Advances in Artificial Intelligence and Machine Learning; Research (Ahead of Print) Received 7-12-2025; Accepted 23-02-2026; Published online 2-03-2026

Detecting Anomalous States Through File Operations Using Unsupervised Learning Algorithms

Islambek Saymanov islambeksaymanov@gmail.com
National University of Uzbekistan,
Uzbekistan Kimyo International University in Tashkent,
Tashkent, Uzbekistan

Firdavs Muxammadiev muhammadiyev$_f$@$nuu.uz$
$NationalUniversityofUzbekistan,$
$UzbekistanEngineeringFederationofUzbekistan,$
$Tashkent, Uzbekistan$

Gayrat Juraev g.jurayev@tsue.uz
Tashkent State University of Economics,
Uzbekistan

Yebo Lu luyebo@zjxu.edu.cn
Jiaxing University,
China

Olga Boiprav smu@bsuir.by
Belarusian State University of Informatics and Radioelectronics,
Belarus

Ruhillo Alaev alayev$_r$@$nuu.uz$
$NationalUniversityofUzbekistan,$
$Uzbekistan$

Obidjon Bozorov o.bozorov@nuu.uz
National University of Uzbekistan,
Uzbekistan

Timur Abdullayev abdulloyev_t@nuu.uz
National University of Uzbekistan,
Uzbekistan

Corresponding Author: Islambek Saymanov

Abstract

To address the challenge of detecting insider threats, this study proposes identifying anomalous cases by analyzing user file operations recorded in organizational system logs. This study involved monitoring user document operations in a special laboratory environment,

Citation: Islambek Saymanov, et al. Detecting Anomalous States Through File Operations Using Unsupervised Learning Algorithms. Advances in Artificial Intelligence and Machine Learning. 2026. (Ahead of Print). https://dx.doi.org/10.54364/AAIML.2026.62284

from which a sample dataset was systematically created for further analysis. Relevant data sources were identified to capture file-based user activities through operating system logs. Features suitable for the study were selected. Cleaning, filtering, and normalization were performed on the data. The cleaned data were consolidated into a single dataset and analyzed using unsupervised learning and statistical methods suitable for classification tasks. The following algorithms were selected: Isolation Forest, Local Outlier Factor, One-Class Support Vector Machine (SVM), and Z-Score. A total of 50 anomalous actions were performed by users in the study. As a result of the evaluation, 36 of these 50 anomalous cases were consistently as anomalies by all algorithms. This study demonstrates the potential for real-time detection of insider threats in the future. The approach is particularly relevant for organizations that handle sensitive data, and it can be integrated into UEBA and DLP systems.

Keywords: Information, Unsupervised learning methods, Anomalous cases, Source, feature, DLP, UEBA, Insider.

1. INTRODUCTION

In the era of developing information technologies, ensuring information security remains one of the main problems. These systems are primarily designed to protect against external attacks. However, nowadays it is also necessary to protect against internal threats.

Internal threats, particularly insider threats perpetrated by users registered in the system, pose a significant danger to organizations. As confirmation of this, we can cite the IBM Security Intelligence report, which states that approximately 60% [1], of data breach incidents are related to insider threats, and this number is increasing year by year.

Such threats are mainly carried out intentionally or through user incompetence [2, 3]. These cases can cause billions of dollars in damage to the organization. In addition, such incidents can cause significant damage to an organization's reputation.

Prevention of insider threats is considered one of the most pressing issues in the field of cybersecurity. In recent years, various approaches have been proposed to eliminate this problem. In particular, methods such as analyzing the text of files, monitoring user behavior, monitoring after-hours activity, determining the size of files transmitted over the network, and analyzing the frequency of user keystrokes are being used [4]. However, most of the existing approaches are based on traditional methods and often do not directly cover processes related to files.

In this study, the goal was to identify anomalies that occur in the operations performed by users on documents in an organization using intelligent methods. The use of intelligent approaches can provide new solutions that allow detecting insider threats at an early stage. This approach serves to detect threats at an early stage by observing anomalies in employee behavior [5, 6]. For this purpose, a special program was installed on the computers of employees at the Cybersecurity Laboratory at the National University of Uzbekistan, and the users' document operations were monitored using this program. As a result of the observation, 2292 cases of users working with documents were collected [7, 8]. Of these, 50 cases were identified as anomalous by industry experts. During the research, the goal was to identify anomalous situations among these cases using intellectual methods

[9]. Based on this, the content of the study was structured as follows: The introduction section stated that insider threats are one of the most critical threats today. The section on related works presents an overview and analysis of articles on data collection, their types, information related to users' work with documents, and the detection of anomalies in these documents [10, 11]. The methodology section described the work carried out. It included identifying sources for detecting anomalous situations in users, identifying features that play a key role in detecting insiders, collecting, cleaning, and normalizing data according to the features. Determining which of the samples are normal and anomalous using unsupervised learning algorithms was described [12]. The results section presented the work that was implemented and tested step by step, and anomalous situations were found among the situations. Also, at the end of the results section, the main conclusions of the study were summarized. The study's contributions and results are briefly described in the conclusion section. It also described how this research work can be enriched in the future and in which areas it can be used.

2. RELATED WORK

In recent years, considerable research has been conducted on the monitoring of anomalous activities. These studies have been applied in various domains, including the field of cybersecurity. Research in this field is mainly focused on data protection, improving network security, and developing network-based intrusion detection systems (IDS). None of the analyzed scientific works fully covers the processes by working directly with documents or files. Most of them are focused on monitoring anomalous activities in the network and user behavior. However, a lot of work has been done in these studies. Below are brief reviews of these works.

The study by Soufleri et al. [13], aimed to preserve privacy by creating a synthetic sample, which helps to train a model without the need for real data. That is, the data in the sample is not real data, but generated data that is closer to real data. However, this data mainly consisted of network data. Krajnc et al. [14], also propose a similar automated data preparation technique, but their sample is primarily focused on medical data. In this study, the data balancing achieved using the SMOTE method is a significant accomplishment. It should also be noted that this approach is not suitable for confidential data, but rather for general samples that may contain confidential information.

IDS systems also use anomaly detection methods as one of the ways to detect attacks. Much research has been done in this direction. For example, Senthilkumar and Arivarasan [15], consider the processes of preparing large amounts of data in the creation of intrusion detection systems (IDS). The achievement of this study is the optimization of data to improve the efficiency of machine learning models.

Alshammari and Aldribi [16], worked on data collection and processing for detecting malicious traffic in cloud computing systems. This study highlights the security problems of cloud infrastructure in data collection and preparation. The study provides valuable information for general security. Both studies focused on intrusion detection systems and did not fully address the issue of detecting anomalies during file operations or user interaction with files but mainly considered network-related anomalies.

Chen [17], focuses on detecting compromised data in systems using unsupervised learning algorithms, while Rathod [18], studies the detection of malicious activity using AI and ML technologies, based on Chellam [19]. Although this study demonstrates the effectiveness of AI technologies in anomaly detection, it does not focus on file-level sensitive data detection.

Chellam et al. [19], demonstrate the effectiveness of Lazy Learning algorithms in IDS systems. These algorithms are effective in monitoring malicious activity in real-time. However, this approach also focuses only on network-level anomaly detection and does not cover file-level issues. D'hooge et al. [20], analyze the generalization of the model for IDS systems under different sampling conditions. They mainly propose general solutions for network and system security.

Ntivuguruzwa and Ahmed [21], propose a local feature optimization method for detecting hidden data in images. Although this study demonstrates new approaches to hiding and detecting confidential information, it focuses on steganographic techniques related to images, rather than on file or general system security. This research work will be handy if applied to DLP systems.

While the remaining research works reviewed [22, 23], are devoted to detecting anomalies in computer devices, detecting network attacks, and monitoring malicious activities in the system, none of them has yet fully addressed the issue of directly detecting insider status by counting user actions on files, as well as creating a sample from this data. In short, the research carried out in this area is primarily focused on network security and intrusion detection systems. Therefore, the approach and technique of identifying insiders by monitoring the files they work on plays a vital role in filling the current scientific gap.

3. METHODOLOGY

Insider activity can be considered anomalous because it deviates from typical user behavior. To detect anomalous activity, the insider's operations on les are monitored by a special program. Monitoring can be done over a specific period of time. The monitoring interval does not need to be long; a few minutes are sufficient. Because taking too much of it can lead to late detection of insider activity. For example, let's say the program monitors user activity for 5 minutes. During this period, a single user can perform multiple operations on a file. For example, he can open a file 2 times or delete a file 3 times and copy it 1 time. Such a situation can be considered normal. However, deleting 25 les or copying 30 files to a USB device may be suspicious. Taking these into account, the collected data will be in digital form. That is, the operations performed by users working with various text les on their computers at specific intervals are counted, collected, and transmitted.

The collected data can be expressed in various formats or forms. Collecting this data is necessary to create a complete and reliable picture of the actions of users in the system. Therefore, it is first required to identify the sources from which the existing data is collected.

Operating system logs, file metadata, information about accessing the operating system, working with confidential information and using them were selected as sources of information for the study. Of these, system logs are considered the primary source of information. They allow you to track user behavior by recording various events that occur in the system. Also, log files represent a sequence of events recorded by the system or programs. It is worth noting that the research used Windows OS and all sources are sources of this operating system.

After identifying the sources of information, it is now necessary to determine the characteristics of the data from these sources that are suitable for the research work. Because the selection of data characteristics is essential for increasing the efficiency of the research. The purpose of selecting characteristics is to reduce the size of the data, optimize the algorithm, and increase the efficiency of anomaly detection. These characteristics were selected based on the above sources of information. Below you can find the characteristics used in the research and their description (TABLE 1):

Table 1: Selection of features for sampling

Features	**Comment**
File Creat	User creates a new file
File Write	User writes to a file
File Rename	User renames a file
File Delete	User deletes a file
File Copy	User copies a file
File Open	User opens a file
File Print	User prints a file
USB Insert	User connects a USB device to the system
System Login	User logs in
Regular File Open	User works on regular files
Confidential File Open	User works on confidential files
Highly_Confidential_File_Open	User works on top-secret files

After identifying the sources, it is necessary to obtain data from these sources. As mentioned above, Event IDs are important for tracking user actions when collecting data. Because each event has its own unique code called Event ID, through these Event IDs, it is possible to know what event is happening. It is necessary to obtain each event in real time. For example, events related to connecting and disconnecting a USB device from the system and copying files in text format to this device are marked with Event IDs 6416, 6417, 4663 in the Windows Security Event Log.

During the research, several problems were encountered when retrieving data from the user's computer from various sources. One of these problems is that the incoming data does not come in the correct format. To solve such a problem, it is necessary to clean the data and handle errors, because the accurate predictions of artificial intelligence algorithms depend on the accuracy of the data. Raw data is usually inaccurate, noisy, and unprocessable, and it is necessary to prepare it for analysis. Since the next steps of the research were to identify anomalous situations using artificial intelligence methods, the following work was carried out to improve the quality of the collected data and bring it into a format convenient for the algorithms:

In some cases, the collected data as a result of the user's operation on the file may contain missing or incorrect property values. For example, the user's action to open the file returned unknown or null instead of open. In such cases, the incorrect values are removed or changed to the agreed-upon default values (TABLE 2).

Table 2: Occurrence of an incorrect value.

Timestamp	User	Action	File	Success
2025-15-01 10:00:24	User1	NULL	example.docx	Success

The same events were recorded multiple times in the log files. For example, records were repeatedly recorded that a user had opened a particular file numerous times. To simplify this process, events were filtered by user ID, action type, and time the action was performed. Redundant events were deleted.

In In the collected data, there were cases where the timestamp of the event was in the incorrect format, or the action type was incorrectly written. For example, the timestamp was written as invalid, and the action type showed an unknown value, such as unknown, and these events were also excluded (TABLE 3). In cases where the timestamp was not in the properties, duplicates were searched.

Table 3: Example of incorrect time.

Timestamp	User	Action	File	Success
Unknown	User1	delete	example.docx	Success

As a result, useful features were extracted from the raw data, incorrect and the information was written to a local data repository on the user's computer. The data collected from the users was compiled into a single file and presented in a unified form (TABLE 4). Then, the sample was checked for errors, identified outliers, removed duplicates, and normalized them. These processes are explained step by step below. The work began with finding and correcting errors in the existing sample.

Table 4: Summary table view.

C	D	O	C	CH	R	W	P	USB	L	RF	CF	HCF
0	2	2	0	2	0	4	0	0	1	2	0	0
2	0	0	0	0	0	0	0	0	1	0	0	0
0	0	5	2	0	0	2	1	0	1	0	0	0
1	0	0	0	0	0	1	0	0	1	0	1	0

After that, the content of the sample was checked. As a result of the check, it was found that there were 2632 event states in the sample, and they were in the following columns:

- Create (C), Write (W), Rename (R), Delete (D), Copy (C), Open (O), Print (P), Change (CH) – file operations
- USB (USB)– USB connection
- System_Login (L)– login to the system.

- Confidential_File (CF), Regular_File (RF), Highly_Confidential_File (HCF) – file opening by confidentiality level

The screen has been defined.

The next step was to check for empty values in the selection. As a result of the check, empty values were detected. These empty values were filled with 0 and converted to int type. After the necessary corrections were made, they were checked again for empty values.

Table 5: Some of the identified duplicates.

Detected duplicate rows	
45 and 675, 567 and 1005, 819 and 1514, 218 and 1895, 1772 and 2550, 842 and 1214, 13 and 900, 109 and 776, 800 and 1668, 405 and 606, 13 and 1459, etc.	Rows 13, 45, 109, 405, 567, 800, 819, 218, 1772, 842 and others were deleted using the *drop_duplicates()* function.

After identifying and filling in the empty values, the sample was moved to identifying duplicates, that is, repeated events. During the identification, 340 duplicates were identified (TABLE 5). This indicates that many users in the system performed actions on the same file. Some of the identified duplicates are listed in the table below. Duplicates were removed from the dataset, resulting in 2292 events remaining in the dataset.

Table 6: Outliner in the selection

C	D	O	C	CH	R	W	P	USB	L	RF	CF	HCF
0	2	2	0	2	0	4	0	0	1	2	0	0
2	0	0	0	1000	0	0	0	0	1	0	0	0
0	0	5	2	0	0	2	1	0	1	0	0	0

The range of values in the columns in the sample is mostly between 0 and 25. This means that the sample does not need to be normalized. Also, the columns in the sample have values very close to each other and rescaling them may be unnecessary.

After collecting, cleaning, and normalizing the data, we now identify anomalous actions performed by the user. For this, we use unsupervised learning algorithms. Algorithms are different, we need classification algorithms depending on the problem statement. The reason is that we need to classify the state from the given data into anomalous or normal. That is, we have two classes: normal (0) and anomalous (1). This process helps to distinguish unusual actions of users from working with normal documents. This makes scientific work scientifically sound and practically useful.

In the next step, using this approach, the process of classifying normal and anomalous user behavior is carried out. From our sample, the following best-fit classification algorithms [24], were selected to detect anomalous cases automatically:

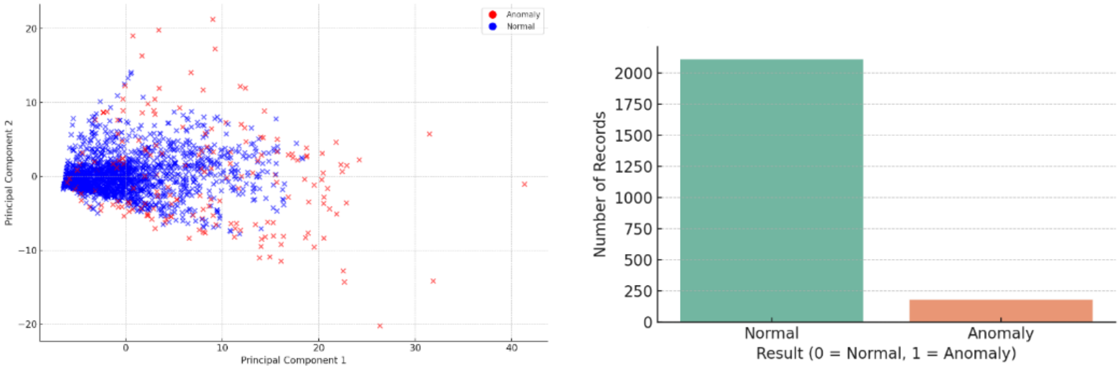
- Isolation Forest

- Local Outlier Factor
- One-Class SVM
- Z-Score

We present our sample to these intelligent methods, and they determine which cases in our sample are anomalous. At the end of our sample, a new column is added, called class, and a method in this column assigns a 1 to the case considered anomalous, and a 0 to the normal case. We test each method separately and finally combine them into one table.

The Isolation Forest method was tested first. The method evaluated each case based on its specific behavior and marked cases that differed from the general structure as anomalies.

Out of 2292 cases, 182 were found to be possible anomalous cases. We can observe the result in a 2D projection based on PCA, where anomalies are shown as red dots and are clearly visible where they depart from the central cluster (FIGURE 1 a). The number of regular and anomalous records is visually compared in a bar chart (FIGURE 1 b), which more clearly demonstrates the effectiveness of the method. The Isolation Forest method demonstrated efficient performance, simplicity, and effective visualization capabilities.

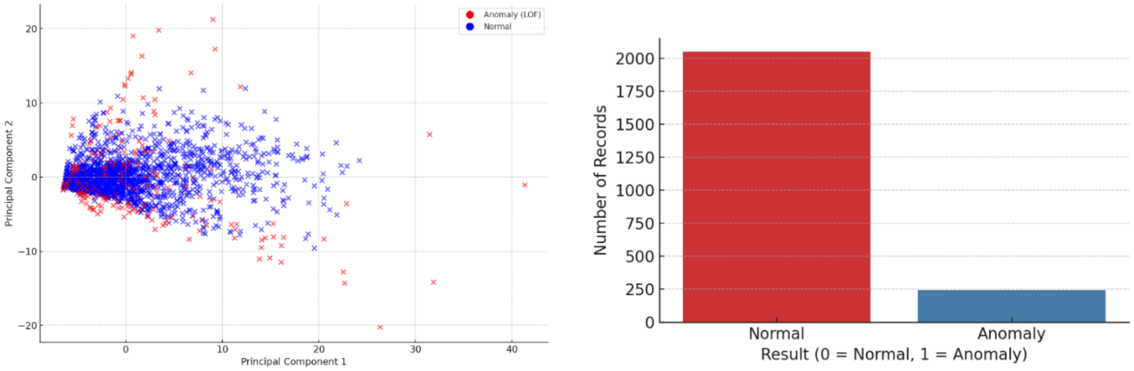


a) b)

Figure 1: Count of normal and anomaly by Isolation Forest

We now continue the analysis using the Local Outlier Factor method — this method estimates the density of each case compared to its surrounding neighbors and identifies anomalies. As a result, 242 cases out of a total of 2292 were identified as anomalies. When visualized in 2D space based on PCA, anomalies stand out as red dots when they leave the central cluster (FIGURE 2 a). The number of normal and anomaly records was easily compared using a bar chart (FIGURE 2 b). Since the Local Outlier Factor method takes into account the assessment of the environment, it effectively revealed unusual behavior at the local level in the system.

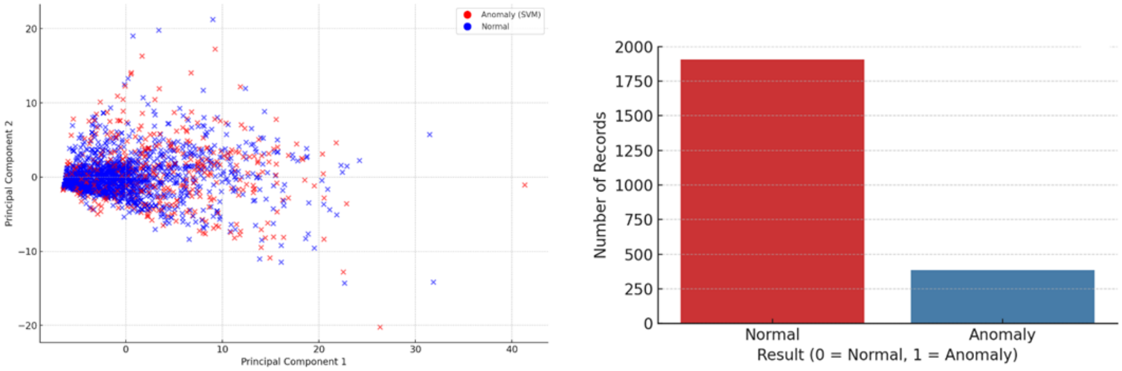
The one-class SVM method aims to identify the boundary of the main normal instances in a dataset. This model creates a hyperplane that envelops the region considered “normal” within the data. If any instance is located outside this boundary, it is evaluated as an anomaly (outlier).



a) b)

Figure 2: Count of normal and anomaly by Local Outlier Factor (LOF)

Based on the One-Class SVM method, 2292 user behavior cases were analyzed to detect anomalies. As a result, 387 cases were identified as anomalous, which accounted for 16.89% of the total cases. In the PCA-based visualization, it was clearly visible that the anomalies were located separately from the cluster in the form of red dots (FIGURE 3 a). However, some anomalous cases are considered normal. They are marked with blue color at the edge of the graph. In the bar chart, normal and anomaly cases were compared numerically (FIGURE 3. b). The strength of this method lies in its ability to distinguish between complex and nonlinear structures clearly. The results showed that the One-Class SVM method is effective in detecting boundary cases in the data.



a) b)

Figure 3: Count of normal and anomaly by One-Class SVM

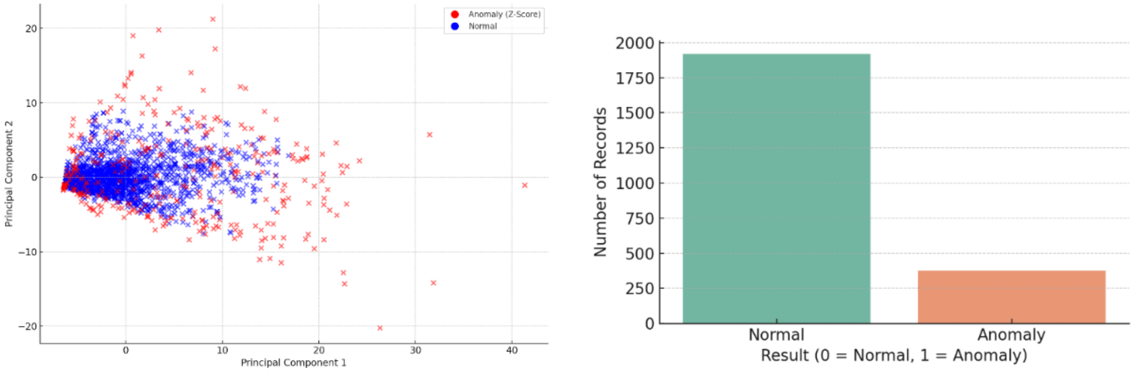
Now we can use one statistical method to identify anomalous cases. Let's try the Z-Score statistical method. The Z-Score statistical method calculates how many standard deviations each case is away from the mean value for its attributes. In this method, the Z-score value for a case is calculated as follows:

$$Z = \frac{x - \mu}{\sigma} \tag{1}$$

Here:
x — status value,
μ — the average value of the attribute,
σ — standard deviation.

If the Z-score value is greater than 3 or less than -3, this is considered an anomaly (outlier). The Z-Score method is a simple, fast, and easy-to-interpret statistical method.

The Z-score method was used to statistically detect anomalies. The distance from the mean value of each case attribute was calculated by how many standard deviations it was. Records with a Z-score value $|Z| > 3$ were identified as anomalies. As a result, 380 out of 2292 records were identified as anomalies. In the 2D projection based on PCA, the anomalies indicated by red dots are located significantly far from the main clusters (FIGURE 4 a). The number of anomalies and normal cases was visually compared using a column chart (FIGURE 4 b). This method is simple and fast, allowing for practical analysis based on clear statistical boundaries. The Z-score method, which relies on statistically justified thresholds, showed promising results in identifying unusual records in the data.



a)

b)

Figure 4: Count of normal and anomaly by Z-score

As a result, we identified the following number of anomalies using the following four methods:

Isolation Forest → 182

Local Outlier Factor → 242

One-Class SVM → 387

Z-Score → 374

4. RESULTS

Now let's compare the results of these methods. First, let's determine how many methods found each case to be anomalous. You can see the results of the analysis in the table below (TABLE 7).

Table 7: Ñomparison of methods.

Number of cases found to be anomalous by the methods	Number of cases	Share (%)
By 0 methods	1618	70.59%
By 1 method	348	15.18%
By 2 methods	177	7.72%
By 3 methods	113	4.93%
By 4 methods	36	1.57%

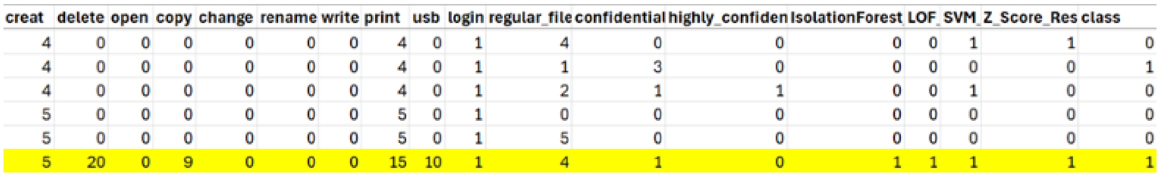
It can be seen from the table that 70.59% of the 2292 cases, or 1618 cases, were not assessed as anomalies by any method. Only 1.57% of the cases were evaluated unanimously as anomalies by all four methods. cases identified as anomalous, 36 were classified as anomalous. This indicates a significant difference between the methods. However, there is a high level of agreement on some cases, which can be assessed as reliable anomalies.

TABLE 8, shows the number of instances each method identified as anomalies. The fewest anomalies were detected by Isolation Forest (7.9%), while the most anomalies were detected by One-Class SVM (16.9%). This shows that each method has different anomaly detection thresholds and model approaches.

Table 8: Methods for detecting anomalous conditions

Metod	Anomaly number	Share (%)
Isolation Forest	182	7.94%
Local Outlier Factor	242	10.56%
One-Class SVM	387	16.88%
Z-Score	374	16.32%

The following table shows the cases where the two methods identified the same condition as an anomaly.



creat	delete	open	copy	change	rename	write	print	usb	login	regular_file	confidential	highly_confiden	IsolationForest	LOF	SVM	Z_Score_Res	class
4	0	0	0	0	0	0	4	0	1	4	0	0	0	0	1	1	0
4	0	0	0	0	0	0	4	0	1	1	3	0	0	0	0	0	1
4	0	0	0	0	0	0	4	0	1	2	1	1	0	0	1	0	0
5	0	0	0	0	0	0	5	0	1	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	5	0	1	5	0	0	0	0	0	0	0
5	20	0	9	0	0	0	15	10	1	4	1	0	1	1	1	1	1

Figure 5: The appearanance of the selected anomaly detection by 4 methods

The most significant similarity was observed between the Local Outlier Factor and Z-score methods, with 159 cases identified as anomalies by both methods (TABLE 9). This shows that these methods act on similar strategies. There is also a high level of agreement between One-Class SVM and Z-Score, and Isolation Forest and Z-Score. This means that Z-score identifies the same cases as many methods, that is, it is a flexible and balanced method in terms of overall sensitivity.

Table 9: Number of cases considered anomalous by the two methods.

Combination of methods	Number of matching cases
Isolation Forest & LOF	85
Isolation Forest & One-Class SVM	79
Isolation Forest & Z-Score	152
Local Outlier Factor & One-Class SVM	102
Local Outlier Factor & Z-Score	159
One-Class SVM & Z-Score	155

Also, during the analysis, 149 cases were identified as anomalies by at least three methods together. These cases can be reliably distinguished as anomalies. To quantitatively assess the detection performance, precision and recall were computed against expert-labeled anomalous cases. Isolation Forest demonstrated the highest recall (0.74) while also achieving the highest precision (0.20) among the evaluated methods, indicating a more balanced performance. Z-Score achieved a recall of 0.56 with a precision of 0.08, showing relatively strong sensitivity. Local Outlier Factor and One-Class SVM exhibited lower recall values of 0.44 and 0.26, respectively, accompanied by reduced precision. These results highlight the trade-off between sensitivity and selectivity across different unsupervised methods and further support the use of a consensus-based ensemble approach for identifying high-confidence anomalous cases. This approach, while covering the specific strategies of each technique, uses the power of an ensemble (using several algorithms) and provides more complete analysis results.

5. CONCLUSION AND FUTURE WORKS

As a result of this study, the detection of insiders among the organization's users through anomalous file handling situations was practically tested using unsupervised learning methods. A total of 2292 cases were used in the study, of which 50 were identified as anomalous cases. Of these, 36 cases were consistently detected by all four methods, while the remaining 14 cases involved elevated activities related to confidential file printing and abnormal USB data transfers. These behaviors indicate high-risk patterns in the studied environment but may not be considered anomalous under different organizational security policies. The research work began with the selection of resources, after which the features suitable for the study were identified from these resources. The problem of data collection, cleaning, and normalization was then analyzed. Finally, the anomaly detection capabilities of unsupervised learning algorithms were utilized to identify anomalous situations within the collected sample. Four unsupervised learning algorithms were used to detect anomalies: Local Outlier Factor, Isolation Forest, One-Class SVM, and Z-score. The Local Outlier Factor algorithm, which analyzed the results, found 242 out of 2292 cases in the sample to be anomalous. One-class SVM gave the highest result. It detected 387 anomalous situations. The second most found algorithm was Z-score. It found 374 anomalies. It found 10 fewer than One-class SVM. Isolation Forest showed the lowest detection rate. It detected 182 anomalies. Comparing these results with those of experts in subsequent studies will further increase confidence in the accuracy of the results. For this, it is advisable to provide the results generated by the algorithms to an expert and have the expert classify them, comparing the results with those of the algorithms. Then, after classification, it can be trained on supervised algorithms.

The results of the study are essential in the field of cybersecurity, as they can be used to identify and eliminate internal threats. Additionally, these results can lead to new opportunities for developing more effective training models to identify insiders with high accuracy.

However, there are some limitations of the study, which are determined by the fact that the sample is used only when working with files. This encourages a deeper analysis of the problem. In the future, it is possible to conduct research in this direction, such as identifying new resource detection features and selective enrichment.

The findings of this study contribute to improving the identification of insider threats among users working with confidential data and demonstrate the practical applicability of intelligent approaches within DLP and UEBA systems.

References

[1] https://securityintelligence.com/articles/83-percent-organizations-reported-insider-threats-2024.

[2] Kabulov A, Normatov I, Saymanov I, Baizhumanov A. On the Completeness of Classes of Correcting Functions of Heuristic Algorithms. Azerb J Math. 2025;15:51-64.

[3] Kabulov A, Baizhumanov A, Saymanov I. Synthesis of Optimal Correction Functions in the Class of Disjunctive Normal Forms. Mathematics. 2024;12:2120.

[4] Kabulov A, Baizhumanov A, Berdimurodov M. On the Minimization of K-Valued Logic Functions in the Class of Disjunctive Normal Forms. J Math Mech Comput Sci. 2024;121:37-45.

[5] Saymanov I, Babadzhanov A, Varisov A, Urinov N, Madjidov A. Numerical Methods of Synthesis of a Correct Algorithm for Solving Recognition Problems. Adv Artif Intell Mach Learn. 2025;5:3534-3547.

[6] Saymanov I. Logical Automatic Implementation of Steganographic Coding Algorithms. J Math Mech Comput Sci. 2024;121:122-131.

[7] Kabulov A, Normatov I, Urunbaev E, Muhammadiev F. Invariant Continuation of Discrete Multi-Valued Functions and Their Implementation. 2021 IEEE International IOT Electronics and Mechatronics Conference. IEMTRONICS. IEEE. 2021:1-6.

[8] Kabulov A, Saymanov I, Yarashov I, Muxammadiev F. Algorithmic Method of Security of the Internet of Things Based on Steganographic Coding. 2021 IEEE International IOT Electronics and Mechatronics Conference. IEMTRONICS. IEEE. 2021:1-5.

[9] Jurayev GU, Bozorov AH, Rakhimberdiyev K. Protection of Transaction Data of Financial Information Systems in Communication Networks Based on Sea80 New Stream Encryption Algorithm. InConference on Internet of Things and Smart Spaces. Springer Nature. 2025:62-73.

[10] Zhang W, Tople S, Ohrimenko O. Leakage of Dataset Properties in Multi-Party Machine Learning. 2020. arXive preprint: https://arxiv.org/pdf/2006.07267v1.

[11] Sun Q, Wei Sh, Saymanov I, Lu Y, Deng W, et al. A Mechanical–Electrical Damage Model for Performance Analysis of Crack-Based Strain Sensor. Int J Appl Mech. 2026;18:2550124.

[12] Ruhillo A. Applying Custom Algorithms in Windows Active Directory Certificate Services. Int J Adv Comput Sci Appl. IJACSA. 2021;12.

[13] Soufleri E, Saha G, Roy K. Synthetic Dataset Generation for Privacy-Preserving Machine Learning. 2022. arXive preprint: https://arxiv.org/pdf/2210.03205v1.

[14] Krajnc D, Spielvogel CP, Grahovac M, Ecsedi B, Rasul S, et al. Automated Data Preparation for in Vivo Tumor Characterization With Machine Learning. Front Oncol. 2022;12:1017911.

[15] Senthilkumar SP, Arivarasan A. Technique for Preparation of Large Data for Machine Learning Algorithms to Generate Intrusion Detection System. ARPN J Eng Appl Sci. 2023:1539-1546.

[16] Alshammari A, Aldribi A. Apply Machine Learning Techniques to Detect Malicious Network Traffic in Cloud Computing. J Big Data. 2021;8:90.

[17] Chen Y, Mao Y, Liang H, Yu S, Wei Y, et al. Data Poison Detection Schemes for Distributed Machine Learning. IEEE Access. 2020;8:7442-7454.

[18] Rathod V, Parekh C, Dholariya D. AI ML Based Anomaly Detection and Response Using Ember Dataset. In: 2021 9th international conference on Reliability. Infocom Technologies and Optimization. Trends and Future Directions. ICRITO. IEEE. 2021:1-5.

[19] Chellam A, L R, S R. Intrusion Detection in Computer Networks Using Lazy Learning Algorithm. Procedia Comput Sci. 2018;132:928-936.

[20] D'hooge L, Wauters T, Volckaert B, Turck FD. Inter-Dataset Generalization Strength of Supervised Machine Learning Methods for Intrusion Detection. J Inf Secur Appl. 2020;54:102564.

[21] De La Croix NJ, Ahmad T. Toward Hidden Data Detection via Local Features Optimization in Spatial Domain Images. In 2023 Conference on Information Communications Technology and Society. ICTAS. IEEE. 2023:1-6.

[22] Zhang W, Tople S, Ohrimenko O. Leakage of Dataset Properties in Multi-Party Machine Learning. IEEE Secur Privacy. 2020.

[23] M B, UMAR, Kumar R K, SS. Data Poison Detection Using Machine Learning. International Journal of Engineering Technology and Management Sciences. 2023;7:560-571.

[24] https://www.eyer.ai/blog/best-ai-models-for-anomaly-detection/.