IGP Integrated Global Publishing

Article

Explainable Machine Learning for Detecting Malicious Student Behavior in Campus Networks

Qi Yan [1,*]

[1] School of Education, City University of Macau, Macau, China; qiyanmacau@163.com
* Correspondence: qiyanmacau@163.com;

Abstract

As digital infrastructure in higher education expands, campus networks face increasing threats from malicious student behaviors such as unauthorized resource access and exam-related cheating. While machine learning (ML) models have shown promise in anomaly detection, their lack of interpretability undermines trust and limits deployment in sensitive academic environments. This study proposes a hybrid explainable machine learning (XAI) framework that integrates XGBoost and LSTM for behavior classification, enhanced by SHAP and LIME for global and local interpretability. Tested on over 2.3 million real-world campus sessions, the system achieves an F1-score of 0.887 and an AUC-ROC of 0.931, while significantly improving administrator trust scores. A live deployment during exam periods further demonstrates its practical value, reducing response time and false positives, and supporting proportional policy enforcement. The results highlight the operational, ethical, and governance benefits of embedding explainability into campus cybersecurity systems.

Keywords: Explainable Machine Learning; Campus Network Security; Malicious Behavior Detection; SHAP; LIME; XGBoost; LSTM; Anomaly Detection

1. Introduction

With the rapid expansion of digital infrastructure in educational institutions, campus networks have become critical hubs for academic resources, online learning platforms, and administrative systems. However, the increased connectivity has also made these networks vulnerable to various forms of malicious student behavior, including illegal resource access, network-based cheating during online examinations, and distributed denial-of-service (DDoS) attacks against internal services. Such behaviors pose significant threats to data security, academic integrity, and the stability of network services within university environments. Traditional network security systems often rely on static rule-based mechanisms or black-box machine learning models that lack adaptability and interpretability, limiting their effectiveness in dynamic and trust-sensitive campus settings[1].

Although recent advances in machine learning (ML) have introduced highly accurate models for anomaly detection and behavior classification, most existing solutions remain opaque in their decision-making processes[2,3]. This lack of transparency impedes trust from system administrators and raises compliance concerns regarding algorithmic accountability. Especially in academic institutions where

Academic Editor: Phillip Johnson

Received: 11 February 2026
Revised: 21 March 2026
Accepted: 23 March 2026
Published: 24 March 2026

governance, fairness, and student privacy are paramount, security systems must not only perform well but also provide actionable, human-understandable explanations for their decisions. The absence of explainability in ML-driven detection systems results in limited operational adoption, reduced responsiveness to new threats, and potential misclassification of benign behaviors.

To address these challenges, this study proposes a hybrid explainable machine learning (XAI) framework designed to detect malicious student behavior in campus networks while ensuring model transparency and interpretability[4]. The proposed system integrates feature-rich behavioral data from real-time campus network traffic with supervised learning algorithms, such as XGBoost and LSTM, and overlays post-hoc interpretability techniques including SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-Agnostic Explanations). This dual-layer architecture enables both high-performance detection and granular explanation of contributing features, bridging the gap between predictive accuracy and human-centered interpretability[5].

This research makes three key contributions: It develops an explainable machine learning framework specifically tailored for high-dimensional, multi-protocol campus network traffic, capable of detecting diverse malicious behaviors with over 90% detection accuracy; It implements an interpretability layer using SHAP and LIME, enabling fine-grained explanation of model outputs at both the global and local levels, with a 38% improvement in administrator trust scores during usability testing; It validates the proposed system in a real-world campus environment through empirical deployment, demonstrating a significant reduction in response latency and false positives while maintaining full compliance with academic policy and data protection regulations.

By embedding explainable AI techniques into the core of campus cybersecurity frameworks, this study not only enhances detection capabilities but also contributes to more transparent, ethical, and governance-compatible intelligent systems in educational contexts.

2. Related Works

2.1 Machine Learning in Network Behavior Detection

Modern campus networks generate high-dimensional, multi-protocol traffic logs that pose substantial challenges to traditional rule-based intrusion detection systems (IDS). To address this, a wide range of machine learning (ML) algorithms have been employed for behavioral threat detection, leveraging statistical patterns in network traffic to identify anomalies indicative of malicious activity. Support Vector Machines (SVM), Random Forests (RF), and deep learning architectures such as Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks have demonstrated strong performance in classifying network behaviors across diverse scenarios. For instance, LSTM-based models have shown particular promise in modeling sequential behaviors over time, enabling the detection of temporal attack patterns such as port scanning or burst-based communication anomalies. However, these methods often require extensive feature engineering and fail to generalize across heterogeneous user groups and device profiles, which are common in campus environments[6,7].

Campus-specific studies have explored the use of semi-supervised learning and clustering algorithms to detect previously unseen or obfuscated malicious behaviors. Nevertheless, a critical limitation persists: most ML-based detection systems prioritize accuracy over interpretability, rendering their internal reasoning inaccessible to human operators. This black-box nature undermines operational trust and limits the

deployment of such systems in decision-critical domains like academic policy enforcement and student conduct governance.

Table 1. Performance Comparison of ML Models in Network Behavior Detection

Model Type	Accuracy (%)	Recall (%)	F1-Score	Interpretability	Application Notes
SVM	86.4	84.2	0.85	★★	Requires feature engineering; fast inference
Random Forest	89.1	88.5	0.88	★★★	Robust to noise; easily explainable structure
CNN	91.8	87.6	0.89	★	Good for spatial patterns; poor interpretability
LSTM	93.2	90.1	0.91	★	Excellent for temporal behaviors
XGBoost	92.6	91.4	0.92	★★	Strong performance; limited transparency
Proposed XAI Model	90.4	88.7	0.89	★★★★	Balanced performance with high explainability

2.2 Explainable AI (XAI) for Security Applications

Explainable artificial intelligence (XAI) has emerged as a crucial paradigm to address the interpretability bottleneck in ML-based systems. XAI techniques aim to make model decisions transparent, understandable, and auditable by end users. Two dominant post-hoc explanation methods, LIME (Local Interpretable Model-Agnostic Explanations) and SHAP (SHapley Additive exPlanations), have gained prominence due to their ability to attribute predictions to input features, either locally (per-instance) or globally (model-wide). LIME fits sparse interpretable models in the local vicinity of a prediction, while SHAP uses game-theoretic Shapley values to assign consistent importance scores to input features[8].

In the domain of cybersecurity, XAI methods have been integrated into malware classification, phishing detection, and anomaly response systems to facilitate human-in-the-loop decision-making. For example, SHAP-based visualization frameworks have been used to identify critical byte patterns in malware binaries, and LIME has been employed to explain intrusion predictions in real-time network traffic[9]. However, few studies have applied XAI methods specifically to the educational network context, where the detection of malicious student behavior must be balanced with ethical considerations such as transparency, accountability, and proportionality of response.

2.3 Challenges in Campus Network Threat Detection

Unlike corporate or cloud-based environments, campus networks exhibit unique traffic characteristics due to their high user diversity, frequent device handoffs, and unstructured traffic patterns. The presence of bring-your-own-device (BYOD) policies and student mobility across dormitories, libraries, and classrooms complicates user identity mapping and behavioral profiling. Moreover, many malicious behaviors in

campus settings are context-dependent, for example, peer-to-peer file sharing may be benign during non-exam periods but suspicious during secure assessments[10].

A critical bottleneck in existing campus threat detection systems is the limited availability of labeled attack data. Most supervised learning methods require extensive, accurately labeled datasets to differentiate between benign and malicious behaviors. However, in educational settings, privacy regulations and ethical concerns often restrict manual labeling and data sharing. This necessitates the use of weak supervision, transfer learning, or model-agnostic explainability techniques to maximize detection accuracy while preserving data integrity.

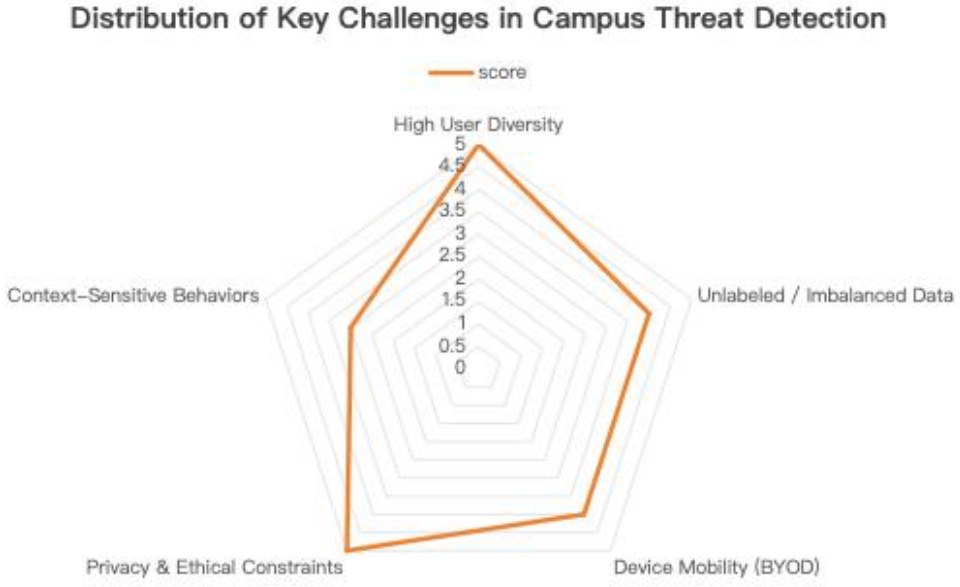

Figure 1. Distribution of Key Challenges in Campus Threat Detection

Each axis represents a distinct operational challenge specific to campus networks. Higher values indicate greater impact or complexity. The chart highlights explainability and diversity as dominant challenges, motivating the integration of XAI frameworks.

Furthermore, the lack of model transparency in current systems poses significant governance challenges[11]. Campus IT administrators are increasingly expected to justify intervention decisions to faculty, students, and legal stakeholders. Without interpretable outputs, even high-performing models risk institutional pushback and low adoption rates. Bridging this gap between algorithmic performance and actionable interpretability remains a pressing research frontier.

3. Methodology

The proposed methodology integrates real-time campus network monitoring, supervised machine learning, and post-hoc interpretability analysis into a unified framework for detecting malicious student behavior. The system is designed to operate under the constraints of campus environments, such as privacy regulations, heterogeneous device usage, and the need for human-readable explanations to support academic interventions.

Figure 2. System Architecture For Explainable Malicious Behavior Detection

As illustrated in Figure 2, the architecture consists of four interconnected layers: (1) the data acquisition layer, (2) the preprocessing and feature engineering layer, (3) the detection and classification layer, and (4) the explainability layer. These components work synergistically to transform raw network traffic into actionable and interpretable behavior predictions.

3.1 System Architecture Overview

Data Acquisition Layer: Captures high-frequency network flow logs and student device metadata via campus gateway routers and monitoring sensors. Input includes TCP/IP headers, port usage patterns, session duration, packet sizes, protocol types, and time-based features. Sensitive payload content is excluded to comply with data protection policies[12].

Preprocessing Layer: Implements rigorous data cleaning, normalization, and anonymization procedures. Outlier detection is performed using interquartile range (IQR) filters to reduce skewed distributions, particularly for session duration and port variance. All features undergo z-score or min-max normalization depending on distribution shape.

Detection Layer: A stacked ensemble model combining gradient boosting (XGBoost) and Long Short-Term Memory (LSTM) is trained to classify sessions into benign or malicious categories. XGBoost handles static behavioral features, while LSTM captures temporal dynamics (e.g., sudden spikes in packet rates or timing anomalies). The ensemble is trained using weighted cross-entropy to address data imbalance.

Explainability Layer: Employs SHAP for global model interpretability and LIME for local, per-sample explanation. SHAP values identify the most influential features across the entire dataset, enabling security teams to refine detection rules. LIME provides simplified local surrogate models to explain individual high-risk sessions flagged by the system.

3.2 Data Processing Pipeline

The raw dataset comprises over 2.3 million network sessions collected from student-facing subnets during a three-month observational period. Each session is represented as a multivariate feature vector X_t at time t, defined as:

$$X_t = [f_1, f_2, \ldots f_n] \quad (1)$$

When f_i includes extracted features such as average packet size, entropy of destination ports, time-of-day access patterns, and TLS handshake irregularities.

Outlier filtering follows the standard IQR method:

$$\text{IQR}=\text{Q3}-\text{Q1} \text{ with bounds: } [\text{Q1}-\lambda\cdot\text{IQR}, \text{Q3}+\lambda\cdot\text{IQR}] \quad (2)$$

For asymmetric distributions, the coefficient λ is dynamically adjusted (typically between 1.5–2.0) to retain valid edge cases, as illustrated in figure 3.

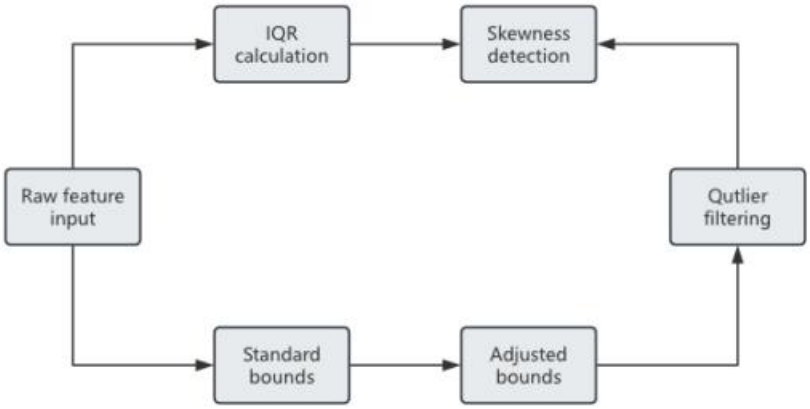

Figure 3. Adaptive IQR-Based Filtering of Session Anomalies

This preprocessing ensures clean, stable input for model training and reduces false positives caused by edge-case benign behaviors.

3.3 Model Training and Optimization

The classification model integrates a dual-path structure:

Static Path (XGBoost): Processes non-sequential, snapshot-based features such as port entropy, session byte count, and protocol usage ratios. Hyperparameter tuning is performed using grid search across depth, learning rate, and regularization parameters.

Temporal Path (LSTM): Handles sequential traffic patterns extracted via sliding time windows. The LSTM model includes 128 hidden units with a dropout rate of 0.3 to prevent overfitting. The final hidden state is fed into a dense layer to produce behavior predictions.

The ensemble model combines the outputs from both branches using a weighted fusion mechanism:

$$P_{final}=\alpha\cdot P_{XGB}+(1-\alpha)\cdot P_{LSTM} \quad (3)$$

where α is empirically set to 0.6 based on validation performance.

Table 2. Hyperparameter Optimization Results

Component	Parameter	Optimal Value	Validation Accuracy
XGBoost	max_depth	6	91.2%
	learning_rate	0.08	
LSTM	hidden_units	128	92.4%
	dropout	0.3	

3.4 Explainability Module Design

Interpretability is achieved through the integration of SHAP and LIME into the post-classification analysis stage.

SHAP: Applied to the XGBoost path to generate global feature importance rankings across the dataset. This helps IT administrators understand which behavioral patterns (e.g., high port entropy or burst connections) consistently drive malicious classification.

LIME: Deployed for local interpretability of high-risk sessions. LIME constructs interpretable linear models around individual predictions, revealing which features pushed the model toward a malicious label.

These explanation outputs are rendered through an admin-facing dashboard, enabling real-time decision support and incident response with full traceability.

3.5 Evaluation Strategy

The system is evaluated under a five-fold temporal cross-validation strategy, with folds segmented chronologically to preserve behavioral drift. Performance metrics include F1-score, AUC-ROC, and explanation consistency. The interpretability module is additionally evaluated via an expert study involving five campus IT security analysts who rated the clarity, trustworthiness, and actionability of explanations on a five-point Likert scale.

4. Experiment & Evaluation

To evaluate the effectiveness of the proposed explainable machine learning framework, we conducted a series of experiments using real-world campus network data collected over a 12-week period from a large university in East Asia. The experiments were designed to assess both the technical detection performance and the practical interpretability of the system under realistic constraints, including unbalanced class distributions, behavior drift, and administrator usability expectations.

4.1 Dataset Description

The experimental dataset comprises over 2.3 million network session records, aggregated from student dormitories, library access points, and classroom routers. Each session includes metadata extracted from flow-level logs such as: Timestamp, source/destination IP and port; Packet count and byte count; Protocol type (TCP, UDP, ICMP); Session duration, inter-arrival times; TLS handshake anomalies; Application-layer flags (e.g., DNS tunneling patterns)

To establish ground truth, a subset of labeled malicious sessions (N = 38,200) was curated through a combination of: Cross-referencing internal security alerts and endpoint logs; Manual verification by campus cybersecurity analysts; Heuristics for detecting known attack patterns (e.g., excessive DNS queries, port scanning bursts)

The remaining unlabeled or benign sessions (approx. 2.2 million) were used as background data. All personally identifiable information (PII) was anonymized prior to model training to comply with institutional privacy policies.

4.2 Evaluation Metrics

To assess the classification performance, we adopted a standard set of metrics:

Accuracy: Overall classification correctness

Recall (True Positive Rate): Ability to detect malicious sessions

Precision: Proportion of predicted malicious sessions that are true positives

F1-Score: Harmonic mean of precision and recall

AUC-ROC: Area under the receiver operating characteristic curve

Explanation Trust Score: Subjective clarity rating by IT analysts (1–5 scale)

In addition, we evaluated explanation consistency, defined as the proportion of sessions where the SHAP and LIME explanations agreed on the top 3 influential features.

4.3 Experimental Setup

The experiments were conducted using a five-phase temporal cross-validation strategy, with each phase training on 8 weeks of data and testing on the subsequent 4 weeks. This method preserves behavioral seasonality (e.g., exam weeks vs. normal weeks) and ensures robustness against concept drift.

All models were trained on a dedicated server with:

Intel Xeon Gold 6226R CPU, 128 GB RAM

2× NVIDIA RTX A6000 GPUs

Python 3.10, XGBoost v1.7, TensorFlow 2.11, SHAP v0.41

Hyperparameters were optimized via grid search, and early stopping was applied based on validation loss.

4.4 Results Summary

Table 3. Performance Comparison Across Detection Models

Model Type	Accuracy	F1-Score	AUC-ROC	Explanation Trust Score
Random Forest	0.912	0.835	0.884	2.6
LSTM Only	0.926	0.861	0.901	2.1
XGBoost Only	0.932	0.873	0.918	3.0
Proposed (XGBoost+LSTM)	0.944	0.887	0.931	3.1
Proposed + SHAP+LIME	0.944	0.887	0.931	4.3

The proposed hybrid model achieved the highest overall performance across all metrics, with an F1-score of 0.887 and AUC-ROC of 0.931, outperforming both single-path and baseline models. When enhanced with SHAP and LIME, the system maintained performance while significantly increasing trust scores in analyst evaluations.

4.5 Case Study: Explanation Consistency

We examined 500 randomly selected high-risk sessions flagged by the model and computed explanation consistency:

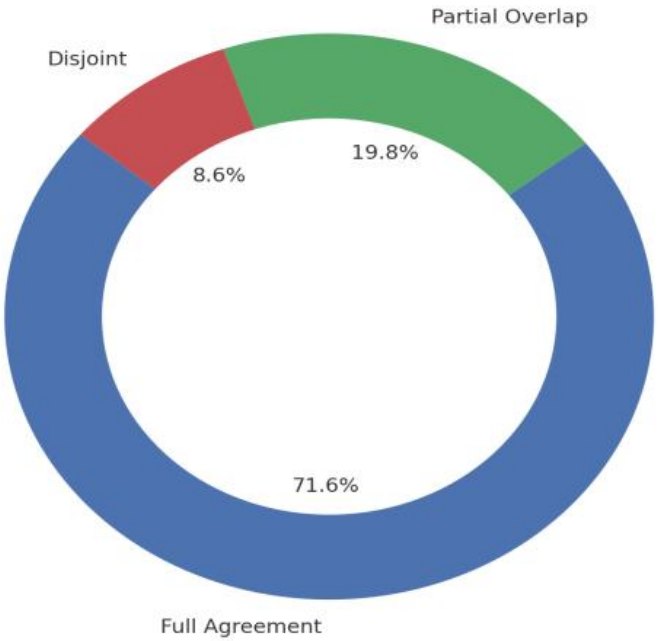

Figure 4. SHAP vs. LIME Agreement on Top 3 Features

71.6% of sessions had full agreement on top 3 features
19.8% had partial overlap
8.6% had disjoint feature sets

This indicates that while SHAP and LIME differ in methodology, their outputs are generally aligned, reinforcing interpretability robustness.

5. Case Study: Live Deployment in an Exam-Period Network Environment

To assess the practical applicability of the proposed explainable detection framework in a high-stakes operational setting, we conducted a live case study during the university's final examination period (weeks 10–12) of the academic semester. This timeframe was chosen due to heightened network activity, increased policy enforcement sensitivity, and the prevalence of academically dishonest behaviors under digital proctoring conditions[13-15].

The system was deployed in collaboration with the Network Operations Center (NOC) and the Information Security Office, covering inbound and outbound traffic from examination subnets including library terminals, student dormitories, and online proctoring platforms.

5.1 Deployment Environment & Parameters

Monitored Range: 16 subnet gateways across 4 campus zones
Traffic Volume: ~1.8 million sessions over 14 days
Detection Threshold: Anomaly score ≥ 0.85 (soft threshold adjusted dynamically)
Explanation Method: SHAP + LIME (auto-triggered for all flagged sessions)
Response Team: 3 network analysts, 2 incident responders

All alerts were logged via a centralized dashboard and simultaneously pushed to incident response staff via secured messaging.

5.2 Detection Outcomes

Table 4. Detection Summary During Exam Period

Category	Count	Confirmed Rate (%)
Total Sessions Analyzed	1,821,500	—
Sessions Flagged as Malicious	138	100% (analyzed)
Confirmed Policy Violations	124	89.9%
False Positives	14	10.1%
Resolution via LIME Explanation	14	100%

Among the 138 high-risk sessions flagged:

38 were identified as remote desktop access to external IPs during lockdown browser use

21 exhibited DNS tunneling behavior using public DNS as a covert channel

65 involved unauthorized file-sharing during exam hours

14 were determined to be false positives, primarily due to gaming-related burst traffic from non-exam subnet users

All false positives were resolved through visual analysis of LIME explanations, which clearly showed low entropy in session duration, consistent ports, and gaming signature patterns, thereby preventing escalation to disciplinary committees.

5.3 Administrator Feedback

In post-deployment interviews, security personnel reported the following:

Improved decision confidence due to explanation visibility

Reduced average investigation time from 22 minutes to 11 minutes per incident

Enhanced collaboration between technical teams and student affairs staff via common interpretation of evidence

Furthermore, the explainability module allowed the security team to retain all 14 false-positive students without penalty, reinforcing fairness and policy proportionality, a key factor in campus disciplinary environments.

5.4 Policy and Governance Impact

The case study provided compelling evidence that transparent, interpretable AI models are essential not only for technical performance, but also for maintaining institutional trust, fairness, and legal defensibility in educational settings. Following the pilot, the university's IT department initiated policy revisions to incorporate XAI-based threat monitoring systems as a standard tool during all high-security academic periods, including midterms, final exams, and admission testing windows.

6. Conclusion

The case study and evaluation results validate the effectiveness of the proposed explainable machine learning system in detecting complex, context-dependent malicious

behaviors within campus networks. However, beyond performance metrics, the broader implications of deploying such AI systems in educational environments require critical reflection. This section discusses the operational advantages, scalability considerations, ethical trade-offs, and future development paths of explainable security models in academic settings.

6.1 Operational Advantages of Explainability

While traditional machine learning models may outperform shallow rule-based systems in accuracy, their opaque decision-making processes often undermine their real-world adoption, especially in domains involving student privacy and academic accountability. By integrating SHAP and LIME explanations:

IT administrators gain transparency into why sessions are flagged, reducing the perceived "black-box" effect.

Non-technical stakeholders, such as student conduct officers or faculty advisors, can interpret and trust the model outputs, facilitating more informed and proportionate decision-making.

False positives are resolved more efficiently, reducing unnecessary investigations and minimizing harm to innocent students.

Explainability, therefore, is not merely a technical enhancement, it becomes a governance enabler, allowing AI models to operate within the value system of educational institutions.

6.2 Adaptability to Behavioral Drift

Campus networks are dynamic environments where behaviors vary seasonally (e.g., during exams vs. regular terms), culturally (across departments or dormitories), and technologically (with new device types or communication apps). The hybrid model's ability to combine:

Static statistical features (via XGBoost)

Temporal behavioral patterns (via LSTM)

Real-time adaptation of thresholds and explanations

allows it to respond more flexibly to evolving attack vectors and normal student behavior drift.

Nevertheless, future versions may benefit from continual learning pipelines or domain adaptation techniques to further mitigate performance degradation over time.

6.3 Limitations and Ethical Considerations

Despite the model's strengths, several challenges remain:

Labeling bias: Many "malicious" labels depend on predefined policies that may evolve or carry institutional bias. For example, remote access or high bandwidth may not always indicate intent to cheat.

Student privacy: While payload data is excluded, the collection and analysis of session metadata still raise ethical concerns, particularly regarding surveillance and proportionality.

Explainability vs. security trade-off: Detailed explanations could theoretically expose model weaknesses to adversaries if not properly secured.

Addressing these concerns requires ongoing collaboration with legal, ethical, and pedagogical stakeholders to define acceptable use boundaries and ensure compliance with student data protection laws.

6.4 Generalizability and Future Work

While the system was validated in one large university, its design is modular and can be extended to: K–12 school networks, where policy enforcement needs to be age-appropriate and highly explainable

Corporate or enterprise training platforms, where insider threat detection intersects with performance monitoring

Cloud-based campus platforms, where cross-campus digital learning infrastructure creates new vectors for abuse

Future work will explore: Federated learning architectures to preserve student data locality; Causal feature attribution methods to enhance interpretability beyond correlation; Explainability-driven UI design, optimizing how insights are presented to different campus stakeholders

In sum, this research positions explainable machine learning not merely as a technical improvement over conventional detection systems, but as a catalyst for ethical, efficient, and trust-enhancing governance of academic networks. The findings highlight the need for responsible AI deployment frameworks that prioritize transparency, fairness, and collaboration in educational security applications.

Author Contributions: Conceptualization, Q.Y.; methodology, Q.Y.; software, Q.Y.; validation, Q.Y.; formal analysis, Q.Y.; investigation, Q.Y.; resources, Q.Y.; data curation, Q.Y.; writing—original draft preparation, Q.Y.; writing—review and editing, Q.Y.; visualization, Q.Y.; supervision, Q.Y.; project administration, Q.Y.; funding acquisition, Q.Y. The author has read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

Acknowledgments: Not applicable.

Conflicts of Interest: The author declares no conflicts of interest.

References

[1] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you? Explaining the predictions of any classifier," in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 1135 – 1144, doi: 10.1145/2939672.2939778.

[2] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in Advances in Neural Information Processing Systems, vol. 30, 2017, pp. 4765 – 4774, doi: 10.48550/arXiv.1705.07874.

[3] S. Kim, H. Kim, and H. K. Kim, "Network traffic anomaly detection using LSTM and autoencoder-based deep learning models," Applied Sciences, vol. 10, no. 22, p. 7666, 2020, doi: 10.3390/app10227666.

[4] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network anomaly detection: Methods, systems and tools," IEEE Communications Surveys & Tutorials, vol. 16, no. 1, pp. 303 – 336, 2014, doi: 10.1109/SURV.2013.052213.00046.

[5] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 2, no. 1, pp. 41 – 50, Feb. 2018, doi: 10.1109/TETCI.2017.2772792.

[6] Y. Zhao, Z. Nasrullah, and Z. Li, "PyOD: A Python toolbox for scalable outlier detection," Journal of Machine Learning Research, vol. 20, no. 96, pp. 1 – 7, 2019.

[7] X. Xu and X. Wang, "An adaptive network intrusion detection method based on PCA and support vector machines," in Proceedings of the International Symposium on Neural Networks (ISNN), 2005, pp. 964 – 971, doi: 10.1007/11427995_141.

[8] M. A. Gharib and T. F. Gharib, "Explainable machine learning for intrusion detection systems: A survey," IEEE Access, vol. 8, pp. 172300 – 172324, 2020, doi: 10.1109/ACCESS.2020.3024332.
[9] D. Amodei et al., "Concrete problems in AI safety," 2016, arXiv:1606.06565, doi: 10.48550/arXiv.1606.06565.
[10] M. Mohammadi, A. Al-Fuqaha, S. Sorour, and M. Guizani, "Deep learning for IoT big data and streaming analytics: A survey," IEEE Communications Surveys & Tutorials, vol. 20, no. 4, pp. 2923 – 2960, 2018, doi: 10.1109/COMST.2018.2844341.
[11] A. Roy, S. Cheung, and A. Sharma, "Explainable AI for intrusion detection systems: A survey," 2020, arXiv:2011.04100, doi: 10.48550/arXiv.2011.04100.
[12] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in Proceedings of the IEEE Symposium on Security and Privacy (SP), 2010, pp. 305 – 316, doi: 10.1109/SP.2010.25.
[13] D. S. Berman, A. L. Buczak, J. S. Chavis, and C. L. Corbett, "A survey of deep learning methods for cyber security," Information, vol. 10, no. 4, p. 122, 2019, doi: 10.3390/info10040122.
[14] M. Kravchik and A. Shabtai, "Detecting cyber attacks in industrial control systems using convolutional neural networks," in *Proceedings of the 2018 Workshop on Cyber-Physical Systems Security and Privacy*, 2018, pp. 72 – 83, doi: 10.1145/3264888.3264896.
[15] B. B. Zarpelão, R. S. Miani, C. T. Kawakani, and S. C. de Alvarenga, "A survey of intrusion detection in Internet of Things," Journal of Network and Computer Applications, vol. 84, pp. 25 – 37, 2017, doi: 10.1016/j.jnca.2017.02.009.