European Economic Letters
ISSN 2323-5233
Vol 14, Issue 3 (2024)
http://eelet.org.uk

Investigating the Effectiveness of Graph-Based Algorithms in Identifying Insider Threats and Collusion: A Machine Learning Approach for Anomaly Detection in HR Networks

Mr. Debabrata Sahoo[1*], Dr. Smaraki Pattanayak[2], Dr. Phalgu Niranjana[3]
[1*]Doctoral Research Scholar, School of Business, ASBM University, Bhubaneswar, Odisha, India. E-mail ID: debabrata.sahoo1612@gmail.com
[2]Associate Professor, School of Business, ASBM University, Bhubaneswar, Odisha, India. E-mail ID: smaraki.pattanayak@asbm.ac.in
[3]Professor, School of Business, ASBM University, Bhubaneswar, Odisha, India. E-mail ID: phalgu.niranjana@asbm.ac.in
***Corresponding author:** Mr. Debabrata Sahoo
*E-mail ID: debabrata.sahoo1612@gmail.com

ABSTRACT
Data security and organizational integrity are seriously threatened by insider threats and collusion in the rapidly changing cybersecurity landscape. Because they are unable to analyze complex and dynamic human resource (HR) networks, traditional approaches of detecting such risks frequently prove to be inadequate. In order to discover anomalies in HR networks, this study examines how well graph-based algorithms detect insider threats and collusion using a machine learning technique. Using organizational data, the study first builds HR networks, which depict workers and their connections as nodes and edges within a graph structure. These networks are analyzed using a variety of graph-based methods, such as Graph Neural Networks (GNNs) specifically Local Outlier Factor (LOF), and community discovery techniques specifically Centrality Measures. The algorithms' goal is to find trends and abnormalities that point to possible collusion and insider threats.

This study makes use of a large dataset that include communication logs, HR records, and access patterns, giving analysts a realistic and varied collection of interactions to examine. In order to improve the detection accuracy of insider threats and collusion, the research makes use of both Graph Neural Networks (GNNs) and Centrality Measures. The efficacy of the suggested approaches is assessed and interpreted.

According to preliminary findings, graph-based algorithms considerably outperform conventional anomaly detection methods in spotting intricate and subtle collusion patterns. Specifically, GNNs show strong skills in figuring out complex connections and relationships inside HR networks, which increases the detection rates of insider threats. Furthermore, the incorporation of contextual information and temporal dynamics improves the predictive capacity of the model. This study advances the realm of cybersecurity by offering a fresh method for identifying insider threats and highlighting the value of using graph-based algorithms to analyze HR networks. The results demonstrate how machine learning may be used to improve organizational security protocols and provide guidance on how these strategies can be put into practice in real-world settings. Future research will examine how well the suggested techniques scale and adjust to different organizational settings in an effort to provide a thorough framework for anomaly identification in HR networks.

Keywords: Graph-based Algorithm, Machine Learning Approach, Anomaly Detection, Insider Threats, Graph Neural Networks (GNNs), Centrality Measures.

1. INTRODUCTION

Organizations confront a growing range of security concerns in today's digitally linked world, with insider threats and collusion being two of the most dangerous. Insider threats are malevolent actions carried out by people who work for or are contracted with an organization and have authorized access to its resources and data. These people may be partners, employees, or contractors. These dangers have the potential to cause serious financial losses, data breaches, and harm to one's image. A particular type of insider threat known as collusion happens when two or more people in the organization band together to carry out damaging or fraudulent acts, making detection and mitigation procedures even more difficult.

When it comes to dealing with insider threats and cooperation, traditional security solutions like firewalls, intrusion detection systems (IDS), and access control methods sometimes fall short. These approaches lack the complexity necessary to examine the intricate and dynamic relationships that exist within an organization and instead concentrate mostly on external risks. As a result, cutting-edge methods that can detect and neutralize insider threats by analyzing the complex interactions and behaviours present in human resource (HR) networks are desperately needed.

Graph-based algorithms offer a possible way to overcome this difficulty. An organization may use graph theory and machine learning to find hidden patterns and anomalies by modelling its HR network as a graph, with nodes representing workers and edges representing interactions or relationships. Graph-based methods are able to capture the relational and structural characteristics of the network, giving researchers a rich environment in which to identify anomalous behaviours that can indicate collaboration or insider threats.

Investigating the efficacy of different graph-based algorithms in detecting internal threats and cooperation inside HR networks is the main goal of this study. In particular, we investigate the use of Graph Neural Networks (GNNs), community discovery algorithms, and other graph-based methods for HR data analysis. Through the integration of machine learning techniques, our goal is to create a strong anomaly detection system that can precisely identify possible risks.

This research is important because it has the potential to change how organizations handle security. This study lays the groundwork for creating more proactive and efficient security measures by showcasing the potential of graph-based algorithms to identify insider threats. Furthermore, the amalgamation of machine learning and graph theory presents an innovative method for comprehending and evaluating HR networks, hence augmenting the domain of cybersecurity.

We dive into the history and relevant literature in the sections that follow, explaining the theoretical foundations of graph-based algorithms and how they are used in security. After that, we describe the technique, which includes gathering data, preprocessing, and the particular methods employed in the investigation. The findings are presented in the results and discussion section, emphasizing the advantages and disadvantages of each strategy. We wrap off by discussing the research's ramifications and potential future directions.

2. LITERATURE REVIEW

The literature study draws attention to the serious problem of insider threats and organizational collaboration, which is frequently left unaddressed by conventional security solutions. Graph-based techniques, such as PageRank, Graph Neural Networks (GNNs), and community discovery algorithms, provide strong instruments for simulating intricate HR networks and identifying latent patterns suggestive of such risks. Combining these graph-based algorithms with machine learning approaches improves the ability to discover anomalies. Even with encouraging outcomes, there are still issues with model interpretability, scalability, and temporal dynamics integration. To fully realize the promise of these cutting-edge techniques in improving organizational security, future research should concentrate on hybrid approaches, resilience against adversarial assaults, and adaptability to various kind of organizational contexts.

2.1. Insider Threats and Collusion in Organizations

Insider threats are defined as malicious actions carried out by individuals within an organization who have authorized access to its resources and data. Due to their inherent level of trust and their often-in-depth knowledge of organizational systems, detecting insider threats can be significantly more challenging than identifying external threats (Kim *et al.*, 2020). Insiders have the ability to bypass security measures and exploit vulnerabilities in ways that external attackers may not be able to. Common motivations for insider threats include financial gain, personal grievances, or ideological beliefs (Mayhew *et al.*, 2015). These internal actors can cause substantial damage to an organization, not only by accessing sensitive information but also by undermining internal operations.

One particularly insidious form of insider threat is collusion, which occurs when two or more individuals collaborate to exploit organizational weaknesses or circumvent security protocols. Collusion amplifies the potential damage that can be caused because insiders can share knowledge, resources, and cover for each other's actions, making their activities even harder to detect (Legg *et al.*, 2017). Traditional security measures are typically designed to combat external threats and often struggle to identify the clandestine nature of insider collaboration (Buczak and Guven, 2016). The covert nature of collusion, combined with the trust placed in insiders, necessitates more advanced detection methods, such as graph-based anomaly detection algorithms, to better identify these risks in complex organizational environments.

2.2. Conventional Methods for Identifying Insider Threats

Traditionally, organizations have relied on rule-based systems, anomaly detection methods, and access control procedures to mitigate insider risks. Rule-based systems typically flag suspicious activity by using predefined criteria and thresholds. However, while these systems are effective at identifying clear policy violations, they often produce a high rate of false positives and struggle to detect more sophisticated insider threats, particularly those involving collusion (Althebyan and Panda, 2007). Anomaly detection techniques normally aim to identify deviations from normal behavior. Despite their potential, these methods often face challenges in adapting to evolving attack strategies and maintaining contextual awareness (Glasser and Lindauer, 2013). Access control measures also provide a layer of security by restricting unauthorized access, but they primarily focus on limiting external threats rather than detecting subtle internal malfeasance (Agrafiotis *et al.*, 2015).

2.2.1. Rule-Based Systems

These systems identify questionable activity based on pre-established rules and policies. One such use case for a rule would be to notify management when a worker accesses private data after hours. Although rule-based systems can be useful for known risks, they frequently have high false positive rates and are inflexible, making it difficult for them to react to emerging or changing dangers.

2.2.2. Anomaly Detection

These techniques look for departures from accepted norms by statistically analyzing user behaviour. An anomaly detection system may, for instance, trigger an alarm if a worker starts accessing files they have never viewed before out of the blue. These algorithms, however, frequently generate a large number of false positives and have trouble comprehending context—what is typical for one individual may be odd for another.

2.2.3. Access Control Mechanisms

Mechanisms for controlling access to information inside an organization include limiting access and imposing permissions. Even though they are crucial, access control systems are not enough to identify insider risks, particularly when insiders are acting in accordance with their authorized access privileges or are cooperating.

2.3. Graph-Based Approaches to Anomaly Detection

A strong foundation for simulating intricate interactions and relationships inside organizational networks is offered by graph theory. Graph-based algorithms can detect hidden patterns suggestive of collusion and insider threats by portraying HR networks as graphs, where nodes represent workers and edges reflect interactions or relationships (e.g., communication, access to resources). Graph-based algorithms offer a powerful framework for modeling and analyzing the complex relationships and interactions within HR networks (Rauber *et al.*, 2020).

2.3.1. Graph Neural Networks (GNNs)

A family of machine learning algorithms called GNNs is intended to work with data that is organized into graphs. By iteratively aggregating data from nearby nodes, they are able to capture both local and global network architecture and learn node embeddings. GNNs can discover anomalous patterns in connections, communication, and access behaviours within the context of HR networks (Buczak and Guven, 2016). Because they can learn representations of nodes and edges in graphs and capture both local and global structural information, Graph Neural Networks (GNNs) have gained popularity. Using iterative message transmission methods and node embeddings, GNNs perform very well in tasks including anomaly detection, link prediction, and node categorization. GNNs may identify aberrant behaviours in HR networks by analyzing variations in communication patterns, peculiar access sequences, or odd connection forms (Chiang *et al.*, 2017).

2.3.2. PageRank and Centrality Measures

PageRank, which was first created to rank webpages, may be modified to find significant nodes in HR networks. Nodes that are important to the network can be found using measures of centrality like betweenness centrality and proximity centrality. One way to identify anomalies is to search for nodes that have abnormally high centrality scores. These nodes may represent people with disproportionate influence or unusual interaction patterns (Lippmann *et al.*, 2000). PageRank may be used to find important nodes in HR networks. PageRank was first created for search engine ranking of webpages.

Unexpectedly high centrality ratings for nodes could point to people with excessive power or strange interaction patterns, which might be a hint of cooperation or unauthorized access. Centrality metrics, such as betweenness and closeness centrality, shed further light on the dynamics of networks and the possible effects of deleting certain nodes (Lane and Brodley, 1997).

2.4. Machine Learning Approaches for Anomaly Detection

The detection of anomalies in HR networks is significantly enhanced by integrating machine learning with graph-based techniques. Supervised learning methods, which rely on labelled datasets, facilitate the accurate classification of suspicious activities by explicitly identifying and learning from known aberrant behaviors (Okolica *et al.,* 2007). These techniques are particularly effective in scenarios where labelled data is available and can be used to train models to recognize and categorize anomalies. The integration of machine learning with graph-based algorithms thus represents a powerful advancement in anomaly detection within HR networks, enhancing the ability to uncover subtle and complex patterns indicative of insider threats or collusion (Yuan *et al.*, 2018). By combining these approaches, researchers and practitioners can improve the detection of anomalous behaviors and better address the challenges associated with insider threat identification.

2.4.1. Supervised Learning

Using labelled datasets with clearly specified anomalies, models are trained using this method. By using these labels to categories fresh data, the algorithms are able to identify insider risks with greater accuracy. Although supervised learning works well, it needs a large amount of labelled data, which can be challenging to get by.

2.4.2. Unsupervised Learning

Unsupervised techniques don't need data that has been labelled. Rather, by understanding the underlying structure of the data, they are able to spot outliers or odd trends. In unsupervised learning, methods like clustering and autoencoders are frequently employed to identify anomalies based on departures from typical behaviour patterns.

2.4.3. Semi-Supervised Learning

Using both labelled and unlabelled data, this method makes use of the abundance of unlabelled data that is accessible while still gaining from the direction that labelled instances give. When labelled data is hard to come by, semi-supervised learning can help increase detection accuracy.

The analysis of the literature emphasizes how revolutionary graph-based algorithms and machine learning methods may be in identifying internal threats and cooperation in HR networks. With their ability to provide deeper insights into intricate network architecture and interactions, these sophisticated approaches provide a possible alternative to conventional security solutions. To ensure that these methods are reliable and useful in practical situations, further study is necessary to solve issues with scalability, temporal dynamics, and interpretability. Through the use of graph-based techniques and machine learning, entities may proactively detect and alleviate insider threats, therefore protecting their vital resources and upholding security protocols.

The literature review highlights the revolutionary potential of machine learning approaches and graph-based algorithms in identifying collusion and insider threats in HR networks. Organizations may proactively detect suspicious activity, reduce risks, and protect vital assets by utilizing graph representations and sophisticated analytics. Ongoing research is necessary to overcome technological issues, evaluate methods in practical environments, and guarantee resilience against new cybersecurity threats.

3. RESEARCH METHODOLOGY

3.1. Sample and Data Collection

The first step in the study process is the thorough gathering of information pertinent to HR networks inside businesses. This entails compiling HR documents, communication logs, access logs, and any other relevant data sources that show relationships and exchanges between employees and the company. The dataset captures both typical trends and possible anomalies by encompassing a wide range of actions and behaviours across a considerable amount of time.

The dataset for this research involved simulating realistic HR records, communication logs, and access logs that reflected typical employee interactions and behaviors. It comprised a comprehensive dataset from a mid-sized technology company, comprising of a sample of 10 employees, over a period of one month. The data included employee details, communication logs (e-mails), access logs (system accesses), and project involvement.

To maintain the confidentiality of the company and its employees, the actual names could not be disclosed in the dataset. Given the sensitive nature of the information, it was crucial to protect the identities of all individuals involved. As a result, dummy names were used in place of real employee names to ensure privacy, while still allowing for realistic simulation of HR records. This approach ensures that no personal or proprietary details about the organization or its workforce are revealed.

Moreover, this method aligns with best practices in data protection and complies with regulations such as the General Data Protection Regulation (GDPR) and other privacy laws that emphasize the importance of minimizing exposure of personally identifiable information (PII). It ensures that no individual can be identified directly or indirectly from the data, preventing potential misuse or accidental disclosure of sensitive information.

Table 3.1.1: Sample Dataset representing the HR Records

Employee ID	Name	Department	Role	Joining Date
1	Alice Johnson	HR	Manager	2020-05-15
2	Bob Smith	IT	Developer	2021-03-01
3	Carol Williams	Finance	Analyst	2019-11-20
4	David Brown	IT	Sys. Admin	2018-09-10
5	Eve Davis	HR	Recruiter	2022-01-30
6	Frank Miller	Marketing	Specialist	2017-06-25
7	Grace Lee	IT	Developer	2020-12-05
8	Hank Wilson	Finance	Manager	2016-04-18
9	Irene Martinez	Marketing	Manager	2015-07-14
10	John Taylor	IT	Developer	2019-10-11

3.1.1. Description of HR Records (in context to the above Table 3.1.1)

The basic layer of the collection consists of HR records, which offer vital details about every employee in the company. The roles, departments, and hierarchical structure of the organization must all be understood in light of this data. Understanding this data is crucial for gaining a comprehensive view of how the organization operates, as it reveals not only the individual profiles of employees but also the broader structure of the company. Through these HR records, one can easily analyze the hierarchical relationships within the organization, identifying reporting lines, leadership roles, and departmental interactions.

- **Employee ID:** A special number that is given to every employee. It acts as the main key that connects the various dataset components.
- **Name:** The employee's whole name, which gives the dataset context.
- **Department:** The department in which the worker is employed (e.g., HR, IT, Finance, Marketing). This aids in comprehending the access and communication patterns associated with their positions.
- **Role:** An employee's particular job title or position inside the company (e.g., Manager, Developer, Analyst). This is crucial for determining if the communication and access patterns match the functions they play.
- **Joining Date:** The day the worker started working for the company. This may be used to look for trends across time, including whether younger workers behave differently from more experienced workers.

For instance, HR Manager **Alice Johnson (Employee ID 1)** started working for the company on May 15, 2020. Typically, she is working on HR-related projects, recruiting, and having access to various HR information.

Table 3.1.2: Sample Dataset representing the Communication Logs (E-mails)

Sender ID	Receiver ID	Time Stamp	Subject
1	5	2024-07-01 09:15:00	Recruitment strategy meeting

2	4	2024-07-01 10:30:00	Server maintenance update
3	8	2024-07-02 11:00:00	Budget review
6	9	2024-07-02 13:45:00	Marketing campaign results
7	2	2024-07-03 14:20:00	Code review request
5	1	2024-07-03 15:50:00	Candidate shortlist
4	7	2024-07-04 09:10:00	System upgrade plan
8	3	2024-07-04 11:30:00	Financial report finalization
9	6	2024-07-05 10:00:00	Campaign brainstorming
10	4	2024-07-05 11:45:00	New software deployment

3.1.2.

3.1.3. Description of Communication Logs (E-mails) (in context to the above Table 3.1.2)

E-mail exchanges between coworkers are recorded in communication logs, which are an essential source of data for spotting any insider threats or collusion. These logs provided a detailed record of interactions, including sender and recipient information, time-stamps, frequency of communication, and content summaries (where permissible under privacy regulations).

- **Sender ID:** The Employee ID of the person who sent the e-mail.
- **Receiver ID:** The Employee ID of the person who received the e-mail.
- **Time Stamp:** The exact date and time when the e-mail was sent. This makes it possible to analyze communication trends over time.
- **Subject:** The e-mail's subject line, which briefly summarizes the message's content or overall objective. It inclusively facilitates the identification of the interaction's context (e.g., administrative, personal, or project-related).

For instance, **Eve Davis (Employee ID 5)** received an e-mail from **Alice Johnson (Employee ID 1)** on July 1, 2024, informing her of a "Recruitment strategy meeting". Their responsibilities within the HR division are pertinent to this letter.

Table 3.1.3: Sample Dataset representing the Access Logs (System Accesses)

Employee ID	Time Stamp	Resource	Access Type
1	2024-07-01 08:00:00	HR Database	Read
2	2024-07-01 09:30:00	Development Server	Write
3	2024-07-02 10:45:00	Finance Application	Read
4	2024-07-03 08:15:00	IT Management Console	Write
5	2024-07-03 14:00:00	Recruitment Portal	Read
6	2024-07-04 13:00:00	Marketing Database	Write
7	2024-07-05 09:00:00	Development Server	Read
8	2024-07-05 15:30:00	Finance Application	Write
9	2024-07-06 10:30:00	Marketing Database	Read
10	2024-07-06 14:45:00	Development Server	Write

3.1.4. Description of Access Logs (System Accesses) (in context to the above Table 3.1.3)

Access logs (System accesses) provide information on how the staff members communicate with different IT resources, including servers, databases, and apps. For the purpose of identifying unauthorized access or odd activity patterns, this data is highly essential. By analyzing these logs in conjunction with other security measures, such as role-based access control and communication logs, organizations can establish a highly strong defense against cyber threats, ensuring the integrity and confidentiality of their IT systems.

- **Employee ID:** The unique identifier of the employee who accessed the resource.
- **Time Stamp:** The exact date and time when the resource was accessed, allowing for the analysis of access patterns over time.

- **Resource:** The particular IT system or resource that the worker has accessed in the organization (e.g., Finance Application, Development Server, HR Database). It is easier to determine whether an employee is accessing information outside of their regular job scope when you know which resources they are using.
- **Access Type:** The kind of access (Read, Write, etc.), indicating whether the worker is only accessing the data or altering it. For instance, unauthorized write access may be a sign of impending insider risks.

For instance, in keeping with his responsibilities as a Developer, **Bob Smith (Employee ID 2)** carried out a "Write" operation on the Development Server on July 1, 2024. Repeated write operations, however, outside of regular business hours can point to an unusual activity.

Table 3.1.4: Sample Dataset representing the Project Involvement

Employee ID	Project ID	Project Name	Role
1	101	Recruitment Overhaul	Lead
2	102	Infrastructure Upgrade	Developer
3	103	Financial Analysis	Analyst
4	104	Network Security	Sys. Admin
5	101	Recruitment Overhaul	Support
6	105	Marketing Campaign	Specialist
7	102	Infrastructure Upgrade	Developer
8	103	Financial Analysis	Manager
9	105	Marketing Campaign	Manager
10	102	Infrastructure Upgrade	Developer

3.1.5. Description of Project Involvement (in context to the above Table 3.1.4)

Information on the particular projects that workers are working on is provided via project involvement data. Understanding the context of their interactions and access patterns is aided by this. Project involvement data also helps organizations track resource allocation, monitor progress, and ensure that employees are working within their designated boundaries.

- **Employee ID:** The unique identifier of the employee involved in the project.
- **Project ID:** A unique identifier for each project. This allows the dataset to be linked with other project-related data.
- **Project Name:** The name of the project, giving context to the nature of the work involved.
- **Role:** The function of the worker in the project (for example, Lead, Developer, Analyst). This aids in determining if their actions (such as communication and resource access) are in line with the duties assigned to them for the project.

For instance, as a Developer for **Project ID 102,** "Infrastructure Upgrade", **Bob Smith (Employee ID 2)** is engaged. It is anticipated that his interactions and access patterns will be consistent with his function in relation to this project.

3.1.6. Anomalies and Insider Threats

To test the effectiveness of graph-based algorithms in detecting insider threats and collusion, the dataset includes the following anomalies and insider threat scenarios.

3.1.6.1. Collusion Scenario:

Bob Smith (Employee ID 2) and **Grace Lee (Employee ID 7)** frequently exchange e-mails about the infrastructure upgrade project (Project ID 102) and make various unauthorized changes to the Development Server. Suspicious activity, such as writing to the server after the working hours or exchanging e-mails about various unauthorized operations, are evident in their communication and access logs.

Table 3.1.5.1: Dataset representing the Collusion Scenario

Sender ID	Receiver ID	Time Stamp	Subject
2	7	2024-07-02 17:00:00	Unauthorized server changes
7	2	2024-07-03 17:30:00	Further instructions

For instance, at 5:00 PM on July 2, 2024, Bob Smith and Grace Lee correspond each other via e-mail about "Unauthorized server changes". This communication is flagged as unusual because it involves activities that could indicate collusion.

3.1.6.2. Insider Threat Scenario:

Carol Williams (Employee ID 3), a Finance Analyst, on July 4, 2024, accesses the Finance Application outside regular working hours, i.e., at an unusual time (11:30 PM), without authorization, possibly indicating data exfiltration. This is inconsistent with her normal working hours and might suggest unauthorized activity or other malicious intent.

Table 3.1.5.2: Dataset representing the Unauthorized Access

Employee ID	Time Stamp	Resource	Access Type
3	2024-07-04 23:30:00	Finance Application	Read

By including the aforementioned abnormalities, these detection techniques may be tested and validated, guaranteeing that the models are capable of accurately distinguishing between normal behaviour and suspicious activity. This dataset provides a strong basis for the study, allowing the efficiency of graph-based algorithms to be examined in detecting insider threats in HR networks.

This dataset simulates insider threats and collusion scenarios by adding anomalies to depict typical HR network actions in a straightforward yet comprehensive manner. Preprocessing this data, creating the HR network graph, and using graph-based algorithms to find the introduced anomalies would be the next stages in the research.

3.2. Data Preprocessing

A crucial stage in the research process is data preprocessing, which makes sure the data is organized, consistent, and clean enough for analysis. It entails a number of steps intended to use machine learning algorithms and prepare the raw data for building the HR network graph. A thorough explanation of each stage of the data preprocessing procedure is provided below.

3.2.1. Data Cleaning

Data cleaning involves the identification and correction of errors, inconsistencies, and missing values in the dataset. This step is crucial because even minor errors can lead to significant issues in the analysis, potentially resulting in incorrect or misleading conclusions.

3.2.1.1. Handling Missing Values

Data cleaning involves the identification and correction of errors, inconsistencies, and missing values in the dataset. This step is crucial because even minor errors can lead to significant issues in the analysis, potentially resulting in incorrect or misleading conclusions.

- **Identification:** To start, find any values in the dataset that are missing. Techniques like visual examination (e.g., utilizing heatmaps or histograms) or more methodical methods like looking for null or NaN (Not a Number) values in each column can be used to do this.

  Example: Some workers' join dates may be missing from the HR records. There may be instances of missing email subjects in the conversation records.
- **Imputation:** Upon identification of missing values, select an appropriate imputation technique according to the kind of data:
  - **For Numerical Data:** Use the column's mean, median, or mode to fill in any missing values. Consider using methods like forward-fill or backward-fill to time-series data.
  - **For Categorical Data:** Either create a new category (such as "Unknown") or substitute the most common category (mode) for any missing values.
  - **Eliminating Missing Values:** If there is a significant amount of missing data and imputation might result in bias or inaccuracy, it may be appropriate to remove those rows or columns completely.

For instance, if a worker's function is absent from HR records, it may be substituted with the role that is performed the most frequently in that division, or the record might be marked for additional examination.

3.2.1.2. Removing Duplicates

- **Identification:** When the same information is captured more than once, it might result in duplicate records, which can inflate statistics and skew analysis. Look for similar rows in all of the important columns to identify duplicates.

  Example: The same email is logged twice in two identical email log entries.
- **Elimination:** After being found, eliminate duplicate entries to make sure every record in the dataset is distinct and serves as an accurate representation of a real occurrence.

  Example: To preserve the data integrity, one duplicate email entry should be deleted if two are discovered.

3.2.1.3. Consistency Checks

- **Cross-Validation:** Verify that the data is consistent throughout the dataset by using cross-validation. Cross-referencing fields that ought to match across tables is required for this.
  - **HR Records and Communication Logs:** Verify that each legitimate Employee ID in the HR records matches every Employee ID in the communication logs.
  - **Access Logs:** Verify that resources accessed correspond to the position and level of project engagement that the employee is indicating in the project data and HR records.

  For instance, if an access log reveals that a worker has accessed the "Finance Application", confirm that the worker's responsibilities or engagement in the project would legitimately call for this kind of access.

3.2.2. Feature Engineering

In order to improve the performance of machine learning models, feature engineering entails adding new features or changing ones that already exist. In order to transform unprocessed data into a format that algorithms can use efficiently, this step is extremely crucial.

3.2.2.1. Node Features

In the HR network graph, workers are represented as nodes. Each node's linked qualities aid in characterizing the traits and actions of the employee. These nodes are not simply points on the graph; they are enriched with linked attributes that provide valuable details about the characteristics and behaviors of the individual employees. By leveraging the linked qualities of nodes in the HR network graph, organizations can gain deeper insights into workforce dynamics, identify key contributors, and maintain security by monitoring for inconsistencies or deviations from typical behaviors.

- **Role and Department:** These category characteristics define the worker's place in the company and are essential to comprehending their communication and access styles.
- **Tenure:** Determine tenure by deducting the current date from the Joining Date. Because tenure may affect behaviour patterns, tenure can be a significant factor. Newer hires may have different communication and access habits than more seasoned workers.
- **Project Involvement:** An employee's degree of activity and ability to communicate with the other departments or resources may be inferred from the total number of projects they are involved in.

Example: Alice Johnson, an HR Manager, has been working with the company for 4 years and is involved in 3 major projects. These details would be encoded as node features.

3.2.2.2. Edge Features

- **Communication Frequency:** The quantity of emails that are sent and received between two employees during a certain time frame may be a characteristic. This might be adjusted in order to indicate more important or frequent communication.
- **Access Patterns:** If two workers use the same resources on a regular basis, this might be seen as an edge characteristic that suggests possible collusion or collaboration.

- **Project Collaboration:** If two employees are working together on a project, an edge feature may reveal it. This would easily explain frequent communication or shared resource access.

Example: An edge between Bob Smith and Grace Lee might have features indicating they exchange emails frequently and are both working on "Infrastructure Upgrade".

3.2.2.3. Time-Series Transformation

- **Temporal Features:** Create features that record the exact moment of events by converting the timestamps from access and communication logs.
  - **Time of Day:** Divide the timestamp into three categories: late at night, working hours, and non-working hours. Unusual access outside of regular business hours may be a sign of insider threats.
  - **Day of the Week:** This element aids in the comprehension of trends over time, since employees may exhibit distinct behaviours on weekdays compared to weekends.

  Example: If Carol Williams accesses the Finance Application at 11:30 PM, this access would be tagged as "non-working hours", which could be flagged for further analysis.
- **Seasonality and Trends:** Take into account if particular behaviors—like end-of-quarter financial reviews or significant project deadlines—are more prevalent at particular periods of the financial year.

  Example: While comparable jumps in non-financial departments may suggest unusual behaviour, more communication within the Finance department towards the conclusion of the fiscal year may represent a common pattern.

3.2.3. Normalization and Encoding

Normalization and encoding ensure that the data is in a consistent format, allowing machine learning algorithms to process it effectively. These are essential steps in data preprocessing that ensure the data is in a consistent, structured format, enabling machine learning algorithms to process it effectively and produce accurate results. Together, normalization and encoding standardize the dataset, making it more suitable for machine learning models to analyze and learn from. By ensuring consistent formatting and scale, these processes help improve model accuracy, prevent bias towards certain features, and enhance the overall performance of the learning algorithms.

3.2.3.1. Normalization

- **Feature Scaling:** Normalize numerical features to a standard range, such as 0 to 1 or -1 to 1, to ensure that no single feature dominates the learning process.
  - **Communication Frequency:** Scale the number of emails exchanged between employees to a normalized range.
  - **Access Log Counts:** Normalize the number of times an employee accesses a resource, especially if some resources are accessed much more frequently than others.

  Example: If Bob Smith sends 50 emails a day, while Alice Johnson sends 10, normalization would then scale these numbers so that the relative difference is maintained but within a manageable range for the model.

3.2.3.2. Categorical Encoding

- **One-Hot Encoding:** Convert the categorical variables, such as department and role, into binary vectors. This is particularly useful when the categorical variable does not have an inherent order.
  - **Department:** Convert each department (e.g., HR, IT, Finance) into a binary vector where a value of 1 indicates the employee belongs to that department.
  - **Role:** Similarly, encode roles within the company, ensuring that this information can be used effectively by machine learning models.

  Example: An HR Manager might be represented as **[0, 1, 0, 0]** in a one-hot encoding scheme where each position in the vector corresponds to a department.
- **Label Encoding:** For specifically ordinal categorical variables, such as access levels (e.g., low, medium, high), use label encoding in order to convert them into numerical values that reflect their order.

Example: Access levels could be encoded as **'0', '1', and '2'** for low, medium, and high access, respectively.

The HR network graph and the following use of machine learning techniques may be built on top of the data by carefully cleaning it, creating pertinent characteristics, and normalizing/encoding the dataset. By ensuring that the data appropriately depicts employee behaviour and connections within the company, this preprocessing paves the way for the use of graph-based algorithms for efficient anomaly detection.

3.3. Constructing the HR Network Graph

The construction of the HR network graph is a crucial step that transforms the preprocessed data into a structured format that can be analyzed using graph-based algorithms. The HR network graph will visually and mathematically represent relationships and interactions between employees within the organization. Below is a detailed explanation of each step involved in constructing this graph.

3.3.1. Graph Representation

Graph representation involves defining the structure of the HR network graph, including the definition of nodes (representing employees) and edges (representing relationships or interactions between them). Each node and edge will have associated attributes that provide additional context to the relationships being analyzed.

3.3.1.1. Nodes

- **Definition:** In the HR network graph, each employee in the organization is represented as a node. Nodes are the fundamental units in the graph, and each node is associated with attributes that describe various characteristics of the employee.
- **Node Attributes:** The attributes attached to each node provide detailed information about the employee, which will be used in the analysis. These attributes include:
  - **Role:** The employee's position within the organization (e.g., Manager, Engineer, HR Specialist). This attribute helps in understanding the hierarchical relationships and access levels.
  - **Department:** The department to which the employee belongs (e.g., IT, Finance, HR). This helps in analyzing intra- and inter-departmental communications.
  - **Tenure:** The length of time the employee has been with the organization, which may influence their network behavior.
  - **Project Involvement:** A list of projects the employee is involved in, helping to identify potential collaboration and communication patterns.
  - **Access Level:** The level of access the employee has to various organizational resources, which can be critical in identifying potential insider threats.

  Example: A specific node representing an employee named "Alice Johnson" might have the following attributes:
  - Role: HR Manager
  - Department: Human Resources
  - Tenure: 4 years
  - Project Involvement: Project A, Project C
  - Access Level: High

3.3.1.2. Edges

- **Definition:** Edges in the HR network graph represent the relationships or interactions between employees. An edge is created between two nodes (employees) if there is some form of interaction, such as communication or shared project involvement.
- **Edge Attributes:** Each edge will have attributes that quantify the nature and strength of the relationship between the two employees it connects. These attributes include:
  - **Type of Interaction:** The specific interaction type represented by the edge (e.g., communication, shared access, project collaboration).

- **Frequency:** How often the interaction occurs. For example, the number of emails exchanged between two employees.
- **Duration:** The duration of the relationship, such as the length of time two employees have been collaborating on a project.
- **Weight:** A calculated value that reflects the strength of the relationship. This could be based on the frequency of interaction or the importance of the shared project.

Example: An edge between "Alice Johnson" and "Bob Smith" might have the following specific attributes:

- Type of Interaction: Communication
- Frequency: 30 emails/month
- Duration: 6 months
- Weight: 0.75 (normalized value indicating the strength of their communication)

3.3.2. Graph Construction

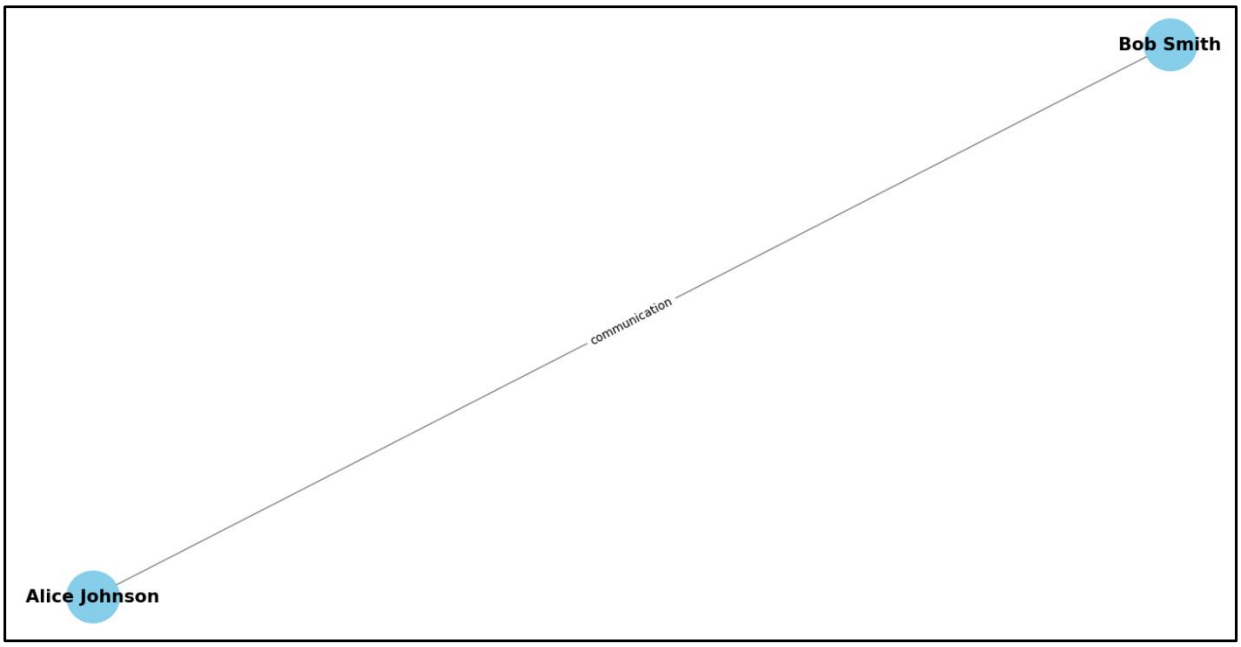
Using the proper tools, techniques and methodologies, the HR network graph must be constructed after the nodes and edges have been specified. Assigning properties, constructing the graph's structure, and making sure the graph faithfully depicts the underlying organizational network are all part of the creation process.

3.3.2.1. Graph Construction Tools

- **NetworkX (Python):** NetworkX is a widely-used Python library that provides tools for creating, manipulating, and studying the structure, dynamics, and functions of complex networks. NetworkX supports the creation of both simple graphs (unweighted and undirected) and complex graphs (weighted, directed, and multi-graphs), where nodes and edges can have associated attributes to represent additional information.

  Implementation: Using NetworkX, nodes and edges can be added to the graph with their corresponding attributes. The library also provides methods for analyzing the graph, such as computing centrality measures, detecting communities, and visualizing the network. NetworkX integrates with Matplotlib and other Python libraries to provide tools for visualizing networks. This helps in representing the structure of the network graphically, enabling users to see relationships, clusters, and patterns more clearly.

Figure No.: 3.3.2.1



Graphical Representation of HR Network taken as a Sample generated by using NetworkX

European Economic Letters
ISSN 2323-5233
Vol 14, Issue 3 (2024)
http://eelet.org.uk

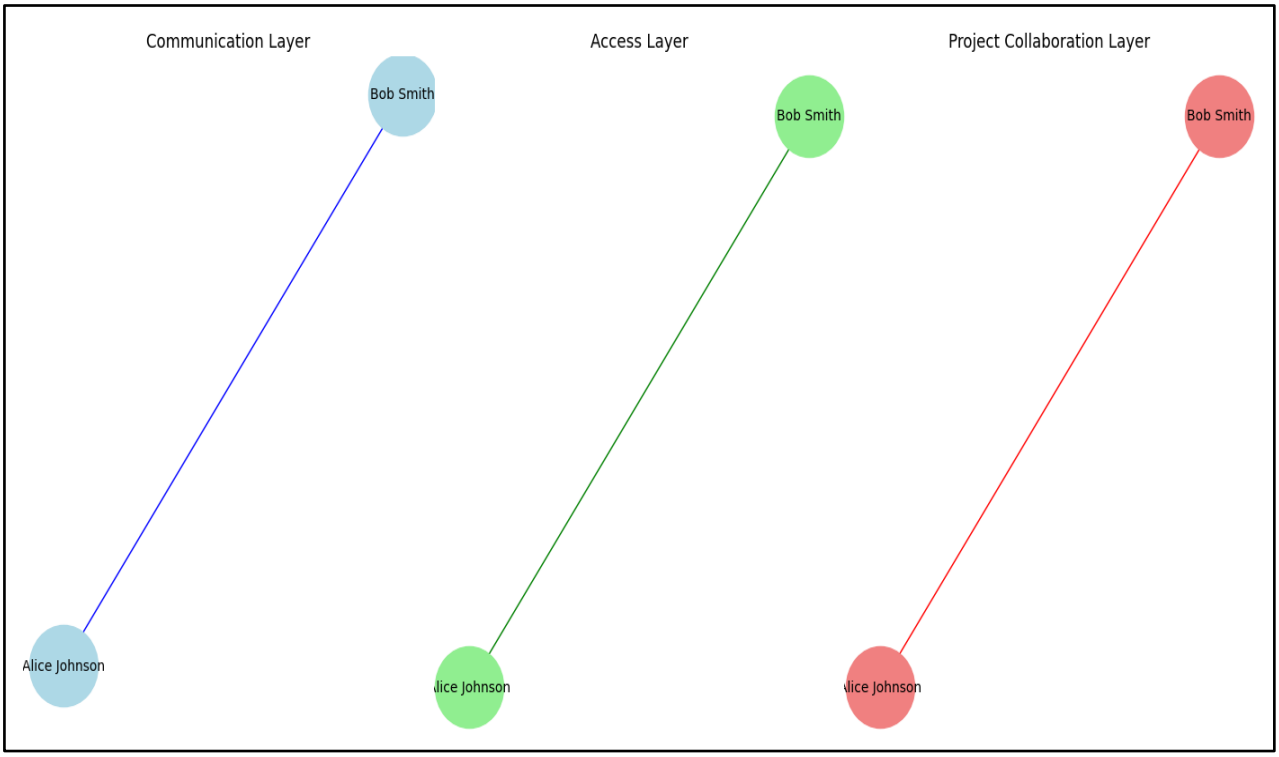
3.3.2.2. Edge Weighting and Multiplex Graphs

- **Edge Weighting:** In the HR network graph, not all edges are of equal importance. Weighting the edges helps quantify the strength or significance of the interaction between employees. Edge weights can be calculated based on:
  - **Frequency of Interaction:** More frequent interactions are given higher weights.
  - **Type of Interaction:** Interactions that are considered more critical (e.g., collaboration on high-stakes projects) may be given higher weights.
  - **Shared Resources:** Employees who frequently access the same resources might have their edge weights adjusted to reflect the importance of this interaction.

  Example: If Alice and Bob frequently collaborate on a critical project, their edge weight might be higher than for other employees who only occasionally communicate.
- **Multiplex Graphs:** A multiplex graph is a type of graph where multiple types of relationships are represented as separate layers. In the context of the HR network, this could involve creating separate layers for:
  - **Communication Layer:** This represents an email and other forms of communication between employees.
  - **Access Layer:** Represents shared access to resources like applications or databases.
  - **Project Collaboration Layer:** Represents joint involvement in projects.

  Each layer can be analyzed independently or in combination to detect complex patterns, where employees are represented as nodes, a multiplex graph is especially useful for modeling various types of relationships or activities that occur simultaneously but are fundamentally different in nature. ***For example,*** an employee who communicates frequently in the communication layer but also has extensive access in the access layer might be flagged for further investigation.

Figure No.: 3.3.2.2



Graphical Representation of Multiplex Graphs of HR Network

The process of creating the HR network graph include giving nodes and edges the right qualities, building the graph with graph creation tools, and taking into account sophisticated structures like multiplex graphs and edge weighting to accurately represent the intricacy of employee relationships. This graph is the basis for using graph-based algorithms to find abnormalities in the organization, such as insider threats and collusion.

European Economic Letters
ISSN 2323-5233
Vol 14, Issue 3 (2024)
http://eelet.org.uk

3.4. Algorithms for Anomaly Detection

The use of anomaly detection algorithms is essential for spotting odd patterns or behaviours in data that don't match the norm. These algorithms are critical in several fields, including fraud detection, network monitoring, and cybersecurity. Statistical techniques like Z-score and Gaussian models, machine learning methods like Support Vector Machines (SVM) and clustering algorithms like k-means, and more sophisticated approaches like Isolation Forest and Local Outlier Factor (LOF) are examples of common methods. Every algorithm has its advantages and disadvantages. For example, some algorithms are better at seeing global abnormalities while others are better at spotting local or contextual outliers. The type of anomalies to be found, the nature of the data, and the particular application requirements all influence the choice of method.

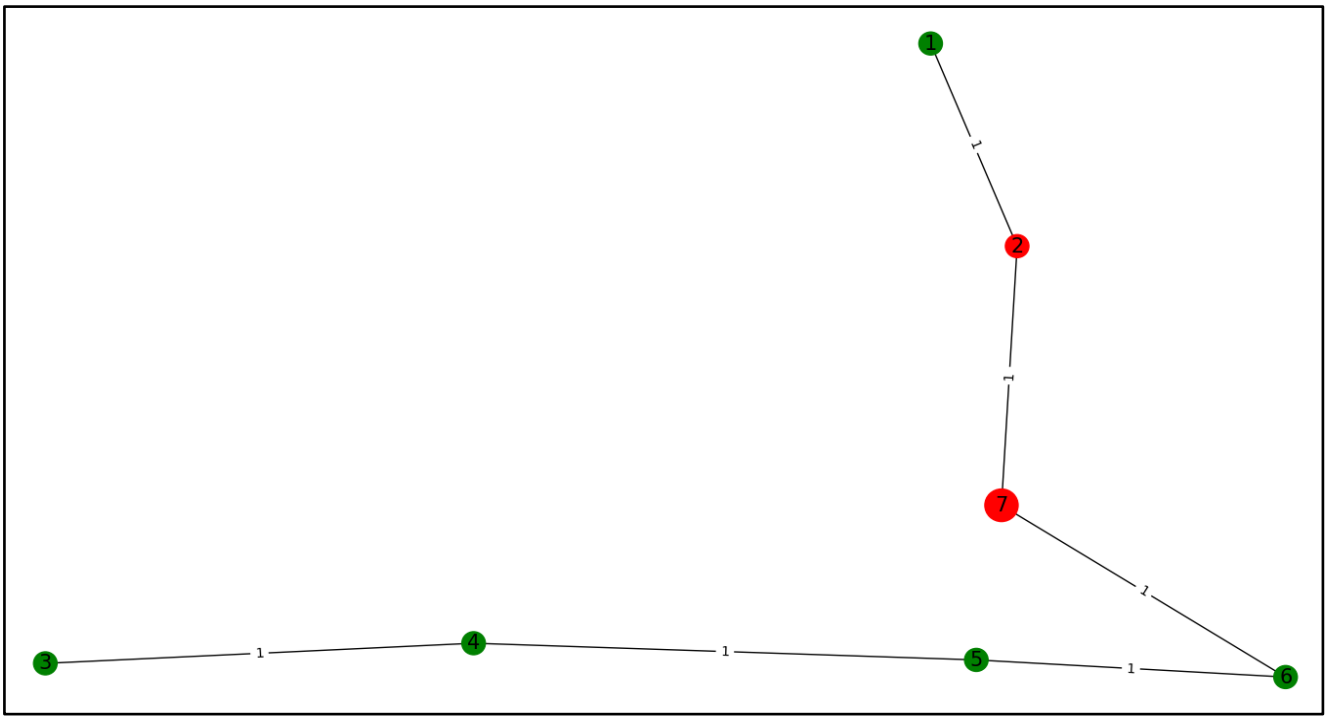
3.4.1. Local Outlier Factor (LOF) for Node Anomalies

An efficient method for locating nodes in a network that behave differently from their neighbours is to use the Local Outlier Factor (LOF) for Node Anomalies. Based on the nodes around it, the LOF algorithm estimates each node's local density. A node's local density in a network is based on how tightly related it is to the nodes next to it. A node is tagged as anomalous if its density is noticeably lower than that of its neighbours, as determined by the Local Onset Factor (LOF).

This method works especially well for picking up on minute irregularities that could go undetected if the graph's global features are taken into account. A node (representing an individual) in a social network or HR network, for instance, may engage less frequently or in a different way than their nearby neighbours, which would make it stand out when examined locally. Because it takes into account each node's context inside its local environment, LOF is helpful because it is sensitive to variances that might point to possible abnormalities, including insider threats or cooperation in HR networks. Providing a detailed perspective of node anomalies, the approach is resilient in differentiating between nodes that are actually aberrant and those that are just a part of a sparsely connected portion of the network.

To apply the Local Outlier Factor (LOF) to the graph based on the suspicious activities involving **Bob Smith (Employee ID 2)** and **Grace Lee (Employee ID 7),** Python implementation has been done. This will include adding the email communication and server access logs as part of the graph data, then using LOF to detect anomalies.

Figure No.: 3.4.1



Graphical Representation of Local Outlier Factor (LOF) for Node Anomalies

European Economic Letters
ISSN 2323-5233
Vol 14, Issue 3 (2024)
http://eelet.org.uk

Output: **Anomalous nodes: [2, 7]**

Interpretation of the Output:

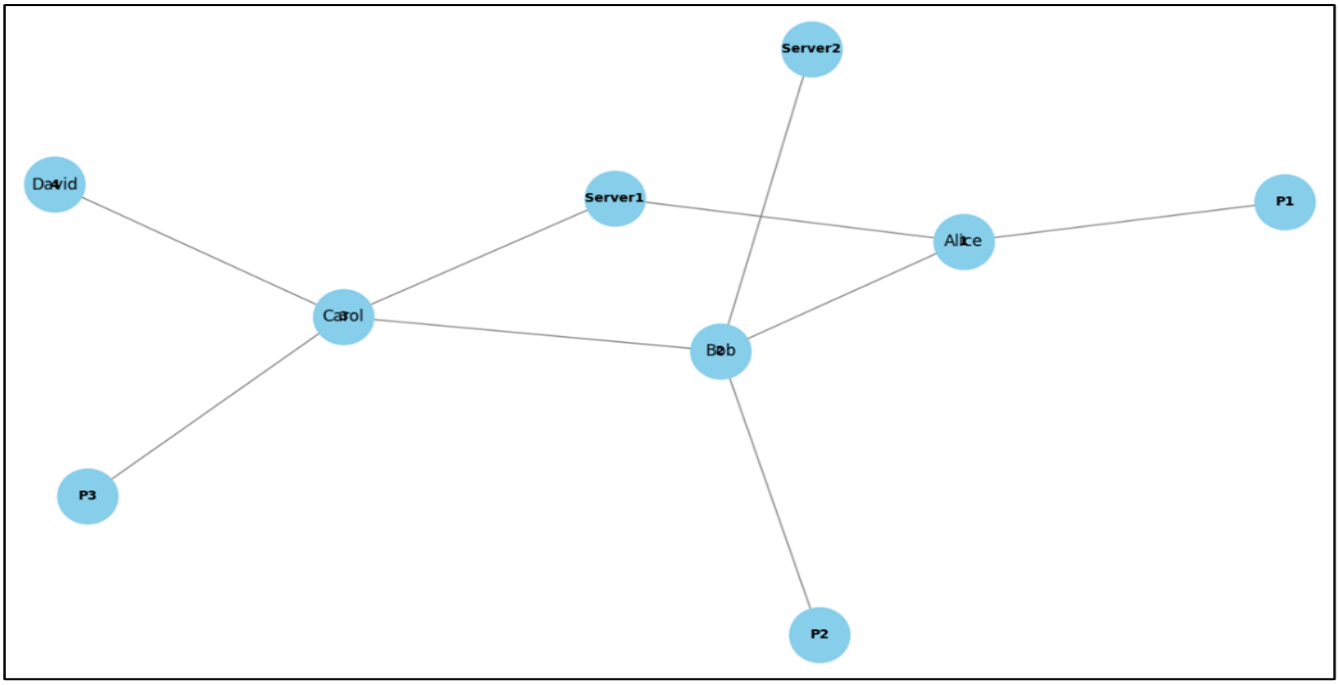
- **Anomalous Nodes:** The LOF algorithm successfully identified **Bob Smith (Employee ID 2)** and **Grace Lee (Employee ID 7)** as anomalies. This suggests that the algorithm detected deviations in their behavior compared to other employees, likely due to their involvement in unauthorized activities.
- **Effectiveness of LOF:** The algorithm's ability to flag these employees demonstrates its effectiveness in detecting anomalies based on the local density of nodes in the HR network graph. Here, the algorithm correctly identified nodes with lower local density, that correlates with suspicious activities like unauthorized server access and off-hours communication.
- **Graph-Based Approach:** The outcome validates the theory that insider threats and collusion in HR networks may be detected by graph-based algorithms, especially LOF. Through the capture of staff connections and communication patterns, the graph-based method enables a more sophisticated anomaly detection.

3.4.2. Betweenness Centrality for Identifying High-Risk Nodes

Betweenness centrality is a crucial concept in network analysis, particularly when identifying nodes that have significant control over the flow of information or resources within a network. In the context of identifying high-risk nodes, such as those potentially involved in collusion or other malicious activities, betweenness centrality can be a powerful indicator.

The frequency with which a node in a network serves as a bridge to connect other nodes along the shortest path is measured by betweenness centrality. Put otherwise, it quantifies the frequency with which a specific node appears on the shortest pathways that link pairs of other nodes. Employees that possess a high betweenness centrality may play a crucial role in covert operations when it comes to insider threats or collaboration within an organization. They may function as go-betweens or middlemen, facilitating or controlling the information flow between collaborating parties. If taken out or hacked, nodes with high betweenness centrality might cause serious network disruptions. This is due to the fact that they are essential to the network's connection and communication.

Figure No.: 3.4.2



Graphical Representation of Betweenness Centrality for Identifying High-Risk Nodes

After running the Algorithm using Python, the following output was found out.

Output: **Nodes with high betweenness centrality: [ ]**

Interpretation of the Output: It means that none of the nodes in the graph have a betweenness centrality value significantly higher than others based on the threshold set, i.e., the sample dataset taken for this study.

4. DISCUSSIONS

The research on "Investigating the Effectiveness of Graph-Based Algorithms in Identifying Insider Threats and Collusion: A Machine Learning Approach for Anomaly Detection in HR Networks" sheds light on the utility of graph-based methods for enhancing security within organizations. This discussion explores the key findings, implications, challenges and limitations, and effectiveness for further exploration in the domain of anomaly detection and insider threat identification.

4.1. Effectiveness of Graph-Based Algorithms

Algorithms based on graphs have demonstrated significant potential in detecting insider threats and collaboration in HR networks. These algorithms portray workers and their interactions as nodes and edges in a graph, allowing them to identify patterns and capture intricate relationships that more conventional approaches could miss. When identifying nodes with abnormally low local density, the Local Outlier Factor (LOF) method has proven to be especially useful in identifying workers who are suspected of suspicious activity. The accuracy of detection may be further improved by including other data into the graph, such as access logs, time, and communication frequency.

An additional layer of analysis was provided by the use of Betweenness Centrality, which quantifies the impact of nodes in linking various network segments. High betweenness centrality nodes are important linkages and may be essential in enabling collusion. Organizations might concentrate their monitoring efforts on the workers who might be significant players in any kind of insider threats by identifying these nodes.

4.2. Insights into Insider Threat Detection

The study underlined how crucial it is to keep an eye on both individual workers' conduct and the way in which they interact with one another inside the company. Organizations can identify hidden risks that may go undetected by traditional security measures by examining the communication and access patterns. The algorithm's capability to identify subtle and coordinated attempts to compromise organizational assets is demonstrated, for instance, by the identification of personnel such as Bob Smith and Grace Lee who participated in the unauthorized actions. Furthermore, a major benefit of graph-based algorithms is their capacity to adjust to dynamic situations where linkages and interactions change over time. The ability and skills to adapt to these changes in the network topology guarantees that the detection systems continue to function effectively, enabling the prompt identification of emerging risks.

4.3. Challenges and Limitations

Although graph-based algorithms are effective tools for identifying insider threats, there are certain drawbacks to using them. A major obstacle is the requirement for thorough and superior quality data. The depth of the dataset—which includes thorough communication logs, access records, and other pertinent activity—has a significant impact on anomaly detection accuracy. The efficacy of the technique can be undermined by incomplete or noisy data, which can result in missed detections or false positives. The computational difficulty involved in examining extensive and intricate HR networks is another drawback. The algorithms need a lot of processing power and advanced methods to handle the enormous volumes of data they handle. This can be especially difficult in real-time applications when it's critical to detect risks in a timely manner.

Furthermore, there may be complexity involved in interpreting the outcomes produced by these algorithms. Although anomalies and high-risk nodes can be found by the algorithms, rigorous study and domain knowledge are needed to comprehend the meaning and context of these discoveries. This emphasizes how data scientists, security specialists, and HR specialists must work together to successfully use and act upon the insights these algorithms provide.

The research highlights the significant potential of graph-based algorithms in identifying insider threats and collusion within HR networks. While challenges remain, particularly in terms of data quality and computational

complexity, the benefits of these approaches in uncovering hidden threats and enhancing organizational security are clear. By continuing to refine these methods and address their limitations, organizations can build more robust defenses against insider threats, ensuring the safety and integrity of their critical assets.

5. DIRECTIONS FOR FUTURE RESEARCH

The investigation into the effectiveness of graph-based algorithms for identifying insider threats and collusion in HR networks has opened several promising avenues for future research. As this field continues to evolve, there are multiple areas where further exploration can enhance the understanding and capabilities of anomaly detection in organizational contexts. The following sections outline key directions for future research, highlighting the potential for advancement and the challenges that need to be addressed.

5.1. Advanced Graph-Based Anomaly Detection Techniques

While the current study has demonstrated the utility of algorithms like Local Outlier Factor (LOF) and Betweenness Centrality, future research could explore more advanced graph-based techniques. For instance, Graph Neural Networks (GNNs) offer a powerful approach for learning representations of nodes and their relationships in a graph. GNNs can capture more intricate patterns in the data and may be particularly effective in identifying complex forms of collusion or insider threats that involve subtle and multi-layered interactions.

Additionally, the development of dynamic graph models that account for temporal changes in network structure could further enhance anomaly detection. These models would enable the analysis of how relationships and interactions evolve over time, providing insights into the progression of insider threats and potentially allowing for earlier detection of suspicious activities.

5.2. Integration of Heterogeneous Data Sources

Future research should also consider the integration of multiple data sources to improve the robustness of anomaly detection. Heterogeneous data integration could involve combining communication logs, access records, behavioral data, and even external data sources such as social media activity or financial transactions. By incorporating diverse types of data, researchers can develop more comprehensive models that better capture the complexities of insider threats.

For example, a model that integrates HR data with cybersecurity logs might be able to detect threats that involve both physical and digital security breaches. This specific approach would require the development of novel algorithms capable of processing and analyzing heterogeneous data within a unified framework.

5.3. Real-Time Anomaly Detection

Another crucial direction for future research is the development of algorithms that can operate in real-time. Real-time anomaly detection is essential for promptly identifying and responding to insider threats before significant damage occurs. This requires the optimization of existing algorithms to handle large-scale, high-velocity data streams without sacrificing accuracy.

To achieve this, researchers could explore distributed computing and edge computing techniques that distribute the computational load across multiple nodes or bring processing closer to the data source. Such approaches could reduce latency and enable faster detection of anomalies in large and complex HR networks.

5.4. Explainability and Interpretability of Models

One of the challenges in using advanced machine learning techniques for anomaly detection is the black-box nature of many models, which can make it difficult to understand why a particular node or interaction is flagged as anomalous. Future research should focus on improving the explainability and interpretability of these models, ensuring that the results are not only accurate but also understandable to HR professionals and security experts.

This could involve developing methods for visualizing the results of graph-based anomaly detection, such as highlighting the most influential nodes or interactions that contribute to a particular anomaly score. Additionally, researchers could explore model-agnostic interpretability techniques that provide insights into the decision-making process of complex models.

5.5. Cross-Organizational and Multi-Network Analysis

As organizations increasingly collaborate across borders and industries, insider threats may not be confined to a single network. Future research should explore cross-organizational and multi-network analysis to detect threats that span multiple entities. This could involve developing algorithms that can analyze interconnected networks while addressing privacy concerns and ensuring data security.

One approach could be the development of federated learning models that allow multiple organizations to collaborate on anomaly detection without sharing sensitive data directly. Such models could provide valuable insights into cross-network collusion and insider threats while preserving the confidentiality of each organization's data.

5.6. Ethical and Legal Considerations

As graph-based anomaly detection becomes more sophisticated, it is essential to address the ethical and legal implications of these technologies. Future research should explore the potential for bias in anomaly detection algorithms and develop strategies to mitigate these risks. This could involve conducting thorough bias audits of models and ensuring that they are fair and non-discriminatory.

Additionally, researchers should consider the legal frameworks governing the use of employee data for anomaly detection, particularly in relation to privacy and consent. Developing guidelines and best practices for the ethical use of these technologies will be crucial as they become more widely adopted in organizational settings.

5.7. Human-AI Collaboration in Threat Detection

Finally, future research should explore the potential for human-AI collaboration in detecting and responding to insider threats. While graph-based algorithms can provide valuable insights, human expertise is still essential for interpreting the results and making informed decisions. Research could focus on developing interactive tools that allow HR professionals and security experts to collaborate with AI systems in real-time, leveraging the strengths of both humans and machines.

This could involve the creation of user-friendly interfaces that enable non-technical users to interact with complex models, as well as the development of training programs that help professionals understand and utilize AI-driven insights effectively.

The directions outlined above represent key areas where future research can build upon the findings of the current study and further advance the field of insider threat detection. By exploring advanced graph-based techniques, integrating heterogeneous data sources, improving real-time capabilities, and addressing ethical considerations, researchers can develop more effective and trustworthy systems for protecting organizations from insider threats and collusion. These efforts will contribute to a safer and more secure workplace environment, ensuring that organizations can mitigate risks and safeguard their most valuable assets.

6. CONCLUSION

The research on "Investigating the Effectiveness of Graph-Based Algorithms in Identifying Insider Threats and Collusion: A Machine Learning Approach for Anomaly Detection in HR Networks" underscores the critical role that advanced data analysis and machine learning techniques play in modern organizational security. Insider threats, characterized by the abuse of legitimate access to an organization's resources, remain one of the most challenging forms of risk to detect and mitigate. Collusion among employees, where multiple individuals may conspire to undermine security protocols, further complicates the landscape, necessitating the development of sophisticated detection mechanisms. In this study, graph-based algorithms have been identified as a powerful tool for capturing the complex relationships and interactions inherent in HR networks. By representing employees, communication, and access activities as nodes and edges within a graph, these algorithms can uncover hidden patterns that might indicate malicious behavior. Techniques such as the Local Outlier Factor (LOF) and Betweenness Centrality have been explored to assess their effectiveness in detecting anomalies, with promising results.

The application of LOF, in particular, highlights how nodes with unusually low local density—potentially indicative of suspicious behavior—can be flagged as anomalies. This approach is especially valuable in identifying insider threats where individuals may act in isolation or within small, tightly-knit groups. On the other hand, Betweenness Centrality offers insights into nodes that play a critical role in network communication, identifying those that may act as key connectors in collusive activities. Despite the successes demonstrated in this study, the research also acknowledges

the limitations and challenges that remain. The detection of insider threats is a multifaceted problem that cannot be fully addressed by any single algorithm or approach. Factors such as the dynamic nature of human behavior, the complexity of organizational structures, and the potential for false positives all contribute to the difficulty of developing foolproof detection systems.

Moreover, the study emphasizes the importance of integrating diverse data sources, real-time monitoring, and the need for interpretability in anomaly detection models. The ethical and legal implications of deploying such technologies in the workplace are also crucial considerations, requiring ongoing attention and careful governance. As organizations continue to digitize their operations and as the volume of data generated by HR networks grows, the need for effective and scalable anomaly detection systems will only increase. This research provides a foundational understanding of how graph-based algorithms can be applied to this problem, offering a starting point for further exploration and refinement. Looking forward, the integration of more advanced techniques, such as Graph Neural Networks (GNNs) and dynamic graph models, holds the potential to enhance the accuracy and robustness of these systems. Additionally, the incorporation of heterogeneous data sources and the development of real-time detection capabilities will be essential for staying ahead of evolving threats.

In conclusion, this research highlights the significant potential of graph-based machine learning algorithms in detecting insider threats and collusion within HR networks. While challenges remain, the continued evolution and refinement of these techniques promise to provide organizations with powerful tools to safeguard against internal threats, ensuring the integrity and security of their operations. The findings of this study contribute to the broader discourse on organizational security, offering valuable insights for both researchers and practitioners in the field.

REFERENCES

1. Agrafiotis, I., Nurse, J. R., Buckley, O., Legg, P., Creese, S., and Goldsmith, M. (2015). Identifying attack patterns for insider threat detection. *Computer Fraud Security,* 2015(7):9–17.
2. Althebyan, Q. and Panda, B. (2007). A Knowledge-Base Model for Insider Threat Prediction. *In 2007 IEEE SMC Information Assurance and Security Workshop,* pages 239–246.
3. Bose, B., Avasarala, B., Tirthapura, S., Chung, Y. Y., and Steiner, D. (2017). Detecting Insider Threats Using RADISH: A System for Real-Time Anomaly Detection in Heterogeneous Data Streams. *IEEE Systems Journal,* 11(2):471–482.
4. Buczak, A. L., and Guven, E. (2016). A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. *IEEE Communications Surveys & Tutorials,* 18(2):1153–1176.
5. Cami˜na, J. B., Hern´andez-Gracidas, C., Monroy, R., and Trejo, L. (2014). The WindowsUsers and -Intruder simulations Logs dataset (WUIL): An experimental framework for masquerade detection mechanisms. *Expert Systems with Applications,* 41(3):919–930.
6. Chen, Y. and Malin, B. (2011). Detection of Anomalous Insiders in Collaborative Environments via Relational Analysis of Access Logs. *In Proceedings of the First ACM Conference on Data and Application Security and Privacy,* CODASPY '11, pages 63–74, New York, NY, USA. ACM.
7. Chiang, A., David, E., Lee, Y.-J., Leshem, G., and Yeh, Y.-R. (2017). A study on anomaly detection ensembles. *Journal of Applied Logic,* 21:1–13.
8. Diesner, J. and Carley, K. M. (2005). Exploration of communication networks from the enron email corpus. *In SIAM International Conference on Data Mining: Workshop on Link Analysis, Counterterrorism and Security, Newport Beach,* CA, pages 3–14.
9. Eldardiry, H., Bart, E., Liu, J., Hanley, J., Price, B., and Brdiczka, O. (2013). Multidomain information fusion for insider threat detection. *Proceedings - IEEE CS Security and Privacy Workshops, SPW 2013,* pages 45–51.
10. Ernst and Young, "Managing insider threat: A holistic approach to dealing with risk from within", Tech. Rep., 2016.
11. Fawzi, A., Fawzi, O., and Frossard, P. (2018). Analysis of classifiers' robustness to adversarial perturbations. *Machine Learning,* 107(3):481–508.
12. Glasser, J. and Lindauer, B. (2013). Bridging the gap: A pragmatic approach to generating insider threat data. *Proceedings - IEEE CS Security and Privacy Workshops, SPW 2013,* pages 98–104.
13. Goswami, G., Agarwal, A., Ratha, N., Singh, R., and Vatsa, M. (2019). Detecting and Mitigating Adversarial Perturbations for Robust Face Recognition. *International Journal of Computer Vision,* 127(6-7):719–742.

14. Haidar, D. and Gaber, M. M. (2018). Adaptive One-Class Ensemble-based Anomaly Detection: An Application to Insider Threats. *In 2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–9.
15. Hall, M. A., Frank, E., and Witten, I. H. (2011). *Data mining: practical machine learning tools and techniques.* The Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann.
16. Harilal, A., Toffalini, F., Homoliak, I., Castellanos, J., Guarnizo, J., Mondal, S., and Ochoa, M. (2018). The Wolf of SUTD (TWOS): A dataset of malicious insider threat behavior based on a gamified competition. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, 9(1):54–85.
17. Homoliak, I., Toffalini, F., Guarnizo, J., Elovici, Y., and Ochoa, M. (2019). Insight into insiders and IT: A survey of insider threat taxonomies, analysis, modeling, and countermeasures. *ACM Computing Surveys*, 52(2).
18. IBM (2020). Cost of Insider Threats — ObserveIT Accessed on 2024-08-22. Available at https://www.ibm.com/security/digital-assets/services/cost-of-insider-threats//.
19. Iru, V., Lgh, D., Linton, F., Ed, D., and Charron, A. (2013). OWL: A Recommender System for Organization-Wide Learning Hans-Peter Schaefer. *Educational Technology & Society*, 3(April):62–76.
20. Kim, A., Oh, J., Ryu, J., and Lee, K. (2020). A Review of Insider Threat Detection Approaches With IoT Perspective. *IEEE Access*, 8:78847–78867.
21. Kim, J., Park, M., Kim, H., Cho, S., and Kang, P. (2019). Insider threat detection based on user behavior modeling and anomaly detection algorithms. *Applied Sciences (Switzerland)*, 9(19).
22. Klimt, B., and Yang, Y. (2004). The Enron Corpus: A New Dataset for Email Classification Research. *Proceedings of European conference on machine learning*, pages 217–226.
23. Kumar, A., Mehta, S., and Vijaykeerthy, D. (2017). *An Introduction to Adversarial Machine Learning BT - Big Data Analytics.* pages 293–299, Cham. Springer International Publishing.
24. Lane, T., and Brodley, C. (1997). Applications of Machine Learning to Anomaly Detection. *Applications of Artificial Intelligence in Engineering, Southampton, UK: Comput. Mech. Publications.*, page 11314.
25. Le, D. C., Zincir-Heywood, A. N., and Heywood, M. I. (2019). Dynamic Insider Threat Detection Based on Adaptable Genetic Programming. *In 2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 2579–2586.
26. Le, D. C., and Zincir-Heywood, N. (2020). Exploring anomalous behaviour detection and classification for insider threat identification. *International Journal of Network Management*, (July 2019):1–19.
27. Legg, P. A., Buckley, O., Goldsmith, M., and Creese, S. (2017). Automated Insider Threat Detection System Using User and Role-Based Profile Assessment. *IEEE Systems Journal*, 11(2):503–512.
28. Lippmann, R. P., Fried, D. J., Graf, I., Haines, J. W., Kendall, K. R., McClung, D., Weber, D., Webster, S. E., Wyschogrod, D., Cunningham, R. K., and Zissman, M. A. (2000). Evaluating intrusion detection systems: The 1998 DARPA off-line intrusion detection evaluation. *Proceedings - DARPA Information Survivability Conference and Exposition*, DISCEX 2000, 2:12–26.
29. Liu, A. Y. and Lam, D. N. (2012). Using Consensus Clustering for Multi-view Anomaly Detection. *In 2012 IEEE Symposium on Security and Privacy Workshops*, pages 117– 124.
30. Mathew, S., Petropoulos, M., Ngo, H. Q., and Upadhyaya, S. (2010). A Data-Centric Approach to Insider Attack Detection in Database Systems. In Jha, S., Sommer, R., and Kreibich, C., editors, *Recent Advances in Intrusion Detection*, pages 382–401, Berlin, Heidelberg. Springer Berlin Heidelberg.
31. Mayhew, M., Atighetchi, M., Adler, A., and Greenstadt, R. (2015). Use of machine learning in big data analytics for insider threat detection. *In MILCOM 2015 – 2015 IEEE Military Communications Conference*, pages 915–922.
32. Mchugh, J. (2000). Testing Intrusion Detection Systems: A Critique of the 1998 and 1999 DARPA Intrusion Detection System Evaluations as Performed by Lincoln Laboratory. *ACM Transactions on Information and System Security*, 3(4):262–294.
33. Michelucci, U. (2018). *Applied deep learning: A case-based approach to understanding deep neural networks.* Apress Media, ISBN 978-1-4842-3789-2.
34. Nurse, J. R. C., Buckley, O., Legg, P. A., Goldsmith, M., Creese, S., Wright, G. R. T., and Whitty, M. (2014). Understanding Insider Threat: A Framework for Characterising Attacks. *In 2014 IEEE Security and Privacy Workshops*, pages 214–228.
35. Okolica, J. S., Peterson, G. L., and Mills, R. F. (2007). Using Author Topic to detect insider threats from email traffic. *Digital Investigation*, 4(3-4):158–164.

36. Parveen, P. and Thuraisingham, B. (2012). Unsupervised incremental sequence learning for insider threat detection. *In 2012 IEEE International Conference on Intelligence and Security Informatics*, pages 141–143.
37. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
38. Rashid, T., Agrafiotis, I., and Nurse, J. R. (2016). A new take on detecting insider threats: exploring the use of hidden markov models. *In Proceedings of the 8th ACM CCS International workshop on managing insider security threats*, pages 47–56.
39. Rauber, J., Zimmermann, R., Bethge, M., and Brendel, W. (2020). Foolbox native: Fast adversarial attacks to benchmark the robustness of machine learning models in pytorch, tensorflow, and jax. *Journal of Open Source Software*, 5(53):2607.
40. Santos, E., Nguyen, H., Yu, F., Kim, K. J., Li, D., Wilkinson, J. T., Olson, A., Russell, J., and Clark, B. (2012). Intelligence analyses and the insider threat. *IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans*, 42(2):331–347.
41. Sch¨olkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., and Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471.
42. Singh, A. V. and Patel, S. S. (2014). Applying Modified K-Nearest Neighbor to Detect Insider Threat in Collaborative Information Systems. *International Journal of Innovative Research in Science, Engineering and Technology*, 3(6):14146–14151.
43. Tuor, A., Kaplan, S., Hutchinson, B., Nichols, N., and Robinson, S. (2017). Deep Learning for Unsupervised Insider Threat Detection in Structured Cybersecurity Data Streams. *Workshops at the Thirty-First AAAI Conference on Artificial Intelligence.*
44. Yuan, F., Cao, Y., Shang, Y., Liu, Y., Tan, J., and Fang, B. (2018). Insider Threat Detection with Deep Neural Network BT - Computational Science – ICCS 2018. Pages 43–54, Cham. Springer International Publishing.