GRAPH-BASED USER BEHAVIOR ANALYSIS APPROACH FOR ANOMALY DETECTION ON E-LEARNING PLATFORMS

ER-ROUGUI Saad , ENNAGACH Ayoub , ENNAJAH Ayoub

ABSTRACT

Massive Open Online Courses (MOOCs)[1] have democratized education by providing open access to learning resources. However, ensuring the authenticity of user engagement and detecting anomalous behaviors, such as account sharing, automated interactions, or fraudulent certification, remains a significant challenge. A deep understanding of user behavior is crucial to identifying deviations that may indicate misconduct or misuse of online learning platforms.

This study introduces a graph-based approach to model and analyze user behavior on e-learning platforms using the MOOCCube dataset. We construct a User-Course Interaction Graph, where learners, courses, sessions, and actions are represented as interconnected nodes, with edges capturing behavioral dynamics such as session duration, action frequency, and learning consistency. This structured representation allows us to detect anomalies, such as users exhibiting sudden shifts in learning patterns, proxies being used for exams, or certificates being earned through unauthorized means. By applying graph analytics and anomaly detection techniques, our approach enhances security, ensures the integrity of learning outcomes, and provides a scalable solution for monitoring user behavior in digital education environments.

1 INTRODUCTION

Massive Open Online Courses (MOOCs) have made education widely accessible, but ensuring the authenticity of user engagement remains a challenge. Fraudulent behaviors such as account sharing, automated interactions, and false certification compromise learning integrity. Traditional rule-based detection methods struggle to capture complex behavioral patterns in large-scale platforms.

To address these issues, we introduce a graph-based framework that models each user's learning sessions, course enrollments, and actions as a personalized graph. This structure allows us to analyze behavioral sequences and detect anomalies, such as sudden shifts in learning behavior or irregular activity timestamps.

Graph Neural Networks (GNNs) serve as the core technique for anomaly detection in these user graphs. By leveraging message passing and node embeddings, GNNs capture temporal relationships between sessions and actions, allowing for the identification of fraudulent behavior. Additional graph mining techniques, such as centrality analysis and sequence-based pattern detection, further enhance anomaly detection, making this a scalable solution for monitoring user behavior on e-learning platforms.

For more details and access to the implementation, please visit our GitHub repository: `https://github.com/ER-ROUGUI/Graph_Analytics.git`

[1]MOOCs are online courses designed for large-scale participation, providing open access to educational content, often from universities and institutions, via digital platforms.

2 PROJECT DESCRIPTION

Our project focuses on leveraging network analysis to study user behavior and detect anomalies in online learning platforms. This work aligns with real-world research applications, particularly in cybersecurity, where anomaly detection is critical for identifying threats. Our team has been actively working on a similar project in cybersecurity, focusing on user/host anomaly detection using a tabular dataset. This experience has motivated us to explore graph-based modeling as a more structured approach to analyzing behavioral deviations in large scale systems.

In this study, we model user interactions in MOOCs using graph-based techniques, constructing an interaction network that captures user sessions, course enrollments, and learning activities. Furthermore, we conduct an experimental evaluation of anomaly detection techniques using Graph Neural Networks (GNNs), assessing their effectiveness in identifying abnormal behavior patterns. This approach allows us to explore how structured network modeling enhances anomaly detection, transitioning from traditional tabular approaches to graph-based anomaly detection in large-scale e-learning environments.

By combining theoretical modeling with empirical analysis, our study provides valuable insights into online education security, user authentication, and behavioral anomaly detection, with potential applications beyond MOOCs, including fraud detection, identity verification, and cybersecurity threat analysis.

3 PROBLEM DEFINITION

Understanding engagement dynamics in MOOCs requires a structured approach to modeling user interactions. Traditional methods of analyzing tabular data fail to capture the sequential and interconnected nature of student behaviors. To address this, we formulate the problem as a set of **individual user graphs**, where each user's sessions, actions, and course interactions are represented as nodes, and their relationships are defined as edges. This representation enables us to track engagement levels, detect anomalies, and predict variations in user activity over time.

We model each user's behavior in MOOCs as an individual directed graph:

$$G_u = (V_u, E_u, X_u, W_u)$$

where:

- $V_u = S_u \cup A_u$ is the set of nodes in user u's graph, categorized as:
  - S_u (Sessions): Time-bounded learning interactions.
  - A_u (Actions): User-generated interactions (e.g., clicks, video plays, quiz attempts).
- E_u is the set of directed edges representing transitions between sessions and actions.
- X_u represents node features, such as session duration, time-of-day activity, and interaction type.
- W_u represents edge weights, encoding time gaps between actions, frequency of interactions, and session continuity.

Each edge $(v_i, v_j) \in E_u$ is weighted based on user interaction metrics:

$$w_{ij} = f(v_i, v_j)$$

where $f(\cdot)$ represents:

- Session duration for **Session** $\rightarrow$ **Action** edges.
- Number of actions performed within a session.
- Time gaps for **Session** $\rightarrow$ **Session** transitions.
- Timestamp differences for **Action** $\rightarrow$ **Action** edges.

The goal of this representation is to analyze individual user behavior, detect anomalies, and identify fraudulent activities in online learning environments. By modeling interactions at a personal level, we capture unique learning patterns and deviations from expected behavior. The per-user graph approach allows us to detect anomalies such as irregular session sequences, abnormally high engagement spikes, and inconsistent activity timestamps that may indicate account sharing or automated cheating behaviors. This structured approach provides a scalable and adaptive framework for monitoring learning integrity, enhancing security, and ensuring the authenticity of online education.

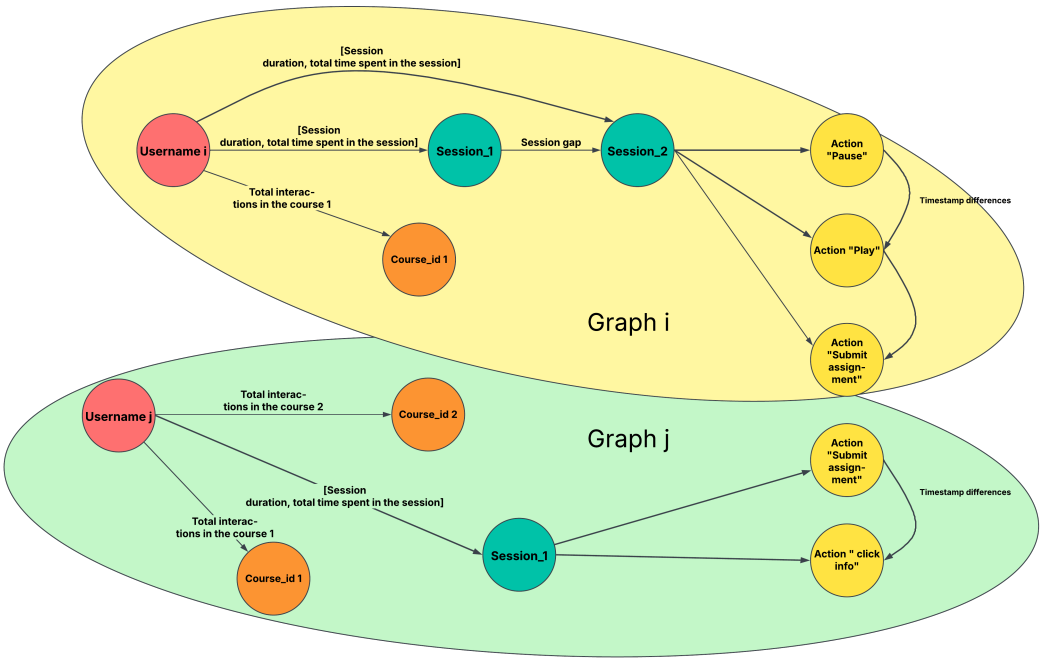

Figure 1: Graph Modeling

This representation effectively models the hierarchical structure of user behavior in MOOCs, capturing the complexities of online learning interactions. As illustrated in Figure 3, our graph consists of four main node types: Users, Sessions, Courses, and Actions. Users interact with multiple courses, initiating learning sessions that include various activities such as video interactions, assignment submissions, and content navigation. The directed edges define relationships between these entities, preserving sequential action flows, session transitions, and enrollment patterns.

The weighted edges session duration, action frequency, and session gap add temporal and behavioral context, allowing for a detailed analysis of user activity consistency and anomalies. This structure enables the detection of irregular engagement behaviors, fraudulent course completion, and potential misuse of accounts (e.g., account sharing). The layered approach ensures scalability and adaptability, making it a powerful framework for fraud detection, behavioral analysis, and security monitoring in online learning environments.

One of the primary use cases of this graph-based modeling is detecting anomalous user behaviors that may indicate account sharing, fraudulent activity, or academic dishonesty. Each user typically follows a consistent learning pattern based on their engagement habits, study routine, and course preferences. For instance, a user who regularly engages with computer science courses may exhibit a stable pattern in terms of session duration, action frequency, and time spent per session. If a sudden deviation from this pattern occurs such as an abrupt increase in session frequency, engagement with courses outside their usual interests, or unusually short session durations this could indicate that another individual is using the account. Such anomalies could point to cases where students share credentials to complete coursework, artificially inflate participation metrics, or obtain certificates fraudulently.

Moreover, this framework enables the detection of users engaging in outsourced coursework completion. For example, if a student's activity exhibits consistent engagement trends but suddenly shifts to erratic session timing, differing engagement levels, or unfamiliar action sequences, it may suggest that another person has taken over their learning process. This is particularly relevant in MOOCs, where certifications hold professional value, and individuals may attempt to game the system by hiring others to complete assignments, exams, or entire courses on their behalf.

By leveraging this structured graph-based approach, MOOC platforms can strengthen security measures, detect fraudulent activity, and ensure the integrity of online learning assessments. This methodology provides an effective way to monitor user authenticity, detect behavioral irregularities, and enforce academic integrity in digital education environments.

4 Related Work

Graph-based modeling has been widely used in various domains, including recommendation systems, fraud detection, and social network analysis. In the context of MOOCs, researchers have explored multiple approaches to analyzing user interactions, engagement, and behavioral patterns.

Several prior works have explored user engagement and anomaly detection using different machine learning and statistical techniques.

User Behavior Analytics for Anomaly Detection Using LSTM Autoencoder Pokharel & Joshi (2020): This study employs Long Short-Term Memory (LSTM) autoencoders to detect anomalous behaviors in insider threat detection. The model learns user behavior patterns over time and identifies deviations indicative of security threats. While primarily used in cybersecurity, this approach inspires behavioral anomaly detection in MOOCs by identifying unusual behavior patterns.

User Navigation Pattern Discovery and Analysis for Web Usage Mining Nasraoui et al. (2000): This work focuses on extracting navigation patterns from web interactions using sequence mining techniques. By analyzing user clickstreams, the study uncovers behavioral trends and potential anomalies in web navigation. Similarly, in MOOCs, such techniques can help detect deviations from typical learning pathways.

Identifying User Behavior in Online Social Networks Maia et al. (2008) is a study that aims to systematically characterize and identify different user behaviors in online platforms. The authors analyze data from YouTube's subscription network to uncover interaction patterns. By leveraging clustering techniques on user engagement data, they differentiate between user types such as content creators, passive consumers, and highly interactive users. The study highlights that attributes derived from social interactions are more effective in characterizing user behavior than individual attributes. This work is relevant to our study as it emphasizes the importance of engagement patterns in identifying behavioral profiles, which aligns with our goal of detecting anomalies in MOOC engagement.

5 Methodology

5.1 Data Description and Preprocessing

The dataset used in this study is derived from the MOOCCube dataset Wang et al. (2021). MOOCCube is a large-scale dataset designed to facilitate research in online learning and contains rich information on student-course interactions, engagement behaviors, and learning outcomes.

5.1.1 Dropout Prediction Dataset

This dataset contains logs of user activities within courses and includes:

Table 1: Columns in the Dropout Prediction Dataset

Column	Description
enroll_id	Unique identifier for user-course pair
username	Unique user identifier
course_id	Unique course identifier
session_id	Identifier for the session
action	Type of user activity
object	Object associated with the action
time	Timestamp of the action

5.1.2 User Profile Dataset

Contains demographic information about users:

Table 2: Columns in the User Profile Dataset

Column	Description
user_id	Unique user identifier
gender	Gender of the user
education	Education level of the user
birth	Birth year of the user

5.1.3 Course Information Dataset

Describes the courses and their attributes:

Table 3: Columns in the Course Information Dataset

Column	Description
id	Numeric course identifier
course_id	String course identifier
start	Course start date
end	Course end date
course_type	Course mode (0: instructor-paced, 1: self-paced)
category	Course category

5.1.4 Dataset Preprocessing

The raw dataset contains various attributes, some of which are not relevant to our study. Since our approach focuses on **anomaly detection in user behaviour patterns**, certain attributes, such as user demographic details (gender, birth year, and education), are not beneficial. These features may be useful in tasks like community detection or course recommendation but do not contribute to our primary objective. Additionally, the dropout labels were not used since we are not performing dropout prediction.

To ensure consistency and reliability in our dataset, we performed the following preprocessing steps:

1. **Data Cleaning:** Removed unnecessary attributes, handled missing values, and ensured the integrity of session-based interactions.
2. **Feature Selection:** Retained only essential attributes related to user activity, session behaviors, and course interactions that align with our graph-based modeling.

3. **Merging Datasets:** Integrated the three datasets (Dropout Prediction, User Profile, and Course Information) into a single dataset that encapsulates meaningful learning interactions.

5.1.5 DATASET FEATURIZATION

To better analyze user behavior, we engineered additional features that help quantify engagement dynamics and learning patterns:

- **session_duration**: Captures the total time a user spends within a session, allowing us to measure engagement levels.
- **session_gap**: Computes the time difference between consecutive sessions to analyze continuity in learning activity.
- **action_count**: Represents the number of interactions within a session, providing insight into session intensity.
- **action_frequency**: Measures the rate of user interactions per second within a session, aiding in detecting engagement variations.

These additional features allow us to track behavioral changes in user interactions and detect anomalies such as unusual session lengths or sudden drops in engagement. For example, an unusually long session gap might indicate reduced engagement, while a high action frequency in a short session could suggest automated activity.

The final dataset consists of the following attributes:

- **username:** Unique identifier for each user.
- **enroll_id:** Unique enrollment identifier per user-course pair.
- **session_id:** Unique identifier for each session.
- **course_id:** Unique course identifier.
- **action:** Type of action performed (e.g., play video, seek video, click_info, submit assignment).
- **time:** Timestamp of the action.
- **category:** Course category (e.g., Science, Business, Engineering).
- **session_duration:** Total time spent in a session.
- **session_gap:** Time difference between consecutive sessions of a user.
- **action_count:** Total number of actions performed in a session.
- **action_frequency:** Number of actions performed per second in a session.

This preprocessing step ensures that the dataset is clean, well structured, and suitable for graph-based analysis.

Now that we have prepared our dataset, we proceed to formally define our problem and outline the methodology used in the following section.

5.2 GRAPH MODELING AND APPROACH

To construct the per-user graph, we define nodes representing sessions and actions. Each user has an independent graph where sessions capture time bounded learning activities, and actions represent engagement events such as video plays, quiz attempts, and content navigation. Edges encode temporal and behavioral relationships, including session transitions, action sequences, and engagement intensity. Edge weights incorporate attributes such as session duration and action frequency, enriching the graph structure for anomaly detection. Unlike prior work that relies solely on tabular or sequential modeling, our approach leverages graph-based representations to capture the unique learning patterns of individual users. By structuring each user's learning activities as an independent graph, we can model engagement holistically and apply graph analytics techniques such as *node*

centrality, *sequence pattern detection*, and *Graph Neural Networks (GNNs)* for behavioral analysis. This per-user modeling enhances the ability to detect deviations in learning behavior, helping identify fraudulent activities such as account sharing and automated engagement.

5.3 GRAPH NEURAL NETWORK MODEL

5.3.1 NODE EMBEDDINGS

Each node $v \in V$ has an embedding initialized as:

$$h_v^{(0)} = x_v \quad \text{for courses, else } h_v^{(0)} \sim \mathcal{N}(0, I)$$

where:

- x_v represents the *category feature* for courses.
- Other nodes are initialized with *learnable embeddings*.

5.3.2 MESSAGE PASSING

At each layer l, node embeddings are updated using heterogeneous aggregation:

$$h_v^{(l+1)} = \sigma\left(W_v^{(l)} h_v^{(l)} + \sum_{(u,v)\in E} \frac{1}{d_v} W_{uv}^{(l)} h_u^{(l)}\right)$$

where:

- $W_v^{(l)}$ and $W_{uv}^{(l)}$ are learnable transformation matrices.
- d_v is the degree of node v.
- σ is an activation function (e.g., ReLU).

We employ *Heterogeneous Graph Transformer (HGT)* layers, allowing the model to adapt weights for each edge type.

5.4 APPROACH 1: SESSION-BASED MODELING

In this approch, we construct session pairs with a user identifier and a label that is set to 1 if both sessions belong to the same user and 0 otherwise. The goal is to train the model to differentiate session graphs for each user and, ultimately, to determine whether a future session genuinely belongs to the same user by analyzing its interaction graph. Each user's session is modeled as a directed graph $G_s = (V, E)$ where:

- V: set of actions with attributes $[count, duration]$
- E: temporal transitions with attributes $[frequency, \Delta t]$

5.4.1 TRAINING PAIR CONSTRUCTION

For n sessions of a user:

- Positive pairs: $\{(s_i, s_j) | i, j \in [1, n], i \neq j\}$
- Negative pairs: (s_i, s_k) where s_k comes from another user
- Train/test split: $80\%/20\%$ stratified per user

5.4.2 Model Architecture

5.4.3 Graph Encoder

$$h_v^{(l+1)} = \sigma\left(\sum_{u \in \mathcal{N}(v)} \frac{1}{\sqrt{d_v d_u}} W^{(l)} h_u^{(l)}\right) \tag{1}$$

where $h_v^{(l)}$ represents the features of node v at layer l.

5.4.4 Classifier

$$\hat{y} = \sigma(MLP(h_G)) \tag{2}$$

where h_G is the graph embedding obtained via mean pooling.

5.4.5 Loss Function

Weighted BCE to handle class imbalance:

$$\mathcal{L} = -\sum_{i=1}^{N} w_i (y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)) \tag{3}$$

where $w_i = \frac{N_{maj}}{N_{min}}$ for the minority class.

5.5 Model Performance Analysis

The model exhibits a strong tendency to predict the majority class (class 1), as indicated by the confusion matrix, where most predictions belong to this class. A significant number of false positives and false negatives are observed, impacting the model's precision and F1-score.

The final metrics show relatively low precision, meaning that many positive predictions are actually incorrect. On the other hand, recall is high, indicating that the model captures a large portion of the positive instances, although this is primarily due to class imbalance.

To improve overall performance, it might be beneficial to explore strategies such as class balancing, using a weighted loss function, or optimizing decision thresholds.

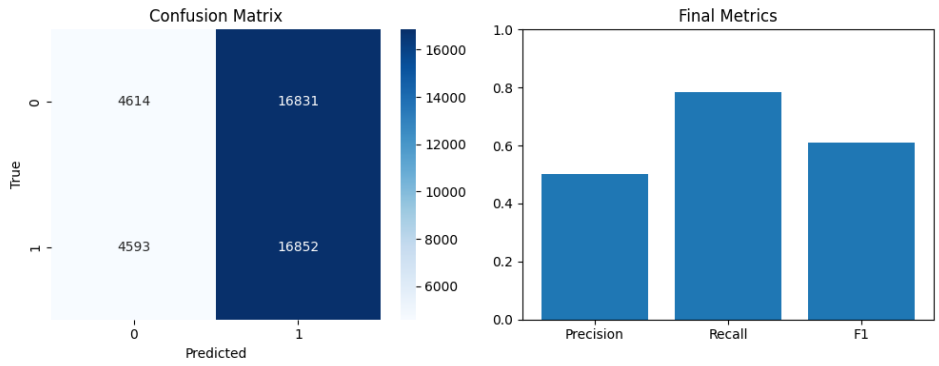

Figure 2: Confusion Matrix and Final Metrics of the Model

5.6 Approch 2 : User based modeling

5.6.1 Model Autoencoder

For each of the top 20 users, we construct individual user-session interaction graphs where nodes represent sessions, actions, and courses, and edges capture the transitions between them. Each node

is assigned a feature vector containing session duration, action frequency, and category information. These graphs are then encoded into a lower-dimensional latent space using a Graph Autoencoder (GAE).

The encoder takes as input:

- $X \in \mathbb{R}^{N \times d}$: Node feature matrix, where N is the number of nodes and d is the feature dimension.
- $A \in \mathbb{R}^{N \times N}$: Adjacency matrix, encoding the graph structure.

It applies Graph Convolutional Networks (GCN) to extract latent embeddings:

$$Z = \sigma(AXW_e + b_e) \tag{4}$$

where W_e is a learnable weight matrix, b_e is a bias term, and $\sigma(\cdot)$ is a non-linear activation function (e.g., ReLU). This produces **compressed embeddings** $Z \in \mathbb{R}^{N \times k}$, capturing essential structural and feature information.

The decoder reconstructs the original node features from Z:

$$X' = \sigma(ZW_d + b_d) \tag{5}$$

where W_d is the decoder's weight matrix. The Mean Squared Error (MSE) loss is minimized to ensure that $X' \approx X$, effectively learning meaningful node representations.

The dataset is split per user into 80% training and 20% testing. The autoencoder successfully learns **low-dimensional embeddings for user interactions, which can be further leveraged for anomaly detection, recommendation, or clustering.

5.6.2 Loss Function: Mean Squared Error (MSE)

To ensure that the reconstructed node features X' are as close as possible to the original features X, we use the Mean Squared Error (MSE) loss function, defined as:

$$\mathcal{L}_{MSE} = \frac{1}{N} \sum_{i=1}^{N} \|X_i - X_i'\|^2 \tag{6}$$

where:

- X_i is the original node feature vector of node i.
- X_i' is the reconstructed feature vector.
- N is the total number of nodes.

This loss function is minimized during training to ensure that $X' \approx X$, leading to meaningful node embeddings.

5.6.3 Training and Results

The dataset is split per user into 80% training and 20% testing. The autoencoder successfully learns low-dimensional embeddings for user interactions, as shown in the training loss curve, which demonstrates steady convergence over 50 epochs. These embeddings can be further leveraged for anomaly detection, recommendation, or clustering.

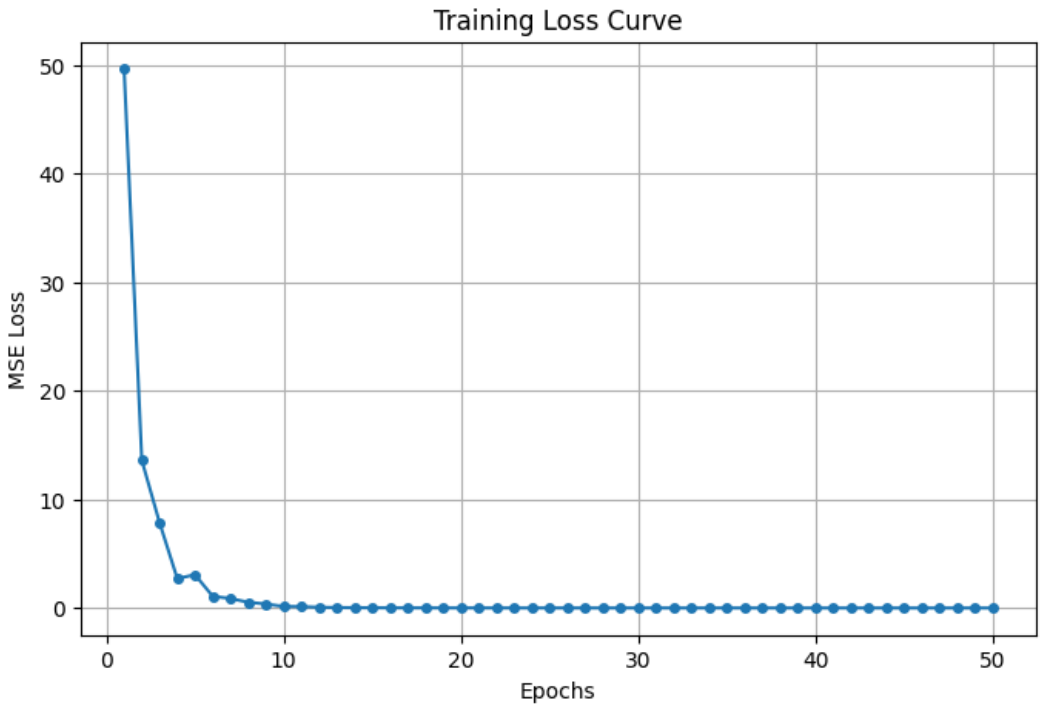

Figure 3: Training loss (MSE) over 50 epochs

Conclusion and Future Work

In this study, we explored two approaches for anomaly detection in user behavior on e-learning platforms. The first approach, session-based modeling, utilized Graph Neural Networks (GNNs) to analyze session graphs and detect inconsistencies in user interactions. The second approach, user-based modeling, employed an autoencoder to learn user-specific behavioral patterns and identify deviations that could indicate anomalies.

Although our methodologies showed promising results, we were limited by computational resources, preventing us from training our models on the entire dataset. This constraint restricted our ability to fully exploit the richness of the data and refine the models' generalization capabilities.

As future work, we aim to explore alternative graph-based representations for this dataset, investigating more sophisticated modeling techniques. Additionally, we plan to apply deeper and more robust models that can capture complex user interactions more effectively. Enhancing computational capabilities will also be a priority, allowing us to leverage the full dataset and further improve anomaly detection performance.

For more details and access to the implementation, please visit our GitHub repository: `https://github.com/ER-ROUGUI/Graph_Analytics.git`

REFERENCES

Marcelo Maia, Jussara Almeida, and Virgílio Almeida. Identifying user behavior in online social networks. In *Proceedings of the First Workshop on Online Social Networks (SocialNets'08)*, pp. 1–6, Glasgow, Scotland, UK, 2008. ACM. doi: 10.1145/1435497.1435498.

Olfa Nasraoui, Raghu Krishnapuram, and Anupam Joshi. Web usage mining: Discovery of the users' navigational patterns using som. In *Proceedings of the 9th International Conference on World Wide Web*, pp. 1–11. ACM, 2000. doi: 10.1145/336597.336668.

Prabhat Pokharel and Basanta Joshi. User behavior analytics for anomaly detection using lstm autoencoder - insider threat detection. *CERT Insider Threat Dataset Study*, 2020.

Jingyuan Wang, Weihao Li, Zhenya Huang, Zhiyuan Liu, and Jie Tang. Mooccube: A large-scale data repository for recommender systems in online education. *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI)*, 2021. URL `http://moocdata.cn/data/user-activity`.