aligned execution. We assess the resulting dataset using the F1-score to evaluate learnability across eight machine learning classifiers and benchmark it against CIC-IDS2017, CSE-CIC-IDS2018, and UNSW-NB15, employing structural fairness metrics to confirm class balance and separability.

This work presents five primary contributions. Firstly, it introduces AR-MANO, an orchestration framework designed to automate the processes of attack reproduction, data collection, and labeling for the generation of dual-labeled IDS datasets. Secondly, it proposes a scalable lifecycle definition approach that utilizes modular configuration files, allowing for the incorporation of new attack scenarios without the need to rewrite existing scripts. Thirdly, it illustrates how this automation facilitates the efficient generation of synchronized, semantically rich datasets from multiple sources, including accounting, Syslog, and traffic data. Fourthly, it evaluates the resulting dataset using the F1-score across eight different classifiers to measure learnability under consistent training conditions. Finally, it benchmarks CREMEv3 against CIC-IDS2017, CSE-CIC-IDS2018, and UNSW-NB15 by employing both performance metrics and structural fairness indicators, such as imbalance ratio, class entropy, and the Gini coefficient, to validate the dataset's quality and practical utility.

II. RELATED WORKS

Table I summarizes prior efforts to generate datasets for Intrusion Detection Systems (IDS), comparing key properties including data sources, attack diversity, ATT&CK labeling (technique and lifecycle), completeness of the attack lifecycle, and support for scalable design and automation. The column "Data Sources" refers to the types of information collected during attack reproduction: A for accounting or host statistics, L for system logs (e.g., Syslog), and T for network traffic. "Attack Diversity" lists the types of adversarial behaviors included, such as Scanning (S), Bruteforce (Bf), Denial of Service (D), Distributed Denial of Service (D2), Botnet (B), and Exploitation (E). The "Complete Kill Chain" column indicates whether a dataset models a full lifecycle of multi-stage attacks rather than isolated actions. The label columns indicate whether ATT&CK technique-level or lifecycle-level labels were applied. The final columns indicate whether the dataset supports a scalable approach and includes automation in lifecycle generation and attack reproduction. The former represents the process of defining reusable and extensible attack lifecycles, typically through structured configuration

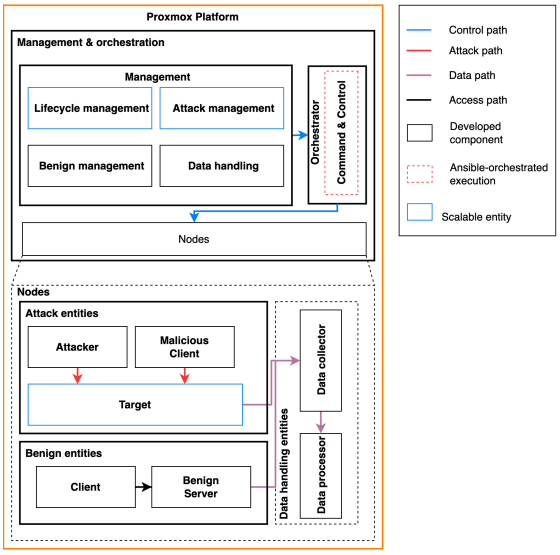


Fig. 1. AR-MANO Architecture

files, while the latter denotes the automatic execution of those lifecycles to generate labeled data.

Early efforts such as [4] and [6] focused solely on traffic data and labeled only a narrow set of attacks, including scanning, brute force, and denial of service. These datasets did not model complete attack progressions or provide semantic labeling of behavior. Later datasets such as [13] and [7] included more diverse attack types (e.g., botnets, distributed denial of service), but continued to rely exclusively on network traffic and did not adopt the ATT&CK framework.

More recent efforts have explored multi-source data collection to improve visibility. For example, [8] combined system logs with traffic data, adding a host-level perspective. However, these datasets still lacked ATT&CK-based labels and focused on isolated attacks rather than modeling structured sequences. The CREME framework introduced by [15] incorporated traffic, log, and accounting data, but did not support ATT&CK technique or lifecycle labeling. Its second generation, CREMEv2 [12], addressed these gaps by adopting dual-labeling aligned with ATT&CK. However, the lifecycle configurations were handcrafted, limited to five scenarios, and required manual scripting, making the system difficult to scale and extend to new attacks.

In contrast, our work introduces a dual-labeled dataset that integrates both ATT&CK techniques and lifecycles, supported by synchronized data collection across accounting, system logs, and traffic sources. We present AR-MANO, an orchestration framework that automates the end-to-end dataset generation process, including lifecycle execution, multi-source data capture, and structured labeling. Compared to prior works, our dataset supports more diverse attacks, full lifecycle modeling, and a scalable, automation-friendly pipeline, making it better suited for training and benchmarking ML-based intrusion detection systems.

III. AR-MANO: Solution for Scalable Dual-Labeled Dataset Generation

To address the challenges involved in dataset generation, this work presents a modular and automated framework known as AR-MANO (Attack Reproduction with Management and Orchestration). AR-MANO serves as a robust foundation for structured attack execution, facilitating synchronized data collection from various sources and dual-label assignment in accordance with the ATT&CK framework.

AR-MANO addresses three fundamental challenges: (1) *Complexity of Dual-Labeling*: this stems from the necessity for precise alignment between data sources and lifecycles; (2) *Lifecycle Scalability*: the scalability constraints associated with manual lifecycle scripting; and (3) *Evaluation and Comparison*: the lack of consistent evaluation practices for multi-datasource, dual-labeled IDS datasets. Although various metrics such as F1-score and label distribution indicators exist, they are rarely applied systematically to evaluate datasets that integrate both techniques and lifecycle labels.

The overall system architecture of AR-MANO is depicted in Figure 1. It operates on a Proxmox virtualization platform and consists of four primary functional components: *Management and Orchestration*, *Attack Entities*, *Benign Entities*, and *Data Handling Entities*. The *Management and Orchestration* layer is further divided into four submodules: *lifecycle management*, *attack management*, *benign management*, and *data handling*. These submodules establish execution logic and workflows based on configuration files. At its core, the *Command and Control* Orchestrator issues control commands and coordinates the behavior of nodes by utilizing Ansible playbooks, facilitating scalable and automated task execution.

The *Attack Entities* group comprises two roles: the attacker, who initiates exploit techniques as outlined in the lifecycle, and the malicious client, which facilitates attacks such as botnets. In contrast, the *Benign Entities* group comprises normal clients and a benign server that simulates legitimate traffic and background noise, ensuring the dataset captures a diverse range of behaviors. These roles operate in parallel to create a realistic and dynamic intrusion environment during data collection.

The *Data Handling* Entities include a Data Collector, which acquires raw data streams through tools such as tcpdump [16] (for traffic), Rsyslog [17] (for Syslog), and Atop [18] (for accounting). Complementing this, the Data Processor extracts features and prepares the data for labeling. The communication pathways are color-coded for clarity: blue lines signify control flow, red lines indicate attack traffic, black lines represent benign traffic, and purple lines illustrate data transfer to the collector. Entities managed via Ansible are denoted with dashed borders, while scalable components are emphasized with blue outlines. This architecture facilitates synchronized, modular, and scalable attack reproduction, with accurate labeling aligned with ATT&CK techniques and lifecycles.

According to the three fundamental challenges mentioned before, we introduce the proposed approaches to conquer them

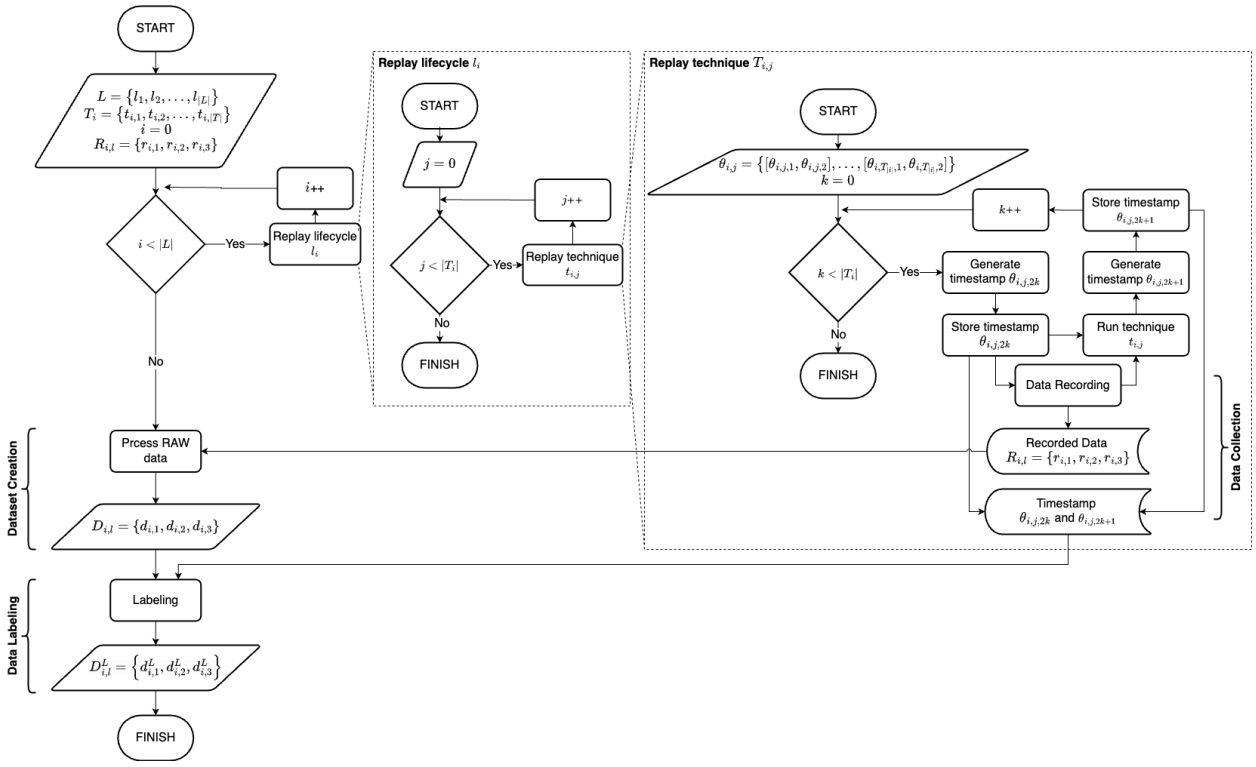


Fig. 2. Dataset Generation Process

below.

A. Complexity of Dual Labeling

Assigning ATTCK technique and lifecycle labels to intrusion detection datasets is challenging, especially when attacks involve multiple stages and sources like accounting, Syslog, and traffic. Temporal alignment of events with their techniques and lifecycles can be difficult, leading to mislabeling and compromised data quality. To address these issues, AR-MANO introduces a time-aligned, breakpoint-driven labeling pipeline that ensures consistency across all modalities.

The overall flow of the dataset generation process is shown in Figure 2. AR-MANO begins with a list of lifecycle configurations $L = \{l_1, l_2, \ldots, l_n\}$, where each lifecycle represents a specific attack scenario, such as ProFTPD 1.3.5 command execution, defined by an ordered sequence of ATT&CK techniques structured across logical stages. For each lifecycle l_i, the framework sequentially executes the associated techniques $T = \{t_{i,1}, t_{i,2}, \ldots, t_{i,j}\}$ as defined in the YAML configuration file. Each technique execution is determined by a pair of breakpoints $\theta_{i,j} = \{\theta_{i,j,2k}, \theta_{i,j,2k+1}\}$, which are logged by the controller at runtime to capture the execution window of each technique.

During execution, AR-MANO collects raw data from three synchronized sources: accounting, Syslog, and traffic. These sources are represented as raw data streams $R_{i,1}$, $R_{i,2}$, and $R_{i,3}$, respectively, where i denotes the lifecycle index and the second index $l \in \{1, 2, 3\}$ corresponds to accounting (1), Syslog (2), and traffic (3). After all techniques in l_i are executed, AR-MANO extracts features from each raw source to produce intermediate datasets $D_{i,l}$, where each $D_{i,l}$ is derived directly from its corresponding raw stream $R_{i,l}$. These datasets are then labeled using the stored breakpoints to assign: a technique label corresponding to the technique $t_{i,j}$ active during that time window, and a lifecycle label identifying the overall attack scenario l_i

The resulting dual-labeled dataset is denoted as $D^L_{i,l}$, which encapsulates both the execution-specific and scenario-level semantics. This notation ensures consistent mapping across all data modalities and enables multi-perspective learning for ML-IDS models.

The automation of the alignment process using breakpoints and lifecycle-aware execution in AR-MANO eliminates manual effort, reduces labeling errors, ensures consistency across data sources, and enhances understanding of adversarial behaviors through fine-grained learning.

B. Lifecycle Scalability

A major challenge in scalable dataset generation is the inflexible and manual definition of attack lifecycles. In previous work, each lifecycle was scripted in a hardcoded manner, making it unmanageable to modify or adapt scenarios. As the number of attack lifecycles grows, maintaining and expanding these lifecycles becomes increasingly inefficient and susceptible to errors.

AR-MANO addresses this limitation by introducing a modular, configuration-driven approach. Each attack lifecycle is defined through a structured YAML configuration that outlines

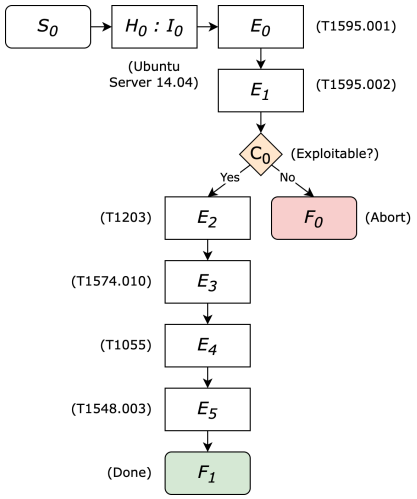


Fig. 3. Lifecycle Execution Graph for ProFTPD 1.3.5 Command Execution

the attack name, variant, and a sequential set of stages. Within each stage, one or more ATT&CK techniques are detailed, along with the corresponding parameters for exploitation modules, such as payloads, IP addresses, ports, and module types (e.g., exploit, shell, or handler). This format facilitates precise control over attack logic while effectively decoupling it from the underlying scripting implementation.

AR-MANO simplifies lifecycle management by using configuration files for execution logic, allowing for quick creation and modification. New scenarios can be added by extending YAML definitions without changing the reproduction engine. Automation can also help generate configurations, such as extracting technique sequences from CTF writeups or threat reports to improve coverage and reduce manual effort.

Figure 3 presents the lifecycle execution flow for a scenario: the ProFTPD 1.3.5 command execution attack. The process begins with reconnaissance techniques (T1595.001 and T1592.002), followed by a conditional check (C_0) to determine whether the target is exploitable. If successful, the attack proceeds through exploitation (T1203), persistence (T1574.010), and privilege escalation stages (T1055 and T1548.003). If the exploit condition fails, the lifecycle terminates early (F_0), capturing a realistic failure branch.

In the execution graph, each node represents a technique step, with conditional nodes (C_0) allowing branching. Failure states (F_0, F_1) indicate alternate exit paths, capturing both successful and failed executions. This structure, along with configuration-based lifecycle definitions, enables AR-MANO to model diverse adversarial behaviors efficiently, minimizing manual scripting efforts.

C. Evaluation and Comparison

In addition to generating a dual-labeled dataset, it is crucial to validate whether the resultant data supports effective machine learning and holds up against established benchmarks. Traditional metrics such as accuracy and precision may fall short in capturing structural quality, particularly in situations characterized by class imbalance or label overlap. Therefore, this work employs an evaluation strategy that emphasizes both learnability, measured through the F1-score, and structural fairness, assessed via imbalance ratio, class entropy, and the Gini coefficient. This approach facilitates a comprehensive quality assessment and allows for an equitable comparison with publicly available IDS datasets.

To evaluate learnability, we train machine learning classifiers on the generated dual-labeled dataset, utilizing standard performance metrics such as F1-score, precision, recall, and accuracy. The F1-score is prioritized as the primary metric due to its ability to balance precision and recall, which is particularly crucial in the context of imbalanced datasets. We assess model performance separately for technique prediction and lifecycle classification. Additionally, results are averaged across multiple classifiers and data sources, including accounting, Syslog, and traffic.

To enhance our understanding of the dataset's impact on classification performance, we compare the results obtained before and after feature selection. By retaining only the most relevant features from the complete dataset, we demonstrate that the dual-labeled structure retains sufficient discriminative information, even with a reduced number of features. This suggests that the dataset effectively captures significant adversarial patterns and facilitates generalizable learning.

To assess the quality of our dataset in a broader context, we perform a comparative analysis with three widely recognized IDS datasets: UNSW-NB15 [6], CIC-IDS2017 [7], and CSE-CIC-IDS2018 [8]. Since these datasets do not incorporate MITRE-based techniques or lifecycle labels, we use their attack labels as proxies for technique labels to maintain consistency in evaluation. Each dataset is subjected to a fair comparison setup, utilizing the same train-test split (80/20), identical machine learning models, and corresponding feature groups (e.g., flow duration, packet length, flag counts, etc.). We exclude metadata-related or source-specific features to enhance generalizability.

In addition to assessing classification performance, we also examine the structural fairness of each dataset through three distribution-aware metrics: imbalance ratio, class entropy, and Gini coefficient. These metrics are widely employed in imbalanced learning and intrusion detection to evaluate the statistical quality of label distribution, especially within multi-class classification scenarios.

- Imbalance Ratio (IR) measures the skewness between the most and least representative classes, defined as the ratio of samples in the majority class to those in the minority class. A lower IR indicates a more balanced distribution, which helps in training fair models that avoid overfitting to dominant classes [19].
- Class Entropy (CE) measures the diversity of class distribution using Shannon entropy on label frequencies. Higher entropy indicates a more uniform label distribution, enhancing generalization across attack types [20].
- The Gini Coefficient (GC) measures the uniformity of sample distribution across classes, helping to evaluate a

dataset's effectiveness in learning decision boundaries for various attack types [21].

These metrics offer a broader perspective on dataset quality, extending beyond typical classifier performance. They help assess class distributions for balance, diversity, and separability, which improves the fairness and robustness of machine learning-based intrusion detection systems (IDS).

In summary, these findings indicate that the dataset generated through AR-MANO is not only well-structured and learnable but also suitable for benchmarking against widely recognized datasets within the machine learning IDS community.

IV. Implementation Details

This section outlines the technical realization of AR-MANO, including testbed setup, attack execution, data collection, labeling, and evaluation. Each component supports the scalable, dual-labeled dataset generation process described in Section III.

A. Testbed Setup

AR-MANO is implemented on a Proxmox-based virtualization platform [22], which manages 12 virtual machines organized into four distinct roles: *Management and Orchestration*, *Attack Entities*, *Benign Entities*, and *Data Handling Entities*. The orchestration is facilitated through Ansible playbooks, which have successfully replaced over 90 legacy scripts with 20 modular workflows. Each lifecycle is designated for a specific target set, and breakpoints are documented to allow for accurate technique labeling.

B. Lifecycle Execution and Dataset Generation

Each attack lifecycle l_i is defined through a YAML configuration that outlines a sequence of ATT&CK techniques. In this study, these YAML-based lifecycles are utilized as predefined inputs. While the focus here is on automating execution, data collection, and labeling based on these configurations, a recent study [23] suggests a method for automatically generating such lifecycle definitions from unstructured Capture The Flag (CTF) writeups. This modular approach allows the current framework to function with reusable lifecycles, facilitating scalable dataset generation without the need for manual scripting.

Once a lifecycle configuration is established, AR-MANO parses the YAML file and executes each technique $t_{i,j}$ in succession. The controller dispatches commands to the attacker nodes and coordinates data capture, logging breakpoints $\theta_{i,j}$ to record the start and stop times of each technique. This arrangement allows for the seamless addition of new lifecycles by simply introducing more YAML files, without the need to alter the orchestration logic.

1) Lifecycle Configuration: Each YAML file specifies a lifecycle's name, type, and relevant ATT&CK techniques, along with associated parameters such as module names, IP addresses, and ports. AR-MANO transforms these specifications into a Python script, which is deployed through Ansible, allowing for branching logic and conditional paths. This configuration-based approach facilitates the scalable expansion of attack scenarios.

2) Multi-Datasource Collection: Three synchronized data sources are gathered: (1) accounting data via Atop, (2) system logs through Rsyslog, and (3) network traffic captured using Tcpdump. The raw files, denoted as $R_{i,1}$ for accounting, $R_{i,2}$ for Syslog, and $R_{i,3}$ for traffic, are stored for subsequent processing.

3) Labeling and Dataset Construction: Raw data is transformed into features, specifically host data (including accounting and Syslog), utilizing methodologies outlined in [24]. Traffic data is processed through NFStream [25]. The datasets denoted as $D_{i,l}$ are labeled using breakpoint techniques, resulting in the labeled datasets $D^L_{i,l}$ for each lifecycle and source. The final datasets are constructed by combining labeled segments across all identified lifecycles.

C. Evaluation and Comparison

We assess the learnability and structural fairness of the dataset. In terms of learnability, eight classifiers are trained using an 80/20 split and evaluated using the F1-score, both before and after feature selection. This approach emphasizes the model's capacity to recognize both technique and lifecycle labels under realistic training conditions.

Structural fairness is evaluated using three standard metrics: imbalance ratio (class distribution), class entropy (label diversity), and Gini coefficient (label separability). These metrics provide insight into the dataset's balance, expressiveness, and clarity in classification.

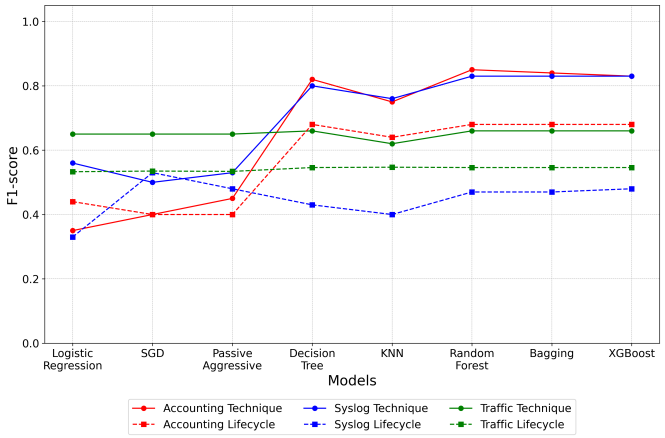
CREMEv3 is benchmarked against the UNSW-NB15, CIC-IDS2017, and CSE-CIC-IDS2018 datasets. Given that these datasets do not include MITRE labels, we utilize their native attack labels as proxies. A consistent feature grouping strategy is employed, retaining only five shared behavior-based groups: packet length, packet count, flow duration, TCP flags, and flow ratios. This normalization ensures a fair and unbiased comparison of models across the different datasets.

V. Evaluation

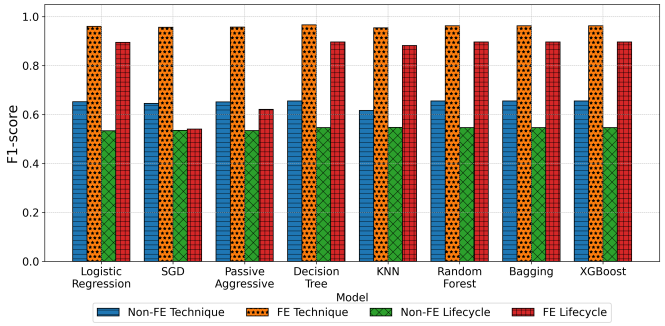
This section presents the experimental evaluation of the dataset generated by AR-MANO, focusing on two aspects: (1) learnability, how well machine learning models can classify techniques and lifecycles; and (2) comparative fairness, how CREMEv3 compares to existing public IDS datasets in terms of model performance and label structure. All evaluations use consistent classifier settings, an 80/20 split, and only non-metadata features, as defined in Section IV-C.

A. Learnability Evaluation

We evaluate eight standard classifiers: Logistic Regression, Stochastic Gradient Descent (SGD), Passive Aggressive, Decision Tree, K-Nearest Neighbors (KNN), Random Forest, Bagging, and XGBoost. Each model is trained using default hyperparameters with an 80/20 train-test split. The F1-score

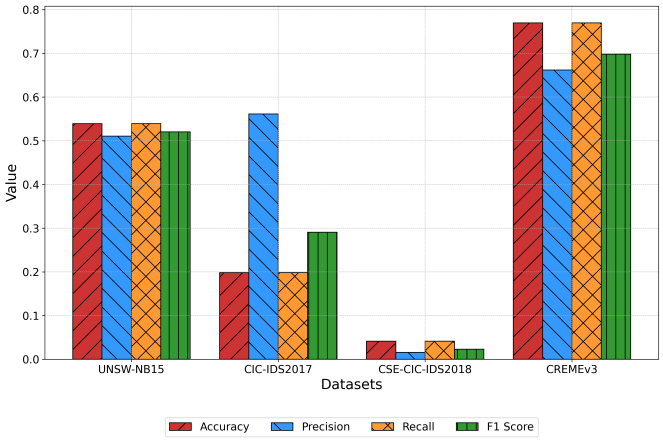


(a)

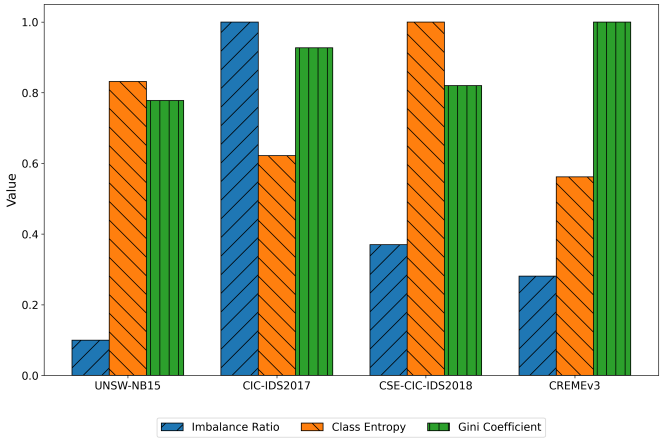


(b)

Fig. 4. Learnability Evaluation



(a)



(b)

Fig. 5. Comparison with Other Datasets

serves as the primary metric due to its reliability in situations with class imbalance.

As illustrated in Figure 4a, CREMEv3 demonstrates robust baseline performance across all three data sources. The technique classification yields average F1-scores of 0.661 for accounting, 0.705 for Syslog, and 0.651 for traffic, indicating that the dataset contains significant behavioral signals. Syslog achieves the highest performance, likely due to the rich contextual information found in log entries. In contrast, lifecycle classification proves to be more challenging, exhibiting a performance decline of 8–25% depending on the source. This decline is expected, as lifecycle labels necessitate abstraction over multiple stages rather than the detection of isolated patterns.

To assess the impact of semantic feature structuring on classification performance, we conduct a feature engineering experiment. In this context, non-FE indicates that feature selection was not applied, while FE refers to the application of feature engineering. Specifically, we implement statistical and semantic selection methods to eliminate irrelevant or redundant features. The results, illustrated in Figure 4b, demonstrate significant improvements: the F1-score rises from 0.6483 to 0.9572 for technique classification and from 0.5410 to 0.8317 for lifecycle classification. These enhancements indicate that aligning features with adversarial semantics boosts model learnability.

Despite using no hyperparameter tuning, CREMEv3 supports high-quality classification. This confirms that the dataset is inherently learnable and semantically rich, making it a strong candidate for benchmarking ML-based intrusion detection systems.

B. Comparison with Other Public Datasets

As indicated in Section IV-C, this study conducts a comparative analysis of CREMEv3 against the UNSW-NB15, CIC-IDS2017, and CSE-CIC-IDS2018 datasets. The comparison employs consistent training protocols and focuses on semantically grouped, non-metadata features.

Figure 5a illustrates that CREMEv3 achieves the highest average F1-score of 0.70, slightly surpassing UNSW-NB15, which has an F1-score of 0.68. Although CIC-IDS2017 demonstrates high precision, its low recall indicates that its models tend to overfit to the more frequent classes and struggle to generalize, likely due to the presence of imbalanced labels. Conversely, CSE-CIC-IDS2018 exhibits the lowest performance, with F1-scores falling below 0.10 across all classifiers. This aligns with previous findings regarding issues of noise, mislabeling, and flaws in feature extraction within the CICFlowMeter datasets [26].

The fairness metrics in Figure 5b further confirm the structural quality of CREMEv3. Its imbalance ratio of 0.1 suggests an even distribution of samples across classes, minimizing majority-class bias. A class entropy of 0.56 indicates a healthy variety of label categories, supporting generalization beyond dominant patterns. The Gini coefficient, which is close to 1.0, reflects strong label separability, a key requirement for building models with clear decision boundaries. Together, these results demonstrate that CREMEv3 is well-structured for both accurate and fair model training.

In summary, CREMEv3 demonstrates improved classification performance and stronger structural balance compared to existing public datasets. Among all evaluated datasets, CREMEv3 achieves the highest class entropy and Gini coefficient, reflecting diverse and well-separated label distributions. While its imbalance ratio is slightly higher than that of UNSW-NB15, it still significantly outperforms CIC-IDS2017 and CSE-CIC-IDS2018. This minor difference stems from UNSW-NB15 having fewer attack classes, resulting in naturally better balance. However, CREMEv3 covers a broader range of techniques and lifecycles while maintaining strong fairness metrics, making it more comprehensive and realistic.

VI. Conclusion

This work presents CREMEv3, a scalable, multi-datasource IDS dataset labeled with ATT&CK techniques and lifecycles. Built using the AR-MANO framework, it automates attack reproduction, synchronized data collection, and dual-label assignment across accounting, Syslog, and traffic. CREMEv3 addresses key limitations in existing datasets, such as coarse labels, single-source dependency, and limited behavioral coverage. Empirical results demonstrate strong learnability: for technique classification across all data sources, the average F1-score improves significantly after feature engineering, showing that the dataset captures semantically rich patterns. In comparison to CIC-IDS2017, CSE-CIC-IDS2018, and UNSW-NB15, CREMEv3 shows improved structural fairness, with a low imbalance ratio of 0.1, a class entropy of 0.56, and a Gini coefficient close to 1.0. These characteristics support balanced training and better class separation. While these results suggest that CREMEv3 is suitable for benchmarking ML-IDS solutions, future work is needed to evaluate how well the models trained on it generalize to real-world traffic.

References

[1] H. Liu and P. Patras, "Netsentry: A deep learning approach to detecting incipient large-scale network attacks," *Computer Communications*, vol. 191, pp. 119–132, 04 2022. [Online]. Available: https://doi.org/10.1016/j.comcom.2022.04.020

[2] Z. Ahmad, A. S. Khan, C. W. Shiang, J. Abdullah, and F. Ahmad, "Network intrusion detection system: A systematic study of machine learning and deep learning approaches," *Transactions on Emerging Telecommunications Technologies*, vol. 32, 10 2020. [Online]. Available: https://doi.org/10.1002/ett.4150

[3] H.-J. Liao, C. R. Lin, Y.-C. Lin, and K.-Y. Tung, "Intrusion detection system: A comprehensive review," pp. 16–24, 09 2012. [Online]. Available: https://doi.org/10.1016/j.jnca.2012.09.004

[4] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the kdd cup 99 data set," in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*. IEEE, 7 2009, pp. 1–6.

[5] G. Creech and J. Hu, "Generation of a new ids test dataset: Time to retire the kdd collection," in *2013 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 4 2013, pp. 4487–4492.

[6] N. Moustafa and J. Slay, "Unsw-nb15: A comprehensive data set for network intrusion detection systems (unsw-nb15 network data set)," in *2015 Military Communications and Information Systems Conference, MilCIS 2015 - Proceedings*. Institute of Electrical and Electronics Engineers Inc., 12 2015.

[7] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *ICISSP 2018 - Proceedings of the 4th International Conference on Information Systems Security and Privacy*, vol. 2018-January. SciTePress, 2018, pp. 108–116.

[8] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, "Developing realistic distributed denial of service (ddos) attack dataset and taxonomy," in *2019 International Carnahan Conference on Security Technology (ICCST)*. IEEE, 10 2019, pp. 1–8.

[9] T. M. Corporation, "Mitre attck®," 4 2022. [Online]. Available: https://attack.mitre.org/

[10] Y. Jiang, Q. Meng, F. Shang, N. Oo, L. T. H. Minh, H. W. Lim, and B. Sikdar, "Mitre attck applications in cybersecurity and the way forward," *arXiv (Cornell University)*, 02 2025. [Online]. Available: http://arxiv.org/abs/2502.10825

[11] 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 attamp;ck framework," *Data*, vol. 8, p. 18, 1 2023.

[12] F. Yudha, Y. Lin, Y. Lai, D. Sudyana, and R. Hwang, "Reproducing attck techniques and lifecycles to train machine learning classifier," *IEEE Network*, pp. 1–1, 03 2025. [Online]. Available: https://doi.org/10.1109/mnet.2025.3551333

[13] 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. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0167404811001672

[14] R. Lippmann, D. Fried, I. Graf, J. Haines, K. Kendall, D. McClung, D. Weber, S. Webster, D. Wyschogrod, R. Cunningham, and M. Zissman, "Evaluating intrusion detection systems: the 1998 darpa off-line intrusion detection evaluation," in *Proceedings DARPA Information Survivability Conference and Exposition. DISCEX'00*. IEEE Comput. Soc, 2000, pp. 12–26.

[15] 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

[16] S. McCanne and V. Jacobson, "The bsd packet filter: A new architecture for user-level packet capture." in *USENIX winter*, vol. 46. Citeseer, 1993, pp. 259–270.

[17] R. Gerhards, "The syslog protocol," https://datatracker.ietf.org/doc/html/rfc5424, 2009, rFC 5424.

[18] G. Langeveld, "atop," https://www.atoptool.nl/, accessed: 2025-05-22.

[19] S. Maheshwari, R. Jain, and R. Jadon, "A review on class imbalance problem: Analysis and potential solutions," *International journal of computer science issues (IJCSI)*, vol. 14, no. 6, pp. 43–51, 2017.

[20] Y. Orenes, A. Rabasa, J. J. Rodriguez-Sala, and J. Sanchez-Soriano, "Benchmarking analysis of the accuracy of classification methods related to entropy," *Entropy*, vol. 23, no. 7, p. 850, 2021.

[21] R. Sanasam, H. Murthy, and T. Gonsalves, "Feature selection for text classification based on gini coefficient of inequality," in *Proceedings of the Fourth International Workshop on Feature Selection in Data Mining*, ser. Proceedings of Machine Learning Research, H. Liu, H. Motoda, R. Setiono, and Z. Zhao, Eds., vol. 10. Hyderabad, India: PMLR, 21 Jun 2010, pp. 76–85. [Online]. Available: https://proceedings.mlr.press/v10/sanasam10a.html

[22] R. Goldman, *Learning Proxmox VE*. Packt Publishing Ltd., 2016.

[23] W.-C. Kew, Y.-D. Lin, F. Yudha, R.-H. Hwang, Y.-C. Lai, and H. G. Goh, "Attack lifecycle extraction and mapping from ctf writeups

using an enhanced llm approach," *(Preprint)SSRN*, 3 2025. [Online]. Available: https://ssrn.com/abstract=5180512

[24] Y.-D. Lin, T.-H. Loo, F. Yudha, Y.-C. Lai, and R.-H. Hwang, "Improving learnability in ml-based hids: Tuple-based aggregation, sentence embedding, and lstm feature extraction in log and accounting data," *(Preprint)SSRN*, 02 2025. [Online]. Available: http://ssrn.com/abstract=5143339

[25] Z. Aouini and A. Pekar, "Nfstream: A flexible network data analysis framework," *Computer Networks*, vol. 204, p. 108719, 2 2022.

[26] G. Engelen, V. Rimmer, and W. Joosen, "Troubleshooting an intrusion detection dataset: the cicids2017 case study," in *2021 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2021, pp. 7–12.