Proceedings of 10th IOE Graduate Conference
Peer Reviewed
ISSN: 2350-8914 (Online), 2350-8906 (Print)
Year: 2021 Month: October Volume: 10

User Behavior Analytics for Insider Threat Detection using Deep Learning

Santosh Nepal [a], Basanta Joshi [b]

[a, b] *Department of Electronics and Computer Engineering, Pulchowk Campus, IOE, Tribhuvan University, Nepal*
Corresponding Email: [a] nep.santo001@gmail.com , [b] joshibasanta@gmail.com

Abstract

In the field of security analysis of an organization, identifying anomalous activities of user from log data for insider threat detection is difficult as well as important. Identification of such anomalous insider behavior is commonly achieved by use of behavior modeling. This paper presents an approach of one class learning, also known as unary classification or class modelling, where the model is exclusively trained on majority class data. The model learns what a model behavior for an employee of an organization is. The proposed paper attempts to detect the insider threat activities and monitor if any unexpected or suspicious behavior are observed by the model, which produces high reconstruction error within the model and are classified as anomalies. Training of the model implements feature vectors extracted form user log activities in a fixed window of per day. This approach implements Gated Recurrent Unit(GRU) based Autoencoder to model user behavior per day and detect anomalous insider threat points. Since the model is overfitted on normal data, the error produced by normal data is very low while the autoencoder produces high error on malicious class of abnormal data. The dataset used in work is Computer Emergency Response Team(CERT) r4.2 and feature vectors are derived according to the number of times a user performs certain activity within a day is used. Behavior learning through GRU autoencoder is used.

At different threshold, performance of model was measured and the model demonstrated good distinction with minimum mis-classfication for both classes with values of true positive and true negative rates at 79.81%.

Keywords

User Behavior Analytics, Anomaly Detection, GRU RNN Autoencoder, Feature Vectors, Classification Error

1. Introduction

In the world of growing connectivity, information is considered to be one of the most prized assets. Due to this reason, the number of threats that are imposed upon corporate data is also high. These threat vectors, hence, pose serious challenge for protecting information. Threat from external sources have received considerable efforts for prevention through the use of various network components installed, like next-gen firewalls, antivirus programs, Intrusion Detection Systems (IDS), etc. On the other hand, insider threats, which have better knowledge of the critical assets within the organization and increased access, are difficult to detect and stop by network components, as these act as a legitimate user and often go undetected. This has encouraged increased amount of insider threats. Compromised users in Advanced Persistent Threat (APTs), careless employees using unsecured application service account instead of named account, users with malicious intents, spies from other organizations and dissatisfied employees constitute are some of the cases which are identified as insider threat[1]. According to latest insider threat report by Gurucul, regardless of the origin, action taken by them potentially harm organization and 49% of those organizaion have no effective detection of insider threat in place[2]. Hacking trails for outsiders are hard to hide whereas malicious insiders are equally difficult [3] to detect based on the signature based profiles of the users.

Effective use of security policies, procedures and controls are preventive measures and detection of these insider threats is measure of minimizing the impact possessed by these threat vectors[1]. For effective detection of user's anomalous behavior within a network, we need to collect his/her activity data over some time thus analyze the pattern [4].

Detection of feed forward traditional machine learning is deemed insufficient as in feed forward networks which cannot remember past inputs and results. Hence, using an RNN with backpropagation through time for long sequences of input is used. Use of gated structures in the recurrent units, as in LSTM and GRU models helps to control the error value propagation and hence prevent vanishing gradient problem using gated mechanism[5]. In this work, an autoencoder model with GRU as processing unit is used for anomaly detection with one class classification.

2. Related Literature

Profiling users for detecting anomalies is an important factor in UBA, where by user activities like login patterns, various application use and website visits function as digital forensic evidence. Signature based algorithms have proved to be ineffective in threat detection due to evolving and rapidly growing attacks and their variants, as reviewed in [6]. Pokharel et.al.[7] proposed anomaly based intrusion detection using hybrid SVM and Naïve Bayes algorithm in their literature featuring user built activities. Rashid et.al. in their literature [8] detailed how Hidden Markov Model(HMM) can be implemented. This uses statistics to detect deviated form normal expected values, however the states needed to be predetermined. Computational complexity increased significantly with increasing number of hidden states in this model. Similarly, various distance measurement techniques with Jaccard, DL and Cosine distance, on HMM was implemented by Owen Lo et.al. [9] to identify change in behavior. were studied and existing approaches based on adopted deep learning architecture were evaluated in literature In literature [10] [14], extensive study of various techniques of Deep Neural Network(DNN) were studied and evaluated and also insights into lot of improvements which can be made to existing models with RNN and Reinforcement Neural Networks. Hu et.al in [11], used multiple log source events which were correlated, as Active Directory(AD),Virtual Private Network(VPN) product, data security products in building user profiles which was an important technique. Significant task on UBA platform with multi algorithm technique combining One Class Support Vector Machine(OCSVM), RNN and isolation forest was used on aggregated data source was done in literature [12] which showed improved effectiveness over individual method. Vanishing gradient problem, where model could not remember past events, resulted in difficulty to train RNN. This hampered to capture term dependencies[12] and subsequently, long short-term memory (LSTM), a recurrent unit, was implemented to rectify this drawback. An sophisticated and enhanced activation function was proposed by Chung et.al. in[13], consisting of affine transformation. Works on network anomaly for IDS can also be seen in literature [14], where Azure tools have used in anomaly detection event. In literature [15] LSTM to model insider activity log is used , where extraction of features for detecting anomalies when log patterns deviate from their trained models. In time series events, significant accuracy is observed in literature [16] to detect anomaly by LSTM using encoder decoder. Using auto encoder in [4] for similar purpose with reference as reconstruction error implementing flexible session period based time window. Comparable to LSTM performance in polyphonic music datasets, proposed architecture of GRU [13], resulted in improvement over traditional tanh model and even outperformed LSTM units in CPU time and parameters update for fixed number of parameters. This, however lacks proper mechanism for fine tuning due to less control, and exposes whole state each time.

3. Methodology

3.1 System Architecture

For modelling insider activities, using GRU is proposed. In past literature, it has which has produced good results for cases where variable length input/output units are used[17] being structurally simple,and with rapid training phase over LSTM [13]. In the standard dataset CERT, version r4.2, are logs from different sources.The events are placed in time basis representing user activities per day on separate files. Aggregating these events, we use numerical representations with frequency aggregation of different these events, which represent a feature vector. These feature vectors are used to train the model. The training set consists exclusively of only normal data This will cause the model to learn to the behavior known as pseudo non-anomalous.

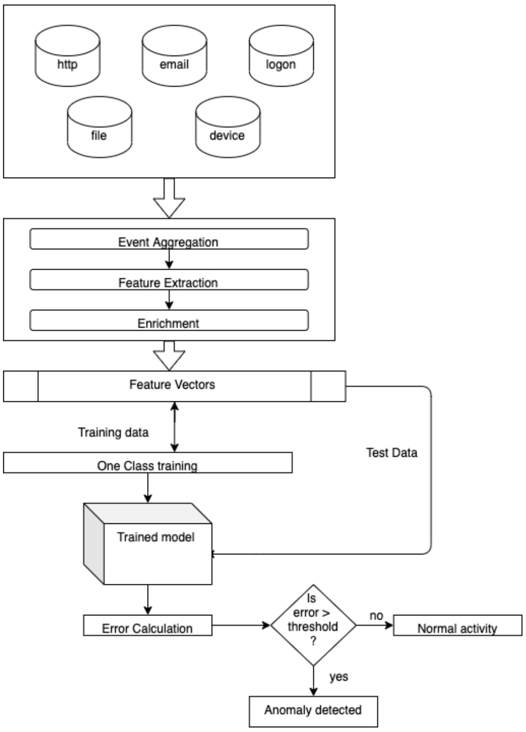


Figure 1: System Architecture

Using an auto-encoder architecture, implementing anomaly detection is achieved.The input feature vector is time-series of n dimension. The encoder model then reads input in sequence to produce n dimensional intermediate encoding. This is used by the decoder to reconstruct the input for minimizing the reconstruction error. User Behavior data patterns can be analysed for a user with iteration over the process.

During testing, both the sets of feature vectors representing are given as input to the neural network. The entries which produces high error values at output are considered anomalous. Mean absolute error (mae) as error parameter is used where there are limited to none outliers in data or in a better way we ignore the outliers while fitting model to data.

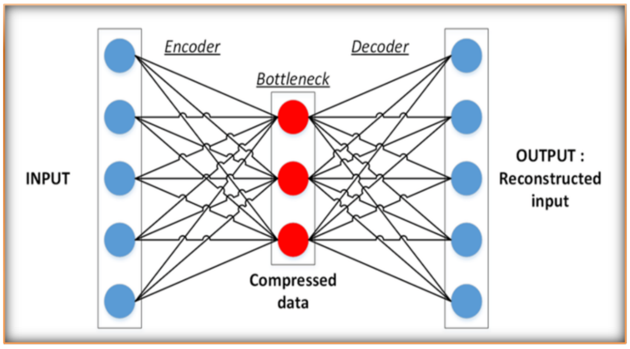


Figure 2: Auto-encoder Model

$$\Delta(x) = |x_i - x| \tag{1}$$

Equation representing mae is:

$$mae = \frac{1}{n}\sum_i |x_i - x| \tag{2}$$

Based on the mae value(score value), the test feature vectors will be labelled as either as normal or malicious.

$$f(x) = \begin{cases} true(anomalous), & \text{if } score \geq threshold \\ false(normal) & otherwise \end{cases} \tag{3}$$

3.2 Log Data Source

The data set for user behavior modeling and evaluation the project is CERT r4.2 insider threat dataset[18]. The logs present comprise various data sets. Such sources include the following sources. Following datasets are used for algorithm development and parameter optimizations.

Table 1: Event Source and Statistics

Log Source	Event Count
http.csv	28,434,424
device.csv	405,381
device.csv	405,381
logon.csv	854,860
file.csv	445,582
psychometric.csv	1000

Featuring 1000 distinct users for 500 users, dataset records these events where no of malicious synthetically injected event is 7323 out of total 32,770,227 events.
Insider Threat scenarios[9]:

1. User with no use of removable device and off hour logins, starts these activities and uploads data to suspicious websites before leaving company

2. Search domain of user consists job seeking sites and starts high amount data transfer to thumb drive for stealing data

3. Unsatisfied system admin download key logger software and use a thumb drive to transfer

spyware to higher privilege account user computer. Uses the key logs for unauthorized access to send alarming email messages causing upheave in company and leaves it.

For feature selection, we extract values from files and convert then into some unique value corresponding to each user that accurately models training, validation and testing.Numerical features are extracted form the log files while the categorial features, present in psychometric files are enrichment for each user. Based on the research and different literature[19, 8, 10, 20, 21], the features were selected for primary consideration. The feature considered is flexible but will have impact in the model preparation. After the feature vector has been obtained, the data for each user will be normalized in the range of (0,1). For this, statistical max model scaling will be used. For a particular user U and any feature

$$X_{(i,m)} = \frac{X_{(i,m)}}{(X_m)_{max}} \tag{4}$$

3.3 GRU Unit

Due to short comings in vanilla RNN models, LSTM model was widely used for time sequence based temporal data analysis. GRU, variant of LSTM, is a gated model structure where the model selects gated structures in place of tanh value traditional models. Opposed to the LSTM, forget along with input gate is merged into single update gate.If X_t, Y_t and H_t be input,output and hidden cell state vectors then:

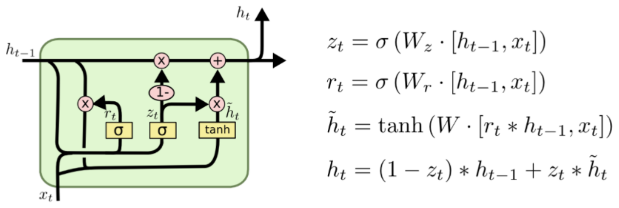

h_t
h_{t-1}
x_t
r_t
z_t
$\tilde{h}_t$
1-
σ
σ
tanh
$z_t = \sigma(W_z \cdot [h_{t-1}, x_t])$
$r_t = \sigma(W_r \cdot [h_{t-1}, x_t])$
$\tilde{h}_t = \tanh(W \cdot [r_t * h_{t-1}, x_t])$
$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$

Figure 3: GRU unit[22]

where, x_t is the input value of a GRU at a time t; h_t is output value of a GRU at a time t; h_{t-1} represents previous instance of current time. Reset gate and update gate are key structures that keep the output through them between 0 and 1 with sigmoid function activation. The multilayered GRU structure is used where arbitrary number of hidden layers can be used.

3.4 Evaluation Metrices

		Predicted Class	
		Normal(-)	Anomaly(+)
Actual	Normal(-)	TN	FN
	Anomaly(+)	FP	TP

Table 2: Confusion Matrix

Based on this table, different performance parameters are calculated in evaluation of model.

True positive Rate OR Recall:

$$TPR = \frac{TP}{TP+FN} \tag{5}$$

False Positive Rate (FPR):

$$FPR = \frac{FP}{TN+FP} \tag{6}$$

Accuracy:

$$Accuracy = \frac{TP+TN}{TN+FP+FN+TP} \tag{7}$$

Precision:

$$Precision = \frac{TP}{FP+TP} \tag{8}$$

F1 score:

$$F1score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{9}$$

ROC:Receiver Operating Characteristic Curve The graph representing cost against benefit to show the relationship between Recall(TPR) and False Postive rate at different threshold values is called ROC curve. This relation between these two quantities differs and plotting these values, we can get a curve. The area covered with the curve is directly proportional to the performance of the model.

3.5 Tools and Libraries Used

Python and related libraries like seaborn numpy, keras, matplotlib, Google Colab, tensorflow, Overleaf

4. Results and Analysis

To achieve the objectives of the project, preparation of suitable data for training a neural network model has achieved a lot of work and the following progresses have been made.

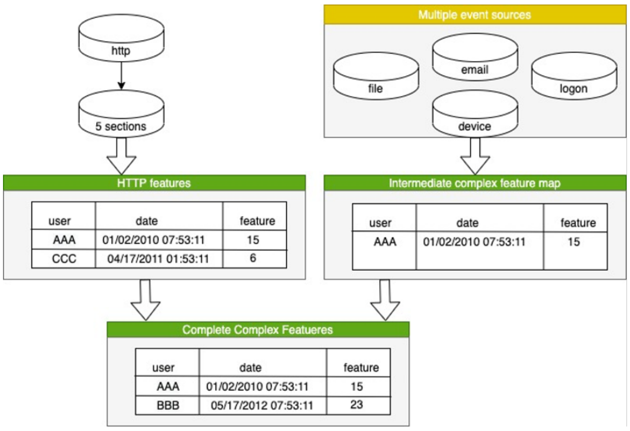


Figure 4: Composite Feature Extraction

Through feature mapping function, data from various sources were passed in different stages.

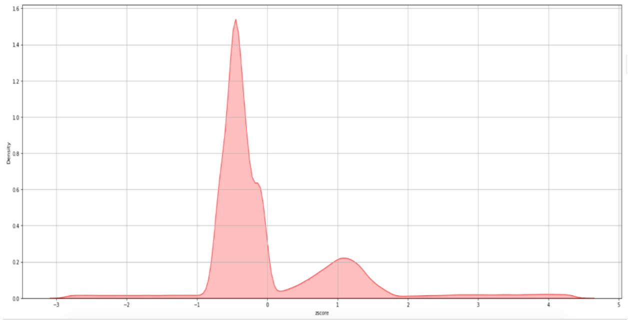


Figure 5: Logon Distribution Curve

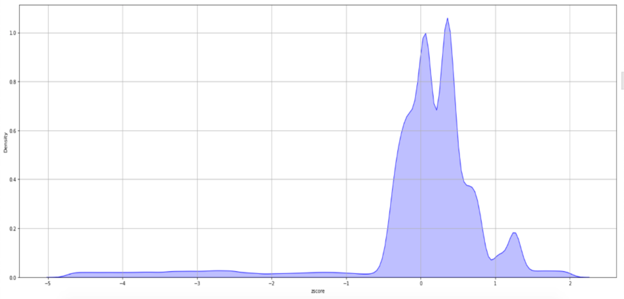
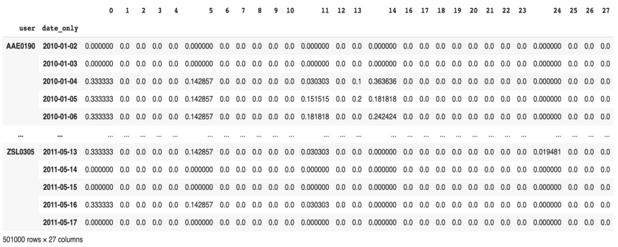


Figure 6: Logoff Distribution Curve

Data set presented time period of 9am-5pm as standard office time.This cutoff period cannot be strictly implemented and results were poorer. For adjusting the new cutoff period, probability distribution chart was constructed and with 1 standard deviation, the normal shifted by 3 hrs: 7am-7pm. The composite feature table was then maintained in multi-index data frame where the frequency distribution table was constructed. The values within this data frame were normalized using max normalization to get to a numerical value of 0-1.



user	date_only	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	16	17	18	19	20	21	22	23	24	25	26	27
AAE0190	2010-01-02	0.000000	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0
	2010-01-03	0.000000	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0
	2010-01-04	0.333333	0.0	0.0	0.0	0.0	0.142857	0.0	0.0	0.0	0.0	0.0	0.030303	0.0	0.1	0.363636	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0
	2010-01-05	0.333333	0.0	0.0	0.0	0.0	0.142857	0.0	0.0	0.0	0.0	0.0	0.151515	0.0	0.2	0.181818	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0
	2010-01-06	0.333333	0.0	0.0	0.0	0.0	0.142857	0.0	0.0	0.0	0.0	0.0	0.181818	0.0	0.0	0.242424	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
ZSL0305	2011-05-13	0.333333	0.0	0.0	0.0	0.0	0.142857	0.0	0.0	0.0	0.0	0.0	0.030303	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.019481	0.0	0.0	0.0
	2011-05-14	0.000000	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0
	2011-05-15	0.000000	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0
	2011-05-16	0.333333	0.0	0.0	0.0	0.0	0.142857	0.0	0.0	0.0	0.0	0.0	0.030303	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0
	2011-05-17	0.000000	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0

501000 rows × 27 columns

Figure 7: Feature Vector

The details implemented are as follows for the auto-encoder model. These parameters were determined by parameter tuning across multiple tests and experimentation.

Table 3: Event Source and Statistics

Parameter	Values
Activation Function	ReLU
Optimizer	Adam
Learning Rate	0.001
Decay rate	1e-6
Loss Function	Mae
No.of Epoch	100
Batch Size	256

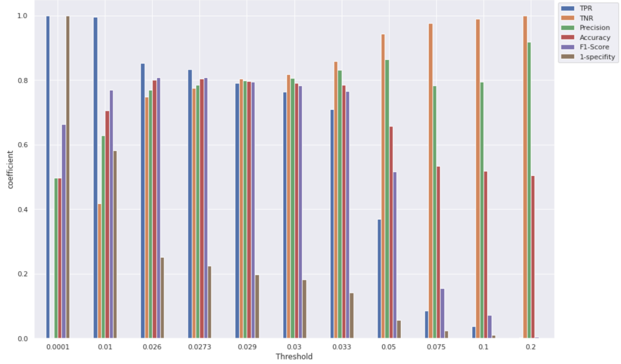


Figure 8: Model Performance at Different Threshold

Based on different value of threshold for anomaly detection, the following values as shown in Figure 8 were calculated. It can be observed from the chart that at lower threshold the model is classifying most of the anomalous events correctly but it fails to efficiently classify negative values. As the threshold value is gradually increased, the overall performance of model in distinguishing both classes of data increases significantly and is found to be optimum at near 0.03. ON further increasing the threshold, the classification of true positives decreases and overall performance of model is also decreased. Based on this varying values, threshold for optimum classification i.e. the point where TPR and TNR chart intersect each other is calculated as shown in the Figure 9. The precision of model was 80.1% and f1 score was 79.4% when the misclassification for both the classes were minimized.

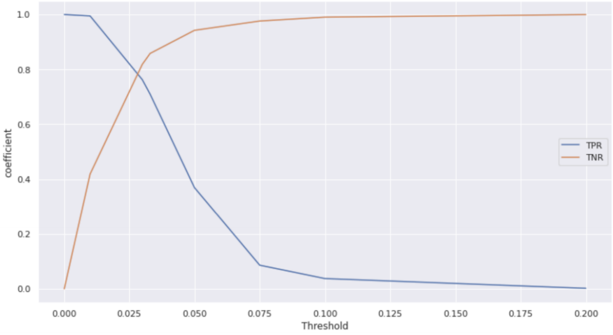


Figure 9: Optimum Threshold Calculation

At optimum threshold, the following ROC curve 10 was obtained with AUC values of 0.8732.

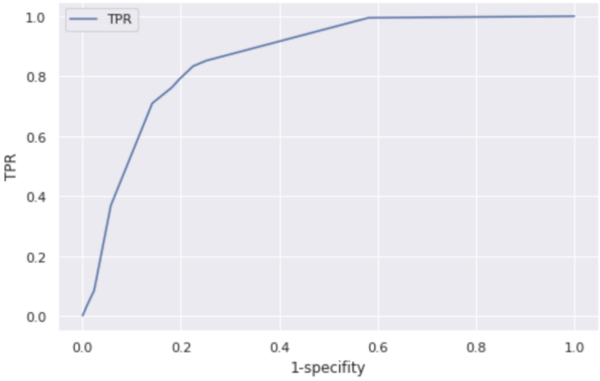


Figure 10: ROC curve

5. Conclusions

This research works presents use of one class modelling on normal data of CERT r4.2 data set. The model overfitted on non-malicious normal during training period data produces high reconstruction error when malicious samples are fed during the testing period. The work focuses on using GRU units in place of traditional LSTM units for use in autoencoder model for modelling non-malicious user behavior.

6. Future Enhancements

The GRU unit is used in place of LSTM unit. In future we intend to compare performance of these two models and research if it is helpful in any way by this replacement other than ease of implementation. Also based on ground truth, we intend to see how the new proposed model fares in case of different usecases defined in the dataset.

References

[1] Challenges of Detecting Insider Threats - Whiteboard Wednesday, July 2017.

[2] 2021 insider threat report gurucul.pdf, July 2017.

[3] 5 Real-Life Examples of Breaches Caused by Insider Threats, November 2020.

[4] Balaram Sharma, Prabhat Pokharel, and Basanta Joshi. User Behavior Analytics for Anomaly Detection Using LSTM Autoencoder - Insider Threat Detection. In *Proceedings of the 11th International Conference on Advances in Information Technology*, pages 1–9, Bangkok Thailand, July 2020. ACM.

[5] How do Long Short Term Memory (LSTM) and Gated Recurrent Unit (GRU) work in Deep Learning?

[6] Ansam Khraisat, Iqbal Gondal, Peter Vamplew, and Joarder Kamruzzaman. Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecurity*, 2(1):20, December 2019.

[7] Roshan Pokhrel, Prabhat Pokharel, and Arun Kumar Timalsina. Anomaly-Based – Intrusion Detection System using User Profile Generated from System Logs. *International Journal of Scientific and Research Publications (IJSRP)*, 9(2):p8631, February 2019.

[8] Tabish Rashid, Ioannis Agrafiotis, and Jason R.C. Nurse. A New Take on Detecting Insider Threats: Exploring the Use of Hidden Markov Models. In *Proceedings of the 2016 International Workshop on Managing Insider Security Threats - MIST '16*, pages 47–56, Vienna, Austria, 2016. ACM Press.

[9] Owen Lo, William J. Buchanan, Paul Griffiths, and Richard Macfarlane. Distance Measurement Methods for Improved Insider Threat Detection. *Security and Communication Networks*, 2018:1–18, 2018.

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

[11] Liu Liu, Chao Chen, Jun Zhang, Olivier De Vel, and Yang Xiang. Insider Threat Identification Using the Simultaneous Neural Learning of Multi-Source Logs. *IEEE Access*, 7:183162–183176, 2019.

[12] Xi Xiangyu, Tong Zhang, Dongdong Du, Guoliang Zhao, Qing Gao, Wen Zhao, and Shikun Zhang. Method and System for Detecting Anomalous User Behaviors: An Ensemble Approach. pages 263–307, July 2018.

[13] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *arXiv:1412.3555 [cs]*, December 2014. arXiv: 1412.3555.

[14] Janardan Bhatta, Kushal Gajurel, Santosh Nepal, Saurav Pandey, and Shekhar Koirala. Anomaly based Intrusion Detection System. 2019. Publisher: Unpublished.

[15] Dongxue Zhang, Yang Zheng, Yu Wen, Yujue Xu, Jingchuo Wang, Yang Yu, and Dan 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 - SecArch'18*, pages 18–20, Toronto, Canada, 2018. ACM Press.

[16] Pankaj Malhotra, Anusha Ramakrishnan, Gaurangi Anand, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. LSTM-based Encoder-Decoder for Multi-sensor Anomaly Detection. *arXiv:1607.00148 [cs, stat]*, July 2016. arXiv: 1607.00148.

[17] Alex Graves. *Supervised Sequence Labelling with Recurrent Neural Networks*, volume 385 of *Studies in Computational Intelligence*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[18] Insider Threat Test Dataset.

[19] Derek Lin. Insider threat detection: Where and how data science applies. *Cyber Security*, 2:8, 2018.

[20] Malvika Singh, B.M. Mehtre, and S. Sangeetha. User Behavior Profiling using Ensemble Approach for Insider Threat Detection. In *2019 IEEE 5th International Conference on Identity, Security, and Behavior Analysis (ISBA)*, pages 1–8, Hyderabad, India, January 2019. IEEE.

[21] Abien Fred Agarap. A Neural Network Architecture Combining Gated Recurrent Unit (GRU) and Support Vector Machine (SVM) for Intrusion Detection in Network Traffic Data. *Proceedings of the 2018 10th International Conference on Machine Learning and Computing - ICMLC 2018*, pages 26–30, 2018. arXiv: 1709.03082.

[22] Understanding LSTM Networks – colah's blog.