User Behavior Analytics for Insider Threat using Transformer Based Approach

(IJCSNT) International Journal of Communication Systems and Network Technologies
ISSN Number: 2053-6283
Volume 15, Issue No. 1, 33–43

https://journals.mirlabs.in/index.php/ijcsnt
DOI-10.18486/ijcsnt/15.1.002
IJCSNT

Aarati Kumari Mahato[1], Om Prakash Mahato[2], Roshan Pokhrel[3], Kashindra Mahato[4]

Abstract

In the past, attacks stemming from the deliberate or inadvertent actions of employees within the organization did not typically worry people. Due to the frequent reports of data breaches involving many organizations and their own employees, businesses are growing increasingly concerned about the need to keep an eye on user's behavior on the network. In light of this, this study suggests a method for user behavior analytics. It makes use of the CERT (version 4.2) insider threat dataset, a five-log file synthetic dataset developed for insider threat research. Various raw events are provided by these log files. The behavior analysis of every user in this work is carried out based on log sources. The structural behavior provided by CERT, is in structural format, which is converted into the contextual data by fine-tuning the Llama3.1 model. These contextual data along with the synthesized contextual data are used to train the SecureBERT model for classification purpose. The model train accuracy was observed to start from 67% when trained and test accuracy was obtained to be 76% with 86% precision on 2500 data samples. The model was then trained on over 16 thousand data, observing the accuracy started with 92% in first epoch and reached to nearly 99.99%, which shows large language models can perform better and accurate in contextual user behavior analysis in insider threat.

Keywords

Anomaly Detection, CERT, Context Generation, Insider Threat Classification, Large Language Model, Llama, SecureBERT, Transformer

Received: 12 November 2025; **Revised:** 15 March 2026; **Accepted:** 12 April 2026;
Published: 30 April 2026;

[1]Department of Electronics and Computer Engineering, Thapathali Campus, Institute of Engineering, Tribhuvan University, Kathmandu, Nepal.
[2]Nepal Telecommunications Authority, Kathmandu, Nepal.
[3]LogPoint Nepal Pvt. Ltd., Kathmandu, Nepal.
[4]Dhuni Software, Kathmandu, Nepal.

Corresponding author:
Email: aarati.079msise01@tcioe.edu.np

Introduction

As digital systems and the internet continue to grow, the world has to reckon with cyber threats that are multiplying and diversifying: malware, ransomware, phishing, and sophisticated, long-term threats that usually originate from external attackers. Yet, threats from within the organization, whether from employees, contractors, or trusted third parties are more difficult to police. These insiders, who already have the keys to the castle, can unwittingly leak sensitive files, sabotage systems, or open doors for external attackers, making the risk both subtle and severe. The 2023 Insider Threat Report makes the size of the issue clear: the overwhelming majority 72% of organizations have already seen an insider incident, and each year the average price tag from these all-too-frequent breaches comes to $15.4 million. Almost 60% of the firms the report surveyed admit that they can't consistently tell the difference between a harmless login pattern and one that signals a leak or a scam in the making. Conventional defenses such as firewalls and intrusion-detection systems are falling short at a moment when benign-looking behavior from employees can switch from routine to dangerous in milliseconds. Analyses in the report suggest that in the majority of cases the root cause comes from human behavior: 43% of interruptions are the result of simple, careless mistakes, another 25% stem from deliberately harmful actions. The blend of accidental and intentional threats, coupled with the shifting patterns seen in hybrid work and the sheer size of data in circulation, renders traditional approaches inadequate. New models that put machine learning and behavior analytics at the core of detection, risk-scoring, and automated risk response are no longer a luxury; they are a prerequisite for defending the organization's most sensitive assets from ill-formed clicks as well as craftily engineered betrayal from within. The initiative responds to the pressing requirement to unveil low-profile, context-sensitive insider actions like atypical email sending, irregular file browsing, or strange login series potential harbingers of fraud, sabotage, or data exfiltration. Present-day solutions excel against overt attacks yet overlook passive insider gestures, underscoring the need for persistent, longitudinal behavior visibility. Datasets harvested weekly from the CERT 4.2 corpus will be sanitized, Llama 3.1 will synthesize auxiliary contextual narratives, and SecureBERT will autonomously earmark patterns as standard or hostile. Our architecture will yield instantaneous, context-enriched anomaly alerts, rule-adaptive operational enforcement, zero-delay triggered countermeasures, and comprehensive retrospective examinations. The cumulative offering constitutes a resilient, behavior analytics as a service stack that fortifies enterprises against retrospective, stealthy insider compromises, data siphoning, and undisclosed actions, while simultaneously assuring comprehensive, ongoing posture refinement for displaced and distributed workers.

Literature Review

The researchers in paper[9], by focusing on just the top 1% of suspicious cases, the anomaly detection model achieved a maximum detection rate of 53.67%, based on daily activity summaries. For two out of the three tasks under review, over 90% of true anomalous behaviors were detected when the monitoring scope was widened to cover the top 30% of anomaly scores. Similarly, when the top 30% of suspicious emails were examined, the detection rate increased to 65.64%, with a peak of 98.67%, compared to a maximum of 37.56% for malicious emails detected using a 1% threshold in the email content analysis. While the model is supported by empirical evidence, there are still some limitations in the current research. There is a lack of proper integration between the various anomaly detection outcomes and this approach cannot detect malicious behavior in real time.

A 2018 evaluation of the insider threat found that 53% of attacks originated from within organizations in the preceding year. Furthermore, according to 27% of the companies surveyed, the attacks originated within. It was shown that the likelihood of insider threats is roughly 66% when there is network access and 27% when there is physical access. A taxonomy of modern insider types, access, level, motivation, insider profiling, effect security property, and techniques used by attackers to conduct attacks were presented, along with an overview of noteworthy recent works on insider threat detection that covered the analyzed behaviors, machine-learning techniques, dataset, detection methodology, and evaluation metrics. They also introduced CERT, NSL KDD OR KDD-99, APEX, RUU, and TWOS dataset.[3]

The authors[16] introduced a User and Entity Behavior Analytics (UEBA) system that employs predefined rules, anomaly detection, and machine learning techniques to detect insider threats. According to the 2019 Insider Threat Report, 68% of organizations regularly encounter insider attacks. The data used in UEBA is gathered from various sources, such as system and application logs, network devices, network traffic, net flows, and includes user activities like login/logout details, email usage, web searches, file operations, server access, and USB usage. This paper reviewed existing UEBA techniques, the design and features of leading UEBA solutions in the market, along with their strengths and weaknesses. Key challenges in UEBA development and deployment include training data duration, minimizing false positives, and calculating risk scores. When evaluating UEBA solutions, it's essential to consider threat detection capabilities alongside scalability and specific use cases. An advanced behavioral analytics system should also integrate data from multiple sources, including social media activities, emails, and network access logs.

Methodology

This study uses Meta AI's Llama 3.1 to translate the raw CERT 4.2 log data into contextual and subsequently passes these through SecureBERT to classify user behavior as normal or malicious. Llama, itself a transformer-based language model, renders raw log entries into human friendly formatted natural language regarding user activity. Pre-processing includes cleaning, normalizing and tokenizing the data, and then doing contextual embedding. The assistance of the post-processing is semantically correct texts distinguishing the behavior. SecureBERT fine-tuned for security tasks classify the generated text descriptions. The malicious text, not the original string, is tokenized and embedded, and classified in this way as either normal or malicious based on PCA of embeddings.

A. Large Language Model by Meta AI (Llama3.1)

The third version of Meta's Llama series, Llama, aims to improve natural language processing (NLP) capabilities. Although Meta's particular implementations would determine the precise architecture details for Llama 3, we can deduce broad concepts from earlier versions (such as Llama 2) and the most advanced transformer models. Llama models use a decoder-only architecture, focusing on generating text based on the input context.

B. SecureBERT

SecureBERT[1,2] is a specialized language model designed for cybersecurity applications, particularly effective in understanding and classifying cybersecurity-related texts, including those related to insider threats. It was developed by fine-tuning the RoBERTa base model with a substantial dataset comprising

98,411 cybersecurity-related textual elements (equivalent to about 1 billion tokens). This fine-tuning process involved introducing noise into the token weights during training to enhance robustness against adversarial attacks. It processes input sequences up to 512 tokens using 12 transformer layers with self-attention to capture deep contextual meaning. Its classification head, with a fully connected layer and SoftMax output, assigns class probabilities. With approximately 123M parameters, the model effectively learns complex text patterns, making it highly suitable for identifying and classifying insider threats[5].

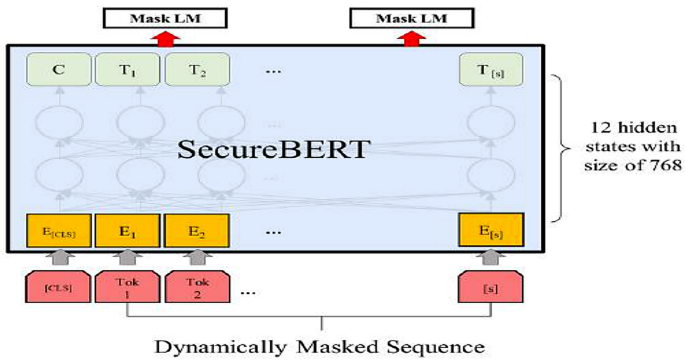

Figure 1. Architecture of SecureBERT (Source of Reference[3,4])

C. DATASET

The CERT Insider Threat Dataset, developed by the CERT Division at Carnegie Mellon University, is a widely used synthetic dataset for insider threat research, offering realistic, privacy free data for benchmarking and model development[6]. Among its versions, CERT 4.2 was chosen for this work as it contains data from 1000 users over 501 days, including 70 malicious actors and over 32 million possible activities with 7327 malicious actions[7]. It provides five log files: logon, file, device, email, and http, capturing diverse user activities essential for holistic analysis. Its advantages include comprehensive and realistic coverage of user behaviors, scenario-based malicious activities reflecting real-world cases, and thorough documentation, making it an ideal dataset for evaluating and improving insider threat detection models[8].

The dataset defines three key insider threat scenarios: **Scenario 1** involves an employee misusing removable media to upload confidential data to *wikileaks.org* before leaving the company[9]. **Scenario 2** shows a user seeking jobs with competitors and stealing data via frequent thumb drive use; and **Scenario 3** depicts a disgruntled system administrator installing a keylogger, impersonating his supervisor, sending a mass email, and resigning immediately after[10].

The LLAMA 3.1 model was fine-tuned on 2,400 GPT-4 Mini–generated samples with three columns: **Input (Logs)** containing raw user and system events, **Output (Contextual Data)** converting logs into natural language descriptions, and **Label** classifying them as normal or scenario-based threats. This process improves the model's ability to contextualize activities and distinguish between normal and malicious behavior for insider threat detection. The fine-tuned model generated 16,800 samples,

Table 1. Attributes of each log file

File Name	Fields
Logon.csv	Id, date, user, pc, activity(login/logoff)
File.csv	Id, date, user, pc, filename, content
Device.csv	Id, date, user, pc, activity(connect/disconnect)
http.csv	Id, date, user, pc, url, content
Psychometric.csv	Employee_name, user_id, O,C,E,A,N
Email.csv	Id, date, user, pc, to, cc, bcc, from, size, attachment_count, content

including 7,200 normal activities and 9,600 malicious ones across three scenarios (3,000 in Scenario 1, 3400 in Scenario 2 and 3200 in Scenario 3). This balanced dataset, with detailed labels and contextual descriptions, supports effective training and evaluation of models for distinguishing normal from insider threat behaviors[11–14].

Experiment Results

The Llama 3.1 model was fine-tuned for the contextual data generation task from the structured data. This fine tuning of the model was initiated by loading the 8 billion parameter version in 4 bit quantization. The model was configured with Parameter efficient Fine-Tuning using LoRA(Low-Rank Adaptation) to fine-tune the model by updating specific projection layers (q proj, k proj, etc.) with a low-rank adaptation (r=16) and scaling (lora alpha=16), no dropout (lora dropout=0), no bias tuning, and using gradient check pointing[15,16]. The model was fine-tuned with learning rate = 2e-4, using AdamW optimizer and weight decay = 0.01. The training loss obtained during the process can be seen in the figure below.

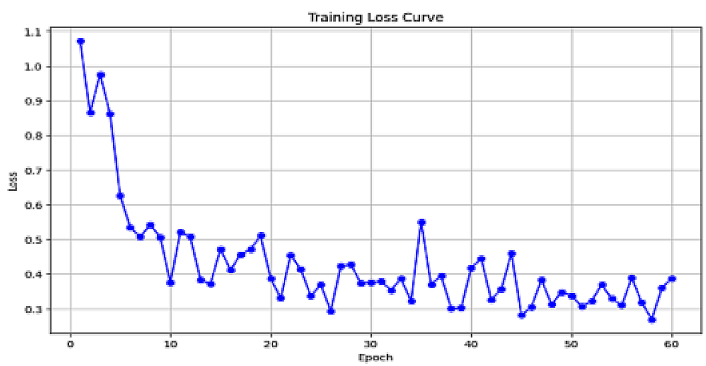

Figure 2. Training loss obtained while fine-tuning Llama3.1 model

The graph shows the training loss obtained when fine tuning the Llama 3.1 model over 60 epochs for a contextual data generation. As seen on the graph, the values of loss in epoch 0 are greater than 1, but they rapidly decrease to about 0.4 within the first ten epochs. These results suggest that the model is

converging though it has some discrepancies or noises during training. Since we observe a trend towards zero in this period with some fluctuations, we can say that model is converging despite some noise or changes during training. The trend suggests successful fine-tuning.

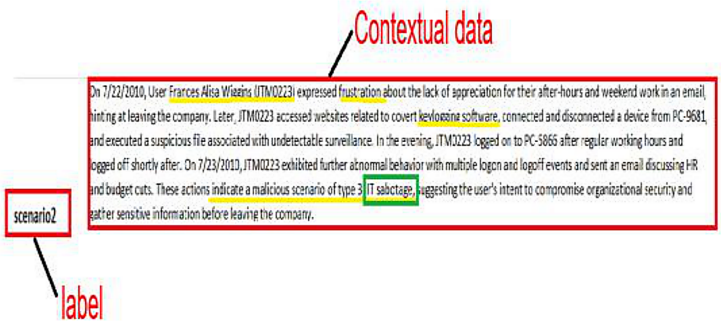

Figure 3. An example of generated contextual data

The SecureBERT model when trained by adding a classification head for multiclass classification, the samples size was divided into validation, test and train set. The model was trained for binary classification with a learning rate of 1×10^{-5}, a batch size of 8, and 25 epochs. A maximum sequence length of 512 tokens was used, along with a weight decay of 0.01 for regularization. Early stopping was applied with a patience of 3. The training and validation loss curve obtained for the above combination of hyperparameters can be seen in table2.

The figure4 depicts the training and validation loss curves over 25 epochs of a deep learning model. The x-axis represents the number of epochs, while the y-axis represents the loss values on a logarithmic scale. Two curves are plotted: the training loss (in blue) and the validation loss (in orange). Throughout the training process, both the training and validation losses decrease consistently, indicating that the model is learning effectively. The initial rapid decline in the loss values suggests that the model quickly captures significant patterns in the data. As the epochs progress, the rate of decrease slows down, indicating that the model is gradually converging towards its optimal performance. The relatively smooth and parallel behavior of the training and validation loss curves suggests that the model is generalizing well to unseen data, without significant overfitting. The validation loss does not exhibit a notable increase or divergence from the training loss, which is a positive indicator of the model's stability and robustness. This behavior reflects a successful training process, where the model maintains good generalization while minimizing the loss on both the training and validation sets.

Hyperparameter tuning using the GridSearchCV and was performed on the same model along with L2 regularization technique and AdamW optimizer. The hyperparameter grid used for GridSearchCV includes learning rates of [1e−5,5e−5,1e−4,2e−5], batch sizes of [8,16,32] and epochs of [3,4,5,10,20,25].

The model was trained using optimal hyperparameters selected via GridSearchCV. As shown in the loss curve, the training and validation losses decrease smoothly without signs of overfitting, indicating effective learning. The accuracy curves also show strong performance, with high and closely aligned

Table 2. Train results obtained after training SecureBERT for classification

Epoch	Train Loss	Train Acc	Val Loss	Val Acc	Prec.	Recall	F1-score
1/25	0.8456	0.6781	0.3083	0.9750	0.9423	0.9516	0.9406
2/25	0.2178	0.9474	0.0719	0.9958	0.9934	0.9919	0.9926
3/25	0.0487	0.9969	0.0054	1.0000	1.0000	1.0000	1.0000
4/25	0.0092	1.0000	0.0017	1.0000	1.0000	1.0000	1.0000
5/25	0.0034	1.0000	0.0009	1.0000	1.0000	1.0000	1.0000
6/25	0.0022	1.0000	0.0006	1.0000	1.0000	1.0000	1.0000
7/25	0.0014	1.0000	0.0003	1.0000	1.0000	1.0000	1.0000
8/25	0.0010	1.0000	0.0003	1.0000	1.0000	1.0000	1.0000
9/25	0.0008	1.0000	0.0003	1.0000	1.0000	1.0000	1.0000

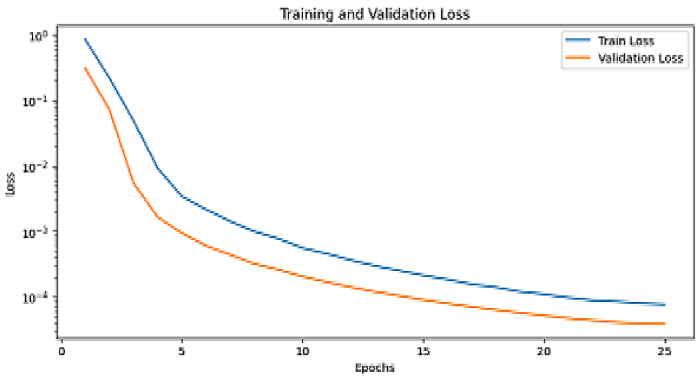

Figure 4. Training and validation loss curve for classification

training and validation accuracy achieved within few epochs. This reflects the model's efficiency, stability, and ability to generalize well, making it suitable for practical classification tasks with reliable and consistent predictions.

Testing on different sets of DATA

Model was evaluated on a separate test set after splitting the data into training, validation, and testing sets. The confusion matrix shows perfect classification across all four classes (0–3), with no misclassifications. All instances, 152 in Class 0, 20 in Class 1, 31 in Class 2, and 37 in Class 3 were correctly predicted. The absence of errors highlights the model's exceptional accuracy and strong ability to distinguish between classes. The trained model was tested on 100 newly generated unseen data, not used during training or validation. The confusion matrix shows strong performance, with all 39 instances of "scenario 1" and all 20 of "scenario 3" correctly classified. For "scenario 2," 18 were correctly identified, while 3 were misclassified. Model also correctly detected 14 out of 20 "normal" cases, with 6 misclassified as "scenario 3." Despite some confusion between "scenario 2" and "scenario 3," and between "normal" and "scenario 3," the model demonstrates high overall accuracy, particularly in classifying "scenario 1" & "scenario 3".

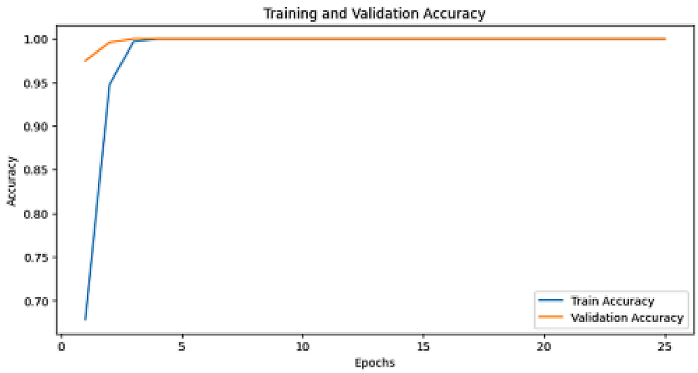

Figure 5. Training vs Validation Accuracy curve after using GridSearchCV

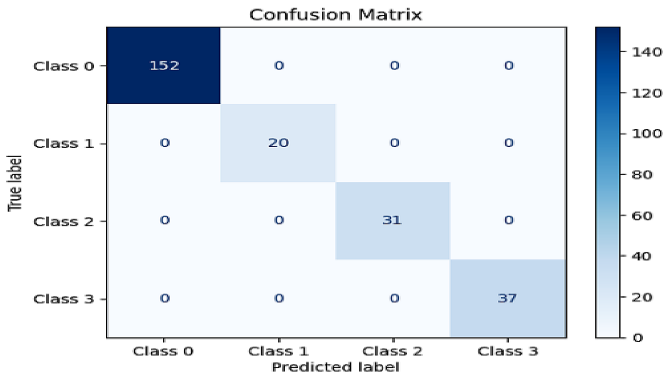

Figure 6. Obtained Confusion matrix when model tested on testing set data

The model was tested on 1,200 unseen samples and showed excellent performance. It correctly classified all instances of scenario 1, 2, and 3. For the normal class, 285 out of 300 were correct, with 15 misclassified as scenario 3. This reflects high overall accuracy with minimal errors.

The ROC curve shows that the model performs exceptionally well in classifying all classes, with AUC values near 1. This indicates excellent discrimination and minimal error. A slight deviation for Class 2 (AUC = 0.999937) suggests a negligible error rate.

Discussion and Analysis

Theoretically, increasing dataset size and training epochs enhances the performance of both the context generation model (Llama 3.1) and the classification model (SecureBERT) by enabling deeper learning and improved generalization, supported by techniques such as early stopping and learning rate reduction. Experimentally, initial fine-tuning of Llama 3.1 on 240 synthetic samples yielded unsatisfactory results. However, with parameter-efficient fine-tuning using Low-rank adaptation and flash attention on an

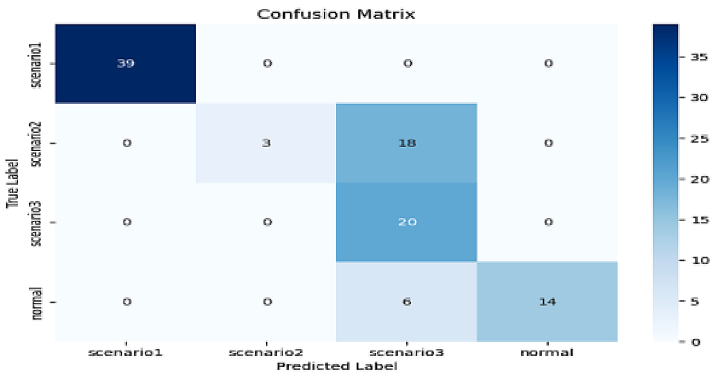

Figure 7. Obtained confusion matrix on 100 unseen test data

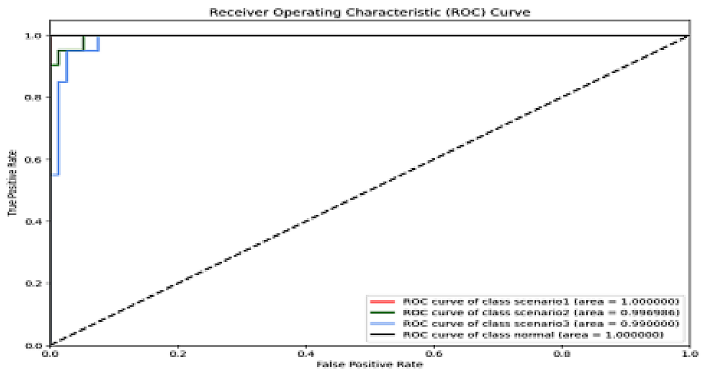

Figure 8. ROC curve for 100 unseen test data

expanded dataset of 1,400 samples over 60 epochs, the model generated high-quality contextual data tailored for insider threat detection, respecting SecureBERT's input token limits. Error sources included data quality issues (noisy and insufficient samples), suboptimal model size and hyperparameters (learning rate, batch size), and challenges in contextual understanding (loss of nuance, inadequate tokenization). These were addressed by increasing sample size, refining data generation prompts, employing advanced tokenization techniques (Grouped Query Attention and WordPiece), quantization, and hyperparameter optimization via grid search.

The best model performance was observed with a learning rate of 1e-5, batch size 16, and 25 epochs, showing steadily decreasing training and validation loss gaps, indicating effective learning and generalization. Conversely, training with 40 epochs, learning rate 5e-6, and batch size 8 led to overfitting after epoch 14. In another experiment, SecureBERT trained on 2,400 diverse samples for 5 epochs achieved 76% accuracy, demonstrating effective learning but room for improvement. Training on a larger dataset of 16,000+ samples with a learning rate of 2e-5 over 25 epochs quickly reached 100% training

accuracy by epoch 3; however, early stopping at epoch 7 prevented overfitting, highlighting the need to balance rapid learning and generalization. Overall, these results emphasize the critical role of dataset size, hyperparameter tuning, and early stopping in achieving robust model performance. Future work should explore further regularization and adaptive learning rate strategies to enhance generalization.

Future Enhancement

Future enhancements focus on evaluating Llama 3.1 and SecureBERT on broader and real-world datasets from diverse domains such as healthcare, finance, and telecommunications, with k-fold cross-validation to assess generalizability and robustness. Optimization of Llama 3.1 will involve expanding training data with complex linguistic structures and domain-specific terms, integrating Reinforcement Learning from Human Feedback (RLHF) for more refined outputs, and improving interpretability for incomplete or ambiguous inputs. Finally, real-time testing and deployment in operational environments like SOCs will validate speed, accuracy, and reliability under practical conditions, while addressing challenges such as latency, data variability, and inference efficiency to ensure readiness for mission-critical applications like intrusion detection and automated threat response.

Conclusion

This study successfully demonstrated how large language models like Llama 3.1 can be fine-tuned to convert structured cybersecurity data into clear, contextual natural language, improving the interpretability of complex security events. Additionally, the SecureBERT model was optimized to classify normal and multiple malicious insider threat scenarios with high accuracy (99–100%), outperforming traditional deep learning approaches. These results highlight the effective integration of NLP and machine learning techniques for enhancing cybersecurity analysis. While the models showed strong performance and scalability, further optimization is needed to improve real-time applicability and handle larger, more diverse datasets efficiently. The combination of natural language generation and robust classification offers a promising pathway to automate security operations, reduce response times, and improve threat detection in dynamic environments. Future work should focus on deploying these models in real-world settings and refining hyperparameters to maximize generalization and operational effectiveness.

References

[1] Ehsan Aghaei, Ehab Al-Shaer, Waseem Ghassan Tahseen Shadid, and Xi Niu. Automated cve analysis for threat prioritization and impact prediction. ArXiv, abs/2309.03040, 2023.

[2] Ehsan Aghaei, Xi Niu, Waseem Shadid, and Ehab Al-Shaer. Securebert: A domainspecific language model for cybersecurity. In International Conference on Security and Privacy in Communication Systems, pages 39–56. Springer, 2022.

[3] Mohammed Nasser Al-Mhiqani, Rabiah Ahmad, Z. Zainal Abidin, Warusia Yassin, Aslinda Hassan, Karrar Hameed Abdulkareem, Nabeel Salih Ali, and Zahri Yunos. A review of insider threat detection: Classification, machine learning techniques, datasets, open challenges, and recommendations. Applied Sciences,10(15), 2020.

[4] B. Lindauer, "Insider Threat Test Dataset." Carnegie Mellon University, 2020. doi: 10.1184/R1/12841247.

[5] Koustav Dutta, Rasmita Lenka, Priya Gupta, Aarti Goel, Janjhyam Venkata, and Janjhyam Ramesh. Swift diagnose: A high-performance shallow convolutional neural network for rapid and reliable sars-cov-2 induced pneumonia detection. EAI Endorsed Transactions on Pervasive Health and Technology, 10, 03 2024.

[6] Fredrik Heiding, Bruce Schneier, Arun Vishwanath, Jeremy Bernstein, and Peter S. Park. Devising and detecting phishing: Large language models vs. smaller human models, 2023.

[7] Fredrik Heiding, Bruce Schneier, Arun Vishwanath, Jeremy Bernstein, and Peter S Park. Devising and detecting phishing emails using large language models. IEEE Access, 2024.

[8] Shadi Jaradat, Richi Nayak, Alexander Paz, Huthaifa I. Ashqar, and Mohammad Elhenawy. Multitask learning for crash analysis: A fine-tuned llm framework using twitter data. Smart Cities, 7(5):2422–2465, 2024.

[9] Dahye Kim, Dongju Park, Honghyun Cho, and Kang. Insider threat detection based on user behavior modeling and anomaly detection algorithms. Applied Sciences,9:4018, 09 2019.

[10] Xuemei Li and Huirong Fu. Securebert and llama 2 empowered control area network intrusion detection and classification. arXiv preprint arXiv:2311.12074, 2023.

[11] Mario Pérez-Gomariz, Fernando Cerdán-Cartagena, and Jess García. Lm-hunter: An nlp-powered graph method for detecting adversary lateral movements in apt cyber-attacks at scale. Available at SSRN 4807938.

[12] Abir Rahali and Moulay A. Akhloufi. Malbertv2: Code aware bert-based model for malware identification. Big Data and Cognitive Computing, 7(2), 2023.

[13] Abir Rahali and Moulay A Akhloufi. Malbertv2: Code aware bert-based model for malware identification. Big Data and Cognitive Computing, 7(2):60, 2023.

[14] Madhu Raut, Sunita Dhavale, Amarjit Singh, and Atul Mehra. Insider threat detection using deep learning: A review. In 2020 3rd International Conference on Intelligent Sustainable Systems (ICISS), pages 856–863, 2020.

[15] Madhu Raut, Sunita Dhavale, Amarjit Singh, and Atul Mehra. Insider threat detection using deep learning: A review. In 2020 3rd international conference on intelligent sustainable systems (ICISS), pages 856–863. IEEE, 2020.

[16] Balaram Sharma, Prabhat Pokharel, and Basanta Joshi. User behavior analytics for anomaly detection using lstm autoencoder-insider threat detection. In Proceedings of the 11th international conference on advances in information technology, pages 1–9, 2020.