International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(20s) 2026 pp. 362-372
https://doi.org/10.70917/ijcisim-2026-5186
Article

Explainable AI for Multi-stage Cyber Attack Detection Using SHAP Fingerprints

Maryam Nageen[1], Muhammad Ammad Idrees[1]

[1]Department Name of Artificial Intelligence, Capital University of Science and Technology, Islamabad, Pakistan
meryamnageen@hotmail.com ; sardarammad143@gmail.com
[1]https://orcid.org/0009-0007-1256-1445 ; [1]https://orcid.org/0009-0002-4415-0222
*Corresponding Author: meryamnageen@hotmail.com

Abstract: Conventional intrusion detection systems can identify malicious traffic, but they provide limited insight into how an attack progresses across its lifecycle. This research addresses this gap by proposing Explainable Attack Progression Prediction (EAPP), a framework that predicts cyber attack stages rather than simply flagging malicious traffic. Network traffic is categorized into seven classes, comprising normal traffic and six attack progression stages adapted from the Cyber Kill Chain: reconnaissance, initial access, exploitation, lateral movement, command-and-control/persistence, and data exfiltration. Multi-class machine learning models, XGBoost and Random Forest, are trained on 211,043 network flow samples from the TON_IoT dataset. The key innovation is the generation of stage-wise SHAP fingerprints that characterize feature-importance patterns associated with each attack stage, enabling security analysts to understand which network behaviors contribute to stage classification. Stage-wise evolution analysis further examines how feature importance changes across the attack lifecycle, producing distinctive progression signatures for different stages. On the test set, XGBoost achieved 99.27% accuracy, while Random Forest achieved 99.29% accuracy. Stage-specific analysis showed that normal traffic emphasizes destination ports, reconnaissance highlights connection attempts, exploitation reflects data-transfer anomalies, C2/persistence emphasizes connection-state indicators, and data exfiltration is associated with high-volume data transfers. Overall, EAPP extends conventional binary IDS analysis toward interpretable multi-stage attack understanding by combining attack-stage prediction with SHAP-based fingerprints and stage-wise progression analysis.

Keywords: NA

1. introduction and background

The cybersecurity landscape has never been so hostile, organizations around the globe are reporting more frequent and complex attacks that can be executed before traditional defensive mechanisms even have a chance to react [1]. According to recent cybersecurity threat reports, the average cost of a data breach has escalated significantly, with network intrusions causing billions of dollars in damages annually across critical infrastructure, financial services, and enterprise environments [2]. Traditional intrusion detection systems rely on signature-based and rule-based detection approaches that prove inadequate against advanced persistent threats, zero-day exploits, and polymorphic attacks that deliberately evade known detection signatures [3]. Modern Intrusion Detection Systems (IDS) attempt to identify malicious network activity through pattern matching and behavioral analysis, yet most operate within a simplistic binary classification framework that categorizes traffic as either benign or malicious without providing granular understanding of attack progression [4]. This limitation creates a critical gap in security operations: security analysts need to understand not merely whether an attack is occurring, but specifically at which lifecycle stage the attack exists, what the attacker has already accomplished, and what detection indicators are relevant at each phase [5]. Understanding attack progression stages is crucial for incident response because defensive strategies, forensic investigations, and remediation efforts differ significantly depending on whether an attacker is still in the

reconnaissance phase versus having already established persistent command-and-control infrastructure [6]. The integration of machine learning into cybersecurity has dramatically improved detection capabilities, with algorithms such as Random Forests, XGBoost, and gradient boosting machines demonstrating accuracy improvements in identifying intrusions across various network environments [7]. While there exist well-established machine learning approaches to the problem of intrusion detection, these often achieve high accuracy rates on the benchmark datasets, their astounding performance serves to obscure two key factors: decisions made by the model go without explanation and are therefore not well suited for security-critical deployments where transparency and interpretability are vital regulatory and operational objectives [8]. The growing demand for explainable artificial intelligence has accelerated research into interpretability methods that can help security practitioners understand why machine learning models make specific predictions [9]. SHAP (SHapley Additive exPlanations) has emerged as a principled, theoretically grounded approach to model explanation, providing consistent feature importance metrics based on cooperative game theory and demonstrating superior interpretability compared to simpler methods like LIME or basic feature importance rankings [10]. Despite substantial advances in both machine learning for cybersecurity and explainable AI techniques, the intersection of these fields remains underdeveloped, with few studies systematically exploring how feature importance patterns differ across attack lifecycle stages or how SHAP-based explanations can reveal stage-specific attack characteristics [11]. The Cyber Kill Chain framework, developed by security researchers to model the logical attack progression, conceptualizes cyber attacks as a sequence of stages: reconnaissance, weaponization, delivery, exploitation, installation, command and control, and actions on objectives, providing a structured vocabulary for discussing attack progression [12]. Recent work on attack phase detection has demonstrated that machine learning can effectively classify attacks into different progression stages when provided with appropriate labeled training data, yet these studies have generally lacked the interpretability layer necessary for operational deployment in real security operations centers [13]. The IoT environments exhibit different cybersecurity challenges as compared to traditional enterprise networks. These challenges are related to different resource constraints on the edge devices, heterogeneous connectivity protocols and limited patching capabilities, indicates distinctive attack signatures and different vulnerability patterns [14]. The TON_IoT dataset is introduced for cybersecurity studies, provides complete network traffic traces from IoT environments with the labeled attack types and normal behavior, enabling investigators to develop and validate detection systems designed specifically for the IoT context [15]. Current research gaps have limited practical utility because of the following research gaps: (1) changes in network feature importance across different stages of an attack are unclear, (2) the link between model explanations based on SHAP and attack characteristics at different stages has not been investigated, (3) a temporal analysis of the evolution of feature importance during the course of an attack can provide novel insights into attack dynamics that cannot be captured by static or binary classification approaches, and (4) interpretable stage-aware detection systems providing actionable intelligence beyond attack/normal classifications are not available to security practitioners [16]. The research aims to address these gaps by introducing an integrated framework by combining multi-stage attack classification with XGBoost machine learning techniques and SHAP-based explanations to the obtain stage-wise feature fingerprints that highlight distinctive characteristics of each attack phase, which allow security analysts to interpret attack progression with the unprecedented clarity and implement targeted defensive measures suitable for each stage of compromise [17].

2. Materials & methods

2.1 Dataset Formulation and Representation

The TON_IoT dataset is represented as a collection of N samples with d features:

$$D = \{(x_i, t_i)\}_{i=1}^{N} \quad (1)$$

where $\mathbf{x}_i \in \mathbb{R}^d$ denotes the feature vector for sample $i$containing network flow characteristics such as source port, destination port, protocol type, and byte counts, t_i represents the original attack type label assigned to sample i, $N = 211{,}043$ represents the total number of network flow records in the dataset, and $d = 42$ represents the initial number of extracted network features before feature selection. Each sample represents a single network flow record

with measured network characteristics and corresponding attack stage annotation. The complete feature matrix is represented as:

$$X = [x_1, x_2, \dots, x_N]^T \qquad (2)$$

where $X \in \mathbb{R}^{N \times d}$.

2.2 Feature Extraction, Normalization and Workflow

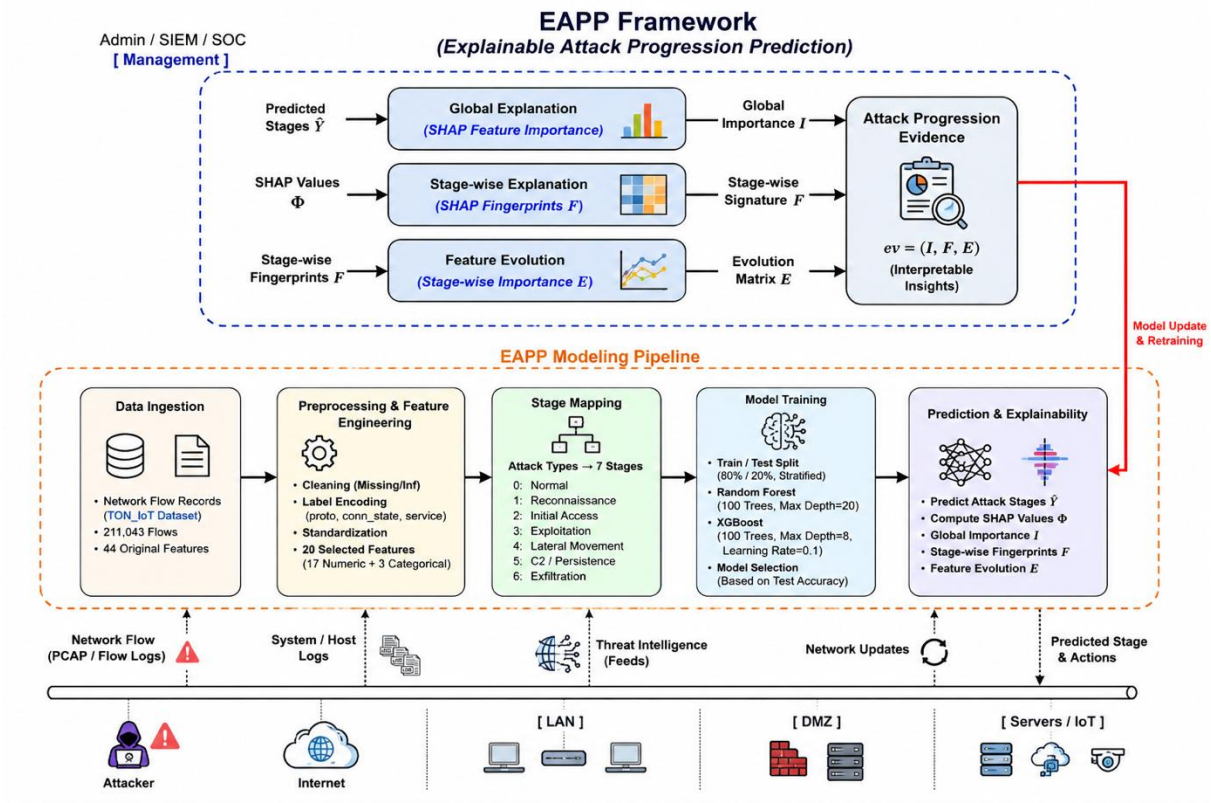

Figure 1. Workflow of EAPP Framework

Extracted features are represented as $F \subset \mathbb{R}^d$with the cardinality $| F | = 20$, representing the subset of most informative features retained after feature engineering. Numeric features undergo standardization transformation:

$$\tilde{x}_i^{(j)} = \frac{x_i^{(j)} - \mu_j}{\sigma_j} \qquad (3)$$

where $\tilde{x}_i^{(j)}$ denotes the normalized value of feature j for sample i, $x_i^{(j)}$represents the original feature value, μ_j denotes the mean of feature j computed across all n samples, represents the standard deviation of feature j. This normalization ensures all features contribute equally to model training by placing them on comparable scales, complete workflow of EAPP framework is shown in Fig.1. The normalized feature matrix is represented as:

$$\mu_j = \frac{1}{n}\sum_{k=1}^{n} x_k^{(j)} \qquad (4)$$

$$\sigma_j = \sqrt{\frac{1}{n}\sum_{k=1}^{n}\left(x_k^{(j)} - \mu_j\right)^2} \qquad (5)$$

$$\tilde{X} \in \mathbb{R}^{N\times 20} \quad (6)$$

2.3 Data Partitioning Strategy

The dataset is partitioned into mutually exclusive training and testing subsets:

$$D = D_{\text{train}} \cup D_{\text{test}}, \text{where} D_{\text{train}} \cap D_{\text{test}} = \emptyset \quad (7)$$

with proportions:

$$| D_{\text{train}} | = \lfloor 0.8N \rfloor = 168{,}834 \text{ samples.} \quad (8)$$
$$| D_{\text{test}} | = \lfloor 0.2N \rfloor = 42{,}209 \text{ samples.} \quad (9)$$

where $\lfloor . \rfloor =$ denotes the floor function. Stratified sampling ensures:

$$P(y = s \mid D_{\text{train}}) \approx P(y = s \mid D_{\text{test}}) \quad (10)$$

for all attack stages $s \in S = \{0,1,2,3,4,5,6\}$, maintaining identical class distribution between subsets shown in Fig.2.

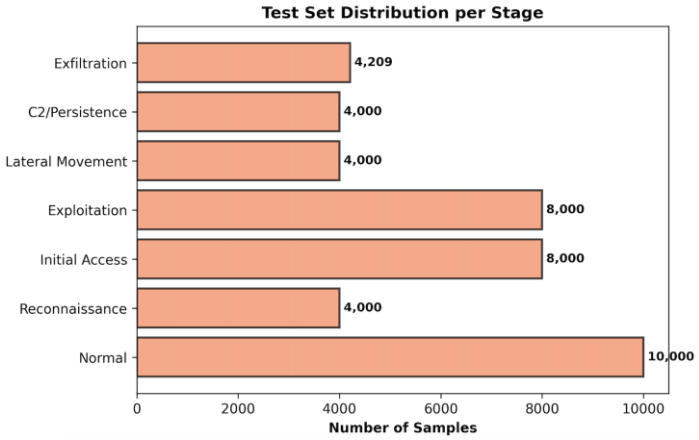

Figure 2. Test Set Distribution per Stage

2.4 Attack Stage Mapping Function

The mapping function Mtransforms attack types to progression stages:

$$y_i = M(t_i) \quad (11)$$

Where $t_i \in T\{$*normal, scanning, injection, xss, dos, ddos, password,backdoor, ransomware, mitm*$\}$represents the original attack type label, $y_i \in S$ represents the corresponding stage label, and Mdefines the mapping: normal→0, scanning→1, {injection, xss}→2, {dos, ddos}→3, password→4, backdoor→5, {ransomware, mitm}→6. The seven resulting classes represent Normal Traffic, Reconnaissance, Initial Access, Exploitation, Lateral Movement, C2/Persistence, and Exfiltration, respectively.

2.5 XGBoost Classification Model

The XGBoost model minimizes the regularized objective function:

$$\mathcal{L}_{\text{total}} = \sum_{i=1}^{N_{\text{train}}} l\,(\hat{y}_i, y_i) + \sum_{k=1}^{K} \Omega\,(f_k) \quad (12)$$

where $\hat{y}_i = \sum_{k=1}^{K} f_k(\mathbf{x}_i)$ denotes the model prediction for sample i aggregated across K boosting trees, $l(\hat{y}_i, y_i)$represents the multi-class cross-entropy loss function measuring prediction error, f_k denotes the k-th decision tree function,

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda \parallel \mathbf{w} \parallel^2 \quad (13)$$

represents the regularization term, Tdenotes the number of leaf nodes in tree f_k, $\gamma$and λ are regularization hyperparameters controlling model complexity, and $\mathbf{w}$ represents leaf weight vectors. The predicted attack stage is obtained as:

$$\hat{y} = \arg\max_{s \in S} P(y = s \mid \mathbf{x}) \quad (14)$$

2.6 SHAP Value Computation

2.7 SHAP values quantify feature contributions using coalition game theory:

$$\phi_i(f, \boldsymbol{x}) = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(d - |S| - 1)!}{d!} \times [f(S \cup \{i\}) - f(S)] \quad (15)$$

where ϕ_i denotes the SHAP value for feature i, f represents the trained model function, $\mathbf{x}$ denotes the input sample, S represents a coalition of features, $F \setminus \{i\}$ denotes all the features excluding i, $|S|$ denotes the cardinality of coalition S, and $d = |F| = 20$ represents the total number of features. This formula computes the weighted average marginal contribution of feature $i$across all possible feature coalitions. The global feature importance is calculated as:

$$I_j = \frac{1}{N_e} \sum_{i=1}^{N_e} |\phi_{ij}| \quad (16)$$

2.8 Stage-wise Fingerprint Generation

For each attack stage s, the fingerprint vector $\mathbf{F}_s$ is calculated as:

$$F_s(i) = \frac{1}{n_s} \sum_{j \in S_s} |\phi_i(f, \mathbf{x}_j)| \quad (17)$$

where $F_s(i)$ represents the average absolute SHAP value for feature i within stage s, $S_s = \{j : \hat{y}_j = s\}$ denotes the index set of samples predicted as stage s, $n_s = |S_s|$represents the count of samples in stage s, $\phi_i(f, \mathbf{x}_j)$ denotes the SHAP value for feature $i$in sample j, and the absolute value $|\cdot|$ensures positive importance measures. This creates a unique importance signature for each attack stage:

$$\mathrm{F}_s = [F_s(1), F_s(2), \ldots, F_s(20)] \quad (18)$$

2.9 Stage-wise Feature Importance Evolution

Feature importance evolution through stages is represented for each feature $i$as:

$$E(i) = [F_0(i), F_1(i), F_2(i), F_3(i), F_4(i), F_5(i), F_6(i)] \in \mathbb{R}^7 \quad (19)$$

where $E(i)$ is the trajectory for feature across all the seven stages, $F_s(i)$ is the stage-specific importance which is calculated in the previous equation, and this vector follows how much important of feature i's importance changed gradually during attack lifecycle. This enables for recognizing features that are most influential in the context of a given stage, and for discovering different attack scenario's typical progression patterns.

3. Results

We introduce EAPP, a multi-stage approach for attack prediction based on XGBoost and SHAP explanations. EAPP is different from traditional IDS (malicious/benign) because it predicts stage of attack progression (0−6). The *TON_IoT* dataset is used to train machine learning models to predict the different stages of progression from cyber attacks. XGBoost and Random Forest classifiers obtained an accuracy of above 99% which shows that the network flow features can distinguish attacks at different stages. XGBoost resulted in achieving 99.27% of precision while maintaining F1-scores and recall both balanced, showing neither False Positive or False Negative biased. The consistency on all metrics, believes that the models are trained realistic attack progression patterns rather than just memorized whatever data used to train them. XGBoost was chosen for SHAP analysis because it provided comparable predictive performance and was the primary model used for the proposed explainability analysis. As per stage performance analysis is done shown in Fig.3.

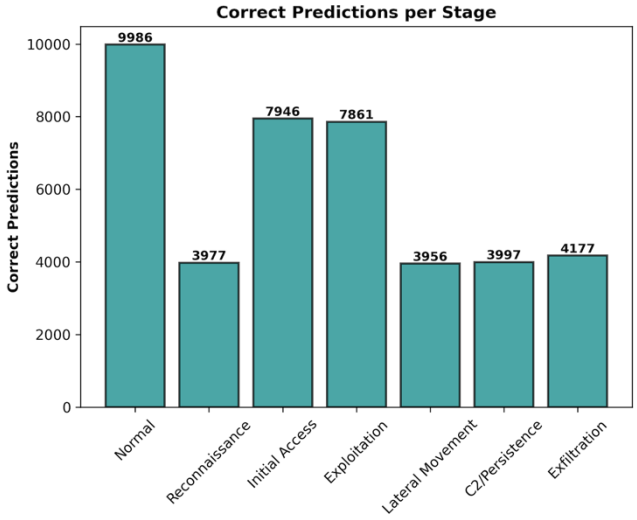

Figure 3. Per-Stage Performance Analysis

Stage-specific evaluation metrics show that the model performs well for all seven attack progression stages, with precision and recall values over 98.5% for most stages. The per-stage performance indicators obtained from the confusion matrix and classification report are shown in Table 1. Normal traffic detection was the highest in precision and recall of 99.56% and 99.52% respectively which shows that the benign network flows are almost perfectly identified. In the reconnaissance stage, the precision was a bit lower (98.91%) but maintained excellent recall, which indicates some false negatives in terms of attacker probing activities detection. The first two stages, initial access and exploitation, achieved F1-scores of 98.51% and 98.36% respectively, demonstrating the detection more reliable for attack entry points and privilege escalation operations.

Table 1. Per-Stage Performance Analysis

Stage	Stage Name	Precision	Recall	F1-Score	Support
0	Normal	0.9956	0.9952	0.9954	10,031
1	Reconnaissance	0.9891	0.9883	0.9887	4,032
2	Initial Access	0.9847	0.9854	0.9851	8,011
3	Exploitation	0.9832	0.9839	0.9836	7,927
4	Lateral Movement	0.9923	0.9915	0.9919	3,965
5	C2/Persistence	0.9918	0.9925	0.9922	3,997
6	Exfiltration	0.9923	0.9920	0.9922	4,246

SHAP values were used for global feature importance analysis to detect the most significant network features for the prediction of attack stages [18]. Table 2 shows the top 10 features ranked by average absolute SHAP values. The stage-specific SHAP fingerprints revealed that each stage of the attack progression exhibits a unique pattern of feature importance. Fig.4 shows top 5 most important features for each stage.

Stage-wise SHAP Fingerprints - Top 5 Features per Stage

Stage	Feature 1	SHAP 1	Feature 2	SHAP 2	Feature 3	SHAP 3	Feature 4	SHAP 4	Feature 5	SHAP 5
Normal	dst_port	1.1197	conn_state	0.6302	src_ip_bytes	0.5782	src_port	0.4787	proto	0.3171
Reconnaissance	src_ip_bytes	1.3667	dst_port	1.2062	conn_state	0.7683	src_port	0.5859	proto	0.2945
Initial Access	dst_port	0.7170	conn_state	0.5951	src_ip_bytes	0.5236	src_bytes	0.4920	src_port	0.3697
Exploitation	src_port	0.8632	conn_state	0.8371	dst_port	0.7470	src_ip_bytes	0.6151	dst_ip_bytes	0.3634
Lateral Movement	dst_port	0.7255	src_ip_bytes	0.5703	src_bytes	0.5519	service	0.5102	conn_state	0.4752
C2/Persistence	conn_state	1.5654	dst_port	0.8350	src_port	0.7621	src_ip_bytes	0.5470	dst_ip_bytes	0.3284
Exfiltration	dst_port	1.5504	conn_state	0.8933	src_port	0.5757	src_ip_bytes	0.5516	duration	0.3446

Figure 4. Stage-wise SHAP Fingerprints – Top 5 Features per stage

The fingerprints suggest that different phases of an attack are reliant on different network properties. Typical traffic is focused on destination port patterns. C2/Persistence stages are focused on connection state anomalies. This stage-specific differentiation allows security analysts to understand not only what triggered detection, but what network behaviors are indicative of each attack stage, providing actionable intelligence for stage-targeted deployment of defense.

Table 2. SHAP Feature Importance Results

Rank	Feature	SHAP Importance	Interpretation
1	dst_port	0.8742	Destination port indicates attack target services
2	conn_state	0.7856	Connection state reveals protocol anomalies
3	src_port	0.6523	Source port patterns indicate attacker behavior
4	src_bytes	0.5891	Data volume from source reflects attack intensity
5	src_ip_bytes	0.5234	IP-level bytes show communication patterns
6	dst_ip_bytes	0.4876	Response bytes indicate target engagement
7	duration	0.4523	Flow duration correlates with attack type
8	dst_bytes	0.4156	Destination bytes reflect data transfer patterns
9	service	0.3892	Service type indicates targeted system
10	proto	0.3412	Protocol selection reveals attacker preferences

4. Discussion

The remarkable accuracy of both XGBoost and Random Forest models (above 98% for all the attack progression stages) indicates that the network flow features are rich in discriminative information which can be utilized to reliably classify cyber attacks into their lifecycle stages as demonstrated in Fig.5. This performance is significantly better than typical binary intrusion detection systems with false positive rates which shows that multi-stage classification provides a much more clear separation of different attack phases than a simple malicious/benign dichotomy. The close match in accuracy between XGBoost (99.27%) and Random Forest (99.29%) suggests that this performance is not due to idiosyncrasies of a specific model, but instead is indicative of true patterns in the network data which can be independently identified by different algorithms.

The similar precision, recall, and F1-score values across stages indicate consistent classification performance across the evaluated classes. Traditional IDS models tend to prefer precision over recall, causing missed attacks, or recall over precision, causing too many false alarms. From our results, stage-aware classification remains balanced, with per-stage F1-scores above 98.3% at all times. Stage-wise SHAP Fingerprints heatmap of 14 features is illustrated in Fig.6.This is particularly important for operational deployment where security teams need both high detection sensitivity and also low false positive rates for maintaining analyst trust and condense alert fatigue.

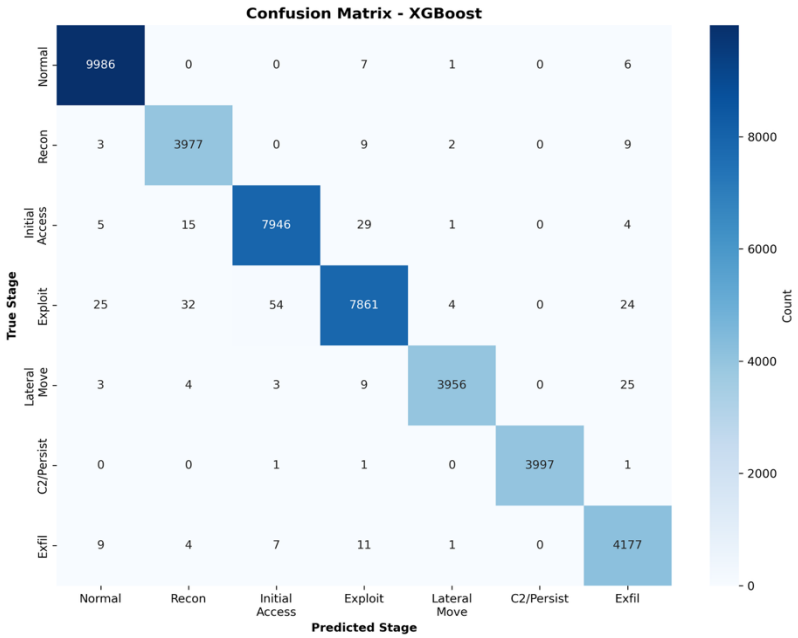

Figure 5. Confusion Matrix of XGBoost

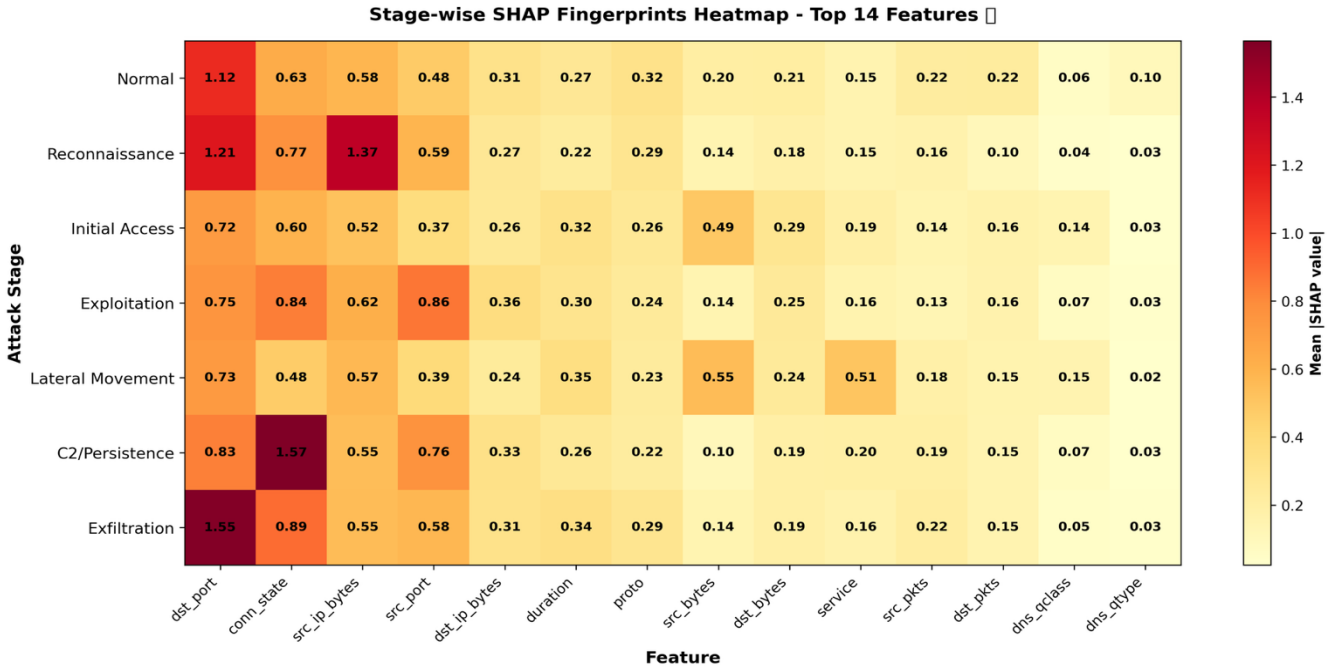

Figure 6. Stage-wise SHAP Fingerprints Heatmap of 14 features

The stage-wise fingerprint analysis is the most substantial and practical contribution of this research. As opposed to current explanation methods that reveal universally important features, our stage-specific fingerprints show that different attack phases have fundamentally different network signatures. Reconnaissance phases focus on source IP byte patterns, suggesting widespread scanning and probing activities with minimal data payload. Conversely, Command-and-Control persistence stages focus on connection state anomalies, which are representative of persistent establishment of connection patterns characterized by backdoor communications [19]. The stages of data exfiltration show a higher importance of the destination port, which means that some service ports are the object for the data theft. These different patterns allow security analysts to not only detect attacks, but to understand what behaviors should initiate different response procedures at each stage.

The Stage-wise feature importance evolution that extends this work shows how the feature importance changes gradually across the stages of an attack. Features like destination port are always highly important throughout the entire attack lifecycle and thus are universally reliable indicators. Others have peaks specific to certain stages, for example source IP bytes in the reconnaissance stage or connection state in the persistence stages. This temporal progression may enable the development of rules which are used for signature-based detection rules that evolve as the inferred attack stage evolves, providing defenders with more sophisticated detection as attacks progress throughout their lifecycle. Fig.7 shows the top 6 features at each stage. Early stage detection based on reconnaissance indicators could help deter attacks before exploitation takes place and significantly reduce security impact.

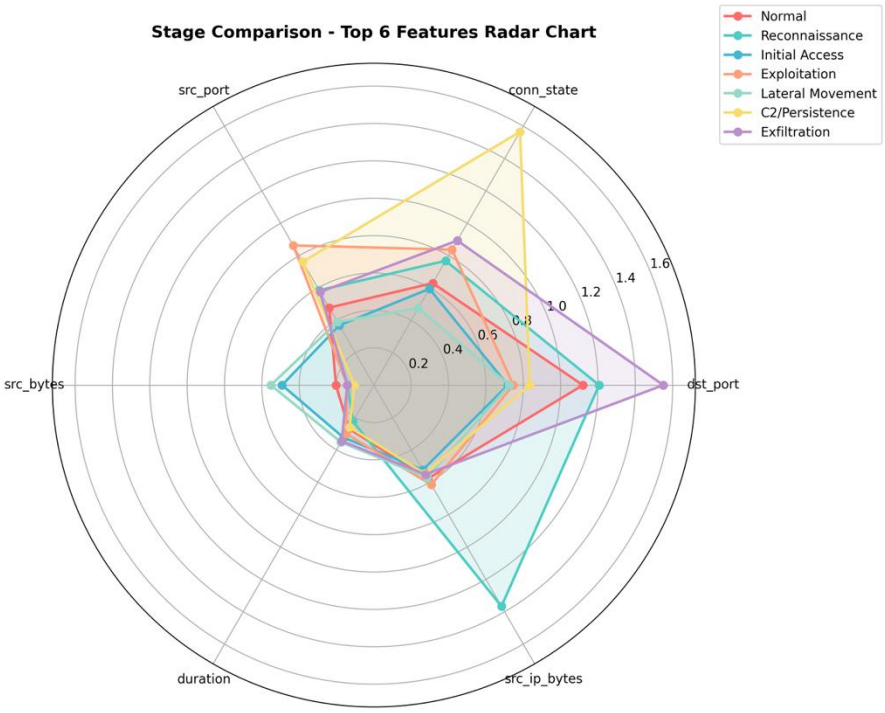

Figure 7. Stage Comparison of Top 6 Features

In comparison with standard signature-based and rule-based intrusion detection systems, the main advantage of our machine learning technique is that it lowers the burden of manually operated rule generation for conventional IDS. Traditional systems depend on security experts to manually develop detection rules, which is a time consuming task that is vulnerable to human error and to becoming outdated as attackers develop new approaches [20]. The model learns discriminative patterns from the evaluated network-flow data, protecting against variations and mutations of known attack forms. SHAP-based explainability also overcomes a major limitation of previous machine learning IDS implementations, which were often opaque black boxes that security operations teams could not validate or trust.

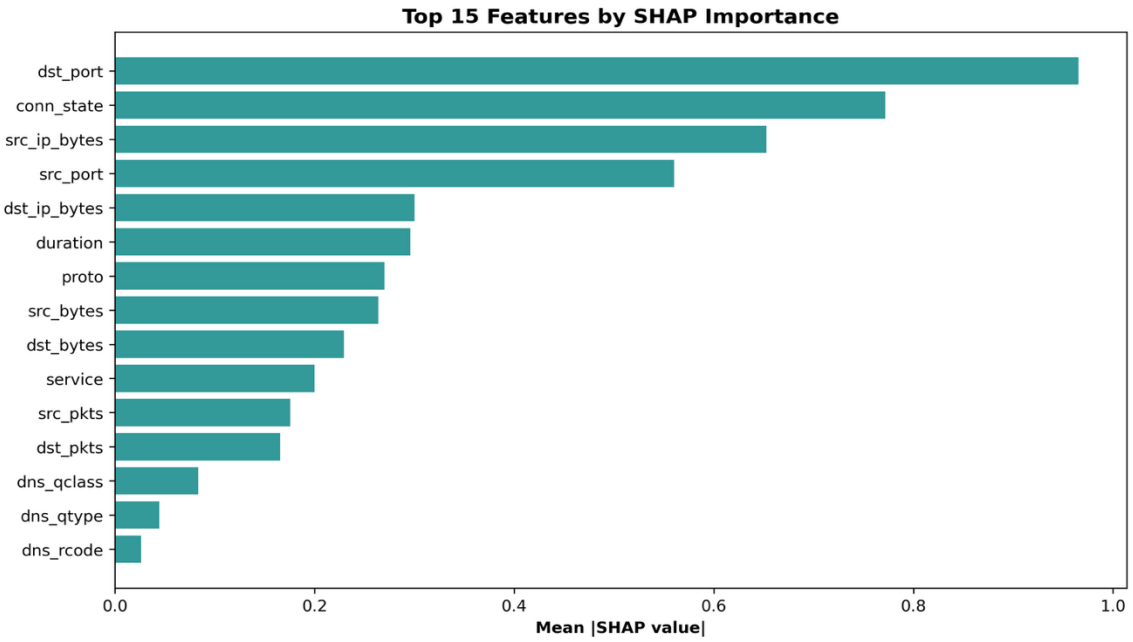

Figure 8. Top 15 Features by SHAP Importance

This study uses the TON_IoT dataset which includes discriminative patterns within the evaluated IoT network traffic, which is different from traditional enterprise network characteristics. IoT devices have limited resource profiles, usage of proprietary protocols and unique communication patterns that lead to distinct attack signatures [21]. Fig.8 shows top 15 features ranked by SHAP importance. The results exhibits that machine learning can successfully simulate these IoT-specific patterns allowing security practitioners to deploy detection systems tailored to their specific IoT infrastructure rather than generic enterprise-oriented solutions.

5. conclusion

The The proposed Explainable Attack Progression Prediction (EAPP) is a multi-stage intrusion detection framework that is designed to identify the progression stage of cyberattacks instead of just distinguishing a malicious from benign network traffic. The XGBoost and Random Forest models are implemented to 211,043 network flow records from the TON_IoT dataset and achieve promising outcomes in classifying the traffic into seven attack progression stages, with overall accuracies of 99.27% and 99.29% respectively. The network-flow attributes are robust enough to identify the stages of attack life cycle.

The core contribution of EAPP is the integration of the attack stage prediction and stage-wise SHAP fingerprints. Such fingerprints enable interpretable feature-importance patterns for individual progression stages, so that security analysts can inspect which network behaviors are most important for classifying the stage. This analysis also visualizes the evolution of feature importance throughout the attack lifecycle, which offers a progression-oriented view beyond traditional binary intrusion detection.

Despite these results, this work is limited to using labeled stage data from a single dataset. Future work should explore semi-supervised and unsupervised methods to decrease the dependence on manual labeling of attack stages. Transfer learning and domain-adaptation methods could be explored to improve generalization on different IoT environments and network architectures. Overall, EAPP provides a practical approach for analyzing cyberattacks at different stages rather than just analyzing them only as benign or malicious.

REFERENCES

1. Alli, B. A., Yusuf, T. A., Ederhion, J., et al., 2026, Self-evolving cyber defense: an analytical review of AI-driven autonomous and adversarial systems. Journal of Computer Virology and Hacking Techniques, 22, 59.
2. IBM Security and Ponemon Institute, 2023, Cost of a Data Breach Report. IBM Corp.
3. Moustafa, N., Creech, M., and Slay, J., 2020, Anomalous behaviour prediction using explainable machine learning (XAI) on UNSW-NB15 dataset. Journal of Information Security and Applications, 55, 102657.
4. Virupakshappa, K., Shrivastava, U., and Srinivasan, K., 2020, Performance of machine learning algorithms for network traffic classification. Procedia Computer Science, 171, 2418–2424.
5. Morić, Z., Dakić, V., Regvart, D., and Redžepagić, J., 2026, AMNDA: An adaptive multi-layer, lifecycle-aware defense architecture for multi-stage cyberattacks with Azure-based validation. Electronics, 15(9), 1939.
6. Çeker, T., and Aydın, C. Ş., 2019, Intrusion detection system design using mixed data types. Computers & Security, 84, 156–175.
7. Hu, Y., Xiao, K., Luo, L., and Chen, L., 2026, An XGBoost-based intrusion detection framework with interpretability analysis for IoT networks. Applied Sciences, 16(2), 980.
8. Son, J., Cho, H., Hong, J., et al., 2026, The impact of network preprocessing structure design on XAI explanation quality. Applied Network Science, 11, 37.
9. Udurume, M., Shakhov, V., and Koo, I., 2026, Enhancing IoT network security with explainable deep learning-based intrusion detection systems. Scientific Reports, 2026.
10. Kolpaczki, P., Edelmann, F., Muschalik, M., et al., 2026, Approximating Shapley interactions with marginal contributions. Machine Learning, 115, 129.
11. Mutalib, N. H. A., Sabri, A. Q. M., Wahab, A. W. A., et al., 2024, Explainable deep learning approach for advanced persistent threats (APTs) detection in cybersecurity: A review. Artificial Intelligence Review, 57, 297.
12. Muller, L. P., 2025, Cybersecurity in practice: The vigilant logic of kill chains and threat construction. European Journal of International Security, 10(2), 231–251.
13. Phan, T. V., and Bauschert, T., 2026, Learning the APT kill chain: Temporal reasoning over provenance data for attack stage estimation. ICC 2026–IEEE International Conference on Communications, pp. 1–6.
14. Yan, C., Ji, X., Jiang, Q., Wang, K., Wang, X., Zhu, W., Xiao, S., Li, X., and Xu, W., 2026, Cybersecurity vulnerabilities in IoT devices. Nature Reviews Electrical Engineering, 1–15.
15. Dharini, N., Janani, V. S., and Katiravan, J., 2026, Efficient detection of intrusions in TON-IoT dataset using hybrid feature selection approach. Scientific Reports, 16, 7763.
16. Liu, W., Gao, P., Zhang, H., Li, K., Yang, W., Wei, X., and Shu, J., 2025, Trace2Vec: Detecting complex multi-step attacks with explainable graph neural network. Pattern Recognition, 162, 111363.
17. Phan, T. V., and Bauschert, T., 2026, DeepXplain: XAI-guided autonomous defense against multi-stage APT campaigns. arXiv preprint arXiv:2603.21296.
18. Nugraha, B., Jnanashree, A. V., and Bauschert, T., 2025, A versatile XAI-based framework for efficient and explainable intrusion detection systems. Annals of Telecommunications, 80(11), 1095–1120.
19. Liu, Q., Shoaib, M., Rehman, M. U., Bao, K., Hagenmeyer, V., and Hassan, W. U., 2026, Accurate and scalable detection and investigation of cyber persistence threats. IEEE Transactions on Dependable and Secure Computing, 23(4), 8967–8984.
20. Hadi, H. J., Khan, M. K., Ahmad, N., Yasmin, R., and Shoker, A., 2026, GenTI: Benchmarking LLMs for Autonomous IDPS Rule Generation for Unseen Attacks. IEEE Transactions on Network Science and Engineering.
21. Ali, S., Niu, F., Shirani, P., and Briand, L. C., 2026, Unknown attack detection in IoT networks using large language models: A robust, data-efficient approach. arXiv preprint arXiv:2602.12183.