TECH-KNOWLEDGE JOURNAL

Insider Attacker Detection Using Light Gradient Boosting Machine

Mohammed A. Mohammed*, Suhad M. Kadhem, Maisa'a A. Ali
Department of Computer Science, University of Technology, Baghdad, Iraq
* mabdallazez4@gmail.com

Keywords: **Insider Attacker, Insider Threat, LightGBM, Detection, Security.**

ABSTRACT

Organizations security suffer from the insider attacker, which is an employee (person) with an authorized access to resources and data of an organization then used the access to harm the organization. In reality, the number of malicious events is very small in relation to the number of normal events of the employee, so it was necessary to use a method that accurately characterized this number of harmful behaviors. Several previous studies used complex methods such as deep learning to solve this problem. In this paper, we used a simpler and faster solution that gave accurate results, where an intelligent approach for detecting insider attacker using Light Gradient Boosting Machine (LightGBM) applied, the cert r4.2 data set used to build and evaluate the model. The results showed the model's ability to distinguish malicious events from data set in its original unbalanced state with accuracy 99.47%.

1.0. INTRODUCTION

The threat is no longer much outside of organizations, whose, firewalls are effective and it no longer targets computers and digital artifacts, which have become more secure; the threat is human which is internal. The human component of the information system constitutes an insider threat to the system's security. A threat that is found inside the organization itself, masters its processes, its firewall and its security policy, whether, they are intentional or accidental, malicious or not [1]. The resource of insider threat is insider attacker "Personnel with an authorized access to resources and data of an organization"[2]. Everyone has the potential do to harm, including your employees, people within the targeted organization who may be either malicious (deliberately seeking to do damage, commit theft, etc.) or inadvertent (careless, poorly trained, etc.); these are the most dangerous because they are already inside system defenses and have access to targeted assets [3]. The insider attacker may be active or passive, the active attacker performs physical operations that cause damage to the organization, while the passive attacker provides information Through what he sees, what he hears, and what he perceives to the opponents or enemies. However, insiders tend to remain hidden and use deceit for activities. One of most important challenges in cybersecurity is detect the inside attacker but how detect the insider attacker, this is the more challenge because in today's technological era the boundary between friend and rival is growing fuzzier [2]. Our motivations to deal with the insider attacker is a great threat that the insider causes to organizations, companies, banks and governments, as it leads to huge losses of money and lives in the cases of security organization. The problem we are trying to solve here is how detect the active insider attacker to avoid losses. In this paper, we propose a model for monitoring employee activities and distinguishing malicious events based on Light Gradient Boosting Machine (LightGBM), the model train and evaluate with r4.2 data set. This paper uses LightGBM framework for the first time to detect the insider attacker, we will notice that all previous

works use complex methods like deep learning. The reminder of this paper is organize as follows. Previous works will be discussed in Sect. 2. Section 3 illustrates the data description. The evaluation metric introduces in Sect. 4, followed by proposed model in Sect. 5. In Sect 6 results and discussion and finally conclusion in Sect 7.

2.0. PREVIOUS WORK

Fangfang Yuan in [4], presented an insider threat detection method with Deep Neural Network (DNN) based on user behavior. Specifically, the LSTM-CNN framework to find user's anomalous behavior. The LSTM with CNN gets best result AUC = 0.9449. Qiujian Lv et al [5], proposed a method for the detection of malicious insiders based on the analysis of both user and role behaviors. First, extract several temporal features for every user corresponding to different types of user behaviors. Then, the multiple features reflecting the deviation between the behavior of a user and that of the user group sharing the similar job role with him/her are then calculated. Those significant features, which influence the detection of insider threat significantly, are select by implementing a PCA method. Finally, an efficient detection model is design by leveraging the Isolation Forest Algorithm. They obtain 0.85% accuracy. Adam James Hall, and other in [6] uses the CERT dataset r4.2 along with a series of machine learning classifiers to predict the occurrence of a particular malicious insider threat scenario - the uploading sensitive information to wiki leaks before leaving the organization. These algorithms are aggregate into a meta-classifier, which has a stronger predictive performance than its constituent models. This meta-classifier has an accuracy of 96.2%. Andreas Nicolaou, and other in [7] they attempt to mitigate insider threat problem by developing a machine-learning model based on Bio-inspired computing. The model was developing by using an existing unsupervised learning algorithm for anomaly detection. Where they collected 50,000 samples for experimentation and divided them at rates 66% for training and 34% for testing, and the best result obtained after using optimization algorithms was TP= 91.4%. Minhae JANG and other in [8], they propose an anomaly-based insider threat detection with local features and global statistics over the assumption that a user shows different patterns from regular behaviors during harmful actions. For each user, they built and trained a seq2seq autoencoder model. The training data is the first 60 days of user behavior logs under the assumption that users act normally during this period.the best result obtained was AUC value of 0.9855. Xiaoyun Ye and other in [9], they used the CERT dataset r4.2 along with double-layer HMM structure to model user behavior. They use 50 insiders and obtain 99% accuracy, and they detect drawback in the system when they face the malicious behavior of users without any data accumulation, they can do nothing about the attack. Shuhan Yuan and Xintao Wu in [10], They mentioned deep learning and its relationship with insider attacker processing and a set of challenges and trends. Mehul S. Raval and other in [2], they mentioned Machine Learning (ML) for an insider threat detection, and some case studies on insider threat defense mechanism based on machine learning. There was not study that dealt with LightGBM to solve insider attacker problem as we presented.

3.0. BODY LANGUAGE

This section provides an overview of the CERT r4.2 dataset [11], which is used for our proposed method to detection of malicious users. Which contains relatively a lot of abnormal events compared to other revisions. A thousand of users generated about 32 million computer usage events during 17 months. The total number of threat events is 7,323. There are seven primary groups of files, which are generated from 1000 simulated users. A description on the contents of each file provided in Table 1; further details can be obtained from the CERT website. In terms of insider threats, version r4.2 of the dataset consists of three primary scenarios described as follows:

1) User who did not previously use removable drives or work after hours begins logging in after hours, using a removable drive, and uploading data to wikileaks.org and leaves the organization shortly thereafter.
2) User begins surfing job websites and soliciting employment from a competitor. Before leaving the company, they use a thumb drive (at markedly higher rates than their previous activity) to steal data.
3) System administrator becomes disgruntled, and downloads a key logger and uses a thumb drive to transfer it to his supervisor's machine. The next day, used the collected key logs to log in as his supervisor and send out an alarming mass email, causing panic in the organization. Leaves the organization immediately [12].

Table 1: Dataset details

Filename	Description
device.csv	Connection and disconnection of Removable devices (e.g., USB hard drive) is describe in this file.
email.csv	Contains logs of user emails.
file.csv	File access activity is provide in this file.
http.csv	This file record the url visited by each user.
logon.csv	Relates to user activity based on logging on and logging off on computing devices.
psychometric.csv	Provides personality and job satisfaction variables for each of the 1000 simulated users.
LDAP	This folder contains a set of LDAP files, which describe the ontology of each simulated user (their role, email, department, supervisor, etc.).

Our focus is on extrapolation of data from the files email.csv, device.csv, file.csv, http.csv and logon.csv. We have chosen to focus on the CERT 4.2 dataset as our data extrapolation methodology is derived from the fact that CERT r4.2 dataset contains a high number of insider threats (Compared with previous and later versions).

4.0. EVALUATION METRIC

To evaluate the performance, we used several typical measures extracted from confusion matrix, including accuracy, Recall, Precision and F1-score as shown in Table 2. According to the confusion matrix as mentioned in [13] [14], several measurements could be used for examining the performance of the model, the accuracy is usually determined by using the confusion matrix. The recall was use for determining the accuracy of every class known. Precision was also inaccurately classify using the equation below. This helped in calculating the F1 scores.

Table 2: Evaluation metric equations

Metric name	Equation
Accuracy	TP+TN/TP+TN+FP+FN
Recall	TP/TP+FN
Precision	TP/TP+FP
F1-score	2×(Recall× Precision)/(Recall+ Precision)

5.0. PROPOSED MODEL

The goal is to analyze the technical behavior of the employee, to detect malicious events, as shown in Figure 1. In this section, explain how the model trained and tested based on LightGBM framework and what data preprocessing give the best results.

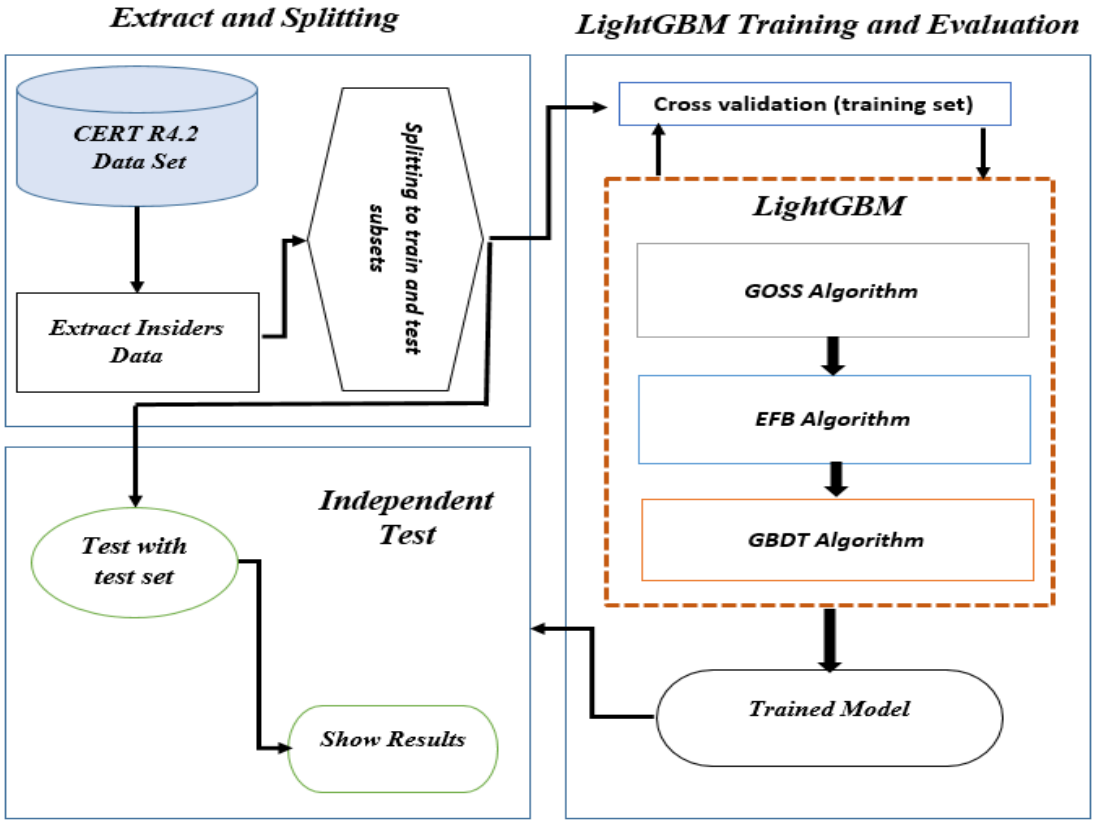

Fig. 1: The general proposed model.

The model consists of three main parts extract and splitting, LightGBM training with cross validation and independent test as explain in following sections:

5.1. Extract and Splitting

The dataset contains a thousand users (whose activities mentioned in the dataset part). Where their activities were record over a period of 17 months, which is an unbalanced dataset. Only 70 of 1000 users represent the insiders, the data of seventy insiders will be extract from the following files (device.csv, email.csv, file.csv, logo.csv and http.csv). Two types for split the dataset will be applied (percentage based and user based), percentage based used 80% for training and 20% for testing, user based used in total 70 users' where 50 users' for training and 20 users for testing.

As mentioned in previous study have been split dataset by using percentage value, this split-let user's behavior occurred in training and testing set. This is our justification for taking another type of division (user based) in this paper. Where, users in the training set have not the same users in the test set. This would be a realistic indication of the model's ability to distinguish as well give the model reliability and generalization to distinguish new users.

5.2. LightGBM Training and Evaluation

LightGBM algorithm used to training and testing a model to make it capable of distinguish malicious events as shown in the Figure 1. Cross validation used to increase the efficiency of the model and achieve the greatest possible accuracy, where it was use 5-Fold cross validation.

Gradient boosting decision tree (GBDT) is a useful algorithm that can be used for both classification and regression problems. Recently, Ke et al [15] proposed a novel gradient boosting decision tree (GBDT) algorithm named LightGBM, which utilize two novel techniques: Gradient-based One-Side Sampling (GOSS) along with Exclusive Feature Bundling (EFB) to deal with the huge number of data samples along with massive amount of features respectively as illustrated in Figure 2.

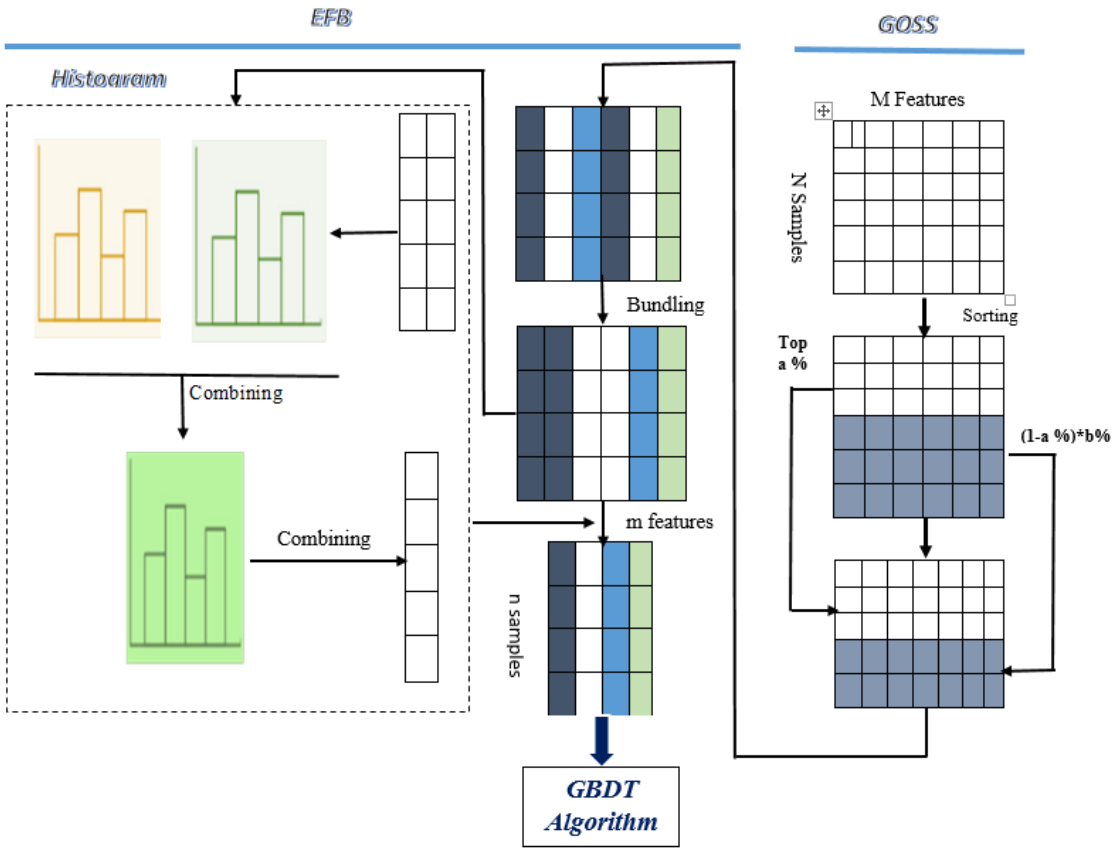

Fig. 2: Gradient-based One-Side Sampling (GOSS) along with Exclusive Feature Bundling (EFB).

GOSS keeps all the examples with large gradients and conducts random sampling on the examples with small gradients. EFB algorithm can bundle many exclusive characteristics to the much fewer dense characteristics, which can dramatically avoid unnecessary calculation for zero feature values. And so on these two algorithms deal with the huge number of data samples along with massive number of features.

The LightGBM algorithm can quickly process large amounts of data. It was developed as an open source project by Microsoft. The Light Gradient Boosting algorithm is explained in Figure 3.

The LightGBM algorithm includes several parameters, termed hyper parameters. The hyper parameters have a significant impact on the performance of LightGBM algorithm. They are typically set manually and then tuned in a continuous trial and error process.

The LightGBM algorithm

Input: Training data D = {(χ1, y1), (χ2, y2), ..., (χN, yN)}, χi_χ, χ ⊆ R, yi_{−1,+1}; loss function: L(y, _(χ)); iterations: M; sampling ratio of large gradient data: a; sampling ratio of small gradient data: b;

1. Merge mutually exclusive features(i.e. features never take nonzero values simultaneously) of χi, $i = \{1, ..., N\}$ by exclusive feature bundling(EFB) method;
2. Initialize $\_0(\chi) = arg\ minc\_Ni\ L(yi, c)$;
3. **for** $m = 1$ toM**do**
4. Compute absolute values of gradients:

$$ri = \left| \frac{\partial L(y_i, \theta(x_i)}{\partial \theta(x_i)} \right|_{\partial \theta(x) = \theta_{m-1(x)}}, i = \{1, \ldots \ldots, N\}$$

5. Resampled dataset by Gradient-based One-Side Sampling (GOSS) method:
   topN = a × len(D); randN = b × len(D);
   sorted = GetSortedIndices(abs(r));
   A = sorted [1 : topN]; B = RandomPick(sorted[topN : len(D)] , randN);
   D_ = A + B;
6. Compute information gains:

$$Vj(d) = \frac{1}{n}\left(\frac{\left(\sum_{x_i \in A_l} r_i + \frac{1-a}{b} \sum_{x_i \in B_l} r_i\right)^2}{N_l^j(d)} + \frac{\left(\sum_{x_i \in A_r} r_i + \frac{1-a}{b} \sum_{x_i \in B_r} r_i\right)^2}{N_r^j(d)} \right)^{nt}$$

7. Get a new decision tree $\theta_m(X)'$ on set D'
8. Update $\theta_m(X) = \theta_{m-1}(X) + \theta_m(X)'$
9. **end for**
10. **return** $\theta_M'(X) = \theta_M(X)$

Fig. 3: The LightGBM algorithm.

5.3. Independent Test

It is an independent test on data that the model has not previously seen, this test to measure the ability of the model to distinguish malicious activities to the user and to give greater reliability to the model.

6.0. RESULTS AND DISSECTION

All data processing tasks in this paper are perform using a PC with Intel Core. i5 2467M @ 1.60GHz CPU and 8.0 GB Dual-Channel DDR, the C# programming language used to paper Implementation.

6.1. Splitting Data and Class Distribution

Total events of the seventy insiders is 207440 events with five features (id, date, user, pc, activity) the class (1=200117 event, 0= 7323 event). Where, 0 is malicious event and 1is non malicious event.

Percentage based splitting is 80% for training and 20% for testing as shown in Table 3.

Table 3: Percentage based Splitting

class	1	0
training set	160100	5852
testing set	40017	1471

User Based Splitting, 50 users selected Randomly, their data extracted for training, and remainder 20 users extracted their data for testing as shown in Table 4.

Table 4: User based splitting

class	1	0
training set	116079	3670
testing set	34156	1498

6.2. Implement LlightGBM with Percentage Based Splitting

The results of training the model with training set and testing it with test set is shown in Table 5 and Table 6, respectively.

Table 5: Confused matrix of training lightgbm with training set (Percentage based).

	Predicted		
actual	**0**	**1**	**recall**
0	5.608	244	**0.9583**
1	80	160.020	**0.9995**
precision	**0.9859**	**0.9985**	

The confused matrix in Table 5 represent the results of the best model among five models of cross validation models. While, the average accuracy of the five models was 99.3% and average F1Score was 97.19%.

Table 6: Confused matrix of test lightgbm with test set (Percentage based).

	Predicted		
actual	**0**	**1**	**recall**
0	1.311	160	**0.8912**
1	60	39.957	**0.9985**
precision	**0.9562**	**0.9960**	

The confused matrix in table VI represent the results of the test the model with test data. While, the accuracy on test data was 99.47%, the Auc was Auc 99.79% and F1Score was 92.26%.

6.3. Implement Lightgbm with User Based Splitting

The dataset is splitting here on the basis of the user. Where the test set contains users who are not in the training set. The results of training the model with training set and testing it with test set is shown in Table 7 and Table 8, respectively.

Table 7: Confused matrix of training lightgbm with training set (user based splitting).

	Predicted		
actual	**0**	**1**	**recall**
0	3.635	35	**0.9905**
1	11	116.068	**0.9999**
precision	**0.9970**	**0.9997**	

The confused matrix in Table 7 represent the results of the best model among five models of cross validation models, That trained on data splitted based on the user. While, the average accuracy of the five models was 99.8% and average F1Score was 96.7%.

Table 8: Confused matrix of test lightgbm with test set (user based splitting).

	Predicted		
actual	**0**	**1**	**recall**
0	838	660	**0.5594**
1	44	34.112	**0.9987**
precision	**0.9501**	**0.9810**	

The confused matrix in Table 8 represent the results of the test the model on testset of 20 users the model has not seen before. While, the accuracy on test data was 98.03 %, the Auc was Auc 97.43% and F1Score was 70.42%.

6.4. Comparison Between Percentage Based and User Based

The comparison was made on the results of the test group for each of the two divisions as shown in the Table 9.

Table 9: Comparison between percentage based and user based

matric	**Percentage based**	**User based**
accuracy	99.47 %	98.03 %
Auc	99.79 %	97.43 %
F1 score	92.26 %.	70.42 %.

As it is clear from the table IX that the percentage based splitting is more accurate than user based splitting, the reason for this is that the behavior that was distinguished in the test set belongs to the same users in the training set. The accuracy in the case of user based splitting is more realistic because the users in the test set have not seen the model before and this corresponds to the situation of the new employee, which we want to find out if he is an insider attacker or not.

7.0. COMPARISON WITH PREVIOUS STUDIES

All previous studies have focused on the use of complex methods such as deep learning, and have dealt with data in a manner that does not suit the important nature of the internal attacker. The Table 10 shows the method of splitting the data in each work with some measurements for comparison.

Table 10: Comparison with previous studies

paper	splitting	accuracy	AUC	F1-score	TP
This work	80%-20% randomly	**99.47 %**	**99.79 %**	**92.26 %.**	-
	50 user training-20 users testing	**98.03 %**	**97.43 %**	**70.42 %.**	-
[4]	~70%-~30%	-	94.49%	-	-
[5]	Basd on user's time	85%	-	-	-
[6]	Use 7260 instances only	96.2%	-	-	-
[7]	66%-34%	-	-	-	91.4%
[8]	Basd on user's days	-	98.55%	-	-
[9]	Basd on user's time	99 %	-	-	-

We note that this work is distinguished by the fact that it adopted two divisions, one of which was tested on 20 users that the model had not seen before, and this did not happen in any of the previous works, in addition to using LightGBM algorithm as it was not used in any of the previous works.

When the behavior belongs to the same user in both the training and testing sets, the identification of the malicious events becomes more clear in this case the model give accuracy 99.47 %. While, when we test the behavior of new users that the model has not seen during the training, the result becomes more realistic, reliable and generlization in this case the model give accuracy 98.03 %., and this is because in the real world, the organizations want to discover new employees if they are insiders or not because the new employee we do not have previous data about him. Also, when using the model in a specific institution, it must be able to detect insiders from its employees, even if it is not trained on data belonging to them.

8.0. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed the insider threat detection model by use Light Gradient Boosting Machine (lightgbm). Because insider threat manifest in various forms, it is not practical explicitly model it. We frame insider threat detection as classification task based on events performed by employee.

the security of many organizations, banks and governaments suffer from the insider attacker, which is an employee with an authorized access to information of an organization then used the access to damage the organization. In reality, the malicious events is very little in relation to the normal events of the employee, so it was necessary to use a method that accurately distinguish this harmful behaviors. Several previous studies used complex methods such as deep learning to solve this problem. we used a simpler and faster solution that gave accurate results, where an intelligent approach for detecting insider attacker using (LightGBM) applied, the cert r4.2 data set used to trainining and test the model. Where two types of division were adopted (percentage based splitting and user based splitting) . The

results showed the model's ability to distinguish malicious events from data set in its original unbalanced state with accuracy 99.47 % In case and 98.03% in case of user based.

Lightgbm algorithm bypassed the most important problem for the attacker's data was an imbalance, As it give high accuracy in detect the malicious events and it is less complexity compared with other method.

In the future, we aspire to increase the accuracy of detection of harmful events in the case of user based splitting.

REFERENCES

[1] Pierre-Emmanuel Arduin,"Insider Threats", Volume 10, © ISTE Ltd 2018.

[2] Mehul S. Raval, Ratnik Gandhi, and Sanjay Chaudhary ,"Insider Threat Detection: Machine Learning Way", © Springer Nature Switzerland AG 2018.

[3] J. M. Borky, T. H. Bradley ," Protecting Information with Cybersecurity", © Springer International Publishing AG, part of Springer Nature 2019.

[4] Yanan Cao, "Insider Threat Detection with Deep Neural Network ", International Conference on Computational Science 2018.

[5] Qiujian Lv, Yan Wang, Leiqi Wang and Dan Wang, "Towards A User and Role-Based Behavior Analysis Method for Insider Threat Detection", Proceedings of IC-NIDC ©IEEE, 2018.

[6] Adam James Hall, Nikolaos Pitropakis, William J Buchanan and Naghmeh Moradpoor, "Predicting Malicious Insider Threat Scenarios Using Organizational Data and a Heterogeneous Stack-Classifier" , arXiv:1907.10272v1 [cs.CR] 24 Jul 2019.

[7] Andreas Nicolaou , Stavros Shiaeles and Nick Savage , "Mitigating Insider Threats Using Bio-Inspired Models" , Appl. Sci. 2020, 10, 5046; doi:10.3390/app10155046.

[8] Minhae JANG, Yeonseung RYU and Jik-Soo KIM, "Against Insider Threats with Hybrid Anomaly Detection with Local-Feature Autoencoder and Global Statistics (LAGS)", Copyright c_ 2020 The Institute of Electronics, Information and Communication Engineers.

[9] Xiaoyun Ye , Sung-Sam Hong , and Myung-Mook Han, "Feature Engineering Method Using Double-Layer Hidden Markov Model for Insider Threat Detection", International Journal of Fuzzy Logic and Intelligent Systems, Vol. 20, No. 1, March 2020, pp. 17-25http://doi.org/10.5391/IJFIS.2020.20.1.17.

[10] Shuhan Yuan and Xintao Wu, "Deep Learning for Insider Threat Detection: Review, Challenges and Opportunities", arXiv:2005.12433v1 [cs.CR] 25 May 2020.

[11] https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=508099.

[12] Owen Lo, William J Buchanan, Paul Griffiths and Richard J Macfarlane , "Distance Measurement Methods for Improved Insider Threat Detection" , Edinburgh Napier University, w.buchanan@napier.ac.uk, Academic Editor: Gerardo Pelosi Copyright © 2017.

[13] Dhafar Hamed Abd , Ahmed T. Sadiq, and Ayad R. Abbas, "Political Articles Categorization Based on Different Naïve Bayes Models" , © Springer Nature Switzerland AG 2020, M. I. Khalaf et al. (Eds.): ACRIT 2019, CCIS 1174, pp. 286–301, 2020.

[14] Dhafar Hamed Abd , Ahmed T. Sadiq, and Ayad R. Abbas, "Classifying Political Arabic Articles Using Support Vector Machine with Different Feature Extraction" , © Springer Nature Switzerland AG 2020, M. I. Khalaf et al. (Eds.): ACRIT 2019, CCIS 1174, pp. 286–301, 2020. https://doi.org/10.1007/978-3-030-38752-5_23.