Insider Threat Detection using Deep Autoencoder and Variational Autoencoder Neural Networks

Efthimios Pantelidis*, Gueltoum Bendiab†, Stavros Shiaeles†, Nicholas Kolokotronis‡
* Faculty of Science and Applied Sciences, Open University of Cyprus (OUC)
Nicosia, Cyprus, efthymios.pantelides@st.ouc.ac.cy
†Cyber Security Research Group, University of Portsmouth, PO1 2UP, Portsmouth, UK
gueltoum.bendiab@port.ac.uk, sshiaeles@ieee.org
‡Department of Informatics and Telecommunications, University of Peloponnese
22131 Tripolis, Greece, nkolok@uop.gr

***Abstract*—Internal attacks are one of the biggest cybersecurity issues to companies and businesses. Despite the implemented perimeter security systems, the risk of adversely affecting the security and privacy of the organization's information remains very high. Actually, the detection of such a threat is known to be a very complicated problem, presenting many challenges to the research community. In this paper, we investigate the effectiveness and usefulness of using Autoencoder and Variational Autoencoder deep learning algorithms to automatically defend against insider threats, without human intervention. The performance evaluation of the proposed models is done on the public CERT dataset (CERT r4.2) that contains both benign and malicious activities generated from 1000 simulated users. The comparison results with other models show that the Variational Autoencoder neural network provides the best overall performance with a higher detection accuracy and a reasonable false positive rate.**

***Index Terms*—Deep Learning, Insider Threat, Network Security, Anomaly Detection.**

I. Introduction

Insider threat is currently one of the biggest security issues for institutions, companies and government agencies [1], [2]. This threat usually originates from users with legitimate access to an organisation asset who use that access either maliciously or unintentionally, to cause data breaches and harms to the organisation [3], [4]. The insider could be a current employee, consultant, former employee, broad member or any other business associate. Internal attacks can be performed intentionally by a malicious insider, for malicious purposes such as information system espionage, intellectual property theft, or disclosure of sensitive data. This kind of internal attacks typically implicates a current or former employee or any other business partners that have legitimate access to sensitive data or privileged accounts within the network of an organisation, and who misuses this access [4]. It could also be performed by exploiting a negligent insider who does not follow proper security measures, such as an employee who did not change a default password or fall victim to a phishing attempt [5]. In either case, negligence is often considered the most expensive type of insider risk [6].

Despite the implemented security measures, the risk of adversely affecting the security of the organisation's data, or information systems, remains high. Insider attacks are particularly more dangerous than outside attacks because an insider already has direct access to the organisation and its network and does not need to hack in over the outer perimeter. Thus they are harder to defend [2], [3]. It is also more challenging to detect insider threats because they can easily evade existing defences. Moreover, it is very difficult to distinguish between a legitimate user's activity and potentially malicious activity, especially for those with high levels of access [7]. According to Verizon (2019) [8], over 57% of data breaches in organisations involve internal threats, and more than 40% of these violations, usually take months or even years to be identified. Another recent report by the Pomeron institute announced that the average annual cost of insider threat was $11.45 million in 2020, with an increase of 31% compared to 2019 [9]. While this issue has been explored for a long time in both industrial and research communities, most proposed techniques rely on feature engineering, which is difficult and time-consuming. In addition, most of them cannot precisely capture the behaviour difference between malicious and normal insider user activities due to several difficulties related to the characteristics of related data. Deep learning techniques give a new powerful paradigm that can automatically discover the features needed for insider threat detection. Although some progress has been done in this field, the topic of applying deep learning for insider threat detection is not well-investigated [6], [10].

Therefore, this paper investigates the application of the deep neural networks Autoencoder (AE) and Variational Autoencoder (VAE) in detecting malicious insiders automatically, without human intervention. These two deep neural networks have proved their effectiveness in discovering high quality, non-linear features for anomaly detection, in various fields. They can learn different levels of representations from the input data based on the multi-layer structures [6]. Moreover, The VAE could generate new data from the source dataset. In this study, the AE and VAE neural networks have been implemented using the Python programming language with the Keras library and the TensorFlow environment. While the validation is performed on the public CERT (version r4.2). This version of the dataset contains both normal and malicious user activities that are generated from 1000 simulated

employees. The comparison results with our previous models in [11], [12] indicate that the Variational Autoencoder neural network provides the best overall accuracy in detecting internal threats with a lower false-positive rate. The rest of this paper is structured as follow: Section 2 reviews the related work in the area of insider threat. Section 3 explains the proposed method to detect insider threats and specific parameters. Section 4 presents the testing design and results, and Section 5 presumes the paper along with some future directions.

II. Related work

In recent years, with a large number of incidents, several research studies have attempted to design new techniques to solve the insider threat detection problem [6], [13]. A number of recent solutions in this field have been studied in [2]. In this study, the authors proposed a novel categorisation of insider threat-related work based on the techniques used in the detection. This study concluded that machine learning-based detection techniques are the most powerful for solving the problem of insider threat. In this technique, a user normal profile is built based on their normal behaviour. In this case, the anomalies are identified as deviations from the normal behaviour [2], [5], [14]. In this context, a variety of machine learning algorithms have used for insider threat detection. For instance, authors in [14] introduced an automated system that has the ability to detect insider threat based on the user's profile. Each user's profile is constructed based on a three-structure approach that involves the details of their activities and job role. The created profile is then used for comparison with other users' activities of the same role. Significant deviations in the user's activities will be considered as an anomaly and potential internal threat.

In another work [15], authors introduced an intrusion sensitive-based trust management model to improve the protection of collaborative Intrusion Detection Systems (IDSs) against insider threats. The proposed model used a supervised machine learning technique to automatically assign a trust value, known as intrusion sensitivity, to each IDS based on expert knowledge. This trust value is used to evaluate the trustworthiness of an IDS in the system. In this work, the authors investigated the performance of three machine learning algorithms, k-nearest neighbours algorithm (KNN), back-propagation neural networks (BPNN) and decision tree (DT), in assigning sensitivity value under various attack scenarios in a real wireless sensor network. In more recent work [16], authors investigated the efficiency of using supervised machine learning and data mining to identify insider threats. The proposed models have been tested on a data set that comprises behaviour traces of 24 users examined over five days of spam. The machine learning algorithms used in the experiments are Adaboost, Naive Bayes (NB), Logistic Regression (LR), KNN, Linear Regression (LR) and Support Vector Machine (SVM). The authors found that Adaboost has outperformed other algorithms with 98.3% Accuracy in identifying malicious emails. In another work [17], the authors used a Hidden Markov Model (HMM) to build the user profile by capture his normal activity per week, and then use this profile to detect significant deviations from this normal behaviour. In more recent work [18], authors tested the efficiency of different machine learning algorithms in detecting anomalies and early quitter indicators, where both indicate a potential internal attack. The proposed detection models are applied to a dataset that consists of 5,500 simulated users.

Many other recent approaches to insider threat have been used deep learning algorithms, which can automatically discover the features needed for detection [19], [20]. Especially, Recurrent Neural Network (RNN), which has been used to model the user activities for insider threat detection [19]. Currently, Long Short-Term Memory (LSTM) [21] and Gated Recurrent Unit (GRU) are the two main variants of the RRN that have been widely applied to model the user activities for building a user normal profile [10], [19], [22], [23]. For instance, in [10], authors employed Long Short-Term Memory (LSTM) to extract temporal features for building the user profile and the Convolutional Neural Network (CNN) to detect insider threat using the extracted features. In [23], authors proposed an online anomaly detection approach using a deep recurrent neural network, to produce anomaly scores. Convolutional Neural Network (CNN), which gained great success in images classification], has also been used to detect insider threat [24] by analysing images generated from the user mouse bio-behavioural characteristics. The proposed approach can perform continuous identity authentication on current computer users with a false acceptance rate of 2.94% and a false rejection rate of 2.28%. In other recent work [25], authors have applied ensemble deep autoencoders for insider threat detection. Each AE is trained using a certain category of audit data, which designates a user's normal behaviour. The authors claimed that the proposed detection system is able to detect all of the malicious insider actions. However, these detection systems have a high false-positive rate. Work in [26] implemented a Deep Neural Network (DNN) model based on the auto-encoder NN to detect insider threat. The validation of this system is done with a real real-world data set composed of 3.6 billion log files and 70.2 million entities. This system achieved good results; however, the user activities were not clearly modelled over time.

III. Proposed methodology

The goal of this study is to evaluate the efficiency of the AE and VAE neural networks in automatic detection of insider threats, without human intervention. As shown in Fig. 1, the overall malicious behaviour detection process involves two main steps. The first stage starts with the data collection from the public dataset CERT r4.2 created by SEI (Software Engineering Institute) for insider threat tests purposes. The dataset includes log files that record an activity including eighteen (18) months, collected between 01.01.2010 and 31.05.2011 [11]. For each user in the dataset, the inputs involved login records, files used by the users, emails sent and received, web browsing history, removable media used, and employee role within the organisation. Collected data are then processed to create

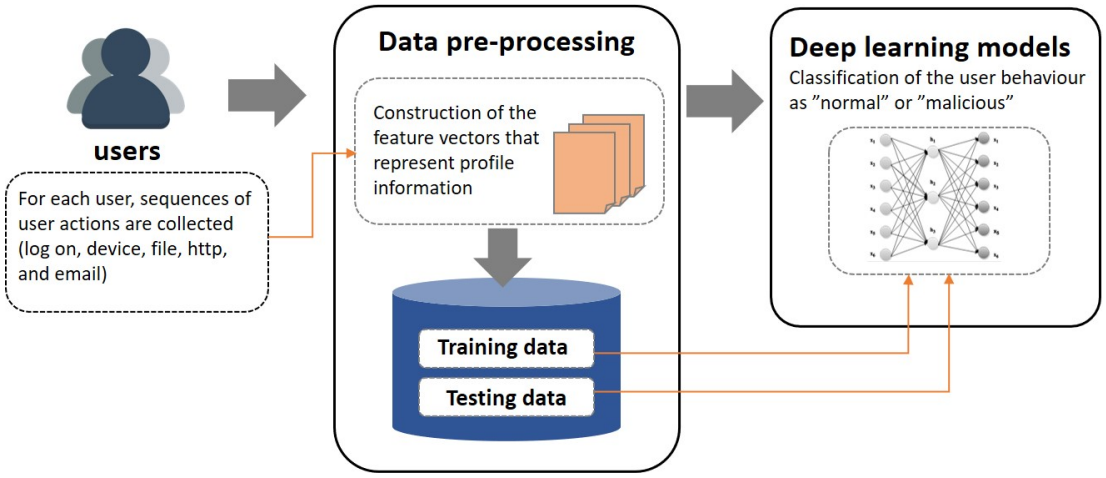

Fig. 1. Overview of the proposed methodology

feature vectors representing profile information and converted into a format (i.e., numerical format) that can be fed into the implemented models. In the next step, built feature vectors are fed to the implemented neural networks autoencoder (AE) and variational autoencoder (VAE) to perform predictions and classify user activities as normal or anomaly. The following sections provide detailed information about the steps we followed to complete the process and classify user behaviour as ”normal” or ”malicious”.

A. Dataset and Data Collection

Data collection is a crucial step in the proposed model. In fact, appropriate data collection and processing allows a successful application of the neural networks models AE and VAE with higher accuracy rate and lowers false alarms. Currently, there is a large number of public data sources that can be used to evaluate insider threat detection models, however, in this paper, we have used a public insider threat dataset created by SEI (Software Engineering Institute) [21]. The institute proposed a collection of ten different test datasets (r1, r2, r3.1, r3.2, r4.1, r4.2, r5.1, r5.2, r6.1, r 6.2) that provide both background and malicious actor synthetic data. In this work, we have chosen version r4.2 of the CERT dataset [21]. This version of the dataset comprises many samples that are collected from three different scenarios of insider threat. The dataset is “free of privacy and restriction limitations”. It contains both benign and malicious user activities generated from 1000 simulated employees. The CERT r4.2 dataset is split into seven log files, which recorded users activity covering eighteen (18) months, collected between 01.01.2010 and 31.05.2011. The log files are presented in Table I.

The version r4.2 of the CERT dataset contains the following insider threat scenarios.

1) The first scenario describes an employee in an organisation who starts logging into the organisation’s system after working hours, using a removable disk. The employee uploads sensitive data from the organisations’ system to the wikileaks.org website and leaves the organisation shortly thereafter.
2) The second scenario represents an employee who starts looking for job websites and requesting employment from an adversary to the organisation. Before moving to another company, he used a USB thrum drive to steal sensible data.
3) The third scenario presents a system administrator in a company who becomes disappointed. Consequently, he downloaded a keylogger and used a USB stick to transfer it to his supervisor’s machine. Then, he logged in as a supervisor and sent an alarming mass email, which caused panic in the organisation. The system administrator left the organisation shortly.

TABLE I
CSV LOG FILES OF THE CERT R4.2 DATASET

File Name	Description
logon.csv	Records the user’s input and output
device.csv	Records USB connection and disconnection
http.csv	Records internet usage
email.csv	Records e-mail usage
file.csv	Records saving files to removable devices
psychometric.csv	Psychometric results
ldap.csv	Users of the organisation and their roles

B. Data Pre-Processing

In the first step of data processing, the raw input files are imported from the CERT r4.2 and processed. Initially, due to the large volume of data and high memory requirements, we have created sample files of 5000 entries per column for each of the CSV files, logon, device, file, HTTP, and email. In order to accurately represent the user behaviour, we have extract seven features that represent the possible user

activities in the system. This includes logon /log-off to a system, connect/disconnect thumb drive, send/receive an e-mail, process a file or surfing the internet. These features are collected in a single CSV file with entries from 1000 rows. The features used to create this file are ”day”, ”time”, ”Logon”, ”Logoff”, ”Connect”, ”Disconnect”, ”email”, ”file”, and ”HTTP”. Because the deep learning models require all input and output variables to be numeric, all features' values in the created CSV file have been converted to numeric values (Integer). first, the seven users' activities (logon, logoff, Connect, disconnect, email, file and HTTP) are converted to integers from 1 to 7. The ”date” column has been also converted to ”day” and ”time”. Then, the hot encoding method is used to convert all the inputs to binary 0 and 1. Table II presents the extracted features at the pre-processing stage.

TABLE II
SUMMARY OF THE EXTRACTED FEATURES AT PRE-PROCESSING PHASE.

Feature	Possible Values
Day	0, 1, 2, 3, 4, 5, 6
Time	1, 2, 3, 4, . . . , 24
Activities	”Logon”=1, ”Logoff”=2, ”Connect”=3, ”Disconnect”4, ”email”=5, ”file”=6, ”http”7

Finally, we have divided the obtained dataset into two subsets, which are train and test sets. The training dataset, which includes 75% of all data, is used to train the AE and VAE models with enriched or labelled data. For that, a set of usernames of the malicious insiders is created according to the scenarios in the CERT r4.2 dataset. Then, a new column has been added to the CSV file, called ”insider” with binary value. The value of this feature is set to 1 if the user is an insider. Otherwise, it is set to 0. While the test set, which contains 25% of all data, is used to validate the efficiency of the implemented models in predicting the test data.

C. Implementing AE and VAE algorithms to detect anomalies

The efficiency of the proposed models is highly dependent on the correct building of the AE and VAE and their related parameters including the loss function to select, the number of layers and for each layer, the best activation function to adopt. The Autoencoder neural network is composed of two connected networks, an encoder and a decoder networks. The encoder learns how to read the input and compress it to an internal representation defined by the bottleneck layer. Then, the decoder network uses the output of the encoder (the bottleneck layer) to reconstruct the input [27]. After training the autoencoder neural network, the decoder is dropped and we only keep the encoder and use it to compress samples of input data to vectors output by the bottleneck layer. For the implementation of the autoencoder model, we have built an input level, encoder network, decoder network and output level. In this work, we have defined the encoder network to have two hidden layers, the first with two times the number of inputs (e.g., 100) and the second with the same number of inputs (i.e., 50), succeeded by the bottleneck layer with the same number of inputs as the dataset (i.e., 50). The decoder network was implemented with a similar architecture, although in reverse. In addition, we have determined the activation function at the levels (tanh, relu) and dense layer.

The Variational Autoencoders (VAEs) are important generative models that can generate random, new output data highly similar to the training data. As shown in Fig. 2, just like the AE, the architecture of the VAE has also two main networks; an encoder and a decoder. The encoder network will be trained to minimise the reconstruction error between the encoded-decoded data and the initial data. However, its encoder network outputs two different vectors of size n: a vector of means, μ, and another vector of standard deviations, σ [12]. This enables the decoder network to efficiently reconstruct the training data (see Fig. 2. For the implementation of the VAE model, we have created decoder and encoder networks in addition to the input and output layers. The optimization is performed with the Nesterov Adam optimizer (NADAM).

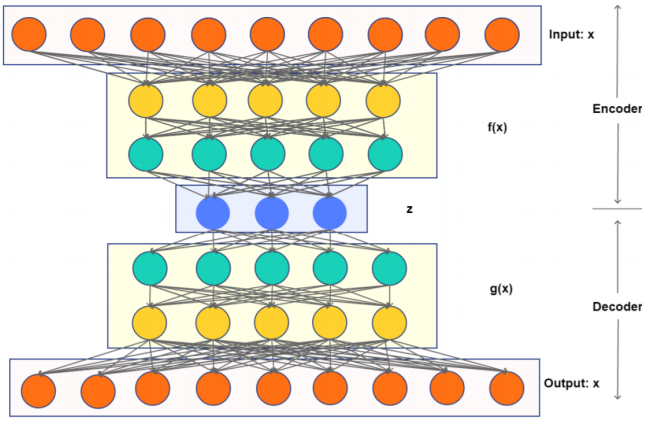

Fig. 2. Variational Autoencoder structure [28]

IV. SYSTEM IMPLEMENTATION & TESTING

In this section, we present the results of the experiments performed over the proposed models using the CERT v4.2 dataset. The experiments were carried out in a virtualised environment. The AE and VEA neural networks were implemented using the Python programming language, TensorFlow library and the open-source library Keras that provides a Python interface for artificial neural networks and the TensorFlow library.

Accuracy refers to the percentage of all correctly classified instances either normal or malicious. In these experiments, normal activities of the user represent positive instances, while malicious activities represent a negative instance. True Positive (TP) is the number of samples that have been correctly classified as normal activities. False Positive (FP) is the number of abnormal instances that have been incorrectly classified as normal activities. True Negative (TN) is the number of samples of abnormal activities that have been correctly classified as anomalous. False Negative (FN) is the number of normal

activities that have been incorrectly classified as anomalous activities.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

Precision represents the percentage of positively classified samples that are truly positive.

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

Recall provides the number of normal activities that were correctly classified. It gives a measure of how accurately our models are able to identify the relevant data.

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

F1-score is a weighted average between precision and recall.

$$F1 - score \times \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

A. Results and discussion

Before running the tests, the training of the AE neural network is done for 30 epochs with a batch size of 256 after running the learning rate function (LRFinder). This function helps to find the optimal values and validate the trained classifiers with just a few experiments. This indicates that the training dataset is perfectly learned without mistakes [29]. For the Variational Autoencoder(VAE) neural network, the training is done for 1000 epochs with a batch size of 128, while the dimension of the latent space is set to 2. After that, many tests were carried out on the trained models to determine their performance.

Fig. 3 presents the overall performance results achieved by the AE and VAE models. As shown in the figure (Fig. 3), VAE clearly outperforms the AE model, where it achieved higher detection performance with an overall accuracy rate of 96%, precision (92%) and recall (96%). Based on these results, the F1-score value is 94%. These values illustrate the efficiency of the VAE in the correct classification of most of the samples. On the other hand, the EA allows slightly lower accuracy detection (95%) and recall (95%) at a cost of a higher false alarm rate (precision rate is 90%).

B. Comparison with other models

The implemented models are compared to our previous work in [11], [12], based on the predefined metrics. In [11], we used the Convolutional Neural Network (CNN) algorithm to identify potential insider threats by using a visual representation of the activity report. The training and testing of the CNN algorithm are done with a dataset of 860 2D images. More specifically, 80% of the samples (i.e., 769 images) were labelled as containing malicious activity and 430 images were labelled as normal. In [12], we examined the efficiency of swarm intelligence algorithms in feature selection optimization. This will greatly enhance the performance of the machine learning model Local Outlier Factor (LOF) in detecting insider threats. In this study, we compare the Bio-inspired model with the nine features "day", "time", "Logon", "Logoff", "Connect", "Disconnect", "email", "file", and "HTTP". The same dataset (CERT dataset) was used for training and testing all the algorithms. The evaluation results are reported in Table III.

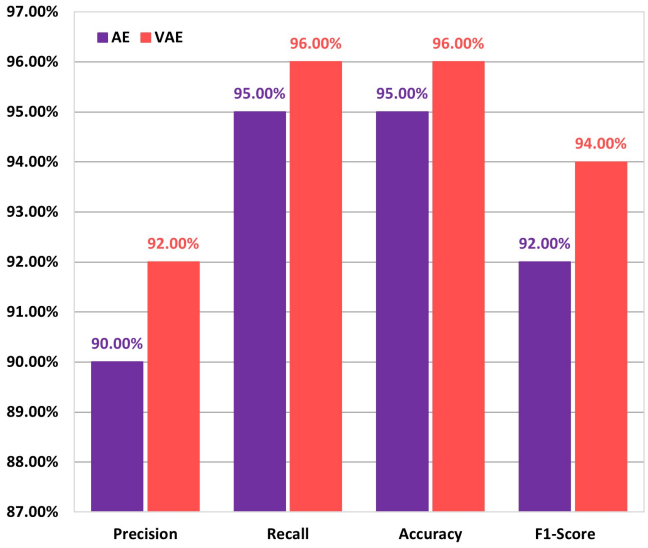

Fig. 3. Evaluation metrics results

TABLE III
COMPARISON WITH OTHER LEARNING ALGORITHMS

	Precision	Recall	Accuracy	F1-Score
Autoencoder (AE)	90%	95%	95%	92%
Variational Autoencoder (VAE)	92%	96%	96%	94%
CNN [11]	n/a	n/a	90%	n/a
Bio-Inspired models [12] (9 features)	n/a	100%	n/a	70%

As presented in Table III, the VAE neural network has achieved the best overall performance compared to other algorithms, with higher accuracy (96%) and recall (96%), and lower false alarms. The CNN model [11] has also achieved very high sensitivity (recall), with a percentage that reached 100%. However, the lack of values for all the evaluation metrics [11], [12] does not help to have a clear view of their performances. Thus, the two models in [11], [12] need to be studied further in future work.

V. CONCLUSION

Insider threat is one of the most complicate security issues that cause significant loss to organisations and businesses. The goal of this work was to examine the utility of using Depp learning techniques to detect insider threats, without human intervention. For that, we have implemented two deep neural

networks autoencoder and Variational Autoencoder to check their efficiency in detecting insider threats automatically. The tests were performed on the public CERT dataset (CERT r4.2). From the tests' results and comparison with other deep learning algorithms, the Variational Autoencoder neural network has proved that it is the most effective in identifying internal threats with an overall accuracy of 96.00%. On the other hand, Autoencoder allows slightly better insider threat detection performance (accuracy 95%) at a cost of higher false alarm rates.

In the future, we intend to perform more experiments on the implemented models by using more data to accurately train and test the neural networks. This can greatly improve the overall accuracy of the proposed models. We also intend to compare the results of our models with the most relevant work in this field and to consider different types of metrics to evaluate them. In this context, it would be good to use the CERT dataset with other deep learning algorithms and compare the results.

ACKNOWLEDGMENT

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 786698. The work reflects only the authors' view and the Agency is not responsible for any use that may be made of the information it contains.

REFERENCES

[1] A. Azaria, A. Richardson, S. Kraus, and V. Subrahmanian, "Behavioral analysis of insider threat: A survey and bootstrapped prediction in imbalanced data," *IEEE Transactions on Computational Social Systems*, vol. 1, no. 2, pp. 135–155, 2014.

[2] 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*, vol. 20, no. 2, pp. 1397–1417, 2018.

[3] C. W. Probst, J. Hunker, M. Bishop, and D. Gollmann, *Insider threats in cyber security*. Springer, 2010, vol. 49.

[4] M. Bishop and C. Gates, "Defining the insider threat," in *Proceedings of the 4th annual workshop on Cyber security and information intelligence research: developing strategies to meet the cyber security and information intelligence challenges ahead*, 2008, pp. 1–3.

[5] D. C. Le, N. Zincir-Heywood, and M. I. Heywood, "Analyzing data granularity levels for insider threat detection using machine learning," *IEEE Transactions on Network and Service Management*, vol. 17, no. 1, pp. 30–44, 2020.

[6] M. B. Salem, S. Hershkop, and S. J. Stolfo, "A survey of insider attack detection research," *Insider Attack and Cyber Security*, pp. 69–90, 2008.

[7] A. Gamachchi, L. Sun, and S. Boztas, "A graph based framework for malicious insider threat detection," *arXiv preprint arXiv:1809.00141*, 2018.

[8] "Insider threat report," Verzion, 2019. [Online]. Available: https://enterprise.verizon.com/resources/reports/insider-threat-report

[9] "2020 cost of insider threats: Global report," Observer IT, 2020. [Online]. Available: https://www.observeit.com/2020costofinsiderthreat/

[10] F. Yuan, Y. Cao, Y. Shang, Y. Liu, J. Tan, and B. Fang, "Insider threat detection with deep neural network," in *International Conference on Computational Science*. Springer, 2018, pp. 43–54.

[11] V. Koutsouvelis, S. Shiaeles, B. Ghita, and G. Bendiab, "Detection of insider threats using artificial intelligence and visualisation," in *2020 6th IEEE Conference on Network Softwarization (NetSoft)*. IEEE, 2020, pp. 437–443.

[12] A. Nicolaou, S. Shiaeles, and N. Savage, "Mitigating insider threats using bio-inspired models," *Applied Sciences*, vol. 10, no. 15, p. 5046, 2020.

[13] F. Meng, F. Lou, Y. Fu, and Z. Tian, "Deep learning based attribute classification insider threat detection for data security," in *2018 IEEE Third International Conference on Data Science in Cyberspace (DSC)*. IEEE, 2018, pp. 576–581.

[14] 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, pp. 503–512, 2015.

[15] W. Li, W. Meng, L.-F. Kwok, and H. Horace, "Enhancing collaborative intrusion detection networks against insider attacks using supervised intrusion sensitivity-based trust management model," *Journal of Network and Computer Applications*, vol. 77, pp. 135–145, 2017.

[16] F. Janjua, A. Masood, H. Abbas, and I. Rashid, "Handling insider threat through supervised machine learning techniques," *Procedia Computer Science*, vol. 177, pp. 64–71, 2020.

[17] T. Rashid, I. Agrafiotis, and J. R. Nurse, "A new take on detecting insider threats: exploring the use of hidden markov models," in *Proceedings of the 8th ACM CCS International workshop on managing insider security threats*, 2016, pp. 47–56.

[18] H. Goldberg, W. Young, M. Reardon, B. Phillips *et al.*, "Insider threat detection in prodigal," in *Proceedings of the 50th Hawaii International Conference on System Sciences*, 2017.

[19] S. Yuan and X. Wu, "Deep learning for insider threat detection: Review, challenges and opportunities," *Computers & Security*, p. 102221, 2021.

[20] D. Zhang, Y. Zheng, Y. Wen, Y. Xu, J. Wang, Y. Yu, and D. Meng, "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*, 2018, pp. 18–20.

[21] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[22] S. Yuan, P. Zheng, X. Wu, and Q. Li, "Insider threat detection via hierarchical neural temporal point processes," in *2019 IEEE International Conference on Big Data (Big Data)*. IEEE, 2019, pp. 1343–1350.

[23] A. Tuor, S. Kaplan, B. Hutchinson, N. Nichols, and S. Robinson, "Deep learning for unsupervised insider threat detection in structured cybersecurity data streams," *arXiv preprint arXiv:1710.00811*, 2017.

[24] T. Hu, W. Niu, X. Zhang, X. Liu, J. Lu, and Y. Liu, "An insider threat detection approach based on mouse dynamics and deep learning," *Security and Communication Networks*, vol. 2019, 2019.

[25] L. Liu, O. De Vel, C. Chen, J. Zhang, and Y. Xiang, "Anomaly-based insider threat detection using deep autoencoders," in *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE, 2018, pp. 39–48.

[26] K. Veeramachaneni, I. Arnaldo, V. Korrapati, C. Bassias, and K. Li, "Aiˆ 2: training a big data machine to defend," in *2016 IEEE 2nd International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing (HPSC), and IEEE International Conference on Intelligent Data and Security (IDS)*. IEEE, 2016, pp. 49–54.

[27] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[28] S. Bhattacharjee, "Variational autoencoder based estimation of distribution algorithms and applications to individual based ecosystem modeling using ecosim," 2019.

[29] L. N. Smith, "Cyclical learning rates for training neural networks," in *2017 IEEE winter conference on applications of computer vision (WACV)*. IEEE, 2017, pp. 464–472.