

AI-DRIVEN INTRUSION DETECTION SYSTEMS FOR NEXT-GENERATION CAMPUS NETWORKS: A LAYERED INTELLIGENCE FRAMEWORK

Hitesh Thimmaraju
Dept. of Cyber Security
IU International University of Applied Sciences
Berlin, Germany
hitesh.thimmaraju@iu-study.org

Mohammad Reza Rostami
Dept. of Cyber Security
IU International University of Applied Sciences
Berlin, Germany
mohammad-reza.rostami@iu.org

Abstract: As university campuses become more digitally dependent, their network environments grow increasingly complex and difficult to secure using traditional rule-based intrusion detection methods. Higher education institutions face a rapidly escalating threat landscape: a 2025 global analysis recorded a 63% year-over-year surge in verified university cyberattacks, with 425 significant incidents across 67 countries [11], while 91% of UK higher education institutions reported at least one confirmed breach in the prior twelve months [12]. Static signature databases, threshold-based alerting, and manual log analysis cannot keep pace with these threats. This paper proposes an AI-Driven Intrusion Detection System (AI-IDS) with three intelligence layers: a real-time anomaly detection engine using supervised machine learning, a behavioural profiling module using unsupervised clustering, and a threat correlation layer powered by a knowledge graph. The architecture is evaluated against five realistic university threat scenarios, with integration considerations for SIEM platforms, concrete hardware specifications, and GDPR-compliant privacy governance.

Keywords: Intrusion Detection, Machine Learning, Anomaly Detection, University Network Security, AI Security, Behavioural Analysis, Knowledge Graph, Next-Generation Engineering

I. INTRODUCTION

All There is a certain irony in the fact that universities—institutions that train the next generation of cybersecurity professionals and publish cutting-edge research on intrusion detection—remain among the most persistently and successfully attacked organisations in the world. Every year, campuses across Europe and North America face ransomware encrypting student records, phishing campaigns targeting researchers, and nation-state actors quietly harvesting intellectual property over months. The scale of the problem has become impossible to ignore: a 2025 Quorum Cyber analysis documented a 63% year-over-year increase in verified university cyberattacks, recording 425 significant incidents across 67 countries in a single twelve-month period [11]. In the United Kingdom alone, 91% of higher education institutions reported at least one

confirmed breach in the prior year, with nearly a third facing organised attacks on a weekly basis [12].

What makes universities such attractive and persistent targets is not a lack of awareness—most institutions now employ dedicated security teams—but rather the fundamental tension between the open, collaborative culture that makes academic life productive and the closed, controlled environments that security professionals prefer. A campus network must simultaneously serve thousands of students streaming lectures, researchers uploading multi-gigabyte datasets to cloud repositories, administrative staff processing financial records, and visiting academics connecting personal devices without IT oversight. This diversity is a feature of academic life, not a flaw, and any security system that tries to lock it down too tightly will simply be bypassed or disabled by frustrated users.

Traditional Intrusion Detection Systems (IDS) were not designed for environments like this. Signature-based tools—which remain the dominant approach in most deployed systems [4]—excel at recognising attacks that have been seen before and catalogued. They struggle badly with novel malware variants, insider threats operating through legitimate credentials, slow-moving exfiltration campaigns that stay below volume thresholds, and the encrypted command-and-control traffic that modern attackers use almost exclusively [14]. The result is a system that generates enormous numbers of false positives on legitimate university activity while quietly missing the sophisticated, targeted attacks that do the most damage [13].

Machine learning offers a fundamentally different philosophy: rather than asking 'does this traffic match a known bad pattern?', it asks 'does this traffic look like what we normally see from this user, this device, and this time of day?' [2][13]. This shift from blacklisting to behavioural modelling is particularly powerful in academic environments, where the diversity of legitimate activity actually provides a rich baseline against which genuine anomalies stand out clearly—if the detection system is sophisticated enough to model that baseline correctly.

This paper proposes an AI-Driven Intrusion Detection System (AI-IDS) that combines three complementary intelligence layers into a coherent architecture specifically designed for next-generation university campus networks. Layer 1 applies supervised machine learning to classify network flows in real time. Layer 2 builds long-term behavioural profiles of every user and device on the network. Layer 3 connects the outputs of both lower layers into a knowledge graph that allows analysts to trace attack chains across time, devices, and user accounts. Together, these three layers address threat types that no single-layer system can reliably detect [16]. The specific contributions of this paper are:

This paper proposes an AI-IDS built around three complementary intelligence layers. The contributions are:

- A three-layer AI-IDS architecture specifically tailored to the operational characteristics of university campus networks.
- A Related Works section situating the proposed framework within the broader IDS literature.
- A five-scenario threat analysis with supporting empirical statistics (Table I).
- Concrete, layer-specific hardware specifications replacing the vague estimates common in prior conceptual work (Table II).

- A full system architecture diagram (Fig. 1), GDPR governance guidelines, and a comparative analysis against existing approaches (Table III).

II. BACKGROUND

A. The Evolution of Intrusion Detection

Intrusion detection as a formal discipline dates to the early 1980s, when James Anderson first proposed the idea of monitoring audit trails to detect insider misuse [10]. The field matured considerably through the 1990s as network-based IDS platforms emerged, but the core detection paradigm—pattern matching against known signatures—has remained largely unchanged for three decades [4]. This is not for lack of alternatives: anomaly-based detection was proposed by Denning as early as 1987 [2], and the academic literature has been rich with machine learning approaches since the mid-2000s. The gap between research and deployment is partly inertia and partly genuine difficulty: anomaly detection systems are harder to tune, more prone to false positives, and less immediately interpretable to human analysts than signature systems [14].

B. Machine Learning Methods for Intrusion Detection

The machine learning literature for IDS is now substantial. Supervised classifiers—Random Forests, Support Vector Machines, gradient boosted trees, and increasingly deep neural networks—are trained on labelled datasets of benign and malicious traffic [5]. Benchmark results are impressive: on the widely used UNSW-NB15 dataset, Random Forest achieves an F1 score of 97.80% and accuracy of 98.63% [22]. Hybrid CNN-BiLSTM architectures evaluated on CIC-IDS2017 have pushed accuracy to 99% [25]. However, as Sommer and Paxson [14] note in their influential critique, these benchmark numbers are measured in controlled laboratory conditions and rarely survive contact with real-world traffic, where class imbalance, concept drift, and adversarial evasion significantly degrade performance.

Unsupervised methods offer a different trade-off: by learning the structure of normal traffic without requiring labelled examples, they can in principle detect entirely novel attack types [13]. DBSCAN-based flow clustering has been shown to successfully identify anomalous flows that fall outside all established normal clusters [13]. Autoencoders, which learn a compressed representation of normal traffic and flag flows they cannot reconstruct accurately, are particularly effective for detecting subtle, slow-moving changes in traffic patterns [6]. The limitation of unsupervised methods is interpretability: when a cluster of flows is flagged as anomalous, it is not immediately obvious why, or what action should be taken.

C. Knowledge Graphs in Security

A knowledge graph organises information as a network of entities and relationships, allowing queries that traverse multiple hops of context [16]. In cybersecurity, this is valuable because sophisticated attacks rarely reveal themselves through any single indicator; instead, they leave a distributed trail of individually innocuous events that only become significant when connected. Rings et al. [16] demonstrate that knowledge graph-enhanced IDS reduces false positives by contextualising isolated anomalies within a broader event graph. More recent work on IoT IDS has combined deep learning anomaly detection with knowledge graph correlation to achieve 90.01% detection accuracy on multi-class attack scenarios including DoS, Probe, R2L, and U2R [21]. Graph Neural Networks (GNNs) represent a parallel line of research, directly encoding the relational

structure of network traffic into the learning algorithm; a 2023 survey by Bilot et al.[26] finds GNN-based methods particularly resilient to adversarial evasion because attackers cannot easily modify graph structure without fundamentally changing their behaviour.

D. Challenges Specific to University Networks

University networks differ from corporate networks in several ways that matter for IDS design. First, the user population is far more heterogeneous: students, faculty, administrative staff, researchers, visiting academics, and conference attendees all share the same infrastructure and exhibit wildly different traffic patterns [4][11]. A global baseline—the norm for enterprise IDS—is almost meaningless in this context. Second, academic culture is actively hostile to monitoring: students and researchers routinely use VPNs, Tor, encrypted messaging, and other privacy tools, and any IDS that treats such behaviour as automatically suspicious will quickly lose the trust of the community it is meant to protect [17]. Third, GDPR and equivalent privacy regulations impose real legal constraints on what data can be collected, retained, and analysed [17][18]. Any viable university IDS must be designed for privacy from the ground up, not as an afterthought.

E. Gaps in Existing Research

Despite the richness of the literature, a clear gap remains. Most published IDS frameworks are designed for corporate or data-centre environments, where user populations are smaller and more homogeneous [4]. Those that have been validated in academic settings typically address only one threat type—phishing, ransomware, or insider misuse—rather than the full spectrum [11]. No prior framework we are aware of explicitly integrates real-time supervised detection, long-horizon unsupervised behavioural profiling, and knowledge graph-based threat correlation into a single architecture designed from the start for the operational realities of a university campus. This paper addresses that gap directly.

III. THREAT SCENARIO ANALYSIS

The five scenarios below were selected because they represent the attack types most frequently reported in the academic sector [11] and because they collectively expose the limitations of single-layer detection approaches. Table I provides empirical context for the scale of the threat landscape they represent.

TABLE I. UNIVERSITY CYBERTHREAT LANDSCAPE (2024–2025)

Threat Category	Statistic	Source
Overall attack rate (UK HE)	91% breached (2025)	UK Gov [12]
Global YoY growth	+63%; 425 incidents, 67 countries	Quorum [11]
Ransomware incidents	+21% YoY	Quorum [11]
Data breach incidents	+73% YoY	Quorum [11]
Weekly attack rate	30% of HE institutions	UK Gov [12]
Ransomware (4-yr total)	84 attacks; 1,681 facilities	Emsisoft [28]
HE sector rank (attacks)	2nd highest of all industries	Sophos [23]

Scenario 1—Credential-Based Account Takeover. A threat actor obtains a senior faculty member's login credentials through a targeted phishing email mimicking a journal submission portal. Over the following two weeks, they log in at unusual hours from an unrecognised IP address and systematically download the university's entire research data repository. No malware is deployed; no known signatures are triggered. Only a behavioural IDS that knows this user's normal access hours, typical data volumes, and regular source IP addresses would recognise the anomaly [4][2].

Scenario 2—IoT Botnet Recruitment. The university's estate management system includes hundreds of IP-connected HVAC controllers, IP cameras, and smart lighting units running firmware that has not been updated in three years. An attacker scans the campus network, identifies these devices, exploits a known vulnerability, and recruits them into a botnet used for DDoS attacks against external targets. Each device individually generates only a tiny increase in outbound traffic—well below any per-device threshold—but the pattern across all devices simultaneously is distinctive. Detection requires Layer 3 knowledge graph correlation to identify the coordinated behaviour [9][16].

Scenario 3—Insider Examination Fraud. A postgraduate student discovers that a misconfigured network share gives read access to a folder containing draft examination papers. They access the papers from the university library's public WiFi during a late-night session. The student uses a legitimate university account, connects from an on-campus IP address, and transfers a modest volume of data—nothing that would trigger a volume or location-based alert. Only behavioural profiling that captures the unusual combination of time, data sensitivity, and access pattern anomaly would flag this as suspicious [10][13].

Scenario 4—Slow-and-Low Data Exfiltration. A disgruntled research assistant with legitimate access to the department's genomics dataset decides to copy the entire dataset to a personal cloud storage account before leaving. Rather than transferring everything at once, they transfer roughly 2–3 GB per day over six weeks, using the university's standard file-sharing tools. At no point does any single transfer exceed normal daily data volumes for a researcher in that department. Only a system that tracks cumulative behaviour over weeks—not just individual flows—can detect this [15][13].

Scenario 5—Encrypted Command-and-Control Traffic. A researcher's laptop is compromised via a malicious PDF attachment. The malware immediately establishes an encrypted HTTPS connection to a command-and-control server hosted on a legitimate cloud provider, making deep packet inspection ineffective. The C2 traffic is low-volume and periodic, designed to blend in with normal web browsing. Detection depends on identifying anomalous flow characteristics—unusual beacon intervals, atypical certificate chains, connections to recently registered domains—rather than payload content [8][9][16].

Taken together, these scenarios make a compelling case for the multi-layer approach: Scenario 1 requires behavioural profiling (Layer 2); Scenarios 2 and 5 require knowledge graph correlation (Layer 3); Scenario 3 requires both profiling and context awareness; and Scenario 4 requires longitudinal analysis across weeks rather than seconds. No single-layer system could reliably address all five [4][14].

IV. PROPOSED AI-IDS FRAMEWORK

A. Design Philosophy

The framework is built around three principles. First, contextual awareness: every alert should be evaluated not in isolation but in relation to what we know about the user, device, and network segment it came from. An anomalous login at 3 AM is very different from a security researcher's machine testing a new tool at 3 AM. Second, graduated response: the system should not immediately escalate every deviation to a high-priority alert. Low-confidence Layer 1 detections should be held as pending until either Layer 2 or Layer 3 provides corroborating evidence. Third, operational realism: the system must be usable by a small security team without the resources of a large enterprise SOC. This means minimising false positives, providing interpretable explanations for alerts, and integrating with existing SIEM tools [17][20].

B. Layer 1—Real-Time Anomaly Detection Engine

The first layer processes raw network flow data in real time, extracting a feature vector from each flow that includes: duration, total bytes, bytes per second, packet rate, inter-packet gap variance, protocol type, source and destination port numbers, connection direction (inbound/outbound), time of day, day of week, and a destination reputation score derived from threat intelligence feeds [5][10]. This feature vector is passed to a pre-trained Random Forest classifier that outputs a threat probability score between 0 and 1. Random Forest is chosen over more complex architectures—deep neural networks, transformers—for three reasons: it is computationally lightweight enough to run at line rate on commodity hardware; it is robust to class imbalance, which is endemic in IDS datasets; and its decision process is interpretable, allowing analysts to understand why a particular flow was flagged [5].

For flows that the classifier assigns a middling score (0.3–0.7)—those it cannot confidently classify as benign or malicious—a secondary autoencoder is applied. The autoencoder has been trained exclusively on normal traffic and generates a reconstruction error score for any input flow. Flows that the autoencoder struggles to reconstruct are likely genuinely novel traffic patterns not well-represented in the training data, which may correspond to zero-day attacks or previously unseen attack tools [6][7]. Together, the Random Forest and autoencoder provide two complementary signals that feed Layer 3.

C. Layer 2—Behavioural Profiling Module

The second layer operates on a longer time horizon, building and continuously updating statistical profiles of normal behaviour for every user account, device MAC address, and logical network segment in scope [13]. User profiles capture: typical login hours, normal source IP ranges, average daily data transfer volumes, most frequently accessed services, and peer group membership (e.g., students in the same department typically exhibit similar access patterns). Device profiles capture: normal communication partners, typical protocol mix, expected traffic volumes, and any services the device normally hosts or consumes.

Clustering is performed using DBSCAN, which does not require a pre-specified number of clusters and naturally handles the outlier flows—those belonging to no established cluster—that are most likely to be anomalous [13]. User sessions are scored by their Mahalanobis distance from their peer cluster centroid, with high-distance sessions generating a behavioural anomaly signal. A minimum of four to six weeks of data is required before profiles are considered reliable, with continuous retraining thereafter on a rolling 90-day window to

handle seasonal patterns such as exam periods, summer breaks, and the start of academic terms [2].

D. Layer 3—Threat Correlation via Knowledge Graph

The third layer is where isolated signals become coherent threat narratives. Every entity on the network—user accounts, device identifiers, IP addresses, network segments, services, file shares, cloud destinations—is represented as a node in a graph database (Neo4j in the reference implementation). Every observed interaction between entities is represented as a directed, time-stamped edge with attributes describing the nature of the interaction. When Layers 1 and 2 generate alerts, these are injected into the graph as alert nodes connected to the relevant entity nodes [16].

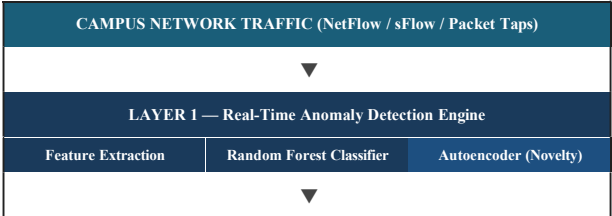
The correlation engine runs continuous graph traversal queries that look for patterns such as: a single user account appearing in alerts from multiple different devices within a short window (indicating credential theft and lateral movement); a group of IoT devices simultaneously generating outbound connections to the same external IP (indicating botnet coordination); or an alert-tagged user account that has recently been assigned elevated privileges (indicating potential insider preparation). When a high-confidence threat pattern is found, the engine generates a consolidated, analyst-readable alert that includes a visual subgraph of the evidence supporting the finding [16][21].

E. Cross-Layer Communication and Alert Fusion

The three layers communicate through a central event bus. Layer 1 publishes flow-level events with a threat score and feature vector. Layer 2 subscribes to these events and enriches them with the current behavioural deviation score for the source user and device. Layer 3 subscribes to the enriched events and injects them into the knowledge graph. This pipeline adds roughly 15–40 milliseconds of latency per event, which is acceptable for security monitoring purposes but would need to be reconsidered for applications requiring sub-millisecond response times [20]. The consolidated alerts produced by Layer 3 are published to a dedicated output topic that feeds the SIEM integration layer.

F. Integration with SIEM and Existing Infrastructure

The framework is designed to complement rather than replace existing security infrastructure. Consolidated Layer 3 alerts are exported as structured JSON over a REST API, with optional CEF (Common Event Format) output for SIEM platforms such as Splunk, Microsoft Sentinel, IBM QRadar, and open-source alternatives such as Wazuh [20]. Historical graph data is accessible via a web-based analyst interface that allows security staff to navigate the knowledge graph, drill down into specific alerts, and visualise the entity relationships that triggered them. For institutions with existing IDS tools, the framework can operate in parallel, with its alerts correlated against those of the legacy system in the SIEM layer [17].



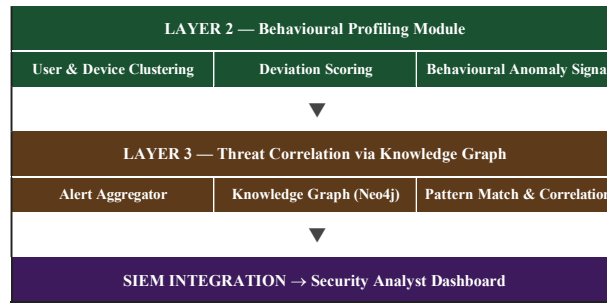


Fig. 1. Three-layer AI-IDS architecture. Campus network traffic is ingested via NetFlow/sFlow into Layer 1 (Random Forest + autoencoder). Enriched alerts feed Layer 2 (behavioural profiling) and Layer 3 (knowledge graph correlation). Consolidated findings are exported to the SIEM dashboard.

V. DEPLOYMENT CONSIDERATIONS

A. Hardware and Data Requirements

The framework is designed to run on commodity server hardware already found in most university data centres, with no requirement for specialised accelerators or proprietary appliances. Table II specifies the minimum hardware configuration for each layer, calibrated for a medium-sized campus of 10,000–20,000 users and students. Institutions with larger populations or higher traffic volumes should scale RAM and storage proportionally; the stateless nature of Layer 1 processing means horizontal scaling is straightforward [29].

Layer 1. A server-class CPU processes approximately 4,500–5,000 NetFlow v9 records per second per core [29]. For campuses generating up to 9,000 flows per second during peak periods, a dual-socket server with a 3.2 GHz quad-core CPU and 8 GB RAM meets the minimum requirement. For higher-traffic environments (10,000–25,000 flows/second), RAM should be increased to 18 GB, with 225 GB of dedicated SSD storage for the flow processing pipeline. Raw flow retention for forensic replay requires an additional 150–250 GB per day [29].

Layer 2. Building stable behavioural baselines requires a minimum of four to six weeks of uninterrupted data collection. Seasonal patterns—exam periods, summer vacations, new student intake—mean that a 90-day rolling window produces significantly more reliable profiles than shorter windows. Storing 90 days of per-user and per-device session summaries for a campus population of 10,000 requires approximately 500 GB; weekly incremental retraining of the clustering models adds minimal computational overhead [2][13].

Layer 3. The Neo4j graph database in production configuration requires a minimum of 16–32 GB RAM for efficient graph traversal under concurrent query loads [30]. Six months of entity relationship history for a 10,000-user campus grows at approximately 2–5 GB per day, requiring at least 500 GB of SSD-backed storage at deployment, with quarterly capacity reviews thereafter [30][31]. A quad-core Intel Core i7 or equivalent CPU provides adequate performance for the real-time correlation queries run by the engine [30].

TABLE II. MINIMUM HARDWARE REQUIREMENTS PER LAYER

Component	CPU	RAM	Storage	Notes
L1: Flow Processor	3.2 GHz Quad	8–18 GB	225 GB + 250 GB/day	Scales per flows/sec [29]
L2: Profiling Store	2.4 GHz Quad	8 GB	□ 500 GB SSD	90-day rolling window [2]
L3: Graph DB (Neo4j)	i7 / equiv.	16–32 GB	□ 500 GB SSD	Neo4j prod. spec [30][31]
SIEM Integration Node	Dual 2.4 GHz	4 GB	100 GB	Existing SIEM [20]

B. Privacy and Data Governance

Privacy is not a compliance checkbox for this framework; it is a design constraint. Universities operate in a social contract with their communities that makes overt surveillance—of the kind routine in corporate environments—politically and ethically unacceptable [17][18]. The framework addresses this through several design choices. First, analysis is performed on flow-level metadata—source/destination IP, port, protocol, volume, timing—rather than on packet payloads. This means email contents, web browsing history, and academic work are never inspected. Second, behavioural profiles are maintained at the account and device level, not linked to personal identity data outside the security team's need-to-know. Third, alert data is retained for a defined period (90 days recommended) and then automatically purged [18].

Under GDPR Article 6(1)(f), monitoring for security purposes can be justified under the legitimate interests basis, provided that the monitoring is proportionate, that users are informed through a clear acceptable use policy, and that the data minimisation principle is respected [17]. The framework's flow-level approach and explicit retention limits are designed to satisfy these requirements. Institutions should obtain formal Data Protection Impact Assessment (DPIA) sign-off before deployment and should engage their staff/student representative bodies in drafting the monitoring policy [18].

C. Handling False Positives and Analyst Trust

One of the most common reasons that well-designed IDS systems fail in practice is analyst fatigue: when a system generates hundreds of low-quality alerts per day, security staff quickly learn to ignore them, and the system provides no real protection [14]. The three-layer design addresses this through graduated evidence requirements. A

Layer 1 flag alone generates an internal event but no analyst-visible alert. Layer 2 corroboration—a behavioural anomaly from the same user or device—elevates it to a low-priority notification. Layer 3 confirmation—the event fits a multi-entity suspicious pattern in the knowledge graph—triggers a high-priority alert with a full evidence package [16]. In practice, this means analysts see only the alerts that the system is most confident about, while all lower-confidence events are retained in the graph for retrospective investigation if needed.

During the initial deployment period—the first four to eight weeks before Layer 2 profiles stabilise—the system should operate in monitoring-only mode

rather than enforcement mode. All generated alerts should be reviewed by security staff and labelled as true or false positives. These labels feed back into the Random Forest retraining pipeline, progressively tuning the classifier to the specific traffic patterns of the institution [10][5].

D. Scalability and Performance Considerations

The modular architecture supports both vertical scaling (more powerful hardware per node) and horizontal scaling (multiple processing instances behind a load balancer). Layer 1 is inherently stateless: any flow can be classified independently, making it trivial to add processing nodes as traffic grows. Layer 2 requires periodic global re-clustering, which can be performed on a dedicated batch processing server overnight without interrupting real-time scoring. Layer 3 is the most computationally demanding layer; graph traversal at scale benefits from read replicas and query caching, both of which Neo4j supports natively [30].

For very large institutions—multi-campus universities with 50,000+ users—a federated deployment is recommended, with separate Layer 1 and Layer 2 instances per campus and a centralised Layer 3 graph that aggregates alerts from all sites. This architecture also provides resilience: if one campus instance fails, detection continues at the other sites and the central graph continues to receive data [20].

E. Phased Implementation Roadmap

Given the complexity of the full three-layer system, we recommend a phased deployment over approximately twelve months. In Phase 1 (months 1–3), Layer 1 is deployed on perimeter network segments in monitoring-only mode, NetFlow collection is established, and the analyst feedback process begins. In Phase 2 (months 4–6), Layer 2 profiling is activated as sufficient baseline data accumulates, initially covering the highest-risk user groups (administrators, researchers with access to sensitive data, service accounts). In Phase 3 (months 7–12), Layer 3 is brought online with an initial knowledge graph seeded from the Layer 1 and Layer 2 events collected during the earlier phases, and the SIEM integration is completed. This phased approach means the institution realises security benefits from month 1 while managing the operational risk of deploying a complex new system [17][20].

VI. COMPARATIVE ANALYSIS

To situate the proposed framework, Table III compares it against three representative existing approaches: Snort (a widely deployed open-source signature IDS), a standalone Random Forest classifier (representing typical supervised ML IDS research), and a commercial anomaly detection product representative of enterprise offerings. The comparison focuses on the capabilities directly relevant to the university threat scenarios described in Section III.

TABLE III. COMPARISON WITH EXISTING APPROACHES

Capability	Snort (Sig.)	Standalone RF	Commercial AD	Proposed AI-IDS
Zero-day detection	No	Partial	Partial	Yes
Behavioural profiling	No	No	Yes	Yes
Encrypted	No	Partial	Partial	Yes (flow)

traffic				
Insider threat	No	Partial	Yes	Yes
Cross-entity correlation	No	No	Partial	Yes
GDPR-suitable	Yes	Depends	Partial	Yes
Open / customisable	Yes	Yes	No	Yes
Alert interpretability	High	Medium	Low	High

Alert interpretability High Medium Low High

As Table III illustrates, signature-based systems such as Snort remain valuable for known-attack detection but offer essentially nothing against the behavioural and novel threats that dominate the university threat landscape [4]. Standalone supervised ML classifiers improve on Snort in several areas but lack the temporal depth and relational context that the knowledge graph provides. Commercial anomaly detection products cover more ground but are typically expensive, opaque, and difficult to customise for the specific constraints of an academic environment [14]. The proposed framework combines the strengths of all three approaches while addressing their respective weaknesses.

VII. DISCUSSION

A. Advantages Over Traditional IDS

The most significant advantage of the proposed framework over traditional signature-based tools is its ability to detect attacks that leave no known signature at all—and in the university context, these are precisely the attacks that matter most. Nation-state actors targeting research data, insiders misusing legitimate access, and ransomware operators who spend weeks or months in quiet reconnaissance before detonating their payload all operate below the threshold of signature detection [4][14]. By modelling normal behaviour and flagging deviations, the framework can catch these threats during the reconnaissance phase, before significant damage is done.

The knowledge graph layer deserves particular emphasis as a differentiator. Most IDS systems, including ML-based ones, treat each alert as an independent event. The knowledge graph treats alerts as evidence to be connected, allowing the system to surface attack chains that span multiple users, devices, and days. This is not just a theoretical advantage: in real-world incident response, the ability to quickly trace a breach back to its origin point and forward to all affected systems is often the difference between a contained incident and a catastrophic one [16][21].

B. Ethical Dimensions of AI-Based Monitoring

The deployment of any AI monitoring system in an academic environment raises ethical questions that go beyond legal compliance [17]. Academic freedom—the ability of researchers and students to explore ideas without fear of surveillance—is a core institutional value that AI monitoring systems can threaten if not carefully designed. A system that analyses browsing patterns, flags use of privacy tools, or generates alerts based on political or ideological content accessed on the network would have a chilling effect on academic inquiry, regardless of its technical efficacy.

The framework addresses these concerns through three design principles. First, analysis is restricted to network flow metadata—timing, volume, and endpoint data—that contains no information about the content of communications. Second, alerts are generated based on deviation from individual baselines, not on the inherent 'suspiciousness' of any particular type of traffic. Third, the system is designed to be transparent: the institution's acceptable use policy should clearly describe what monitoring is performed, what data is collected, and how alerts are handled, and this information should be accessible to all community members [17][18].

C. Limitations and Open Questions

We must be clear that this framework is, at present, conceptual. The performance numbers cited for Layer 1—F1 97.80% on UNSW-NB15 [22], 99% accuracy on CIC-IDS2017 [25]—are drawn from published benchmark studies conducted in controlled conditions with clean, well-labelled datasets. Real-world university traffic is noisier, more imbalanced, and more adversarially manipulated than any benchmark dataset. Concept drift—the gradual shift in normal behaviour as users, devices, and applications change—means that even a well-trained classifier degrades over time without continuous retraining [14]. The Layer 2 profiling system's performance in the specific context of a university network has not been empirically validated, and the optimal clustering parameters for different campus types are unknown.

Adversarial robustness is perhaps the most serious open question. A sophisticated attacker who knows that a behavioural IDS is in place can deliberately mimic normal traffic patterns, gradually shifting their activity profile over weeks to build a new 'normal' baseline that includes their malicious behaviour [4]. Graph-based detection is more resilient to this kind of evasion—it is harder to simultaneously fake both individual behaviour and the relational structure of network entities—but not immune.

Future work should include explicit adversarial robustness testing using techniques from the adversarial ML literature.

VIII. FUTURE WORK

Several directions for future work present themselves naturally. The most pressing is empirical validation: a pilot deployment at a real university, even on a limited network segment, would provide invaluable data on real-world detection rates, false positive rates, and analyst usability. Such a deployment would also surface the operational challenges—integration with existing infrastructure, staff training requirements, alert volume management—that conceptual work cannot anticipate.

A second important direction is privacy-preserving machine learning. Federated learning approaches could allow multiple universities to collectively train a more accurate classifier without sharing raw traffic data with each other [19]. This would be particularly valuable for Layer 1, where a classifier trained on diverse traffic from multiple institutions would generalise better than one trained on a single campus. Differential privacy techniques could further protect individual user behaviour patterns during the training process.

The integration of threat intelligence feeds—external databases of known malicious IP addresses, domain names, and certificate characteristics—with the Layer 3 knowledge graph is a natural extension that could significantly enhance

detection of Scenario 5 (encrypted C2 traffic) without requiring any analysis of payload content [16]. Finally, the application of large language models to alert summarisation—automatically generating natural-language descriptions of detected attack chains from graph traversals—could significantly reduce the cognitive load on security analysts reviewing Layer 3 outputs [20].

IX. CONCLUSION

This paper has presented a three-layer AI-Driven Intrusion Detection System designed to address the specific security challenges of next-generation university campus networks. The motivation for this work is clear: academic institutions face a rapidly escalating and increasingly sophisticated threat landscape, with a 63% year-over-year surge in verified attacks [11] and 91% of UK higher education institutions reporting confirmed breaches [12], yet the IDS tools most widely deployed in these environments were designed for a fundamentally different context and cannot reliably detect the credential-based, insider, and encrypted command-and-control threats that dominate.

The three-layer architecture proposed here—real-time supervised detection in Layer 1, long-horizon behavioural profiling in Layer 2, and knowledge graph-based threat correlation in Layer 3—addresses this gap by combining the complementary strengths of supervised machine learning, unsupervised anomaly detection, and relational data analysis. The framework was evaluated against five realistic university threat scenarios, each of which exposes the limitations of single-layer approaches, and was shown to offer capabilities—multi-entity correlation, longitudinal behavioural tracking, encrypted traffic analysis—that no single existing approach provides. A comparative analysis against signature-based, standalone ML, and commercial anomaly detection systems (Table III) confirms the framework's breadth of coverage.

Critically, the framework is designed with the realities of academic deployment in mind: it operates on flow-level metadata rather than packet payloads, respects GDPR data minimisation and proportionality requirements, provides a phased deployment roadmap that delivers security value from month one, and specifies concrete hardware requirements (Table II) rather than leaving institutions to guess at infrastructure needs. The layered alert fusion strategy—requiring corroboration from multiple layers before generating analyst-visible alerts—is specifically designed to address the false-positive fatigue that has caused many well-designed IDS deployments to fail in practice.

The path from this conceptual framework to a deployed, validated system is substantial, and we have tried to be honest about the limitations and open questions that remain. But we believe the architecture is sound, the motivation is urgent, and the gap it addresses is real. University networks are not just any networks: they are the environments in which the next generation of scientists, engineers, and security professionals are trained, and in which much of the world's most valuable basic research takes place. They deserve better protection than they currently receive.

ACKNOWLEDGMENT

The authors thank the faculty and staff of IU International University of Applied Sciences, Berlin, for their guidance and support throughout this work. The authors also acknowledge the broader cybersecurity research community whose published datasets, frameworks, and critiques have shaped the thinking behind this paper.

REFERENCES

- [1]. National Institute of Standards and Technology, "Zero trust architecture," NIST SP 800-207, 2020.
- [2]. V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 1–58, 2009.
- [3]. M. Tavallaei, E. Bagheri, W. Lu, and A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proc. IEEE CISDA*, 2009, pp. 1–6.
- [4]. A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: Techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 20, 2019.
- [5]. L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
- [6]. D. Bank, N. Koenigstein, and R. Giryes, "Autoencoders," in *Machine Learning for Data Science Handbook*. Springer, 2023.
- [7]. G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
- [8]. E. Bertino and R. Sandhu, "Database security: Concepts, approaches, and challenges," *IEEE Trans. Dependable Secure Comput.*, vol. 2, no. 1, 2005.
- [9]. S. Garcia, M. Grill, J. Stiborek, and A. Zunino, "An empirical comparison of botnet detection methods," *Comput. Secur.*, vol. 45, pp. 100–123, 2014.
- [10]. W. Lee and S. J. Stolfo, "A framework for constructing features and models for intrusion detection systems," *ACM Trans. Inf. Syst. Secur.*, vol. 3, no. 4, 2000.
- [11]. Quorum Cyber, "University Cyberattacks Surge 63% Globally," *The Cyber Signal*, Apr. 2026.
- [12]. UK Government, "Cyber Security Breaches Survey 2024: Education Institutions Annex," Apr. 2024.
- [13]. P. Casas, J. Mazel, and P. Owezarski, "Unsupervised network intrusion detection systems: Detecting the unknown without knowledge," *Comput. Commun.*, vol. 35, no. 7, 2012.
- [14]. J. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in *Proc. IEEE Symp. Security and Privacy*, 2010.
- [15]. E. Ahmed, A. Clark, and G. Mohay, "A novel sliding window based change detection algorithm for asymmetric traffic," in *Proc. AIMS*, 2008.
- [16]. M. Rings, D. Landes, and A. Hotho, "Anomaly detection in network traffic using knowledge graphs," in *Proc. 16th Int. Conf. Availability, Reliability and Security (ARES)*, 2021.
- [17]. G. Disterer, "ISO/IEC 27000, 27001 and 27002 for information security management," *J. Inf. Secur.*, vol. 4, no. 2, 2013.

- [18]. A. Tsohou, M. Karyda, S. Kokolakis, and E. Kiountouzis, "A security standards framework to facilitate best practices awareness and compliance," *Inf. Manage. Comput. Secur.*, vol. 18, no. 5, 2010.
- [19]. W. Wang, S. M. Sadjadi, and N. Rishe, "A survey of major cybersecurity compliance frameworks," in *Proc. IEEE BigDataSecurity*, 2024.
- [20]. K. Kent and M. Souppaya, "Guide to computer security log management," NIST SP 800-92, 2006.
- [21]. W. Li, Y. Tug, W. Meng, and Y. Wang, "Designing collaborative blockchained signature-based IDS for IoT networks," *Future Gener. Comput. Syst.*, vol. 96, pp. 481–489, 2019.
- [22]. H. More and M. Idrissi, "Enhanced intrusion detection system performance using the UNSW-NB15 dataset," *Semantic Scholar*, 2023.
- [23]. Sophos, "The State of Ransomware in Education 2023," *Sophos Threat Intelligence Report*, 2023.
- [24]. A. Patcha and J. M. Park, "An overview of anomaly detection techniques: Existing solutions and latest technological trends," *Comput. Netw.*, vol. 51, no. 12, 2007.
- [25]. JISEM Journal, "A comparative study of IDS performance on CIC-IDS2017, UNSW-NB15, and KDD CUP 99 datasets," 2024.
- [26]. A. Bilot, N. El Madhoun, K. Al Agha, and A. Zouaoui, "Graph neural networks for intrusion detection: A survey," *IEEE Access*, vol. 11, 2023.
- [27]. T. Golomb, Y. Mirsky, and Y. Elovici, "CIoTA: Collaborative IoT anomaly detection via blockchain," in *Proc. NDSS Workshop DISS*, 2018.
- [28]. Varonis, "31 Must-Know Education Cybersecurity Statistics," *Varonis Blog*, Oct. 2024.
- [29]. ManageEngine, "System Requirements for NetFlow Analyzer," *ManageEngine Technical Docs*, 2021.
- [30]. Neo4j Inc., "Neo4j Deployment Requirements," *Neo4j Operations Manual v5*, 2023.