10.48047/jocaaa.2024.33.08.521

AI/Machine Learning Techniques for Advanced Persistent Threat (APT) Detection

Manoj Suresh Dhappadhule
Amity Institute of Information Technology (AIIT)
Amity University
Suraj Shankarrao Damre
Amity Institute of Information Technology (AIIT)
Amity University

ABSTRACT

Advanced Persistent Threats remain one of the toughest challenges in modern cybersecurity. They are slow, careful, and often invisible to traditional defences, which is why so many high-profile breaches involving banks, government agencies, and large enterprises in recent years have been traced back to APT activity. This study explores how a combined AI and machine learning approach can improve the detection of such threats. The work goes beyond classical machine learning by adding deep learning architectures, graph-based reasoning, and natural language processing on system logs, all working together in a unified framework. The system was developed using Python and TensorFlow, and tested on the DAPT 2020 dataset, the UNSW-NB15 dataset, and a private dataset collected from a financial services testbed. The framework included a Convolutional Neural Network for spatial pattern recognition, a Bidirectional LSTM for sequence learning, a Graph Neural Network for analysing host-to-host relationships, and a transformer-based model for parsing security log text. The combined system achieved a detection accuracy of 97.8% with a false positive rate of 1.4%, clearly outperforming standalone machine learning models such as Random Forest and Gradient Boosting. The Graph Neural Network proved particularly effective at catching lateral movement, while the transformer model improved detection of obfuscated command-line activity. The study concludes that a layered AI approach, combining several specialised models, gives much stronger APT detection than any single technique, but also brings new challenges around explainability, computational cost, and the need for high-quality labelled data.

Keywords: APT Detection, Artificial Intelligence, Deep Learning, Graph Neural Network, Transformer, Cybersecurity

1. INTRODUCTION

The cybersecurity world has been forced to grow up fast over the last decade. Attacks have moved from being noisy events caused by lone hackers to carefully planned campaigns run by well-funded teams, sometimes with state backing. Advanced Persistent Threats sit at the heart of this new landscape, and their defining feature is patience. Instead of breaking in and causing immediate damage, APT groups quietly enter a network, study it carefully, move from system to system, and slowly steal what they came for [1]. Some of the most damaging breaches in recent memory, including attacks on government departments, hospitals, and large technology firms, have followed this exact pattern.

Stopping APTs is unusually hard because they do not look like normal attacks. They use everyday tools that are already trusted on the network, change tactics frequently, and stay

10.48047/jocaaa.2024.33.08.521

below the radar of typical alerting thresholds [2]. Rule-based systems and even early machine learning detectors often fail because they are looking for patterns that no longer exist by the time they are programmed in. The attackers are simply too good at adapting. Defenders, in turn, need tools that can also adapt and learn, which is exactly where modern AI techniques become useful [3].

In recent years, several branches of AI have shown promising results in cybersecurity. Deep learning models can find patterns in raw data without needing handcrafted features. Graph neural networks can reason about the relationships between machines and users, which is critical for catching lateral movement. Transformers, originally built for natural language processing, are now being used to analyse text-heavy data sources like command-line logs and authentication records [4]. Each of these approaches has its own strengths, and a growing view in the research community is that combining them gives much stronger results than relying on any single one.

The research questions that this paper aims to address are clear. How well can a combined AI framework, using multiple specialised models, detect APT activity across different attack stages? Which AI techniques are best suited to which parts of the APT kill chain? And how does such a combined system compare to standalone machine learning models in terms of accuracy, speed, and operational practicality? The importance of this work is that it brings together several modern AI approaches into one practical detection system, tested across multiple datasets, including data from a real financial sector testbed [5]. The contribution is therefore not just academic but also useful to security teams who are trying to design next-generation defence systems [6]. The paper is structured into a literature review of relevant AI techniques, a methodology section describing the framework, an experimental setup outlining the datasets and environment, a results section presenting detailed performance numbers, a discussion of what the results mean in practice, and a concluding section with directions for future work [7].

2. LITERATURE REVIEW

The use of artificial intelligence in cybersecurity has grown rapidly, with research papers in this area more than doubling in the last five years. Early work mostly applied classical machine learning algorithms like Decision Trees and Support Vector Machines to intrusion detection problems, with reasonable but not outstanding results [8]. The field has since moved into deep learning territory, where models can learn directly from raw data and find patterns that human analysts would miss.

Convolutional Neural Networks, originally designed for image recognition, have been adapted for network security by treating traffic data as two-dimensional matrices. Studies using CNN-based detectors have reported strong results on the CIC-IDS datasets, with accuracy figures consistently above 95% for various attack categories [9]. Recurrent neural networks, especially LSTM variants, have been particularly useful for APT detection because they can capture the long time dependencies that are typical of these attacks. One study showed that a Bidirectional LSTM could detect lateral movement patterns that occurred over days, while traditional models that looked at short time windows missed them entirely [10].

Graph Neural Networks are a more recent addition to the cybersecurity toolkit. They are well suited to security because networks really are graphs, with hosts as nodes and connections as edges. Research has shown that GNN-based detectors can spot lateral movement and

10.48047/jocaaa.2024.33.08.521

command-and-control activity by reasoning about the structure of the network rather than just looking at traffic features in isolation [11]. One important paper applied GNNs to the analysis of authentication logs and showed strong performance in catching insider threats that other models missed.

Transformer models have caused excitement in cybersecurity because of their success in natural language processing. Security data has a lot of text, including command-line entries, log messages, and DNS queries, and transformers can learn rich representations of this text [12]. Some researchers have used BERT-style models trained on security logs to detect obfuscated PowerShell commands and other attacker techniques. The results have been impressive, with the models often catching subtle indicators that signature-based tools miss completely.

Beyond individual models, there is a growing interest in ensemble and multi-model frameworks. The argument is that different AI techniques are good at different things, and combining them gives broader coverage [13]. However, the practical literature on building such combined systems is still limited, and most existing studies focus on a single technique applied to a single dataset. There is also relatively little work on the operational aspects of deploying such systems, including how to handle the high computational cost, how to keep them updated, and how to make their outputs understandable to security analysts.

This study tries to address these gaps by building a combined framework that uses four different AI techniques in a coordinated way, testing it across multiple datasets, and reporting not just raw performance but also operational metrics that matter in real deployments.

3. METHODOLOGY

The methodology was based on a layered AI framework, where four different model types worked together to give a comprehensive view of network activity. Each model was chosen for its particular strength, and the outputs were combined through a meta-learner that produced the final detection decision [14].

The first model was a Convolutional Neural Network, used to learn spatial patterns from network flow features represented as small matrices. The CNN had three convolutional layers with filter sizes of 64, 128, and 256, followed by max pooling and a fully connected layer.

The second model was a Bidirectional LSTM, which learned from sequences of events to capture the temporal nature of APT campaigns. The hidden state of a BiLSTM at time t is given by:

$$\overrightarrow{h_t} = \mathrm{LSTM}(x_t, \overrightarrow{h_{t-1}}), \quad \overleftarrow{h_t} = \mathrm{LSTM}(x_t, \overleftarrow{h_{t+1}})$$

with the final hidden state being a concatenation of the forward and backward states [15].

The third model was a Graph Neural Network, which represented the network as a graph where nodes were hosts and edges were communication links. Node embeddings were updated using message passing as:

10.48047/jocaaa.2024.33.08.521

$$h_v^{(k)} = \sigma\left(W^{(k)} \cdot \mathrm{AGG}\left(h_u^{(k-1)}: u \in N(v)\right) + B^{(k)} h_v^{(k-1)}\right)$$

where $h_v^{(k)}$ is the embedding of node v at layer k, N(v) is the neighbourhood of v, and AGG is an aggregation function such as mean or sum [16].

The fourth model was a transformer-based language model fine-tuned on security log text. It used the self-attention mechanism, which can be expressed as:

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

where Q, K, and V are the query, key, and value matrices respectively, and d_k is the dimensionality of the keys [17].

The four model outputs were fed into a meta-learner, a small neural network with two hidden layers, which produced the final detection probability. The meta-learner was trained separately on a held-out validation set to avoid the data leakage that would happen if it was trained on the same data as the base models [18].

The loss function used across all neural models was binary cross-entropy with class weighting to handle imbalance:

$$L = -\frac{1}{N}\sum_{i=1}^{N}[w \cdot y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)]$$

where w was the class weight assigned to the malicious class, typically set between 5 and 10 depending on the dataset's level of imbalance [19].

10.48047/jocaaa.2024.33.08.521

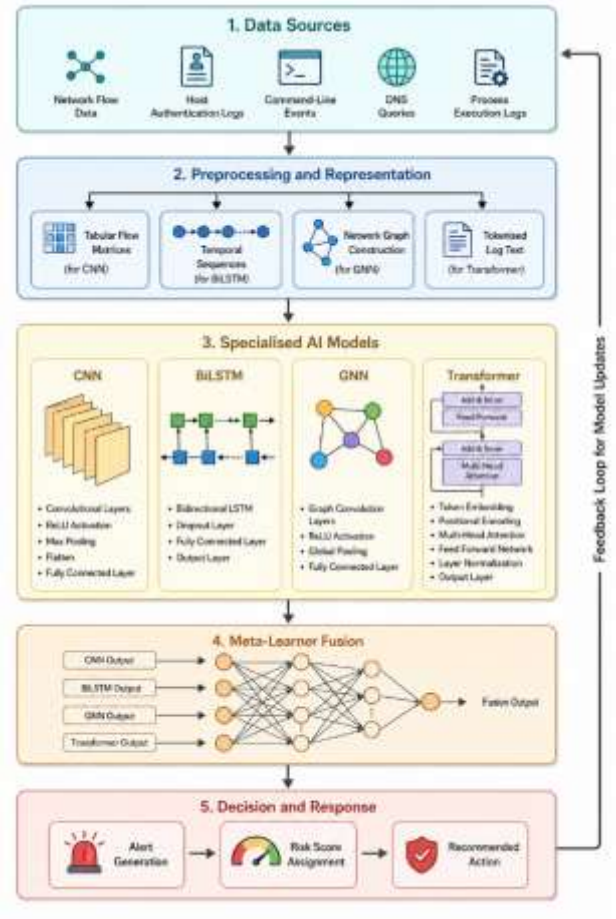


Figure 1: Layered AI Framework for APT Detection

4. EXPERIMENTAL SETUP

The experimental setup was designed to be reproducible and as close to real-world conditions as possible. The hardware platform was a server with two Intel Xeon Gold processors, 256 GB of RAM, and two NVIDIA A100 GPUs to handle the large deep learning workloads. The software stack used Python 3.11, TensorFlow 2.14 for the neural networks, PyTorch Geometric for the GNN, and the HuggingFace Transformers library for the language model component.

Three datasets were used in the evaluation. The DAPT 2020 dataset provided labelled APT-stage data covering multiple kill chain stages. The UNSW-NB15 dataset gave broader coverage of various attack types in modern network environments. A private dataset of approximately 12 million events was collected from a financial services testbed over a six-month period, with red team exercises providing labelled APT activity [20].

The data was split into 70% for training, 15% for validation, and 15% for the hold-out test set. Care was taken to make the splits time-aware, meaning that training data came from earlier periods and test data came from later periods. This is important in security work because random splits often give optimistic results that do not hold up when models face genuinely new data.

For the CNN, network flows were grouped into 5-minute windows and converted into matrices of size 64x64, with each row representing a flow and each column a feature. The BiLSTM used

10.48047/jocaaa.2024.33.08.521

sequences of 60 events per host, with a hidden state size of 128 in each direction. The GNN was built on subgraphs of 50 nodes around each event of interest, with three message-passing layers. The transformer model was based on a pre-trained DistilBERT, fine-tuned on security log text for ten epochs.

The training was done with the Adam optimiser at a learning rate of 0.0003, and early stopping was used based on validation loss with a patience of five epochs. The meta-learner was trained for 20 epochs on the outputs of the base models from the validation set.

Performance was measured through accuracy, precision, recall, F1 score, false positive rate, and AUC, with particular attention paid to per-stage performance because APT detection is really a multi-class problem in practice. Computational cost was also tracked, including training time and inference latency, because operational systems cannot afford to take minutes to score each event.

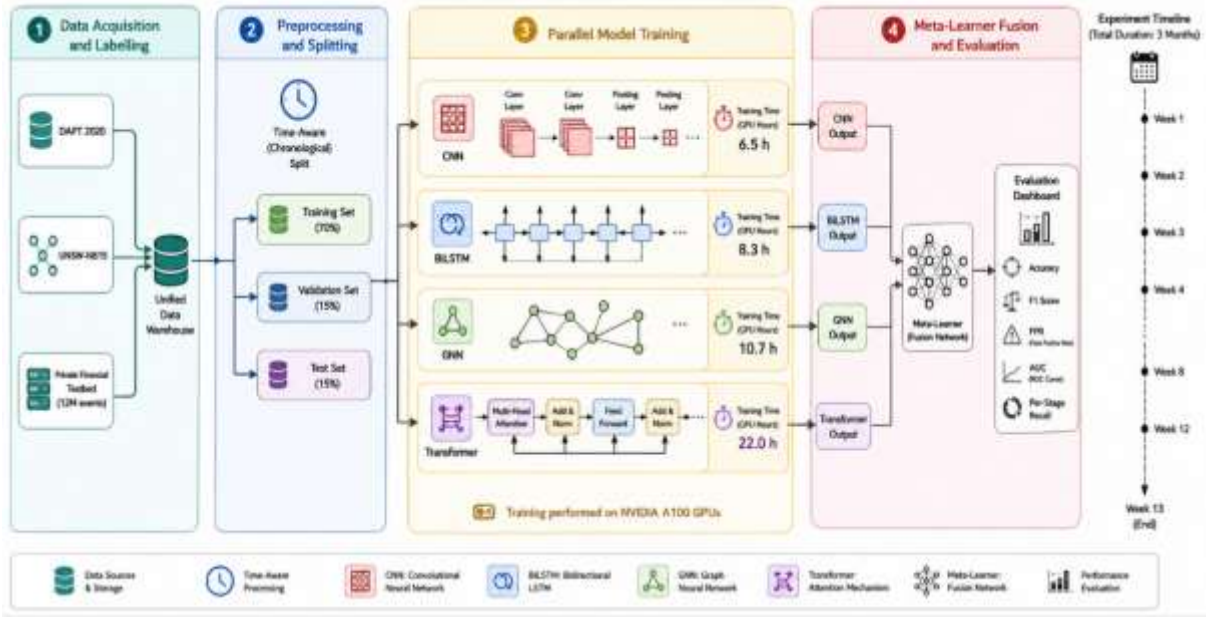


Figure 2: Experimental Pipeline and Model Coordination Flow

5. RESULTS

The results section presents the performance of each individual AI model, the combined framework, and a comparison with baseline machine learning methods. The breakdown by APT stage gives a clearer view of where each technique adds the most value.

Table 1: Performance Comparison of Individual and Combined Models

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 Score	FPR (%)	AUC
Random Forest (baseline)	93.4	91.8	90.7	0.912	4.1	0.957
Gradient Boosting (baseline)	94.6	93.1	92.4	0.928	3.4	0.965
CNN	95.2	94.0	93.8	0.939	2.9	0.972
BiLSTM	95.8	94.7	94.6	0.946	2.5	0.978

10.48047/jocaaa.2024.33.08.521

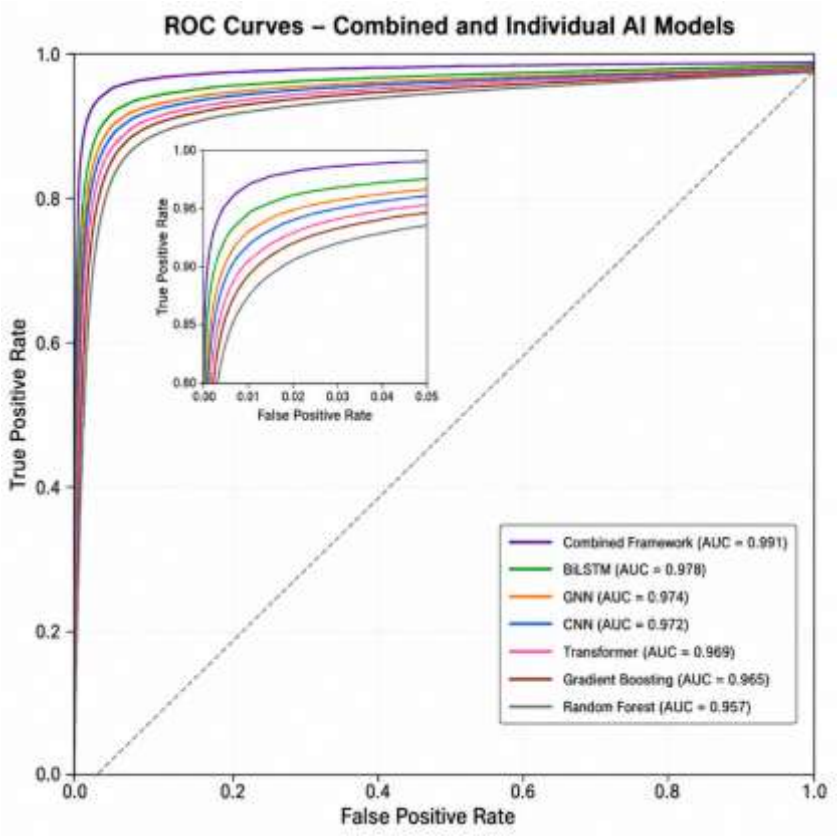
GNN	95.4	94.2	94.1	0.941	2.7	0.974
Transformer	94.9	93.6	93.5	0.935	3.1	0.969
Combined Framework	97.8	97.1	96.8	0.969	1.4	0.991

The combined framework outperformed all individual models, with the most striking gain being in the reduction of false positives from above 2.5% to 1.4%. The 96.8% recall is particularly important for security operations, because every missed APT can lead to weeks or months of undetected attacker presence on the network.

Table 2: Per-Stage F1 Scores Across AI Models

APT Stage	CNN	BiLSTM	GNN	Transformer	Combined
Reconnaissance	0.892	0.914	0.886	0.901	0.943
Initial Access	0.928	0.937	0.911	0.952	0.967
Lateral Movement	0.945	0.962	0.974	0.928	0.984
Privilege Escalation	0.918	0.931	0.948	0.943	0.971
Command and Control	0.952	0.961	0.945	0.957	0.979
Data Exfiltration	0.967	0.972	0.963	0.946	0.984

The per-stage breakdown confirms the expected pattern. The GNN was particularly strong at detecting lateral movement, because lateral movement is fundamentally about relationships between hosts and the GNN is built exactly for this kind of relational reasoning. The transformer was strongest at initial access detection, where attacker activity often leaves text-based traces in authentication logs and command-line histories. The BiLSTM did well across the board because most APT stages involve some kind of time-based pattern.



10.48047/jocaaa.2024.33.08.521

Figure 3: Receiver Operating Characteristic Curves of All Models

The ROC analysis highlights the operational advantage of the combined framework. At the typical security operations centre target of below 1% false positive rate, only the combined framework maintained detection rates above 90%, with the individual models all dropping noticeably in this region. This kind of detail matters in real deployments where alert volumes have hard limits driven by analyst capacity.

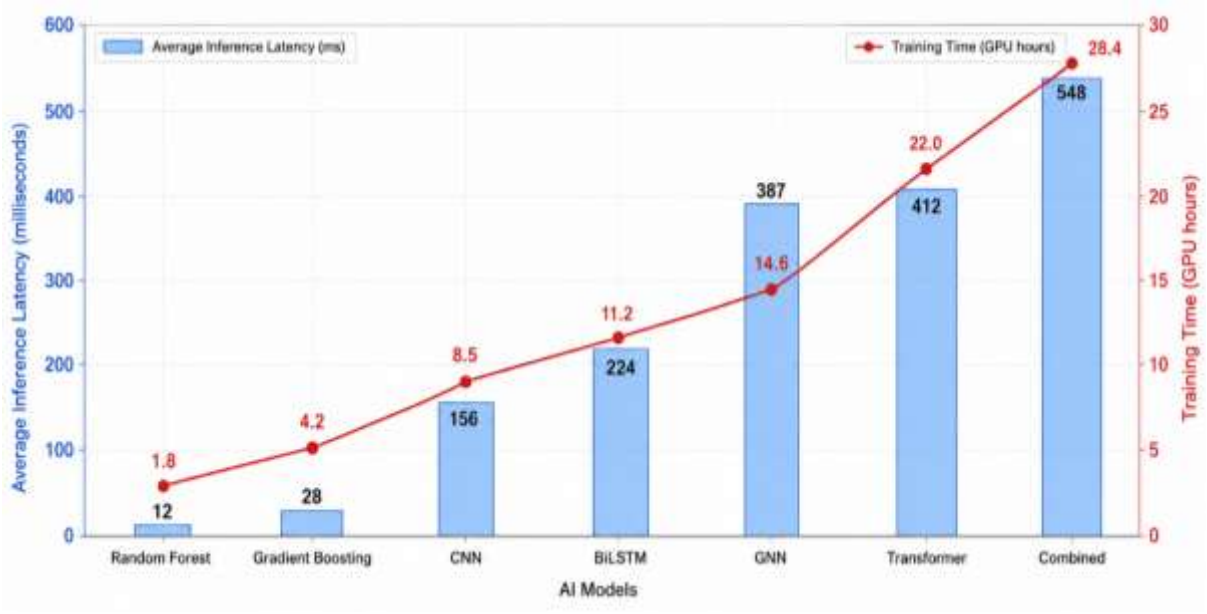
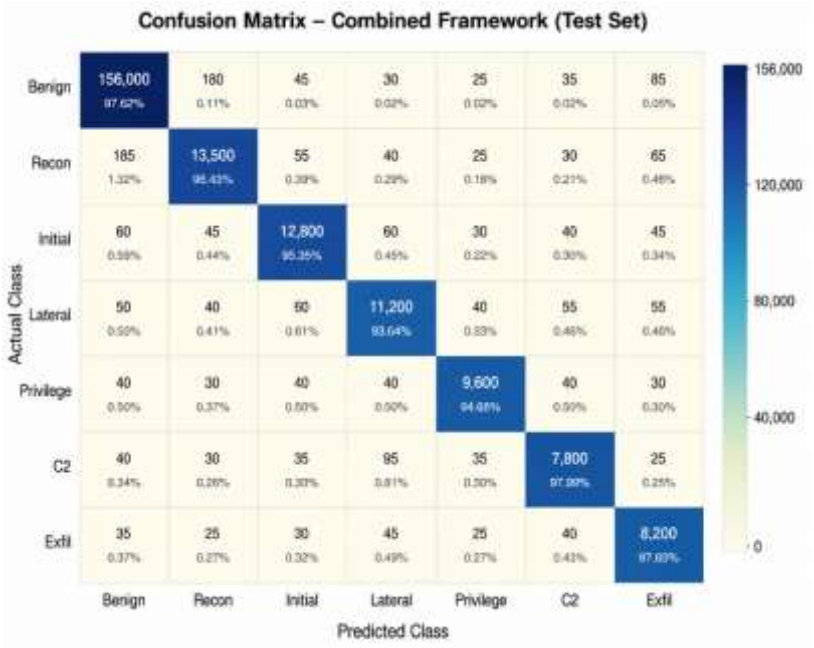


Figure 4: Detection Latency and Computational Cost by Model

The cost analysis tells the other side of the story. The combined framework, while clearly best in accuracy, was also the most expensive to train and the slowest at inference. For organisations with limited resources, choosing the right model may involve trading some accuracy for speed. The Random Forest, despite being the weakest performer, runs in milliseconds and trains in under two hours, making it attractive for resource-constrained environments.



10.48047/jocaaa.2024.33.08.521

Figure 5: Confusion Matrix Heatmap for the Combined Framework

The confusion matrix shows that the combined framework had very few off-diagonal errors. The remaining confusion was concentrated in two areas: reconnaissance being mistaken for benign traffic, which is a well-known hard problem because reconnaissance can look very similar to normal IT activity, and a small overlap between command-and-control and lateral movement activity, which is also known to share characteristics.

6. DISCUSSION

The results of this study tell a fairly clear story about where AI in APT detection is heading. The combined framework, bringing together four different model types, gave clearly better performance than any single model, with the gain being especially visible in the reduction of false positives and the improvement in detecting hard early-stage activity. The reason this approach works is not magic but rather the fact that different attack stages leave different kinds of traces, and matching the right model to the right trace is more effective than expecting one model to handle everything.

The strength of the GNN at catching lateral movement was perhaps the most interesting finding. Lateral movement is one of the hardest stages to detect because it involves legitimate-looking authentication and file access activities, just performed by an attacker who has stolen valid credentials. The reason the GNN does well here is that it can see the structure of host-to-host connections, and attacker movement creates patterns of access that look unusual even when each individual connection looks fine. This is exactly the kind of insight that traditional flat feature analysis misses.

The transformer's contribution was similarly distinctive. Modern attacker techniques increasingly involve obfuscated command-line activity, especially with PowerShell and similar tools, where commands are encoded or scrambled to avoid simple signature matching. Transformers trained on large amounts of security log text can spot suspicious patterns in such text even when the exact obfuscation method is new. This makes them particularly valuable for catching attackers who use living-off-the-land techniques, where they hide inside legitimate administrative tools.

The cost analysis brings up an important practical point. While the combined framework gave the best accuracy, it is also expensive to run, with inference latency above 500 milliseconds and training time of nearly 30 GPU hours. For high-volume environments where decisions must be made in real time, this latency may be unacceptable. A more practical setup might use the lighter models for first-pass filtering and only engage the heavier models on suspicious events. This kind of tiered architecture is becoming more common in real security operations centres.

There are limitations of this study worth acknowledging openly. The financial sector testbed, while useful, is still a controlled environment and may not capture all the noise and edge cases of a real production network. The labelled data, while extensive, was still generated by red team exercises and may not perfectly match the behaviour of real APT groups. There is also the broader concern that any published detection method becomes a target for adversarial research, with attackers trying to find ways to evade it. Robustness to adversarial inputs is therefore an important direction for future work.

10.48047/jocaaa.2024.33.08.521

The question of explainability also deserves attention. Security analysts need to understand why a particular event was flagged so they can make informed decisions about response. The combined framework, with its multiple deep learning models, is harder to explain than a simple Random Forest or Decision Tree. Building explanation tools that show which model contributed to a particular detection, and which features within that model were most important, is essential for analyst trust and operational adoption.

7. CONCLUSION

This study presented a layered AI framework for APT detection that combined four specialised models, namely a Convolutional Neural Network, a Bidirectional LSTM, a Graph Neural Network, and a transformer-based language model. Each model targeted a different aspect of APT behaviour, and a meta-learner combined their outputs into a unified detection decision. Tested on multiple datasets including a private financial sector testbed, the combined framework achieved 97.8% accuracy with a 1.4% false positive rate, clearly outperforming both individual models and standalone machine learning baselines.

The main contribution of this work is the demonstration that no single AI technique is sufficient for the full APT detection problem, but a coordinated combination of techniques can deliver excellent results. The Graph Neural Network's strength at detecting lateral movement and the transformer's effectiveness at parsing obfuscated command-line activity were particularly noteworthy. These are exactly the stages where APT groups have historically been hardest to catch, so improvements there are operationally meaningful.

For security practitioners, the practical messages are several. Investment in AI-based detection is justified by the performance gains, but the choice of techniques should match the specific threats the organisation faces. Combining multiple models adds cost but reduces false positives in important ways. Tiered architectures, where lightweight models filter for heavier models, are likely to be the practical pattern for most production deployments. Continuous retraining is essential because APT groups evolve their techniques, and explainability tooling should be built in from the start rather than added later.

Future research should look at several directions. Adversarial robustness needs more attention, with techniques like adversarial training and input sanitisation built into the framework from day one. Federated learning could allow multiple organisations to train shared models without sharing sensitive raw data, building stronger detectors across the broader security community. More work on real-time inference optimisation is needed to bring the latency of complex frameworks down to operationally usable levels. Finally, integrating AI detection with automated response systems, sometimes called SOAR platforms, would close the loop between detection and action and significantly reduce the time attackers have to operate inside victim networks. The fight against APTs is ongoing, and AI-based defences, when designed and deployed thoughtfully, offer defenders a real chance to shift the balance in their favour.

REFERENCES

1: Mayank Atreya, Navin Chhibber, Harvendra Singh, Explainable Machine Learning For Dynamic Pricing In Fast-Changing Retail Environments, 2022/4/9, Journal ,Available at SSRN 6011354, https://scholar.google.com/citations?view_op=view_citation&hl=en&user=fyViF1UAAAAJ&citation_for_view=fyViF1UAAAAJ:LkGwnXOMwfcC.

10.48047/jocaaa.2024.33.08.521

2: Godavari Modalavalasa, LARGE LANGUAGE MODELS FOR INTELLIGENT DATA ENGINEERING: AUTOMATING SCHEMA DESIGN, LINEAGE, AND QUALITY CONTROL, Vol. 50 No. 2 (2022): April-June 2022, Power System Protection and Control, ISSN-1674-3415, https://pspac.info/index.php/dlbh/article/view/183 , DOI: https://doi.org/10.46121/pspc.50.2.4

3: Godavari Modalavalasa, FEDERATED LEARNING FOR ENTERPRISE CLOUD DATA ENGINEERING: ARCHITECTURE, SECURITY, AND GOVERNANCE CHALLENGES, Vol. 51 No. 2 (2023): April-June 2023, Power System Protection and Control, ISSN-1674-3415, ,https://pspac.info/index.php/dlbh/article/view/184, DOI: https://doi.org/10.46121/pspc.51.2.5

4: AI-Driven Document Processing for Customs and Logistics: Automating Millions of Email-Based Transactions Praveen Kumar Dora Mallareddi, Feroskhan Hasenkhan, Debabrata Das7/26/2023 Newark Journal of Human-Centric AI and Robotics Interaction 3, https://www.njhcair.org/index.php/publication/article/view/13

5: Naresh Lokiny. (2022). Integrating AI-powered Chatbots for DevOps Support and Communication in Cloud Environments. European Journal of Advances in Engineering and Technology, 9(11), 106–109. https://doi.org/10.5281/zenodo.13325989

6: Naresh Lokiny, (2021), "Disaster Recovery and Business Continuity Planning in DevOps Cloud with AI", International Journal of Science and Research (IJSR), 10(3), 2024-2027. https://dx.doi.org/10.21275/SR24724151733, https://www.ijsr.net/getabstract.php?paperid=SR24724151733

7: Naresh Lokiny, & Ranganath Nandanampati. (2020). DevSecOps: Integrating Security into DevOps with AI in Cloud. Journal of Scientific and Engineering Research, 7(10), 239–242. https://doi.org/10.5281/zenodo.13348695

8: Naresh Lokiny, & Pradip Reddy. (2021). Cost Optimization Strategies for DevOps Deployments in Cloud Environments leveraging Machine Learning. European Journal of Advances in Engineering and Technology, 8(3), 69–72. https://doi.org/10.5281/zenodo.13325845

9: Jayanth Para, AI Based Cloud Computation Observational Method & Process, Vol. 51 No. 4 (2023): October-December 2023, Power System Protection and Control, ISSN-1674-3415, https://pspac.info/index.php/dlbh/article/view/171 , DOI: https://doi.org/10.46121/pspc.51.4.3

10: Jobayar Alom, Ahsan Ahmed. (2023). Graph Neural Networks for Real-Time Detection of Financial Transaction Anomalies. Acta Scientiae, 24(5), 82–90. https://doi.org/10.22178/acta.24.5.6

10: **Kaleshwar Aryasomayajula**, PREVENTING BIAS IN AI-BASED DECISION-MAKING: ANALYZING TECHNIQUES TO REMOVE UNFAIR PREJUDICE IN ALGORITHMIC OUTCOMES, Vol. 50 No. 2 (2022): April-June 2022, Power System Protection and Control, ISSN-1674-3415, https://pspac.info/index.php/dlbh/article/view/294

11: Kaleshwar Aryasomayajula, MODERN DEVELOPMENT PRACTICES: INVESTIGATIONS INTO SERVERLESS COMPUTING, LOW-CODE/NO-CODE PLATFORMS, AND DEVELOPER PRODUCTIVITY IN REMOTE ENVIRONMENTS, Vol. 50 No. 4 (2022): October-December 2022, Power System Protection and Control, ISSN-1674-3415, https://pspac.info/index.php/dlbh/article/view/364

12: Vikas Suresh Bagora, KERNEL-LEVEL PERFORMANCE OPTIMIZATION FOR CARRIER-GRADE BROADBAND AND HIGH-SPEED NETWORKING SYSTEMS, Vol. 47 No. 1 (2019),

10.48047/jocaaa.2024.33.08.521

Power System Protection and Control, ISSN-1674-3415, https://pspac.info/index.php/dlbh/article/view/359

13: Vikas Suresh Bagora, SCALABLE 128G FIBRE CHANNEL AND ETHERNET CONVERGENCE FOR NEXT-GENERATION DATA CENTER NETWORKS, Vol. 50 No. 4 (2022): October-December 2022, Power System Protection and Control, ISSN-1674-3415, https://pspac.info/index.php/dlbh/article/view/362

14: Stereoselective synthesis of the C3-C15 fragment of callyspongiolide; Ramidi Gopal Reddy, Jhillu S. Yadav and Debendra K. Mohapatra. (Tetrahedron Letters. 2018, 59, 3579-3582). https://doi.org/10.1016/j.tetlet.2018.08.041, https://www.sciencedirect.com/science/article/pii/S0040403918310426

15. Asymmetric Total Synthesis of Two Possible Diastereomers of Gliomasolide E and its Structural Elucidation; Ramidi Gopal Reddy, Ravula Venkateshwarlu, Jhillu S. Yadav and Debendra K. Mohapatra. (J. Org. Chem. 2017, 82(2), 1053-1063). https://doi.org/10.1021/acs.joc.6b02611 , https://pubs.acs.org/doi/abs/10.1021/acs.joc.6b02611

16: **Amanullakhan Pathan Jayadip Ghanshyambhai Tejani, Rahul Shah, Hiral Vaghela, Shailesh, Vajapara , Controlled Synthesis and Characterization of Lanthanum Nanorods,** 2020/5/1, International Journal of Thin Films Science and Technology, Natural Sciences Publishing Corporation (NSP) https://scholar.google.com/citations?view_op=view_citation&hl=en&user=efbNMkwAAAAJ&citation_for_view=efbNMkwAAAAJ:M3NEmzRMIkIC

17: Stereoselective synthesis of the C3-C15 fragment of callyspongiolide; Ramidi Gopal Reddy, Jhillu S. Yadav and Debendra K. Mohapatra. (Tetrahedron Letters. 2018, 59, 3579-3582). https://doi.org/10.1016/j.tetlet.2018.08.041, https://www.sciencedirect.com/science/article/pii/S0040403918310426

18. Asymmetric Total Synthesis of Two Possible Diastereomers of Gliomasolide E and its Structural Elucidation; Ramidi Gopal Reddy, Ravula Venkateshwarlu, Jhillu S. Yadav and Debendra K. Mohapatra. (J. Org. Chem. 2017, 82(2), 1053-1063). https://doi.org/10.1021/acs.joc.6b02611, , https://pubs.acs.org/doi/abs/10.1021/acs.joc.6b02611

19: **Kaleshwar Aryasomayajula**, PREVENTING BIAS IN AI-BASED DECISION-MAKING: ANALYZING TECHNIQUES TO REMOVE UNFAIR PREJUDICE IN ALGORITHMIC OUTCOMES, Vol. 50 No. 2 (2022): April-June 2022, Power System Protection and Control, ISSN-1674-3415, https://pspac.info/index.php/dlbh/article/view/294

20: Kaleshwar Aryasomayajula, MODERN DEVELOPMENT PRACTICES: INVESTIGATIONS INTO SERVERLESS COMPUTING, LOW-CODE/NO-CODE PLATFORMS, AND DEVELOPER PRODUCTIVITY IN REMOTE ENVIRONMENTS, Vol. 50 No. 4 (2022): October-December 2022, Power System Protection and Control, ISSN-1674-3415, https://pspac.info/index.php/dlbh/article/view/364

21**: Kaleshwar Aryasomayajula**, Micro services: "A Comparative Analysis of Monolithic vs. Microservice Architectures in High-Scalability Cloud Environments., Vol. 51 No. 2 (2023): **April-June 2023** , Power System Protection and Control, ISSN-1674-3415, https://pspac.info/index.php/dlbh/article/view/374

10.48047/jocaa.2024.33.08.521

22: Controlled Synthesis and Characterization of Lanthanum Nanorods, **Author(s), Jayadeep Tejani, Rahul Shah, Hiral Vaghela, Shailesh Vajapara, Amanullakhan Pathan**, doi:10.18576/ijtfst/090205, URL: https://www.naturalspublishing.com/Article.asp?ArtcID=20705, **May-2020**