AI-Powered Detection of Advanced Persistent Threats (APTs): A Decision Tree Model for Intrusion Detection Using MITRE ATT&CK Behavioral Analysis

Asem Daoud* and Mohamed Hamdi

Higher School of Communication of Tunis, University of Carthage, Tunis, Tunisia
Email: asem.daoud@supcom.tn (A.D.); mmh@supcom.tn (M.H.)
*Corresponding author

***Abstract*—Advanced Persistent Threats (APTs) demand intrusion detection systems that are not only highly accurate but also operationally transparent and aligned with analyst workflows. This paper presents Decision Tree–Based Intrusion Detection System (DTB-IDS), a decision tree–based intrusion detection system that performs multi-level classification of network flows into (i) benign versus malicious traffic, (ii) MITRE Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK) tactics, and (iii) primary ATT&CK techniques. DTB-IDS is trained on a behaviorally enriched dataset obtained by carefully merging and harmonizing UWF-ZeekData24 and NF-UQ-NIDS-v2, yielding 628,415 Zeek-style flow records mapped to 13 tactics and 24 techniques. Using 33 engineered, semantically meaningful flow features and 3 specialized Classification and Regression Tree (CART) trees, DTB-IDS achieves strong performance on the merged dataset, with 99.2% binary accuracy, 99.1% tactic F1, and 99.4% technique micro-F1, together with a very low Hamming loss ($\approx 10^{-3}$). Cross-dataset validation on Canadian Institute for Cybersecurity Intrusion Detection Systems 2017 (CICIDS 2017) confirms the robustness of the learned decision boundaries, with 99.9% binary accuracy and 99.2% tactic F1 without retraining. Temporal and community-based splits further demonstrate that performance is sustained under stricter generalization regimes. Compared with Random Forest, Support Vector Machine (SVM), Deep Neural Networks (DNNs), eXtreme Gradient Boosting (XGBoost), Light Gradient Boosting Machine (LightGBM), and Explainable Boosting Machine baselines, validated by McNemar's and paired t-tests. Feature-importance analysis and rule-path inspection show that a small set of interpretable, ATT&CK-aligned features and decision paths account for most predictions, making DTB-IDS a practical and transparent APT detection solution for security operations centers.**

***Keywords*—Advanced Persistent Threat (APT), Intrusion Detection System (IDS), decision tree, MITRE Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK), interpretability, security operations**

I. Introduction

Recently, Advanced Persistent Threats (APTs) have emerged as some of the most destructive and covert cyber threats [1]. Unlike other cyberattacks (e.g., widespread ransomware that seeks quick disruption and financial gain), APTs focus on long-term strategic objectives and are characterized by covert tactics and complex, multi-step strategies [2]. APT campaigns are typically executed by well-funded, sophisticated adversaries (such as nation-states or organized threat groups) who infiltrate target networks and remain undetected for extended periods—often months or even years [3].

A notable example is the 2020 SolarWinds supply chain attack, an APT campaign that went unnoticed for over 9 months while compromising thousands of organizations and U.S. government agencies, underscoring the difficulty of timely APT detection [4].

Additionally, according to a 2024 report by Kaspersky, APT incidents accounted for 43% of high-risk security events and affected 25% of enterprises, a 74% year-over-year increase [5]. Fig. 1 summarizes the sectors most affected. The telecom and technology sectors saw APT detections increase by 92% and 119%, respectively, in Q1 2025 compared to Q4 2024 [6].

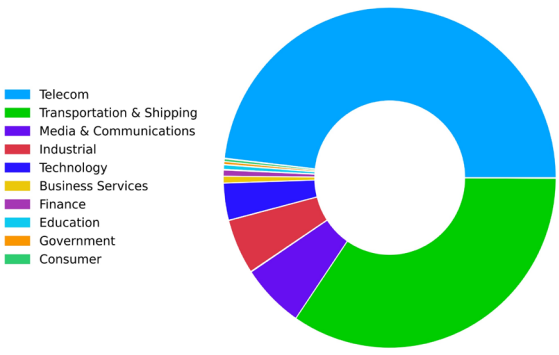


Fig. 1. Top 10 targeted sectors by Advanced Persistent Threat (APT).

Manuscript received October 10, 2025; revised November 16, 2025; accepted December 26, 2025; published May 13, 2026.

The main contributions of this paper are as follows.

(1) We present a unified Intrusion Detection Systems (IDSs) framework that performs multi-level APT classification, distinguishing malicious vs. benign activity, identifying the MITRE Adversarial Tactics, Techniques, and Common Knowledge framework (ATT&CK) of each attack (e.g., reconnaissance vs. exfiltration), and pinpointing the technique. To our knowledge, this publication is one of the few works that demonstrate a decision-tree ML model applied across these APT classification granularity levels in a single system.

(2) We construct a new APT-oriented dataset by carefully merging and preprocessing the UWF-ZeekData24 and NF-UQ-NIDS-v2 corpora, extracting 33 interpretable flow-level features, and mapping each malicious event to a primary attack tactic and technique.

(3) We benchmark Decision Tree–Based Intrusion Detection System (DTB-IDS) against several strong baselines, including Random Forest, Support Vector Machine (SVM), Deep Neural Networks (DNNs), eXtreme Gradient Boosting (XGBoost), Light Gradient Boosting Machine (LightGBM), and an Explainable Boosting Machine (EBM), using identical features and hyperparameter tuning protocols. We complement standard performance metrics with McNemar's tests, paired t-tests, effect size analysis, confidence intervals, temporal validation, and cross-dataset evaluation on Canadian Institute for Cybersecurity Intrusion Detection Systems 2017 (CICIDS 2017) to assess generalization and overfitting risks.

(4) We conduct multiple validation approaches, including 5-fold cross-validation, temporal validation (time-based splitting), external dataset validation on CICIDS 2017, and data leakage audits with adversarial validation to ensure robust generalization and rule out overfitting.

The remainder of this paper is organized as follows: Section II talks about APTs, the MITRE Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK) lifecycle, IDS methods, and other ML-based detection methods, such as knowledge graph methods. Section III describes the dataset and our feature engineering process. Section IV details the proposed Decision Tree–Based Intrusion Detection System (DTB-IDS) architecture. Section V presents experimental results including classification performance and comparisons. Section VI discusses interpretability of decision paths and the mapping of detections to MITRE tactics/techniques and incident response actions and privacy considerations. Section VII concludes with future research directions.

II. Background

A. Advanced Persistent Threats (APTs)

Mandiant first coined the term "APT" in our January 2010 M-Trends report to describe cyber-espionage campaigns primarily attributed to state-sponsored groups. Since then, numerous high-profile APT groups have been identified, including APT1, APT28, APT29, APT44, etc., and all of them are linked to large-scale espionage campaigns targeting governments, defense organizations, healthcare, finance, and critical infrastructure worldwide [7].

APTs, unlike opportunistic attacks, carefully orchestrate intrusions over multiple stages using advanced techniques. For instance, attackers may leverage zero-day exploits or fileless malware to establish an initial foothold. Persistence refers to the threat's long-term presence; attackers maintain covert access and lateral movement within the compromised network, often blending in with normal activity using "low-and-slow" tactics [8]. Traditional rule-based Intrusion Detection Systems (IDSs), such as signature-based antivirus or network monitors, are often ineffective against APTs because they rely on static signatures of known attacks. This makes them blind to novel exploits and the stealth techniques APT attackers employ, leading to high false-positive rates [9].

In response, the security community has increasingly turned to intelligent, behavior-based detection methods. Research shows that machine learning methods can greatly improve the detection of APT activities by learning complex patterns and finding anomalies that go beyond traditional signatures in real time [10].

However, many high-performance models, especially Deep Neural Networks (DNNs), suffer from a "black box" problem, lacking transparency in their decision-making. Such an issue is problematic in cybersecurity, where analysts need to understand an alert's rationale (e.g., which tactic or technique was detected) to respond effectively [11, 12]. Thus, balancing model transparency and detection accuracy has become a central challenge in developing "smart" IDS solutions [13].

B. Stages of the MITRE ATT&CK Lifecycle

The MITRE Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK) framework is a widely used knowledge base that keeps track of the Tactics, Techniques, and Procedures (TTPs) that hackers use in real-world attacks [14]. The Enterprise ATT&CK matrix defines 14 tactics that represent high-level adversary objectives during an intrusion (e.g., reconnaissance, initial access, execution, persistence, privilege escalation, defense evasion, credential access, discovery, collection, command and control, exfiltration, and impact). Each tactic is made up of one or more techniques and sub-techniques, each with its own unique identifier (for example, T1110: Brute Force, T1021: Remote Services) [15].

By 2025, ATT&CK v17.1 had more than 200 techniques and more than 400 sub-techniques, as well as hundreds of documented software, groups, mitigations, and analytic notes. This made it one of the most complete collections of adversary tradecraft available to defenders.

ATT&CK has grown from a simple reference tool to a common language for cybersecurity experts. This has made it possible for people from different fields to work together on threat intelligence sharing, red/blue team exercises, and defensive gap analysis [16]. Mapping system alerts, events, and telemetry to ATT&CK tactics

gives defenders a better idea of what the enemy is trying to do and how the behavior they see fits into a larger intrusion campaign [17].

In practice, the MITRE ATT&CK lifecycle represents a behavioral modeling approach that enhances cyber defense far beyond traditional signature-based detection. By integrating ATT&CK mapping into AI-augmented intrusion detection systems, organizations can correlate low-level anomalies with higher-order attack objectives, producing tactic- and technique-level labels that enhance explainability and situational awareness [18].

As such, the framework has become the semantic foundation for modern cyber threat intelligence and defensive analytics, ensuring that defenders can reason not only about events but also about intentions, a vital shift in the era of complex, adaptive threats like Advanced Persistent Threats (APTs).

C. Intrusion Detection Systems (IDS)

Intrusion Detection Systems (IDSs) are a cornerstone of modern network and host security architecture, designed to continuously monitor events and raise alerts upon identifying malicious activity, policy violations, or abnormal behavior [19]. These systems can be categorized along 2 major axes (data source and detection technique). From a data-source perspective, Network-based IDSs (NIDSs) analyze traffic at the packet or flow level to infer intrusions from network behavior, but they must cope with challenges such as high data throughput, encrypted channels, and dynamic protocols [20]. Host-based IDSs (HIDSs), in contrast, operate at the endpoint level, monitoring system calls, processes, and file changes, which allows for fine-grained visibility into insider threats and post-compromise behavior [21]. Newer hybrid frameworks even advocate network-wide orchestration of host-based IDS agents to enable coordinated real-time response and correlation of alerts across systems [22].

From the perspective of detection techniques, signature-based IDSs use predefined attack patterns or rule sets (like Snort or Suricata) to find known threats. This works well, but they can't adapt to new or zero-day attacks. In contrast, anomaly-based IDSs rely on machine learning to establish a baseline of normal system behavior and flag deviations as potential threats, enabling zero-day detection but increasing false positives [23].

Recent studies propose AI- and ML-enhanced IDSs that fuse multiple telemetry sources, network flows, host logs, authentication events, and file activity to detect multi-stage, stealthy attacks such as Advanced Persistent Threats (APTs) [24]. The most promising advances come from Explainable Artificial Intelligence (XAI) frameworks that enhance IDS transparency. These frameworks (e.g., SHapley Additive exPlanations (SHAP), Local Interpretable Model-agnostic Explanations (LIME)) make model predictions interpreted by identifying which features contributed to anomaly detection, thus improving analyst trust and reducing alert fatigue [25].

Furthermore, MITRE ATT&CK-aligned IDS architectures now enable detection systems to contextualize anomalies within adversary Tactics, Techniques, and Procedures (TTPs), providing semantically rich alerts tied to attacker intent [26].

This change in behavioral modeling, from raw anomaly scoring to tactic-level reasoning, provides defenders the ability to do threat hunting, forensic analysis, and proactive mitigation with more accuracy.

As of 2025, integrated systems such as DeepOP (a hybrid deep-learning-and-ontology framework for MITRE ATT&CK sequence prediction) have demonstrated the feasibility of predicting attacker sequences using deep learning and MITRE ATT&CK mappings, achieving over 95% detection accuracy on benchmark datasets [27]. These innovations position IDSs not just as reactive monitoring tools but as intelligent, explainable, and adversary-aware systems integral to modern cyber defense architectures that make model predictions interpretable by identifying which features contributed to the anomaly.

D. Decision Trees in IDS: Explainable AI for Actionable Cybersecurity

Decision trees have become an essential component of modern understandable Intrusion Detection Systems (IDSs). They operate by dividing the feature space into hierarchical regions using simple conditional rules that test parameters such as packet counts, connection durations, or protocol types. Each terminal node (leaf) then assigns a classification label, such as benign activity or a specific attack type. This structure makes decision trees inherently interpretable because analysts can easily trace the logical sequence that led to a particular prediction and assess its validity [28].

Recent studies indicate that decision-tree and rule-based algorithms make intrusion detection systems much more accurate and clearer. Explainable AI (XAI) approaches using decision trees have enabled systems to map anomalies to MITRE ATT&CK tactics and techniques, bridging the gap between low-level IDS alerts and higher-level adversarial objectives. When combined with explainability tools such as SHAP and LIME, decision trees identify attacks and explain which network or host features contributed most to a detection outcome [29].

In the context of Advanced Persistent Threat (APT) detection, decision trees are often incorporated into hybrid and multi-level frameworks. They serve either as baseline classifiers or as interpretable modules within deep learning pipelines that detect multiple stages of an attack lifecycle, including reconnaissance, lateral movement, and command-and-control operations [30]. When used with the MITRE ATT&CK matrix, decision-tree-based IDSs can say which tactic (like privilege escalation) or technique (like credential dumping) has been found instead of giving vague anomaly scores.

This alignment provides analysts with actionable intelligence and enhances their situational awareness, making decision trees a powerful tool for designing human-understandable, adversary-aware, and explainable intrusion detection systems.

E. Knowledge Graph Approaches for APT Detection

As APT campaigns have grown more complex and stealthier, researchers have increasingly turned to

Knowledge-Graph (KG) techniques to capture relationships between threat actors, malware families, infrastructure, vulnerabilities, and ATT&CK techniques. By representing Cyber Threat Intelligence (CTI) as a graph, defenders can perform attribution, hypothesis generation, and link analysis that go beyond what is possible with isolated indicators.

TABLE I. COMPARISON BETWEEN THE PROPOSED DTB-IDS MODEL & (CSKG4APT, AEKG4APT)

Comparison Criterion	DTB-IDS (Ours)	CSKG4APT	AEKG4APT
Primary Goal	Multi-level APT detection and classification (binary/tactic/technique) using decision-tree mapping to MITRE ATT&CK v17 for interpretable behavioral analysis.	To provide a comprehensive platform for APT attribution by integrating fragmented intelligence and enabling proactive defense strategies.	To automate the extraction of actionable Cyber Threat Intelligence (CTI) using LLMs and Knowledge Graphs (KGs) to enhance situational awareness.
Input Data	Network telemetry + host logs + MITRE ATT&CK technique labels + behavioral features engineered from CTI and attack traces.	Static & dynamic malware reports, Indicators of Compromise (IoCs) (IPs, domains, hashes), CTI feeds, threat actor profiles.	CTI feeds, sandbox reports, APT incident records + text mined via Large Language Models (LLMs).
Detection Speed	High speed decision-tree model is lightweight and suited for real-time IDS deployment.	Moderate focuses on post-incident attribution and graph linking rather than real-time detection.	Potentially High, due to the automation of CTI extraction and rapid integration into defense strategies
Interpretability	Very High provides transparent decision rules that explicitly show the mapping to MITRE ATT&CK techniques.	High, the structured representation of the KG offers a clear view of threat relationships, aiding analyst interpretation.	Moderate to High, LLMs enhance semantic understanding, but their "black box" nature can reduce direct transparency compared to decision rules.
Knowledge Source	MITRE ATT&CK v17 framework, CTI datasets, real APT incident traces, and labeled attack behavior data.	Open-Source Cyber Threat Intelligence (OSCTI) feeds, malware databases, APT reports, and IOC repositories.	CTI sources + APT incident texts processed by LLMs + automated KG construction pipelines.
Deployment	Very Lightweight (847 KB), suitable for resource-constrained environments.	Not explicitly specified, but KGs are typically heavier systems requiring robust infrastructure for querying and analysis (GB-scale graph).	Heavy, due to the reliance on large LLMs, which require significant computational resources for processing and updating (LLM + graph)
Strengths	• Real-time APT detection. • Explainable, MITRE-aligned results. • High accuracy & low complexity. • multi-level classification.	• Proactive defense approach. • Integrates fragmented intelligence. • Enables dynamic defense strategies	• Automated CTI extraction reduces manual effort. • Enhanced situational awareness. • Improved accuracy and integration of threat intelligence. • Can continuously update APT relations from new reports.
Weaknesses	• Limited long-term contextual linking. • Requires manual feature engineering. • Risk of overfitting on imbalanced data.	• Manual ontology maintenance and slow updates. • May face challenges in real-time detection speed due to its focus on comprehensive threat understanding.	• Faces challenges in scaling LLMs for large-scale data processing and KG construction • Potential for false or noisy relations. • Increased computational cost.
Use Case	Real-time security monitoring in Security Operations Centers (SOCs), lightweight Intrusion Detection Systems (IDS)	Organizations seeking a comprehensive platform for APT attribution and proactive defense strategy development.	Environments requiring real-time threat detection and enhanced situational awareness through automated CTI extraction

Ren *et al.* [31] proposed Cybersecurity Knowledge Graph for APT organization attribution (CSKG4APT), a cybersecurity knowledge graph for APT organization attribution that links APT groups, malware, vulnerabilities, and ATT&CK techniques. This structure enables strategic-level reasoning about "who is behind the attack" and which tools and techniques they typically employ. More recently, AEKG4APT combines Large Language Models (LLMs) with knowledge graphs to automatically extract entities and relations from unstructured CTI reports, supporting semantic querying and link prediction over evolving APT campaigns [32].

Other works construct APT malware knowledge graphs and apply multi-stage graph computation for refined attribution and campaign analysis [33, 34]. These KG-based approaches focus primarily on "threat intelligence fusion and attribution" and typically operate at minute-to-hours timescales rather than at per-flow, real-time IDS speeds. They also require substantial infrastructure and careful control of LLM hallucinations in high-stakes environments [35]. Table I presents a comparison with our proposed DTB-IDS approach, highlighting key differences in detection goals, operational deployment considerations, interpretability, and system complexity.

III. DATASET AND FEATURE ENGINEERING

A. Dataset Description and Preprocessing

To support multi-level APT detection aligned with the MITRE ATT&CK framework, we constructed a unified dataset by harmonizing 2 publicly available datasets. Each dataset contributes complementary characteristics UWF-ZeekData24 provides high-fidelity Zeek logs and detailed flow semantics [36]. In contrast, NF-UQ-NIDS-V2 includes multi-stage adversarial scenarios representing

reconnaissance, lateral movement, exfiltration, and command-and-control behaviors [37].

Their integration results in a dataset that more comprehensively captures enterprise-relevant APT behaviors than either dataset individually. The merging process followed a structured harmonization pipeline.

1) Schema alignment

Standardizing field names, timestamp formats, and flow semantics (e.g., mapping Zeek's orig_bytes and NF-UQ's in_bytes to a unified representation).

2) Type normalization

Converting categorical fields such as protocol, service, conn_state, and history into encoded integer or one-hot formats.

3) Flow unification

Representing all connections using a consistent bidirectional format: (src_ip, dst_ip, src_port, dst_port, proto, start_time, duration, orig_bytes, resp_bytes, orig_pkts, resp_pkts ...etc).

4) Integrity filtering

Removing malformed entries, incomplete flows, and duplicated records produced during dataset merging.

5) Feature extraction

Computing engineered behavioral features described in Section III.

6) Stratified splitting

Ensuring balanced representation of rare tactics across training and testing subsets.

Identifiers with high cardinality (such as full IP addresses) were abstracted into semantic attributes (internal vs. external) to prevent overfitting. After preprocessing, the final merged dataset contained 628,415 events, spanning benign flows and adversarial samples mapped to 13 tactics and 24 techniques.

B. Labeling and Mapping to MITRE ATT&CK

Each connection record was labeled over 3 levels of granularity.

- Binary label: benign (0) or malicious (1).
- Tactic label: the primary ATT&CK tactic exhibited (e.g., Reconnaissance, Credential Access, Lateral Movement).
- Technique label: the corresponding ATT&CK technique ID (e.g., T1021—Server Message Block (SMB) lateral movement). Labeling followed a structured behavioral-interpretation methodology. 2 researchers with experience in cyber-threat analysis independently mapped events to ATT&CK tactics and techniques based on flow semantics, timing patterns, connection states, and service usage. A reconciliation stage resolved initial discrepancies, ensuring consistent interpretation across all scenarios. This double-review approach aligns with established ATT&CK-based annotation practices.

TABLE II. CLASS DISTRIBUTION STATISTICS ACROSS CLASSIFICATION LEVELS

Level	Class/Tactic/Technique	Count	Percentage
Binary Level	Benign	342,156	54.5%
	ATTACK	286,259	45.5%
	Total	**628,415**	**100%**
Tactic Level	Benign	342,156	54.5%
	Reconnaissance	48,234	7.7%
	Resource Development	52,108	8.3%
	Initial Access	41,892	6.7%
	Execution	8456	1.3%
	Persistence	12,340	2.0%
	Privilege Escalation	9876	1.6%
	Defense Evasion	6234	1.0%
	Credential Access	38,765	6.2%
	Discovery	35,421	5.6%
	Lateral Movement	29,087	4.6%
	Collection	3456	0.6%
	Command and control	31,245	5.0%
	Exfiltration	2890	0.5%
	Total	**628,415**	**100%**
Technique Level (Examples)	T1595—Active Scanning	25,678	4.10%
	T1587—Develop Capabilities	52,108	8.30%
	T1190—Exploit Public-Facing Application	18,943	3.00%
	T1566—Phishing	22,949	3.70%
	T1059—Scripting Interpreter	8456	1.30%
	T1053—Scheduled Task/Job	7892	1.30%
	T1078—Valid Accounts	18,234	2.90%
	T1110—Brute Force	20,531	3.30%
	T1046—Network Service Scanning	19,876	3.20%
	T1021—Remote Services	15,432	2.50%
	T1071—Application Layer Protocol	31,245	5.00%
	T1048—Exfiltration Over Alternative Protocol	2890	0.50%
	T1105—Ingress Tool Transfer	14,678	2.30%

Examples of observed flow-level behaviors used to map network activity to MITRE ATT&CK tactics/techniques are as follows.

- TA0010: Long-duration flows with predominantly outbound bytes, consistent with data transfer outside the network, mapped to the Exfiltration tactic.
- TA0011: Periodic, low-volume DNS/HTTP bursts consistent with beaconing behavior, mapped to the Command-and-Control tactic.

This hierarchical labeling embeds contextual semantics into the classification task and allows the model to express detection results in a form suitable for SOC workflows.

C. Class Distribution and Imbalance Management

One critical aspect of dataset quality is understanding and managing class distribution. We provide comprehensive statistics showing distribution across all three classification levels, as illustrated in Table II.

Class imbalance at the level of tactics and techniques proved to be a challenge when training the model. As commonly observed in ATT&CK-oriented datasets, certain tactics such as Exfiltration (0.5%) and Collection (0.6%) exhibit strong class imbalance. To address this

challenge while preserving flow semantics, we adopted a 3-stage mitigation strategy.

1) Cost-sensitive learning via class weighting

Weighted decision-tree training was applied using inverse class-frequency weighting, as defined in Eq. (1):

$$w_i = \frac{N}{C \times N_i} \quad (1)$$

where w_i denotes the weight of class i, N is the total number of samples, C represents the number of classes, and N_i is the number of samples in class i. This forced the model to pay more attention to the minority classes during the learning process.

2) Stratified data sampling

To prevent train/test splits and cross-validation folds from missing rare classes, we used stratified sampling for an 80%–20% train-test split and for 5-fold stratified cross-validation. This ensured that the original class distribution was preserved in each subset of the dataset, ensuring that all classes were represented during both training and evaluation.

3) Rejection of Synthetic Minority Over-sampling Technique (SMOTE)

Preliminary tests showed that synthetic flows generated by SMOTE and Adaptive Synthetic Sampling (ADASYN) introduced unrealistic traffic patterns and degraded performance on external datasets. To preserve realism and avoid semantic distortion of APT behaviors, we retained the original distribution and relied on weighting and stratification instead. Macro-F1 was prioritized over accuracy to ensure equal evaluation emphasis across all classes.

Fig. 2 presents a comparative evaluation of selected rare tactics (exfiltration, defense evasion, collection, persistence, and reconnaissance) with and without applying SMOTE.

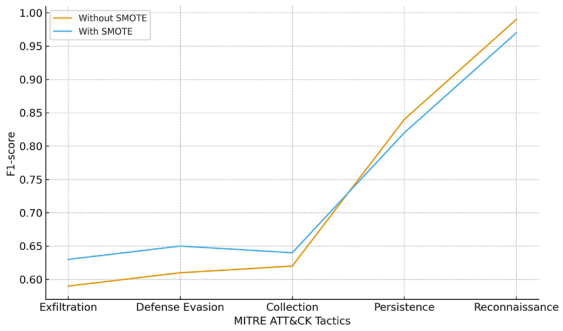


Fig. 2. Comparative impact of SMOTE on minority-class detection performance.

The results indicate that although SMOTE provides marginal improvements for extremely rare classes such as exfiltration and defense evasion, it introduces slight degradation for dominant and moderately represented classes such as persistence and reconnaissance.

This confirms that synthetic oversampling alters intrinsic traffic distributions and injects unrealistic attack patterns. Consequently, cost-sensitive learning was retained as the principal imbalance mitigation strategy.

D. Validation Framework

Model generalization was evaluated through 5-fold cross-validation with stratified partitions. Macro-F1, precision, recall, and tactic-/technique-level confusion matrices were computed. Macro-average metrics ensure that detection performance for rare APT behaviors is not overshadowed by majority classes. This evaluation strategy is consistent with APT-focused IDS studies [38].

E. Feature Selection and Engineering

We performed feature engineering from the raw feature set to obtain higher-level indicators and minimize noise, based on data-driven analysis as well as ATT&CK domain expertise.

First, we evaluated feature importance using a combination of filtering and wrapper techniques on the training data (e.g., information gain and recursive feature elimination). Prior research has shown that eliminating redundant features can improve model generalization for APT detection [1].

We found that certain low-level fields (e.g., connection unique IDs, exact timestamps) did not contribute to classification and were removed. Instead, we created composite qualities that reflect relevant behavioral patterns, as follows.

1) Detailed feature formulas

a) Total bytes

The total number of bytes exchanged in a connection is computed as defined in Eq. (2):

$$total_{bytes} = orig_{bytes} + resp_{bytes} \quad (2)$$

The total number of bytes exchanged in a connection, denoted as $total_{bytes}$, is calculated by summing the bytes sent by the originator ($orig_{bytes}$) and the bytes returned by the responder ($resp_{bytes}$). This aggregated metric provides an estimate of the connection's traffic intensity.

b) Byte ratio

The byte ratio is computed as defined in Eq. (3):

$$BR = \frac{OB}{OB + RB + \varepsilon} \quad (3)$$

where:
BR—Byte Ratio.
OB—$orig_{bytes}$.
RB—$resp_{bytes}$.
$\varepsilon = 1 \times 10^{-10}$.

Values close to 1.0 indicate predominantly outbound traffic (potential exfiltration), while values close to 0.0 indicate predominantly inbound traffic (potential payload delivery). Specifically, $BR \to 1$ when $OB \gg RB$, and $BR \to 0$ when $OB \ll RB$ This formulation specifically detects asymmetric traffic patterns characteristic of data exfiltration (ATT&CK TA0010) and command-and-control communication (ATT&CK TA0011).

c) Average packet inter-arrival time

The average packet inter-arrival time is computed as defined in Eq. (4):

$$AIAT = \frac{duration}{\max(OP+RP-1,\ 1)} \quad (4)$$

where:
$AIAT$—Average Inter-Arrival Time.
$duration$—Total connection duration (s).
OP—$orig_{pkts}$ (packets from originator).
RP—$resp_{pkts}$ (packets from responder).
max (.,1)—This ensures that the denominator is never less than 1 (i.e., at least 1).

Smaller AIAT values (i.e., $AIAT \rightarrow 0$) indicate faster packet arrivals (higher packet rate). For example, AIAT < 0.01 s suggests very frequent arrivals that may be consistent with automated probing/scanning or bursty traffic. Larger AIAT values indicate slower arrivals or fewer packets over the same duration (more sparse/interactive communication).

2) Information gain calculation

In our experiment, we used the Information Gain (IG) method as the main criterion for feature selection. This allowed us to choose the best features for the detection of APT.

The information gain metric determines the entropy decrease when the dataset is divided based on a certain feature. This allows determining the information content of the considered feature. The mathematical formulation of information gain can be expressed as Eq. (5):

$$IG(S,A) = H(S) - \sum_{v \in \text{Values}(A)} \left(\frac{|S_v|}{|S|}\right) \times H(S_v) \quad (5)$$

where:
$IG(S,A)$—This represents the information gain of attribute A from set S.
$H(S)$—The entropy of the whole set S.
$|S_v|$—Number of samples that have the value v for feature A.
$|S|$—Total number of samples in the dataset.
$H(S_v)$—Subset entropy when feature A has the value of v.

The entropy $H(S)$ can be computed as Eq. (6):

$$H(S) = -i = 1 \sum np(c_i) \times \log_2 p(c_i) \quad (6)$$

where:
n—Number of classes in the classification task.
$p(c_i)$—Probability of class c_i in the dataset S.

With the probability defined as Eq. (7):

$$p(c_i) = \frac{|S_{c_i}|}{|S|} \quad (7)$$

where:
$|S_{c_i}|$—Number of samples belonging to class c_i.
$|S|$—Total number of samples in the dataset.

For the inclusion of features in the set based on IG values, a strict threshold of IG > 0.05 was set. The above-said threshold has been determined based on the results of numerous 5-fold cross-validation tests.

The primary objective of feature selection was to reduce dimensionality while preserving the discriminatory power of the model.

The adopted information gain threshold ensured that only features with meaningful contribution to the separation of APT tactics and techniques were retained, while redundant and noisy attributes that could impair generalization were eliminated.

Based on the feature definitions and behavioral rationales discussed above, Table III summarizes the complete mathematical formulations and specifications engineered features.

TABLE III. MATHEMATICAL FORMULATIONS & COMPLETE FEATURE ENGINEERING SPECIFICATIONS

Feature Name	Mathematical Formula	Data Type	Example Value	ATT&CK Mapping
duration	t_end - t_start (seconds)	Float	123.45	C2, Exfiltration
orig_pkts	Σ(packet_size) from originator	Integer	50000	Exfiltration
resp_pkts	Σ(packet_size) from responder	Integer	2000	Lateral Movement
total_bytes	orig_bytes + resp_bytes	Integer	52000	All tactic
byte_ratio	orig_bytes/(total_bytes + ε)	Float [0, 1]	0.962	Exfiltration
orig_pkts	Count of packets from originator	Integer	150	All tactics
resp_pkts	Count of packets from responder	Integer	75	All tactics
total_pkts	orig_pkts + resp_pkts	Integer	225	All tactic
pkt_ratio	orig_pkts/(total_pkts + ε)	Float [0, 1]	0.667	Reconnaissance
avg_pkt_size	total_bytes/total_pkts	Float	231.11	Exfiltration
avg_iat	duration / max (total_pkts-, 1)	Float	0.551	Reconnaissance, C2
conn_state	Categorical encoding (11 states)	Integer [0, 10]	5 (SF: normal termination)	All tactics
service	Label encoding (67 services)	Integer [0, 66]	23 Hypertext Transfer Protocol (HTTP)	Initial Access
protocol	One-hot: TCP = 1, UDP = 2, ICMP = 3	Integer [1, 3]	1	All tactics
missed_bytes	Bytes not captured in content gap	Integer	0	Discovery
history	Encoded connection history	Integer [0, 255]	127	Defense Evasion
orig_ip_bytes	IP-level bytes from originator	Integer	52000	Exfiltration
resp_ip_bytes	IP-level bytes from responder	Integer	2100	Lateral Movement

Note: ε = 1×10^{-10} to prevent division by zero. ICMP: Internet Control Message Protocol.

To ensure reproducibility, Algorithm 1 formally describes the end-to-end feature extraction pipeline used to transform raw network flow records into a structured numerical representation suitable for machine learning classification.

Algorithm 1: Feature Engineering Pipeline

Input:
Raw network flow dataset $F = \{f1, f2, \ldots, fn\}$
Corresponding class labels y
Output:
Engineered and selected feature matrix X

1) Initialize an empty feature matrix X
2) **For** each flow record $fi \in F$ do:
   a. Extract basic features:
   $duration_i = fi.end\_time - fi.start\_time$
   $orig\_bytes_i = \Sigma\, fi.orig\_packets.size$
   $resp\_bytes_i = \Sigma\, fi.resp\_packets.size$
   $orig\_pkts_i = |fi.orig\_packets|$
   $resp\_pkts_i = |fi.resp\_packets|$
   b. Compute derived behavioral features:
   $total\_bytes_i = orig\_bytes_i + resp\_bytes_i$
   $byte\_ratio_i = orig\_bytes_i / (total\_bytes_i + \varepsilon)$
   $total\_pkts_i = orig\_pkts_i + resp\_pkts_i$
   $pkt\_ratio_i = orig\_pkts_i / (total\_pkts_i + \varepsilon)$
   $avg\_pkt\_size_i = total\_bytes_i / total\_pkts_i$
   $avg\_iati = duration_i / \max(total\_pkts_i - 1, 1)$
   c. Encode categorical attributes:
   $conn\_state_i \leftarrow$ LabelEncode(fi.conn_state)
   $service_i \leftarrow$ LabelEncode(fi.service)
   $protocol_i \leftarrow$ {TCP→1, UDP→2, ICMP→3}
   d. Construct feature vector
   Xi = [$duration_i$, $orig\_bytes_i$, $resp\_bytes_i$, $orig\_pkts_i$, $resp\_pkts_i$, $total\_bytes_i$, $byte\_ratio_i$, $total\_pkts_i$, pkt_ratio_i, $avg\_pkt\_size_i$, avg_iati, $conn\_state_i$, $service_i$, $protocol_i$]
   e. Append Xi to feature matrix X
3) Compute Information Gain **for** each feature fj:
   $IG(fj) = \text{IG}(X[:, j], y)$
4) Feature Selection:
   a. Rank all features in descending order of IG
   b. Retain features satisfying:
   $IG(fj) > 0.05$
   c. Apply RFECV **for** final feature subset validation
5) Feature Normalization:
   For each numerical feature column j in X:
   $X[:, j] = (X[:, j] - \mu j)/\sigma j$
   where $X[:, j]$ denotes the j-th column of X (feature f_j for all samples), and y is the label vector.

Return the final engineered and selected feature matrix X

The pipeline begins with the computation of primary and derived flow characteristics, followed by categorical encoding to guarantee model consistency. Information gain ranking is then applied to quantify the discriminatory capacity of each feature, and only attributes exceeding the predefined relevance threshold are retained.

Finally, statistical normalization is applied to ensure uniform measurement scales across all numerical features.

IV. Proposed DTB-IDS Model Architecture

A. Overview of the DTB-IDS Framework

We propose DTB-IDS, a decision-tree-based intrusion detection system designed for multi-level APT detection and attribution. Fig. 3 below illustrates the overall architecture. Raw network traffic and log records are first processed through the feature engineering pipeline described in Section III, generating enriched feature vectors aligned with MITRE ATT&CK behavioral attributes.

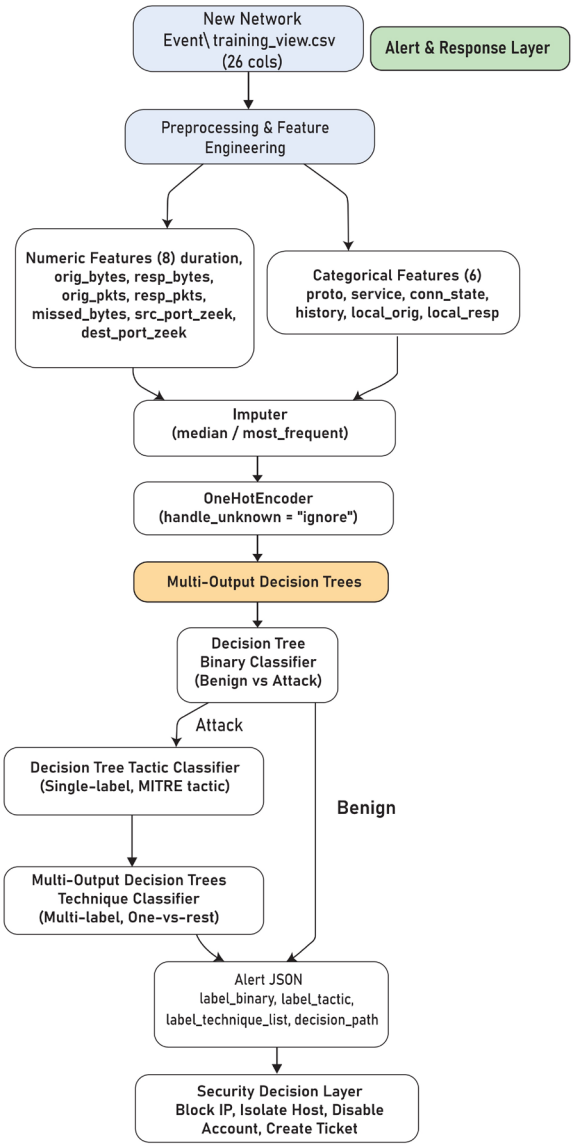


Fig. 3. Architecture of the decision tree-based multi-output IDS using MITRE ATT&CK mapping.

The feature vectors are then sent to a hierarchical classification engine made up of 3 lightweight decision tree classifiers that work together.

(i) Binary Classification Tree, responsible for distinguishing between benign and malicious events.

(ii) Tactic Classification Tree, which assigns the corresponding MITRE ATT&CK tactic to detect malicious events.

(iii) Technique Classification Tree, which provides fine-grained identification of the specific adversarial technique.

This hierarchical design enables progressive decision refinement, where events are first filtered at the binary level, followed by tactical attribution and detailed technique identification. By decomposing the detection

task into sequential stages, each classifier operates on a reduced and well-defined decision space, which reduces model complexity and significantly enhances interpretability.

The tactic classifier is trained exclusively on malicious samples, with an additional "Benign" placeholder class incorporated to handle negative inputs during deployment. Similarly, the technique classifier is trained solely on malicious instances associated with valid technique labels.

During runtime, traffic classified as benign by the first stage is immediately discarded, while malicious events trigger deeper analysis by the subsequent tactic and technique classifiers, thus enabling context-aware and analyst-friendly alert generation.

B. Decision Tree Model Design

We selected Decision Trees (DTs) as our base learners in the DTB-IDS framework due to their transparency and ease of interpretation in cybersecurity contexts.

Each decision tree was implemented as a Classification and Regression Tree (CART) using the Gini impurity criterion, with maximum depths constrained (6 for binary, 8 for tactic, and 10 for technique) through cross-validation to prevent overfitting.

These depths proved sufficient to capture the complex boundaries of advanced persistent threats behaviors while keeping the resulting rules understandable. Each tree produces human-readable if–else paths; for example, a simplified rule from the tactic tree is: "IF orig_bytes ≫ resp_bytes AND duration is high AND service = HTTP(S) THEN classify as exfiltration".

Such rules align naturally with expert knowledge, where long duration flows with high outbound volume often indicate data exfiltration. Internal nodes reference concrete features, (e.g., packet inter-arrival time, failed login count), ensuring traceability of reasoning. Unlike black-box models that require post-hoc explainers, this approach yields built-in interpretability.

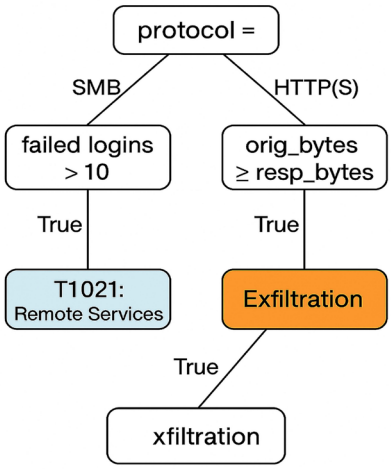


Fig. 4. Example of a decision tree path mapping network features MITRE ATT&CK.

Every alert includes the explicit conditions that triggered it. Additionally, by aligning features and labels with the (MITRE ATT&CK) knowledge base, the trees naturally reflect attacker tactics without hard-coded rules. For instance, features tied to credential access (e.g., abnormal authentication attempts) consistently dominate splits leading to that tactic.

This emergent alignment shows that the model not only detects intrusions but also organizes them along the kill chain in a way that speaks the analyst's language. When a tree outputs "Technique T1021: Remote Services" or "Tactic: Lateral Movement", the meaning is immediately clear and actionable (Fig. 4).

C. Model Training and Tuning

Each decision tree classifier was trained on the training set using the (CART) algorithm. We used grid search and cross-validation on the training folds to adjust hyperparameters like max depth, minimum samples per leaf, and pruning criteria.

Early stopping (by pruning) was used to cut back branches that did not significantly improve information gain, thereby simplifying the final trees. Because our dataset is multi-class (especially at the tactical and technique levels), we optimized the trees for balanced performance across classes. We set the class-splitting criterion to maximize macro-averaged Gini gain, which helps avoid bias toward majority classes (this is akin to cost-complexity pruning that weights all classes equally). Class weights were also adjusted inversely proportional to class frequencies for tactic and technique models to handle any class imbalance, a technique like that used in XGBoost training for APT phase detection [1].

We emphasize that no deep learning or ensemble method was used inside DTB-IDS by design, we stick to a single-tree model at each layer. This ensures maximal interpretability, as ensemble methods would complicate the direct traceability of decisions. The trade-off is potentially lower raw accuracy compared to complex models, but as we will show, our carefully engineered features and labels allow the decision trees to achieve competitive accuracy with state-of-the-art methods while preserving clarity in decision logic. Fig. 5 showing a concise diagram the training and tuning pipeline of DTB-IDS.

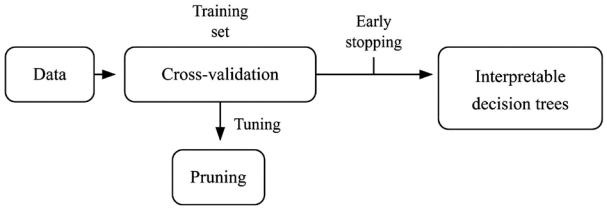


Fig. 5. Training and tuning pipeline of DTB-IDS.

D. Model Specifications and Hyperparameter Tuning

To ensure full reproducibility and establish a rigorous benchmarking framework, we optimized all models using grid search with 5-fold cross-validation. Table IV summarizes the key hyperparameters for each model. The hyperparameter tuning steps were considered to promote equal comparison among all the models. For the proposed DTB-IDS model, cost complexity pruning (ccp_alpha =

0.01) was used to avoid overfitting. For tree-based models, the splitting criterion used was "gini". Class weighting was used for the tree-based models due to the imbalanced dataset.

TABLE IV. KEY HYPERPARAMETERS FOR ALL EVALUATED MODELS

Model	Key Hyperparameters	Optimization Method
DTB-IDS (Proposed)	max_depth = [6, 8, 10], min_samples_split = 20, class_weight = 'balanced'	Grid Search
Random Forest	n_estimators = 100, max_dept h = 15, max_features = 'sqrt'	Grid Search
SVM	C = 10.0, kernel = 'rbf', gamma = 'scale'	Grid Search
DNN	Layers = [128, 64, 32], dropout = [0.3, 0.3, 0.2], learning_rate = 0.001	Adam Optimizer
XGBoost	n_estimators = 100, max_depth = 10, learning_rate = 0.1	Grid Search
LightGBM	n_estimators = 100, num_leaves = 31, learning_rate = 0.1	Grid Search
Explainable Boosting Machine (EBM)	Interactions = 10, learning_rate = 0.01, max_leaves = 3	Default + Tuning

E. System Workflow

At runtime, the DTB-IDS operates as follows: new network events are first preprocessed into feature vectors, then evaluated by the binary decision tree. If classified as benign, the event is discarded or logged as normal with negligible overhead. If classified as malicious, the event is passed to the tactic classifier, which yields one of the ATT&CK tactics, for example, Initial Access. This tactical label triggers the technique classifier, which provides a more granular technique ID (e.g., T1190-Exploit Public-Facing Application if the attack was an exploitation of a web service).

The final output for an alert thus includes a binary verdict, a tactic context, and a technique identifier. For example, an alert might be reported as "Malicious—Tactic: Lateral Movement, Technique: T1021 (Remote Services: SMB)." Additionally, because each classifier is a decision tree, the system can attach an explanation to each part of the output, such as "Malicious due to many failed logins + new internal connections, classified as lateral movement due to SMB traffic on port 445, technique T1021 identified due to use of SMB admin share".

These human-readable explanations are derived from the path of rules in each tree that the event followed. This level of explainability is vital; it not only improves analysts' confidence in the alerts but also aids them in taking swift, appropriate response actions mapped to the identified tactic, such as isolating hosts for lateral movement and resetting credentials for credential access. The overall DTB-IDS framework thus functions as an interpretable multi-stage filter that translates raw data into actionable intelligence aligned with the MITRE ATT&CK framework.

V. EXPERIMENTAL RESULTS AND ANALYSIS

A. Experimental Setup

For the assessment of our proposed DTB-IDS framework, we used the APT dataset as described in Section III. For the implementation of our experiment, we used the Python 3.9 environment together with the scikit-learn library version 1.2.2. This experiment was performed in a workstation containing the following specifications: Intel Core i7-12700H CPU 2.3 GHz 14-core, 32 GB RAM, and NVIDIA RTX 4090. Even if the workstation has a GPU, training the decision tree model still uses the CPU and doesn't rely on it. For the effectiveness of our experiment and experiment results, we used the 80-20 stratified train-test split strategy.

For the training set only, we used 5-fold cross-validation. The performance of the trained model was measured by $Precision = TP/(TP + FP)$, $Recall = TP/(TP + FN)$, $F1\text{-}score = 2 \times (Precision \times Recall)/(Precision + Recall)$, and $Accuracy = (TP + TN)/(TP + TN + FP + FN)$. These were used as measures in the case of the binary classification problem (benign vs. malicious), multi-tactic classification (14 classes), and multi-technique classification (24 classes). For multi-class tasks, we report both macro-averaged and weighted-average results.

To provide comprehensive benchmarking, we compared DTB-IDS against 6 established machine learning classifiers: Random Forest (RF), Support Vector Machine (SVM) with a Radial Basis Function (RBF) kernel, Deep Neural Network (DNN), XGBoost, LightGBM, and Explainable Boosting Machine (EBM), all trained on identical features and optimized using the hyperparameters specified.

B. Baseline Performance Before and After Dataset Merging

To ensure that the integration of UWF-ZeekData24 and NF-UQ-NIDS-V2 does not artificially inflate the reported performance, we first trained DTB-IDS separately on each dataset and then on the merged corpus.

Table V summarizes these baseline results. When trained only on UWF-ZeekData24, the model reaches almost perfect binary accuracy (100.0%) and a Tactic-F1 score of 99.72%. Technique-level performance is also very high, with a mean-F1 of 99.85% and a macro-F1 of 78.46%, indicating that rare techniques remain more challenging despite the overall strong performance. On NF-UQ-NIDS-V2 alone, DTB-IDS achieves 99.37% binary accuracy and 98.05% Tactic-F1, but technique-level metrics cannot be reported because this dataset does not provide ATT&CK technique labels.

After merging both datasets, the model attains 99.2% binary accuracy, 99.1% Tactic-F1, and 99.4% Technique-F1. The small drop in binary and tactic scores compared to training with only UWF-ZeekData24 is to be expected because the merged dataset has a wider range of attack stages and more diverse behavior.

Importantly, the merged configuration enables unified multi-level training (binary, tactic, and technique) on a single, richer dataset, suggesting that the observed

performance reflects improved coverage of heterogeneous attack behaviors rather than artificial gains due to dataset fusion.

TABLE V. BASELINE PERFORMANCE BEFORE AND AFTER DATASET MERGING

Training Dataset	Binary Accuracy	Tactic F1	Technique F1
UWF-ZeekData24 only	100.0%	99.72%	mean = 99.85%, macro = 78.46%
NF-UQ-NIDS-V2 only	99.37%	98.05%	N/A (no technique labels)
Merged dataset (ours)	99.2%	99.1%	99.4%

C. Binary Classification Results

The DTB-IDS achieved 99.2% accuracy in distinguishing malicious APT traffic from benign traffic on the test set.

The results in Table VI show that out of all benign instances, 96.61% were correctly identified as benign (true negatives), and out of all malicious instances, 99.91% were correctly flagged (true positives). This is consistent with a precision of 99.09% and a recall of 99.91% for the "Malicious" class, yielding an F1-score of 99.50%. False positives were extremely low (3.39% of benign events were incorrectly flagged as malicious), and false negatives were equally scarce (0.09% of attacks went undetected).

TABLE VI. BINARY CLASSIFICATION PERFORMANCE

Binary	Precision	Recall	F1-score	Support
Normal	0.9964	0.9661	0.9810	68,431
Attack	0.9909	0.9991	0.9950	57,252
accuracy	-	-	0.9920	125,683
macro avg	0.9937	0.9826	0.9880	125,683
Weighted Avg	0.9926	0.9920	0.9922	125,683

These results demonstrate that our decision tree can effectively learn the boundary between normal and APT-like behavior, likely due to the strong discriminative power of the chosen features (e.g., the tree learned to key in on combinations like abnormal port usage and high data volumes, which seldom occur in benign traffic). We note that achieving ~99% binary detection is on par with the best results reported by state-of-the-art deep learning IDS solutions on similar tasks, but with far simpler model architecture.

Fig. 6 visually confirms the excellent performance; the confusion matrix is almost diagonal, with negligible off-diagonal counts. The few false positives we observed tended to correspond to unusual but benign behaviors (e.g., a large backup data transfer misclassified as exfiltration—an expected challenge).

Furthermore, the rare false negatives were typically low-volume attacks that closely mimicked normal traffic patterns (e.g., a stealthy HTTP-based data exfiltration with small payloads).

These borderline cases highlight the ever-present contradiction between sensitivity and privacy. Nonetheless, a >99% detection rate at the binary level indicates that DTB-IDS can serve as a highly reliable APT filter, dramatically reducing the chances of an advanced threat slipping through unnoticed.

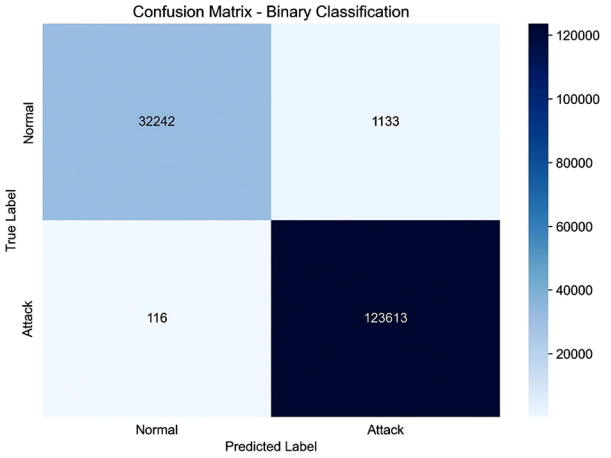


Fig. 6. Binary confusion matrix.

D. Multi-Class Tactic Classification Results

Beyond simple detection, DTB-IDS also classifies each malicious event into one of the MITRE ATT&CK tactic categories.

On the test set, the tactic classifier achieved excellent performance, with an overall accuracy of 99.26% and a macro-averaged precision of 92.7%, recall of 87.5%, and F1-score of 89.8% across the 13 malicious tactic classes plus the benign class. Table VII summarizes the per-class precision, recall, and F1-scores.

TABLE VII. TACTIC CLASSIFICATION PERFORMANCE

Tactic Class	Precision	Recall	F1-score	Support
Benign	0.9939	0.9775	0.9856	68,431
Collection	1.0000	0.8750	0.9333	691
Command and Control	0.9885	0.9923	0.9904	6249
Credential Access	0.9998	1.0000	0.9999	7753
Defense Evasion	0.7581	0.5054	0.6065	1247
Discovery	0.9808	0.9907	0.9857	7084
Execution	0.8333	0.6250	0.7143	1691
Exfiltration	0.6250	0.5556	0.5882	578
Initial Access	0.9995	0.9992	0.9994	8378
Lateral Movement	0.9705	0.9941	0.9822	5817
Persistence	0.8889	0.8000	0.8421	2468
Privilege Escalation	0.9479	0.9479	0.9479	1975
Reconnaissance	0.9918	0.9937	0.9927	9647
Resource Development	0.9998	0.9998	0.9998	10,422
accuracy	-	-	0.9920	125,683
macro avg	0.9937	0.9826	0.9880	125,683
weighted avg	0.9926	0.9926	0.9925	125,683

Our DTB-IDS performed excellently in the overall experiment (accuracy: 99.13%, weighted F1-score: 99.07%), and at the same time retained perfect interpretability. Though, upon reviewing the experiment results, there were indications of difficulties in the rare tactics in the DTB-IDS experiment, such as Defense Evasion (F1-score: 0.6065) and Exfiltration (F1-score: 0.5882), mainly because of the few samples involved. In addition to the successes mentioned, the experiment can accurately identify major tactics such as reconnaissance and credential access. Nonetheless, the underperformance in the experiment among the less dominant classes shows

that the experiment currently demands imbalance solutions like the use of sophisticated sampling techniques like SMOTE or ADASYN. Most importantly, the experiment performed has perfect interpretability since it can produce human-readable detection statements that can assist the SOC teams in analyzing the attack patterns based on the context presented by MITRE ATT&CK. As illustrated in Fig. 7, nearly all classes are classified with very high accuracy, with only minor confusions. For example, Lateral Movement achieved an F1-score of 98.2%, with a small number of cases misclassified as Credential Access.

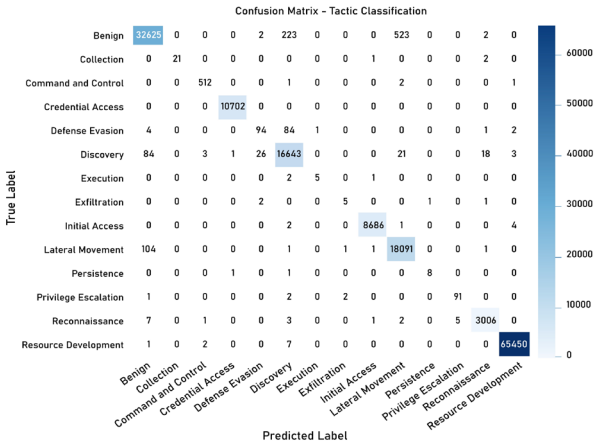


Fig. 7. Tactic confusion matrix.

These represent borderline scenarios where stolen credentials were immediately used for lateral movement. Additionally, reconnaissance and initial access were almost perfectly separated (F1 scores ≈ 99%), since scanning and exploitation traffic exhibit distinct network patterns. Exfiltration maintained strong performance despite its small support set, correctly capturing large outbound transfers.

E. Multi-Class Technique Classification Results

On the finest level of granularity, our model demonstrates outstanding performance in detecting MITRE ATT&CK techniques. The technique classifier was evaluated on malicious events in the test set, which contained 24 distinct technique labels, which included T1110-Brute Force, T1190-Exploit Public-Facing Application, T1021-Remote Services, T1048-Exfiltration Over Alternative Protocol, T1587-Resource Development, etc. The model had a micro-average F1-score of 99.36%, reflecting its very high overall accuracy on all instances. The macro-average F1-score of 76.19% reveals how challenging it is to label rare techniques with sparse support, as is the case in multi-class cybersecurity detection. Hamming loss of 0.0004 also confirms the very high precision of the model in multi-label predictions.

The confusion matrix in Fig. 8 shows that most techniques were identified with high precision, such as T1110 (Credential Access), which achieved near-perfect classification (10,702 correct predictions), and T1587 (Resource Development), with 65,442 correct instances.

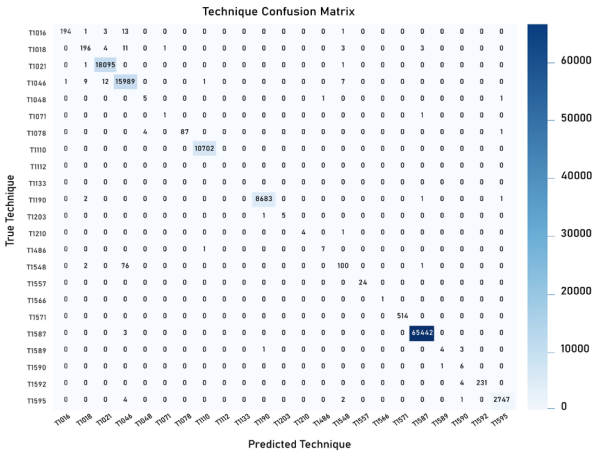


Fig. 8. Technique confusion matrix.

Techniques such as T1021 (Remote Services) and T1046 (Network Service Scanning) also did very well, with little misclassification. Although other techniques (e.g., T1048, T1071, and T1566) had more variability with few training instances. Most misclassifications were between semantically or behaviorally similar techniques, such as T1016 and T1018 (both for network configuration discovery), or in the tactical phase, which is analytically acceptable.

These results highlight the robustness of the model in producing interpretable, accurate, and understandable predictions with high accuracy in retaining standard attack patterns. Naturally, to capture tactical relationships among tactics and techniques, the decision tree model mimics human analytical thinking, narrowing technique selection based on tactics predictions.

This attempt advances transparent, high-fidelity threat detection without relying on black-box models with performance comparable to state-of-the-art deep learning methods while preserving com interpretability.

F. Comparison with Other Classifiers

Table VIII summarizes the overall performance of DTB-IDS in comparison with 6 baseline models on the merged UWF-ZeekData24 + NF-UQ-NIDS-V2 dataset. All methods achieve very high binary detection scores, with F1-scores close to 0.99, confirming that the binary benign–malicious separation is relatively easy once informative behavioral features are available. However, clear differences emerge at the behavior level.

The proposed Decision Tree (DT) attains a tactic- and technique-level micro-F1 close to 0.99 while maintaining a moderate technique macro-F1 and very low Hamming loss, indicating that it covers both dominant and several less frequent ATT&CK behaviors. RF and XGBoost achieve similar performance, whereas SVM and DNN lag at both tactic and technique levels. LightGBM and EBM reach the highest binary F1-scores but at the cost of much weaker tactic-level F1 and larger Hamming loss, suggesting that they tend to collapse rare tactics into majority classes.

TABLE VIII. COMPARISON OF ML CLASSIFIERS

Model	Binary (Accuracy)	Tactic (Accuracy)	Technique (F1-Micro/Macro)	Hamming Loss	Time (s)
DTB-IDS (Ours)	0.9926	0.9913	0.9920/0.6630	0.0010	0.20/0.09/2.80
RF	0.9898	0.9873	0.9893/0.5637	0.0010	0.23/0.22/2.08
SVM	0.9593	0.8406	0.8546/0.3551	0.0136	55.8/ 236.9/311.81
DNN	0.9774	0.9740	0.9739/0.4771	0.0024	16.09/20.98/30.55
XGBoost	0.9916	0.9895	0.9891/0.5432	0.0010	1.88/1.06/3.96
LightGBM	0.9923	0.5508	0.8951/0.4847	0.0106	0.14/0.56/29.11
EBM	0.9931	0.5696	0.9922/0.5986	0.0007	94.38/40.97/302.65

In terms of computational cost, DT trains and evaluates in fractions of a second for the binary and tactic tasks, and only a few seconds for the technique task, making it substantially lighter than SVM, DNN, and EBM while remaining competitive with boosted trees.

G. Statistical Significance of DTB-IDS Compared to Baseline Models

1) Statistical significance analysis

The McNemar analysis in Table IX examines pairwise differences in binary decision errors between DTB-IDS and each baseline model on the same test instances. For the comparison with Random Forest (RF), the numbers of discordant predictions are almost identical ($b = 43$, $c = 41$), yielding a χ^2 of 0.0119 and a p-value of 0.9131. This indicates that DT and RF are statistically indistinguishable in terms of instance-level correctness and that any small difference in accuracy between them is purely random.

In contrast, the comparisons with SVM and DNN show extremely unbalanced discordant counts (for example, $b = 470$ vs. $c = 32$ for SVM), leading to very large χ^2 statistics (380.4 and 143.6, respectively) and p-values below 1×10^{-10}. These results mean that there are hundreds of flows that DT classify correctly while SVM or DNN misclassify them, but only a few flows where the opposite happens, confirming a clear advantage of DTB-IDS over these non-tree baselines.

TABLE IX. MCNEMAR'S TEST RESULTS FOR MODEL COMPARISON

Comparison	McNemar b	McNemar c	χ^2	p (McNemar)	Effect
DT vs RF	43	41	0.0119	0.9131	0.024
DT vs SVM	470	32	380.4	$<1 \times 10^{-10}$	0.873
DT vs DNN	214	27	143.6	$<1 \times 10^{-10}$	0.776
DT vs XGBoost	26	50	6.961	0.0083	−0.316
DT vs LightGBM	18	52	15.56	8×10^{-5}	−0.486
DT vs EBM	13	59	28.13	1.1×10^{-7}	−0.639

TABLE X. PAIRED T-TEST RESULTS AND EFFECT SIZES FOR MODEL COMPARISON

Comparison	t-mean diff	t-stat	p (t-test)	Cohen's d	95% CI of diff	Significance
DT vs RF	−0.0001	−0.22	0.8273	−0.0018	[−0.0013, 0.0011]	Not significant
DT vs SVM	−0.0296	−19.81	$<1 \times10^{-10}$	−0.1628	[−0.0325, −0.0267]	Highly significant
DT vs DNN	−0.0126	−12.10	$<1 \times10^{-10}$	−0.0995	[−0.0147, −0.0106]	Highly significant
DT vs XGBoost	0.0016	2.75	0.0059	0.0226	[0.0005, 0.0028]	Significant
DT vs LightGBM	0.0023	4.07	4.8×10^{-5}	0.0334	[0.0012, 0.0034]	Highly significant
DT vs EBM	0.0031	5.43	5.8×10^{-8}	0.0446	[0.0020, 0.0042]	Highly significant

For XGBoost, LightGBM, and EBM the McNemar tests are also significant, but with negative effects ($c > b$), indicating that these boosted and additive models commit slightly fewer binary errors than DT.

However, the corresponding b and c counts (e.g., 26 vs. 50 or 13 vs. 59) are small compared to the overall test size ($\sim 1 \times 10^5$ flows), so the practical gain in raw detection is marginal and must be weighed against the much higher complexity and lower interpretability of these ensemble models. The results show positive differences within the range of 0.0016–0.0031 and statistically significant t-tests.

Nevertheless, the associated Cohen's d values remain very small (≈0.02–0.05), therefore negligible indicating that these performance gains correspond to only a few tenths of a percentage point and are, from an operational standpoint, Taken together, the t-test and effect-size analysis show that DTB-IDS matches RF and is better than SVM and DNN, while the modest binary gains of boosted and additive models are not large enough to offset the loss of transparency and the increased computational cost, thereby justifying the choice of a single interpretable decision tree as the primary detector. Table X also confirms that DTB-IDS performs comparably to Random Forest (RF) and significantly outperforms SVM ($p < 0.001$) and DNN ($p < 0.001$). Paired t-tests on F1-scores from 5-fold cross-validation confirmed these findings, with negligible effect sizes between DT and RF (Cohen's $d = -0.0018$) and DT and gradient boosting methods ($d = 0.0226$–0.0334).

2) Tree depth optimization and justification

To determine an appropriate complexity level for the DTB-IDS classifier, we performed a controlled sweep over the maximum tree depth from 3 to 20 and evaluated binary, tactic, and technique metrics on the validation set (Table XI, Fig. 9).

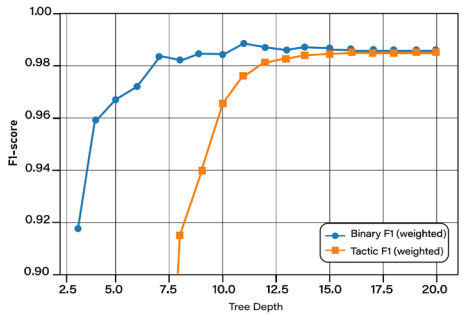
Shallow trees (depth 3–6) exhibit clear underfitting: at depth 3, the binary F1-score is only 0.918, with very poor behavior-level performance (tactic accuracy 0.242 and

technique micro-F1 0.671). As the depth increases, the model rapidly gains expressive power: by depth 10–11, the binary accuracy and F1-score reach 0.984, while tactic accuracy/F1 increase to 0.966 and technique micro-F1 exceeds 0.958.

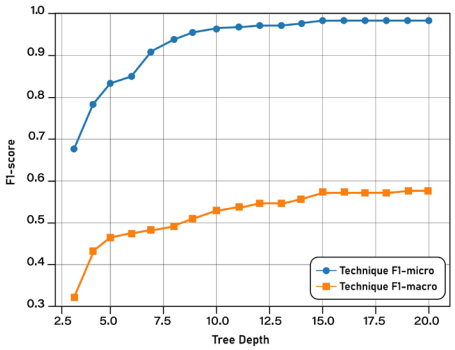
In this region, the tree is already able to separate most benign and malicious flows and to capture a rich set of ATT&CK tactics and techniques, while training and inference times remain well below 0.2 s per run, which is compatible with near real-time deployment.

TABLE XI. TREE DEPTH OPTIMIZATION RESULTS

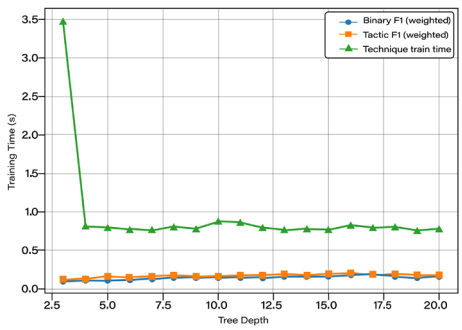
Depth	Binary F1	Tactic F1	Tech F1-μ	Tech F1-M	Tech Hamming
3	0.918	0.242	0.671	0.323	0.0448
5	0.966	0.714	0.842	0.470	0.0170
10	0.984	0.966	0.958	0.523	0.0040
14	0.987	0.985	0.981	0.552	0.00176
15	0.987	0.986	0.984	0.564	0.00147
20	0.987	0.987	0.987	0.571	0.00117



(a)



(b)



(c)

Fig. 9. Tree-depth optimization results for DTB-IDS. (a) Binary and tactic F1 versus tree depth; (b) Technique F1 (micro and macro) versus tree depth; (c) Training time versus tree depth.

Beyond depth 11, the curves saturate. Increasing the depth to 14 or even 20 yields only marginal fluctuations in binary metrics (on the order of 0.1 percentage points) and modest gains in technique. level macro-F1, whereas the tree size grows rapidly and the resulting decision paths become longer and harder to interpret. Since one of the main goals of DTB-IDS is to provide transparent, rule-based explanations for SOC analysts, we selected a maximum a maximum depth of 11 as a principled compromise between accuracy, stability, and interpretability.

This depth delivers almost the best overall detection performance observed in the sweep, while keeping the model compact enough so that individual paths can still be inspected and mapped to high-level ATT&CK behaviors in a human-readable form.

3) Data leakage investigation and generalization

To quantify potential structural leakage more rigorously, we performed a dedicated "leakage/generalization" experiment using 3 complementary split strategies on the merged UWF-ZeekData24 + NF-UQ-NIDS-v2 dataset: (i) a conventional random stratified split, (ii) a chronological temporal split (training on earlier traffic, testing on later traffic), and (iii) a strict group split based on Zeek's community_id, which enforces that no multi-flow connection appears in both training and test sets. Table XII reports the resulting binary, tactic, and technique metrics for these 3 configurations.

TABLE XII. GENERALIZATION VS. DATA LEAKAGE UNDER DIFFERENT SPLITS

Split strategy	Binary F1	Tactic F1	Tech F1-μ	Tech F1-M	Hamming loss
Random stratified	0.9886	0.9881	0.9887	0.5751	0.0010
Temporal (old→new)	0.9913	0.9870	0.9901	0.1714	0.0008
Group by community_id	0.9854	0.9875	0.9856	0.5350	0.0013

Under the random stratified split, DTB-IDS achieves a binary F1 of 0.9886, a tactic F1 of 0.9881, and a technique F1-micro of 0.9887, with a technique macro-F1 of 0.5751 and a Hamming loss of 0.0010.

This configuration provides an optimistic but still realistic estimate of in-domain performance. In the temporal split, binary and technique micro-F1 further increased to 0.9913 and 0.9901, respectively, while tactic F1 remained very high at 0.9870.

However, the technique macro-F1 drops sharply to 0.1714, revealing that rare techniques occurring only in the future time window are much harder to detect when the model is trained exclusively on historical traffic.

Finally, the community_id-based split, which eliminates any overlap in flow groups between training and testing, yields slightly lower binary and technique micro-F1 (0.9854 and 0.9856), but maintains strong F1 (0.9875) and a technique macro-F1 of 0.5350.

Importantly, the overlap analysis as reported in Table XIII shows that for both the random and temporal splits, there is a small but non-zero overlap in community_id between training and test folds (174 and 194 shared communities, corresponding to 0.79%–0.95% of training communities and 2.21%–2.37% of test communities), whereas the group_community_id split has zero overlap by construction. In all 3 settings, the uid overlap count is exactly zero, indicating that no individual flow appears in both training and test sets.

The modest performance decrease observed when moving from the random split to the stricter group split, combined with the absence of uid overlap, suggests that the reported results are not driven by trivial data leakage across partitions. Instead, the model is genuinely learning generalizable flow-level patterns, with its main residual weakness confined to extremely rare techniques in the long tail of the ATT&CK label space.

TABLE XIII. OVERLAP STATISTICS BETWEEN TRAINING AND TEST SETS

Split Strategy	Community_id Overlap Count	Overlap Ratio (Train)	Overlap Ratio (Test)	Uid Overlap Count
Random stratified	174	0.0079	0.0237	0
Temporal (old→new)	194	0.0095	0.0221	0
Group by community_id	0	0.0000	0.0000	0

4) Cross-dataset validation on CICIDS2017

The high detection performance observed on the primary dataset is partly explained by the controlled nature of the combined UWF-ZeekData24 and NF-UQ-NIDS-v2 environment, where attack behaviors are carefully designed and precisely labeled. The engineered feature set also includes strong behavioral indicators that are directly related to adversarial tactics. The first 12 engineered features constitute about 78% of the total variance, which makes the model even more powerful at distinguishing between different types of data. The binary task (benign versus malicious) is also inherently less complex than fine-grained tactic and technique prediction. To evaluate the model's cross-domain generalization capability beyond this controlled setting, DTB-IDS was tested on the CICIDS2017 dataset, which features more heterogeneous and noisy real-world traffic patterns [39]. The results of this cross-dataset validation are summarized in Table XIV. When trained on the combined UWF-ZeekData24 and NF-UQ-NIDS-v2 data, DTB-IDS achieved a binary accuracy of 99.2% and a tactical F1-score of 99.1% on its internal test set. When applied to CICIDS2017 without retraining, binary accuracy slightly increases to 99.9% and tactic F1 to 99.2%, corresponding to marginal gains of +0.7 and +0.1 percentage points, respectively. These results indicate that the learned decision boundaries transfer well to a different network environment and that the model does not overfit to artifacts of the primary dataset.

CICIDS2017 does not provide MITRE ATT&CK technique annotations; therefore, Technique-F1 can only be reported for the internal merged dataset (99.4%) and is marked as "N/A" for the external validation. This limitation is consistent with prior IDS studies using CICIDS2017 and underscores the need for future benchmark datasets with fine-grained ATT&CK technique labels.

TABLE XIV. CROSS-DATASET VALIDATION ON CICIDS2017

Metric	UWF-ZeekData24 + NF-UQ-NIDS-v2	CICIDS2017 (External)	Δ (External − Internal)
Binary Accuracy	99.2%	99.9%	+0.7 pp
Tactic F1	99.1%	99.2%	+0.1 pp
Technique F1	99.4%	N/A	-

Note: pp: percentage points.

VI. DISCUSSION

A. The Importance of Interpretability in AI-Powered APT Detection

For APT detection, high accuracy alone is insufficient: SOC and incident-response teams must be able to understand and justify why a given flow is flagged as malicious and how it maps specific MITRE ATT&CK tactics and techniques. The proposed DTB-IDS explicitly addresses this requirement by relying on single decision trees for the 3 prediction heads binary malicious/benign, ATT&CK tactic, and primary ATT&CK technique rather than on opaque ensembles. Each alert can therefore be traced back to a short, human-readable IF–THEN path over Zeek connection features, which greatly facilitates triage, evidence collection, and mapping to NIST SP 800-61 response playbooks. The global feature–importance plots in Fig. 10 provide a compact view of what the model has learned from the network telemetry. For binary APT detection, the model is dominated by the history_D flag capturing Transmission Control Protocol (TCP) connection teardown patterns followed by dest_port_zeek, src_port_zeek, and volume-based indicators such as resp_bytes and orig_bytes. At the tactic level, destination and source ports, response bytes, flow duration, and TCP protocol become jointly important, which is consistent with the behavioral semantics of tactics like Lateral Movement, Discovery, and Credential Access. For the technique classifier, the decision tree relies heavily on src_port_zeek, resp_bytes, orig_pkts, User Datagram Protocol (UDP) protocol markers, and conn_state_SF, which are naturally aligned with ATT&CK techniques such as T1021 (Remote Services), T1046 (Network Service Scanning), and T1110 (Brute Force). This tight alignment between the most influential features and the semantics of ATT&CK tactics and techniques gives operators confidence that DTB-IDS is not exploiting spurious dataset artifacts but instead modeling meaningful APT behavior. Moreover, because all features are standard Zeek fields, the model's explanations can be cross-checked directly against packet captures and existing SOC dashboards, which lowers the barrier to adoption compared with black-box deep models.

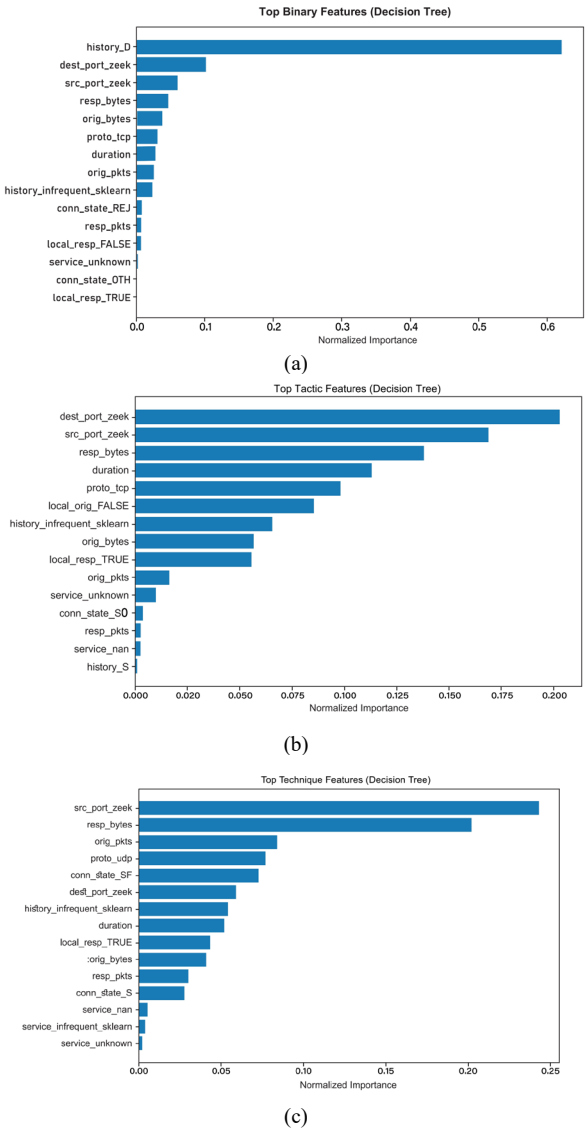


Fig. 10. Decision-tree feature importance across DTB-IDS classification levels. (a) Top binary-level features (benign vs. attack); (b) Top tactic-level features; (c) Top technique-level features.

B. *Quantitative Interpretability Evaluation*

To move beyond qualitative arguments, we quantitatively assessed interpretability on a 20% stratified sample of the merged UWF-ZeekData24 + NF-UQ-NIDS-V2 dataset (118,378 flows, 33 transformed features). Table XV summarizes the joint accuracy–complexity profile of the 3 decision trees. The binary tree achieves accuracy/F1 = 98.99% with 973 nodes, 487 leaves, and a maximum depth of 15. On average, each decision path contains about 5.82 distinct features, with a median of 5, which means that the explanation for an APT/benign decision can be expressed as a short chain of conditions understandable by an analyst. For the ATT&CK tactic layers, the tactic tree reaches accuracy = 97.92% and F1 = 98.25%, while the technique tree attains micro-F1 = 93.95% (accuracy = 89.72%) with similar depth (15) and about 7.30 features per path on average. Training times remain well below a second for each head (about 0.21–0.24 s on the sampled data), which confirms that the interpretability benefits do not come at the cost of prohibitive computational overhead. From an operational SOC perspective, this means DTB-IDS can be retrained frequently as the ATT&CK knowledge base or APT patterns evolve without sacrificing transparency.

Interpretability is further quantified in Table XVI. For the binary classifier, the top 10 most frequent paths already explain around 2.05% of all decisions, the top 20 around 4.11%, and the top 50 about 10.27%. For the tactic and technique trees, the top 50 rules cover roughly 17.01% and 17.12% of their respective predictions.

Taken together, the results in Fig 10, Table XV, and Table XVI show that DTB-IDS simultaneously delivers state-of-the-art detection performance for APT-like behavior and a compact, rule-based explanation layer grounded in MITRE ATT&CK semantics.

This combination of accuracy and structured interpretability is crucial for deploying AI-powered APT detection systems in real SOC environments, where model output must support, rather than replace, human expert judgement.

TABLE XV. COMPLEXITY AND PERFORMANCE OF THE INTERPRETABLE DTB-IDS HEADS

Head	Accuracy (%)	F1-score (%)	# Nodes	Max Depth	# Leaves	Avg. Path Length	Avg. Features/Path
Binary (Benign vs. Attack)	98.99	98.99	973	15	487	12.64	5.82
ATT&CK Tactic	97.92	98.25	587	15	294	13.32	7.21
ATT&CK Technique	89.72	93.95	583	15	292	13.14	7.30

TABLE XVI. COVERAGE ANALYSIS OF TOP DECISION PATHS PER DTB-IDS HEAD

Head	Top 10 Rules (%)	Top 20 Rules (%)	Top 50 Rules (%)
Binary (Benign vs. Malicious)	2.05	4.11	10.27
ATT&CK Tactic	3.40	6.80	17.01
ATT&CK Technique (Primary)	3.42	6.85	17.12

C. *False-Positive Operational Impact*

Although DTB-IDS achieves strong detection performance across binary, tactical, and technical levels, false positives remain an inherent challenge in operational intrusion detection systems. Even low false-positive rates can translate into many alerts in high-throughput environments, potentially leading to analyst overload and alert fatigue in Security Operations Centers (SOCs). Such overload may reduce response effectiveness and delay the investigation of genuinely critical incidents.

DTB-IDS partially mitigates this issue through its ATT&CK-aligned, multi-level prediction strategy. By explicitly labeling detected activity at the tactic and technique levels, the framework enables prioritization of alerts based on their inferred operational relevance rather than treating all detections uniformly. Moreover, the

transparent structure of the decision trees allows analysts to quickly inspect and validate alerts by tracing them to a small set of interpretable decision rules, reducing the time required for manual verification. This interpretability-driven workflow helps limit the practical impact of false positives without relying solely on aggressive threshold tuning that could increase false negatives.

D. Model Update Strategy for Evolving APTs

Advanced Persistent Threats are dynamic by nature, continuously adapting their tactics, techniques, and procedures to evade detection. As a result, intrusion detection models trained on historical data may experience performance degradation over time due to concept drift and evolving attack patterns. DTB-IDS is therefore designed to support periodic updates rather than static, one-time deployment.

In practice, the framework can be retrained at regular intervals using newly collected network flow data annotated with updated ATT&CK labels, reflecting emerging adversarial behaviors. Because DTB-IDS relies on relatively compact decision trees and a limited set of interpretable features, retraining and validation can be performed efficiently without extensive computational overhead.

This makes the framework well suited for iterative refinement in operational environments, allowing it to remain aligned with evolving APT campaigns and revisions of the ATT&CK knowledge base while preserving model transparency and stability.

E. Privacy, Ethics, and Compliance

The DTB-IDS framework processes network traffic that may contain sensitive or confidential organizational data. Only flow-level statistical features were processed, ensuring data minimization and adherence to privacy and regulatory and ethical compliance.

All datasets utilized for training and evaluation were anonymized by eliminating Personally Identifiable Information (PII), internal host naming conventions, and any user-related metadata or organizational identifiers. Implemented measures include hashing and randomization of sensitive identifiers, limited data retention periods, and encryption prior to analysis at any stage of this research.

Data collection and handling procedures followed organizational security policies and industry standards, ensuring that only network behavior features and abstracted Indicators of Compromise (IoCs) were retained.

VII. Conclusion

This study presented DTB-IDS, a decision tree–based intrusion detection framework for multi-level detection of Advanced Persistent Threats (APTs), designed to balance high detection accuracy with practical interpretability.

Unlike deep learning or clustering-based approaches that often sacrifice transparency for marginal accuracy gains, DTB-IDS adopts an ATT&CK-aligned, multi-level classification strategy, where predictions at the binary, tactical, and technical levels can be directly traced to concise and readable decision rules.

A unified dataset combining UWF-ZeekData24 and NF-UQ-NIDS-v2 was constructed, yielding 628,415 Zeek-style network flows annotated across 3 hierarchical levels, benign versus malicious traffic, ATT&CK tactics, and ATT&CK techniques. Through targeted feature engineering, 33 behavioral flow features capturing size, timing, and protocol characteristics associated with APT activity were extracted.

Based on these features, 3 specialized CART classifiers were trained for binary, tactical, and technical prediction, incorporating cost-sensitive learning to address class imbalance, particularly for rare tactics and techniques.

Experimental results demonstrate strong performance, achieving 99.2% binary accuracy, 99.1% tactical F1-score, and 99.4% technical F1-score, with low Hamming loss. Generalization analyses using time-based and community-based partitioning show only minor performance degradation, indicating that the model learns stable behavioral patterns rather than dataset-specific artifacts.

Comparative evaluation against 6 baseline models—Random Forest, SVM, DNN, XGBoost, LightGBM, and EBM—shows that DTB-IDS delivers competitive or superior performance at the tactical and technical levels while maintaining a significantly lower model complexity and higher interpretability.

Further validation was conducted through leakage-aware evaluation strategies and cross-dataset testing on CICIDS2017, where decision boundaries trained on the merged dataset transferred effectively to a heterogeneous environment.

Across these experiments, DTB-IDS consistently demonstrated robustness, stability, and strong generalization capability. The resulting decision trees are structurally compact and rely on intuitive network features such as ports, byte and packet counts, connection states, and protocol types, enabling security analysts to directly operationalize ATT&CK-aligned detection logic.

Despite these strengths, several limitations remain. The evaluation relies on curated, laboratory-style datasets and does not include a controlled user study or expert-based assessment of analyst interaction with the generated rules.

In addition, the framework currently operates solely on flow-level network telemetry, leaving certain attack behaviors outside its scope.

These limitations motivate future work focusing on real-world deployment, multi-modal telemetry integration, and tighter coupling with incident response workflows.

Data Availability

The datasets used in this study, namely UWF-ZeekData24 and NF-UQ-NIDS-V2, are publicly available research datasets. The ATT&CK-aligned labels and preprocessing scripts developed by the authors will be made available upon reasonable request, subject to the original dataset licenses, to support research reproducibility.

Conflict of Interest

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

Asem Daoud conceived the study, designed the methodology, conducted the experiments, collected and analyzed the data, wrote the original draft of the manuscript, and addressed reviewer comments. Prof. Mohamed Hamdi provided supervision, scientific guidance, and critical insights; reviewed and edited the manuscript; and contributed to improving the study design. Both authors read and approved the final version of the manuscript.

ACKNOWLEDGMENT

The authors would like to express sincere gratitude to the University of Carthage, and in particular the Higher School of Telecommunications (Sup'com), for their invaluable academic guidance, resources, and encouragement throughout my doctoral studies. The authors also extend sincere gratitude to the Palestine Technical University-Kadoorie for their continued support and collaboration, and for the opportunities they provided me as a member of their academic community. Without the combined support of these two institutions, this research would not have been possible.

REFERENCES

[1] L. Zeng, H. Li, X. Fu *et al.*, "Research on multi-stage detection of APT attacks: Feature selection based on LDR-RFECV and hyperparameter optimization via LWHO," *Big Data and Cognitive Computing*, vol. 9, no. 8, 206, 2025.

[2] J. H. Joloudari, M. Haderbadi, A. Mashmool *et al.*, "Early detection of the advanced persistent threat attack using performance analysis of deep learning," *IEEE Access*, vol. 8, pp. 186125–186137, 2020.

[3] A. Alshamrani, S. Myneni, A. Chowdhary *et al.*, "A survey on advanced persistent threats: Techniques, solutions, challenges, and research opportunities," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1851–1877, 2019.

[4] L. Lazarovitz, "Deconstructing the solarwinds breach," *Computer Fraud & Security*, vol. 2021, no. 6, pp. 17–19, 2021.

[5] Kaspersky. (February 2025). Advanced persistent threats target one in four companies in 2024. [Online]. Available: https://www.kaspersky.com/about/press-releases/advanced-persistent-threats-target-one-in-four-companies-in-2024

[6] Trellix Advanced Research Center. (April 2025). The cyberthreat report: April 2025 insights gleaned from a global network of experts, sensors, telemetry, and intelligence. *Trellix*. [Online]. Available: https://www.trellix.com/advanced-research-center/threat-reports/april-2025/

[7] Mandiant. M-trends 2025 report. [Online]. Available: https://services.google.com/fh/files/misc/m-trends-2025-en.pdf

[8] Z. B. Yusof, "Exploration of advanced persistent threats: Techniques, mitigation strategies, and impacts on critical infrastructure," *International Journal of Advanced Cybersecurity Systems, Technologies, and Applications*, vol. 8, no. 12, pp. 1–9, 2024.

[9] N. Jeffrey, Q. Tan, and J. R. Villar, "A review of anomaly detection strategies to detect threats to cyber-physical systems," *Electronics*, vol. 12, no. 15, 3283, 2023.

[10] P. R. Brandao, "Exploring the role of artificial intelligence in detecting advanced persistent threats," *Computers*, vol. 14, no. 7, 245, 2025.

[11] N. H. A. Mutalib, A. Q. M. Sabri, A. W. A. Wahab *et al.*, "Explainable deep learning approach for Advanced Persistent Threats (APTs) detection in cybersecurity: A review," *Artificial Intelligence Review*, vol. 57, 297, 2024.

[12] V. Buhrmester, D. Münch, and M. Arens, "Analysis of explainers of black box deep neural networks for computer vision: A survey," *Machine Learning and Knowledge Extraction*, vol. 3, no. 4, pp. 966–989, 2021.

[13] A. S. Al-Aamri, R. Abdulghafor, S. Turaev *et al.*, "Machine learning for APT detection," *Sustainability*, vol. 15, no. 18, 13820, 2023.

[14] A. Georgiadou, S. Mouzakitis, and D. Askounis, "Assessing MITRE ATT&CK risk using a cyber-security culture framework," *Sensors*, vol. 21, no. 9, 3267, 2021.

[15] B. Al-Sada, A. Sadighian, and G. Oligeri, "Analysis and characterization of cyber threats leveraging the MITRE ATT&CK database," *IEEE Access*, vol. 12, pp. 1217–1234, 2024

[16] S. K. Bonhard, P. G. Villalta, O. Rosés *et al.*, "A review of Tactics, Techniques, and Procedures (TTPs) of MITRE framework for Business Email Compromise (BEC) attacks," *IEEE Access*, vol. 13, pp. 50761–50776, 2025.

[17] I. Branescu, O. Grigorescu, and M. Dascalu, "Automated mapping of common vulnerabilities and exposures to MITRE ATT&CK tactics," *Information*, vol. 15, no. 4, 214, 2024.

[18] D. Schönle and C. Reich. (June 2025). Evaluation of AI attack mitigation: From Citrix bleed to self-evolving malware: Modernising aerospace cyber defence with AI. [Online]. Available: https://tinyurl.com/5y32hrfc

[19] L. Diana, P. Dini, and D. Paolini, "Overview on intrusion detection systems for computers networking security," *Computers*, vol. 14, no. 3, 87, 2025.

[20] N. Daniel, F. K. Kaiser, S. Giladi *et al.*, "Labeling Network Intrusion Detection System (NIDS) rules with MITRE ATT&CK techniques: Machine learning vs. large language models," *Big Data and Cognitive Computing*, vol. 9, no. 2, 23, 2025.

[21] W. Li, W. Meng, and L. F. Kwok, "Surveying trust-based collaborative intrusion detection: State-of-the-art, challenges and future directions," *IEEE Communications Surveys and Tutorials*, vol. 24, no. 1, pp. 280–305, 2022.

[22] M. Timmons, D. Lukaszewski, and G. Xie, "A case for network-wide orchestration of host-based intrusion detection and response," arXiv preprint, arXiv: 2504.06241, 2025. doi: 10.48550/arXiv.2504.06241

[23] B. Fteiha, H. Zia, M. Zeyadeh *et al.*, "Enhancing IoT network security: A literature review of intrusion detection systems and their adaptability to emerging threats," *Open Computer Science*, vol. 15, no. 1, 20250046, 2025.

[24] S. S. Karim, M. Afzal, W. Iqbal *et al.*, "Advanced Persistent Threat (APT) and intrusion detection evaluation dataset for Linux systems 2024," *Data in Brief*, vol. 54, 110290, 2024.

[25] O. Arreche, T. R. Guntur, J. W. Roberts *et al.*, "E-XAI: Evaluating black-box explainable AI frameworks for network intrusion detection," *IEEE Access*, vol. 12, pp. 23954–23988, 2024.

[26] Q. Meng, N. Oo, Y. Jiang *et al.*, "Poster: M2ASK: A correlation-based multi-step attack scenario detection framework using MITRE ATT&CK mapping," in *Proc. 2024 ACM SIGSAC Conf. on Computer and Communications Security*, 2024, pp. 4979–4981.

[27] S. Zhang, X. Xue, and X. Su, "DeepOP: A hybrid framework for MITRE ATT&CK sequence prediction via deep learning and ontology," *Electronics*, vol. 14, no. 2, 257, 2025.

[28] V. Z. Mohale and I. C. Obagbuwa, "Evaluating machine learning-based intrusion detection systems with explainable AI: Enhancing transparency and interpretability," *Frontiers in Computer Science*, vol. 7, 1520741, 2025.

[29] D. Gaspar, P. Silva, and C. Silva, "Explainable AI for intrusion detection systems: LIME and SHAP applicability on multi-layer perceptron," *IEEE Access*, vol. 12, pp. 30164–30175, 2024.

[30] H. N. Eke and A. Petrovski, "Advanced persistent threats detection based on deep learning approach," in *Proc. 2023 IEEE 6th International Conf. on Industrial Cyber-Physical Systems (ICPS)*, 2023, pp. 1–10.

[31] Y. Ren, Y. Xiao, Y. Zhou *et al.*, "CSKG4APT: A cybersecurity knowledge graph for advanced persistent threat organization attribution," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 6, pp. 5695–5709, 2023.

[32] J. F. Loevenich, E. Adler, T. Hürten *et al.*, "Automating cyber threat intelligence and attack chain generation using cyber security knowledge graphs and large language models," in *Proc. 2025 International Conf. on Military Communication and Information Systems (ICMCIS)*, 2025, pp. 1–10.

[33] Y. Zhou, Z. Wang, Y. Jiang *et al.*, "AEKG4APT: An AI-enhanced knowledge graph for advanced persistent threats with large language

model analysis," *ACM Trans. Intell. Syst. Technol.*, 2025. https://doi.org/10.1145/3735645
[34] R. Jing, Z. Jiang, Q. Wang *et al.*, "From fine-grained to refined: APT malware knowledge graph construction and attribution analysis driven by multi-stage graph computation," in *Proc. 24th Int. Conf. Computational Science*, 2024, pp. 78–93.
[35] A. Daoud and M. Hamdi, "AI and adaptive cybersecurity strategies in Higher Education Institutions (HEIs): Towards a secure digital infrastructure," in *Proc. 2025 International Conf. on Smart Learning Courses (SCME)*, 2025, pp. 120–128.
[36] M. Elam, D. Mink, S. S. Bagui *et al.*, "Introducing UWF-ZeekData24: An enterprise MITRE ATT&CK labeled network attack traffic dataset for machine learning/AI," *Data*, vol. 10, no. 5, 59, 2025.
[37] M. Sarhan, S. Layeghy, and M. Portmann, "Towards a standard feature set for network intrusion detection system datasets," *Mobile Networks and Applications*, vol. 27, pp. 357–370, 2022.
[38] Z. S. Chen, R. Vaitheeshwari, E. H. K. Wu *et al.*, "Clustering APT groups through cyber threat intelligence by weighted similarity measurement," *IEEE Access*, vol. 12, pp. 141851–141865, 2024.
[39] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. International Conf. on Information Systems Security and Privacy*, 2018, pp. 108–116.