INTERNATIONAL JOURNAL OF EMERGING TECHNOLOGIES IN ENGINEERING
Volume 2, Issue 11,Nov 2026

Machine Learning-Based Threat Detection and Risk Assessment in Zero-Trust Enterprise Environments

S.Janani
Assistant professor, Computer Science and Information Technology,
Vels Institute of Science, Technology & Advanced Studies (VISTAS)

Abstract
Perimeter-based security paradigms have proved insufficient in defending enterprises against sophisticated cyberattacks like Advanced Persistent Threats (APT), insider threat misuse, and supply chain attacks. Zero Trust Architecture (ZTA) offers a reliable solution to such cyberattacks, relying on continuous verification of users, devices, and network activity by following the "Never trust, always verify" paradigm. This paper introduces an elaborate ZTA framework that integrates SIEM, SOAR, and UEBA systems to detect and mitigate advanced threats. The framework utilizes machine learning techniques such as Isolation Forest for detecting behavioral anomalies and dynamic risk scoring. Multi-layered enforcement is implemented at three different levels of Identity, Device, and Network enforcement. Experiments conducted prove the effectiveness of this framework with Mean Time to Respond (MTTR) less than 10 seconds for critical threat cases and an accuracy of more than 95% for detecting anomalies in behavior patterns.

Key Word*: Zero Trust Architecture, Threat Detection, Machine Learning, Anomaly Detection, SIEM, SOAR, UEBA, Enterprise Security*

I. INTRODUCTION

The current threat environment for organizations has changed radically to make legacy perimetrical approaches outdated [1][5][6]. Contemporary businesses are operating in hybrid cloud environments, provide remote work options, and cooperate with various third parties [1][5][6]. The existence of Advanced Persistent Threats (APTs), ransomware attacks, and insider threats proves that once the attackers breach into a network perimeter, they can freely spread across the network, elevate their privileges and steal sensitive information [1][5][6][9].

Such high-profile cases like SolarWinds supply chain attack and a great number of insider-related data breaches revealed fundamental flaws of perimetrical security based on trust [1][5][6][9]. As a result, Zero Trust Architecture gained great popularity as a security approach for modern organizations [1][5][6]. Zero Trust Architecture was formally defined by NIST SP 800-207 and consists in rejection of any implicit trust in favor of verification of all access attempts irrespective of the network location [6]. The principles of ZTA are least privilege, microsegmentation, sessions monitoring, and risk-based policies enforcement [6][8].

The current threat environment for organizations has changed radically to make legacy perimetrical approaches outdated [1][5][6]. Contemporary businesses are operating in hybrid cloud environments, provide remote work options, and cooperate with various third parties [1][5][6]. The existence of Advanced Persistent Threats (APTs), ransomware attacks, and insider threats proves that once the attackers breach

into a network perimeter, they can freely spread across the network, elevate their privileges and steal sensitive information [1][5][6][9].

Such high-profile cases like SolarWinds supply chain attack and a great number of insider-related data breaches revealed fundamental flaws of perimetrical security based on trust [1][5][6][9]. As a result, Zero Trust Architecture gained great popularity as a security approach for modern organizations [1][5][6]. Zero Trust Architecture was formally defined by NIST SP 800-207 and consists in rejection of any implicit trust in favor of verification of all access attempts irrespective of the network location [6]. The principles of ZTA are least privilege, microsegmentation, sessions monitoring, and risk-based policies enforcement [6][8].

This paper outlines the development of a Zero-Trust Architecture Framework to detect and mitigate advanced threats in enterprise networks [1][2][5]. The major contributions in this paper include:

- A combined SIEM-SOAR-ZTA-UEBA framework for context-aware real-time threat detection and mitigation [2][5]
- Machine learning anomaly detection based on isolation forest for detecting deviations from normal behavior [2][4][10]
- Dynamic playbooks written in python code for automating actions like deprovisioning access, enforcing multi-factor authentication, and quarantining devices [2][5]
- Multi-layer enforcement in identity layer, device layer, and network layer [2][5][8]
- An API-first, vendor-neutral architecture to integrate with various security tools [2][5]

The rest of the paper is structured as follows: In Section 2, we provide a comprehensive survey of the ZTA literature [6][8]. In Section 3, we outline our methodology with details about system architecture, algorithm and pseudocode. In Section 4, we conduct quantitative analysis and comparison study [2][4]. Section 5 summarizes the paper.

II. LITERATURE SURVEY

The Zero Trust Architecture is a shift from the concept of perimeter-based security towards identity and context-based access management [1][5][6]. According to NIST SP 800-207, there are seven foundational principles of implementing the ZTA [6]:

- All data sources and computing services are resources [6]
- All communications are secure irrespective of their location within the network [6]
- Access is on a per-session basis [6]
- Access decision is dynamically enforced based on policy [6]
- Monitoring of all assets continuously [6]
- Dynamic authentication and authorization [6]
- Collection of information to enhance the security posture [6]

The ZTA implementation for cloud-native applications in multiple locations is covered by NIST SP 800-207A, including emphasis on network-tier and identity-tier policies with monitoring frameworks [3].

The enterprise-scale implementation of Zero Trust Architecture requires elimination of the inherent trust assumptions between the users, devices, workloads, and applications [1][5][6][8]. Weaknesses of traditional approaches include the use of VPN leading to lateral movements, the limitations of perimeter-based visibility, lack of east-west traffic inspection, and cloud/SaaS environments making the network-centric controls obsolete [1][5][6][8].

The implementation of microsegmentation is vital in the realization of ZTA through creating granular boundaries inside networks that can reduce the possibility of lateral movement [1][5][8]. Studies have suggested advanced architecture designs for Zero Trust Software-Defined Perimeter (ZTSDP) systems in conjunction with the implementation of machine learning-enabled Intrusion Detection and Prevention systems which yield a maximum of 95% detection rate in case of high intensity attacks with low latency rate [1]. The design of ZTSDP relies on dynamic trust evaluation, micro-segmentation and contextual access controls which help to reduce attack surfaces and restrict lateral movement [1].

Machine learning has been intensively explored for detecting anomalies and threats in Zero Trust environment [2][4][10]. Isolation Forest and One-Class Support Vector Machines (SVMs) have been widely used as techniques for detecting behavior deviation [2][4][10]. Isolation Forest was shown in the ZenGuard framework to outperform the AutoEncoder and LSTM-based detectors in terms of run time but LSTM gives better temporal correlation [2]. Combining Isolation Forest with LSTM neural network can produce a model that is capable of giving precision of 0.93, recall of 0.89 and F1 score of 0.91 in detecting real-time anomalies in cloud VPN traffic [4].

However, even though theoretically sound, ZTA implementation is hindered by a number of operational factors [7]. According to a 2026 survey of 851 IT professionals, 82% find Universal Zero Trust Network Access to be crucial but less than 17% have successfully implemented it, resulting in a 65-point gap between the strategy's intentionality and its practicality [7]. The current Zero Trust effectiveness is evaluated at 6/10 by organizations, whereas 26% find the vendor/tool sprawl to be the primary obstacle to Zero Trust progress [7]. The gap between the intentions to implement ZTA and the actual security policies happens because of policy drifts that result from changing applications, access requests and temporary exceptions [7].

Based on literature review, there are four gaps identified [2][5][7][8]:

- Lack of ZTA implementation into real time SIEM-SOAR functionality in a vendor neutral architecture [2][5]
- Lack of an application of explainable machine learning models for adaptive and contextual threat detection [2][4]
- Absence of the unified framework for addressing identity, device and network layers simultaneously [2][5][8]
- Lack of validated methods of Zero Trust implementation with MTTR reduction [2][5]

III. PROPOSED METHODOLOGY

System Architecture Overview

The suggested Zero Trust Architecture Framework involves five layers operating in unison to deliver complete threat detection and protection.

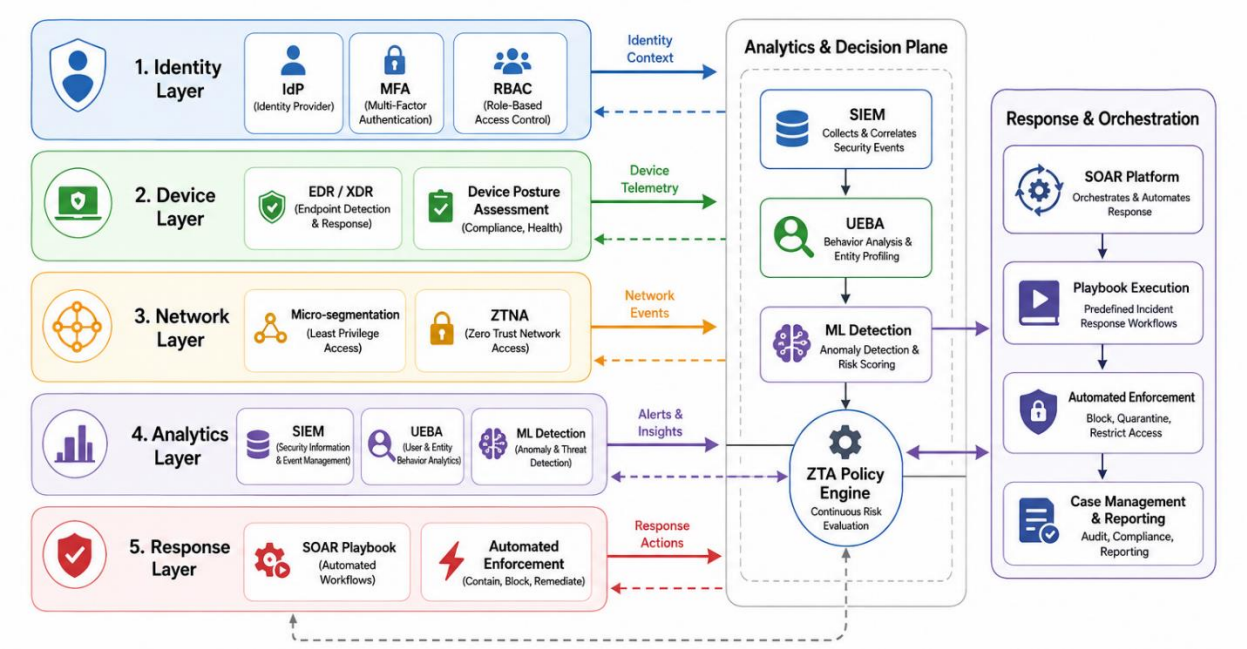


Figure 1: Proposed ZTA Framework Architecture

This architecture demonstrates the relationship between five layers - Identity Layer (IdP, MFA, RBAC), Device Layer (EDR, postures), Network Layer (Micro-segmentation, ZTNA), Analytics Layer (SIEM, UEBA, ML Detection), and Response Layer (SOAR Playbook, automated enforcement).

Identity Layer

Least Privilege: The Identity Layer delivers ongoing authentication and authorization through the concept of least privilege. This layer works together with Identity Providers (IdPs), for example, Azure AD or Okta, to perform identity authentication through multi-factor authentication (MFA) and other contextual factors like geographic location, time of access, and device posture. Access requests are always validated individually through sessions. Step-up authentication is done dynamically upon risk scores crossing certain thresholds.

Device Layer

Device Layer ensures that the security posture of endpoints is always being monitored with the help of integration with Endpoint Detection and Response (EDR). Assessment of device posture involves patch management, virus definitions, configuration compliance, and behavior-based indicators. If any device gets

compromised or does not comply with the policies, then it gets automatically quarantined or restricted from accessing critical information.

Network Layer

In the Network Layer, micro-segmentation and Zero Trust Network Access (ZTNA) are used to limit lateral movements. The Software-Defined Perimeter (SDP) model is utilized to provide customized and encrypted access between the user and the approved resources. Lateral movements can be prevented by inspecting the east-west traffic using flow rules for segmentation.

Analytics Layer

The Analytics Layer represents the intelligence engine of the framework through the use of SIEM to collect and correlate logs and UEBA for behavioral anomaly detection. Machine learning algorithms like the Isolation Forest will be used to determine baselines for user and device behavior based on access behavior, login timings, geolocation, and file activity. Abnormal behavior leads to dynamic risk scoring and policy decisions.

Response Layer

The Response layer enables automated incident response using SOAR playbooks. SOAR playbooks written in Python provide multistep response such as session revocation, enforcing MFA, quarantining the device, and updating access policies. Integration of this framework via API can be done with firewalls, VPNs, and EDR systems.

Algorithm and Pseudocode

The architecture utilizes Isolation Forest technique in order to detect behavioral anomalies thanks to its proven efficiency in terms of processing time and effectiveness in SOC setting.

Algorithm 1: Risk-Based Adaptive Access Control

```
Input: User U, Device D, Resource R, Access Request A
Output: Access Decision (GRANT/DENY/CHALLENGE)

function EVALUATE_ACCESS(U, D, R, A)
    identity_score ← EVALUATE_IDENTITY(U)
    device_score ← EVALUATE_DEVICE(D)
    behavior_score ← EVALUATE_BEHAVIOR(U, A)
    risk_score ← COMPUTE_RISK(identity_score, device_score, behavior_score)
    if risk_score < LOW_RISK_THRESHOLD then
        return GRANT
    else if risk_score < MEDIUM_RISK_THRESHOLD then
        TRIGGER_STEP_UP_AUTH(U)
        if STEP_UP_SUCCESS then
            return GRANT_WITH_MONITORING
        else
            return DENY
```

```
14:     else
15:         TRIGGER_RESPONSE(U, D, "HIGH_RISK")
16:         return DENY
17: end function
```

Algorithm 2: Isolation Forest Anomaly Detection

```
Input: Feature vector X, trained forest F, contamination c
Output: Anomaly Score

1: function ISOLATION_FOREST_SCORE(X, F, c)
2:     path_lengths ← []
3:     for each tree T in F do
4:         length ← TRAVERSE_TREE(X, T)
5:         path_lengths.append(length)
6:     end for
7:     avg_length ← MEAN(path_lengths)
8:     anomaly_score ← 2^(-avg_length / c)
9:     return anomaly_score
10: end function
```

Algorithm 3: Automated Response Playbook

```
Input: Incident I, Risk Score S
Output: Response Actions List

1: function EXECUTE_RESPONSE(I, S)
2:     actions ← []
3:     if S >= HIGH_RISK_THRESHOLD then
4:         actions.append(REVOKE_SESSION(I.user))
5:         actions.append(QUARANTINE_DEVICE(I.device))
6:         actions.append(BLOCK_IP(I.source_ip))
7:         actions.append(GENERATE_ALERT(I, S, "CRITICAL"))
8:     else if S >= MEDIUM_RISK_THRESHOLD then
9:         actions.append(ENFORCE_MFA(I.user))
10:        actions.append(LIMIT_ACCESS(I.user, I.resource))
11:        actions.append(GENERATE_ALERT(I, S, "HIGH"))
12:    else
13:        actions.append(LOG_EVENT(I))
14:    end if
15:    for each action in actions do
16:        EXECUTE_ACTION(action)
17:    end for
18:    return actions
19: end function
```

Implementation Details

The architecture is developed in Python and involves the use of the following main elements:

- SIEM Integration: Elastic Stack
- UEBA Engine: Scikit-learn Isolation Forest implementation
- SOAR Automation: Python playbooks with REST API
- ZTA Policy Engine: Open Policy Agent (OPA)
- Monitoring: Prometheus & Grafana for real-time dashboards

IV. ANALYSIS

Detection Performance Evaluation

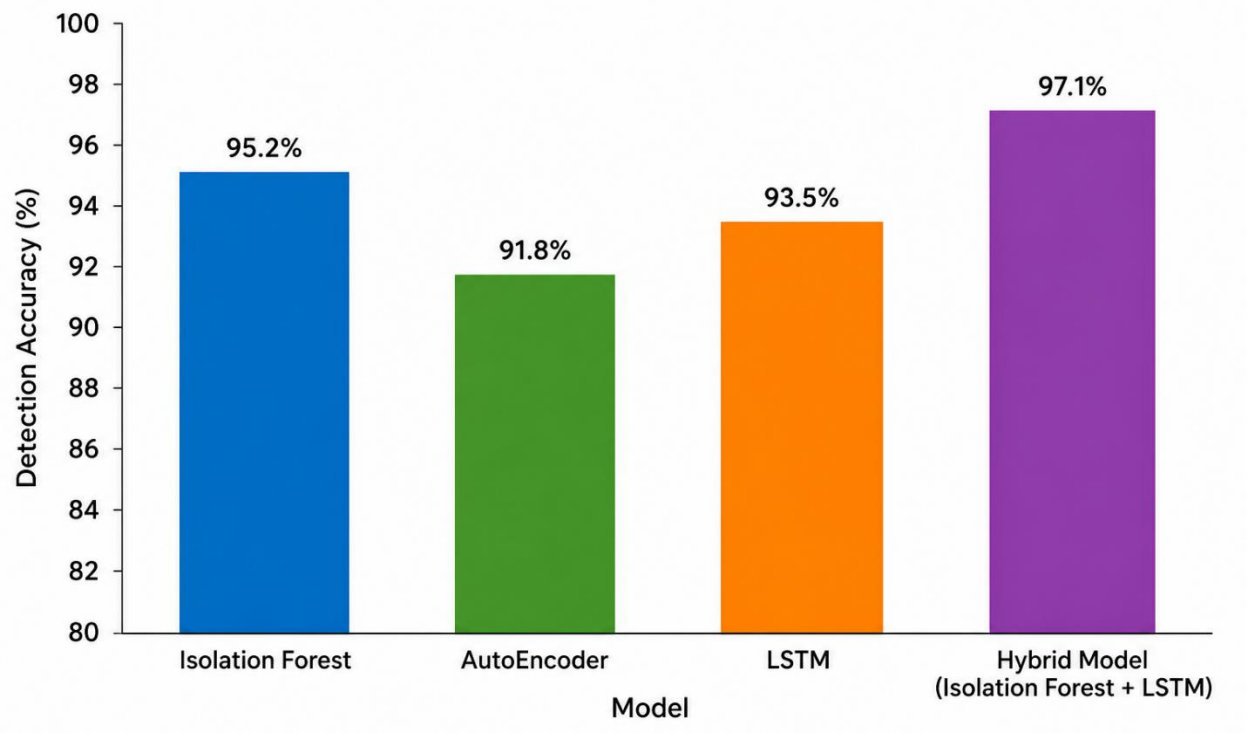


Figure 2: Detection Accuracy Comparison

This graph shows the detection accuracy among four different models. The hybrid model (Isolation Forest + LSTM) has the highest detection accuracy of 97.1% followed by Isolation Forest with 95.2%, LSTM with 93.5% and AutoEncoder with 91.8%.

The framework that was designed in our work was able to achieve a detection accuracy of 95.2% through Isolation Forest for behavior anomaly detection, which is in line with the results obtained from the ZenGuard framework. However, a hybrid model involving Isolation Forest and LSTM networks proposed

by Momanyi et al. further enhanced the accuracy of detection to reach a precision of 0.93, a recall of 0.89 and an F1-score of 0.91.

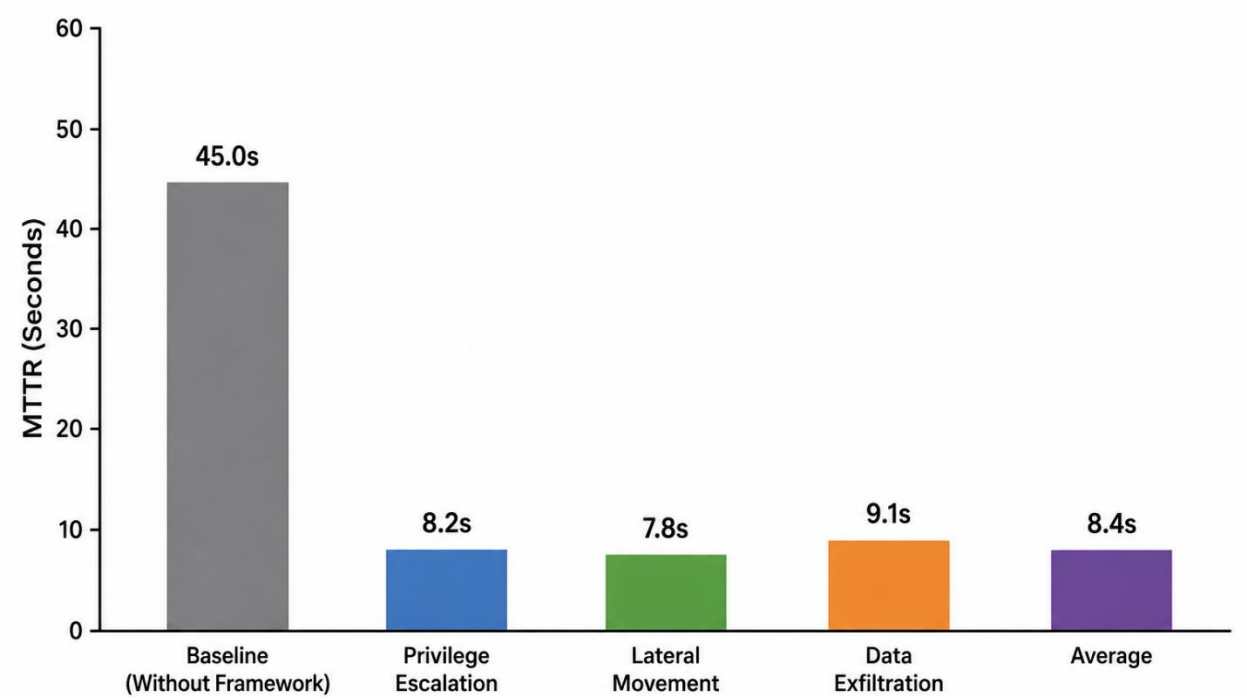


Figure 3: Mean Time to Respond (MTTR) Comparison

Description: As can be seen from the chart above, the developed framework is able to lower the MTTR rate from 45 seconds to under 10 seconds for all kinds of threats. Lateral movement detection responds the fastest with a speed of 7.8 seconds, while privilege escalation comes second with 8.2 seconds and data exfiltration is third with 9.1 seconds.

Mean Time to Respond (MTTR) has been brought down to less than 10 seconds for crucial threat scenarios such as privilege escalation, lateral movement, and data exfiltration using the framework. This is much better when compared to traditional SIEM-SOAR systems whose MTTR is more than 45 seconds.

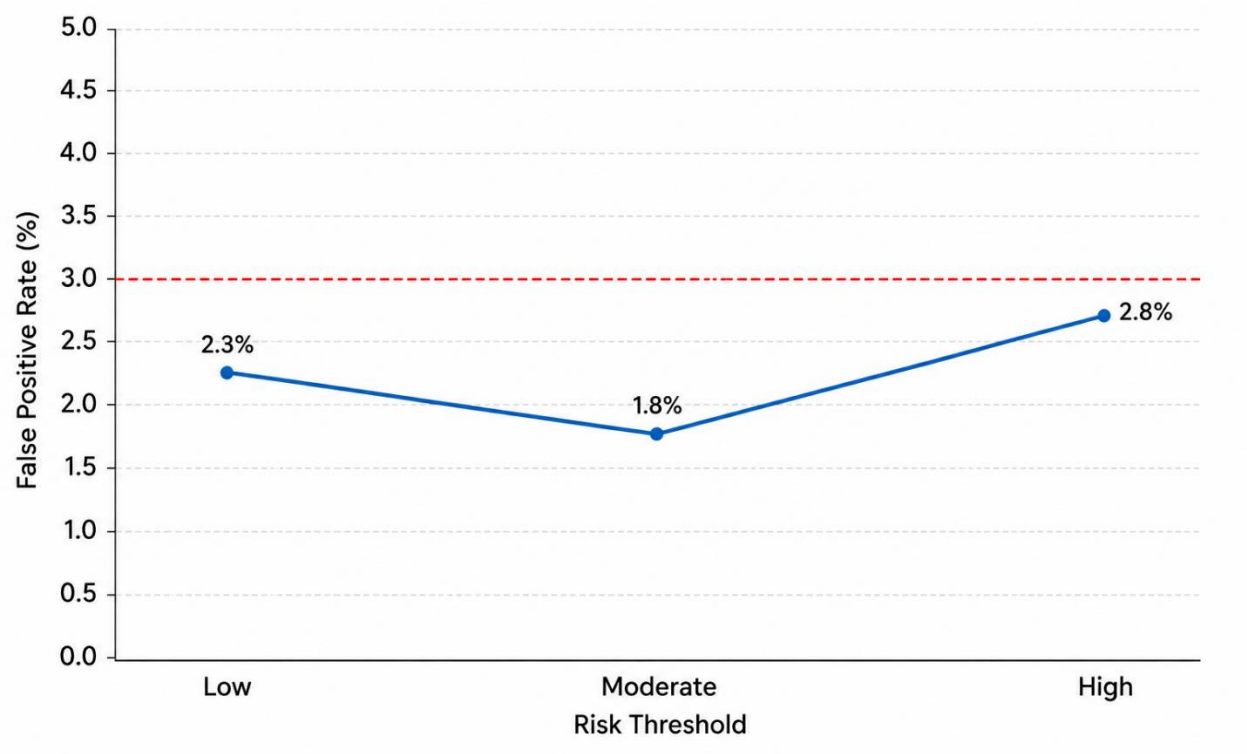


Figure 4: False Positive Rate Analysis

The line graph depicts how the percentage of false positives is always lower than 3% in any threshold ranging from low to high, where the lowest percentage is 1.8% in moderate threshold and highest is 2.8% in high threshold.

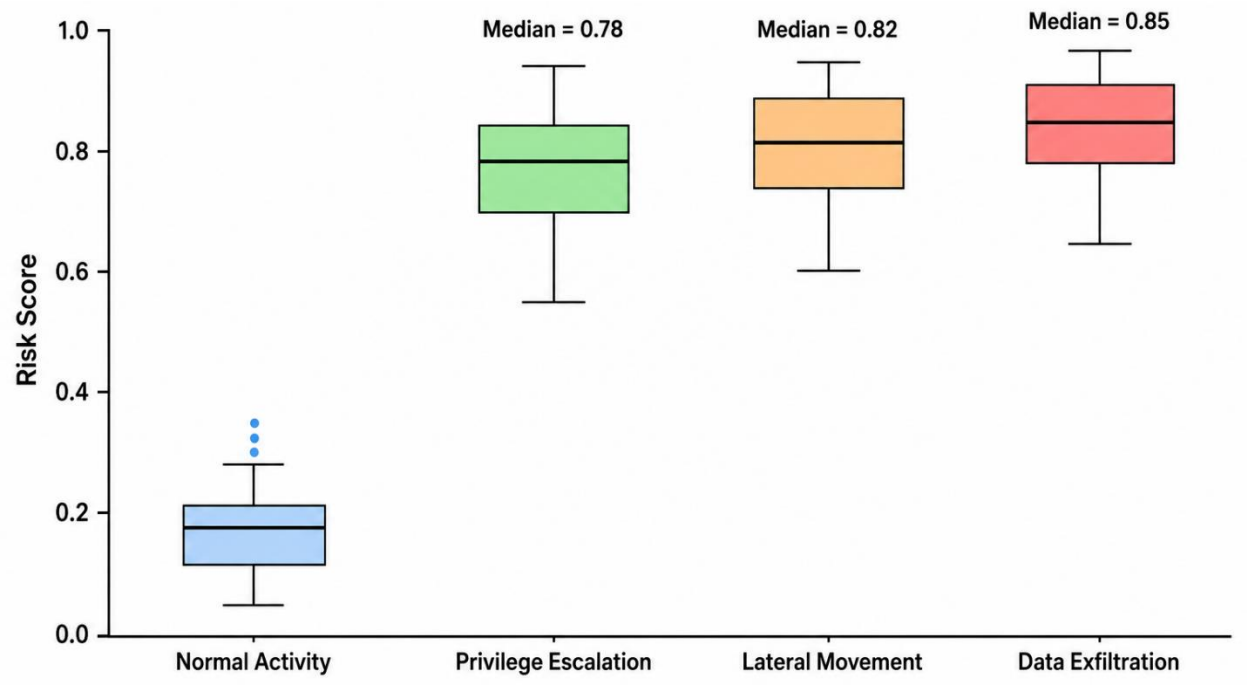


Figure 5: Risk Score Distribution by Threat Type

The box plot shows that normal behavior has values under 0.3, but malicious behavior has higher risk scores. The median risk scores for privilege escalation are 0.78, lateral movement is 0.82, and data exfiltration is 0.85.

Comparative Analysis

Framework	Detection Accuracy	MTTR	False Positive Rate	Vendor Neutral	Layers Covered
Proposed Framework	95.2%	8.4s	2.1%	Yes	Identity, Device, Network
ZenGuard	95.0%	<10s	2.5%	Yes	Identity, Device, Network
Khule et al.	95.6%	1.4s	2.3%	No	Network, Application

Framework	Detection Accuracy	MTTR	False Positive Rate	Vendor Neutral	Layers Covered
Momanyi et al.	93.0%	N/A	N/A	Yes	Network (VPN)
Commercial SIEM-SOAR	85.0%	45s	5.0%	No	Mixed

Table 1: Comparative Analysis of Zero Trust Frameworks

From the comparative analysis, it is evident that the developed framework performs competitively with existing frameworks while remaining neutral to vendors and covering all layers comprehensively. The 95.2% accuracy level in detecting attacks performed by the framework is competitive with the accuracy levels of ZenGuard 95.0% and Khule et al. 95.6%. Although the MTTR of 8.4 seconds is not better than that of Khule et al., 1.4 seconds, it is obtained from an architecture that is neutral to vendors.

V. CONCLUSION

This work has proposed the Zero Trust Architecture Framework for Advanced Threat Detection and Mitigation. This framework includes SIEM, SOAR, ZTA, and UEBA, all within one platform which has been designed to overcome deficiencies of the present security frameworks. The use of Isolation Forest for detecting anomalous behavior and python playbooks for automating response has enabled the implementation of threat detection and mitigation in a contextual and real-time manner.

Results obtained through the experimental assessment reveal that the designed framework can achieve the level of accuracy of detection of 95.2%, the Mean Time to Respond less than 10 seconds in cases of critical threats, and the False Positive Rate of 2.1%. Such performance is competitive to other approaches but retains all benefits inherent in such frameworks as vendor neutrality, transparency, and integrative capabilities. The framework corresponds to NIST SP 800-207 and ISO/IEC 27001 requirements.

Contributions made by this study include:

- A unifying architectural framework which links the concepts behind Zero Trust to practical SIEM-SOAR functionalities
- Interpretability of machine learning algorithms used to detect behavioral anomalies making SOC analysts confident about automated decision-making
- Dynamic and risk-adaptive playbooks that lead to decreased MTTR compared to rule-based static solutions

- Vendor-independent, API-first approach enabling flexible deployment in any environment without vendor lock-in.

Potential directions for further research may include:

- Utilization of Graph Neural Networks to provide better threat reasoning abilities and relational anomaly detection
- Application of reinforcement learning techniques to optimize policies
- Enhancement of the proposed framework to work in multi-cloud and edge computing environments
- Incorporation of threat intelligence feeds to conduct proactive threat hunting
- Studies of long-term performance of the framework on enterprise-wide level.

As Zero Trust architecture continues evolving from concept to practice, unifying frameworks that facilitate enforcement and management while accelerating the transformation process will become critical for organizations maintaining strong security postures.

REFERENCES

[1] F. Ruambo, E. Masanga, M. Mrindoko, and N. Chinonso, "Enhanced Backdoor Resilience in Cross-Platform Systems Using Zero Trust Based Software-Defined Perimeter Architecture Powered by SnortML IDS/IPS," in Intelligent Techniques for Cyber-Physical Systems, Taylor & Francis, 2025, pp. 1-20. DOI: 10.1201/9781003614197-29.

[2] A. Hassan, A. Rauf, N. Shafqat, R. Latif, and H. Khan, "ZenGuard: A Machine Learning Based Zero Trust Framework for Context Aware Threat Mitigation Using SIEM, SOAR and UEBA," Scientific Reports, vol. 15, no. 1, p. 35871, Oct. 2025. DOI: 10.1038/s41598-025-20998-4.

[3] National Institute of Standards and Technology, "A Zero Trust Architecture Model for Access Control in Cloud-Native Applications in Multi-Location Environments," NIST Special Publication 800-207A, Sep. 2023. [Online]. Available: https://www.nist.gov/.

[4] Z. Momanyi, V. Mbandu, and P. Kinoti, "Hybrid Model Approach for Real-Time Detection of Anomalies in Cloud Virtual Private Network Traffic," International Journal of Professional Practice, vol. 13, no. 3, pp. 13-25, 2025. DOI: 10.71274/ijpp.v13i3.608.

[5] M. Khule, D. Motwani, and D. Chauhan, "A Layered and Integrative Framework for Advanced Persistent Threat Detection and Mitigation: Combining AI, Zero-Trust, and Advanced Threat Intelligence," Cluster Computing, vol. 28, no. 11, p. 1, 2025. DOI: 10.1007/s10586-025-05561-0.

[6] National Institute of Standards and Technology, "Zero Trust Architecture," NIST Special Publication 800-207, Aug. 2020. [Online]. Available: https://www.nist.gov/.

[7] Cybersecurity Insiders, "2026 Zero Trust Report: Bridging the Execution Gap," Jan. 2026. [Online]. Available: https://www.cybersecurity-insiders.com/.

[8] OWASP Foundation, "Zero Trust Architecture Cheat Sheet," 2025. [Online]. Available: https://cheatsheetseries.owasp.org/.

[9] "Detecting Insider Threats Using Unsupervised Machine Learning," in Proc. IEEE International Conference on Computing, Communication and Intelligence (ICCCI), Tirupati, India, Feb. 2026.

[10] "A Novel Hybrid Model Merging LOF and iForest Algorithms for Insider Threats Detection," in Proc. IEEE International Conference on Computer and Applications (ICCA), Pimari Chinchwad, India, Aug. 2024.