Machine learning based Insider Threat Modelling and Detection

Duc C. Le and A. Nur Zincir-Heywood
Faculty of Computer Science, Dalhousie University, Halifax, Nova Scotia, Canada

Abstract

Recently, malicious insider attacks represent one of the most damaging threats to companies and government agencies. This paper proposes a new framework in constructing a user-centered machine learning based insider threat detection system on multiple data granularity levels. System evaluations and analysis are performed not only on individual data instances but also on normal and malicious insiders, where insider scenario specific results and delay in detection are reported and discussed. Our results show that the machine learning based detection system can learn from limited ground truth and detect new malicious insiders with a high accuracy.

Index Terms

insider threat, cyber-security, machine learning

I. Introduction

Some of the most damaging security threats that companies and government agencies are facing is insider threat, where the malicious acts are performed by authorized personnel in the organizations. Recent reports show that 53% of organizations and 42% of U.S. federal agencies are suffering from insider threats every year [1], [2].

Insider threat related activities can be carried out both intentionally, such as information system sabotage, intellectual property theft and disclosure of classified information, as well as unintentionally, such as careless use of computing resources. One of the challenges of insider threat related research is that a malicious insider is authorized to access the organization's computer systems and has knowledge about the organization's security procedures. Moreover, in organizational environments, malicious insiders' activities may only make up a small portion of user activities in a wide range of domains that are recorded, from web and file access, to email history. Any proposed system for insider threat detection needs to overcome the challenges in learning from highly skewed data of heterogeneous sources in order to distinguish malicious activities from the legitimate ones, where all are from authorized users.

This research presents a system that focuses on user-centered analysis for insider threat detection. Proposed and evaluated on publicly available CERT dataset of corporate data for insider threat detection [3], our system seeks to learn from only a small number of normal / malicious insider actions for identifying threats in unknown data. In this context, we assume the point of view of cybersecurity analysts, where the amount of work is proportional not only to the amount of alerts but also to the number of users flagged, where a range of user's actions need to be taken into account for adequate decisions on alerts [4]. Hence system results are reported per user as well as per data point.

The rest of the paper is organized as follows. Section II summarizes previous research in machine learning applications for insider threat detection. Section III introduces the proposed system as well as the machine learning algorithms employed, and the data processing steps performed. Section IV details the experimental settings while Section V presents the evaluation results. Finally, conclusions are drawn and the future work is discussed in Section VI.

II. Related Work

Insider threat research attracts attention from many government organizations and cybersecurity firms. Common guides to prevent and mitigate insider threats in organizations were released by the CERT Insider threat center and U.S. National Insider threat task force [5], [6]. Liu et al. summarized the field of insider threat research in a recent survey [7]. They reviewed not only insider threats but also related cybersecurity threats, such as advanced persistent threats and malware, which can lead to masquerader cases, where outside attackers have control of inside user accounts.

Early research particularly focused in applying machine learning for insider threat detection goes back a decade ago. Caputo et al. introduced Elicit, a system for monitoring user activities and producing malicious indicators using Bayesian networks [8]. In [9], [10], graph based anomaly detection algorithms were applied for insider threat detection.

Many insider threat detection systems [11]–[14] were stemmed from DARPA's project ADAMS [15], which aims to identify patterns and anomalies in very large datasets to combat insider threats. In [11], [12], various anomaly detection algorithms, including Hidden Markov Model (HMM) and Gaussian Mixture Model were employed on user activity log data for identifying indicators of insider threats. Eldardiry et al. proposed approaches employing hybrid of anomaly detectors on combined information from multiple domains (user activities) to detect blend-in anomalies and unusual change anomalies [13]. Gavai et al. applied different machine learning-based methods on organizational activity data for anomaly and quitter detection [14].

More recently, Rashid et al. used HMMs to model users' weekly activity sequences and detect possible insider threats from the subtle changes [16]. Self Organizing Maps, another unsupervised learning algorithm, has been used in [17] for clustering and visualization of normal and insider user activities. In [18], a framework for modelling the insider threat problem based on behavioural and psychological observations is proposed. The framework allows analyst reasoning on the user data and constructing hypothesis trees describing potential insider threats.

Given the fact that organizational data stream is non-stationary, where user behaviour changes over time, several streaming machine learning approaches have been employed for insider threat detection as well. Some approaches are based on moving weighted sum/average schemes for generating anomaly scores [10], [19]. To this end, Tuor et al. proposed a deep learning based anomaly detection system [19]. Bose et al. proposed a system for analyzing fusion of heterogeneous data streams, based on supervised and unsupervised learning to detect anomalies [20]. On the other hand, Le et al. applied genetic programming algorithms, on two data assumptions: stationary and non-stationary, and demonstrated promising performance of bio-inspired algorithms in insider threat detection [21].

In this paper, different from previous work – we propose an user-centered insider threat detection system. We evaluate the proposed system on publicly available CERT insider threat dataset [3], which is generated as organizational data specifically for insider threat detection system evaluations. Furthermore, the proposed system assumes the point of view of a security analyst in evaluating suspicious user activities, where the results are reported by not only the correctly detected data instances, e.g. weekly or daily user data points, but also per normal and malicious insiders identified. Finally, we also evaluate different levels of granularity in data preprocessing, from a session to a week of user actions. This stimulates the understanding of system behaviours in malicious insider detection and response time.

III. System Description

A. System Overview

Figure 1 illustrates an overview of our proposed approach for malicious behaviour and insider threat detection. The data collection and data processing steps are detailed in Section III-B. Based on given data, initial anomaly detection steps can be performed to obtain an initial set of confirmed malicious and normal users' data. Alternatively, security analysts may notice suspicious activities or unusual changes in

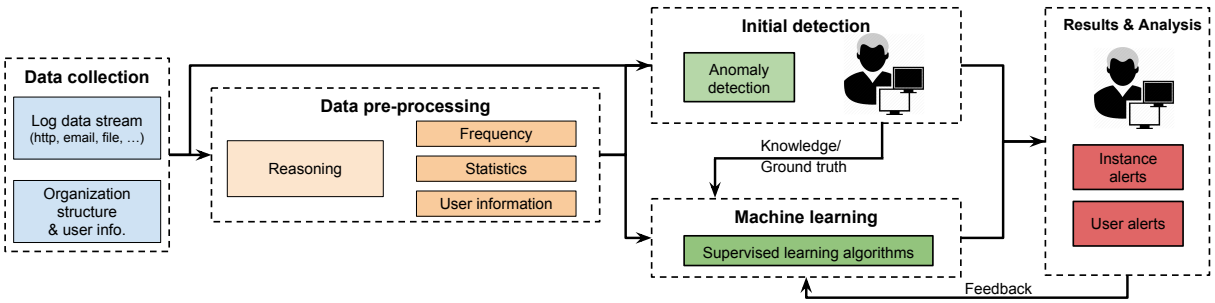

Fig. 1. Overview of the proposed system

user behaviours and may perform more detailed analysis. Either way, this step is expensive in terms of labor, and may result in high false alarm rate per each malicious user detected [7].

Supervised learning algorithms are employed in the next step to learn from the initial knowledge on malicious / normal user behaviours and generalize for detecting previously unseen malicious insider cases. The obvious advantage of supervised learning algorithms in this case is that they are able to generate classifiers with much higher precision than unsupervised learning / anomaly detection algorithms. Additionally, security analyst's verdicts on alerts / warnings can be used as an important source to reinforce machine learning models.

On result reporting, the proposed system shows not only how many malicious insiders are detected but also how quickly they are detected. In many cases, user actions in a long time range need to be taken into account in order to process an alert regarding suspicious user behaviour. We believe user-based result reporting would play an important role when the system is applied in real-world scenarios.

In this paper, the focus is in applying machine learning algorithms on the ground truth obtained from the initial detection step for identifying unknown malicious insiders. To achieve this, we aim to examine the capabilities of machine learning algorithms under limited ground truth conditions in generalizing user behaviours for detecting insider threats to support security analysts. It is noteworthy that the initial detection step can be performed similar to some of the previous work in [11], [13], [17].

B. Data Collection and Pre-processing

Data collection and pre-processing is crucial for insider threat detection in particular but also for cybersecurity tasks in general. A good monitoring procedure in combination with adequate data collection enables a successful application of machine learning techniques and support security analysts in making correct decisions.

In this paper, based on the publicly available CERT dataset for insider threat detection (see IV-B), we employ the following data sources from corporate environments as input to our system: (i) activity log data stream, such as web history, email log, file and device access, or log on/off records, and (ii) organization structure and user information as background or context for data analysis. In the case of CERT insider threat data, the Lightweight Directory Access Protocol (LDAP) is employed to maintain the second source of data.

Based on available sources of data, suitable processing steps need to be carried out to provide data in appropriate formats for machine learning algorithms. In many cases, the data collected does not accompany sufficient background information that is necessary for feature extraction, hence there is a reasoning step in data pre-processing. This step is performed to obtain helpful auxiliary information for further processing. Specifically, for the CERT dataset, we had to examine the data and design appropriate schemes for determining user-user relationships, user-PC relationships, regular work hours, and website and file categories.

From the collected data and auxiliary information extracted in the previous step, feature extraction is performed. In this work we extract two types of features from the data: (i) frequency features, specifically

number of actions over a time period, e.g. *number of emails sent, number of log on after hour, or the number of USB connects to a shared PC*, and (ii) statistical features, specifically mean and standard deviation of data, e.g. *email size, file size, or the number of words in websites visited*. Additionally, the user information, which is mostly categorical data encoded in numeric format, is included in the processed data in order to provide context for machine learning algorithms.

Based on the chosen duration for feature extraction: session, day, or week, there are different levels of granularity. Session data points capture actions over time from each log on to corresponding log off of each user. Session extracted data tends to have less number of features, as a session is usually short and concentrated in one time category and one PC only. But session-based data may be helpful in isolating malicious actions, since malicious users tend to perform malicious actions in certain sessions and other sessions in the same day or week may still be normal. Day and week based data points summarize users' actions over the corresponding time period. These types of data may provide better overview of behaviours in a day or a week with a higher feature count, and may accelerate learning by lower amount of data. However, they may also lead to average out the real malicious actions and require longer reaction time once insiders are detected. Finally, it is noteworthy that the data collection and processing steps are designed to be flexible and will need to be adjusted to fit to the specific organization conditions and limitations.

C. Machine Learning Algorithms

As mentioned above, in this study we focus on the applications of supervised learning algorithms on limited ground truth and detecting unknown insider threat cases. To that end, we employ the following popular learning algorithms [22].

1) Logistic Regression (LR): Logistic regression is a linear statistical model that uses a logistic function to model a binary dependent variable, which is in our work the indicator of normal or malicious insider behaviour. The aim of logistic regression, with $l2$-regularization, is to find the coefficient vector, w, that minimizes the sum of squared errors between logistic function $\sigma(w^T x)$ and labels, where x denotes a data sample. As a linear model, logistic regression has the advantage of being highly interpretable. Furthermore, logistic regression returns a probability for belonging to a class, which can be helpful in supporting security analyst in prioritizing the most suspicious actions for investigation. In this work, logistic regression is included as a baseline model.

2) Random Forest (RF): Decision tree learns tree-like nonlinear classification model, where each leave of the tree represents a "decision", or predicted class label. Decision tree is widely used for its interpretability and efficiency, where each decision at a tree's node can be represented as a rule over the input space. In this work, we employ Random Forest algorithm, which is an ensemble learning method. Random forest trains a set of individual decision trees, each with a random subset of training data and input features, which are called bootstrap aggregating and random subspace methods. The output of random forest is usually taken as the majority of votes from all the individual tree. The individual trees of random forest are trained using CART algorithm, which seeks to maximize information gain at each split of the tree, measured by "gini" impurity.

3) Artificial Neural Network (ANN): A neural network consists of neurons, which are organized by a number of layers, including input, output and hidden layers, and connections between neurons of consecutive layers. Usually the weight vectors of the connections are trained by back propagation algorithm over a number of epochs, or until stopping criterion is met. A neural network with a large amount of connection weights can represent highly non-linear mappings between input data and the output. However, neural network models are very limited in interpretability, which is often crucial in cybersecurity. In this paper, we employed feed forward neural networks with up to three fully connected hidden layers and rectified linear unit and logistic as the activation functions for hidden layers and the output layer, respectively.

IV. Experiments

A. Dataset Employed

This work employs the CERT insider threat dataset, which is a publicly available dataset for research, development, and testing of insider threat mitigation approaches [3]. The release 5.2 of the dataset (CERT r5.2) simulates an organization with 2000 employees over the period of 18 months. For the data collection step (III-A), CERT r5.2 consists of user activity logs in following categories: log on/off, email, web, file and thumb drive connect, as well as organizational structure and user information. Each malicious insider in CERT r5.2 belongs to one of four popular insider threat scenarios, from data exfiltration (scenario 1), intellectual property theft (scenarios 2, 4) to IT sabotage (scenario 3). Details of the insider scenarios can be found in [3].

Based on the original CERT data, we performed data processing steps to obtain data in three levels of data granularity, namely user-session, user-day, and user-week, as described in III-B. On feature count, pre-processed user-session, user-day, and user-week data have 221, 824, and 1092 features, respectively. Table I shows the distribution of CERT r5.2 data in the three categories and the number of normal and malicious users. It is obvious that the data distribution is extremely biased toward normal user data, where malicious insider related data only accounts for 0.18%, 0.19%, and 0.39% of user-session, user-day, and user-week data, respectively. Moreover, inspecting the insider threat scenarios, one can easily see that malicious actions in scenarios 2 and 4 span over longer periods – averaged to 8 weeks or 22 to 33 sessions per user – than scenarios 1 and 3. This represents the malicious insiders' attempts in avoiding detection by performing malicious actions over a long period of time.

TABLE I
Summary of the dataset

	Normal	**Insider threat scenarios**			
		1	**2**	**3**	**4**
user-week	141572	49	245	10	248
user-day	692455	85	863	20	339
user-session	1002803	65	1070	33	678
# users	1901	29	30	10	30

B. Evaluations

In real-world environments, ground-truth, or label for training detection systems is scarce. To simulate that condition, in the evaluations we assume that ground truth is obtained from only a restricted set of users. Specifically, the training data for the supervised machine learning algorithms is limited to the data of maximum 400 identified "normal" and "malicious" users (among 2000 users in the organization), from only the first 37 weeks – 50% of the time period that the dataset covers. By user count, the training data accounts for 18% of "normal" users in the training weeks and 34% of all malicious insiders. Its noteworthy that from detector's point of view, the "normal" users in the training set is only confirmed to be benign in the first 37 weeks, and they may or may not turn "malicious" later, i.e. in the test data.

The binary classifier models trained on this small training data is then used to predict user behaviours in the later weeks of the CERT r5.2 data. Furthermore, models trained on CERT r5.2 are used to test against CERT r5.1, which has a similar organization structure, but only has one insider per scenario, and CERT r6.2, which simulates a larger organization with one more insider scenario. The results are reported in terms of data instances correctly detected (instance-based detection rate – IDR), or users correctly detected (user-based detection rate – UDR). In the case of UDR:

- A normal user is misclassified if at least one of his data instances is classified as "malicious",
- A malicious insider is detected if at least one of his data instances is classified as "malicious".

C. Machine Learning Algorithm Training

We implemented data processing steps on Python 3.7 and used machine learning implementations from Scikit-learn [23]. For the Logistic Regression based classifier (LR), *lbfgs* solver is used to speed up training by parallelization, while all other parameters are left as default. For the Random Forest based classifier (RF), we used random search with cross-validation to adjust the following hyperparameters: the number of individual decision trees (50, 100, and 200), the number of features available for training individual trees (all features, square root and base-2 logarithm of all features), and the maximum number of leaf nodes in a tree (100, 200, or unlimited). Finally, for ANN, we used Adam optimization algorithm for training, up to 250 epochs, and again employed random search for tuning the hyperparameters. For the number of hidden layers, we searched between 1 and 3, where each hidden layer has the size set to a half of the previous layer. Other tuned hyperparameters include batch size (32, 64, and 256), $l2$ penalty (10^{-4}, 10^{-2}, and 10^{-1}), and Adam's epsilon (10^{-8}, 10^{-4}, and 10^{-1}).

V. Results

In the following, due to the extremely skewed data, we present results – averaged from 5 runs – of ML algorithms on the aforementioned datasets based on detection rates (DR), and malicious insider detection precision. Specifically, $DR = TP/(TP + FN)$ and $Precision = TP/(TP + FP)$, where TP, FN, FP are true positive, false negative, and false positive, respectively. As presented in IV-B, there are two types of DR based on data instance (IDR) or user (UDR).

A. Results on Test Data

TABLE II
INSTANCE-BASED TEST RESULTS

Algorithm	User-Session			User-Day			User-Week		
	IDR-Normal	IDR-Malicious	Precision	IDR-Norm	IDR-Malicious	Precision	IDR-Normal	IDR-Malicious	Precision
LR	99.74%	26.40%	0.194	99.27%	**44.80%**	0.127	98.53%	**57.52%**	0.171
RF	**99.99%**	**30.39%**	**0.884**	**99.99%**	37.50%	**0.880**	**100.00%**	46.42%	**0.994**
ANN	99.89%	28.65%	0.383	99.61%	42.27%	0.202	99.61%	46.76%	0.391

TABLE III
USER-BASED TEST RESULTS

Algorithm	User-Session			User-Day			User-Week		
	UDR-Normal	UDR-Malicious	Precision	UDR-Normal	UDR-Malicious	Precision	UDR-Normal	UDR-Malicious	Precision
LR	92.93%	**90.46%**	0.314	91.60%	**92.62%**	0.283	91.60%	**93.23%**	0.284
RF	**99.54%**	82.15%	**0.864**	**99.69%**	79.69%	**0.902**	**99.99%**	69.54%	**0.996**
ANN	97.13%	86.77%	0.519	96.65%	91.38%	0.493	97.55%	78.15%	0.532

Tables II and III present the results of machine learning algorithms on the unseen test data part of CERT r5.2 based on data instances and users. The tables show that there are trade-offs between malicious insider DR (recall) and precision, or normal user DR. Logistic Regression usually achieves high DR-malicious, while suffers from low precision, which translates to high expense of an analyst's investigation time on false alarms. On the other hand, RF and ANN show better balance in maintaining low false positive rates and high malicious insider detection rates. While RF shows very good precision, which results in more efficient use of analyst's time on false alarms, ANN gives better malicious insider DR in most of the cases tested. RF's higher precision can be explained through its resistance to class imbalance in training data [24]. In this case, RF is able to maintain a higher malicious DR while keeping a lower false positive rate than ANN and LR.

By investigating and reporting results based on users, we can see that the classifiers could learn to recognize at least one instance which represents malicious behaviour for most of the malicious insiders (80 to 90%). This shows the shortcomings of simply reporting results per data instance rather than per user. Given that the purpose of an insider threat detection system is to detect malicious instances as well as malicious insiders, we believe that reporting results as in Table III provides a more realistic estimation of the detector's capabilities.

B. Scenario Specific DRs and Detection Delay

TABLE IV
SCENARIO SPECIFIC TEST RESULTS: DR AND AVERAGE DETECTION DELAY*

Data type	**Alg.**	**Metric**	**Scen 1** 18 users	**Scen 2** 16 users	**Scen 3** 7 users	**Scen 4** 24 users
User-Session	LR	UDR	100%	61.25%	100%	100%
		Delay	0	6.16	0.29	4.25
	RF	UDR	100%	27.50%	100%	100%
		Delay	0	5.5	0.29	0
	ANN	UDR	100%	50.00%	91.43%	100%
		Delay	0	7.47	0.44	0.4
User-Day	LR	UDR	100%	70%	100%	100%
		Delay	0.44	4.79	0.00	2.13
	RF	UDR	100%	46.25%	34.29%	100%
		Delay	0.44	3.16	0.00	2.99
	ANN	UDR	100%	67.50%	94.29%	100%
		Delay	0.44	4.93	0.45	3.20
User-Week	LR	UDR	100%	73.75%	100%	99.17%
		Delay	0	1.24	0	1.24
	RF	UDR	100%	12.50%	34.29%	95%
		Delay	0.02	1.8	0	1.28
	ANN	UDR	100%	58.75%	2.86%	96.67%
		Delay	0.03	1.62	0	2.18

*The unit of 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.

As discussed in part IV-A, there are four insider scenarios in CERT r5.2. Table IV shows results on each scenario by DR and detection delay for the malicious insiders 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. Detection rates on scenario 4 are high too, however classifiers take much more time to detect malicious behaviours. On scenario 3, results on user-session data are better than on user-week and user-day data. Scenario 2 is shown to be the hardest to detect in all cases. This may be due to the nature of the actions in this scenario. Dataset description states that in this scenario, "user surfing job websites and soliciting employment from a competitor, and before leaving the company, they use a thumb drive to steal data" [3], which shows less obtrusive actions than other scenarios. Furthermore, the insiders in scenario 2 purposely hid malicious actions by performing them over a longer time period (2 months in average, see IV-A), making it harder to detect.

C. Results on Other CERT Datasets

Other than CERT r5.2, there are different releases of the CERT dataset. In this part, we test the trained models on CERT r5.1, which simulates a similar organization structure with the same number of users,

TABLE V
USER-BASED TEST RESULTS ON CERT R5.1 AND R6.2 OF MODELS TRAINED ON CERT R5.2

Test	Alg.	User-Session			User-Day			User-Week		
		UDR-Normal	UDR-Mal.	Precision	UDR-Normal	UDR-Mal.	Precision	UDR-Normal	UDR-Mal.	Precision
r5.1	LR	88.40%	80.00%	0.014	87.05%	85.00%	0.013	87.62%	75.00%	0.012
	RF	99.42%	75.00%	0.205	99.48%	60.00%	0.187	99.93%	50.00%	0.588
	ANN	95.10%	75.00%	0.030	94.84%	75.00%	0.028	96.81%	75.00%	0.045
r6.2	LR	14.41%	80.00%	0.001	40.33%	88.00%	0.002	72.43%	64.00%	0.003
	RF	89.21%	56.00%	0.006	97.39%	56.00%	0.026	98.89%	48.00%	0.052
	ANN	77.35%	60.00%	0.003	93.04%	48.00%	0.009	99.96%	12.00%	0.273

and CERT r6.2, which has a different organization structure and more users, 4000. Both CERT r5.1 and r6.2 has only one insider per scenario, making the detection task much more challenging. Results are shown in Table V. It is clear that on CERT r5.1, the models perform well with low false alarm rates and 75% malicious insider detection rate, where only the scenario 2 insider is missed. On CERT r6.2, due to a different organization structure, accuracy is lower. On user-week data, most of the insider cases are missed, while on user-session data, although malicious insider detection rates are acceptable, the precision is low. On the other hand, accuracy could be increased on CERT r6.2 by using RF or ANN with user-day data. This suggests that on a different organization, the previously trained machine learning models can only be used as an initial detection step, and new models need to be trained for better accuracy.

VI. CONCLUSION AND FUTURE WORK

In this paper, we present a machine learning based system for insider threat detection. ANN, RF and LR learning algorithms are trained on limited ground truth to support cybersecurity analysts in detecting malicious insider behaviours on unseen data. Results show that the proposed system was able to generalize on limited training data to detect new malicious insiders, with high precision, especially when user based results are considered. On data granularity, user-session data appears to be the best candidate, as it allows a system with high malicious insider detection rate and fast response times. On machine learning algorithms employed, RF and ANN show good performances. RF can be employed where manpower for investigating the alarms is limited, as it gives high precision. On the other hand, ANN gives higher malicious insider DR.

In the future work, we will investigate similar systems under more limited learning conditions. More sophisticated data pre-processing techniques as well as feature analysis can be used to improve system performance.

ACKNOWLEDGMENT

This research is supported by Natural Science and Engineering Research Council of Canada (NSERC). Duc C. Le gratefully acknowledges the supports by Killam, Mitacs, and Nova Scotia graduate scholarships. The research is conducted as part of the Dalhousie NIMS Lab at: https://projects.cs.dal.ca/projectx/.

REFERENCES

[1] Meritalk, "The 2017 federal insider threat report," Symantec, Tech. Rep., 2017. [Online]. Available: https://www.meritalk.com/study/inside-job-the-sequel/

[2] Crowd Research Partners, "2018 insider threat report," 2018. [Online]. Available: https://crowdresearchpartners.com/insider-threat-report/

[3] J. Glasser and B. Lindauer, "Bridging the gap: A pragmatic approach to generating insider threat data," *IEEE Security and Privacy Workshops*, 2013.

[4] J. Polverari, "Why less is more when it comes to cybersecurity," 2018. [Online]. Available: https://www.forbes.com/sites/forbestechcouncil/2018/06/01/why-less-is-more-when-it-comes-to-cybersecurity/

[5] 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.

[6] National Insider Threat Task Force, "2017 NITTF insider threat guide," 2017. [Online]. Available: https://www.dni.gov/files/NCSC/documents/nittf/NITTF-Insider-Threat-Guide-2017.pdf

[7] 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.

[8] D. Caputo, M. Maloof, and G. Stephens, "Detecting insider theft of trade secrets," *IEEE Security & Privacy Magazine*, vol. 7, no. 6, 2009.

[9] W. Eberle, J. Graves, and L. Holder, "Insider threat detection using a graph-based approach," *J. Applied Security Research*, vol. 6, no. 1, 2011.

[10] P. Parveen and B. Thuraisingham, "Unsupervised incremental sequence learning for insider threat detection," in *IEEE International Conference on Intelligence and Security Informatics*, 2012.

[11] T. E. Senator, E. Chow, I. Essa, J. Jones, V. Bettadapura, D. H. Chau, O. Green, O. Kaya, A. Zakrzewska, E. Briscoe, R. I. L. Mappus, H. G. Goldberg, R. McColl, L. Weiss, T. G. Dietterich, A. Fern, W.-K. Wong, S. Das, A. Emmott, J. Irvine, J.-Y. Lee, D. Koutra, A. Memory, C. Faloutsos, D. Corkill, L. Friedland, A. Gentzel, D. Jensen, W. T. Young, B. Rees, R. Pierce, D. Huang, M. Reardon, and D. A. Bader, "Detecting insider threats in a real corporate database of computer usage activity," in *19th ACM SIGKDD*, 2013.

[12] H. G. Goldberg, W. T. Young, M. G. Reardon, B. J. Phillips, and T. E. Senator, "Insider threat detection in PRODIGAL," in *Annual Hawaii International Conference on System Sciences*, 2017.

[13] H. Eldardiry, E. Bart, J. Liu, J. Hanley, B. Price, and O. Brdiczka, "Multi-domain information fusion for insider threat detection," in *IEEE Security and Privacy Workshops*, 2013.

[14] G. Gavai, K. Sricharan, D. Gunning, J. Hanley, M. Singhal, and R. Rolleston, "Supervised and unsupervised methods to detect insider threat from enterprise social and online activity data," *J. Wireless Mobile Networks, Ubiquitous Computing, & Dependable Applications*, vol. 6, no. 4, 2015.

[15] Defense Advanced Research Projects Agency, "Anomaly detection at multiple scales (ADAMS)," https://www.darpa.mil/program/anomaly-detection-at-multiple-scales.

[16] T. Rashid, I. Agrafiotis, and J. R. Nurse, "A new take on detecting insider threats," in *Intl. Workshop on Managing Insider Security Threats*, 2016.

[17] D. C. Le and A. N. Zincir-Heywood, "Evaluating insider threat detection workflow using supervised and unsupervised learning," in *IEEE Security and Privacy Workshops*, 2018.

[18] P. A. Legg, O. Buckley, M. Goldsmith, and S. Creese, "Automated insider threat detection system using user and role-based profile assessment," *IEEE Systems Journal*, vol. 11, no. 2, 2017.

[19] A. Tuor, S. Kaplan, B. Hutchinson, N. Nichols, and S. Robinson, "Deep learning for unsupervised insider threat detection in structured cybersecurity data streams," in *AAAI Workshop on AI for CyberSec.*, 2017.

[20] B. Bose, B. Avasarala, S. Tirthapura, Y. Y. Chung, and D. Steiner, "Detecting insider threats using radish: A system for real-time anomaly detection in heterogeneous data streams," *IEEE Systems Journal*, 2017.

[21] D. C. Le, S. Khanchi, A. N. Zincir-Heywood, and M. I. Heywood, "Benchmarking evolutionary computation approaches to insider threat detection," in *Genetic and Evolutionary Computation Conference*, 2018.

[22] E. Alpaydin, *Introduction to Machine Learning*. The MIT Press, 2014.

[23] 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.

[24] D. J. Dittman, T. M. Khoshgoftaar, and A. Napolitano, "The effect of data sampling when using random forest on imbalanced bioinformatics data," in *IEEE International Conference on Information Reuse and Integration*, 2015, pp. 457–463.