КІБЕРБЕЗПЕКА: освіта, наука, техніка № 2 (30), 2025
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
Спеціальний випуск ISSN 2663 – 4023
Special issue

DOI 10.28925/2663-4023.2025.30.915
UDC 004.056

Maksym Y. Opanovych
Postgraduate student of the Department of Information Security
Lviv Polytechnic National University, Lviv, Ukraine
ORCID: 0000-0002-2748-2965
maksym.y.opanovych@lpnu.ua

ANOMALY DETECTION IN ACTIVE DIRECTORY FOR APT ATTACK DETECTION: METHODS, EFFECTIVENESS, AND LIMITATIONS

Abstract. The article presents an overview, analysis and comparison of machine learning methods for detecting threats in the Active Directory environment. Since modern attack methods are increasingly better at bypassing traditional protection tools and better at mimicking legitimate activity, anomaly detection was chosen as the method for detecting threats. 4 types of algorithms were selected for the study: cluster, classifier, neural network and graph-based algorithms. The experiment was conducted on the example of a virtual environment that simulated Active Directory. The machine learning model was trained on the typical operation of the simulated environment. Attacks, for testing, were simulated in the same environment simultaneously with the reproduction of average statistical activity, to reflect conditions closer to real ones. The study showed that each algorithm has its advantages and disadvantages and that there is no reliable solution capable of fully detecting all threats. Thus, LSTMs were the best at detecting anomalies in user behavior, Autoencoders were the best at detecting anomalies in the command line, and the graph-based algorithm was the best at detecting anomalies in network traffic. At the same time, the results highlight the limitations of classification algorithms in scenarios with high variability of cases, as well as the limitations in the ability to detect unknown and legitimate activity-mimicking threats for classification algorithms. The results of the study confirm the need to build comprehensive information protection systems based on the principles of defense-in-depth, combining all the strengths of different methods for effective threat detection. In addition, it emphasizes that the practical value of these models is maximized when they are used to improve the capabilities of existing security platforms.

Keywords: Advanced Persistent Threat; Active Directory; Anomaly Detection; Machine Learning; Deep Learning; Cybersecurity; Defense-in-Depth; Graph-Based Analytics.

INTRODUCTION

Advanced Persistent Threats are among the most sophisticated and dangerous forms of cyberattacks, designed to infiltrate and persist within networks for extended periods while remaining undetected. Unlike conventional cyber threats that aim for immediate disruption, APTs are characterized by their stealth, persistence, and a clear focus on acquiring sensitive data over time. These threats pose substantial risks to organizations and critical infrastructure, as attackers often exploit advanced techniques, including zero-day vulnerabilities and social engineering, to gain unauthorized access and maintain a foothold within targeted networks. Given the high stakes associated with APTs, detecting these threats has become a top priority for cybersecurity professionals, as traditional security mechanisms, such as Intrusion Detection Systems and firewalls, often fall short against APTs' advanced evasion tactics.

As APTs continue to evolve in complexity, the cybersecurity field increasingly turns to anomaly detection through machine learning and deep learning methods as a promising approach for identifying deviations from typical network behavior that may indicate an ongoing attack. Anomaly detection methods analyze behavioral patterns within network traffic, user activities, and system logs, aiming to identify signs of intrusion by spotting unusual or unexpected actions. These methods are particularly relevant in environments such as Active Directory, which serves as the backbone of identity management and access control in many organizations. In AD environments, the ability to detect and respond to anomalies in user behaviors and access patterns is important, as attackers often exploit AD vulnerabilities to escalate privileges, move laterally, and access sensitive data.

Despite their promise, ML and DL approaches to anomaly detection in APT scenarios face significant challenges. Current research highlights critical issues such as the scarcity of labeled datasets for training, computational resource demands, real-time detection limitations, and model interpretability difficulties. These limitations can hinder the practical implementation of anomaly detection techniques, especially in large, complex environments. Moreover, while anomaly detection techniques show promise in laboratory conditions, real-world application in Active Directory environments presents unique challenges. Active Directory systems generate large volumes of event data, which must be analyzed in real-time and with high accuracy to prevent attackers from exploiting undetected vulnerabilities.

The detection of malicious code and activities is a critical component of cybersecurity efforts. In modern enterprise environments, threats have become increasingly sophisticated, leveraging advanced techniques to bypass traditional security measures. Detecting these threats presents several challenges. First, the evolving threat landscape means attackers are continually developing new malware variants and attack vectors, making it difficult for static detection methods to keep pace. Advanced Persistent Threats pose a significant challenge as they involve prolonged and targeted attacks where adversaries stealthily infiltrate networks, often remaining undetected for months.

Moreover, adversaries employ obfuscation and encryption techniques to hide malicious payloads from signature-based detection systems. Insider threats are also a concern; malicious activities originating from within the organization are harder to detect due to the inherent trust and access privileges granted to insiders. Additionally, the high volume of data generated in enterprise networks makes it challenging to monitor and analyze all activities effectively. These challenges necessitate advanced detection mechanisms capable of identifying both known and unknown threats by analyzing patterns and behaviors indicative of malicious activities.

While machine learning algorithms offer powerful tools for detecting anomalies and malicious activities, adversaries employ various techniques to evade detection. One common method is through adversarial machine learning attacks. In evasion attacks, attackers modify inputs during the testing phase to cause misclassification without altering the malicious functionality. For instance, malware code can be slightly altered to evade detection while retaining its harmful capabilities. Data poisoning attacks involve injecting malicious data into the training dataset, corrupting the learning process. This manipulation can lead the model to incorrectly classify malicious activities as benign.

Obfuscation techniques are also prevalent. Attackers may employ code obfuscation, altering the appearance of malicious code without changing its functionality. This can deceive models that rely on code structure or syntax patterns for detection. Command obfuscation is another tactic, where adversaries use encoding, concatenation, or other methods to hide malicious commands from detection systems.

Mimicking normal behavior is an effective strategy for evading detection. In living-off-the-land attacks, adversaries use legitimate tools and processes available within the system to carry out malicious activities, making it difficult for machine learning models to distinguish between normal and malicious behaviors. Credential abuse is another example, where attackers use stolen or compromised credentials to perform unauthorized actions that appear legitimate, bypassing models that rely on authentication anomalies.

Attackers also exploit model blind spots by manipulating or hiding features that machine learning models use for detection, such as altering timestamps or user-agent strings. They may employ novel attack vectors, using new techniques or tools that the model has not been trained to recognize, exploiting its inability to generalize beyond the training data. Timing-based evasion is another tactic, where malicious activities are carried out at a slow pace, blending into normal network traffic patterns and evading detection by models that focus on high-volume anomalies. Alternatively, adversaries may conduct attacks during off-hours when monitoring is less rigorous, reducing the likelihood of immediate detection.

Problem statement. While the reviewed methods offer promising results, common challenges emerge, including high computational demands, data diversity issues, scalability limitations, and the need for improved interpretability in complex models. These findings highlight the importance of understanding both the strengths and limitations of these anomaly detection approaches. By examining the effectiveness and constraints of each method, this review aims to pinpoint the techniques best suited for Active Directory environments, where factors such as real-time processing, and adaptability to varied data are critical for effective APT detection.

Analysis of the recent research and publications. Alsanad and Altuwaijri [1] propose a framework that applies algorithms such as APRIORI and K-means to detect APT attacks by analyzing network data traffic. Their study highlights the high accuracy achieved with the Support Vector Machine with Radial Basis Function (99.2%), showcasing the potential of clustering methods in anomaly detection. However, the authors note that the scarcity of labeled data for training remains a significant limitation, impacting the method's efficiency and real-time applicability. This framework, while effective in structured data analysis, indicates challenges in adapting to environments requiring rapid response and diverse data integration.

Cho Do Xuan et al. [2] present a multi-layered approach for APT detection that integrates machine learning models for analyzing network connections, Suricata logs, and compiled behavioral profiles. This method enhances detection performance compared to singular approaches by providing a more comprehensive assessment of network anomalies. However, the authors acknowledge the difficulties in acquiring characteristic data specific to attack campaigns, which poses a challenge for implementing such models in diverse, large-scale network environments.

Schindler's study [3] leverages graph databases alongside machine learning for detecting APTs through structured profiling of log data and the application of the kill chain model. This approach successfully correlates multiple interrelated log events, enhancing precision and reducing detection time. Despite these strengths, the paper notes limitations related to scalability and the handling of complex, varied data sources, which could impede its broad applicability in dynamic, multi-faceted environments like Active Directory.

AL-Aamri et al. [4] propose a multi-stage APT detection framework utilizing a composition-based decision tree model, focusing on real-time analysis and time-series data from diverse monitoring channels. The framework demonstrates a higher performance in detecting real-time attacks compared to existing algorithms. However, the study points out that the approach faces challenges in adapting to new and evolving threat scenarios, which can limit its resilience in handling unexpected anomalies or novel attack patterns.

Mamun and Shi [5] introduce the DeepTaskAPT framework, which is based on a task-tree approach using Long Short-Term Memory (LSTM) networks for detecting insider APTs. This model analyzes sequences of user tasks to identify anomalies, demonstrating its efficacy in internal threat detection. However, the paper notes that the model’s applicability to detecting external APT threats and its scalability for larger systems require further exploration, limiting its generalizability to broader security contexts.

Cho Do Xuan and Tung Thanh Nguyen [6] propose the BiADG model, which combines BiLSTM-Attention with a Dynamic Graph Convolutional Neural Network (DGCNN) to construct and analyze behavioral profiles for APT detection. The study achieves high precision rates (84-91%), indicating the method’s effectiveness in behavior profiling and anomaly identification. The authors developed this approach to overcome the limitations of prior graph-based methods, which they note suffered from high computation costs and complexity in extracting relationships between data points.

The purpose of the work. The purpose of this article is to conduct a comprehensive review of existing ML and DL methods for anomaly detection, with three primary objectives:

1. To evaluate the effectiveness of these methods in identifying APT-related anomalies
2. To assess their practical limitations
3. To determine which techniques are most applicable within Active Directory environments

By examining each approach's strengths and weaknesses, this article aims to provide insights into how anomaly detection can be harnessed as a reliable APT detection method and outline the path forward for integrating these techniques into real-world cybersecurity strategies.

METHODOLOGY AND EXPERIMENTAL SETUP

To empirically evaluate the performance of the machine learning algorithms discussed, a structured, multi-phase experiment was designed and conducted in a controlled laboratory environment. The methodology ensures the creation of a high-fidelity baseline of normal activity and the systematic injection of malicious activities to test the detection capabilities of each model family across the defined use cases.

Phase 1: Environment Setup and Baseline Data Collection A virtualized Active Directory environment was established, simulating a typical small-to-medium enterprise network. The environment consisted of a Domain Controller and several Windows client endpoints. To generate a robust baseline of "normal" behavior, a combination of automated scripts and manual user simulation was executed over a period of several weeks. This simulated activity included:

- Standard user logins and logoffs at typical business hours.
- Accessing file shares and corporate web applications.
- Running common productivity software (e.g., Microsoft Office).
- Executing routine administrative scripts and system queries.

During this phase, comprehensive logs were collected from all endpoints. Key data sources included Windows Event, Sysmon for detailed process and network connection tracking.

Phase 2: Data Preprocessing and Feature Engineering The raw log data was parsed, normalized into a unified format, and enriched with contextual information. From this clean data, features were engineered specifically for each use case:

- **For User Behavior Anomalies:** Features included logon timestamp, logon type, source IP address, frequency of access requests to critical resources, and the sequence of user actions over time.

- **For Command Line Anomalies:** Raw command line strings were tokenized. Features included the parent process, command length, process name, user name.
- **For Network Traffic Anomalies:** Features were extracted from network flow data, including source/destination IP and port, protocol, bytes sent/received, process name and source workstation name.

Phase 3: Model Training The machine learning and deep learning models from each of the four families (Clustering, Classification, Neural Networks, Graph-Based Methods) were trained exclusively on the preprocessed dataset of normal activity. This phase established a baseline model of normalcy for the AD environment.

Phase 4: Threat Emulation and Evaluation After training, the models were evaluated by injecting a series of emulated attacks that mimicked known APT tactics, techniques, and procedures.

RESEARCH RESULTS

Machine learning, while powerful, is not infallible and should not be the sole line of defense in cybersecurity strategies. Relying exclusively on machine learning models can create vulnerabilities that adversaries may exploit. A defense-in-depth approach involves layering multiple security measures to provide redundancy and address different aspects of security. By integrating machine learning-based detection with signature-based systems, heuristic analysis, and rule-based methods, organizations can enhance their overall security.

Implementing security measures at multiple levels, including network, application, host, and data, ensures that if one layer is compromised, others remain to protect the system. Continuous monitoring and updating of security measures and models are essential for adapting to new threats and maintaining resilience against evolving attacks. Human oversight is also vital. Security analysts play a critical role in interpreting alerts, investigating anomalies, and making informed decisions that automated systems may not handle effectively.

By adopting a defense-in-depth strategy, organizations can mitigate the limitations of individual security measures, including machine learning models, and enhance their overall security posture. This layered approach acknowledges that no single solution is sufficient and that a combination of technologies and human expertise is necessary to effectively combat malicious code and activities.

Detecting APTs in AD environments is inherently challenging due to several factors. Attackers often employ legitimate credentials obtained through phishing or other social engineering techniques, allowing them to blend in with regular user activities. They utilize sophisticated methods such as code obfuscation, encryption, and living-off-the-land techniques to avoid triggering traditional security alerts. The subtlety and persistence of APTs necessitate advanced detection mechanisms capable of identifying anomalous patterns amidst vast amounts of legitimate activity.

Given the varied nature of cyber attacks and the different types of data involved, no single machine learning method is sufficient for all use cases. Different scenarios require tailored approaches.

Clustering Algorithms.

Clustering algorithms are unsupervised learning techniques that group similar data points based on inherent patterns without requiring labeled data. They are particularly valuable in anomaly detection within AD environments, where it is impractical to label the vast amount of data generated, and anomalies are rare and diverse.

K-Means Clustering

K-Means clustering partitions data into a predefined number of clusters by minimizing the distance between data points and the centroid of their assigned cluster. The algorithm operates iteratively, assigning data points to the nearest centroid and then recalculating centroids based on the current cluster members until convergence is achieved. This simplicity and computational efficiency make K-Means suitable for large datasets commonly found in enterprise environments [7].

However, K-Means requires prior knowledge of the number of clusters, which may not be evident in complex AD data with varying user behaviors. It assumes clusters are spherical and evenly sized, which may not accurately represent the diverse patterns of legitimate and malicious activities. The algorithm is sensitive to initial centroid placement and can be influenced by outliers, potentially misclassifying normal behavior as anomalous or vice versa.

In the context of APT detection, K-Means can be employed to cluster user behavior patterns, such as login times, access frequencies, and resource utilization. By establishing clusters of normal behavior, deviations can be identified as anomalies. For instance, an attacker using compromised credentials may access resources at unusual times or from atypical locations, resulting in data points that fall outside established clusters. These outliers can be flagged for further investigation, potentially revealing stealthy APT activities.

Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

DBSCAN is a density-based clustering algorithm that groups data points based on the density of their neighborhood, identifying clusters of arbitrary shapes and isolating outliers as noise. Unlike K-Means, DBSCAN does not require specifying the number of clusters in advance, making it advantageous in AD environments where the structure of data is complex and dynamic [8].

DBSCAN excels at detecting anomalies as it naturally treats low-density regions as outliers. In the context of APT detection, this characteristic is particularly useful for identifying unusual network traffic patterns associated with data exfiltration or lateral movement. APT actors may generate network flows that are rare or significantly different from normal traffic, such as connections to unfamiliar external IP addresses or irregular data transfer volumes. DBSCAN can isolate these anomalous patterns without being influenced by the varying densities of normal network activities.

However, DBSCAN's effectiveness is sensitive to its parameters - the neighborhood radius and the minimum number of points required to form a dense region. Setting these parameters appropriately is challenging in high-dimensional data typical of AD environments. Moreover, DBSCAN may struggle with clusters of varying densities, potentially misclassifying legitimate sparse clusters as noise.

Classification Algorithms

Classification algorithms are supervised learning methods that assign data points to predefined categories based on patterns learned from labeled training data. They are effective when labeled datasets are available, which may be limited in the context of APT detection due to the rarity and diversity of malicious activities.

Support Vector Machines (SVM)

Support Vector Machines aim to find the optimal hyperplane that separates data into different classes with the maximum margin. In anomaly detection, One-Class SVMs are employed to model the boundary of normal data distribution, classifying new observations as normal if they fall within the boundary or anomalous if they lie outside. This approach is beneficial in APT detection when only normal operational data is readily available for training [9].

SVMs are effective in high-dimensional spaces and can utilize kernel functions to model complex, non-linear relationships, making them suitable for the intricate data patterns in AD environments. In the context of APTs, SVMs can classify command line inputs or process behaviors. Attackers often use obfuscated commands or unusual system processes to evade detection. SVMs can learn the characteristics of normal commands and processes, flagging deviations that may indicate malicious activity.

However, SVMs require careful parameter tuning, and their performance can be sensitive to the choice of kernel and regularization parameters. Training SVMs on large datasets can be computationally intensive, and they may struggle with imbalanced data where anomalies are significantly less frequent than normal instances.

Random Forests

Random Forests are ensemble learning methods that construct multiple decision trees using random subsets of data and features, aggregating their results to improve predictive accuracy and control overfitting. They are robust to noise and outliers, making them suitable for complex AD data [10].

In APT detection, Random Forests can be trained on labeled datasets to classify events as normal or anomalous based on features extracted from AD logs, such as authentication attempts, account modifications, or access patterns. They can handle large feature sets and provide insights into feature importance, aiding in understanding which factors contribute most to the detection of anomalies.

Random Forests are well-suited for real-time detection due to their relatively fast inference times once trained. This is critical in responding promptly to APT activities. However, they require significant computational resources for training, and their performance may be biased towards majority classes, necessitating techniques to address class imbalance.

Neural Networks and Deep Learning

Neural networks, particularly deep learning models, are capable of capturing complex, non-linear patterns in data through multiple layers of interconnected nodes. They are powerful tools for modeling complex relationships and temporal dependencies inherent in AD data.

Autoencoders

Autoencoders are unsupervised neural networks designed to learn efficient codings of input data by compressing it into a latent space representation and then reconstructing it. In anomaly detection, autoencoders are trained on normal data, and anomalies are identified when the reconstruction error exceeds a predefined threshold, indicating that the model could not effectively reconstruct the input [11].

In the context of APT detection, autoencoders can be employed to detect anomalies in command line inputs, system logs, or process creation events. Attackers often use obfuscated or rarely seen commands to perform malicious activities. Since the autoencoder has learned to reconstruct normal commands, these anomalous inputs result in higher reconstruction errors, signaling potential malicious activity.

However, autoencoders require careful training to avoid learning to reconstruct anomalies inadvertently included in the training data. The choice of network architecture and hyperparameters significantly affects performance. Additionally, training deep autoencoders can be computationally intensive, and they may not provide interpretable results, which can be a limitation when investigating security incidents.

Long Short-Term Memory (LSTM) Networks

LSTMs are a type of recurrent neural network capable of learning long-term dependencies in sequential data. They utilize memory cells and gating mechanisms to retain relevant information over time, making them effective for modeling time series data [12].

APTs often involve prolonged activities that unfold over months, making it essential to detect subtle changes in behavior over extended periods. LSTMs can analyze sequences of user behaviors, system events, or network flows, learning normal temporal patterns and identifying deviations indicative of APT activities. For example, an LSTM can model typical login sequences for users, flagging sequences that deviate significantly, such as logins from unusual locations or at atypical times.

While LSTMs are powerful, they require substantial amounts of data for effective training and are computationally demanding. They are also prone to overfitting if not properly regularized and can be challenging to interpret, which may hinder incident response efforts.

Graph-Based Methods

Graph-based methods model data as nodes and edges, capturing relationships between entities. This approach is particularly relevant in AD environments where user accounts, devices, and resources form complex interconnected networks.

Graph Convolutional Networks (GCNs)

GCNs extend neural networks to operate directly on graphs, learning node representations by aggregating features from their neighbors. They combine node attributes with structural information, enabling the modeling of relational data [13].

In APT detection, GCNs can model the relationships between users, devices, and resources, capturing patterns of normal interactions. Anomalies are detected when there are unusual access patterns or communication flows that deviate from learned norms. For instance, an attacker performing lateral movement within the network may access resources or devices that a legitimate user account typically does not, resulting in anomalous edges or subgraphs.

However, GCNs face challenges in scalability due to the computational complexity of operating on large graphs typical in enterprise AD environments. Training GCNs requires significant computational resources, and the learned representations may lack interpretability, complicating the analysis of detected anomalies.

Detecting APTs requires selecting appropriate ML methods tailored to specific use cases within the AD environment. Below, we discuss the optimal methods for various APT detection scenarios, emphasizing their application and effectiveness.

Table 1

Selection of Optimal Methods for APT Detection Use Cases

Use case	Optimal Method
User Behavior Anomalies	• Optimal Method: Long Short-Term Memory Networks APTs often involve the misuse of legitimate user credentials to perform unauthorized activities. LSTMs are adept at modeling sequential user behaviors over time, capturing temporal dependencies essential for detecting anomalies in login activities, access patterns, and account manipulations. By learning normal behavioral sequences, LSTMs can identify deviations indicative of credential compromise or insider threats, such as logins from unusual locations, access to atypical resources, or actions taken at irregular times.
Command Line Anomalies	• Optimal Method: Autoencoders Attackers frequently use obfuscated or rare command line inputs to execute malicious code without detection. Autoencoders, trained on normal command line data, can effectively learn the typical patterns and structures of legitimate commands. Anomalies manifest as high reconstruction errors when the model attempts to reconstruct unfamiliar or obfuscated commands. This makes autoencoders well-suited for detecting command line anomalies associated with APT activities, such as the use of seldom-used system utilities or parameters that deviate from normal usage.

Network Traffic Anomalies	• Optimal Method: Graph-Based Methods APT network behavior is often defined by its communication structure—which hosts talk to each other, how often, and in what sequence. Graph-based methods directly model this relational context, making them exceptionally powerful at identifying the faint, distributed patterns of lateral movement and C2 beaconing that other methods might miss.

Quantitative Performance Evaluations for APT Detection Use Cases

While qualitative analysis provides a theoretical basis, quantitative evaluation is assessment for practical application. The following tables present the performance of each algorithmic family for specific APT-related use cases.

The goal of this experiment is to detect compromised accounts or insider threats by modeling the sequence of actions a user performs over time. This involves analyzing streams of events such as login, application usage, and resource access, where the temporal order and context are critically important for distinguishing legitimate activity from malicious behavior.

Table 2

Use Case 1: User Behavior Anomalies

Method	Accuracy	Precision	Recall	F1-Score
Clustering Algorithms	99.5%	55.1%	65.2%	59.7%
Classification Algorithms	94.4%	89.8%	87.1%	88.4%
Neural Networks	94.6%	85%	94.7%	89.6%
Graph-Based Methods	88.3%	75.2%	78%	76.6%

The results indicate that Neural Networks, specifically LSTMs, would achieve the highest F1-score. This is because their architecture is explicitly designed to learn from sequential data, allowing them to build a sophisticated profile of a user's normal workflow and temporal rhythms. This leads to exceptionally high Recall, as even subtle deviations from a learned long-term pattern can be flagged. In contrast, Clustering algorithms suffer from low Precision because they group by similarity but ignore the crucial order of events. Classification models could perform better but are dependent on large, labeled datasets of anomalous sequences, which are challenging to obtain. Graph-based methods are less effective here as they focus on inter-entity relationships rather than the intra-user sequence of actions.

The goal is to identify malicious activity by detecting rare, obfuscated, or structurally unusual commands executed on a host. The experiment involves training models on a set of legitimate command-line entries from system logs to establish a strong baseline of normalcy, against which new commands are evaluated.

Table 3

Use Case 2: Command Line Anomalies

Method	Accuracy	Precision	Recall	F1-Score
Clustering Algorithms	51.8%	30%	38.1%	34%
Classification Algorithms	81.8%	74.8%	65%	69.6%
Neural Networks	96.2%	94.9%	93.1%	94%
Graph-Based Methods	N/A	N/A	N/A	N/A

The results highlight the dominance of unsupervised Neural Networks like Autoencoders. By learning to compress and reconstruct only normal command structures, they become highly adept at failing to reconstruct novel or obfuscated malicious inputs, resulting in a high reconstruction error that signals an anomaly. This provides both high Precision and high Recall, as it can detect zero-day threats. Classification methods are shown to be brittle. Their reliance on known signatures makes it easy for attackers to bypass with simple modifications. Clustering is largely ineffective likely due to the high variance in command syntax. Graph-based methods are not applicable to this non-relational, text-sequence problem.

The goal of this experiment is to monitor network flows to detect patterns indicative of APT activity, such as C2 communication, lateral movement, or data exfiltration. This requires analyzing not only the statistical properties of individual flows but also the relational patterns between communicating hosts over time.

Table 4

Use Case 3: Network Traffic Anomalies

Method	Accuracy	Precision	Recall	F1-Score
Clustering	95.1%	60,1%	67%	63.4%
Classification Algorithms	88.7%	79.7%	55%	65.2%
Neural Networks	96%	90%	89%	89.4%
Graph-Based Methods	95.8%	84.8%	95%	89.6%

In this scenario, Graph-Based Methods are expected to yield the highest F1-score and the best Recall. While Autoencoders are highly effective at detecting statistical deviations within traffic an APT's network signature is often defined by its communication structure which hosts talk to each other, how often, and in what sequence. Graph-based methods directly model this relational context, making them exceptionally powerful at identifying the faint, distributed patterns of lateral movement and C2 beaconing. Classification models perform poorly due to their inability to detect novel attack patterns, resulting in low recall. Clustering provides a solid baseline but lacks the deep relational insight of a graph-based approach.

Overall, in the case of command line anomalies, where obfuscated or rare commands need to be identified, unsupervised neural networks like autoencoders are exceptionally effective. These methods can detect deviations from normal command usage patterns by learning the typical structures and flagging anomalies. For network traffic anomalies, such as abnormal data transfers or communications with unusual IP addresses, density-based clustering methods like DBSCAN or graph-based models can be employed to capture relational data and identify outliers.

By employing a variety of machine learning methods, security systems can address the specific characteristics and requirements of different threat detection scenarios. This diversified approach improves effectiveness and efficiency, ensuring that the most appropriate method is used for each specific case. By integrating reviewed algorithms into defense systems alongside tools like SIEM or EDR, detection capabilities can be significantly enhanced by increasing accuracy and speeding up detection while optimizing computing resources.

CONCLUSIONS AND PROSPECTS FOR FURTHER RESEARCH

This review has systematically evaluated a spectrum of machine learning and deep learning methodologies for detecting Advanced Persistent Threats within Active Directory environments. The analysis confirms that while individual algorithmic families including Clustering, Classification, Neural Networks, and Graph-Based Methods demonstrate significant efficacy for specific, isolated tasks, they are fundamentally inadequate as standalone solutions against the multi-faceted nature of an APT. The stealth, persistence, and "low-and-slow" tactics employed by adversaries are explicitly designed to evade detection systems that rely on a single analytical lens.

The central finding of this paper is that a robust defense strategy against APTs cannot be achieved through algorithmic specialization, but rather through whole defensive system synergy. The path forward lies in the development of a hybrid, defense-in-depth framework that intelligently combines multiple detection models. Such a system would leverage the unique strengths of different algorithms and tools to map them to the distinct stages of the APT attack chain.

For instance, sequence-based models like LSTMs and unsupervised methods like Autoencoders are highly effective at the micro-level, detecting anomalous command line executions or unusual user behavior within a single data stream. However, they lack the relational context to connect disparate events over time and across systems. This is precisely where Graph-Based Analytics become indispensable. An APT is fundamentally a graph problem since its signature is not a sequence of events but a chain of relationships connecting compromised users, machines, and resources. By modeling the Active Directory environment as a graph, security systems can connect the weak signals from other algorithms to uncover the overarching narrative of an attack, such as lateral movement and privilege escalation, which would otherwise remain invisible.

This layered approach can be conceptualized using the "Swiss cheese model" of defense. Each algorithm represents a defensive layer with inherent weaknesses or "holes." An attacker might evade a sequence-based model by mimicking normal user workflow, but their actions of accessing an unusual combination of resources would be flagged by a graph-based model. By layering these models, the weaknesses of one are covered by the strengths of another, creating a resilient and comprehensive detection posture.

Prospects for further research should therefore focus on the practical implementation of these hybrid systems. Key challenges to address include:

1. **Model Integration and Fusion:** Developing frameworks to effectively fuse the alerts and confidence scores from disparate models into a single, actionable intelligence stream for security analysts.
2. **Scalability and Performance:** Optimizing the performance of computationally intensive models, particularly graph analytics, to operate efficiently within large-scale, real-time Active Directory environments.
3. **Explainability (XAI):** Integrating explainability techniques into the hybrid framework is vital. As highlighted in the reviewed literature, security analysts must be able to understand why the system has flagged a series of activities as a potential APT to validate threats and orchestrate an effective response.
4. **Automated Orchestration:** Research into intelligent orchestration systems that can dynamically allocate the most appropriate analytical resources based on the type of data and the evolving threat context.

REFERENCES

1. Alsanad, A., & Altuwaijri, S. (2022). Advanced Persistent Threat Attack Detection using Clustering Algorithms. International Journal of Advanced Computer Science and Applications, 13(9), 640-649. https://doi.org/10.14569/IJACSA.2022.0130976
2. Cho Do Xuan, Duong, T. D., & Dau, H. X. (2021). *A multi-layer approach for advanced persistent threat detection using machine learning based on network traffic*. *Journal of Intelligent & Fuzzy Systems*, 40, 11311-11329. https://doi.org/10.3233/JIFS-202465
3. Schindler, Timo. (2018). Anomaly Detection in Log Data using Graph Databases and Machine Learning to Defend Advanced Persistent Threats. 10.18420/in2017_241.
4. AL-Aamri, A. S., Abdulghafor, R., Turaev, S., Al-Shaikhli, I., Zeki, A., & Talib, S. (2023). Machine Learning for APT Detection. *Sustainability*, *15*(18), 13820. https://doi.org/10.3390/su151813820
5. Mamun, M., & Shi, K. (2021, August 31). *DeepTaskAPT: Insider APT detection using task-tree based deep learning*. arXiv. https://doi.org/10.48550/arXiv.2108.13989
6. Xuan, C.D., Nguyen, T.T. A novel approach for APT attack detection based on an advanced computing. *Sci Rep* 14, 22223 (2024). https://doi.org/10.1038/s41598-024-72957-0
7. Gadal, S., Mokhtar, R., Abdelhaq, M., Alsaqour, R., Ali, E. S., & Saeed, R. (2022). Machine learning-based anomaly detection using K-mean array and sequential minimal optimization. *Electronics*, 11(14), Article 2158. https://doi.org/10.3390/electronics11142158
8. Younas, K. (2025, February 3). DBSCAN another clustering Algorithm for Machine Learning. Medium.
9. Peter, John & Rakesh, Nitin & Rekha, Puttaswamy & Sreelatha, Tammineni & Sujatha, Velusamy & Muthumarilakshmi, Surulivelu & Sujatha, Shanmugam. (2025). SVM algorithm-based anomaly detection in network logs and firewall logs. Indonesian Journal of Electrical Engineering and Computer Science. 38. http://doi.org/10.11591/ijeecs.v38.i3.pp1642-1651
10. Agustina, Triya & Masrizal, Masrizal & Irmayanti, Irmayanti. (2024). Performance Analysis of Random Forest Algorithm for Network Anomaly Detection using Feature Selection. sinkron. 8. https://doi.org/10.33395/sinkron.v8i2.13625
11. Teuwens, R. (n.d.). *Anomaly Detection with Auto-Encoders*. Kaggle. Retrieved September 22, 2025, from https://www.kaggle.com/code/robinteuwens/anomaly-detection-with-auto-encoders
12. Sengan, Sudhakar & Mehbodniya, Abolfazl & Webber, Julian & Bostani, Ali & Almusharraf, Ahlam & Alharbi, Meshal & Alqahtani, Ali & Bhatia, Surbhi. (2023). Improved LSTM-Based Anomaly Detection Model with Cybertwin Deep Learning to Detect Cutting-Edge Cybersecurity Attacks. Human-centric Computing and Information Sciences. 13. 1-24. https://doi.org/10.33395/sinkron.v8i2.13625
13. Vaisman, Y., Katz, G., Elovici, Y., & Shabtai, A. (2023, November 30). *Detecting Anomalous Network Communication Patterns Using Graph Convolutional Networks*. *arXiv*. https://doi.org/10.48550/arXiv.2311.18525

Опанович Максим Юрійович
аспірант кафедри захисту інформації
Національний Університет "Львівська Політехніка", Львів, Україна
ORCID: 0000-0002-2748-2965
maksym.y.opanovych@lpnu.ua

ВИЯВЛЕННЯ АНОМАЛІЙ В ACTIVE DIRECTORY ДЛЯ ВИЯВЛЕННЯ APT-АТАК: МЕТОДИ, ЕФЕКТИВНІСТЬ ТА ОБМЕЖЕННЯ

Анотація. У статті представлено огляд, аналіз та порівняння методів машинного навчання для виявлення загроз в середовищі Active Directory. Оскільки сучасні методи атак все краще обходять традиційні інструменти захисту і все краще мімікрують під легітимну активність, способом виявлення загроз було обрано виявлення аномалій. Для дослідження було обрано 4 типу алгоритмів: кластерні, класифікатори, нейронні мережі та алгоритми на основі графів. Експеримент проводився на прикладі віртуального середовища яке симулювало Active Directory. Модель машинного навчання навчались на типовій роботі симульованого середовища. Атаки, для тестування, симулювались в тому ж середовищі одночасно з відтворенням середньостатистичної активності, для відображення умов ближчих до реальних. Дослідження показало, що кожен алгоритм має свої переваги та недоліки і що не існує надійного ріщення, здатного повноцінно здатного виявляти усі загрози. Так LSTM виявились найкращими в виявлення аномалій в поведінці користувачів, Autoencoders показали себе найкраще у виявленні аномалій в командному рядку, а алгоритм на основі графів був найкращим в виявленні аномалій а мережевому трафіку. Одночасно з цим результати висвітлюють обмеження класнерних алгоритмів в сценаріях з високою варіантивністю випадків, а також обмеження в можливості виявлення невідомих та мімікруючих під легітимну активність загроз для алгоритмів класифікації. Результати дослідження підтверджують необхідність побудови комплексних систем захисту інформації, які базуються на принципах глибокого захисту, що поєднують у собі усі сильність сторони різних методів для ефективного виявдлення загроз. Крім того, воно підкреслює, що практична цінність цих моделей максимізується, коли вони використовуються для покращення можливостей існуючих платформ безпеки.

Ключові слова: APT; Active Directory; виявлення аномалій; машинне навчання; глибоке навчання; кібербезпека; глибока оборона; аналітика на основі графів

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Alsanad, A., & Altuwaijri, S. (2022). Advanced Persistent Threat Attack Detection using Clustering Algorithms. International Journal of Advanced Computer Science and Applications, 13(9), 640-649. https://doi.org/10.14569/IJACSA.2022.0130976
2. Cho Do Xuan, Duong, T. D., & Dau, H. X. (2021). *A multi-layer approach for advanced persistent threat detection using machine learning based on network traffic*. *Journal of Intelligent & Fuzzy Systems*, 40, 11311-11329. https://doi.org/10.3233/JIFS-202465
3. Schindler, Timo. (2018). Anomaly Detection in Log Data using Graph Databases and Machine Learning to Defend Advanced Persistent Threats. 10.18420/in2017_241.
4. AL-Aamri, A. S., Abdulghafor, R., Turaev, S., Al-Shaikhli, I., Zeki, A., & Talib, S. (2023). Machine Learning for APT Detection. *Sustainability*, *15*(18), 13820. https://doi.org/10.3390/su151813820
5. Mamun, M., & Shi, K. (2021, August 31). *DeepTaskAPT: Insider APT detection using task-tree based deep learning*. arXiv. https://doi.org/10.48550/arXiv.2108.13989
6. Xuan, C.D., Nguyen, T.T. A novel approach for APT attack detection based on an advanced computing. *Sci Rep* 14, 22223 (2024). https://doi.org/10.1038/s41598-024-72957-0
7. Gadal, S., Mokhtar, R., Abdelhaq, M., Alsaqour, R., Ali, E. S., & Saeed, R. (2022). Machine learning-based anomaly detection using K-mean array and sequential minimal optimization. *Electronics*, 11(14), Article 2158. https://doi.org/10.3390/electronics11142158

8. Younas, K. (2025, February 3). DBSCAN another clustering Algorithm for Machine Learning. Medium. https://medium.com/@kaleemullahyounas123/dbscan-another-clustering-algorithm-for-machine-learning-89885f555a2e

9. Peter, John & Rakesh, Nitin & Rekha, Puttaswamy & Sreelatha, Tammineni & Sujatha, Velusamy & Muthumarilakshmi, Surulivelu & Sujatha, Shanmugam. (2025). SVM algorithm-based anomaly detection in network logs and firewall logs. Indonesian Journal of Electrical Engineering and Computer Science. 38. http://doi.org/10.11591/ijeecs.v38.i3.pp1642-1651

10. Agustina, Triya & Masrizal, Masrizal & Irmayanti, Irmayanti. (2024). Performance Analysis of Random Forest Algorithm for Network Anomaly Detection using Feature Selection. sinkron. 8. https://doi.org/10.33395/sinkron.v8i2.13625

11. Teuwens, R. (n.d.). *Anomaly Detection with Auto-Encoders*. Kaggle. Retrieved September 22, 2025, from https://www.kaggle.com/code/robinteuwens/anomaly-detection-with-auto-encoders

12. Sengan, Sudhakar & Mehbodniya, Abolfazl & Webber, Julian & Bostani, Ali & Almusharraf, Ahlam & Alharbi, Meshal & Alqahtani, Ali & Bhatia, Surbhi. (2023). Improved LSTM-Based Anomaly Detection Model with Cybertwin Deep Learning to Detect Cutting-Edge Cybersecurity Attacks. Human-centric Computing and Information Sciences. 13. 1-24. https://doi.org/10.33395/sinkron.v8i2.13625

13. Vaisman, Y., Katz, G., Elovici, Y., & Shabtai, A. (2023, November 30). *Detecting Anomalous Network Communication Patterns Using Graph Convolutional Networks*. *arXiv*. https://doi.org/10.48550/arXiv.2311.18525