Vol. 8, No. 1, February 2026

Journal of Information Systems and Informatics

Published By Asosiasi Doktor Sistem Informasi Indonesia

A Hybrid Machine Learning and Signature-Based Approach for Detecting Network Pivoting in BYOD Environments

Nassor Suleiman Amour[1], Judith Leo[2], Mussa Ally Dida[3]

[1,2,3]School of Computational and Communication Science and Engineering, Nelson Mandela African Institution of Science and Technology, Tanzania, United Republic of

Received:
November 28, 2025
Revised:
January 15, 2026
Accepted:
January 29, 2026
Published:
February 25, 2026

Corresponding Author:
Author Name*:
Nassor Suleiman Amour
Email*:
amourn@nm-aist.ac.tz

DOI:
10.63158/journalisi.v8i1.1428

Abstract. This study addresses the challenge of detecting network pivoting, a lateral movement technique that is difficult to identify in insider and BYOD environments because malicious transitions can resemble normal internal activity. The objective was to improve detection of both known and unknown pivoting behaviours while supporting practical triage in resource-constrained institutions. A hybrid detection framework was developed that fuses Snort signature alerts with machine learning classification and unsupervised anomaly detection using behavioural features derived from BYOD-like network traffic. The approach was evaluated in a controlled testbed and supported by organisational survey findings on awareness and monitoring practice. Results show the hybrid system achieved 96.2% classification accuracy with a 4.5% false positive rate when distinguishing normal traffic, suspicious activity, and pivoting attacks. Compared with signature-only and machine-learning-only baselines, the hybrid design detected simulated pivoting attempts earlier and more consistently. User acceptance testing also reported strong satisfaction with the integrated dashboard for monitoring, filtering, and reporting. The key contribution is a unified, dashboard-oriented fusion of signature and behavioural evidence that strengthens early lateral movement detection and reduces manual correlation effort.

Keywords: Network pivoting detection, lateral movement, BYOD security, hybrid intrusion detection, insider threat detection

ISI Journal of Information Systems and Informatics
Published By Asosiasi Doktor Sistem Informasi Indonesia

1. INTRODUCTION

Modern organizations operate in highly interconnected environments where external security controls have improved significantly, yet attackers increasingly exploit internal weaknesses. Bring Your Own Device (BYOD) practices widen these weaknesses because personal devices routinely move between protected and unprotected networks. When such devices are compromised outside the institutional environment, they can return as unnoticed entry points once reconnected. This creates favorable conditions for network pivoting, a technique used to move from one compromised host to other internal systems after initial access. Marques et al. describe pivoting as a method through which attackers exploit weak points in a network, particularly in insider contexts and BYOD environments where traditional defense layers may not offer sufficient protection [1].

In practical terms, pivoting is closely linked to lateral movement, where an adversary expands reach inside a network in search of high-value assets. In widely used attack taxonomies, including the MITRE ATT&CK model, lateral movement describes the stage in which an attacker transitions between internal hosts to extend access and progress toward objectives [16]. Pivoting operationally supports this stage by enabling movement across internal boundaries and into segments that were not initially reachable [7]. Detection remains difficult because pivoting can resemble legitimate internal activity, particularly when encrypted channels, remote administration tools, or normal user behaviour are involved. Recent survey work on lateral movement detection shows that many approaches either depend heavily on endpoint visibility or struggle to deliver reliable detection from network evidence alone, especially when attacks blend into routine traffic [16]. Prior pivoting-focused research also reports that methods can produce overwhelming false positives when moved from constrained settings to more realistic traffic conditions, which limits operational usefulness in operational environments [6].

Machine learning has become an important analytical tool for detecting complex and subtle behavioural patterns within network traffic. Prior work shows that machine learning can recognize tunneling and other elusive forms of communication that conventional signature systems often fail to detect [2]. At the same time, signature-based methods remain valuable because they provide rapid identification of known threats and

can be deployed with low complexity. Several studies support combining these ideas, highlighting the advantages of merging rule-based detection with behavioural machine learning for improved intrusion detection accuracy [3]–[5].

Insider-enabled pivoting remains difficult to detect because adversarial actions can be embedded in legitimate workflows, and early lateral movement often produces weak network-level signals that are hard to distinguish from routine activity [1], [7], [16]. In many public and resource-constrained institutions, this challenge is amplified by limited monitoring coverage, inconsistent endpoint controls, and fragmented security tooling, which reduces consistent visibility across devices and network segments.

Although several studies have explored intrusion detection more broadly, recent literature still lacks a complete framework that combines behavioural machine learning with signature techniques specifically for detecting pivoting in insider and BYOD environments. Existing work often emphasizes either behavioural analysis or signature detection, without offering a unified mechanism that can address both known and unknown pivoting patterns in a way that is operationally usable. This gap motivates the present study, which designs and validates a hybrid detection system that integrates behavioural analysis with signature rules to improve the identification of pivoting attempts in institutional BYOD networks.

The study contributes in the following ways. First, it proposes a hybrid pivoting detection framework that integrates multi-stage machine learning models with Snort signature rules to cover both unknown behaviours and known indicators. Second, the framework is empirically evaluated using BYOD-like traffic generated through controlled pivoting experiments and correlated behavioural logs, enabling reproducible performance assessment. Third, the detection outputs are integrated into a dashboard-oriented workflow that supports alert triage and investigation in institutional environments.

2. METHODS

2.1. Research Design and Workflow

This study used an experimental design within a design science research (DSR) framework to develop and evaluate a hybrid machine learning and signature-based system for

detecting network pivoting in insider and BYOD environments. DSR was selected because it offers a structured pathway from problem identification to artefact construction and evaluation, with emphasis on demonstrating utility, rigour, and relevance through appropriate evaluation activities [9], [19], [21]. Established guidance on positioning and presenting artefact contributions also informed how the solution and evaluation outcomes were framed in the manuscript [20]. The experimental component enabled controlled measurement of detection performance under defined pivoting scenarios.

The workflow started with a literature review and a baseline organisational survey to characterise insider and BYOD-related pivoting threats, current monitoring practice, and limitations of conventional intrusion detection. These findings were translated into functional requirements such as real-time alerting, low false positives, and visibility into suspected pivoting paths, maintaining traceability between the identified problem conditions and the artefact design goals [19], [21]. System development then proceeded iteratively using the Scrum framework (Figure 1), selected for incremental, feedback-driven development and its suitability for complex software projects where requirements evolve through repeated validation cycles [10], [22]. Sprint reviews applied predefined acceptance criteria focused on functional completion, technical correctness of the data and detection pipeline, validity of generated outputs, and readiness of key user workflows for alert review and reporting.

Evaluation criteria were applied per iteration. Iteration 1 was accepted when all research objectives were mapped to system requirements and modules with no unmapped items, and when the pivoting scenarios were fully specified and traceable to those requirements. Iteration 2 was accepted when Zeek and Snort logs were ingested, parsed, joined, and stored correctly, and when repeated runs produced consistent record counts and key fields. Iteration 3 was accepted when feature computation and model inference ran end-to-end without failures and evaluation outputs, including confusion matrix and summary metrics, were generated correctly on held-out data. Iteration 4 was accepted when dashboard filtering, triage, reporting, and export results matched the alerts and records stored in the database and supported the intended monitoring tasks.

Pilot testing with administrators, who reported using SIEM platforms alongside Zeek, Snort, and custom scripts, reinforced the practical challenges of fragmented monitoring

and supported the need for an approach that consolidates behavioural indicators and signature evidence into a single workflow [27]. These insights informed model selection, correlation logic, and interface design choices aimed at improving operational triage in BYOD-intensive environments. The organisational survey and system evaluation were conducted within public institutions in Tanzania and used controlled pivoting scenarios for labelled assessment; the findings should therefore be interpreted as context-specific evidence rather than a universal baseline.

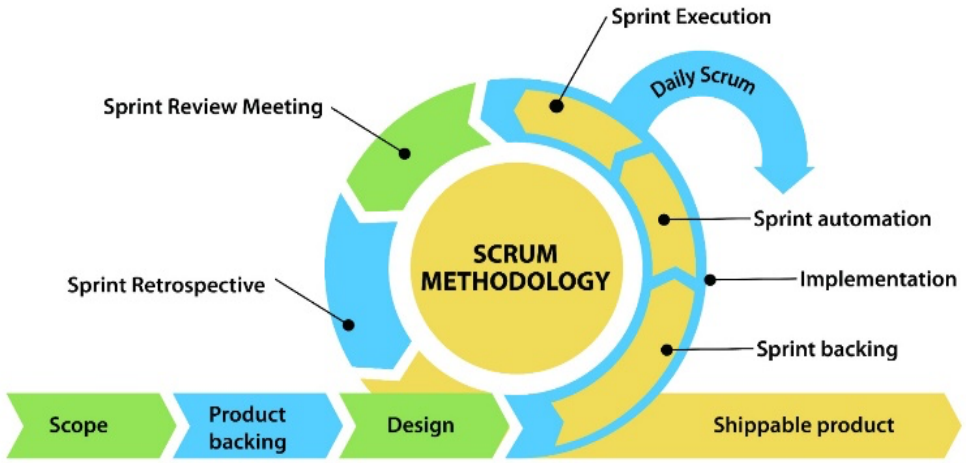

Figure 1. Scrum Methodology

2.2. Participants and Data Collection Instruments

The empirical component involved participants from public organisations. Two groups were targeted: network and system administrators, who manage enterprise networks and security tools, and general staff who routinely use organisational networks and personal devices. Together, they provided complementary perspectives on insider behaviour, BYOD usage, and detection challenges. The empirical component focused on public organisations within Tanzania, with participants drawn from institutions in Dar es Salaam and Zanzibar. This geographic scope provides a realistic institutional setting for evaluation, but generalisability to other regions may vary due to differences in BYOD policies, infrastructure maturity, and threat profiles. Future work should validate the framework across multiple institutions and in cross-country settings to confirm robustness under broader operational conditions.

Simple random sampling was used for regular staff, while administrators were selected purposively due to their specialised expertise. Data were collected using two structured questionnaires: one for staff (demographics, cyber-threat awareness, BYOD use, insider incidents, policy perceptions) and one for administrators (deployed tools, frequency and nature of pivoting attempts, visibility gaps, expectations for a hybrid system). In parallel, a BYOD-like testbed with virtual machines generated normal activity and staged pivoting attacks. Zeek captured detailed connection, DNS, HTTP, and SSL logs [11], and Snort produced alerts based on pivoting-focused signature rules [12], forming the main dataset for training and evaluating the hybrid system.

2.3. Data Preparation and Feature Engineering

Survey data were checked for completeness, cleaned of invalid entries, and coded into numerical form, while open-ended responses were grouped into themes. This produced a dataset suitable for descriptive and comparative analysis between staff and administrators and supported interpretation of the technical findings.
For the network experiments, raw Zeek and Snort logs were parsed, normalised, and joined using connection identifiers and timestamps to align behavioural features (for example connection duration and byte counts) with corresponding Snort alerts. The resulting dataset included IP addresses, ports, protocols, timestamps, and connection attributes. Additional features were derived, including per-host connection counts, port and protocol frequencies, and temporal indicators (hour of day and day of week). The final feature space comprised enriched behavioural and contextual attributes, statistical measures, frequency-based metrics, temporal patterns, and encoded categorical variables for protocols, services, and flags (Table 1). In the controlled attack scenarios, Snort alerts were treated as reference labels to support supervised learning, and were also used as binary indicators within the hybrid fusion logic. Missing values were handled using simple, explicit strategies to avoid introducing artificial behaviour into network traces: zeros were used for count-based features to represent the absence of observed events within the relevant connection window, and reserved tokens were used for categorical fields to preserve model input consistency while keeping "unknown/not observed" distinct from valid categories. This supports reproducible preprocessing and reduces the risk of biased imputation in high-variance traffic data.

Table 1. Feature groups used for pivoting detection

Feature group	What it captures	Example features (Illustrative)	Why it matters for pivoting
Connection identifiers and context	Basic session attributes	src/dst IP, src/dst port, protocol, timestamp	Provides session context for correlation and sequencing
Flow and volume behaviour	Communication intensity	duration, orig/resp bytes, packet-like counts where available	Pivoting often changes volume and session patterns across hops
Frequency and aggregation	Host and port activity patterns	per-host connection counts, port/protocol frequencies	Pivoting may increase scanning, new port usage, or unusual service mixes
Temporal indicators	Time-based activity signals	hour of day, day of week	Helps separate normal business patterns from anomalous pivoting timing
Encoded categorical indicators	Service and flag semantics	protocol/service/flag encodings	Captures behavioural signatures that are not numeric by default
Signature signals and labels	Known-pattern evidence	Snort alert indicator; Snort-derived label (where used)	Improves precision for known pivoting patterns and supports fusion logic

2.4. Modelling and Data Analysis Techniques

Analysis combined survey statistics, machine learning experiments, and system-level evaluation. Survey responses were summarised using descriptive statistics and cross-

tabulations to characterise awareness, tool usage, and reported insider incidents, and to compare administrators with general staff. These results supported interpretation of the experimental findings and informed the design rationale for the hybrid detection architecture.

On the technical side, supervised and unsupervised models were trained using the enriched network dataset. Because normal traffic typically dominates security datasets, the labelled data were partitioned using a stratified train test split to preserve proportional class representation in both training and testing sets, consistent with standard guidance for imbalanced learning settings [31]. To reduce bias toward majority classes and improve recognition of rare attack patterns, the Synthetic Minority Over-sampling Technique (SMOTE) [17] was applied to the training split to increase the representation of minority attack classes, including pivoting, while the test split was left unchanged to preserve unbiased evaluation.

For supervised learning, multiple candidate classifiers were evaluated to support fair selection across different algorithm families. Hyperparameter optimisation was performed automatically using a successive-halving style search procedure, which allocates more resources to promising configurations while terminating weaker settings early [18]. The search was executed using the HalvingGridSearchCV implementation in scikit-learn with 5-fold cross-validation [24]. Weighted F1-score was used as the principal selection criterion to ensure performance was not dominated by majority classes, with evaluation choices aligned to common recommendations for imbalanced classification settings [31], [32]. Based on this process, a tree-based ensemble classifier (Random Forest) was retained as the primary supervised model due to its robust detection performance and suitability for intrusion detection settings [13]. Hyperparameter search targeted model-specific settings for the candidate classifiers, while parameters not included in the search space followed standard scikit-learn defaults [24].

For unsupervised detection, Isolation Forest and One-Class SVM were trained exclusively on normal or unlabeled traffic to learn typical behavioural patterns without relying on predefined attack categories. Deviations from the learned baseline were flagged as anomalies and later correlated with signature outputs in the hybrid fusion stage. Isolation Forest and One-Class SVM were selected for their suitability for high-dimensional

anomaly detection and boundary-based novelty detection, respectively [14], [15]. Snort alerts were incorporated as signature evidence, and the supervised predictions, anomaly indicators, and Snort outputs were combined in the fusion logic to strengthen coverage for both known pivoting patterns and stealthy behaviours lacking explicit signatures. Table 2 summarises the modelling components, training data assumptions, and tuning scope used in the hybrid detection workflow.

Performance evaluation was conducted on the held-out test set using accuracy, precision, recall, F1-score, false positive rate, and confusion matrix analysis. System-level evaluation further included predefined test cases and feedback summaries to confirm functional correctness and to assess the operational usefulness of the detection outputs within the dashboard workflow.

Table 2. Summary of models, training setup, and tuning scope

Component	Model	Training data used	Purpose in the hybrid system	Tuning and selection approach
Supervised classification	Candidate classifiers (multiple families)	Labelled data (stratified split)	Predict attack class including pivoting	Hyperparameters optimised automatically with successive-halving search (HalvingGridSearch CV) using 5-fold CV; selected by weighted F1-score
Supervised classification (final)	Random Forest	Labelled data (stratified split)	Primary supervised predictor	Retained as best-performing supervised model after automated tuning; non-searched parameters kept at defaults

Component	Model	Training data used	Purpose in the hybrid system	Tuning and selection approach
Unsupervised anomaly detection	Isolation Forest	Normal-only or unlabeled traffic	Detect behavioural deviations without attack labels	Trained on normal baseline; anomaly scores used as indicators in fusion
Unsupervised anomaly detection	One-Class SVM	Normal-only or unlabeled traffic	Boundary-based novelty detection	Trained on normal baseline; anomaly indicators used in fusion
Signature detection	Snort rules/alerts	Network traffic	Detect known patterns quickly	Alerts used as binary indicators in fusion; treated as reference labels only in controlled scenarios

2.5. System Implementation and Architecture

The hybrid detection system was implemented as a modular architecture that integrates data capture, feature computation, machine learning inference, and alert visualisation. The end-to-end design is summarised in Figure 7. Mirrored BYOD traffic was processed by Snort and Zeek in parallel. Snort generated signature-based alerts for known attack patterns [12]. Zeek produced behavioural logs that capture connection and protocol activity for higher-level analysis [11]. Both outputs were stored in a PostgreSQL backend. A Python processing pipeline then parsed, normalised, and transformed records into model-ready features.

The machine learning module handled feature engineering, model loading, and prediction. Its outputs were passed to a fusion component together with Snort alerts. The fusion logic assigned a risk score and a final label to each observed session. Results were presented through a web-based dashboard. The dashboard supports near-real-time alert monitoring, filtering, and review of suspected pivoting paths. The architecture was designed to remain extensible. New models, features, and signature rules can be added

without disrupting the existing workflow, which supports long-term maintainability in BYOD-intensive environments.

Operational deployment considerations were incorporated during implementation. Continuous Zeek logging and Snort alerting can produce high volumes of records, so storage, indexing, and compute capacity must scale with traffic load. In this implementation, detection operates in near-real-time. Alerts are produced after log ingestion and feature computation, rather than through strict per-packet inline inspection. End-to-end delay depends on sensor load, database throughput, and feature computation rate. In resource-constrained settings, this can be managed through log retention windows, batching non-critical aggregation tasks, and scheduling model or rule updates during low-traffic periods. Future work will benchmark throughput and end-to-end latency under different traffic loads to quantify operational limits more precisely.

3. RESULTS AND DISCUSSION

3.1. Demographics of Participants

A total of 276 participants were surveyed, comprising 252 general staff and 24 network or system administrators from public organisations in Dar es Salaam and Zanzibar (Table 3). The sample is intentionally weighted toward general staff, who represent the largest group of BYOD users and therefore the main source of routine traffic and insider-enabled exposure. The smaller administrator subgroup provides operational context for interpreting differences in awareness, tool usage, and incident handling, although administrator comparisons are treated cautiously due to the group size. Overall, this structure reflects a common institutional reality in which a small technical team must monitor a large user population, supporting the need for detection that combines signature evidence with behavioural analytics.

Table 3. Demographic Breakdown of Survey Respondents

Category	Subgroup	Percentage
Age	Under 25	10 %
	25–34	45 %

Category	Subgroup	Percentage
	35–44	30 %
	45+	15 %
Gender	Male	62 %
	Female	38 %
Organizational Tenure	1–3 years	40 %
	4–6 years	35 %
	> 6 years	25 %
Role	General Staff	56 %
	Mixed IT/Non-IT	28 %
	IT Admin	16 %

3.2. Awareness and Security Behaviour

Figures 2 and 3 summarise awareness of insider threats and network pivoting among general staff and administrators. General staff report mostly low to medium awareness, while administrators report higher awareness overall, although gaps remain in both groups. The results also indicate that training on pivoting is not consistently received, which reinforces the need for targeted awareness support for both technical and non-technical users. In response, the system design prioritised an analyst-facing dashboard that categorises alerts and shows, for each event, whether it was triggered by signature evidence, anomalous behaviour, or both, together with a risk score to guide review. This supports simple triage workflows that reduce reliance on raw IDS outputs and helps institutions with limited specialist capacity.

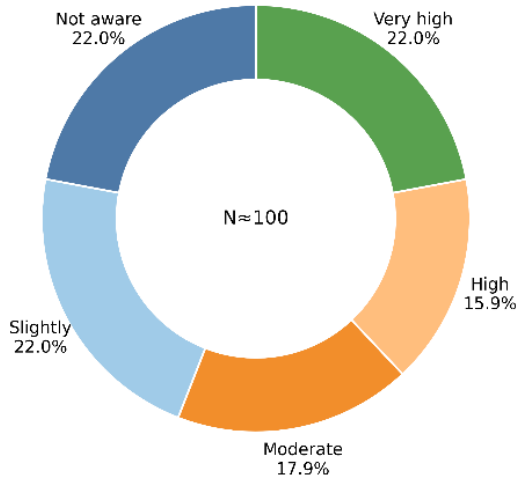

Figure 2. Staff Awareness

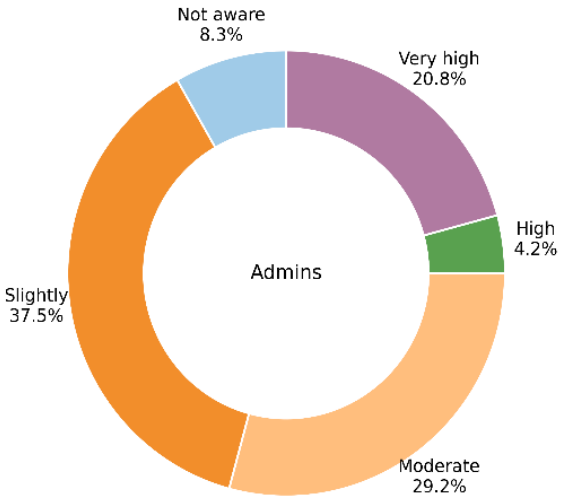

Figure 3. Administrator Awareness

3.3. Use Of Monitoring Tools

Administrators reported using a mix of monitoring tools, most commonly SIEM platforms and network monitoring tools such as Zeek, with some also using Snort or custom scripts (Figure 4). A few indicated that no specific tool is available for detecting pivoting. Overall, tool coverage and maturity vary across institutions, increasing investigation effort because alerts and behavioural evidence must be correlated across separate systems. In response, the hybrid system integrates Snort signatures and Zeek-derived behavioural indicators in a single pipeline and presents fused outputs in one dashboard view, reducing manual correlation and supporting consistent triage in BYOD-intensive environments.

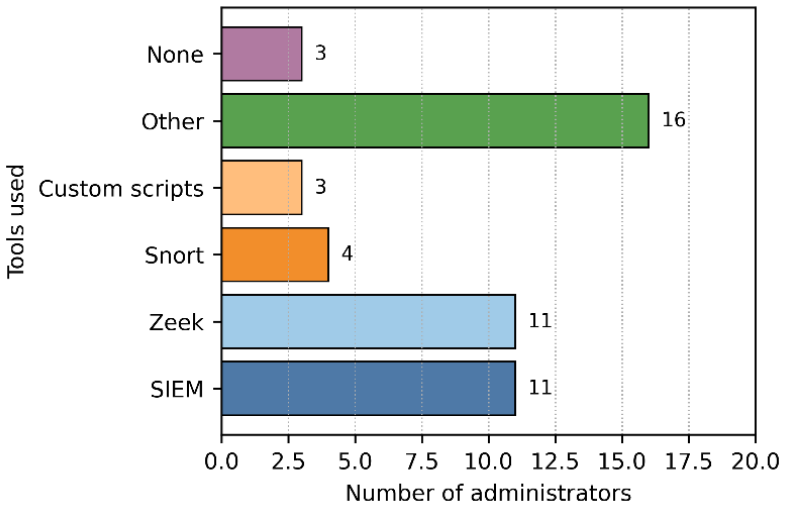

Figure 4. Tools and methods used by administrators to detect pivoting

3.4. Frequency of Insider Related Incidents

Results on detection frequency showed that some administrators reported noticing pivoting attempts often, while others rarely or never observed them (Figure 5). Staff responses also indicate that insider-related problems, such as improper access or compromised devices, occur in practice, even when they are not recognised or reported as pivoting. Overall, the findings suggest that pivoting threats are present, but that observed detection depends strongly on monitoring coverage and operational visibility. This variation supports a hybrid approach that combines signature-based and behavioural detection to improve coverage under partial visibility, and it motivates risk-informed dashboard triage so analysts can filter and prioritise higher-risk events when monitoring resources are limited.

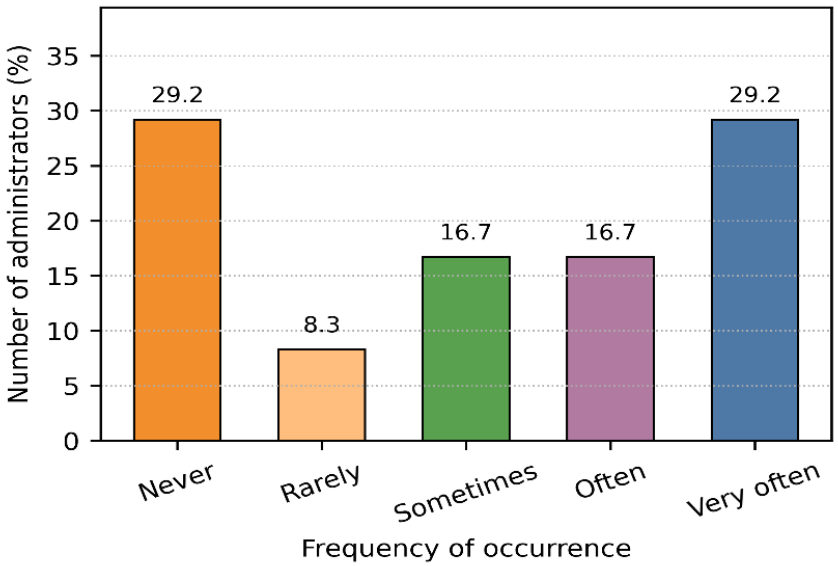

Figure 5. Frequency of detected network pivoting attempts (administrator survey).

3.5. Machine Learning Analysis and Feature Importance

To improve interpretability, Figure 6 presents the six most influential features for pivoting detection as a ranked bar chart. Importance values are normalised scores from the trained supervised model, reported to indicate which features most influenced pivoting predictions. The top signals concentrate on lateral movement behaviour and multi-host traversal patterns. Lateral movement score and destination host count indicate repeated internal connections and expansion across hosts, which is consistent with pivoting workflows. Anomaly score adds a behavioural indicator that helps flag stealthier transitions that may not match fixed signature rules. The remaining high-

impact features capture access and execution traces that often accompany internal progression, supporting the use of enriched behavioural evidence alongside signature alerts. These findings directly support the hybrid design: behavioural indicators strengthen detection when signature coverage is incomplete, while signature outputs help confirm known patterns and reduce false positives during triage.

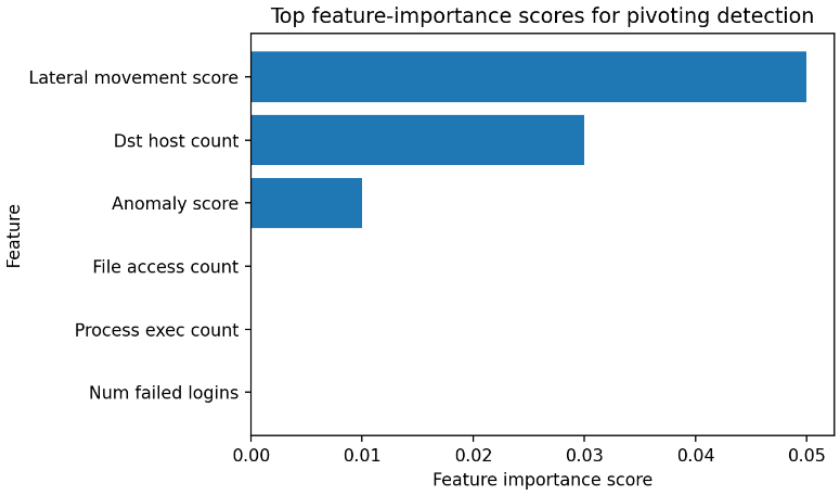

Figure 6. Features for pivoting detection

3.6. System Development

The final hybrid detection system combined machine learning classification, anomaly detection and Snort signature rules into a single operational platform. The overall architecture, shown in Figure 7, routes mirrored BYOD traffic in parallel to Zeek and Snort, stores their outputs in PostgreSQL, and then passes them to a Python-based machine learning module and an alert correlator before displaying results on the web dashboard. The implemented interfaces provide administrators with authentication, near-real-time monitoring, report generation and user management, and show how each action interacts with the detection pipeline and data stores. Together, these design artefacts demonstrate that the system requirements from the earlier stages were implemented in practice: alerts from known signatures and behavioural anomalies are fused into a single view, reports can be generated for different time frames and sources, and administrators can manage users and configurations through one interface. This integrated design supports the goal of improving visibility of pivoting activity in BYOD environments while keeping the workflow manageable for security teams.

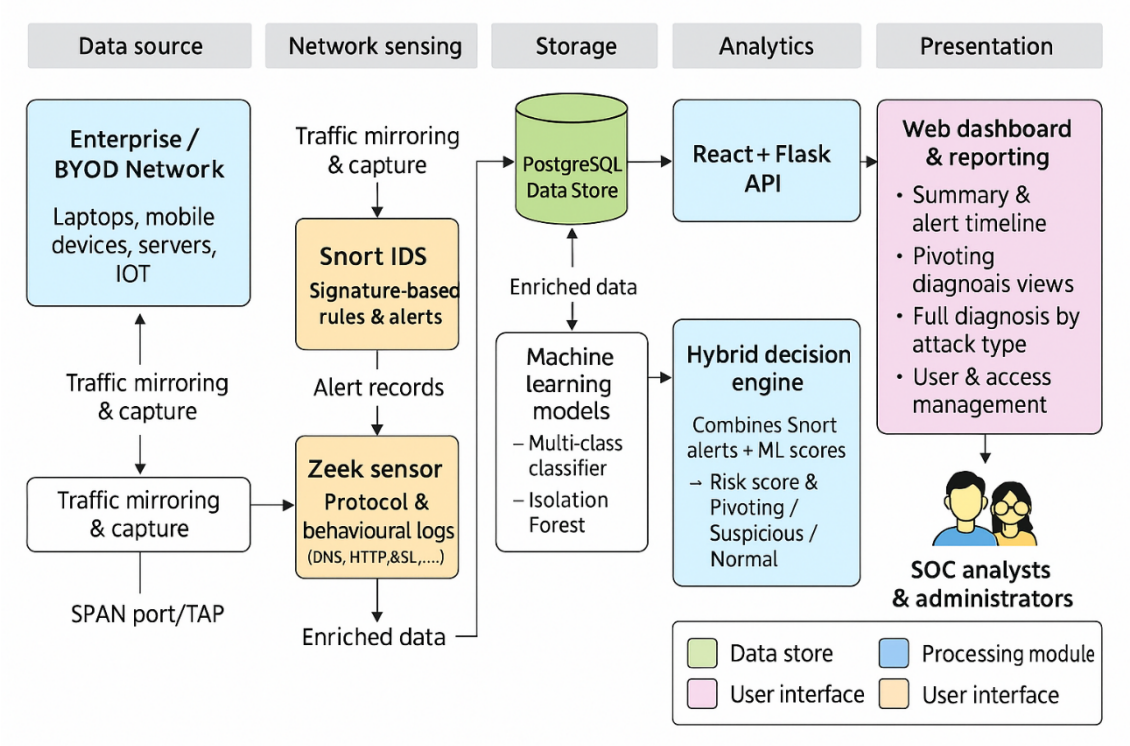

Figure 7. System Architecture

3.7. System Interface Implementation

The hybrid detection system was operationalised through a web-based platform that translates Snort, Zeek and machine learning outputs into actionable monitoring and reporting functions for administrators and analysts. The implemented interfaces confirm that the study delivered not only a detection model but also a usable institutional monitoring solution aligned with routine security workflows. The main functional coverage of the system is summarised in Table 4, while representative interface screenshots are presented in Figures 8–12 to demonstrate how authentication, alert monitoring, case handling, reporting and diagnostic review are supported within a single integrated environment.

Table 4. Key system interfaces and operational roles

Interface group	Main purpose and output
Authentication & user management	Supports secure access, registration and controlled user roles to protect sensitive monitoring data.

ISI Journal of Information Systems and Informatics
Published By Asosiasi Doktor Sistem Informasi Indonesia

Interface group	Main purpose and output
Dashboard overview	Provides a high-level view of alert volumes, risk categories and detection sources for quick situational awareness.
Alerts & filtering	Displays enriched alert details (IPs, ports, timestamps, protocol/service context) and allows prioritisation by risk, time and source.
Case/incident grouping	Enables analysts to organise related alerts into unified investigations for structured response and documentation.
Reports & export	Generates full, source-based and risk-based summaries with export support for management review and offline analysis.
Diagnostic and detail lookup	Allows drill-down into alert-level context to support validation, investigation and reduction of false positives.
System status/administration	Supports rule/model oversight and operational monitoring of the platform's health and update readiness.

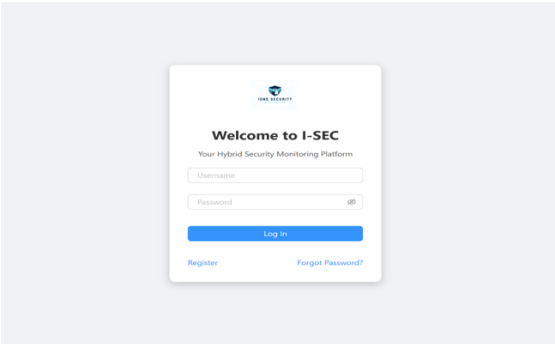

Figure 8. Login / user access interface

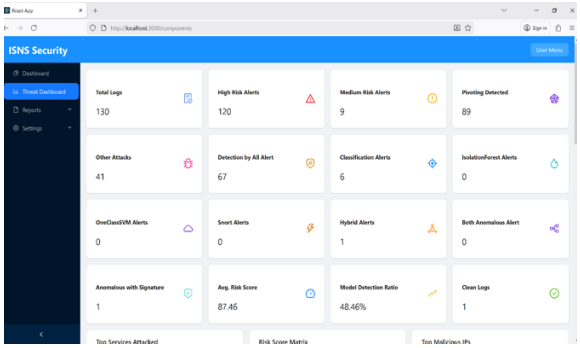

Figure 9. Dashboard overview

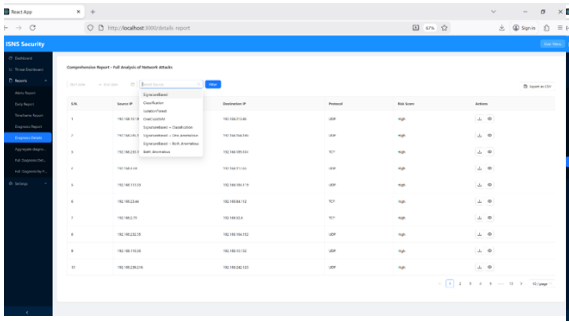

Figure 10. Alerts + filters

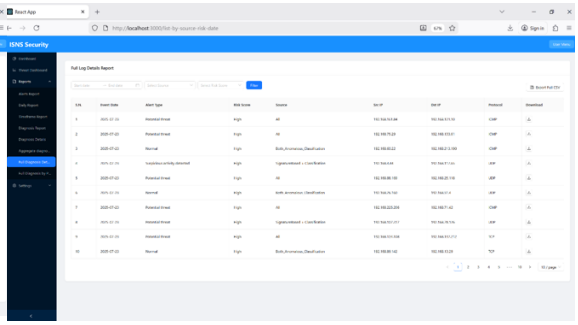

Figure 11. Reporting / export

ISI Journal of Information Systems and Informatics
Published By Asosiasi Doktor Sistem Informasi Indonesia

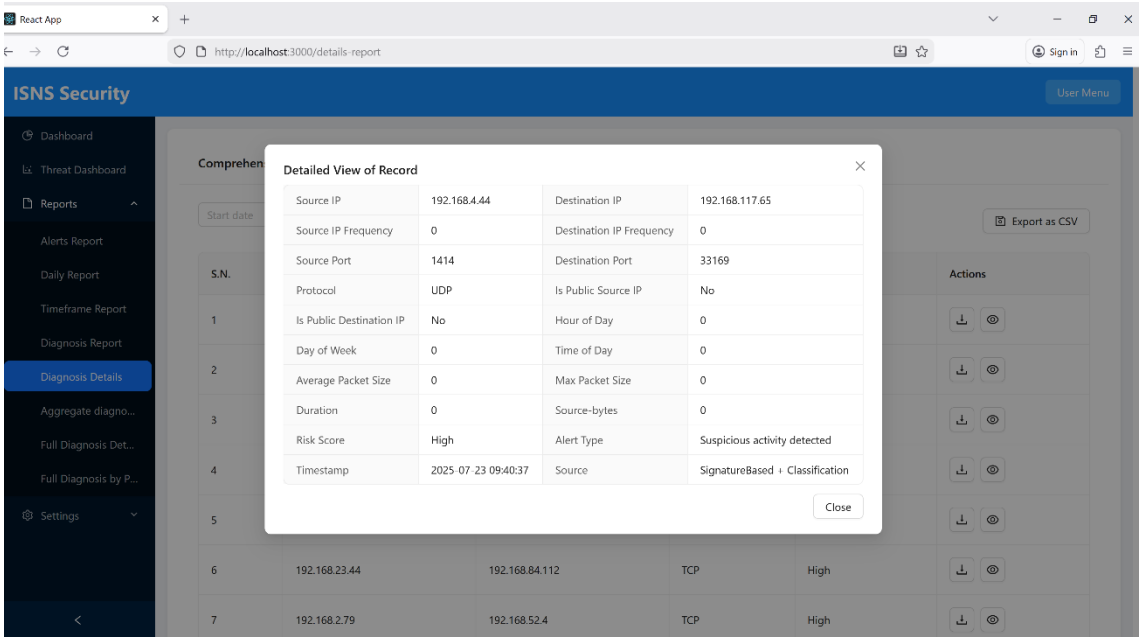

Figure 12. Diagnostic detail view

3.8. System Testing and User Feedback

The complete system was tested across all main interfaces, including the admin login page, alert dashboard, detailed alert view, reporting screens, and export functions. The system interfaces and their operational roles are summarised in Table 4, and the main workflow is illustrated in Figures 8–12. Figure 8 shows secure login, Figure 9 the main dashboard overview, Figure 10 alert filtering, Figure 11 reporting and export, and Figure 12 drill-down diagnostic details. Collectively, these interfaces demonstrate that the system supports end-to-end operational monitoring, while the user acceptance ratings provide evidence of usability and operational fit. The tests confirmed that users could log in securely, that Snort and the machine learning models raised alerts correctly, and that these alerts were stored, filtered, and displayed as expected on the dashboard. User acceptance feedback is summarised in Figure 13 and indicates consistently strong ratings across the tested functions. "High" ratings ranged from 85% (detection accuracy) to 96% (admin login and alert filtering), with an average "High" rating of 92.1% across the eight evaluated functions. Satisfaction was strongest for the dashboard overview, incident details, and report export features. Some users requested clearer explanations of risk scores and slightly simpler navigation between filters and report views, suggesting that brief analyst orientation (training) on risk-score interpretation and dashboard navigation would further strengthen day-to-day use.

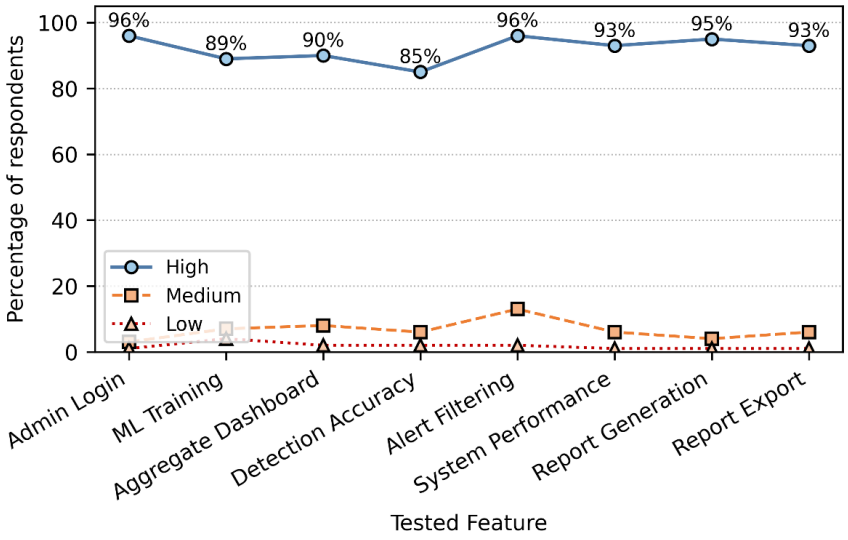

Figure 13. UAT responses

3.9. Discussion

The combined survey and experimental results highlight why pivoting detection remains uneven in practice and why a hybrid design is operationally justified in BYOD-heavy institutional networks. Survey responses show clear asymmetry in awareness: administrators generally report higher familiarity with pivoting and insider threats than general staff, yet knowledge gaps persist in both groups. This matters because pivoting often succeeds in the "grey zone" between policy and practice—users may unintentionally introduce compromised devices, while administrators must distinguish malicious lateral movement from routine internal access. In BYOD settings, that challenge intensifies because devices regularly traverse trusted and untrusted environments, increasing behavioural diversity and reducing the effectiveness of uniform enforcement and monitoring policies across organisations [28]. Consistent with insider-threat literature, the survey pattern aligns with the broader problem that malicious activity can be embedded in normal workflows, making reliable detection difficult when intent is not directly observable and signals are weak or ambiguous [29].

A second practical finding is the fragmentation of monitoring capability across institutions. Administrators reported using SIEM platforms, Zeek, Snort, and custom scripts in different combinations, and some indicated limited tooling for pivoting detection. This variability implies that pivoting visibility is not purely a function of threat prevalence; it is strongly shaped by sensor coverage, data integration, and analyst

capacity. When evidence is split across tools, analysts must perform manual correlation across separate alert feeds and log sources, which increases triage time and creates inconsistent detection outcomes—especially in teams with limited staffing and constrained operational maturity. This observation mirrors prior evidence that many real monitoring environments depend on fragmented SIEM and network-monitoring stacks where correlation becomes a practical bottleneck for timely response [27]. The survey also showed variation in how often administrators "notice" pivoting attempts, which may reflect these visibility differences rather than true differences in attack occurrence.

The experimental results support the study's core design choice: combining complementary detection signals improves practical coverage for pivoting and lateral movement. Signature-based alerts (Snort) are valuable because they deliver fast, high-confidence detection for known patterns and can anchor investigations with interpretable evidence. However, pivoting frequently involves novel sequences, tool variations, or subtle behavioural shifts that do not map cleanly to predefined rules. Behavioural indicators derived from Zeek telemetry (e.g., host traversal patterns, destination host count, temporal irregularities, and anomaly scores) help increase sensitivity to such activity, but they also introduce the well-known trade-off: anomaly-oriented methods broaden detection but can elevate noise and false positives unless contextualised and prioritised [25], [26]. The hybrid fusion strategy directly responds to this trade-off by correlating Snort outputs with supervised predictions and anomaly indicators, then presenting results as risk-scored, consolidated alerts. In effect, signature evidence helps raise precision and interpretability, while behavioural modelling expands coverage when signatures are incomplete or evasions occur—consistent with prior work advocating blended rule-based and machine learning detection for complex network threats [3]–[5].

Importantly, this fusion is not only a modelling choice but also an operational workflow intervention. The survey results indicated that manual correlation across tools is a major burden; the implemented system reduces that burden by joining Snort and Zeek evidence in a single pipeline and exposing the results through a unified dashboard for filtering, drill-down, reporting, and export. This design aligns with findings from Zeek-based intrusion detection research showing that behavioural indicators can be reliably derived from Zeek-style telemetry and strengthened using anomaly-oriented techniques under

realistic traffic variability [30]. It also reflects the design science objective of translating organisational constraints (uneven awareness, limited staff, and fragmented tooling) into artefact requirements (consolidated evidence, triage support, and actionable output).

User acceptance testing further suggests that the integrated presentation of alerts is meaningful for day-to-day monitoring. High ratings across dashboard overview, filtering, diagnostics, and reporting indicate that the system's primary value is not merely detection accuracy but workflow usability—making pivoting evidence easier to interpret and prioritise in settings where security teams must manage large user populations with limited specialist capacity. At the same time, requests for clearer explanations of risk scores highlight a common adoption barrier for ML-assisted detection: analysts need transparent reasoning cues (e.g., "which signals drove this risk?") to trust prioritisation outputs and reduce alert fatigue. This points toward future enhancement through explainability features (feature-attribution summaries, rule/behaviour tags, or "reason codes") and brief analyst orientation to standardise risk-score interpretation.

Interpretation of the results should consider the evaluation context. The study reflects public organisations in Tanzania and uses controlled pivoting scenarios to produce labelled assessment data. While this supports reproducibility and clear measurement, it may not capture the full diversity of real-world traffic, attacker behaviours, encryption patterns, and organisational heterogeneity. In addition, using Snort alerts as labels in controlled experiments provides practical ground truth, but operational deployments may require validation against incident-response outcomes or curated datasets to ensure that labels are not biased toward signature-visible behaviours. Future work should therefore prioritise (i) multi-institution and cross-country validation, (ii) benchmarking throughput, storage, and end-to-end latency under varied traffic loads, and (iii) longitudinal evaluation to assess stability as network behaviour and policies evolve. These steps would strengthen confidence in generalisability and clarify operational limits in higher-volume environments.

The evidence indicates that uneven awareness and fragmented monitoring are real institutional constraints, and that a hybrid, dashboard-oriented approach can improve pivoting detection and triage by unifying signature confidence with behavioural sensitivity. The results support the view that effective pivoting defence in BYOD contexts

is not achieved by a single technique, but by integrated evidence, prioritised workflows, and institutional readiness—training, policy alignment, and sustainable monitoring capacity.

4. CONCLUSION

This study examined insider-related pivoting risks in BYOD environments by combining evidence from organisational practice and controlled technical evaluation. Survey findings highlighted practical gaps in awareness and monitoring consistency that can delay early detection and coordinated incident handling in public institutions. In response, a hybrid detection approach was implemented that integrates Zeek behavioural logs, Snort signature alerts, and machine learning models to capture both known pivoting patterns and stealthy behavioural deviations. Experimental results showed strong detection performance (96.2% accuracy with a 4.5% false positive rate), while system testing and user feedback indicated that the dashboard supports practical monitoring tasks through secure access, consolidated alert review, and reporting, with consistently high acceptance across key workflow functions. Overall, the study demonstrates that combining behavioural analytics with signature evidence can improve pivoting visibility and operational triage in BYOD-intensive, resource-constrained environments.

ACKNOWLEDGMENT

Sincere gratitude is extended to Dr. Judith Leo and Dr. Mussa Dida for their guidance, technical insights, and sustained support throughout the development and evaluation of the hybrid detection system. Appreciation is also extended to the Nelson Mandela African Institution of Science and Technology (NM-AIST) for providing an enabling academic environment and resources that supported this study. Thanks are further extended to family and friends for encouragement and moral support throughout the research journey. Finally, appreciation is given to all participants and administrators who contributed their time and experience during the survey and user acceptance testing phases.

ISI Journal of Information Systems and Informatics
Published By Asosiasi Doktor Sistem Informasi Indonesia

REFERENCES

[1] R. S. Marques, H. Al-Khateeb, G. Epiphaniou, and C. Maple, "APIVADS: A novel privacy-preserving pivot attack detection scheme based on statistical pattern recognition," *IEEE Trans. Inf. Forensics Secur.*, vol. 17, pp. 700–715, 2022, doi: 10.1109/TIFS.2022.3146076.

[2] U. Aslam, E. Batool, S. N. Ahsan, and A. Sultan, "Hybrid network intrusion detection system using machine learning classification and rule based learning system," *Int. J. Grid Distrib. Comput.*, vol. 10, no. 2, pp. 51–62, Feb. 2017, doi: 10.14257/ijgdc.2017.10.2.05.

[3] D. Vinod and M. Prasad, "A novel hybrid automatic intrusion detection system using machine learning technique for anomalous detection based on traffic prediction," in *Proc. Int. Conf. Netw. Commun. (ICNWC)*, Chennai, India, Apr. 2023, pp. 1–7, doi: 10.1109/ICNWC57852.2023.10127442.

[4] Z. Sui, H. Shu, F. Kang, Y. Huang, and G. Huo, "A comprehensive review of tunnel detection on multilayer protocols: From traditional to machine learning approaches," *Appl. Sci.*, vol. 13, no. 3, Art. no. 1974, Feb. 2023, doi: 10.3390/app13031974.

[5] R. Palanisamy, A. A. Norman, and M. L. M. Kiah, "Compliance with bring your own device security policies in organizations: A systematic literature review," *Comput. Secur.*, vol. 98, Art. no. 101998, Nov. 2020, doi: 10.1016/j.cose.2020.101998.

[6] M. Husák, G. Apruzzese, S. J. Yang, and G. Werner, "Towards an efficient detection of pivoting activity," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manag. (IM)*, 2021, pp. 980–985.

[7] G. Apruzzese, F. Pierazzi, M. Colajanni, and M. Marchetti, "Detection and threat prioritization of pivoting attacks in large networks," *IEEE Trans. Emerg. Top. Comput.*, vol. 8, no. 2, pp. 404–415, Apr.–Jun. 2020, doi: 10.1109/TETC.2017.2764885.

[8] E. Espinal and Z. Castro, "Machine learning techniques for network systems," HAL Open Archive, hal-04598713, Jun. 2024.

[9] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A design science research methodology for information systems research," *J. Manag. Inf. Syst.*, vol. 24, no. 3, pp. 45–77, Dec. 2007, doi: 10.2753/MIS0742-1222240302.

[10] K. Schwaber, *Agile Project Management with Scrum*. Redmond, WA, USA: Microsoft Press, 2004.

[11] V. Paxson, "Bro: A system for detecting network intruders in real-time," *Comput. Netw.*, vol. 31, no. 23–24, pp. 2435–2463, Dec. 1999, doi: 10.1016/S1389-1286(99)00112-7.

[12] M. Roesch, "Snort: Lightweight intrusion detection for networks," in *Proc. 13th USENIX Conf. Syst. Admin. (LISA '99)*, Seattle, WA, USA, Dec. 1999, pp. 229–238.

[13] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: 10.1023/A:1010933404324.

[14] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Proc. IEEE Int. Conf. Data Min. (ICDM)*, Pisa, Italy, Dec. 2008, pp. 413–422, doi: 10.1109/ICDM.2008.17.

[15] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Comput.*, vol. 13, no. 7, pp. 1443–1471, Jul. 2001, doi: 10.1162/089976601750264965.

[16] C. Smiliotopoulos, G. Kambourakis, and C. Kolias, "Detecting lateral movement: A systematic survey," *Heliyon*, vol. 10, no. 4, Art. no. e26317, Feb. 2024, doi: 10.1016/j.heliyon.2024.e26317.

[17] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002, doi: 10.1613/jair.953.

[18] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar, "Hyperband: A novel bandit-based approach to hyperparameter optimization," *J. Mach. Learn. Res.*, vol. 18, no. 185, pp. 1–52, 2018, doi: 10.48550/arXiv.1603.06560.

[19] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design science in information systems research," *MIS Q.*, vol. 28, no. 1, pp. 75–105, Mar. 2004, doi: 10.2307/25148625.

[20] S. Gregor and A. R. Hevner, "Positioning and presenting design science research for maximum impact," *MIS Q.*, vol. 37, no. 2, pp. 337–355, Jun. 2013, doi: 10.25300/MISQ/2013/37.2.01.

[21] J. Venable, J. Pries-Heje, and R. Baskerville, "FEDS: A framework for evaluation in design science research," *Eur. J. Inf. Syst.*, vol. 25, no. 1, pp. 77–89, Jan. 2016, doi: 10.1057/ejis.2014.36.

[22] T. Dybå and T. Dingsøyr, "Empirical studies of agile software development: A systematic review," *Inf. Softw. Technol.*, vol. 50, no. 9–10, pp. 833–859, Aug. 2008, doi: 10.1016/j.infsof.2008.01.006.

ISI Journal of Information Systems and Informatics
Published By Asosiasi Doktor Sistem Informasi Indonesia

[23] A. Hinderks, F. J. Domínguez Mayo, J. Thomaschewski, and M. J. Escalona, "Approaches to manage the user experience process in agile software development: A systematic literature review," *Inf. Softw. Technol.*, vol. 150, Art. no. 106957, Oct. 2022, doi: 10.1016/j.infsof.2022.106957.

[24] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011, doi: 10.48550/arXiv.1201.0490.

[25] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," *Comput. Secur.*, vol. 28, no. 1–2, pp. 18–28, Feb. 2009, doi: 10.1016/j.cose.2008.08.003.

[26] T. Pietraszek and A. Tanner, "Data mining and machine learning—Towards reducing false positives in intrusion detection," *Inf. Secur. Tech. Rep.*, vol. 10, no. 3, pp. 169–183, 2005, doi: 10.1016/j.istr.2005.07.001.

[27] G. González-Granadillo, S. González-Zarzosa, and R. Diaz, "Security information and event management (SIEM): Analysis, trends, and usage in critical infrastructures," *Sensors*, vol. 21, no. 14, Art. no. 4759, Jul. 2021, doi: 10.3390/s21144759.

[28] M. Ratchford, O. El-Gayar, C. Noteboom, and Y. Wang, "BYOD security issues: A systematic literature review," *Inf. Secur. J. Glob. Perspect.*, vol. 31, no. 3, pp. 253–273, May 2022, doi: 10.1080/19393555.2021.1923873.

[29] Y. Gong, S. Cui, S. Liu, B. Jiang, C. Dong, and Z. Lu, "Graph-based insider threat detection: A survey," *Comput. Netw.*, vol. 254, Art. no. 110757, Dec. 2024, doi: 10.1016/j.comnet.2024.110757.

[30] F. Moomtaheen, S. S. Bagui, S. C. Bagui, and D. Mink, "Extended isolation forest for intrusion detection in Zeek data," *Information*, vol. 15, no. 7, Art. no. 404, Jul. 2024, doi: 10.3390/info15070404.

[31] H. He, *Imbalanced Learning: Foundations, Algorithms, and Applications*, 1st ed. Somerset, NJ, USA: John Wiley & Sons, 2013.

[32] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets," *PLoS One*, vol. 10, no. 3, Art. no. e0118432, Mar. 2015, doi: 10.1371/journal.pone.0118432.