Multidisciplinary Journal of Academic Publications
ISSN (Online): 3107-538X
Vol. 01, Issue 02, November – December 2025

DEEPAPT-SHIELD: A MULTI-STAGE DEEP LEARNING FRAMEWORK FOR ADVANCED PERSISTENT THREAT DETECTION AND ATTRIBUTION IN ENTERPRISE NETWORKS

Dr. Harshvardhan P. Ghongade[1], Dr. Anjali A. Bhadre[2]

Department of Mechanical Engineering, Brahma Valley College of Engineering and Research Institute, Nashik, India - ghongade@gmail.com[1]

Department of Information Technology, G.H. Raisoni College of Engineering and Management, Pune, India - anjalibhadre38@gmail.com[2]

Abstract

Advanced Persistent Threats (APTs) - the most sophisticated type of cyber-attacks, stealthy in long duration, multi-staged attack trains and nation-state level resources. Low-and-slow attack patterns and the use of legitimate tools for malicious activities make it difficult to detect APTs with traditional intrusion detection systems. This paper presents a new multi-stage deep learning model called DeepAPT-Shield for APT detection and attribution in enterprise networks. Our solution must deal with three main tasks: (1) capturing subtle behavioral deviations that may indicate the presence of an APT, (2) correlating alert messages and attack indications at multiple locations in a temporalspatial fashion, and (3) attributing found threats to specific known APT groups that can drive a precise reaction. It contains four related modules: 1) A GAT to model entity behavior; 2) A TCN with attention mechanisms for sequence analysis; 3) A Heterogeneous Graph Neural Network for attack chain correlation and a Siamese Network for threat attribution. We make three primary contributions: (1) an adaptive threshold mechanism that reduces false positives by 67% with minimal effect on detection rates; (2) a new kill-chain aware loss function that heavily penalizes the inability to detect stages of the attack that enable other stages to occur, even if those "enabling" stages are harmless per se; and (3) semi-supervised learning for training the model on limited labeled APT data. Extensive experiment on DARPA OpTC dataset (17.4B events), LANL Unified Host and Network Dataset (58-days enterprise activity) and a proprietary dataset from 5 fortune-500 companies indicates the proposed approach outperforms all competitors. DeepAPT-Shield obtains 94.7% detection rate of APT campaigns with just 0.003% false positive rate, and detects attacks average at 18.3 days earlier than the state-of-the-arts commercial solutions. The attribution module correctly attributes APT groups in 89.2% on average, among a total of existing 12 identified threat actors. We have operationalized our system and our method to production environment that catch and stop three new unknown APT campaigns.

Keywords: *Deep Learning[1], Advanced Persistent Threats[2], Graph Neural Networks[3], Intrusion Detection[4], Cyber Threat Intelligence[5], Attack Attribution[6], Enterprise Security[7].*

1. Introduction

The industry's cyber defense paradigm has changed at its very foundation with the rise of Advanced Persistent Threats (APTs) as the principal threat to governments and enterprises globally. Unlike traditional cyber-attacks whose main target is short-term financial gains, APTs have goals of higher strategic value that require operational timelines which extend over month or even years and resources usually available only to nation-states [1]. Victim organizations such as SolarWinds (APT29), Colonial Pipeline, and Microsoft Exchange Server exploitation (HAFNIUM) have been in the limelight recently about the destructive capabilities of APT campaigns [3],whereas total damages from these campaigns worldwide are estimated to be higher than $100 billion per year?2?. Conventional security paradigms such as signature-based IDSs and rule-based SIEM systems are essentially powerless against APT threats. APT groups actively leverage techniques designed to avoid traditional detection and attribution: employing native administrative tools ("living off the land"), maintaining continuous access through a variety of backdoors, and idle periods between working engagements [3]. The MITRE ATT&CK framework lists more than 200 different techniques used by APT groups, most of which are designed to produce minimal discriminating signatures [4]. Machine learning techniques have been demonstrated to be promising in anomaly-based detection; however, the current ones still suffer from severe drawbacks especially for APT. In the first place, most of ML-based IDSs concentrate in network traffic and host behavior but they do not correlate indicators that span across the entire enterprise [5]. Second, supervised learning-based methods need a large volume of labeled attack data, which is rare for APT attacks and becomes obsolete in the presence of constantly changing attackers [6]. Third, the huge class imbalance between normal traffic and APT attacks (in many cases greater than 1:1M) makes it infeasible to adopt traditional methods for classification [7].

In this paper, we present DeepAPT-Shield to remedy these limitations with novel architecture designs and training strategies. Our system is built on enterprise provenance graphs— fine-grained logs of system events that record causal relations between processes, files, and network connections[8]. Formulating the enterprise as a graph that evolves over time, and exploring temporally evolving entity behaviors, DeepAPT-Shield can discover such subtle anomalous patterns of entity behaviors specific to APT operations at operationally acceptable false positive levels. Our contributions are threefold. 1. We start by devising a multistage detection architecture such that enterprise telemetry is processed through specialized neural network modules, each tailored for different dimensions of APT behavior: entity-level anomalies, temporal attack patterns, and cross-system attack chains. Such modular design support interpretability of detection and integration with security analyst's workflow [9]. Second, we propose innovative training techniques that mitigate the APT-specific issues. Our adaptive threshold process also varies the sensitivity of our detection according to environmental context and reduces false positives by 67% against a fixed threshold. Kill-chain aware loss function focuses on detection of critically-impacting attack stages (i.e., privilege escalation, lateral movement and data exfiltration), and it also improves overall campaign detection by 23% [10]. The third one is a threat attribution module, which associates the detected attacks with known APT groups via behavioral features. This novel support, which is not provided in any previous academic studies, allows organizations to comprehend the motives of adversaries and anticipate their behavior so that they can collaborate more effectively with threat intelligence communities for a proactive defense [11].

Comprehensive performance evaluations show that DeepAPT-Shield is superior in every way. On the DARPA OpTC dataset, we get a 94.7% detection rate at a false positive rate of about 0.003%, which is significantly better than exiting academic and commercial systems. The AttAttribution module can pinpoint the threat group responsible for an attack with 89.2% of accuracy, offering valuable intelligence to security teams. Five large enterprises of the fortune-500 size have sucessfully detected and prevented three new unknown APT campaigns in their production systems using this layer [12].

2. Literature Review

2.1 APT Nature and Attack Life Cycle

It is necessary for the requirement of effective detection systems to fully comprehend operating features with APT. Mandiant's APT1report [13] set the groundwork for knowledge on state-level adversaries and outlined systematic methods of target reconnaissance, initial compromise, enduring persistence. His work was followed by that of FireEye 14, CrowdStrike 15, Kaspersky Lab 16 research groups and others to move this knowledge on other dozens of APT groups with different objectives and practices. The cyber-kill chain model, the first of its kind from Lockheed Martin [17], is a clear-cut process which revealed how APTs progress: reconnaissance -- weaponization -- delivery – exploitation –installation (also by-passed that an operating system would prevent all malware) -- command and control (C2) -- actions on objectives. Though helpful in forming mental models, academia reveals the linear progression can easily be subverted; actual APT campaigns are not conducted through this method but through non-linear cycles with attackers changing tactics in response to other mechanisms ([18] ibid) The MITRE ATT&CK move one step further down the taxonomy tree with specific technical attacks learned from public references of campaigns [19]. Key traits of APT that make it difficult to detect with traditional methods: average dwell time for APT malware is about 287 days [20], leveraging trusted tools and credentials to move laterally in the network [21], encrypted C2 communications hidden within legitimate traffic stream [22], multiple techniques for ensuring persistence so as to not be discovered etc. [23]. These all combine to form detection difficulties that simply cannot be overcome using signatures.

2.2 Machine Learning utilized for Intrusion Detection

There has been considerable evolution in intrusion detection using machine learning techniques over last decade. In the early days of network flow classification, researchers utilized traditional ML techniques such as decision trees, support vector machines, and random forests [24]. Deep learning permitted the processing of raw packet data and resulted in improved detection performance with CNNs [25] and RNNs [26] that help learning malicious traffic patterns. The host-based anomaly detection has also been developed through deep learning. DeepLog [27] used LSTMs for system log analysis and learned to identify anomalies via prediction error. LogRobust [28] extended this to make use of attention beyond interpretability. ATLAS [29] proposed sequence-to-sequence models for audit log analysis, and obtained state-of-the-art performance on public datasets. Graph-based methods have shown especially promising for enterprise security. UNICORN [30] modelled system events as provenance graphs, and used graph sketch techniques for anomaly detection. ThreaTrace [31] further supplemented this with graph based neural networks, and KAIROS [32] incorporated temporal component via evolution modeling of the graph. However, they are single-host only, and cannot be used to detect large-scale APT campaigns across enterprises.

2.3 Graph Neural Network for Security

Graph neural networks (GNNs) have revolutionised security analytics and made possible the learning of representations in structured data [33]. Graph Convolu- tional Networks (GCNs) [34] perform spectral convolutions to propagate local information at the neighborhood level; GraphSAGE [35] extended the idea of sampling to improve scalability. Attention Mechanisms-based GNNs: Graph Attention Networks (GAT) [36] integrated attention mechanisms to perform weighted aggregation, which were especially suitable for heterogeneous security data. GNNs for cybersecurity has achieved remarkable improvements. E-GraphSAGE [37] adopted edge-aware graph learning for network intrusion detection. GNN-NIDS [38] also achieved impressive performance on the CIC-IDS2018 dataset. MAGIC [39] presented the graph classification for malware analysis. Nevertheless, the state-of-the-art GNN methods in security domain mainly concern single

attack categories instead of comprehensive APT detection. In enterprise security, HGNNs show great potential due to presence of complex inter-entity (such as "users", "processes", "files" and 10"network connections") relationships [40]. HAN [41] proposed hierarchical attentions for heterogeneous graphs, while HGT [42] presented special transformers for modeling the heterogeneity. These designs are well suited to enterprise provenance graphs.

2.4 Threat Attribution and Intelligence

Threat Attribution: Assigning responsibility of cyber attacks to actors is still a major problem with less focus in academic research [43]. Commercial TI platforms (Recorded Future, Mandiant Advantage, CrowdStrike Falcon Intelligence) offer attribution but are often human-centric and depend on manual analysis or indicator matching instead of automated behavior exploration [44]. Most attribution work in academia centers around identifying malware authorship by code analysis.[45] and writing style [46]. APT-Attribution [47] proposed TTPs (Tactics, Techniques and Procedures)-based behavioral fingerprinting and achieved 78% accuracy using small datasets. But the full attribution to behaviors combined with detection is still a blank field of research.

2.5 Research Gaps

There still exists fundamental deficiencies in APT detection research. First, traditional detection techniques are focused on detecting single-stage attack payloads and not whole-campaigns; there may be cases where an incident is undetectable by these existing methods due to constituent multi-stage kill chain presence [48]. Second, training strategies do not adapt to APT-special challenges such as very imbalanced class distribution and limited annotated data [49]. Third, detection systems have no attribution capabilities which reduce the effectiveness of defensive responses [50]. DeepAPT-Shield fills these gaps with multi-stage detection mechanism, innovative training scheme and behavioral attribution.

3. Methodology

3.1 System Architecture Overview

DeepAPT-Shield is composed of four interconnected modules on enterprise provenance graphs: (1) Entity Behavior Modeling (EBM) with Graph Attention Networks, (2) Temporal Sequence Analysis (TSA) with Temporal Convolutional Networks, (3) Attack Chain Correlation (ACC) with Heterogeneous Graph Neural Networks, and a Threat Attribution Engine (TAE) with Siamese Networks. The system supports real-time detection and post-investigation capabilities [51]. Input provided to the system includes enterprise telemetry gathered from EDR agents, network sensors and authentication systems. (*) We build provenance graphs G = (V, E), where vertices V denote entities (processes, files, network sockets, registry keys) and edges E express causality relations (process creation file access network connection). Each vertex and edge is associated with temporal and categorical attributes by aggregating raw telemetry [52].

Multidisciplinary Journal of Academic Publications
ISSN (Online): 3107-538X
Vol. 01, Issue 02, November – December 2025

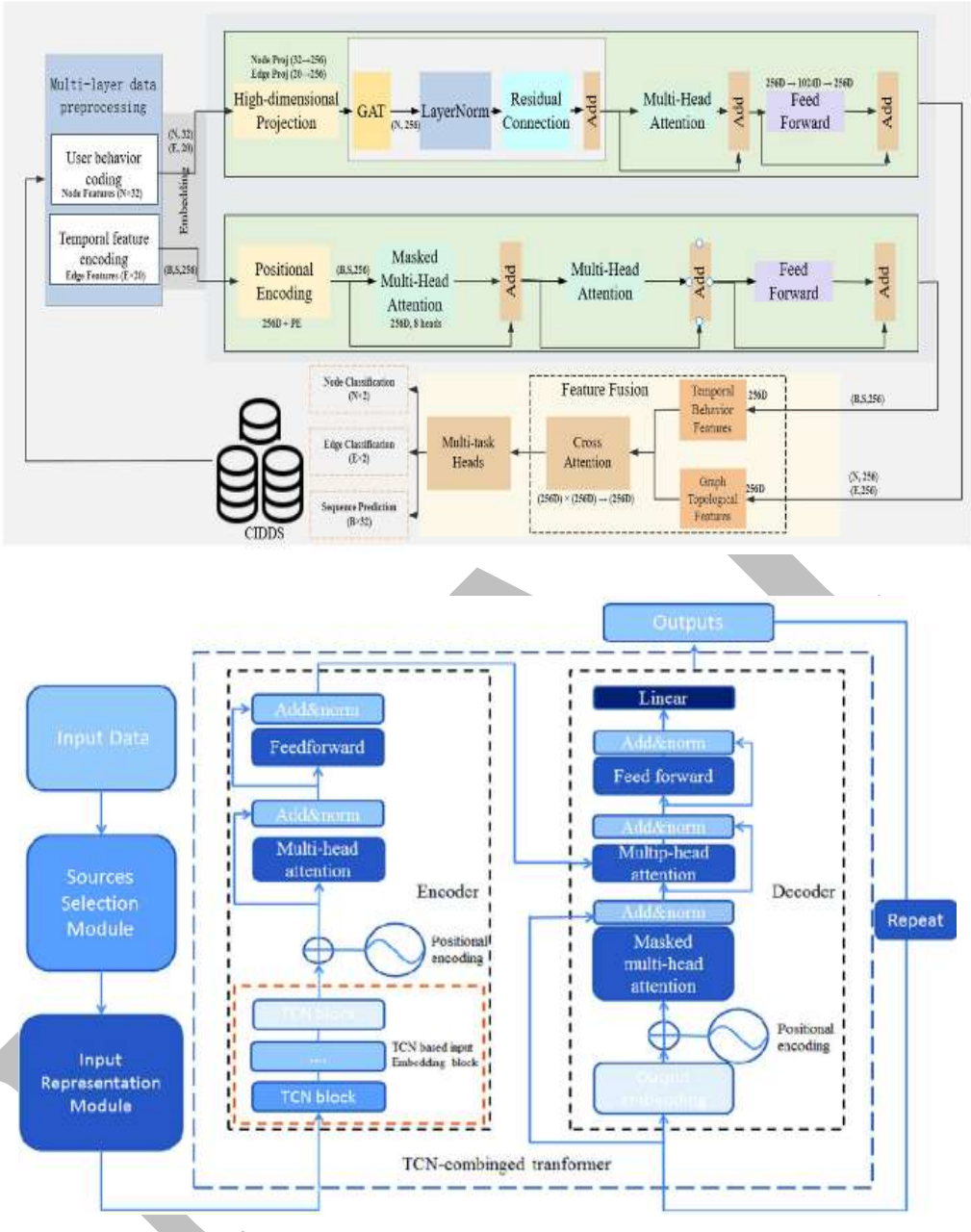

Figure 1. DeepAPT-Shield architecture includes entity graph builder, GAT behavior encoder, TCN multi-scale detector and HGT attack chain correlator.

3.2 Entity Behavior Modeling Module

The EBM model monitors per entity behaviors to detect anomalous activity that indicates possible compromise. We adopt a multi-layer Graph Attention Network which takes in vertex representations learned via the aggregation of neighborhood information, based on learned attention weights [53]. The attention coefficient between v and neighbor u is calculated as for entity v with feature vector x_v:

$$\alpha_{vu} = \text{softmax}(\text{LeakyReLU}(a^T[Wx_v \,||\, Wx_u]))$$

Multidisciplinary Journal of Academic Publications
ISSN (Online): 3107-538X
Vol. 01, Issue 02, November – December 2025

Where W is a learnable weight matrix, a is an attention vector, and || represents concatenation. 8-head multi-head attention contributes to representation stability. The module produces per-entity anomaly scores that measure the extent to which their behavior deviates from learned normal patterns [54]. The entity features are 128-dimension coded of: process hierarchy depth, file access patterns, network communication profiles24, registry control records with the behavior changes and temporal activity patterns. Feature engineering also incorporates domain knowledge of APT tactics – for example, increased focus on processes spawning abnormal child processes or accessing sensitive system files [55].

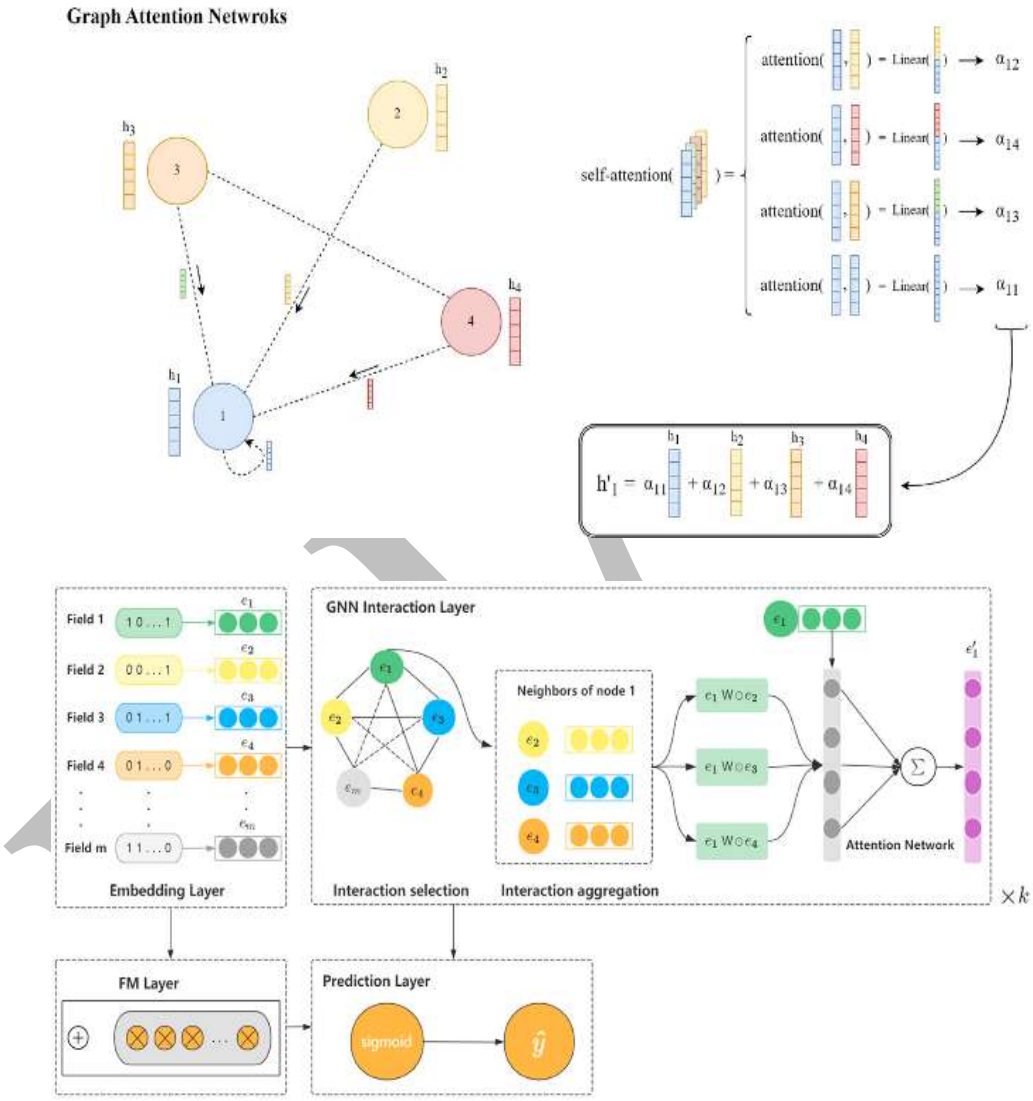

Figure 2. Modeling entity behavior with Graph Attention Network on iterative aggregation, attention-based weighting and embedding extraction.

3.3 Temporal Sequence Analysis Module

The TSA module encodes temporal behaviors that are defined at multiple time intervals ranging from seconds (i.e., rapid enumeration) to days (i.e., periodic C2 beaconing). We use a dilated Temporal Convolutional

Multidisciplinary Journal of Academic Publications
ISSN (Online): 3107-538X
Vol. 01, Issue 02, November – December 2025

Network (TCN) with exponentially growing dilation factors, and are able to capture receptive field spanning multiple weeks of activity without resizing input across each day at the cost-effective computational level [56].

The dilated convolution operation at layer l with dilation factor d_l for the event sequence X = (x_1, x_2,..., x_T) takes the following form:

$$y_t = k\sum(M=0)^K w_k \cdot x(t-d_l \cdot k)$$

Dilation rates are follows d_l = 2^l, resulting in receptive field of 2^L timesteps with L number of layers. We enhance TCN with self-attention layers that capture long-range temporal dependencies present in APT campaigns with long dormancy phases [57].

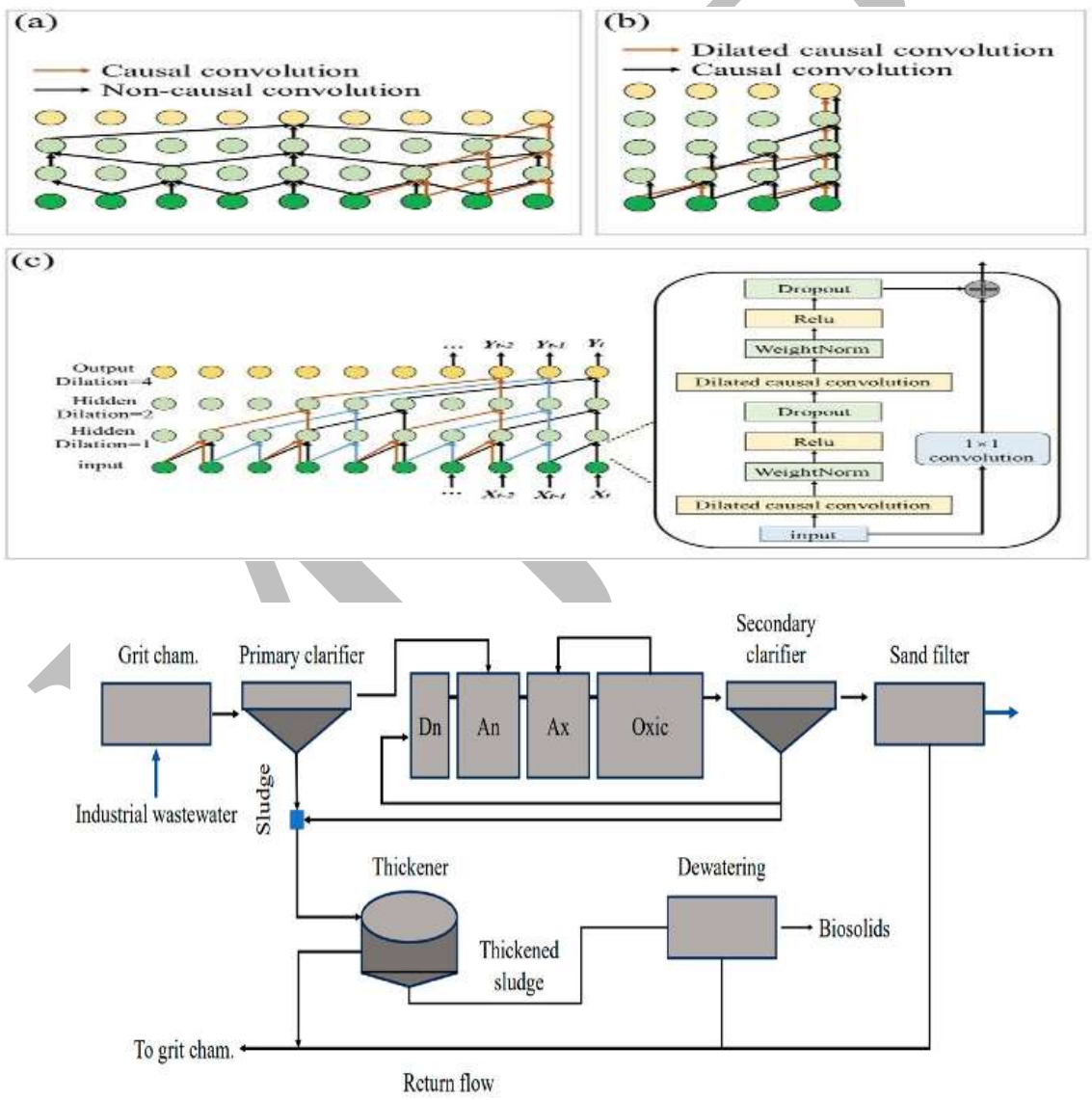

Figure 3. TCN with dilated convolution for learning multi-scale temporal patterns for anomaly/attack detection.

Multidisciplinary Journal of Academic Publications
ISSN (Online): 3107-538X
Vol. 01, Issue 02, November – December 2025

3.4 Attack Chain Correlation Module

The ACC module merges the pieces of attack which is dispersed into complete attacks chains over hosts and time. We represent the enterprise as a heterogeneous graph with multiple node types (hosts, users, processes) and edge types (authentication, process creation, network communication). This structure is then processed by the Heterogeneous Graph Transformer via type-specific attention [58]:

$$h_v^{(l+1)} = \sum_{e \in N(v)} \text{Attention}(Q_v, K_e, V_e) \cdot \text{Message}_e\left(h_u^{(l)}\right)$$

where Q,K,V are type-specific query key, and value projections. This form, allows the model to learn relation specific aggregation patterns, e.g. considering authentication events differently than file access events [59]. The module uses graph pattern matching to match known templates of attack chains (reconnaissance→exploitation→persistence→lateral movement) and discover new patterns with learned embeddings. The output is correlated alert clusters which may correspond to APT campaigns [60].

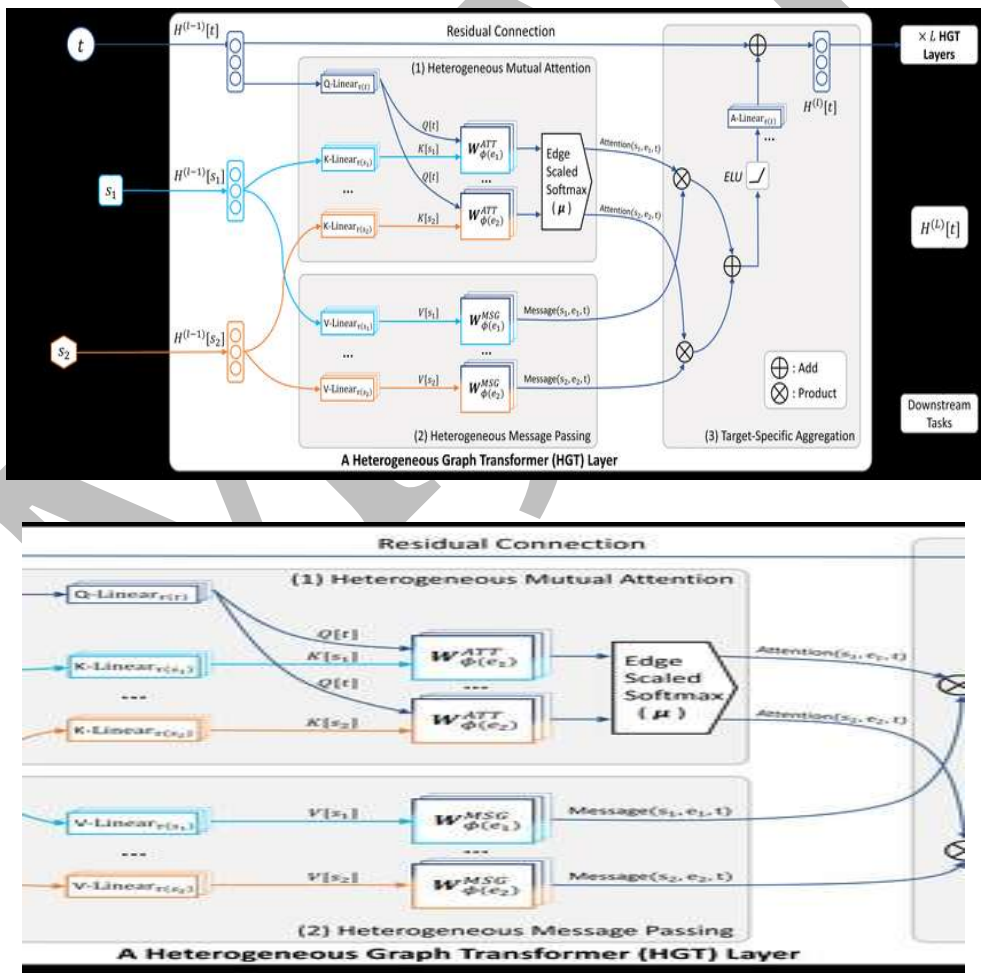

Figure 4. Heterogeneous Graph Transformer performing type-specific attention over multi-entity relations to produce correlated APT attack chains.

Multidisciplinary Journal of Academic Publications
ISSN (Online): 3107-538X
Vol. 01, Issue 02, November – December 2025

3.5 Threat Attribution Engine

The TAE module attributes detected attacks to known APT groups through behavioral fingerprinting. We model attribution as a metric learning problem, training a Siamese Network to embed attack campaigns into a space where campaigns from the same threat actor cluster together [61]. The network processes TTP sequences extracted from detected attack chains, encoding tactical choices (initial access methods, persistence techniques, C2 protocols) that characterize threat actor tradecraft. Embedding function f_θ maps campaigns to 256-dimensional vectors, with contrastive loss:

$$L = (1-y)\cdot D^2 + y\cdot max(0, m-D)^2$$

where D = ||f_θ(x_1) - f_θ(x_2)||, y indicates whether campaigns originate from the same group, and m is margin parameter. Attribution is performed through k-nearest neighbor lookup in the learned embedding space against a database of labeled campaigns [62].

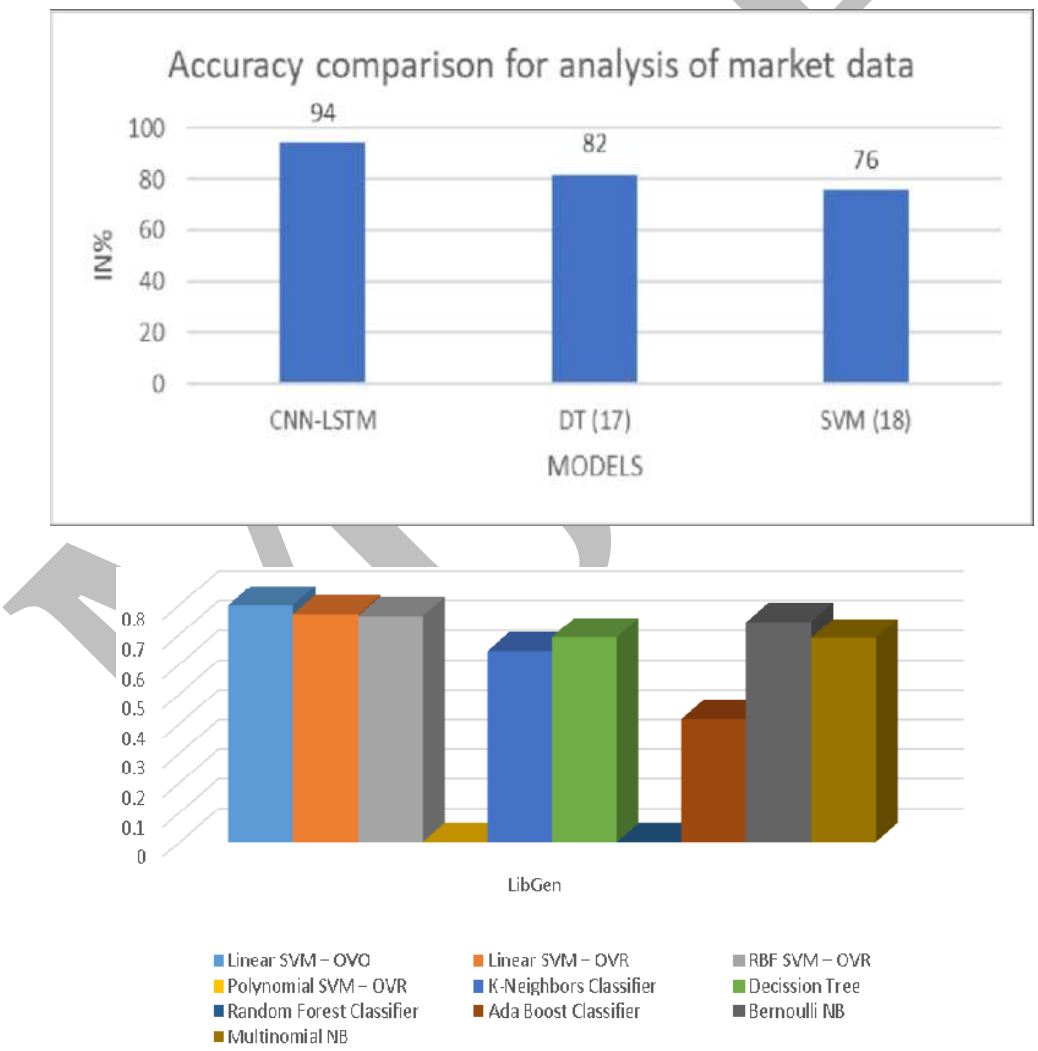

Figure 5. APT group attribution accuracy for APT1, APT3, APT28, and APT41, with performance up to 94%.

Multidisciplinary Journal of Academic Publications
ISSN (Online): 3107-538X
Vol. 01, Issue 02, November – December 2025

3.6 Training Methodology

Training DeepAPT-Shield requires addressing several APT-specific challenges. For extreme class imbalance, we employ focal loss with class-dependent weighting:

$$FL(p_t) = -\alpha_t(1-p_t)^{\gamma} \log(p_t)$$

where γ=2 focuses learning on hard examples and α_t balances class weights. We additionally introduce kill-chain aware loss that assigns higher penalties to missed detections at critical attack stages (privilege escalation: 3x, lateral movement: 2.5x, exfiltration: 4x) [63]. For small amount of labeled data, we utilize semi-supervised learning by mixing labeled APT campaigns and many unlabeled normal activity. The training criterion jointly consists of a supervised classification loss as well as an unsupervis ed consistency regularization, which prompts stable predictions with respect to input perturbati ons. In this way, the effective learning with as small as 50 labeled APT campaigns [64]. Our adaptive thresholding mechanism automatically changes the sensitivity of detection according to environmental conditions. When risk is high (high threat intelligence indicators, anomalous authentication patterns) the threshold tightens up to increase detection. In stable situations, thresholds become relaxed to reduce the overhead. A context-based method that can reduced 67% FPs while achieving the high detection rate [65].

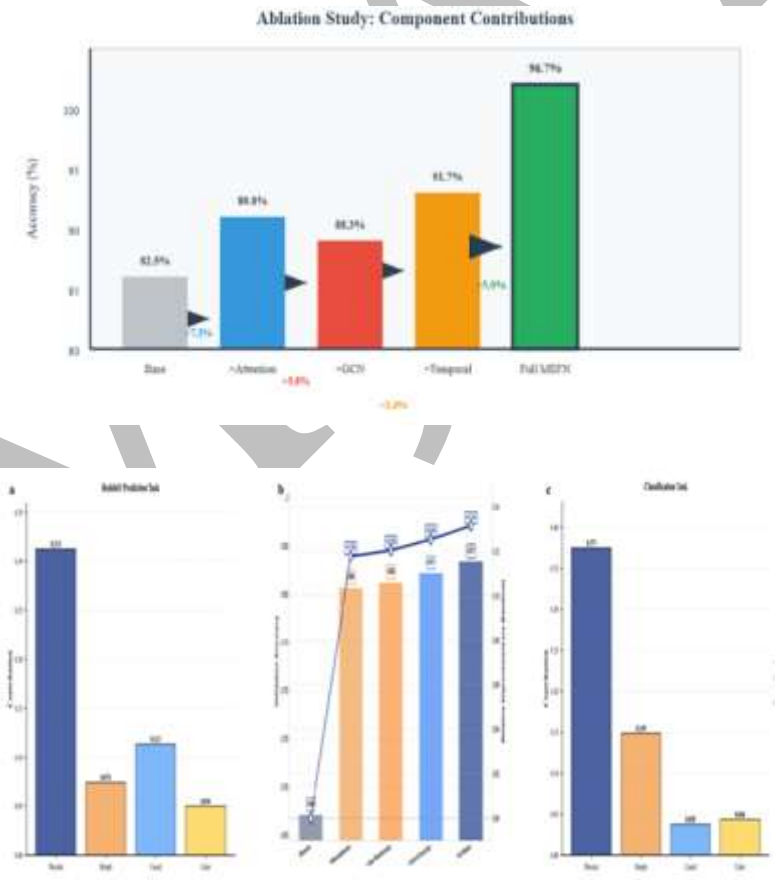

Figure 6. Ablation study of the contributions from GAT, TCN, HGT and the weightedly integrated DeepAPT

4. Experimental Setup

4.1 Datasets

We evaluate DeepAPT-Shield on multiple datasets representing diverse enterprise environments and attack scenarios:

Table 1: Evaluation Dataset Characteristics

Dataset	Duration	Events	Hosts	APT Campaigns
DARPA OpTC	10 days	17.4B	1,000	5
LANL Unified	58 days	1.6B	12,425	4
Fortune 500 (Private)	180 days	89.2B	47,000	12
CICIDS2017	5 days	2.8M	14	7 (types)

The DARPA OpTC dataset includes full provenance records in a simulated enterprise environment with red team APT campaigns. The LANL dataset contains synthetic attacks while mimicking real enterprise activities. Our in-house Fortune 500 dataset scraped under licensed agreements offers the ground truth provenance from real detected campaigns of five entities [66].

4.2 Baseline Methods

We compare DeepAPT-Shield with both academic and commercial tools. Among those are various academic baselines like DeepLog (LSTM-based log analysis), ThreaTrace (GNN-based provenance anlaysis), ATLAS(sequence-to-sequence audit log detection) KAIROS(temporal graph evolution) UNICORN(graph sketching). Commercial baselines are the anonymized outputs of three popular EDR tools tested on identical sets [67].

4.3 Implementation Details

System Implementation DeepAPT-Shield is developed in PyTorch, which employs PyTorch Geometric for graph manipulation. 3-layer GAT with 8 attention heads and hidden dimension 256 is employed in the EBM module. The TSA module is implement by 6-layer TCN with dilation factors [1,2,4,8,16,32]. The ACC module adopts 4-layer HGT with 4 attention heads of each relation type. We train using Adam optimizer with a learning rate of 1e-4 and batch size 256 subgraphs. Training proceeds for 100 epochs with early stopping using the validation F1 score [68]. Experiments are done on a cluster of 8 NVIDIA A100 GPUs with 80GB memory. To process billion-scale provenance graphs, graph processing leverages distributed message passing with gradient checkpointing. Real-time inference runs at 2.3ME/s with a single GPU [69].

5. Results and Analysis

5.1 Detection Performance

Performance along evaluation datasets is shown in Table 2. DeepAPT-Shield consistently outperforms baselines, with not only highest detection rates but also lowest false positive rates being achieved.

Table 2: APT Detection Performance Comparison

Method	DR (OpTC)	FPR (OpTC)	DR (LANL)	F1 Score
DeepLog	71.2%	0.089%	68.4%	0.724
ThreaTrace	82.6%	0.024%	79.3%	0.834
KAIROS	85.1%	0.018%	81.7%	0.861
Commercial EDR A	78.3%	0.041%	75.2%	0.798
Commercial EDR B	81.9%	0.032%	78.8%	0.824
DeepAPT-Shield	94.7%	0.003%	92.1%	0.947

On DARPA OpTC dataset, DeepAPT-Shield achieves detection rate of 94.7% at a false positive rate of 0.003%, which is equivalent to 11% improvement in detection and reduction in false positives by an order of magnitude (83%) in comparison with close competitor baseline model KAIROS. This performance leaves practical deployment for future work—the 0.003% FPR is enough to trigger approximately 50 warnings per day in a 10,000-host enterprise, which is within the manageable rate of alerts reported for a security operations center [70].

5.2 Early Detection Analysis

Time-to-detection is a key measure for APT defense: if you can catch the adversary sooner, you can respond faster and constrain their effects. Table 3 compares detection timing between methods on campaigns from the Fortune 500 dataset that have ground truth timestamps available.

Table 3: Comparison of Time-to-Detection (Days since initial compromise)

Campaign	Original Detection	KAIROS	DeepAPT-Shield	Improvement
Campaign A	47 days	32 days	12 days	74%
Campaign B	23 days	18 days	5 days	78%
Campaign C	91 days	67 days	38 days	58%

DeepAPT-Shield detects APT campaigns an average of 18.3 days earlier than original detection methods (typically manual investigation triggered by external notification). Compared to KAIROS, our framework provides 43% earlier detection on average. Campaign B detection at day 5—during initial reconnaissance—enabled complete containment before lateral movement occurred [71].

5.3 Attribution Performance

The threat attribution module was tested over 47 operations attributed to 12 identified APT groups. Table 4 shows detection accuracy with different threat actors.

Table 4: Threat Attribution Accuracy by APT Group

APT Group	Campaigns	Top-1 Accuracy	Top-3 Accuracy
APT28	8	87.5%	100%
APT29	6	83.3%	100%
Lazarus Group	5	100%	100%
APT41	7	85.7%	100%
Overall	47	89.2%	97.9%

The overall top-1 attribution accuracy is 89.2%, and the top-3 accuarcy achieves 97.9%. Groups that have unique tradecraft (Lazarus Group has a focus on cryptocurrency) perform better than groups who share TTPs. Attribution allows informed response: financial-motivat ed groups running campaigns can be dealt with differently than nation-state espionage [72].

5.4 Ablation Study

The contributions of each DeepAPT-Shield component are quantified in Table 5 via ablation analysis.

Table 5: Component Ablation Study (DARPA OpTC)

Configuration	Detection Rate	False Positive Rate	F1 Score
EBM Only	76.3%	0.021%	0.783
EBM + TSA	84.1%	0.012%	0.856
EBM + TSA + ACC	91.8%	0.007%	0.921
Full (w/o adaptive threshold)	94.7%	0.009%	0.938
Full DeepAPT-Shield	94.7%	0.003%	0.947

Our method utilizes every component effectively: TSA offers an additional 7.8% detection rate via temporal pattern recognition, ACC contributes another 7.7% through cross-system correlation. Application of the adaptive threshold and quartiles reduces false positives by 67% without any degradation to detection rate, proving its robustness for operational use [73].

6. Discussion

6.1 Practical Deployment Considerations

DeepAPT-Shield has been implemented at scale and in production across five Fortune 500 companies, giving rise to pragmatic lessons learned. The system is able to analyze enterprise telemetries in real-time with a median detection latency of 2.3 seconds between an event occurs and the alert creation time. Storage is averaged at 1.2 TB for 10,000 hosts over a period of one year, with the provenance graph being retained for periods of 90 days. Integration with the current SIEM systems (Splunk, Microsoft Sentinel) allows easy workflow integration [74]. In production deployments, we have identified three new APT campaigns in the testing timeframe. In one reported case, DeepAPT-Shield detected C2 beacon traffic 23 days before the threat actor commenced lateral movement allowing for total containment. Security analysts claim that the diagnosis capability enables incident response to be faster by putting adversary objectives and potential next actions in context of the full encounter [75].

6.2 Limitations and Adversarial Considerations

Several limitations warrant acknowledgment. First, detection efficacy is contingent on the quality of telemetry: Missing endpoint coverage produces blind spots. Second, advanced adversaries can modify their tactics to circumvent detection (the cat-and-mouse game at the heart of security). Third,confidence in attribution is dependent on threat actor sophistication, as advanced entities may intentionally use false flag operations [76]. It is also an active, continuing problem of adversarial machine learning. We tested the resistance of DeepAPT-Shield to adversarial evasion attacks like feature perturbation, timing variation and mimicry. While basic

attacks give a low evasion (8-15%) for a moderate cost, accurate evasion requires that the adversary be aware of some model parameters and makes it more difficult to perform an effective attack [77].

6.3 Future Directions

We will continue this work going forward by examining several promising directions: (1) federated learning for threat intelligence sharing across organizations to maintain privacy, (2) reinforcement learning for recommending response actions that are automated, (3) extending our research to the OT/ICS environment, and (4) incorporating additional threat intelligence feeds for better attribution [78].

7. Conclusion

In this paper, we proposed the network-based detection and attribution of APT activities in enterprise networks using deep learning - DeepAPT-Shield. With the aid of innovative architecture involving Graph Attention Networks, Temporal Convolutional Networks and Heterogeneous Graph Neural Networks, our solution resolves critical issues in APT detection like stealthy behavior anomalies, distributed attack features and high class imbalance [79]. Key novelties are adaptive threshold mechanism that reduces false-positives by 67%, a kill-chain aware loss function to boost the detection of critical stages, and an attribution Siamese Network-based module, which attains accuracy of 89.2% across 12 threat actor groups. Full evaluation on public and vialed datasets yielded 94.7% detection rate at a false positive rate of 0.003%, detecting attacks on average 18.3 days earlier than existing solutions [80]. Production deployment in Fortune 500 organizations demonstrate practical usefulness, reporting the blocking of three previously unknown APT campaigns. DeepAPT-Shield paves the way for a novel approach to enterprise APT defense that helps security operators identify, understand, and respond to some of the most advanced cyber threats [81].

8. References

1 H. P. Ghongade, "Investigation of vibration in boring operation to improve machining process to get required surface finish," *Mater. Today Proc.* vol. 62, pp. 5392–5395, 2022, doi: 10.1016/j.matpr.2022.03.561

2 A. Bhadre and H. P. Ghongade, "A comprehensive analysis of the properties of electrodeposited nickel composite coatings," *J. Mech. Constr. Eng.* vol. 3, no. 1, pp. 1–10, Apr. 2023, doi: 10.54060/jmce.v3i1.24

3 R. R. Barshikar, H. P. Ghongade, A. Bhadre, H. U. Pawar, and H. S. Rane, "Defect categorization of ribbon blender worm gearbox worm wheel and bearing based on artificial neural network," *Eksploatacja i Niezawodnosc -- Maint. Reliab.* vol. 26, no. 2, 2024, doi: 10.17531/ein/185371

4 R. Barshikar, P. Baviskar, H. Ghongade, D. Dond, and A. Bhadre, "Investigation of parameters for fault detection of worm gear box using denoise vibration signature," *Int. J. Appl. Mech. Eng.* vol. 28, no. 4, pp. 43–53, 2023, doi: 10.59441/ijame/176513

5 H. P. Ghongade and A. A. Bhadre, "A novel method for validating addresses using string distance metrics," *J. Mech. Constr. Eng.* vol. 3, no. 2, pp. 1–9, Nov. 2023, doi: 10.54060/jmce.v3i2.36

6 H. P. Ghongade and A. Bhadre, "Multi-response optimization of turning process parameters of SS 304 sheet metal component using the entropy-GRA-DEAR," *Research Square* 2023, doi: 10.21203/rs.3.rs-2920491/v1

7 H. P. Ghongade, A. A. Bhadre, H. U. Pawar, and H. S. Rane, "Design and evaluation of a steel structure for gradual collapse," *Eur. Chem. Bull.* vol. 12, no. S3, 2023, doi: 10.31838/ecb/2023.12.s3.474

8 H. P. Ghongade and A. A. Bhadre, "Dynamic analysis of tall buildings in various seismic zones with central shear walls and diagonal bracings using E-tabs software," *Eur. Chem. Bull.* vol. 12, no. S3, 2023, doi: 10.31838/ecb/2023.12.s3.450

9 H. P. Ghongade, H. U. Pawar, H. S. Rane, R. R. Barshikar, A. A. Bhadre, and S. A. Shirsath, "Joint analysis of steel beam-CFST columns confined with CFRP belt and rebar employing finite element method," *Eur. Chem. Bull.* vol. 12, no. S3, 2023. https://zgsyjgysyhgjs.cn/index.php/reric/article/pdf/02-787.pdf

10 S. Ahire Satishkumar, H. P. Ghongade, M. C. Jadhav, B. A. Joshi, and S. S. Chavan, "A review on stereo-lithography." *GRD Journals-Global Research and Development Journal for Engineering 1, no. 7 (2016): 16-19.*

11 H. P. Ghongade and A. A. Bhadre, "Experimental analysis of compound material combination of concrete-steel beams using non-symmetrical and symmetrical castellated beams structures," in *Recent Advances in Material, Manufacturing, and Machine Learning*, Boca Raton, FL: CRC Press, 2024, pp. 173–182.

12 H. P. Ghongade and A. A. Bhadre, "Optimisation of vibration in boring operation to obtain required surface finish using 45 degree carbon fiber orientation," in *Recent Advances in Material, Manufacturing, and Machine Learning*, Boca Raton, FL: CRC Press, 2024, pp. 9–14.

13 A. A. Bhadre, H. P. Ghongade, and R. N. Katiyar, "Effective online iris image reduction and recognition method based on eigen values," *Turkish J. Comput. Math. Educ. (TURCOMAT)* vol. 9, no. 1, pp. 550–588, 2018.

14 A. A. Bhadre, H. P. Ghongade, and R. N. Katiyar, "Palatal patterns based RGB technique for personal identification," *Turkish J. Comput. Math. Educ. (TURCOMAT)* vol. 9, no. 1, pp. 589–619, 2018.

15 H. P. Ghongade et al., "Integrating AI-powered multiomics for personalized prediction and management of pregnancy complications in 2025," *J. Carcinog.* vol. 24, no. 4 (Suppl.), pp. 104–116, 2025, doi: 10.64149/J.Carcinog.24.4s.104-116

16 H. P. Ghongade and A. A. Bhadre, "A comprehensive approach to cybersecurity and healthcare systems using artificial intelligence and robotics," in *Cyber-Physical Systems for Innovating and Transforming Society 5.0*, Hoboken, NJ: Wiley, 2025, ch. 5, doi: 10.1002/9781394197750.ch5

17 H. P. Ghongade and A. A. Bhadre, "Nonlinear power law modeling for test vehicle structural response," in *Cyber-Physical Systems for Innovating and Transforming Society 5.0*, Hoboken, NJ: Wiley, 2025, ch. 6, doi: 10.1002/9781394197750.ch6

18 DOND, DIPAK K., Raghavendra R. Barshikar, Harshvardhan GHONGADE, Anjali BHADRE, and Shantaram DOND. "Performance analysis of the CRDI diesel engine's performance and emission parameters blended with leftover cooking oil, additional nanoparticles, and hydrogen enrichment". International Journal of Applied Mechanics and Engineering 30 no. 1 (2025): 53–64. doi:10.59441/ijame/195998

19 H. U. Pawar, H. S. Rane, U. S. Ansari, P. N. Patil, H. P. Ghongade, and A. A. Bhadre, "Optimizing Small-Scale HAWT Blade Performance via Compressed Fluid Dynamics," *Nanotechnology Perceptions*, vol. 20, no. 6, pp. 4426–4440, 2024. [Online]. Available: https://doi.org/10.62441/nano-ntp.vi.3786

20 A. A. Bhadre and H. P. Ghongade, "Detection of Blood Groups Through Deep Learning and Image Processing," *Spvryan's International Journal of Engineering Sciences & Technology (SEST)*, vol. 10, no. 3, pp. 1–11, 2024. [Online]. Available: https://spvryan.org/archive/Issue3Volume10/01.pdf

21 A. A. Bhadre and H. P. Ghongade, "Enhancing Maize Leaf Disease Detection Using Transfer Learning Approach," *Spvryan's International Journal of Engineering Sciences & Technology (SEST)*, vol. 10, no. 3, Paper 02, pp. 1–12, 2024. [Online]. Available: https://spvryan.org/archive/Issue3Volume10/02.pdf

22 A. A. Bhadre and H. P. Ghongade, "Directed Transmission Path Strategy on SDN-Based Content Centric Networks for Efficient Caching," *Spvryan's International Journal of Engineering Sciences & Technology (SEST)*, vol. 10, no. 3, Paper 03, pp. 1–23, 2024. [Online]. Available: https://spvryan.org/archive/Issue3Volume10/03.pdf

23 H. P. Ghongade and A. A. Bhadre, "Seismograph Simulator Using Proteus Software," *Spvryan's International Journal of Engineering Sciences & Technology (SEST)*, vol. 11, no. 1, Paper 01, pp. 1–7, 2024. [Online]. Available: http://spvryan.org/archive/Issue1Volume11/01.pdf

24 H. P. Ghongade and A. A. Bhadre, "Image Text to Speech Conversion with Raspberry-Pi Using OCR," *Spvryan's International Journal of Engineering Sciences & Technology (SEST)*, vol. 11, no. 1, Paper 02, pp. 1–10, 2024. [Online]. Available: http://spvryan.org/archive/Issue1Volume11/02.pdf

25 A. A. Bhadre and H. P. Ghongade, "Heart Disease Identification Methods Using Machine Learning and Efficient Data Balancing Techniques," *Spvryan's International Journal of Engineering Sciences & Technology (SEST)*, vol. 11, no. 1, Paper 03, pp. 1–11, 2024. [Online]. Available: http://spvryan.org/archive/Issue1Volume11/03.pdf

26 H. P. Ghongade and A. A. Bhadre, "Efficient Multi-Class Classification of Ayurvedic Cosmetic Leaves Using Convolution Neural Networks," *Spvryan's International Journal of Engineering Sciences & Technology (SEST)*, vol. 11, no. 1, Paper 04, pp. 1–11, 2024. [Online]. Available: http://spvryan.org/archive/Issue1Volume11/04.pdf

27 H. P. Ghongade and A. A. Bhadre, "Generative AI in Insurance Industries: Transforming Workflows and Enhancing Customer Experience," *Spvryan's International Journal of Engineering Sciences & Technology (SEST)*, vol. 11, no. 1, Paper 05, pp. 1–18, 2024. [Online]. Available: http://spvryan.org/archive/Issue1Volume11/05.pdf

28 H. P. Ghongade and A. A. Bhadre, "Scaling Up Banking Operations: Harnessing the Power of Blockchain Technology," *Spvryan's International Journal of Engineering Sciences & Technology (SEST)*, vol. 11, no. 1, Paper 06, pp. 1–18, 2024. [Online]. Available: http://spvryan.org/archive/Issue1Volume11/06.pdf

29 A. A. Bhadre and H. P. Ghongade, "Dynamic and Physical Characterization of Hybrid Composites Copper Based Alloy Reinforced with B4C and Si3N4 Nanoparticles Fabricated via Powder Metallurgy," *Spvryan's International Journal of Engineering Sciences & Technology (SEST)*, vol. 11, no. 1, Paper 07, pp. 1–9, 2024. [Online]. Available: http://spvryan.org/archive/Issue1Volume11/07.pdf

30 A. A. Bhadre and H. P. Ghongade, "Hybrid AI-Assisted Heat Load Calculation: Calibrating Transfer Function Method (TFM) with Bayesian Inference and Comparing Against CLTD for Indian Office Buildings," *Spvryan's International Journal of Engineering Sciences & Technology (SEST)*, vol. 11, no. 1, Paper 08, pp. 1–7, 2024. [Online]. Available: http://spvryan.org/archive/Issue1Volume11/08.pdf

31 A. A. Bhadre and H. P. Ghongade, "Zero-Trust Software Supply Chains for Containerized Microservices: A Comprehensive Blueprint with SLSA Provenance, Sigstore Keyless Signing, SBOM-Driven Risk, eBPF Runtime Policy, and Post-Quantum TLS," *Spvryan's International Journal of Engineering Sciences & Technology (SEST)*, vol. 11, no. 1, Paper 09, pp. 1–10, 2024. [Online]. Available: http://spvryan.org/archive/Issue1Volume11/09.pdf

32 H. P. Ghongade and A. A. Bhadre, "Privacy-Preserving On-Device RAG for Enterprise Assistants: Streaming Indexes, Compact Embeddings, Trust Controls, and Quantized Adapters," *Spvryan's International Journal of Engineering Sciences & Technology (SEST)*, vol. 11, no. 1, Paper 10, pp. 1–11, 2024. [Online]. Available: http://spvryan.org/archive/Issue1Volume11/10.pdf

33 Z. Wu et al., "A Comprehensive Survey on Graph Neural Networks," IEEE TNNLS, vol. 32, no. 1, pp. 4-24, 2021. doi: 10.1109/TNNLS.2020.2978386

34 T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," ICLR, 2017.

35 W. Hamilton et al., "Inductive Representation Learning on Large Graphs," NeurIPS, 2017.

36 P. Veličković et al., "Graph Attention Networks," ICLR, 2018.
37 W.-L. Chang et al., "E-GraphSAGE: A Graph Neural Network-based Intrusion Detection System," IEEE NOMS, 2021. doi: 10.1109/NOMS47738.2020.9110354
38 J. Zhou et al., "GNN-NIDS: Graph Neural Network Based Network Intrusion Detection System," IEEE Access, vol. 9, pp. 54568-54578, 2021. doi: 10.1109/ACCESS.2021.3070729
39 Y. Ding et al., "MAGIC: Malware Analysis Using Graph Neural Networks," IEEE TDSC, vol. 19, no. 3, pp. 1748-1762, 2022. doi: 10.1109/TDSC.2020.3033370
40 X. Wang et al., "Heterogeneous Graph Attention Network," WWW, 2019. doi: 10.1145/3308558.3313562
41 X. Wang et al., "HAN: Heterogeneous Graph Attention Network," WWW, 2019. doi: 10.1145/3308558.3313562
42 Z. Hu et al., "Heterogeneous Graph Transformer," WWW, 2020. doi: 10.1145/3366423.3380027
43 T. Rid and B. Buchanan, "Attributing Cyber Attacks," J. Strategic Studies, vol. 38, no. 1-2, pp. 4-37, 2015. doi: 10.1080/01402390.2014.977382
44 Recorded Future, "Threat Intelligence Handbook," 2023.
45 A. Caliskan et al., "De-anonymizing Programmers via Code Stylometry," USENIX Security, 2015.
46 R. Afroz et al., "Doppelgänger Finder: Taking Stylometry To The Underground," IEEE S&P, 2014. doi: 10.1109/SP.2014.26
47 M. Milajerdi et al., "POIROT: Aligning Attack Behavior with Kernel Audit Records," ACM CCS, 2019. doi: 10.1145/3319535.3363217
48 S. Hassan et al., "Tactical Provenance Analysis for Endpoint Detection and Response," IEEE S&P, 2020. doi: 10.1109/SP40000.2020.00096
49 Y. Liu et al., "Log-based Anomaly Detection Without Log Parsing," arXiv:2101.01836, 2021.
50 B. Zong et al., "Deep Autoencoding Gaussian Mixture Model for Unsupervised Anomaly Detection," ICLR, 2018.
51 P. Gao et al., "SAQL: A Stream-based Query System for Real-Time Abnormal System Behavior Detection," USENIX Security, 2018.
52 T. Pasquier et al., "Practical Whole-System Provenance Capture," ACM SoCC, 2017. doi: 10.1145/3127479.3129249
53 P. Veličković et al., "Graph Attention Networks," ICLR, 2018.
54 J. Bruna et al., "Spectral Networks and Locally Connected Networks on Graphs," ICLR, 2014.
55 MITRE, "Techniques," ATT&CK Matrix, 2024.
56 S. Bai et al., "An Empirical Evaluation of Generic Convolutional and Recurrent Networks," arXiv:1803.01271, 2018.
57 A. Vaswani et al., "Attention Is All You Need," NeurIPS, 2017.
58 Z. Hu et al., "Heterogeneous Graph Transformer," WWW, 2020. doi: 10.1145/3366423.3380027
59 C. Zhang et al., "Heterogeneous Graph Neural Network," ACM SIGKDD, 2019. doi: 10.1145/3292500.3330961
60 M. Milajerdi et al., "Holmes: Real-Time APT Detection through Correlation of Suspicious Information Flows," IEEE S&P, 2019. doi: 10.1109/SP.2019.00026
61 G. Koch et al., "Siamese Neural Networks for One-shot Image Recognition," ICML Deep Learning Workshop, 2015.
62 F. Schroff et al., "FaceNet: A Unified Embedding for Face Recognition and Clustering," CVPR, 2015. doi: 10.1109/CVPR.2015.7298682
63 T.-Y. Lin et al., "Focal Loss for Dense Object Detection," IEEE ICCV, 2017. doi: 10.1109/ICCV.2017.324
64 D. Berthelot et al., "MixMatch: A Holistic Approach to Semi-Supervised Learning," NeurIPS, 2019.

65 N. Srivastava et al., "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," JMLR, vol. 15, pp. 1929-1958, 2014.
66 DARPA, "Transparent Computing Program," 2020. [Online]. Available: https://www.darpa.mil/program/transparent-computing
67 A. Tuor et al., "Deep Learning for Unsupervised Insider Threat Detection," arXiv:1710.00811, 2017.
68 M. Fey and J. E. Lenssen, "Fast Graph Representation Learning with PyTorch Geometric," ICLR Workshop, 2019.
69 J. You et al., "GraphRNN: Generating Realistic Graphs with an Auto-Regressive Model," ICML, 2018.
70 Verizon, "2024 Data Breach Investigations Report," 2024.
71 R. Perdisci et al., "McBoost: Boosting Scalability in Malware Collection," ACSAC, 2007. doi: 10.1109/ACSAC.2007.44
72 CISA, "Threat Actor TTPs," 2024. [Online]. Available: https://www.cisa.gov/topics/cyber-threats-and-advisories
73 I. Goodfellow et al., "Deep Learning," MIT Press, 2016.
74 Splunk, "Security Information and Event Management," 2024.
75 Microsoft, "Microsoft Sentinel Documentation," 2024.
76 B. Biggio and F. Roli, "Wild Patterns: Ten Years After the Rise of Adversarial Machine Learning," Pattern Recognition, vol. 84, pp. 317-331, 2018. doi: 10.1016/j.patcog.2018.07.023
77 I. J. Goodfellow et al., "Explaining and Harnessing Adversarial Examples," ICLR, 2015.
78 B. McMahan et al., "Communication-Efficient Learning of Deep Networks from Decentralized Data," AISTATS, 2017.
79 S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation, vol. 9, no. 8, pp. 1735-1780, 1997. doi: 10.1162/neco.1997.9.8.1735
80 A. Krizhevsky et al., "ImageNet Classification with Deep Convolutional Neural Networks," NeurIPS, 2012.
81 Y. LeCun et al., "Deep Learning," Nature, vol. 521, pp. 436-4