IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

DETECTION OF INSIDER THREAT FORMING USING MACHINE LEARNING

[1]Prof. Megha Zala

[1]Assistance Professor
[1]Information and Technology
[1]Parul University, Gujarat, India

Abstract: Malicious insider threat attacks signify one of the most destructive threats to networked systems of establishments and companies. There is a unique set of challenges that come with insider threat detection in term of hugely unbalanced data, limited ground truth, as well as behavior drifts and shifts. In this research, we study and evaluate an insider threat detection workflow using supervised learning algorithms. To this end, we study on multiple levels of granularity under realistic condition for identifying not only malicious behaviors but also malicious insider. We evaluate several supervised learning algorithm LR, MN, RF and XG using this workflow. Detailed analyses of widespread insider threat scenarios with different performance measures are presented to facilitate the realistic estimate of system performance. Evaluation results show that the machine learning based detection system can learn from limited ground truth and detect new malicious insider in unseen data with a high accuracy.

Index Terms **- Malicious, insider threat, Supervised learning, ground truth.**

I. INTRODUCTION

An insider threat is a malicious threat to an organization that comes from people within the organization, such as employees, former employees, contractors or business associates, who have inside information concerning the organization's security practices, data and computer systems. [5] The threat may involve fraud, the theft of confidential or commercially valuable information, the theft of intellectual property, or the sabotage of computer systems. [7] Insider threat detection approaches can be categorized based on the data source into host-based user profiling, sensors network-based, contextual data-based analytics and integrated approaches. [7] All these approaches aim to build an understanding of an insider's activity patterns, and the intent of the attack so as to prevent any damage due to a security breach. [7]

Categories of Insider Threat

Malicious insiders, which are people who take advantage of their access to inflict harm on an organization.
Negligent insiders, which are people who make errors and disregard policies, which place their organizations at risk.
Infiltrators, who are external actors that obtain legitimate access credentials without authorization.

The actual process of behavior analysis, threat detection, categorization and risk scoring can be a complex endeavor depending on what machine learning algorithms are used. However, a common approach used by many solutions is 'anomaly detection', also known as 'outlier detection'. The idea is: a user's behavior should match with the rest in their group or past activities, called a baseline. Events or observations that deviate from this baseline are an anomaly. Typically, such an anomaly might be an indicator of fraud, sabotage, collusion, data theft or other malicious intent. Once an early deviation is detected, the algorithm can flag the incident for further investigation or if designed to do so, compare the incident with similar events recorded in the past. This record(s) could be the result of a previously executed Supervised algorithm where the anomalies were labeled as 'normal' or 'abnormal' by a human security analyst, acquired from previous training data or a crowd-sourced knowledgebase (for example, multiple customers sharing a threat intelligence database). Finally, the threat is reported with a risk score factoring in the frequency, resources involved, potential impact, number of nodes it's affecting and other variables.

Here are some basic steps and process a machine learning system might go through to detect insider threats:

Data mining input: The first step in machine learning involves getting the user behavior and entity datasets, i.e. the monitored objects like apps/websites, email, file system, network, meta data such as time of monitoring, user roles/access levels, content, work schedule etc. The more granular the data is the better the accuracy of the system.

User profiling: Information such as user roles, department/groups, access levels etc. are fed into the system from the employee records/HR systems, Active Directory, system audit logs, slice and dice data and other sources. This can be utilized for personalized profiling in the behavior models or integrated with an access control and privilege management system later.

Behavioral model(s): Different algorithms such as, Feature Extraction, Eigen-Value Decomposition, Density Estimation, Clustering etc. are used to generate behavior models. Sometimes specialized statistical/mathematical frameworks are adapted for this purpose. For example, Regression-based models can be used to predict future user action or to detect credit card frauds. Whereas, a Clustering algorithm can be used to compare business processes with compliance objectives.

Optimizing baselines: Once the behavior model generates a baseline, it can be fine-tuned for specific purposes. For example, adding a time or frequency component to trigger different rules at different levels of deviation, assign risk scores etc. Additional layers of filtering can also be used to increase efficiency of the algorithm and reduce false positives. For example, adding a domain filter to website anomalies to limit the number of incidents they system needs to check. In most cases, such baselines can be customized for individual, group/department or at organizational levels.

Policies and rules integration: Behavior baselines are used to identify threats and trigger alerts when something out of the ordinary happens. Some of the employee monitoring/UEBA/DLP combines these baselines with a policy and rules engine to proactively prevent threats. The engines support actions such as: warning the user, blocking an action, notifying admin, running specific commands or recoding the incident to facilitate forensic investigation.

Human feedback: At the end of the day, no matter how good a machine learning system is, it will still make mistakes, generate false positives or fail to identify a threat. After all, modeling human behavior is beyond the reach of any current technology. So, a security analyst will need to take the output from the machine learning system and conduct threat assessment manually from time to time. The good news is, these systems are designed to be responsive to human input. With enough human training, the system can be improved requiring less and less intervention over time.

Anomaly Detection

Anomaly detection is the identification of rare items, events or observations which raise suspicions by differing significantly from the majority of the data. Typically the anomalous items will translate to some kind of problem such as bank fraud, a structural defect, medical problems or errors in a text. Anomaly detection is heavily used in behavioral analysis and other forms of analysis in order to aid in learning about the detection, identification and prediction of the occurrence of these anomalies. Anomaly detection is also known as outlier detection.

Supervised Learning

Supervised learning is when the model is getting trained on a labeled dataset. Labeled dataset is one which has both input and output parameters. Supervised learning is indicates the presence of the supervisor as a teacher. It is learning in which we teach or train the machine using data which is well labeled that means some data is already tagged with the correct answer. After that, the machine is provided with a new set of examples (data) so that supervised learning algorithm analyses the training data (set of training examples) and produces a correct outcome from labeled data.

II. PROPOSED METHOD

The principal interest of this proposed architecture of this research is to assess the capability of Machine learning techniques in detecting insider threats in a corporate and organizational network. To this this end, the workflow for applying ML techniques is presented and designed to be modular and easily expandable for a wide range of corporate environments, data acquisition conditions, as well as learning and analysis method. The data collection and pre-processing steps, where features are constructed and different level of granularity are defined.

Figure: 1 proposed architecture

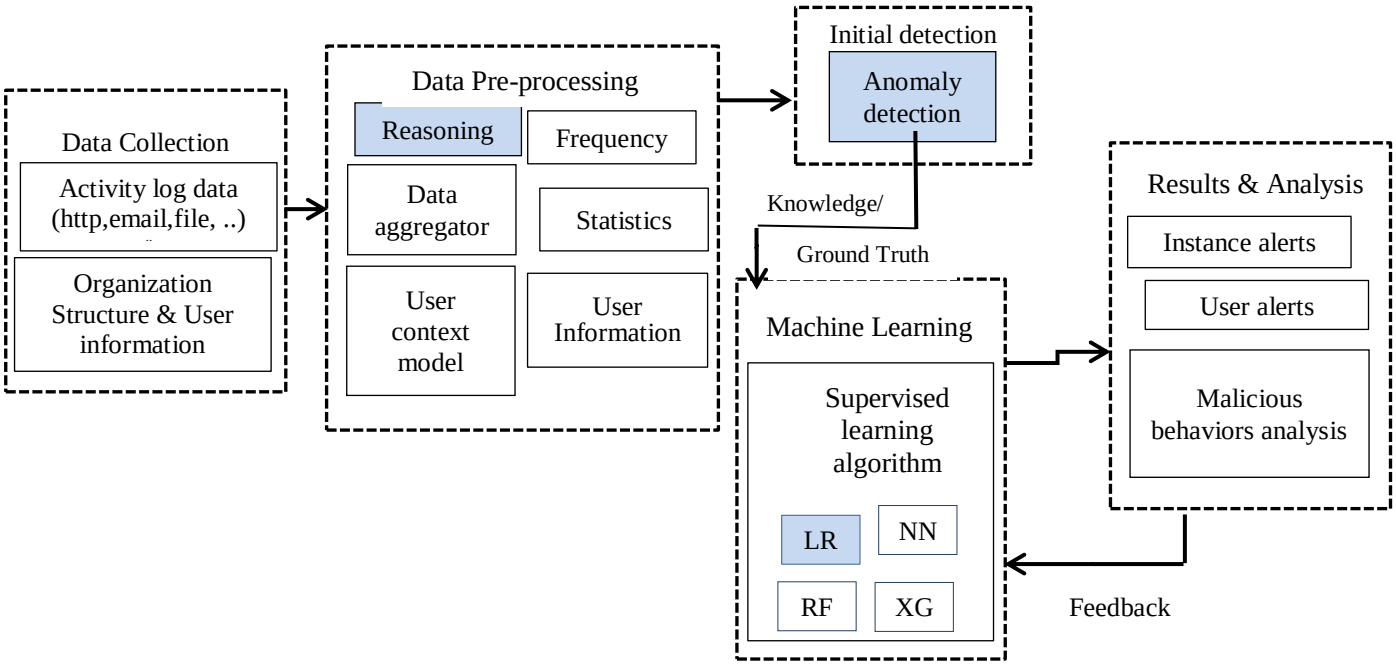
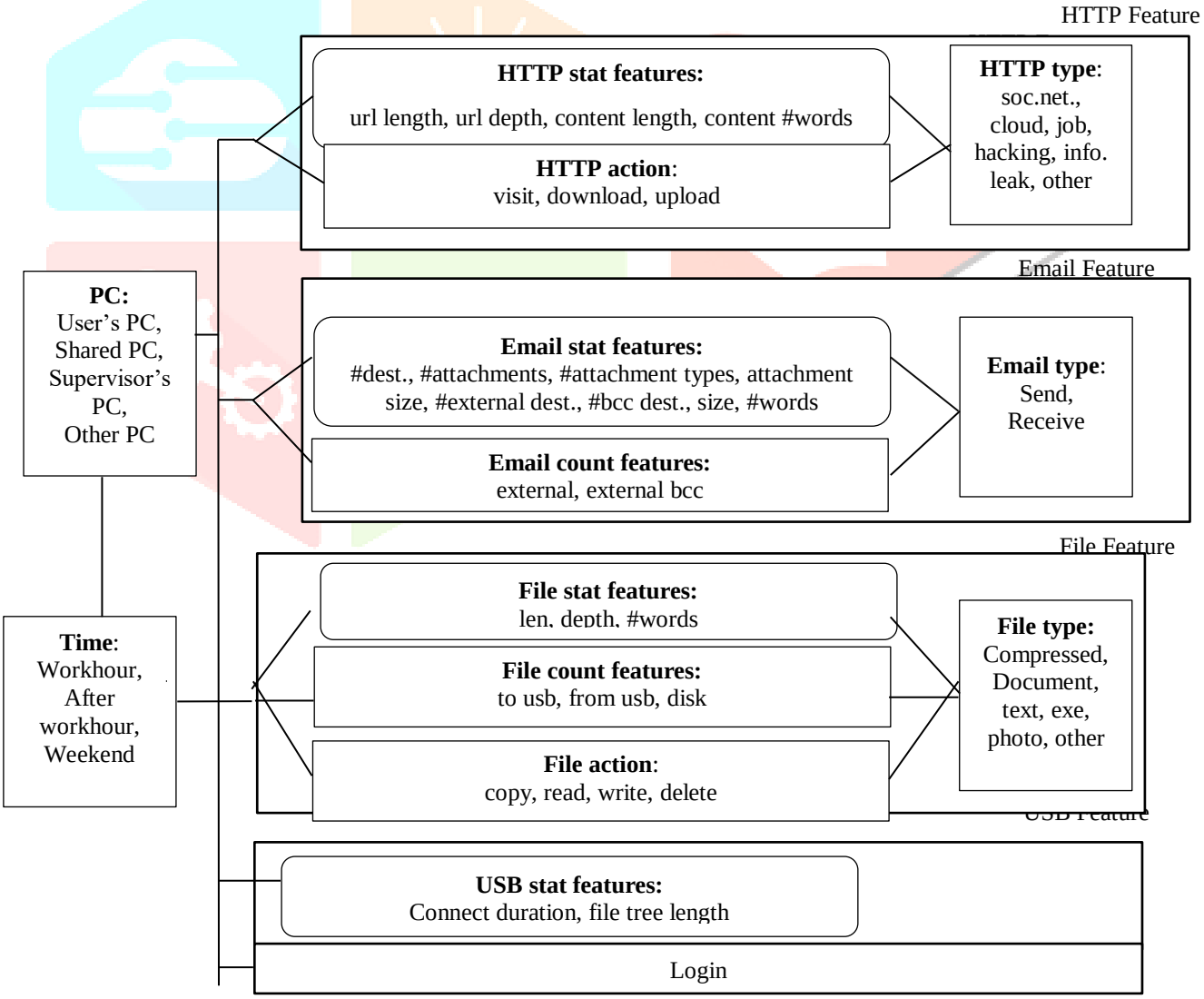


Figure: 2 Feature Extraction Process



Scikit-learn

Scikit-learn is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy. [12]

Simple and efficient tools for data mining and data analysis. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means, etc.

CERT Dataset

A computer emergency response team is a historic term for an expert group that handles computer security incidents. "'CERT"' should not be generically used as an acronym for this term as it is registered as a mark in the United State Patent and Trademark Office as well as in other jurisdictions around the world. [13]

Figure: 3 CERT Dataset

Filename	Description
device.csv	Connection and disconnection of removable devices (e.g., USB hard drive) is described in this file.
email.csv	Contains logs of user emails.

III. IMPLEMENTATION AND RESULTS

Performance Metrics:

Detection Rate: - TP/ (TP + FN)
False Alarm Rate: - FP/ (FP+TN)
Precision: - TP/ (TP+ FP)
F1 Score: - 2* ((Precision * Recall) / (Precision + Recall))
Where, TP: - True Positive Rate, TN: - True Negative Rate, FP: - False Positive Rate, FN: - False Negative Rate

Calculation of Detection Rate, False Positive Rate, Precision & F1 Score

Total dataset: - 10000
True Positive: - 8000, True Negative: - 1000, False Positive: - 600, False Negative: - 400

Implementation:
Detection Rate: - TP/ (TP + FN)
= 8000/ (8000+400)
= 8000/ 8400
= 0.9523
Detection Rate = 95.23%
False Alarm Rate: - FP/ (FP+TN)
= 600/ (600+1000)
= 600/ 1600
False Alarm Rate = 0.375
Precision: - TP/ (TP+ FP)
= 8000 / (8000+600)
= 8000 / 8600
= 0.9302
= 93.02 %
F1 Score: - 2* ((Precision * Recall) / (Precision + Recall))
= 2*((93.02 * 95.23) / (93.02 + 95.23))
= 2*((8858.2946) / (188.25))
= 2* 47.056
F1 Score = 94.11%

Result:

Figure: 4 Instance Based Result on User-week Data type

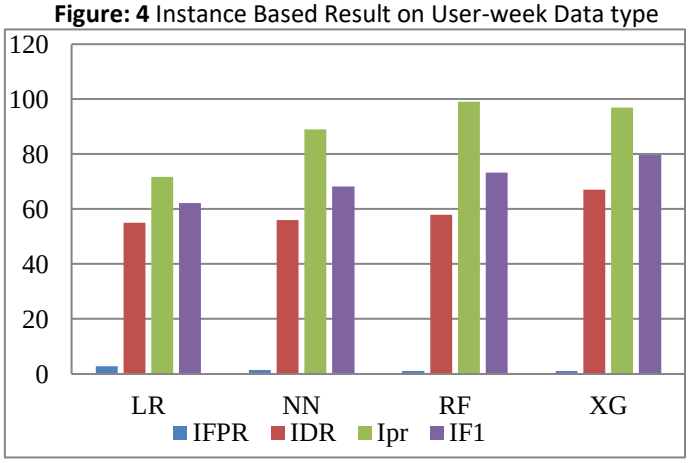


Figure: 5 Instance Based Result on User-day Data type

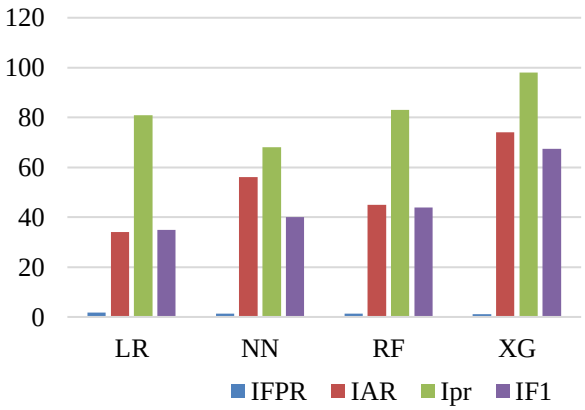


Figure: 6 Instance Based Result on User-Session Data type

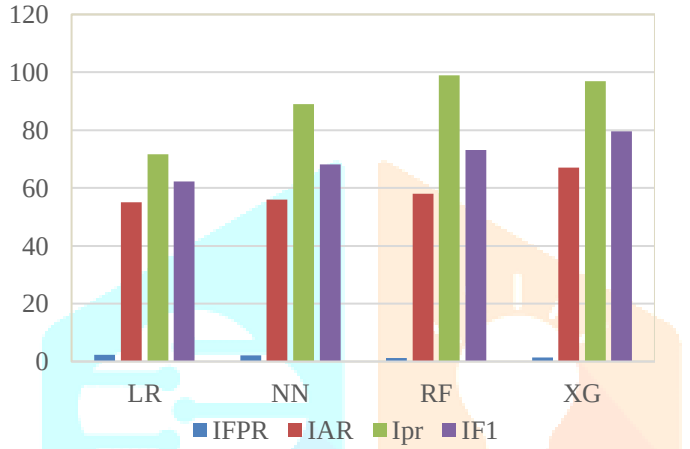


Figure: 7 User Based Result on User-week Data type

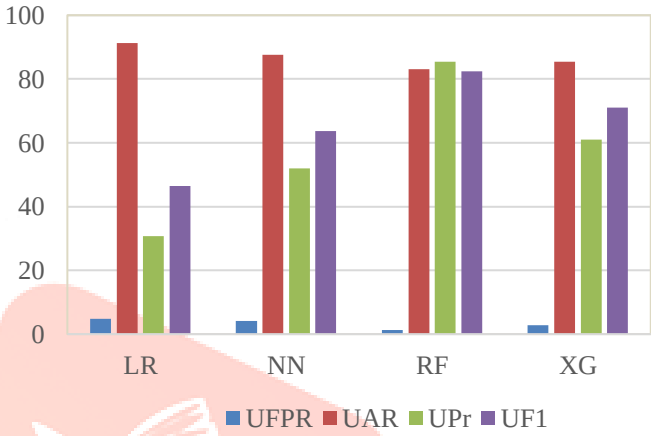


Figure: 8 User Based Result on User-Day Data type

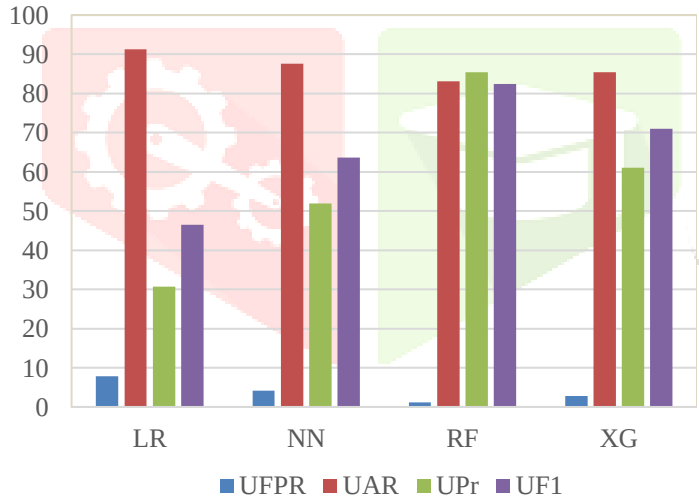
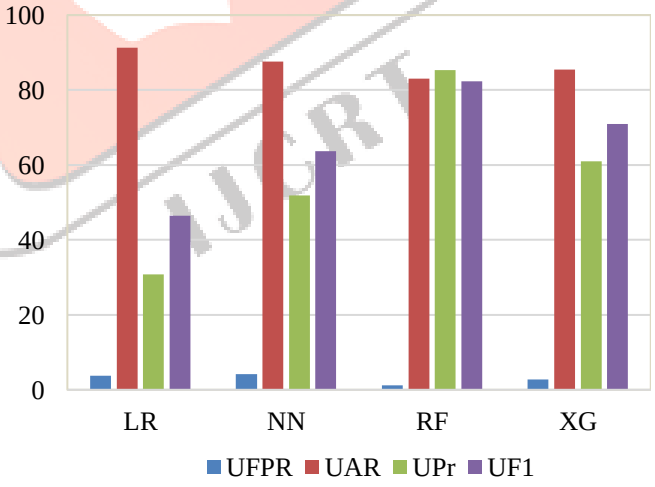


Figure: 9 User Based Result on User-Session Data type



As per above result figure RF Presents the best results in term of F1- score, Precision and False, Positive Rates in both case User based and Instance based metrics. NN shows promising UDR about 4% higher than RF .LR suffers from high UFRP and low F1-scores. XG achieves the best performance (IF1) on user-week data.

Figure: 10 Scenario Specific Test Result: DR and Average Detection Delay

Data Type	Algorithm	Metric	Scen 1 10 Users	Scen 2 20 Users	Scen 3 30 Users
User- Session	LR	UDR	100%	61.25%	100%
		Delay	0	6.16	4.25
	NN	UDR	100%	30.36%	97.35%
		Delay	0	5.5	1.35
	RF	UDR	100%	50.00%	91.43%
		Delay	0	7.47	0.45
	XG	UDR	100%	62.45%	99.74
		Delay	0	4.12	2.17
User- Day	LR	UDR	100%	70%	100%
		Delay	0.45	4.59	2.13
	NN	UDR	100%	47.54%	100%
		Delay	0.35	3.16	3.20
	RF	UDR	100%	67.50%	94%
		Delay	0.44	4.93	2.99
	XG	UDR	100%	78.54%	100%
		Delay	0.49	6.51	3.04
User- Week	LR	UDR	100%	73.75%	99.17%
		Delay	0	1.24	1.24
	NN	UDR	100%	58.75%	95%
		Delay	0.02	1.62	1.28
	RF	UDR	100%	15.98%	96.67%
		Delay	0.04	1.6	2.18
	XG	UDR	100%	84.75%	94.29%
		Delay	0.05	2.06	0.45

The unit detection delay depends on the data type. For example, on user session data, detection delay of 6 means that the malicious insider is detected after 6 sessions from the first malicious action. Above figure shows results on each scenario by DR and detection delay for malicious insider detected. Detection delay is defined as the delay from the time of the first malicious action to the time when the insider is detected. It is apparent that scenario 1 is the easiest to detect, where nearly all classifiers can detect them with 100% rate and very low delay.

IV. CONCLUSION AND FUTURE WORK

Insider threat is one of the most serious for many organizations. We present a machine learning based system for insider threat detection. LR, NN, RF, XG supervised learning algorithms are trained on limited ground truth to detect the malicious insider behaviours on unseen data. Evaluation results show proposed system is able to successfully learn from the limited training data and generalize to detect new users with malicious behaviours. The system achieves a high detection rate and precision, especially when user based result are considered.

More sophisticated data pre-processing techniques as well as feature analysis can be used to improve system performance.

V. REFERENCES

[1] Malek Ben Salem, Shlomo Herhkop, Salvatore J. Stolfo, "A Survey of Insider Attack Detection Research", Insider Attack and Cyber Security pp. 69-90, 2019.
[2] David A. No ever, Sr. Technical Fellow, "Classifier Suites for Insider Threat Detection", pp. 1-11, 2018.
[3]T.O. Oladimeji, C.K. Ayo, S.E. Adewumi, "Review on Insider Threat Detection Techniques", 3rd International Conference on Science and Sustainable Development (ICSSD), pp. 1-10, 2019.
[4] Ioannis Ayrafiotis, Jason RC Nurse, Oliver Buckley, Phil Legg. Sadie Creese, Michael Goldsmith, "Identifying Attack Patterns for Insider Threat Detection",Computer Fraud & Security, pp. 9-17, 2015.
[5] Mehul S. Raval, Ratnik Gandhi, Sanjay Chaudhary, "Insider Threat Detection: Machine Learning Way", Versatile CyberSecurity, Advances in Information Security,Springer, pp. 19-53, 2018.
[6] Duc C. Le, A. Nur Zincir-Heywood, "Machine learning based Insider Threat modelling and Detection", Workshop on Security for Emerging Distributed Network Technologies(DISSECT), IEEE, no. 05, pp. 1-6, 2019.
[7] Maryam Aldairi, Leila Karimi, James Joshi, "A Trust Aware Unsupervised Learning Approach for Insider Threat Detection", 20th International Conference on Information Reuse and Integration for Data Science (IRI), IEEE, no. 20, pp. 89-98, 2019.
[8] Chih-Hung Hsieh, Chia-Min Lai, Ching-Hao Mao, Tien-Cheu Kao, Kuo-Chen Lee,"AD2 : Anomaly Detection on Active Directory Log Data for Insider Threat Monitoring" 49th Annual International Carnahan conference on Security Technology, IEEE, no. 49, pp. 287-292, 2015.
[9] Duc C. Le, A. Nur Zincir-Heywood, "Evaluating Insider Threat Detection Workflow Using Supervised and Unsupervised Learning", Symposium on Security and Privacy workshops, IEEE, pp. 270-275, 2018.
[10] Duc C. Le, A. Nur Zincir-Heywood, "Analyzing Data Granularity Levels for Insider Threat Detection Using Machine Learning", IEEE, pp. 1-14, 2019.
[11] Guang Yang, Lijun Cai, Aimin Yu, Dan Meng, "A General and Expandable Insider Threat Detection System Using Baseline Anomaly Detection and Scenario-driven Alarm filters", 17th International Conference On Trust, Security And Privacy In Computing And Communication / 12th International Conference On Big Data Science And Engineering, IEEE, pp. 763-773, 2018.

[12] L. Liu, O. De Vel, Q.-L. Han, J. Zhang, and Y. Xiang, "Detecting and Preventing Cyber Insider Threats: A Survey," IEEE Communications Surveys & Tutorials, 2018.
[13] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M.Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D.Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, 2011.
[14]Chih-Hung Hsieh, Chia-Min Lai, Ching-Hao Mao, Tien-Cheu Kao, Kuo-Chen Lee, 46"AD2 : Anomaly Detection on Active Directory Log Data for Insider Threat Monitoring" 49th Annual International Carnahan conference on Security Technology, IEEE, no. 49, pp. 287-292, 2015.
[15] M. L. Collins, M. C. Theis, R. F. Trzeciak, J. R. Strozer, J. W. Clark, D. L. Costa, T.Cassidy, M. J. Albrethsen, and A. P. Moore, "Common sense guide to mitigating insider threats, fifth edition," The CERT Insider Threat Center, Tech. Rep. CMU/SEI- 2015-TR-010, 2016.
[16] S. C. Roberts, J. T. Holodnak, T. Nguyen, S. Yuditskaya, M. Milosavljevic, and W.W. Streilein, "A model-based approach to predicting the performance of insider threat detection systems," in IEEE SPW, 2016.
[17] N. Nguyen, P. Reiher, and G. H. Kuenning, "Detecting insider threats by monitoring system call activity," in Information Assurance Workshop, 2003. IEEE Systems, Man and Cybernetics Society. IEEE, 2003, pp.45–52.
[18] M. B. Salem, S. Hershkop, and S. J. Stolfo, "A survey of insider attack detection research," in Insider Attack and Cyber Security. Springer, 2008, pp. 69–90.