Threat Analysis using N-median Outlier Detection Method with Deviation Score

Pattabhi Mary Jyosthna, Konala Thammi Reddy
Department of Computer Science and Engineering
GITAM (Deemed to be University)
Visakhapatnam, India

Abstract—**Any organization can only operate optimally if all employees fulfil their roles and responsibilities. For the majority of tasks and activities, each employee must collaborate with other employees. Every employee must log their activities related with their roles, responsibilities, and access permissions. Some users may deviate from their work or abuse their access rights in order to gain a benefit, such as money, or to harm an organization's reputation. Insider threats are caused by these types of users/employees, and those users are known as insiders. Detecting insiders after they have caused damage is more difficult than preventing them from posing a threat. We proposed a method for determining the amount of deviation a user has from other users in the same role group in terms of log activities. This deviation score can be used by role managers to double-check before sharing sensitive information or granting access rights to the entire role group. We first identified the abnormal users in each individual role, and then used distance measures to calculate their deviation score. In a large data space, we considered the problem of identifying abnormal users as outlier detection. The user log activities were first converted using statistics, and the data was then normalized using Min-Max scalar standardization, using PCA to transform the normalized data to a two-dimensional plane to reduce dimensionality. The results of N-Median Outlier Detection (NMOD) are then compared to those of Neighbour-based and Cluster-based outlier detection algorithms.**

Keywords—Organizational roles; insider threats; outlier detection; deviation score

I. Introduction

In a distributed environment, all resources such as infrastructure and data are to be distributed among the employees of an organization to obtain better performance and economic growth of the business. But security becomes a major concern in this distributed environment to avoid unexpected loss of reputation or money of their business. In general, security breaches might occur either from externals who have no rights to access any sort of the organization's resources or from the internals who have legitimate rights to access the infrastructure within the organization [1]. The purpose of insiders is may be to gain money or sensitive data to disrupt the operation or functionalities of an organization. Comparatively, internal threats are harder than external threats to detect. As per the Insider Threat Report by Cyber security Insiders in 2019 [2], 68% of organizations are getting experience with the frequent insider threats. Insider threats can happen by the people purposely or accidentally. Accidental breaches may happen due to careless users or naïve users. 30% of organizations are using some analytical tools to determine insider threat details like user activity management and summary reports in order to reduce the loss caused by these insider threats. Organizations still need to respond quickly in response to the attacks and should be able to identify or predict future threat possibilities. Finding insiders in an organization is a very challenging task to the organizations.

Various Machine Learning (ML) approaches are evolving for carrying out complex and challenging problems that would help to identify and predict malicious intents [4]. In general, a user will be treated as an insider if he/she shows a different behaviour from their previous behaviour and from their peer's behaviour. The abnormal behaviour of an insider within his allotted role can be defined as the deviation score of a user. Behaviour of a user is nothing but his/her activities or computer system usage in the organization [3]. Researchers are applying either classification or clustering algorithms based on the data that they have gathered regarding insiders. If the dataset includes details of the user's activities in some insider threat incidents, then the researchers can use classification algorithms to build a model with that data. This model will be used in future to classify whether the new user activities can lead to internal threat or not. If the data is about user roles and their activities within the organization, then ML clustering algorithms can be used to cluster the users.

To work on or to analyse the historical data about insider's activities, The Computer Emergency Response Team (CERT) Division, in partnership with Exact Data, LLC, and under sponsorship from Defense Advanced Research Projects Agency (DARPA) I2O [5], generated a collection of synthetic insider threat test datasets which will be available publicly. The CERT r6.1 dataset simulates an organization with 4000 users' activities like login/logoff, thumb drive connectivity, file access and their roles over the period of 12 months. The purpose of this paper is to apply existing outlier detection techniques to analyse user activities which are assumed to be generated from different sources and proposed a new N-Median Outlier Detection (NMOD) model to find role wise outliers. Here, a role is nothing but a job role within the organization. This proposed model can able to do the following:

- Aggregate all log files generated from different monitoring tools based on the user activities in an organization.

- Finds the outliers based on the user behavioural patterns when compare with the other users of the same role.
- Generate deviation score of each user based on their activities in a specific role.

The rest of this paper is organized as follows. Section II describes Literature Review; Section III describes the Model for finding the deviation score of a user; Section IV Analyses the Results; and Section V ends with the Conclusion.

II. Literature Review

Insider is an employee in the organization with authorized access rights to access the system resources and knows the vulnerabilities of an organization's infrastructure. The insider is malicious if he/she misuses their access rights to gain benefit out of it. Research on detecting malicious insiders helps the Organizations to take preventive measures.

In the recent years, researchers [4][6][7][8][9][10][11][12] have come up with new supervised and unsupervised Machine Learning (ML) analytical techniques to detect abnormal behaviour of those insiders based on their daily log activities or based on their digital footprints. Researchers of [4] use unsupervised learning techniques such as Isolation Forest and One- class SVM to identify abnormalities in large datasets. They use a trust score which is generated from the previous cycle. Furthermore, they considered the psychometric score of users in their model and checked its effectiveness in identifying insiders. Researchers of [6] mentioned that supervised learning approaches are useful if they have large and balanced data. Otherwise, unsupervised learning approaches are best to predict insiders. They used an unsupervised learning approach called Graph Based Anomaly Detection (GBAD) which is used to detect anomalies in streams. Weekly data is considered as Streams.

William T. Young, et al. [7], uses domain knowledge to develop indicators, anomalies, and scenarios as starting points for analyzing and detecting susceptible insiders. They defined indicators as if any user activity causes any specific attack, then that activity will be considered as an indicator. They defined anomalies as unusual patterns of user behaviour and different log activities are considered as scenarios. They applied unsupervised anomaly detection (AD) algorithms to detect insiders based on the features derived from the previous indicators.

Owen Lo, et al. in [8] uses distance measurement to find the changes in user behaviour and then anomalous insiders. The three distance vector methods that they have used are Damerau– Levenshtein Distance, Cosine Distance, and Jaccard Distance. Duc C. Le, et al. in [9] uses both supervised and unsupervised algorithms on publicly available CERT datasets to detect malicious insiders. They used Self Organizing Maps (SOM) on the datasets and compared it to Hidden Markov Models (HMM) and C4.5 Decision Trees (DT). Duc C. Le, et al. in [10] [11] uses supervised ML techniques such as LR, RF & ANN on publicly available CERT dataset to detect new insider threat cases and considers the data as multiple levels of data granularity to detect malicious insiders and malicious activities. In [12], the researchers transform the security logs to text using Word2vec method to identify the behavioural probabilities. All these are detecting the abnormal behaviour of a user in their log activities not considering their job roles at their working place.

Few researchers [13] [14] [15] have considered the role group of a user to find the deviation score of that user. A. Legg, et al. [13] defined tree-structured profiles for individual user activity and combined role activity. These tree-structured profiles are used to assess how the user's current activities differ from his previous activities and with their peers. The variance of user behaviour from the previous behaviour is treated as deviation score of that user. Researchers of [14] did a sequential analysis using activity tree structure of user behavioural activities. It identifies whether the new activity belongs to the normal behaviour sequence in tree or malicious behaviour sequence of tree. The author in [15] uses a neural network model to do role-based classification of users by learning their behavioural patterns.

None of the above works are generating deviation score of a user and their level of threat severity. Clustering is the unsupervised techniques which can usually groups the entire data into clusters. Deviation score can be found in clustering technique as it clusters the data points based on the distance. But they cluster even an abnormal user to any one of the clusters whereas outlier detection techniques separate the abnormal users from the group of users [23][24][26][27]. But they are using a single threshold value for the entire dataset to find the outliers. That may lead to inaccurate results.

The objectives of the proposed work are:

- Partition the entire dataset into groups by user's job role in the organization.
- Find the threshold value for each group using N-Median distance plot.
- Labeling the outliers in every role group.
- Generating deviation scores of users to predict the possibility of insider threat in an organization.

III. Model to Find the Deviation Score of a User

From the literature review we observed that, most of the researchers have done their insider threat analysis using synthetic data which simulates the real data of the user activities in an organization. Due to the reputation and security concerns, organizations might not reveal their insider threat incidents and their user activities to the outside world. We did analysis of user activities on CERT insider threat dataset r6. 1, which includes 4000 user's activity log files, to produce an activity score of a user and his deviation score from other users within his allotted role group. The details of the datasets are mentioned in Table I.

We used Exploratory Data Analysis on the datasets to understand the correlation and significance of attributes of each dataset. We transformed the features from object type to numeric values before applying suitable algorithms.

TABLE I. DESCRIPTION ABOUT DATASETS

S. No	File	Features	Description
1	Logon	#id, #date, #user, #pc, #activity	gives the information about the user's logon/logoff activity
2	Device	#id, #date, #user, #pc, #filetree, #activity	Each record gives the information about the user's thumb drive connection to the system.
3	LDAP	#Userid, #role	Each record gives the information about the users and their roles in the organization

The system architecture in Fig. 1 shows the processing steps to find the deviation score of users whose behaviour is abnormal comparatively with the other users of the same group. The three datasets login, device & user-role are processed independently using data pre-processing and feature generation techniques to make numerical data ready for applying outlier detection techniques.

A. *Data Pre-processing*

Data Pre-processing is the initial step that every data analyst should perform before applying meaningful data analysis on the data. The main reason for doing data pre-processing is to understand and extract significant features of data [16].

B. *Feature Transformation*

CERT r6.1 dataset contain the raw data which will not give meaningful insight of the data. In a dataset, each represents the details of user activity in the organization like logon/logoff, device connect/disconnect, file open/close and email to within the group/outside the group. Features of those datasets are mapped to labels to count the activities or to apply the statistical analysis on the data. Each data set transformed and extracted features are mentioned in Table II.

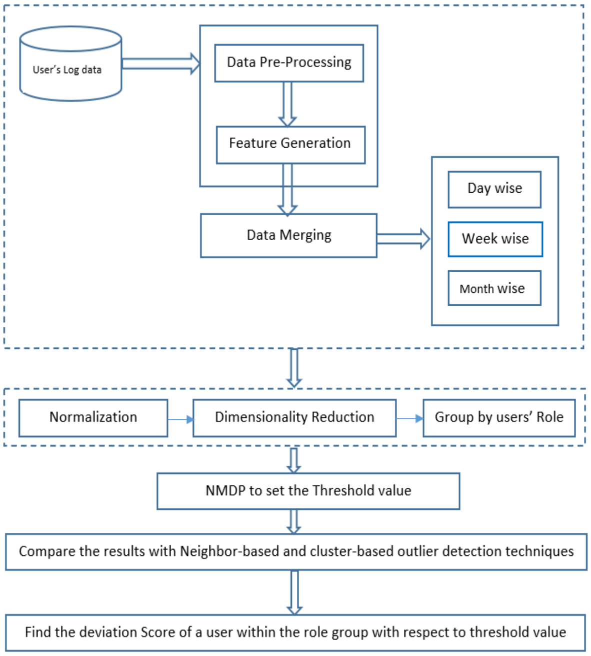

User's Log data
Data Pre-Processing
Feature Generation
Data Merging
Day wise
Week wise
Month wise
Normalization
Dimensionality Reduction
Group by users' Role
NMDP to set the Threshold value
Compare the results with Neighbor-based and cluster-based outlier detection techniques
Find the deviation Score of a user within the role group with respect to threshold value

Fig. 1. System Architecture to find the Deviation Score of a user within the Role.

TABLE II. FEATURE TRANSFORMATION

S. No	Name of the File	Transformed Features
1	Logon	#user, #day_only, #pc, #logon_time, #logoff_time, #time_duration
2	Device	#user, #day_only, #pc, #noof_activities, #filelength, #lastactivity_time
3	Role	#user, #pc, #role,

Feature transformation will not change the nature of data or relation between features of data; however, it will influence a lot towards the analytics. It is used to perform data analysis and then to find significant information from the data. We apply aggregate functions on the features by grouping the user activities on a daily basis or weekly basis or monthly basis. The features in the aggregated dataset are of two types.

1) Features that contain count of a particular value like number of PCs, number of late hours, number of logins and logouts, number of devices connected.

2) Features that contain statistical values like variance of user activities per a day, week, month and year.

To find the abnormal behaviour of a user in a week, the mean value of a total number of activities in a week will not reveal accurate behaviour of a user because if a user performs more activities in one day and zero activities in remaining days will give as normal behaviour. But he did malicious activity on the weekend. So, the variation or standard deviation of user activities will produce accurate results.

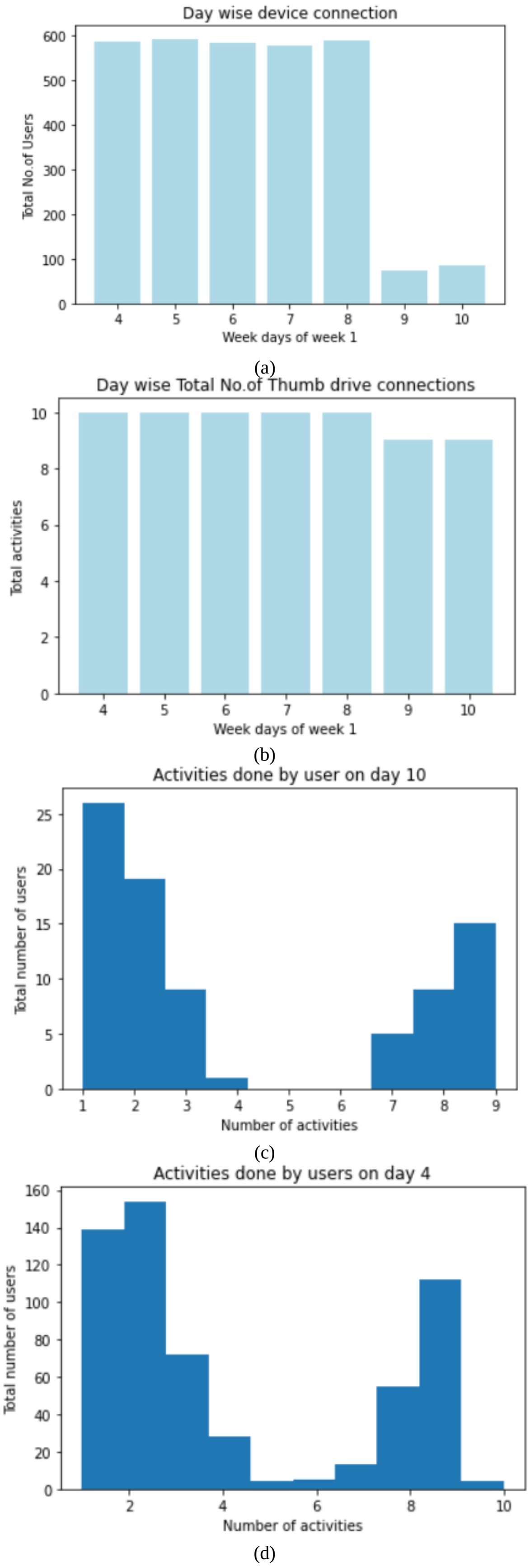
C. *Data Visualization*

Data visualization reveals a lot of insights in a dataset. We can understand CERT r6.1 datasets and relationships among the features in each dataset. We can visualize the log activities of users on a daily or weekly or monthly basis.

The bar plot is used to visualize the total number of users connected the thumb drives in a week, tells that, a smaller number of users who works on weekends were used thumb drives as shown in Fig. 2(a), the highest number of times device connectivity on a day is shown in Fig. 2(b), the distribution of activities on day10 and day4 is shown in Fig. 2(c) and Fig. 2(d).

We can also observe the variance of a user's behaviour in a week and how their activities are correlated with their variance in login time, numbers of PCs they used and the number of files in the file tree. Fig. 3 shows sample user behaviour in a week.

We can observe that the user has a different behavioural pattern, that is, few days he connected the external device a high number of times and few days he connected a smaller number of times. If we take average external drive connectivity, it may bias the truth. So, we need to transform the raw data into some standard form. Next section we discuss the standardization of data.



(a)

(b)

(c)

(d)

Fig. 2. Day Wise Activities of users.

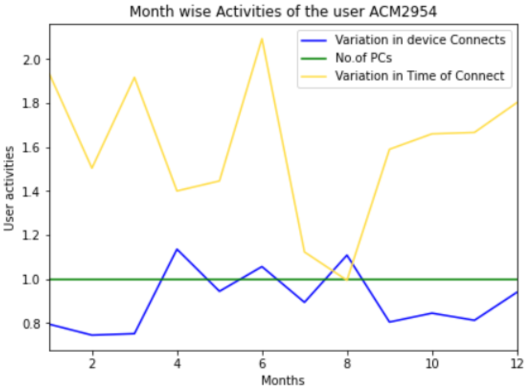


Fig. 3. Month Wise Activities of a user ACM2954.

D. Data Merging

To apply algorithms on the collected data, the data from different sources should be aggregated and transform their features as numerical vectors. We combined day-by-day sessions, days into weeks, and weeks into monthly log activities.

E. Data Normalization

The main objective of this paper is to produce the deviation score of a user from other users in an organization specific role. We are using clustering algorithms to group the users based on their behaviour in the log activities. All clustering techniques form clusters based on the distance computation and they are highly influenced by outliers. As shown in Fig. 2(c) and Fig. 2(d), CERT r6.1 dataset attribute value is not in Gaussian Distribution manner. Large-scaled features will dominate other features [16]. Therefore, for better results, we applied the Min-Max Normalization method before applying the clustering techniques.

Min-max normalization transforms every value in the feature column between the range [0 ,1]. The values will be transformed using the following formula

$$x_i^{'} = \frac{x_i - (F_i)}{(F_j) - (F_i)}$$

Where,

$x_i^{'}$ is transformed feature in the dataset,

x_i is value in the feature column F_i

(F_i) is minimum value in the feature column F_i

(F_j) is the maximum value in the feature column

The minimum value in the column will be transformed as 0 and the maximum value will be transformed as 1. All datasets of CERT r6.1 will be normalized according to the formula to avoid bias.

F. Dimensionality Reduction

As the CERT r6.1 dataset are unlabeled data, we find the deviation score of a user in his allotted role by grouping the users based on user's activity using clustering techniques. AK Jain et al. in [17] given that, feature selection & feature extraction are key steps to obtain the appropriate clusters. All clustering algorithms check the similarities between data points using distance measuring techniques to form groups or clusters. The prominent distance measuring technique is Euclidean distance.

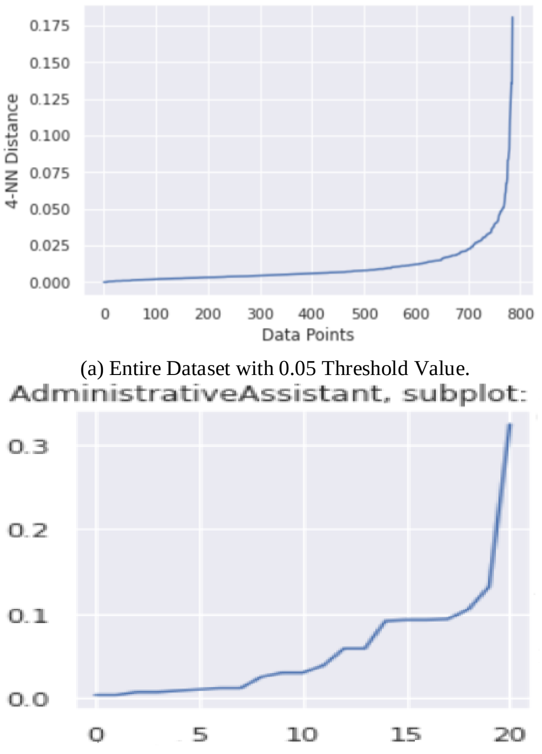
Let $U_i(d_{i1}, d_{i2})$ & $U_j(d_{j1}, d_{j2})$ are two user records, the Euclidean distance between these two user points is

$$dist = \sqrt{(d_{i1} - d_{j1})^2 + (d_{i2} - d_{j2})^2}$$

Clustering on a small number of dimensions would give better results than the large scaled dimensionality sample. According to [18] [19], dimensionality reduction reduces the computational load of clustering and also removes the redundant data. Our final dataset consists of nearly 15 features after merging the datasets and which leads to high dimensional problems. We used a popular linear technique called Principal Component Analysis (PCA) for dimensionality reduction without significant loss of information.

G. Group by users' Role

To identify whether the user is behaving normally or not, we need to have some threshold value for comparisons. If we set the threshold value for the entire dataset, global threshold, it will check every point with that threshold value and assigns few data points as outliers even if they are normal within their local groups. We partitioned the dataset into groups based on the user roles in the organization and assigns threshold value for each group individually based on their role behaviour. Fig. 4 shows the k-distance plot to estimate the optimal threshold value for finding the abnormality of the user behaviours.



(a) Entire Dataset with 0.05 Threshold Value.

(b) Administrative Assistant Group with 0.13 Threshold Value.

Fig. 4. K-Distance Plot to get Optimal Threshold Value.

H. Comparing Unsupervised Outlier Detection Techniques with NMOD

Unsupervised ML techniques are used to analyze the unlabeled data in the dataset. Our purpose of analyzing the unlabeled data is to identify the abnormal behaviour of users and their deviation score with respect to their role group. This problem is considering as the detection of outliers [20][21] in the dataset. Outlier detection finds the different patterns exist in the data and a data point is to be considered as an outlier if it shows the different pattern other than the defined one [22]. The outlier detection approaches are categorized as Statistical, distance-based, density-based and cluster-based. The outlier detection techniques were applied on individual role groups instead of applying on the entire dataset because the access privileges of a user are generated based on his/her role group in an organization.

Statistical Outlier detection techniques are good when the data is univariate and have pre-assumed distribution [27]. CERT r6.1 data sets are merged to find the behavioural patterns of users which is multi feature space. So, statistical outlier detection techniques alone are not applying in the proposed method on the dataset.

1) Neighborhood-based outlier detection: Distance-based Outlier detection identifies a data point as an outlier based on its distance from its k nearest neighbours. Euclidean distance is the distance function used in various neighbour-based outlier detection methods. According to [28], a data point is an outlier if it has less than k neighbours within the predefined distance R. Researchers in [25] defines an object is an outlier if the ratio of k-nearest neighbour distance of an object and the average distance among k-nearest neighbours is greater than 1. The author in [29] says that a point is an outlier if it has highest k^{th}-nearest neighbour distance when compare with all other data points. The author in [30] defines a point as an outlier if it has highest average distance to its k nearest neighbours.

The author in [31] proposed a distance-based outlier detection method to find the top n outliers in the large high dimensional dataset by considering the weights of the data points. Weight of a data point is the sum of the distances from its k-nearest neighbours. Researchers in [32] detects outlier detection solving set for the dataset which is also a distance-based outlier detection. The author in [33] identified outliers by finding the frequency of k-occurrences of a point in the k-NN list of all other data points. Like distance-based outlier detection, Density-based outlier detection methods are also neighbourhood-based detection methods. It finds the outliers based on the density estimation of each data point with respect to their neighbourhood's density distribution [34][35].

2) Cluster-based Outlier detection: Clustering techniques are unsupervised techniques used to group the data points into clusters based on the likenesses between the data points in terms of distance or density of data points. Cluster-based Outlier detection techniques treat a data point as an outlier if it does not belong to any of the cluster. Density Based Spatial Clustering of Applications with Noise (DBSCAN) is the unsupervised clustering algorithm [23] used to form the clusters based on the density-reachability and density-connectivity between the points. Minimum number of points/neighbours (MinPts) and the radius (Eps) are the main parameters to decide the density level, core points and border points. Objects with more than MinPts neighbours within this radius Eps considered to be a core point [24].

Our purpose of using this DBSCAN algorithm is to find the outlier points out of D points in the CERT r6.1 database. We identified 0.13 as the optimal Eps value for Administrative Assistant role group based on K-distance elbow plot as shown in the Fig. 4. It forms clusters and considers the data point that does not belong to these clusters as a noise point. It is not giving the percentage of deviation of a user from his normal behaviour with respect to their role group.

3) N-Median Outlier Detection (NMOD): As per the Hawakin's definition [26], a data point is an outlier if it deviates from the other data point. In our context, we considered any specific user is an outlier in the group if he/she is largely deviating from the other users of the same group. We are using a group wise threshold value to find whether the user is deviating from other the other users of the same role group.

The following are the steps to find the threshold distance value and outliers in the Role group G_i:

- Calculate the (*n* x *n)* Euclidian distance matrix for all data points of the dataset. Where n is the number of data points in a role group G_i.
- Find the median distance m_{P_i} for every point P_i in the matrix. Where each row is distances between a point P_i to each of the other points $\{P_1, P_2, P_3, \ldots\ldots P_n\}$ in a role group G_i.
- Plot a graph with those median values. The first raising edge's corresponding y value consider as the threshold distance TD_{G_i} value for the given role.
- Label a data point as an outlier if its median distance exceeds the threshold distance value.

4) Deviation score of a user in a Role Group: Deviation score is a value between 0 to 1 scale to identify the amount of deviation in the specified role group. If a user's deviation score is greater than 0, we considered them as insiders and role manager can estimate the causes of those insiders. This score helps the role manager to predict the insider threat possibility.

The following are the steps to calculate deviation score:

- Compare every point's median distance value *m* with the threshold value TD_{G_i}.
- Deviation score $DV_{P_i} = \begin{cases} m_{P_i} - TD_{G_i} & if\ m_{P_i} > TD_{G_i} \\ 0 & otherwise \end{cases}$

The complete procedure to find the deviation score of a user in the role group is listed as algorithmic steps in Algorithm1.

Algorithm 1. The pseudo code of outlier detection

```
Input:
  D: Dataset (Device& Log datasets)
  $G_i$: Role Group
Output:
  Label each user with deviation score
Algorithm
  Merge datasets into D
  for i in range (1,13): // Month wise
    Data standardization using MinMaxScaler
    Dimensionality reduction using PCA
    Partitioned the Data in to Role groups
    Call NMOD by passing role group $G_i$
    if (label = = -1)
      Df← Extract the User details
    end if
    Find the deviation score $DV_{P_i}$ of each user whose label is -1
  end for
  print outliers and their score
```

IV. Analysis of Result

We combined day-by-day sessions, days into weeks, and weeks into monthly log activities. Finally, we iteratively applied the kth-distance, DBSCAN, and NMOD techniques to 12 months of data to find the cluster groups and list of outliers. The proposed N-median outlier detection method outperforms the kth-distance and DBSCAN outlier detection methods. Eps and MinPts are hyper parameters in DBSCAN for clustering users, and k value is user specific for the kth-distance method to find outliers. The k-distance plot is used by both DBSCAN and kth-distance methods to determine their Eps and threshold value, as shown in Fig. 3. The proposed NMOD does not depend on k value and it gives the deviation score for each user in the scale of 0 to 1. Our main objective is to find the deviation score of users to predict the possibility of insider threat in the organization. Table III shows the list of outliers detected by Kth-distance, DBSCAN and NMOD in the role group Chief Engineer.

The listed users are identified as outliers in Chief Engineer group when Eps and Threshold value in both DBSCAN and Kth-distance is 0.12 and k/Minpts value is 3. In N-Median the threshold value is 0.35 from the N-Median distance plot shown in Fig. 5. If we change the Eps and Minpts values in DBSCAN, it will form the two clusters and noise points in the same role group. So, Eps and Minpts are hyper parameters in DBSCAN where as in NMOD it will not take any assumed values and produce the same results.

TABLE III. LIST OF OUTLIERS IN CHIEF ENGINEER ROLE GROUP

Userid	Role	DBSCAN	K-Distance	N-Median
KMP2999	ChiefEngineer	-1	-1	-1
PCJ2393	ChiefEngineer	-1	-1	-1
SCR0171	ChiefEngineer	-1	-1	-1
WCH0096	ChiefEngineer	-1	-1	-1

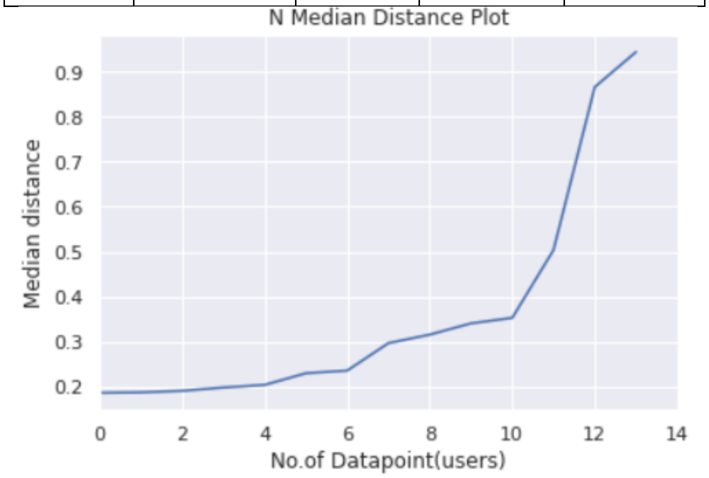


Fig. 5. N-Median Distance Plot.

If we change the k value in kth-distance method, it will generate different list of outliers as shown in Table IV whereas NMOD always gives the same result and there is no ambiguity in choosing the threshold value.

The deviation score of the outliers listed out in NMOD method is shown in the Table V. NMOD also label the users in the role as High (H), Medium (M), Low (L) users for the purpose of role manager to predict the insider threat possibility.

The role manager can use these values to take decision for distributing sensitive data with them in an organization or they can remove from the role group.

TABLE IV. LIST OF OUTLIERS IN CHIEF ENGINEER ROLE GROUP WHEN K=4

Userid	Role	DBSCAN	K-Distance	N-Median
DSL1727	ChiefEngineer	0	-1	0
GJF2381	ChiefEngineer	0	-1	0
KMP2999	ChiefEngineer	-1	-1	-1
PCJ2393	ChiefEngineer	-1	-1	-1
SCR0171	ChiefEngineer	-1	-1	-1
WCH0096	ChiefEngineer	-1	-1	-1
WJV3002	ChiefEngineer	0	-1	0

TABLE V. LIST OF OUTLIERS WITH THEIR DEVIATION SCORE & LEVEL

Index	Userid	Role	DevScore	Level
432	KMP2999	ChiefEngineer	0.154	M
585	PCJ2393	ChiefEngineer	0.594	H
668	SCR0171	ChiefEngineer	0.003	L
746	WCH0096	ChiefEngineer	0.516	H

V. CONCLUSION AND FUTURE WORK

The proposed N-Median outlier detection technique detects the users who exhibit the aberrant behavior when compared to other users of the same job role in an organization. It finds the threshold value using n-median distance plot for each role group. The method will compute the deviation score of each user with respect to their role's threshold value. If an employee deviates from their job role, the role manager must be notified. The deviation score enables the role manager to forecast the possibility of an insider threat in an organization. This approach additionally categorizes the people in the role as High (h), Medium (m), or Low (l) severity based on their deviation score. In this paper, the results of the proposed N-Median outlier detection techniques are compared with the results of kth-distance and DBSCAN outlier detection techniques. N-Median outlier identification does not rely on any of the k values used in kth-distance, nor does it construct numerous groups like DBSCAN does. In future work, we will leverage a user's deviation score in a role to create a framework for safe data delivery to an organization's users via Cloud servers.

REFERENCES

[1] Liu, L., De Vel, O., Han, Q. L., Zhang, J., & Xiang, Y. (2018). Detecting and preventing cyber insider threats: A survey. *IEEE Communications Surveys & Tutorials*, *20*(2), 1397-1417.

[2] https://www.fortinet.com/content/dam/fortinet/assets/threat-reports/insider-threat-report.pdf.

[3] J. P. Anderson, "Computer Security Threat Monitoring and Surveillance," James P Anderson Co, Fort Washington, Pennsylvania, Tech. Rep., April 1980.

[4] Aldairi, M., Karimi, L., & Joshi, J. (2019, July). A Trust Aware Unsupervised Learning Approach for Insider Threat Detection. In 2019 IEEE 20th International Conference on Information Reuse and Integration for Data Science (IRI) (pp. 89-98). IEEE.

[5] Glasser, J., & Lindauer, B. (2013, May). Bridging the gap: A pragmatic approach to generating insider threat data. In 2013 IEEE Security and Privacy Workshops (pp. 98-104). IEEE.

[6] Parveen, P., Evans, J., Thuraisingham, B., Hamlen, K. W., & Khan, L. (2011, October). Insider threat detection using stream mining and graph mining. In 2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing (pp. 1102-1110). IEEE.

[7] Young, W. T., Goldberg, H. G., Memory, A., Sartain, J. F., & Senator, T. E. (2013, May). Use of domain knowledge to detect insider threats in computer activities. In 2013 IEEE Security and Privacy Workshops (pp. 60-67). IEEE.

[8] Lo, O., Buchanan, W. J., Griffiths, P., & Macfarlane, R. (2018). Distance measurement methods for improved insider threat detection. Security and Communication Networks, 2018.

[9] Le, D. C., & Zincir-Heywood, A. N. (2018, May). Evaluating insider threat detection workflow using supervised and unsupervised learning. In 2018 IEEE Security and Privacy Workshops (SPW) (pp. 270-275). IEEE.

[10] Le, D. C., & Zincir-Heywood, A. N. (2019, April). Machine learning based insider threat modelling and detection. In 2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM) (pp. 1-6). IEEE.

[11] Le, D. C., Zincir-Heywood, N., & Heywood, M. I. (2020). Analyzing data granularity levels for insider threat detection using machine learning. IEEE Transactions on Network and Service Management, 17(1), 30-44.

[12] Liu, L., Chen, C., Zhang, J., De Vel, O., & Xiang, Y. (2019). Insider threat identification using the simultaneous neural learning of multi-source logs. IEEE Access, 7, 183162-183176.

[13] Legg, P. A., Buckley, O., Goldsmith, M., & Creese, S. (2015). Automated insider threat detection system using user and role-based profile assessment. IEEE Systems Journal, 11(2), 503- 512.

[14] Agrafiotis, I., et al., Towards a User and Role- based Sequential Behavioral Analysis Tool for Insider Threat Detection. Journal of Technology Transfer, 2014. 4(forthcoming): p. 127-137.

[15] Zhang, D., Zheng, Y., Wen, Y., Xu, Y., Wang, J., Yu, Y., & Meng, D. (2018, January). Role-based log analysis applying deep learning for insider threat detection. In Proceedings of the 1st Workshop on Security-Oriented Designs of Computer Architectures and Processors (pp. 18-20).

[16] Famili, A., Shen, W. M., Weber, R., & Simoudis, E. (1997). Data preprocessing and intelligent data analysis. Intelligent data analysis, 1(1), 3-23.

[17] Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: a review. ACM computing surveys (CSUR), 31(3), 264-323.

[18] Kambhatla, N., & Leen, T. K. (1997). Dimension reduction by local principal component analysis. Neural computation, 9(7), 1493-1516.

[19] Paukkeri, M. S., Kivimäki, I., Tirunagari, S., Oja, E., & Honkela, T. (2011, November). Effect of dimensionality reduction on different distance measures in document clustering. In International Conference on Neural Information Processing (pp. 167-176). Springer, Berlin, Heidelberg.

[20] Xu, R., & Wunsch, D. (2005). Survey of clustering algorithms. IEEE Transactions on neural networks, 16(3), 645-678.

[21] Gogoi, P., Bhattacharyya, D. K., Borah, B., & Kalita, J. K. (2011). A survey of outlier detection methods in

[22] Kriegel, H. P., Kröger, P., & Zimek, A. (2010). Outlier detection techniques. Tutorial at KDD, 10, 1-76.

[23] Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996, August). A density-based algorithm for discovering clusters in large spatial databases with noise. In Kdd (Vol. 96, No. 34, pp. 226-231).

[24] Schubert, E., Sander, J., Ester, M., Kriegel, H. P., & Xu, X. (2017). DBSCAN revisited, revisited: why and how you should (still) use DBSCAN. ACM Transactions on Database Systems (TODS), 42(3), 1-21.

[25] Zhang, K., Hutter, M., & Jin, H. (2009, April). A new local distance-based outlier detection approach for scattered real-world data. In Pacific-Asia Conference on Knowledge Discovery and Data Mining (pp. 813-822). Springer, Berlin, Heidelberg.

[26] Kriegel, H. P., Kröger, P., & Zimek, A. (2010). Outlier detection techniques. Tutorial at KDD, 10, 1-76.

[27] Zhang, Y., Meratnia, N., & Havinga, P. (2010). Outlier detection techniques for wireless sensor networks: A survey. IEEE communications surveys & tutorials, 12(2), 159-170.

[28] Knorr, E. M., & Ng, R. T. (1998, August). Algorithms for mining distance-based outliers in large datasets. In VLDB (Vol. 98, pp. 392-403).

[29] Ramaswamy, S., Rastogi, R., & Shim, K. (2000, May). Efficient algorithms for mining outliers from large data sets. In Proceedings of the 2000 ACM SIGMOD international conference on Management of data (pp. 427-438).

[30] Angiulli, F., & Pizzuti, C. (2002, August). Fast outlier detection in high dimensional spaces. In European conference on principles of data mining and knowledge discovery (pp. 15-27). Springer, Berlin, Heidelberg.

[31] Angiulli, F., & Pizzuti, C. (2005). Outlier mining in large high-dimensional data sets. IEEE transactions on Knowledge and Data engineering, 17(2), 203-215.

[32] Angiulli, F., Basta, S., Lodi, S., & Sartori, C. (2012). Distributed strategies for mining outliers in large data sets. IEEE transactions on knowledge and data engineering, 25(7), 1520-1532.

[33] Radovanović, M., Nanopoulos, A., & Ivanović, M. (2014). Reverse nearest neighbors in unsupervised distance-based outlier detection. IEEE transactions on knowledge and data engineering, 27(5), 1369-1382.

[34] Breunig, M. M., Kriegel, H. P., Ng, R. T., & Sander, J. (2000, May). LOF: identifying density-based local outliers. In Proceedings of the 2000 ACM SIGMOD international conference on Management of data (pp. 93-104).

[35] Papadimitriou, S., Kitagawa, H., Gibbons, P. B., & Faloutsos, C. (2003, March). Loci: Fast outlier detection using the local correlation integral. In Proceedings 19th international conference on data engineering (Cat. No. 03CH37405) (pp. 315-326). IEEE.