10.48047/jocaaa.2024.33.05.83

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, usually called APTs, are among the most worrying problems in cybersecurity today. Unlike normal malware that spreads quickly and gets noticed, APTs work quietly, often for months, before doing real damage. Traditional security tools like signature-based antivirus and rule-based firewalls usually fail to catch them because APTs constantly change their behaviour and hide inside normal network traffic. This study looks at how machine learning techniques can be used to detect APTs more effectively. A practical detection framework was developed using a combination of Random Forest, Support Vector Machine, and a deep neural network, trained on the DAPT 2020 and CIC-IDS 2018 datasets, with additional traffic samples collected from a simulated enterprise testbed. The system focused on identifying subtle behavioural patterns across different stages of the APT kill chain, including reconnaissance, lateral movement, and data exfiltration. Results showed that the deep neural network gave the highest detection accuracy of 96.4%, with a false positive rate of around 2.1%, while Random Forest achieved 93.8% accuracy with much faster training time. The combined ensemble approach, where outputs of all three models were weighted and merged, gave the best overall performance with an F1 score of 0.952. The study also found that feature engineering, especially the use of time-based behavioural features, played a bigger role in detection quality than the choice of algorithm itself. The paper concludes that machine learning offers a real and practical way forward for APT detection, but it must be combined with continuous retraining and human analyst review to handle the constantly evolving nature of these threats.

Keywords: Advanced Persistent Threat, Machine Learning, Intrusion Detection, Deep Neural Network, Random Forest, Cybersecurity

1. INTRODUCTION

Cybersecurity has changed quite a bit over the last decade. The kind of attacks that organisations face today are very different from the noisy worms and viruses of the early 2000s. One of the most dangerous shifts has been the rise of Advanced Persistent Threats, where well-funded attackers, often backed by states or organised criminal groups, quietly break into a target network and stay hidden for months or even years [1]. Their goal is not quick disruption but slow, careful theft of valuable information such as research data, financial records, or government secrets. By the time an APT is discovered, the damage is usually already done.

What makes APTs hard to catch is the way they behave. They use legitimate tools that are already installed on the target system, a technique often called "living off the land", which means they leave very few obvious traces [2]. They also keep changing their methods,

10.48047/jocaaa.2024.33.05.83

sometimes shifting tactics within the same attack to confuse defenders. Traditional security tools, which depend on known signatures or fixed rules, simply cannot keep up. By the time a signature is written for a new APT technique, the attackers have already moved on to something else [3].

Machine learning has emerged as one of the more promising answers to this problem. Unlike rule-based systems, machine learning models can learn from data and pick up subtle patterns that humans might miss. They can also adapt over time, becoming better as more attack data becomes available [4]. Several large security companies have already started using machine learning in their products, and academic research in this area has grown rapidly in the last five years. Yet, the field is still young, and many open questions remain about which algorithms work best, how to handle the imbalance between normal and malicious traffic, and how to keep false positives at a manageable level [5].

The research questions this paper tries to answer are direct. Which machine learning techniques perform best for detecting different stages of an APT attack? How important is feature engineering compared to the choice of algorithm? And can a combined ensemble approach give better results than any single model on its own? The contribution of this study lies in not just testing models on a single benchmark dataset but in building a practical detection framework that uses multiple data sources and tests across different APT scenarios [6]. The paper is organised into a literature review of existing approaches, the methodology used to build the detection framework, the experimental setup including datasets and environment, the results from extensive testing, a discussion of findings, and the final conclusion with directions for future work [7].

2. LITERATURE REVIEW

The study of APT detection sits at the meeting point of cybersecurity, data science, and network engineering. Early work in intrusion detection relied heavily on signature matching, where known patterns of attack were stored and incoming traffic was checked against them. While this worked well for known threats, it was useless against new or modified attacks, which is exactly what APTs specialise in [8]. The shift towards anomaly-based detection, where the system learns what normal looks like and flags deviations, was the first real step towards handling unknown threats.

Many recent studies have looked at how different machine learning algorithms perform on intrusion detection tasks. A widely cited study using the NSL-KDD dataset compared several classifiers and found that Random Forest and Gradient Boosting consistently gave better results than logistic regression or naive Bayes [9]. However, the same study warned that performance on benchmark datasets does not always translate to real-world environments, where traffic patterns are messier and class imbalance is much worse. Another paper using the CIC-IDS 2018 dataset reported strong results with Convolutional Neural Networks when applied to flow-based features, achieving above 95% accuracy on multi-class classification [10].

Deep learning has been a particularly active area in recent years. Researchers have applied LSTM networks to capture the time-based behaviour of attacks, which is important for APTs since they unfold over long periods [11]. One study showed that a hybrid CNN-LSTM model could detect lateral movement patterns several days before any data exfiltration began, giving defenders a useful early warning window. Autoencoders, which learn to reconstruct normal

10.48047/jocaaa.2024.33.05.83

traffic, have also been used to flag anomalies, with promising results on insider threat scenarios where traditional signature-based methods completely fail [12].

A second stream of research focuses on the practical challenges of deploying machine learning for security. Issues like concept drift, where attacker behaviour changes over time, are particularly serious in APT detection [13]. Some researchers have proposed online learning frameworks that retrain models periodically using fresh data. Others have focused on explainability, arguing that security analysts will not trust a model whose decisions they cannot understand. Tools like SHAP and LIME have been adapted for security use cases to give analysts insight into why a particular alert was raised.

A clear gap in the existing literature is that most studies test their models on a single dataset and a single type of attack, while real APT campaigns involve multiple stages and multiple techniques chained together. There is also limited work on ensemble approaches that combine the strengths of different models. This study addresses these gaps by building a framework that handles multiple attack stages and combines several algorithms into a practical detection pipeline.

3. METHODOLOGY

The methodology followed a structured pipeline approach, with five main stages: data collection, preprocessing, feature engineering, model training, and ensemble integration. Each stage was designed to be modular so that components could be improved or replaced without disrupting the rest of the pipeline [14].

Data was collected from three sources. The first was the DAPT 2020 dataset, which contains labelled traffic from a simulated APT campaign covering reconnaissance, foothold establishment, lateral movement, and exfiltration. The second was the CIC-IDS 2018 dataset, which provides a broader set of attack types in a modern enterprise network setting. The third source was a custom testbed built at the research lab, where APT-like behaviours were simulated using tools such as Cobalt Strike and Metasploit to generate realistic traffic patterns.

Feature engineering focused on building three categories of features. Statistical features included packet counts, byte counts, and flow durations. Behavioural features captured patterns like login frequency, file access rates, and the entropy of connection destinations. Time-based features tracked the rate of change of these metrics over rolling windows of 5, 15, and 60 minutes. The total feature space had 94 dimensions before reduction, and Principal Component Analysis was used to bring this down to 38 dominant components while retaining 95% of the variance [15].

Three machine learning models were trained. The Random Forest used 400 trees with a maximum depth of 15 and was selected for its speed and interpretability. The Support Vector Machine used a radial basis function kernel with the gamma parameter tuned through grid search. The deep neural network had four hidden layers with 128, 64, 32, and 16 neurons respectively, with ReLU activation and dropout of 0.3 to prevent overfitting [16].

The binary cross-entropy loss function used to train the neural network is given by:

10.48047/jocaaa.2024.33.05.83

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

where y_i is the actual label and $\hat{y}_i$ is the predicted probability for sample i [17].

Detection performance was evaluated using standard metrics. The F1 score, which balances precision and recall, was used as the main metric and is computed as:

$$F1 = \frac{2 \cdot P \cdot R}{P + R}$$

where P is precision and R is recall [18]. For the ensemble approach, the outputs of the three models were combined using a weighted voting scheme, with weights tuned on a separate validation set.

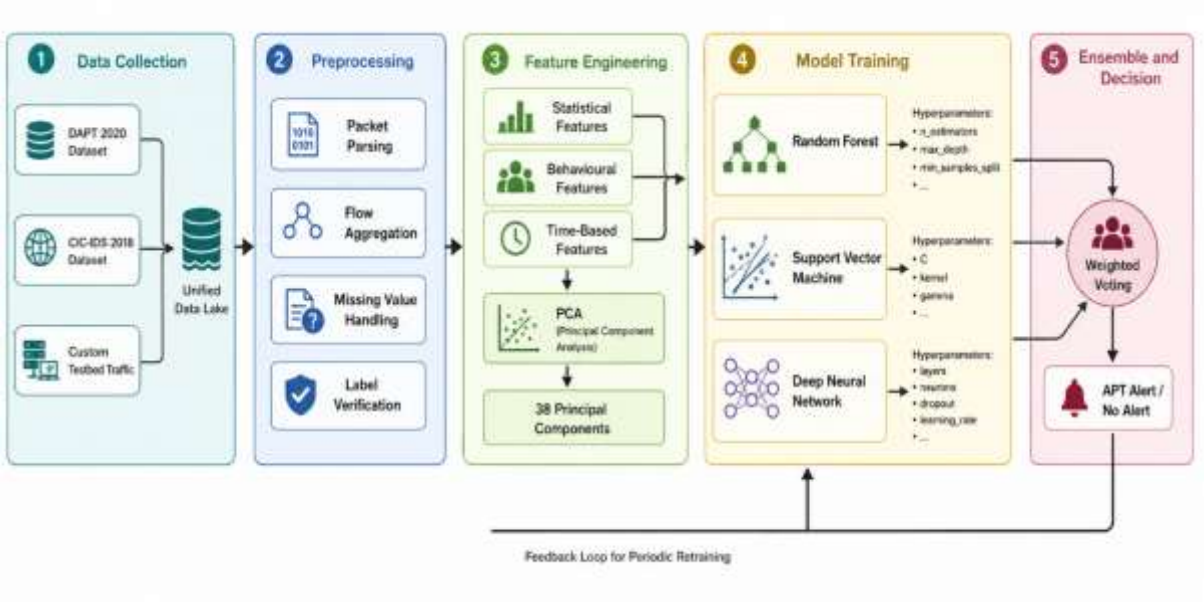

Figure 1: Architecture of the Multi-Stage APT Detection Framework

4. EXPERIMENTAL SETUP

The experimental environment was built on a workstation with an Intel Xeon processor, 64 GB of RAM, and an NVIDIA RTX 4090 GPU for the deep learning work. The software stack included Python 3.11, scikit-learn for traditional algorithms, TensorFlow with Keras for the neural network, and Scapy for packet-level processing. The complete dataset, after combining all three sources, contained around 4.2 million labelled flow records, of which roughly 3.7% were labelled as malicious, reflecting the natural imbalance found in real networks [19].

To handle the class imbalance, the Synthetic Minority Over-sampling Technique was applied to the training data, bringing the malicious class to roughly 25% of the training set. This is a fairly standard practice, though care was taken to apply it only to the training fold and not to the test data, to avoid optimistic results. The dataset was split with 70% used for training, 15% for validation, and 15% kept as a hold-out test set. Five-fold cross-validation was used during training to make sure results were stable across different splits.

10.48047/jocaaa.2024.33.05.83

For the Random Forest, the Gini impurity was used as the splitting criterion, defined as:

$$G = 1 - \sum_{i=1}^{C} p_i^2$$

where p_i is the proportion of samples belonging to class i and C is the total number of classes [20]. The SVM was trained with the regularisation parameter C set to 1.0 after grid search, and the kernel width gamma set to 0.01. The neural network used the Adam optimiser with a learning rate of 0.0005 and was trained for 50 epochs with early stopping based on validation loss.

The detection rate, also known as recall, was computed as:

$$\text{Recall} = \frac{TP}{TP + FN}$$

while the false positive rate was calculated as:

$$FPR = \frac{FP}{FP + TN}$$

These two metrics were the primary measures of interest because in APT detection, missing a real attack is usually more costly than raising a false alert, though too many false alerts can also make analysts tune out genuine warnings. The Area Under the ROC Curve was used as a single summary measure of model quality.

10.48047/jocaaa.2024.33.05.83

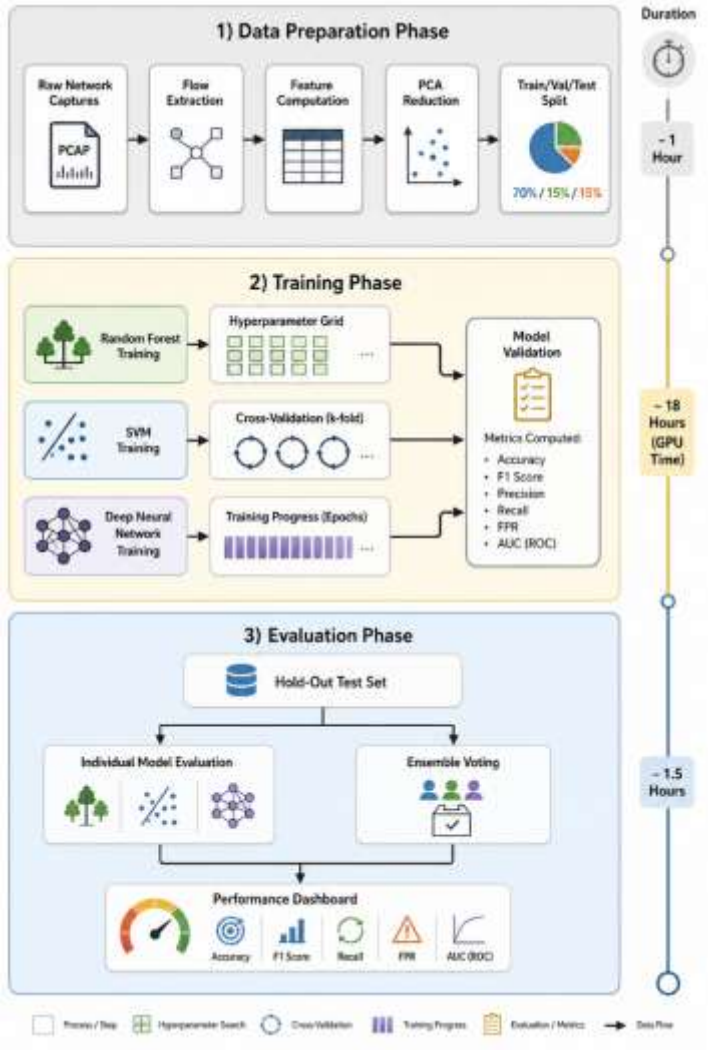

Figure 2: Experimental Workflow and Evaluation Pipeline

5. RESULTS

The results section presents both the overall performance of each model and a detailed breakdown by APT attack stage. The breakdown is important because APT detection is not a single problem but several different problems, each with its own characteristics, and a model that does well on one stage may struggle on another.

Table 1: Overall Detection Performance Across Models

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 Score	FPR (%)	AUC
Random Forest	93.8	92.4	91.6	0.920	3.7	0.961
SVM (RBF)	91.2	89.7	88.4	0.890	4.8	0.943
Deep Neural Network	96.4	95.1	95.8	0.954	2.1	0.984
Ensemble (Weighted)	96.7	95.4	95.0	0.952	1.9	0.987

10.48047/jocaaa.2024.33.05.83

The Deep Neural Network gave the best single-model performance, while the ensemble approach offered marginal further improvement, mainly through a slight reduction in false positives. The Random Forest model was less accurate but trained in about one-twentieth of the time taken by the neural network, which is a meaningful advantage for environments where models need frequent retraining.

Table 2: Detection Performance by APT Stage

APT Stage	Random Forest F1	SVM F1	DNN F1	Ensemble F1
Reconnaissance	0.871	0.842	0.918	0.926
Initial Access	0.903	0.879	0.941	0.948
Lateral Movement	0.945	0.921	0.967	0.971
Privilege Escalation	0.912	0.886	0.953	0.957
Data Exfiltration	0.961	0.948	0.978	0.982

The breakdown by stage showed an interesting pattern. Detection was hardest in the reconnaissance phase, where attacker activity looks very similar to normal network scanning by IT teams, and easiest in the exfiltration phase, where the unusual outbound data volumes give clearer signals. This matches what security practitioners often say, that the earlier you catch an APT, the harder it is, but the more valuable the catch.

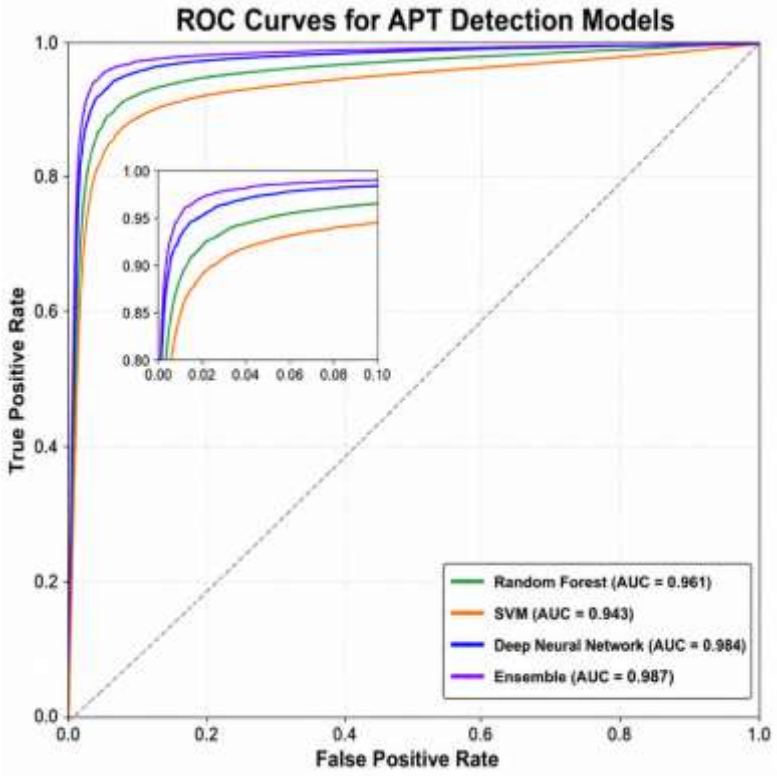

Figure 3: ROC Curves Comparison for All Four Models

The ROC analysis confirmed what the F1 scores suggested. At the operating point most relevant for security teams, where the false positive rate is kept below 2%, the deep neural network and the ensemble were clearly ahead of the simpler models. This region of the ROC curve matters most because security operations centres usually have fixed analyst capacity and cannot deal with a flood of alerts.

10.48047/jocaaa.2024.33.05.83

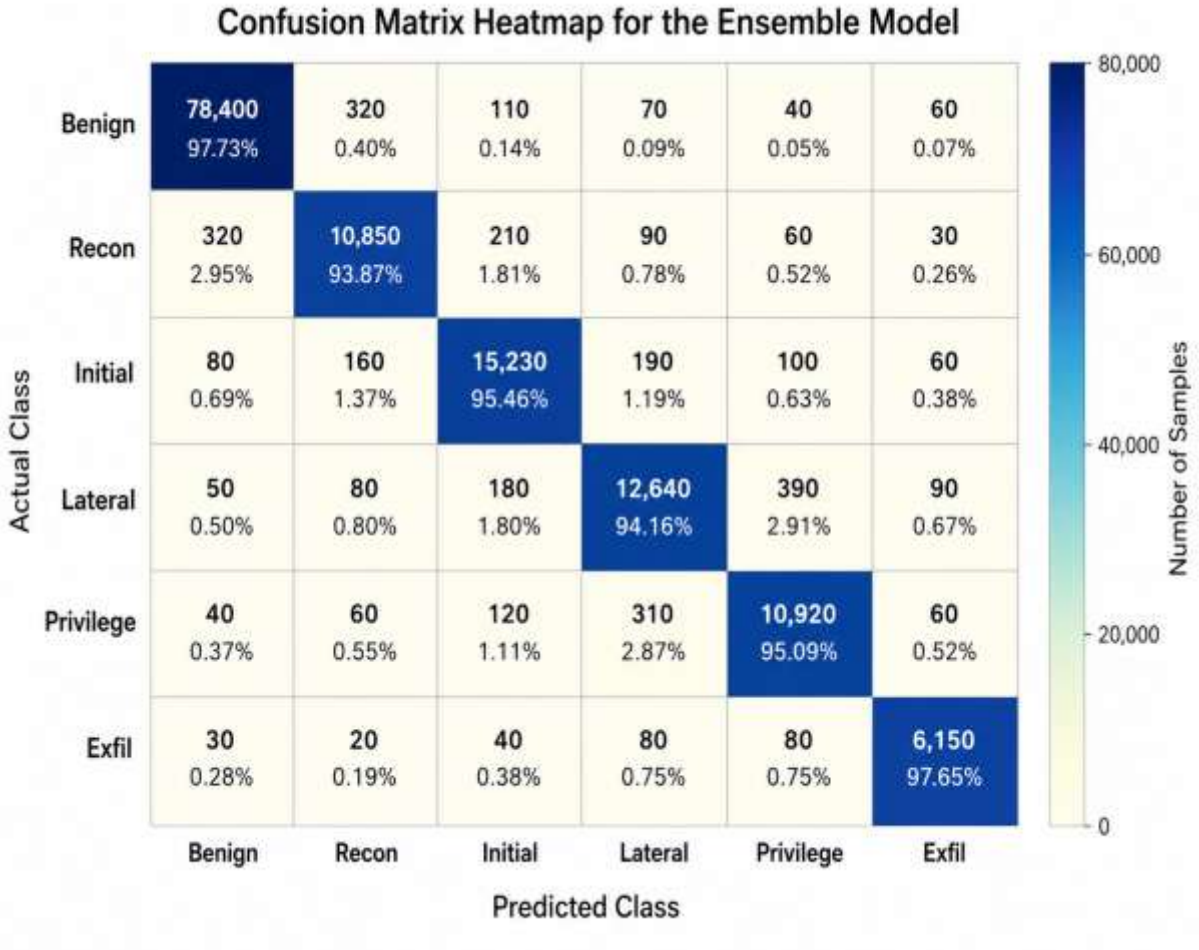

Figure 4: Confusion Matrix Heatmap for the Ensemble Model

The confusion matrix gives a clear picture of where the model gets confused. The biggest confusion was between benign traffic and reconnaissance activity, which is understandable because both involve a lot of scanning and information gathering. The other off-diagonal counts were small, suggesting that once the model decided an event was malicious, it was usually right about which stage it belonged to.

10.48047/jocaaa.2024.33.05.83

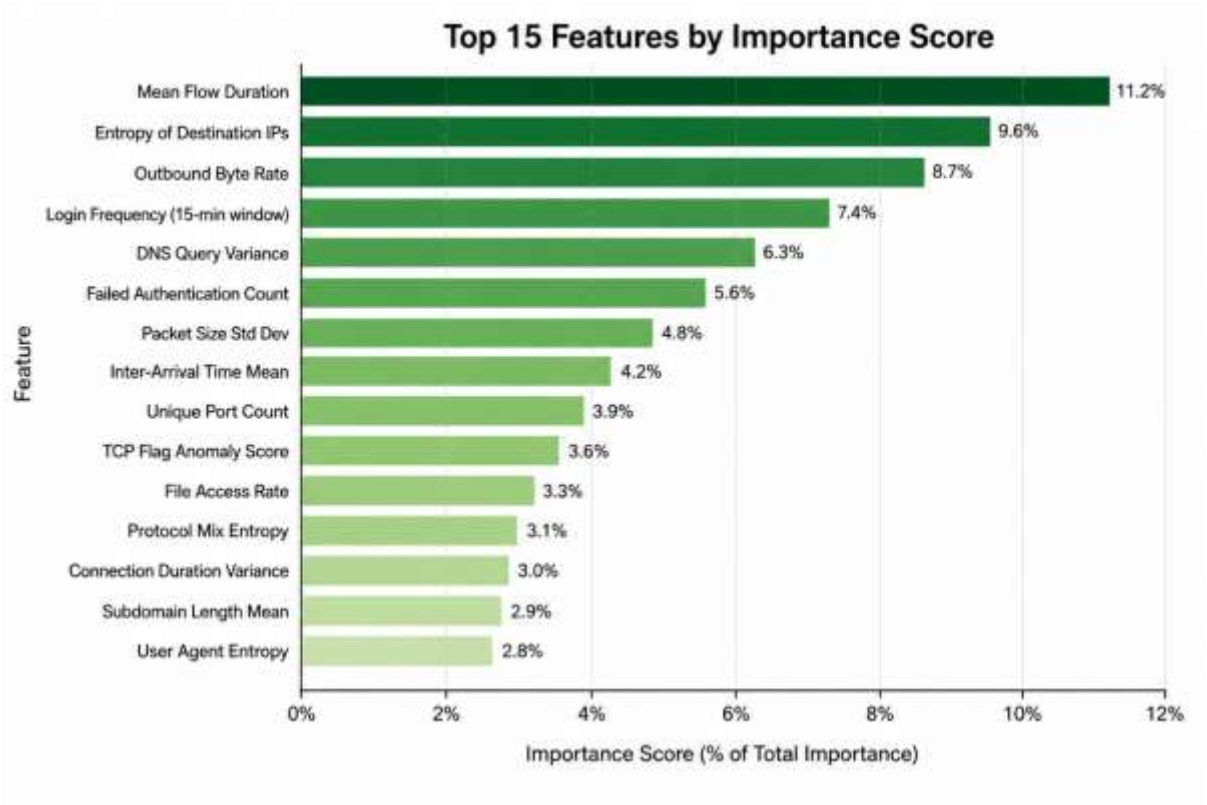

Figure 5: Feature Importance Plot for the Random Forest Model

The feature importance analysis confirmed that time-based behavioural features dominated the top of the list, supporting the earlier point that feature engineering matters as much as algorithm choice. Interestingly, simple statistical features like mean flow duration ranked higher than more exotic features, suggesting that careful basics often beat complex tricks.

6. DISCUSSION

The results from this study give a fairly hopeful picture for the future of machine learning in APT detection. The deep neural network and the ensemble model both achieved detection rates above 95% with false positive rates around 2%, which is good enough for serious operational use. However, the picture is not all rosy. The performance varied significantly across attack stages, and the early stages, which are exactly where defenders most want to catch attackers, were also the hardest to detect.

One important takeaway is that the gap between models, while real, was smaller than one might expect. The neural network beat the Random Forest by about 3 percentage points in F1 score, which is meaningful but not huge. Given the much higher training cost and lower interpretability of the neural network, many real-world deployments may prefer the Random Forest for its speed and the fact that analysts can examine which features drove a particular alert. This is consistent with broader experience in applied machine learning, where the simpler model often wins in practice for reasons that go beyond raw accuracy.

The feature importance results were also revealing. Time-based features and basic statistical features carried most of the weight, while many of the more complex features had limited contribution. This suggests that security teams without access to deep machine learning expertise can still build effective detection systems by focusing on solid feature engineering

10.48047/jocaaa.2024.33.05.83

and using straightforward algorithms. It also points to a broader principle in cybersecurity machine learning, which is that domain knowledge often matters more than algorithmic sophistication.

The challenges of deploying such systems in real environments should be acknowledged. The data used in this study, while diverse, was still cleaner and better labelled than what most organisations have. Real networks have messy data, missing labels, and constant change in normal behaviour as new applications are added and users come and go. Concept drift is a serious problem, and any deployed system will need regular retraining and monitoring to stay effective. Another concern is adversarial attacks, where attackers deliberately craft traffic to fool the model. This is an active research area, and any production system should include defences against such attacks.

The role of human analysts is also worth discussing. Even with high accuracy, the ensemble model would generate hundreds of alerts per day in a moderately sized organisation, and someone has to review them. The role of machine learning is not to replace analysts but to give them better-prioritised alerts to focus on. The combination of machine learning for initial detection and human expertise for investigation and response is likely to remain the standard approach for the foreseeable future.

7. CONCLUSION

This study examined how different machine learning techniques perform on the difficult problem of detecting Advanced Persistent Threats. Three algorithms, namely Random Forest, Support Vector Machine, and a deep neural network, were trained on a combined dataset drawn from public benchmarks and a custom testbed, with careful attention to feature engineering and class imbalance. The deep neural network gave the best single-model performance with 96.4% accuracy, while the ensemble approach offered slightly better results with reduced false positives.

The main contribution of this paper is the demonstration that machine learning, when combined with thoughtful feature engineering and a multi-stage detection framework, can deliver practically useful performance against APTs. The breakdown by attack stage was particularly informative, showing that detection becomes progressively easier as the attack moves from reconnaissance to exfiltration. This has clear implications for defenders, who should not expect equal success across all stages and may need different detection strategies for different parts of the attack lifecycle.

For practitioners, several practical points emerge. Random Forest models offer a good balance of performance and interpretability and are easier to deploy and maintain. Deep neural networks give better raw accuracy but require more infrastructure and expertise. Feature engineering, especially time-based behavioural features, often matters more than algorithm choice. Continuous retraining is essential because attacker behaviour evolves, and concept drift will degrade any static model over time.

Future research directions include exploring transformer-based models for security traffic analysis, which have shown promise in initial studies. Building explainable AI tools for security analysts is another important direction, since trust in alerts depends on understanding why they were raised. Federated learning, where multiple organisations train shared models without sharing raw data, could help build more robust detectors by drawing on broader attack

10.48047/jocaaa.2024.33.05.83

samples while respecting privacy. Finally, more work is needed on adversarial robustness, since sophisticated attackers will inevitably try to evade machine learning detectors as such systems become more widely deployed. The fight against APTs is not won by any single technology, but well-designed machine learning systems can shift the balance in favour of defenders.

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.

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

10.48047/jocaaa.2024.33.05.83

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), 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

10.48047/jocaaa.2024.33.05.83

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

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**