IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 25, 2021, accepted September 26, 2021, date of publication October 5, 2021, date of current version October 27, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3118297

Behavioral Based Insider Threat Detection Using Deep Learning

RIDA NASIR[1], MEHREEN AFZAL[1], RABIA LATIF[2], AND WASEEM IQBAL[1]
[1]Department of Information Security, National University of Sciences and Technology (NUST), Islamabad 44000, Pakistan
[2]College of Computer and Information Sciences, Prince Sultan University, Riyadh 11586, Saudi Arabia

Corresponding author: Waseem Iqbal (waseem.iqbal@mcs.edu.pk)

This work was supported by the Artificial Intelligence Data Analytics Lab (AIDA) CCIS, Prince Sultan University, Riyadh, Saudi Arabia.

ABSTRACT The most detrimental cyber attacks are usually not originated by malicious outsiders or malware but from trusted insiders. The main advantage insider attackers have over external elements is their ability to bypass security checks and remain undiscovered, this may cause serious damage to the organizational assets. This paper focuses on insider threat detection through behavioral analysis of users. User behavior is categorized as normal or malicious based on user activity. A series of events and activities are analyzed for feature selection to efficiently detect adversarial behavior. Selected feature vectors are used for model training during the implementation phase. A deep learning based approach is proposed that detects insiders with greater accuracy and low false positive rate. A rich event / user role based feature set containing Logon/Logoff events, User_role, Functional_unit etc are used for detection. The dataset used is the CMU CERT synthetic insider threat dataset r4.2. Performance of our proposed algorithm has been compared to other well-known techniques i.e. long short term Memory- convolutional neural network, random forest, long short term memory- recurrent neural network, one class support vector machine, Markov chain model, multi state long short term memory & convolutional neural network, gated recurrent unit & skipgram. The comparison proved that our novel approach produces relatively good accuracy(90.60%), precision(97%) and F1 Score (94%).

INDEX TERMS Insider threat, deep learning, machine learning, user behavior, information security.

I. INTRODUCTION

One of the most basic, yet hard to solve problem in cyber security is the identification of adversarial behavior. The exploitation and leakage of sensitive data and information by malicious insiders is getting worse day by day. According to ''Insider Report 2018'' 90% of the organizations are prone to insider attack [1]. Around 60% organizations encountered one or more insider attacks in 2019 [2]. Since an insider has authorized access to an organization assets, therefore they might have better opportunity to undermine the confidentiality, availability or integrity of data than an external attacker. Various primary and secondary elements that may serve as an inspiration for an insider includes financial gain or greed, revenge, anger, thrill, pressure, treachery, discontentment, jealousy, organizational politics and acknowledgement [3] as shown in Figure 1.

FIGURE 1. Insider attack motivations.

Mostly ''Insider Threat'' is associated with malicious employees who aim to harm the company by theft and vandalism. However in actual, negligent and careless

The associate editor coordinating the review of this manuscript and approving it for publication was Mervat Adib Bamiah.

employees might often accidentally cause a high impact damage (66%) [2]. Many types of insiders can be found in theory, as shown in Figure 2.

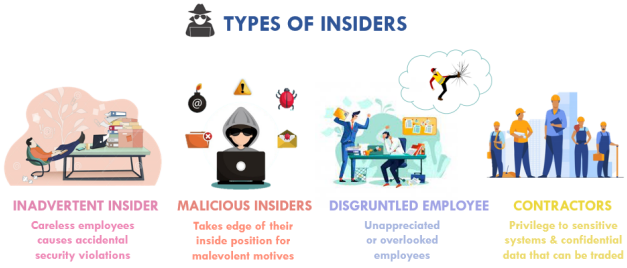

FIGURE 2. Insider types.

The biggest insider threat actors are regular employees (49%) and privileged IT users (59%) followed by contractors (52%) [2] as shown in Figure 3. The main enablers of insider attack are the increased number of users with unnecessary access rights, increasing devices accessing confidential data, rapid increase in the technological complexities, lack of user awareness & training and increase in sensitive data. The most upsetting truth is that the potential loss caused by successful insider attack ranges from $100,000 to $500,000. [1].

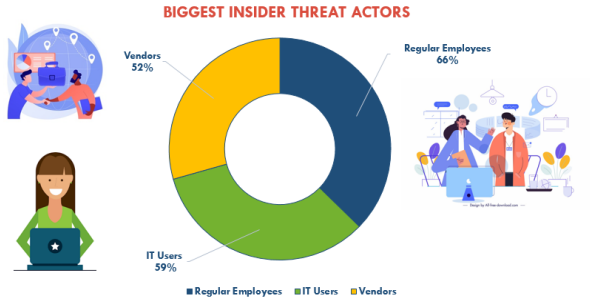

FIGURE 3. Biggest insider threat actors.

Deep learning (DL) is a trending research topic and is being applied in various security frameworks due to its enormous advantages. It can be used in both supervised and unsupervised manner. DL is a subfield of machine learning which has enormous advantages. The algorithms outperform traditional machine learning algorithms in both performance and accuracy. Therefore, DL based algorithms can be used to improve insider detection and results can be obtained with high accuracy and lower false positives. This can result in enabling organizations to have a robust insider threat detection mechanism.

This research focuses on user behavior based insider detection. Based on the user's activity, their behavior is categorized as either normal or malicious. For feature selection, a set of events and activities are analyzed to detect malicious behavior efficiently. In the implementation phase the model is trained with the selected feature vectors. DL based approach is proposed to detect insiders with higher accuracy and low false positive rate. In the training phase, the model is trained using the normal user behavior and any deviation from the normal behavior is classified as malicious or insider in the testing phase.

Our research stands out because of the following main contributions:

- A detailed literature review of the existing techniques for insider detection using machine learning.
- The proposal of a novel deep learning based insider detection technique which is simple and yet not processing and memory intensive. The feature set is richer as compared to many existing researches thus giving better results.
- A comparative study of existing techniques with our proposed model.

II. A REVIEW OF INSIDER DETECTION TECHNIQUES USING ML AND DL

Insider detection techniques using machine and deep learning can be broadly categorized in user behavior based detection and graph based detection. Both techniques have multiple models. Few other techniques also exist which are discussed below.

A. USER BEHAVIOR BASED INSIDER DETECTION TECHNIQUES

In user behavior based detection, user behavior is categorized in to two types, that is normal and malicious. Each user behavior is logged and is compared with a standard rule set (created by experts). If the behavior deviates from normal, it is considered malicious.

A supervised time series based solution using two layer deep auto-encoder can also be used for insider attack detection [4]. A technique using LSTM-CNN algorithm has been shown to identify user anomalous behavior in [5], by monitoring user activities and extracting temporal features. While in [6], detection algorithm XGBoost has been used and behavior characteristic features are extracted from audit logs. Technique proposed in [7] extracted features and fields from user behavior logs for behavior auditing, and then these log files are used to train the Improved Hidden Markov Model (IHMM) for detection of malicious behavior. Random forest algorithm can be used for behavior analysis of individual user by analyzing its activities over a period of time [8]. A user sentiment profile has been designed in [9] to give a prediction scheme using user's network browsing content and emails. A framework known as ''Insider Catcher'' is proposed in [19]. The proposed framework uses LSTM, a deep learning technique to model system logs as an organized sequence. Work done in [12] uses distance measurement techniques (DL distance, Jaccard Distance and Cosine Distance) through analysis of user activity for insider detection. Kernel PCA and LSTM-RNN for insider detection is proposed in [14].

Work done in [18] used an unsupervised DBN for unseen feature selection, from multi-domain features obtained from logs. Extracted features from DBN are used to train a One-Class SVM (OCSVM). In [21], user behavior data has been extracted by examining shell commands flow, keystrokes and mouse functions during GUI interaction. De-noising auto-encoders has been used for encoding user log file in [24], while anomalous data has been identified using integrated methods like GMM, buck covariance, OCSVM, isolation forest and local outlier factor. In [29] Multi State Long Short Term Memory (MSLSTM) and CNN based hybrid ML approach has been introduced which works by using time series anomaly detection method for outlier detection in user behavioral patterns. Aspect based sentiment analysis and social network information of the user using hybrid DL techniques such as Gated Recurrent Unit (GRU) and skip-gram are proposed in [30] to detect insider threat. Variational autoencoders and deep autoencoders are used for insider detection in [39]. A Hybrid approach AD-DNN is used in [40] which uses an adaptive synthetic approach for class imbalance problem and Deep neural network for detecting insider threat.

B. GRAPH BASED INSIDER DETECTION TECHNIQUES

With the technological advancement, user's data is now heterogeneous and multi dimensional. The heterogeneous data generated from various sources consists of network activity, psychological factors, organizational dynamics, employee behavior etc. the data thus forming structural patterns. For detecting insider threat in this complex, structural and heterogeneous data, graph based approach is used.

Technique proposed in [20] detects the malicious conduct of an employee based not only on its own activities but also on the malicious activities of the employees with same job roles. Prospective insiders are recognized by designing a graph signal processing technique. Employees normal data usage patterns are reported and compared to identify anomaly in [25]. An insider attack detection mechanism using Gaussian Mixture model is proposed in [11] which included security expert knowledge and other non-technical indicators of insider threat as key elements of the system. A graph analysis and anomaly detection based hybrid framework, which consists of two modules ''Graphical Processing Unit'' (GPU) and ''Anomaly Detection Unit'' (ADU) for insider threat detection has been given in [13].

Attributed graphs for showing high dimensional, diverse data are used for insider threat detection [35]. A framework which uses the combined approach of Structural Anomaly Detection (SA) and Psychological Profiling (PP) of users for insider threat detection is proposed in [36].

C. INSIDER ATTACK DETECTION USING OTHER TECHNIQUES

A network based insider attack flexible approach ''Gargoyle'' is proposed in [15]. The trustworthiness of the context of an access request is evaluated through a new set of contextual attributes called Network Context Attribute (NCA), information such as the user's device capacity, security-level, network connection status etc. are obtained from network traffic analysis. Network packet inspection has been used for insider threat detection [16], while honey pot sensors have been used within the company's local network to detect insiders [22]. The model proposed in [23] works by properly designing rules and regulations into complex events and investigating whether employees conduct conforms to the rules and regulations.

To predict and detect insider threat, disturbing psychological patterns of individual users are obtained by analyzing electronic communications in [32]. A state machine system is proposed in [34] that can efficiently integrate policies from rule based anomaly detection systems in order to create models which are followed by the insiders to launch an attack.

D. A COMPARATIVE STUDY OF EXISTING TECHNIQUES

The overview and comparison shown in the Table 1 & 2 respectively gives a clear picture of various ML and DL techniques used for insider threat detection. Some of the techniques are very efficient, but have some deficiencies in terms of complexity, performance evaluation metrics and evaluation dataset. Some models are not evaluated on real life scenarios and are processing & memory intensive. Some have relatively small test data, which does not fully evaluate the performance of the technique.

It is also observed that most Role based or Behavior based techniques produce significant quantitative results as compared to graph based and other techniques. The most widely used technique is LSTM [5], [10], [14], [19], [29], [32]. Another widely used technique is Deep AutoEncoders [4], [24], it has the ability to be used on real valued datasets and are quick & concise. Keeping in view the above discussion, a novel hybrid DL approach has been designed in this research to detect insiders efficiently, with low processing and memory requirements, with low false positive rate and higher accuracy.

III. PROPOSED SCHEME FOR INSIDER THREAT DETECTION

We are using **"LSTM-Autoencoder"** for insider threat detection. Our proposed approach consists of multiple stages. At first user data is gathered from various csv files, the data is then processed and a rich event /user_role based feature set containing Logon/Logoff events, user_role, user_functional_unit, user_department, user_activity etc are extracted. Selected features are then used for model training and evaluation. An overview of our approach can be seen in Figure 4.

Insider threat dataset has a multivariate time-series data which consists of several variables discovered over a time interval. On this multivariate time-series data for insider attack detection an **LSTM Autoencoder** has been built.

LSTM is very effective in natural language processing. It can automatically learn features. It is very good in process-

TABLE 1. Overview of insider detection techniques.

Sr.	Paper Ref.	Approach			Technique			Remarks
		Behavior Based	Graph Based	Other	ML	DL	Hybrid	
1	[6]	✓	✗	✗	✗	✗	✓	Claims to be the first one to use XGBoost algorithm for insider detection.
2	[5]	✓	✗	✗	✗	✗	✓	Experimental setup, platform used, memory and processing requirements are all missing
3	[8]	✓	✗	✗	✓	✗	✗	The evaluation is performed on both CMU-CERT and private datasets NextLabs.
4	[9]	✓	✗	✗	✗	✗	✓	Sentiment classification accuracy of 100% for http and 96% for email content is achieved.
5	[14]	✓	✗	✗	✗	✓	✓	The proposed scheme produces good results with a precision=95.12% and accuracy=93.85% .
6	[18]	✓	✗	✗	✓	✗	✗	User characteristics and domain understanding is not required, and the technique is purely data driven.Performance evaluation scale are missing.
7	[17]	✓	✗	✗	✗	✓	✗	Evaluation platform , memory & processing requirement and model train and test time are all missing
8	[24]	✓	✗	✗	✗	✗	✓	The performance of the technique is compared with an existing technique Pearson-OCSVM
9	[12]	✓	✗	✗	✗	✗	✓	Real life scenario evaluation of the model is missing. DM techniques only compare previous week user activity with the current week, which creates high significance of false positives.
10	[29]	✓	✗	✗	✗	✓	✓	The platform used for evaluation is not mentioned.
11	[30]	✓	✗	✗	✗	✗	✓	The dataset used was publically available email dataset ENRON. Comparison with other latest techniques is missing.
12	[31]	✓	✗	✗	✓	✗	✗	Only 15% of the CMU dataset was used for evaluating the model
13	[11]	✗	✓	✗	✓	✗	✗	Proposed scheme looks complex and time intensive.
14	[13]	✗	✓	✗	✓	✗	✗	The scheme does not take into account the social behavior, content analysis of emails and web browsing of users
15	[25]	✗	✓	✗	✗	✗	✓	Enterprise graph database Neo4j is used for analyzing and visualizing anomalous patterns.
16	[35]	✗	✓	✗	✗	✗	✓	The approach used is complex and difficult to understand
17	[36]	✗	✓	✗	✗	✗	✓	Dataset used is generated from a multi-player online game, World of Warcraft (WoW).
18	[16]	✗	✗	✓	N/A			A graph based approach, in which a weight is assigned to each activity. Wireshark is used for traffic capturing at proxy server.
19	[22]	✗	✗	✓	N/A			Use of honey pot sensor for keeping track of system calls in real time is cost effective.
20	[34]	✗	✗	✓	✗	✗	✓	Effective with minimal time and memory requirements. Capable to produce real time alerts.

TABLE 2. Comparison between different ML & DL techniques.

Sr.	Paper Ref.	Algorithm	Performance Matrices					Dataset Used		
			Accuracy	Precision	F-Score	AUC	Graph	CMU-CERT V4.2	CMU-CERT V6.2	Other
1	[6]	XGBoost	✓	✗	✓	✗	✗	✗	✓	✗
2	[5]	LSTM-CNN	✗	✗	✗	✓	✗	✓	✗	✗
3	[8]	Random Forest	✓	✗	✗	✗	✗	✓	✗	✓
4	[9]	Dictionary method & CNN	✓	✗	✗	✗	✗	✓	✗	✗
5	[14]	LSTM-RNN	✓	✓	✗	✗	✗	✗	✓	✗
6	[18]	One-Class SVM	✓	✗	✗	✗	✗	✓	✗	✗
7	[17]	Deep Autoencoder	✗	✗	✗	✗	✗	✗	✓	✗
8	[24]	Denoising autoencoders, GMM	✗	✗	✗	✗	✗	✗	✗	✗
9	[12]	Distance Measurement Techniques	✗	✗	✗	✗	✗	✗	✗	✗
10	[29]	Multi State LSTM & CNN	✗	✗	✗	✓	✗	✗	✗	✗
11	[30]	Gated Recurrent Unit & Skipgram	✗		✗	✓	✗	✗	✗	✓
12	[31]	Markov chain model	✓	✗	✗	✗	✗	✗	✗	✗
13	[11]	GMM	✓	✓	✓	✗	✗	✓	✗	✗
14	[13]	Isolation Forest	✓	✗	✗	✗	✗	✓	✗	✗
15	[25]	GBAD & Neo4j	✗	✗	✗	✗	✓	✗	✗	✓
16	[35]	EDCAR & GAMER	✗	✗	✗	✗	✓	✗	✗	✗
17	[36]	Structural Anomaly Detection (SA) and Psychological Profiling (PP)	✗	✗	✗	✗	✓	✗	✗	✓
18	[16]	Network Packet Inspection	✗	✗	✗	✗	✓	✗	✗	✓
19	[22]	Honeypot sensors	✗	✓	✓	✗	✗	✗	✗	✓
20	[34]	State Machine System	✗	✗	✓	✗	✓	✗	✗	✗

ing sequence and time series data due to which it is used to model user behavior. While autoencoder has the ability to be used on real valued datasets and are quick & concise.

LSTM auto-encoders are explicitly designed to avoid the long-term dependency problem. Remembering information for long period of time is practically their default behavior and hence they have an advantage over normal auto-encoders. So it is one of the best technique for finding anomalies in time series data.

LSTM is used for model training. As LSTM are designed to look at the historical data to make predictions, it processes data up to (t-lookback) to make a prediction at a given time t. It takes a 3D array as input.

samples × *lookback* × *features*

Once the data is ready, it is divided into the train-data and test-data. Train-data is used for model training and parameter tuning, while test-data is used for model evaluation.

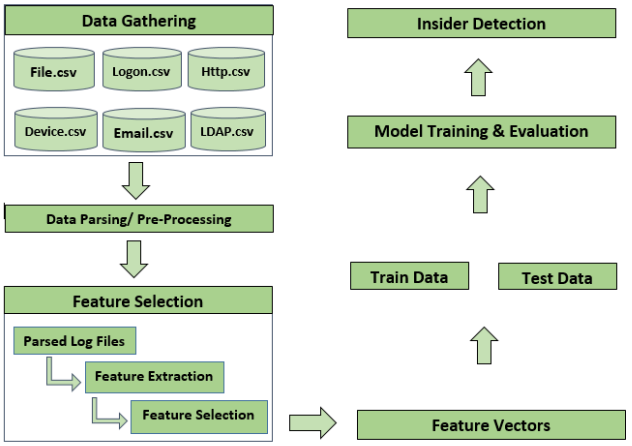

FIGURE 4. Structure of our proposed solution.

A. DATA GATHERING

The dataset used is the CMU CERT synthetic insider threat dataset r4.2. The dataset consists of synthetic data of both normal and malicious insiders. To make the approach simple all the csv files are aggregated and most relevant features are extracted. The dataset consists of 1000 synthetic users out of which 70 are malicious insiders. The dataset consists of various csv files [38], in which following are included:

1) **logon.csv**: Log of users logging in and out on a computer
2) **device.csv**:Log of users connecting and disconnecting external devices (USB)
3) **http.csv**:Users browser history
4) **email.csv**:Email logs
5) **file.csv**:Log of user activity on files (coping file to an external device)
6) **psychometric.csv**:Contains user personality attributes [69] i-e. OCEAN (Openness, Conscientiousness, Extraversion, Agreeableness, Neuroticism).
7) **LDAP (Lightweight Directory Access Protocol)**:Set of files describing all users and their assigned job roles.

Total number of rows are 32,770,220.

The reason for selecting version r4.2 is that most datasets had one instance of each scenario. Dataset 4.2 was a ''dense needle'' dataset and had many instances of each scenario.

B. INSIDER THREAT SCENARIO

In this dataset, malicious insider user is designed to accomplish one out of the following two scenarios at some point in time.

1) Use of external hard drives, or work after hours, login activity after office hours by the user who did not have such previous routine, using of a flash drive, uploading data to wikileaks.org and then leaving the organization shortly thereafter.
2) User visiting job sites and seeking employment from a competitor. Use of a flash drive (at markedly higher rates than their previous activity) to steal data before leaving the organization.

C. DATA PRE-PROCESSING

The dataset consists of various csv files (logon, file, HTTP, email, device, LDAP) and each file contained raw data. This data cannot be fed to the algorithm and it needs to be pre-processed. All the csv files are parsed and an aggregated csv file i-e. Master file was created which contained data from all csv files. A features set was extracted from this aggregated master file containing both integers and text strings. The values will have to be suitably encoded to be used as input for our proposed algorithm. There were some missing data in the synthetic dataset 4.2, due to which it does not look like real life data collected from sensors. The algorithms cannot work with this missing data. So at this step data is pre-processed by replacing missing values with the estimated mean value of the relevant feature. The machine learning algorithms uses numerical or integer values so the values are integer encoded. Data pre-processing is a tedious job, so a code was written to automate this proves in order to perform the job efficiently and save time and resources.

D. FEATURE EXTRACTION

All the CSV files are parsed and relevant data fields are identified that can efficiently learn and predict user behavior. Moreover, it depends on the insider scenarios with which we are dealing. For example, a user normal working hours are from 8AM to 7PM, so it is normal if a user login and logout during this time. However, if a user login after office hours, use some flash drive or USB and leaves shortly thereafter, this behavior is considered malicious.Psychometric.csv is not used in feature selection. Features from all other csv files included integer encoded day, time, pc, user_id, PC, user_role,user_functional_unit, user_department and activity features. Id of the features is redundant and is not included.

Most previous approaches used fixed time window based features, however it may reduce the likelihood of detecting anomalous behavior. Instead of using fixed time based windows, user session based flexible time window is used. A user session includes various activities i-e. it starts with logon activity followed by other activities (http,email,file etc.) and ends with logoff activity. Day, time, user_id, PC and activity feature reveals user session based activity and information. However, user_role, functional_unit and department reveals important information related to user job role and responsibilities, and are populated against each user. Feature values used in this work are shown in the Table 3.

The collected features from various csv files contains multiple categorical and ordinal values given as text strings, and will not be used as input for our algorithm. Therefore, the values will have to be suitably encoded for the algorithms to make correct prediction. Presence of a feature is represented by ''1'' while the absence by ''0''. Activities are labeled as per Table 4.

TABLE 3. Feature values.

Features	Values
Day	0-6
Time	1-24
Activity	1-7
User_id	1-1000
User_role	1-42
User_functional_unit	1-6
User_department	1-7
PC	Unique number

TABLE 4. Activity labels.

Activity	Label
Logon	1
Logoff	2
Connect	3
Disconnect	4
E-mail	5
File	6
Http	7

TABLE 5. User_functional_unit encoding.

User_Functinal_Unit	Label
Administration	1
Research And Engineering	2
Manufacturing	3
Finance	4
Sales And Marketing	5
Purchasing And Contracts	6

Days are labeled as "0" to "6" for Monday to Sunday respectively. Each user has an assigned role inside the organization. The encoding scheme of user functional unit is as follows:

Once the features are extracted and useful features are selected, feature vector for each user U at time interval t, U_t^i(where i = 1 –> number of users) is created as shown in Figure 5.

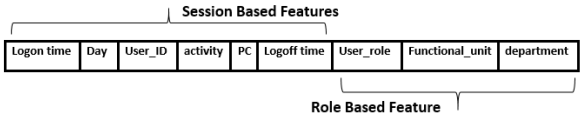

FIGURE 5. Feature vector.

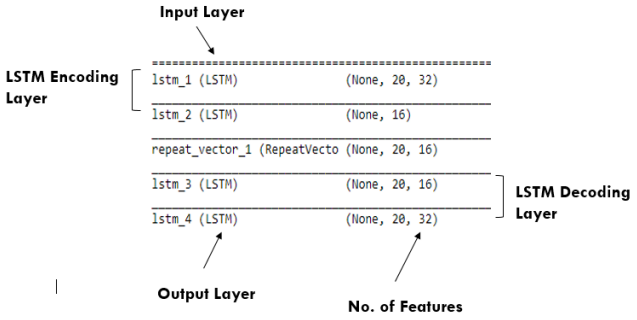

FIGURE 6. LSTM-autoencoder architecturer.

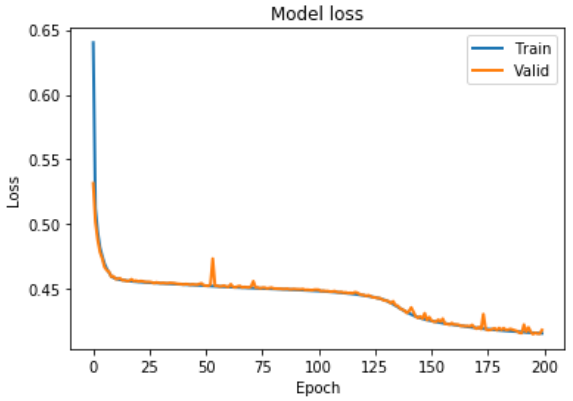

FIGURE 7. Model loss over the epochs.

IV. EXPERIMENTAL SETUP

An experimental setup was established to evaluate and study the importance and usefulness of the proposed technique. The experimental environment consists of AMD A8 pro 1.9 GHz CPU,8 GB RAM, Windows 10 Home. The testing is carried out on Anaconda 2018.12, Build: py37_0 using Jupyter Notebook 5.7.4, which is a web based, interactive programming environment enabling user to run and edit human readable documents.

A. LSTM AUTOENCODER TRAINING

The model is trained with Epochs = 200, batchsize = 64, learning rate = 0.0001, activation function is "relu" and optimizer used is "Adam". The dimensions used for input and output are same. The architecture of LSTM autoencoder is shown in Figure 6.

The difference between input and output sequence is calculated as loss function. The loss function is calculated as Mean Squared Error (MSE). The loss over the epochs is plotted as shown in Figure 7.

The input is reconstructed to the output, during the model training phase. The model is trained over normal or negatively labeled data i-e. Insider = 0. During testing phase if the reconstruction error is high, it is considered as anomalous behavior. The model is tested using both positive and negative samples. The reconstruction error for insider user is very high as compared to normal users. A threshold value is defined to segregate normal and malicious behavior. If the value is higher than threshold it is considered as "**Insider**" and if the value is lower, it is considered "**Normal**" as shown in Figure 8.

V. EXPERIMENTAL RESULTS AND DISCUSSION

The dataset used is the CMU CERT synthetic dataset r4.2 which consists of 1000 synthetic users out of which 70 are malicious insiders. The dataset contains only 0.03% anomalous and 99.7% normal instances. In order to solve the problem of class imbalance, the technique of random over

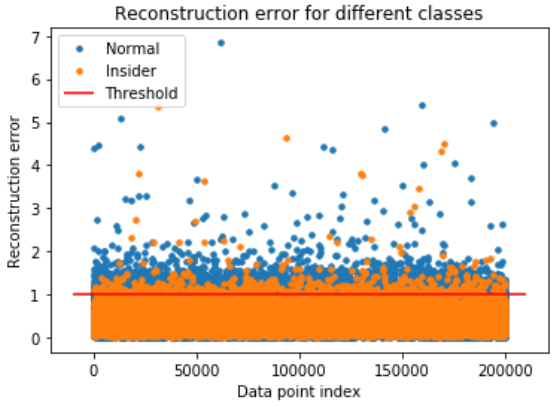

FIGURE 8. Reconstruction error.

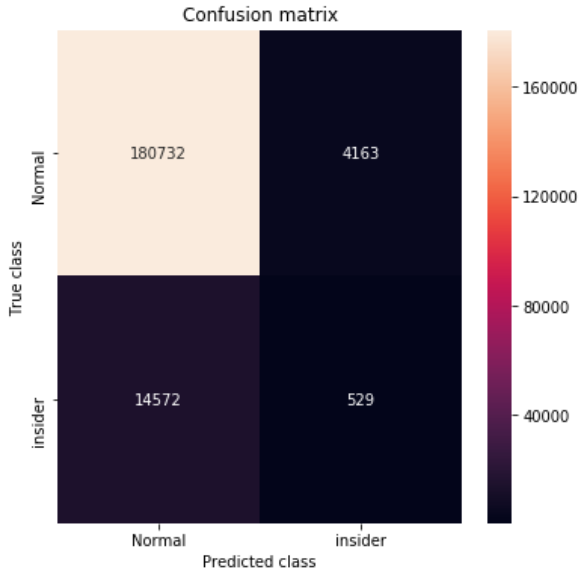

FIGURE 9. Confusion matrix.

sampling is used. In which copies of anomalous instances are diffused within the dataset. The dataset is divided into train, valid, and test data for model training, validation and testing. The proportion in which data is divided includes: 70% training data, 10% validation data and 20% testing data. Testing data contains instances of both normal and malicious data. Total number of logs are 32,770,220 out of which 17,193 are trainable parameters.

During the training phase LSTM autoencoder reconstructs input sequence to the output sequence and a loss function as MSE is calculated to identify the difference. A threshold value is set to segregate insider and normal users. If the reconstruction error is higher than threshold than it is considered ''Insider'', and if lower than the threshold than it is considered ''Normal''. The reconstruction error for normal user is low because the model is trained with normal data. Once the model is trained, it is then tested on mix data samples including both normal and malicious instances. The reconstruction error for insider user is very high as compared to normal

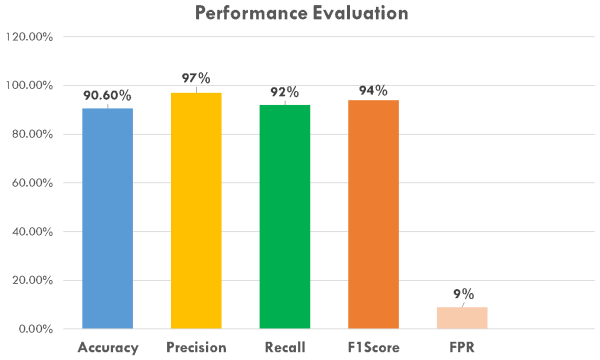

FIGURE 10. Experimental results.

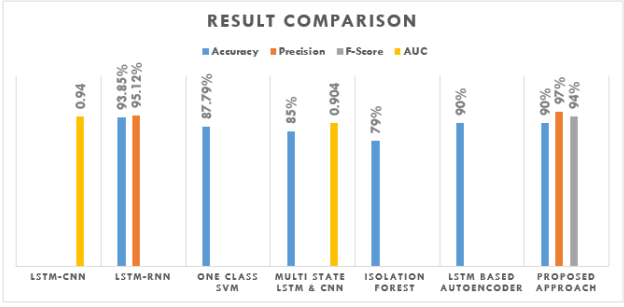

FIGURE 11. Result comparison.

users. A confusion matrix is generated which evaluates the performance of our classification model. Each row represents a class: Normal and Insider, while each column cell shows the predicted values i-e. True Positives, False Positives, True Negative and False Negative as shown in Figure 9.

It helps us to identify classification accuracy along with other performance evaluation scales (Precision, F-Score, Recall etc.) The performance in terms of accuracy, precision and F1Score is calculated as shown in Figure 10. The model achieves a remarkable accuracy = 90.60%, precision = 97%, F1Score = 94% and FPR = 9%.

A. COMPARISON OF RESULTS WITH OTHER TECHNIQUES

Performance of our proposed algorithm is compared to other well-known techniques i-e. LSTM-CNN, LSTM-RNN, One Class SVM, Multi State LSTM & CNN and Isolation forest in terms of performance evaluation metrics, dataset upon which evaluation is performed and feature set used. LSTM-CNN used user activity based feature set. Detail about feature set used in LSTM-RNN is missing, One-Class SVM used domain based features, Multi State LSTM & CNN used user behavior based features, Isolation Forest used psychometric observations & web access patterns and our proposed approach used user session based rich feature set i-e. LogonLogoff Events,UserId, User_role, Functional_unit, Department, Day, Time,PC. The parameters which has a significant impact on results include: feature set, insider scenarios and the dataset used. r4.2 has multiple instances of each scenario which fully evaluate the performance of the algorithms. Upon comparison with other techniques it is

TABLE 6. Comparison of results.

Sr.	Paper Ref.	Technique Used		Algorithm	Performance Matrices				
		Behavior Based	Graph Based		Accuracy	Precision	F-Score	AUC	Dataset
1	[5]	✓	✗	LSTM-CNN	✗	✗	✗	0.94	V4.2
2	[14]	✓	✗	LSTM-RNN	93.85%	95.12%	✗	✗	V6.2
3	[18]	✓	✗	One-Class SVM	87.79%	✗	✗	✗	V4.2
4	[29]	✓	✗	Multi State LSTM & CNN	85%	✗	✗	0.9047	V6.2
5	[13]	✗	✓	Isolation Forest	79%	✗	✗	✗	V4.2
6	[38]	✓	✗	LSTM based Autoencoder	90%	✗	✗	✗	V4.2
7	Proposed Approach	✓	✗	LSTM-AutoEncoder	90%	97%	94%	✗	V4.2

observed that our novel approach produces relatively good accuracy (90.6%), precision (97%) and F1 Score (94.4%) .

The detailed comparison is shown in Table 6 and a graph is shown in Figure 11.

VI. CONCLUSION AND FUTURE DIRECTIONS

We study the insider threat problem, and identified that mitigating this problem is a challenging task. Now a days, mitigation against insider threat is achieved by implementing user access controls, user behavior monitoring and physical security controls. In this work a Deep Learning based insider attack detection scheme is presented. The main aim behind the development of this scheme is its application on user technical data within an organization with low processing and memory requirements. Moreover, the developed system is simple and adaptable with minimum domain knowledge requirement.

Different insider threat scenarios are used and "LSTM-Autoencoder" is used for insider threat detection. Model is trained and tested on CMU CERT V4.2. The platform used for evaluation of the scheme is Anaconda 2018.12, Build: py37_0 using Jupyter Notebook 5.7.4. Moreover the performance of the proposed algorithm has been compared to other well-known techniques i.e. LSTM-CNN, Random Forest, LSTM-RNN, One Class SVM, Markov Chain Model, Multi State LSTM & CNN, Gated Recurrent Unit & Skip-gram. The comparison showed that our novel approach produces relatively good accuracy(90.60%), precision(97%) and F1 Score (94%).

In order to create a robust Insider detection system, we need to create more diverse insider threat scenarios, as there is a lack of publicly available threat scenarios. This will help us in solving the insider problems with more creativity, high quality and accuracy.

ACKNOWLEDGMENT

This work was supported by the Artificial Intelligence Data Analytics Lab (AIDA) CCIS, Prince Sultan University, Riyadh, Saudi Arabia. Authors are thankful for the support.

REFERENCES

[1] *Insider Report 2018*, CA Technol., New York, NY, USA, 2018.
[2] *Insider Threat Report 2019*, CA Technol., San Jose, CA, USA, 2019.
[3] S. R. Band, D. M. Cappelli, L. F. Fischer, A. P. Moore, E. D. Shaw, and R. F. Trzeciak, "Comparing insider IT sabotage and espionage: A model-based analysis," Carnegie Mellon Univ., Softw. Eng. Inst., Pittsburgh, PA, USA, Tech. Rep. CMU/SEI-2006TR-026, 2006.
[4] P. Chattopadhyay, L. Wang, and Y.-P. Tan, "Scenario-based insider threat detection from cyber activities," *IEEE Trans. Comput. Social Syst.*, vol. 5, no. 3, pp. 660–675, Sep. 2018.
[5] F. Yuan, Y. Cao, Y. Shang, Y. Liu, J. Tan, and B. Fang, "Insider threat detection with deep neural network," in *Proc. Int. Conf. Comput. Sci.* Cham, Switzerland: Springer, 2018.
[6] W. Jiang, Y. Tian, W. Liu, and W. Liu, "An insider threat detection method based on user behavior analysis," in *Proc. Int. Conf. Intell. Inf. Process.* Amsterdam, The Netherlands: International Federation for Information Processing, 2018, pp. 421–429.
[7] C. Liu, Y. Zhong, and Y. Wang, "Improved detection of user malicious behavior through log mining based on IHMM," in *Proc. 5th Int. Conf. Syst. Informat. (ICSAI)*, Nov. 2018, pp. 1193–1198.
[8] Z. Zamanian, A. Feizollah, N. B. Anuar, L. B. M. Kiah, K. Srikanth, and S. Kumar, "User profiling in anomaly detection of authorization logs," in *Computational Science and Technology*. Singapore: Springer. 2019.
[9] J. Jiang, J. Chen, K.-K.-R. Choo, K. Liu, C. Liu, M. Yu, and P. Mohapatra, "Prediction and detection of malicious insiders' motivation based on sentiment profile on webpages and emails," in *Proc. MILCOM*, Oct. 2018, pp. 1–6.
[10] 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 *Proc. SecArch*, Toronto, ON, Canada, Jan. 2018, pp. 18–20.
[11] K. A. Tabash and J. Happa, "Insider-threat detection using Gaussian mixture models and sensitivity profiles," *Comput. Secur.*, vol. 77, pp. 838–859, Aug. 2018.
[12] O. Lo, W. J. Buchanan, P. Griffiths, and R. Macfarlane, "Distance measurement methods for improved insider threat detection," *Secur. Commun. Netw.*, vol. 2018, pp. 1–18, Jan. 2018.
[13] A. Gamachchi, L. Sun, and S. Boztas, "Graph based framework for malicious insider threat detection," in *Proc. 50th Hawaii Int. Conf. Syst. Sci. (HICSS)*, 2017, p. 10.
[14] F. Meng, F. Lou, Y. Fu, and Z. Tian, "Deep learning based attribute classification insider threat detection for data security," in *Proc. IEEE 3rd Int. Conf. Data Sci. Cyberspace*, Jun. 2018, pp. 576–581.
[15] A. Shaghaghi, S. S. Kanhere, M. A. Kaafar, E. Bertino, and S. Jha, "Gargoyle: A network-based insider attack resilient framework for organizations," in *Proc. IEEE 43rd Conf. Local Comput. Netw. (LCN)*, Oct. 2018, pp. 553–561.
[16] D. Patil and B. Meshram, "Network packet analysis for detecting malicious insider," in *Proc. 3rd Int. Conf. Converg. Technol.*, 2018, pp. 1–8.
[17] L. Liu, O. De Vel, C. Chen, J. Zhang, and Y. Xiang, "Anomaly-based insider threat detection using deep autoencoders," in *Proc. IEEE Int. Conf. Data Mining Workshops (ICDMW)*, Nov. 2018, pp. 39–48.
[18] L. Lin, S. Zhong, C. Jia, and K. Chen, "Insider threat detection based on deep belief network feature representation," in *Proc. Int. Conf. Green Informat. (ICGI)*, Aug. 2017, pp. 54–59.
[19] J. Lu and R. K. Wong, "Insider threat detection with long short-term memory," in *Proc. ACSW*, Sydney, NSW, Australia, Jan. 2019, pp. 1–10.
[20] J. Wang *et al.*, "Learning correlation graph and anomalous employee behavior for insider threat detection," in *Proc. 21st Int. Conf. Inf. Fusion (FUSION)*, Jul. 2018, pp. 1–7.
[21] X. Wang, Q. Tan, J. Shi, S. Su, and M. Wang, "Insider threat detection using characterizing user behavior," in *Proc. IEEE 3rd Int. Conf. Data Sci. Cyberspace*, Jun. 2018, pp. 476–482.

[22] M. M. Yamin, B. Katt, K. Sattar, and M. B. Ahmad, "Implementation of insider threat detection system using honeypot based sensors and threat analytics," in *Proc. Future Inf. Commun. Conf.* Cham, Switzerland: Springer, 2020, pp. 801–829.
[23] Z. Li and K. Liu, "An event based detection of internal threat to information system," in *Proc. Int. Conf. Harmony Search Algorithm*, in Advances in Intelligent Systems and Computing. Cham, Switzerland: Springer, 2019, pp. 44–53.
[24] Z. Zhang, S. Wang, and G. Lu, "An internal threat detection model based on denoising autoencoders," in *Advances in Intelligent Information Hiding and Multimedia Signal Processing*. Singapore: Springer, 2020.
[25] S. Velampalli, L. Mookiah, and W. Eberle, "Discovering suspicious patterns using a graph based approach," in *Proc. 32nd Int. Florida Artif. Intell. Res. Soc. Conf. (FLAIRS)*, 2019, pp. 382–386.
[26] M. Aldairi, L. Karimi, and J. Joshi, "A trust aware unsupervised learning approach for insider threat detection," in *Proc. IEEE 20th Int. Conf. Inf. Reuse Integr. Data Sci. (IRI)*, Jul. 2019, pp. 89–98.
[27] J. Kim, M. Park, H. Kim, S. Cho, and P. Kang, "Insider threat detection based on user behavior modeling and anomaly detection algorithms," *Appl. Sci.*, vol. 9, no. 19, p. 4018, Sep. 2019.
[28] 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 Netw., Ubiquitous Comput., Dependable Appl.*, vol. 6, pp. 13–20, May 2015.
[29] M. Singh, B. M. Mehtre, and S. Sangeetha, "User behavior profiling using ensemble approach for insider threat detection," in *Proc. IEEE 5th Int. Conf. Identity, Secur., Behav. Anal. (ISBA)*, Jan. 2019, pp. 1–8.
[30] C. Soh, S. Yu, A. Narayanan, S. Duraisamy, and L. Chen, "Employee profiling via aspect-based sentiment and network for insider threats detection," *Expert Syst. Appl.*, vol. 135, pp. 351–361, Nov. 2019.
[31] D.-W. Kim, S.-S. Hong, and M.-M. Han, "A study on classification of insider threat using Markov chain model," *KSII Trans. Internet Inf. Syst.*, vol. 12, no. 4, pp. 1887–1898, Apr. 2018.
[32] S.-S. Tan, S. Duraisamy, and J.-C. Na, "Unified psycholinguistic framework: An unobtrusive psychological analysis approach towards insider threat prevention and detection," *J. Inf. Sci. Theory Pract.*, vol. 7, no. 1, pp. 52–71, 2019.
[33] E. E. Schultz, "A framework for understanding and predicting insider attacks," *Comput. Secur.*, vol. 21, no. 6, pp. 526–531, Oct. 2002.
[34] H. Zhang, I. Agrafiotis, A. Erola, S. Creese, and M. Goldsmith, "A state machine system for insider threat detection," in *Proc. Int. Workshop Graph. Models Secur.* Cham, Switzerland: Springer, 2019, pp. 111–129.
[35] A. Gamachchi and S. Boztas, "Insider threat detection through attributed graph clustering," in *Proc. 16th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun.*, Aug. 2017, pp. 112–119.
[36] O. Brdiczka, J. Liu, B. Price, J. Shen, A. Patil, R. Chow, E. Bart, and N. Ducheneaut, "Proactive insider threat detection through graph learning and psychological context," in *Proc. IEEE Symp. Secur. Privacy Workshops*, May 2012, pp. 142–149.
[37] *Insider Threat Test Dataset*, Softw. Eng. Inst., 2016. [Online]. Available: https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=508099
[38] B. Sharma, P. Pokharel, and B. Joshi, "User behavior analytics for anomaly detection using LSTM autoencoder—Insider threat detection," in *Proc. IAIT*. Bangkok, Thailand: Association for Computing Machinery, Jul. 2020, pp. 1–9.
[39] E. Pantelidis, G. Bendiab, S. Shiaeles, and N. Kolokotronis, "Insider threat detection using deep autoencoder and variational autoencoder neural networks," in *Proc. IEEE Int. Conf. Cyber Secur. Resilience*, Jul. 2021, pp. 129–134.
[40] M. N. Al-Mhiqani, R. Ahmed, Z. Zainal, and S. N. Isnin, "An integrated imbalanced learning and deep neural network model for insider threat detection," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 1, pp. 1–6, 2021.