Intelligent Automation & Soft Computing
DOI:10.32604/iasc.2022.021430
Article

Tech Science Press

Insider Threat Detection Based on NLP Word Embedding and Machine Learning

Mohd Anul Haq[1], Mohd Abdul Rahim Khan[1,*] and Mohammed Alshehri[2]

[1]Department of Computer Science, College of Computer and Information Sciences, Majmaah University, Al-Majmaah 11952, Saudi Arabia
[2]Department of Information Technology, College of Computer and Information Sciences, Majmaah University, Al-Majmaah 11952, Saudi Arabia
*Corresponding Author: Mohd Abdul Rahim Khan. Email: m.khan@mu.edu.sa
Received: 02 July 2021; Accepted: 09 November 2021

Abstract: The growth of edge computing, the Internet of Things (IoT), and cloud computing have been accompanied by new security issues evolving in the information security infrastructure. Recent studies suggest that the cost of insider attacks is higher than the external threats, making it an essential aspect of information security for organizations. Efficient insider threat detection requires state-of-the-art Artificial Intelligence models and utility. Although significant have been made to detect insider threats for more than a decade, there are many limitations, including a lack of real data, low accuracy, and a relatively low false alarm, which are major concerns needing further investigation. In this paper, an attempt to fulfill these gaps by detecting insider threats with the novelties of the present investigation first developed two deep learning hybrid LSTM models integrated with Google's Word2vec LSTM (Long Short-Term Memory) GLoVe (Global Vectors for Word Representation) LSTM. Secondly, the performance of two hybrid DL models was compared with the state-of-the-art ML models such as XGBoost, AdaBoost, RF (Random Forest), KNN (K-Nearest Neighbor) and LR (Logistics Regression). Thirdly, the present investigation bridges the gaps of using a real dataset, high accuracy, and significantly lower false alarm rate. It was found that ML-based models outperformed the DL-based ones. The results were evaluated based on earlier studies and deemed efficient at detecting insider threats using the real dataset.

Keywords: Natural language processing; insider threats; lstm; word2vec; global vectors for word representation

1 Introduction

In telecommunication, we are exchanging and sharing several petabytes of information over computer networks. It requires the protection of information from insider and outsider threats. While the detection of outsider threats has an adequate level of security benchmarks, local insider attackers are increasing data vulnerability due to expansion in technology. Indeed, inside-network threats are extremely difficult to

detect, posing a high-security risk [1–3]. Insider threats can cause serious destruction of an organization's reputation, financial assets, and intellectual resources. As per a year report of 2018, insider threat was more than 53% due to insider attacks. Moreover, a survey report showed that 27% of the security attacks were insider attacks [4]. Some well-known insider threat incidents leading to data leakage have even reached the media [1].

Thus, in the current scenario, most organizations try to implement an Intrusion Detection System (IDS), Intrusion Prevention System (IPS), and Virtual Private Networks (VPN) to defend from outside threats while using the anti-malicious technique to defend against insider threats. Unfortunately, most insider threats remain undetected as the current detection approaches have serious limitations. Thus, detection of insider threats requires in-depth research [5], given the lack of pertinent information in the present scenario.

According to a survey report (2019) [6] of insider threat, the organization felt 68% vulnerable from moderately to extremely, 73% inside attacked confirmed by organizations, 53% of inside attacked migrate to the cloud become more challenging, and 59% privileged user of the organization poses the biggest insider security risk.

In some cases, [7–9], insider threat leads to the exposure of an organization sensitive information and makes it highly vulnerable. Therefore, insider threats are the biggest challenge that needs to be tackled inside organizations to increase mobile devices to connect with the network. An insider threat has been defined as [10] "an existing or ex-employee, vendor, business partner, or contractor who can access to the network, system, or data of any organization and intentionally abused that access to compromise the integrity, availability, and confidentiality of the information."

The word embedding is an essential representation for document vocabulary or vector representation of any word; additionally, it is beneficial to detect insider threat from text data such as e-mails. Word embedding is used for extracting the context, semantic and syntactic, based on the words in a document.

Previous insider threat detection methods have many limitations, including a lack of real data, low accuracy, and a relatively low false alarm. There are three main objectives of the present investigations to address the gaps by detecting insider threats with the novelties of the present investigation first developed two deep learning hybrid LSTM models integrated with Google's Word2vec LSTM (Long Short-Term Memory) GLoVe (Global Vectors for Word Representation) LSTM. Secondly, the performance of two hybrid DL models was compared with the state-of-the-art ML models such as XGBoost, AdaBoost, RF (Random Forest), KNN (K-Nearest Neighbor) and LR (Logistics Regression). Thirdly, the present investigation bridges the gaps of using a real dataset, high accuracy, and significantly lower false alarm rate.

1.1 Word2Vec

Word2Vec is a popular NLP technique based on neural network to learn word embeddings from 100 billion words [11–13]. Google's Word2vec model has the ability to detect synonyms of words or suggest suitable words for an incomplete sentence. Word2vec signifies each individual word with a vector. The vectors are selected using the cosine angle between the vectors, which specifies the semantic match between the words symbolized by corresponding vectors. Consider the two similar statements: 'you are a good person' and 'you are a great person'. These two statements have a similar meaning. If we want to develop a vocabulary V it will be V = {You, are, a, good, great, person}. We can create a one-hot encoding vector for V of length six. We will assign the value 1 for a particular element while the rest will be zero vector. i.e.,; You = {1, 0, 0, 0, 0, 0} T; are = {0, 1, 0, 0, 0, 0} T; a= {0, 0, 1, 0, 0, 0} T good= {0, 0, 0, 1, 0, 0} T; great= {0, 0, 0, 0, 1, 0} T; person= {0, 0, 0, 0, 0, 1} T (where T represents transpose). It will represent a 6-d (dimension) space to visualize the mentioned encodings, where an individual word signifies 1-d and the word great and good are different. The cosine angle between the vectors should be close to 1, i.e., angle close to 0 based on Eq. (1).

$$\text{Sim(A, B)} = \cos(\lambda) = \frac{A.B}{||A||||B||} \tag{1}$$

Word2Vec embedding can be achieved using the Common Bag of Words (CBOW) model.

1.1.1 CBOW Model One-Word Context

CBOW model uses the context of the individual word as the input and attempts to predict the word according to the corresponding context [11]. Suppose V is the one-hot encoded vector of size V; WVXN and W`NXV are the weight matrix to map the input → hidden layer and hidden layer →output layer respectively. Let an input xk = 1 and yj' = 0 for j' ≠ j, we have

$$\text{h} = W^T\text{y} = \text{w}^{\text{T}}_{(\text{j},\cdot)} := \text{v}^{\text{T}}_{\text{wI}} \tag{2}$$

where vw is the N-d vector representation of the associated word of the input layer.

From the hidden→output layer, W' = {w'ik}, the score sj can be computed with weight matrix based on individual words of the vocabulary,

$$\text{s}_\text{j} = \text{v}'^{T}_{\text{wk}} \tag{3}$$

where v'_{wk} is the kth column of W′. Then softmax classification model can be applied to get the subsequent information of words.

$$p(wk|wI) = yk = \frac{\exp(\text{s}_\text{j})}{\sum_{\text{k}'=1}^{V}\exp(\text{sj}')} \tag{4}$$

where yk is the output of the jth unit. Substituting (2) and (3) into (4), we get

$$p(wk|wI) = \frac{\exp(v'^{\,T}_{wk}v_{wI})}{\sum_{\text{k}'=1}^{V}\exp(v'^{\,T}_{wk'}v_{wI})} \tag{5}$$

where the interpretations of the w represented by vw and v′w.

1.1.2 CBOW Model Multi Word Context

During computation of hidden layer, the output of multi word model is computed based on the average of the vector (Av).

$$Av = \frac{1}{n}\text{W}^{\text{T}}(\text{x1} + \text{x2} + \ldots + \text{xn} \tag{6}$$

$$\text{Av} = \frac{1}{n}(\text{v}_{w1} + \text{v}_{w2} + \ldots + \text{v}_{wn})^T \tag{7}$$

where the total number of words denoted by n, $w_I \ldots, w_{I,N}$, v_w is the input vector of a word w, the loss function is given by

$$L = -logp(w_O|w_{I,1}, \ldots, w_{I,N}) \tag{8}$$

$$L = -u_{k*} + log\sum_{\text{j}'=1}^{V}\exp(u_{k'}) \tag{9}$$

$$L = (-v_{wo}^{'T}.Av) + log\sum_{k'=1}^{V}\exp(v_{wk}^{'T}.Av) \tag{10}$$

The updated Eq. (11) for the hidden to the output was same as in the one-word case.

$$v_{wk}^{'}(updated) = v_{wk}^{'}(previous) - \tau.e_k.h \quad for\ k = 1,\ 2,\ 3\ldots,\ V \tag{11}$$

The updated equation for input→hidden weights is similar to one word context, except that in present case this equation is applied for every word instead of one word.

$$v_{wI,c}^{(updated)} = v_{wI,c}^{(previous)} - \frac{1}{C}\tau.EH^T\ for\ c = 1,\ 2,\ 3\ldots,\ C \tag{12}$$

where $v_{wI,c}$ = input vector of the cth word; τ is a positive learning rate; and $FH = \frac{\partial F}{\partial h_i}$ is given by Eq. (12).

1.2 GloVe

GloVe model is a log-bilinear model based on weighted least-squares to get the word embedding using 6 billion words [14–16]. The training is performed on combined global word-word co-occurrence statistics, and the output depicts the linear substructures of the vector space. The GLoVe model also uses cosine similarity to obtain the semantic similarity. GloVe was developed to address the vector differences of possible meaning matching by the association of two words. The training of this model learned the vectors in a way that their corresponding dot product equals the logarithm of the words' probability of co-occurrence [14–16]. The most general form of the model is given by:

$$F(w_x,\ w_y,\ w_z) = \frac{P_{xz}}{P_{yz}} \tag{13}$$

Selecting the best parameters is an essential step in selecting and optimizing the AI/ML model; most investigations used default parameters that are not the quality of the present investigation because all the rigorous parameter tuning was performed based on GridSearchCV for all seven developed models.

2 Related Works

This paper proposes an unsupervised-based learning algorithm [17] for the detection of anomaly-based detection. The proposed approach classifies a real dataset to ascertain the accuracy for data streams containing insider threat anomalies, using the LZW (Lempel Ziv- Welch) algorithm and the shell command line. This paper proposes data mining and forensic techniques to identify the representative System Call (SC) patterns for users [18]. It employs the term frequency-inverse document frequency (TF-IDF) to count the SCs collected in a user log file. The proposed approach describes two types of information handling: analysis and verification [19]. Here, the authors improve the classification sequence to evade quick decision-making when the environment is not clear.

In this approach, it was required to analyze the monitor sequences. Whenever a suspicious situation arises, divides the long sequence into subsequences of minor intrusions more noticeable and easily analyzed. We applied the verification scheme based on the numerical non-parametric U-test. The proposed approach used the supervised learning method [20,21] to detect real insider threats, used one-class SVM (OCSVM) and Multiple OCSVM to detect the insider threat. Here, it showed effectiveness over OCSVM in terms of True Positive and False Positive. This approach used a user-generated dataset classified by using unsupervised learning [22]. It formulated 2 working clusters that classified tradition patterns that show average-to-low and average-to-high stress points. The primary goal, authors,

distributed user-generated datasets into chronologically defined sub-periods to learning potential tradition variations over time.

The proposed approach design employs the reference point-based Local Outlier Factor (LOF) method that measures outlines of data points regarding a set of known data points. This approach was attentive to abnormal behavior detection applications where multiple users share the application and system [23]. It used the LOF method to evaluate the good anomalous samples from the behavior of other users. The approach gave a better result in comparing a single-class classifier and designed the distributions of abnormal samples for training. This approach proposed signature-based intrusion detection, preparing the signature by collecting and merging databases from various belief sources and updates [24]. In database detection approaches, it is believed that the intra-transactional features are adequate for detecting insider threats. Moreover, they add three different sensitivity levels of attributes to monitor the modified malicious activities more carefully.

This paper presented a software component-based framework for anomaly detection [25]; the framework used runtime-based unsupervised learning, which gives rapid anomaly detection. This estimation of the approach alongside a real Emergency Deployment System has established positive results and presented the framework that can detect secret attacks and insider threats; recent intrusion detection approaches may miss that. This paper applied a hybrid approach, including graph-based detection, unsupervised learning, insider threat detection, and stream mining, which is more effective than any single anomaly-based detection approach. This approach generally identified the insider threats, which attempt to conceal their activities by changing their behaviors over time [26]. This paper designed a system that automatically detects anomalies using critical content and context-based information [27]. After this process, based on information to detect the insider threats. Moreover, this system maintains the database of historical logs, which is helpful to detect the typical level of criticality of data.

This approach used behavioral analytic anomaly-based detection, which used a framework to analyze Active Directory domain service logs and keep track of the insider threats [28]. The experiment was applied to real datasets. The proposed idea of this framework is more feasible to keep track of cyber security-based detection. The proposed approach is based on any unauthorized movement of data by insider threats [29]. It uses file repositories, which is the statistical method for authorized or legitimate users. In this case, every user has profile access and repository access logs. It can be worked for the detection of a large set of data exfiltration activities. The proposed approach [30] identifies normal behaviors using the Hidden Markov Models (HMMs). Moreover, this model detects the deviations from those behaviors. The result showed that it could detect insider threats and learn user behaviors efficiently. The present study applied the unsupervised learning approach of clustering and one-class support vector machine (OCSVM) [31]. This approach is weak for the detection of network attacks. However, this approach is more suitable for anomaly detection, network flows, and signature-based detection. The proposed approach is to detect insider threats [32]. The authors used the CERT dataset and analyzed it using various distance-vector methods to detect behavior deviation. This approach presented graph-based techniques [33]. It used flow algorithms to use both types of mutual similarities from user modes and respected similarities from queries to choose from user profiles for a more dependable composed detection. The proposed model is used to detect insider threats by using the deep belief network (DBN) [34]. This approach extracted the hidden feature from the audit logs; it used the One-Class SVM (OCSVM) and applied the extracted feature by DBN. This approach has a significant strength in that it used the three different supervised classifiers to find an accurate result [35]. This model showed trust in different attack categories, such as actual wireless sensor attacks and insider attacks. This model has better accuracy for detecting the accuracy of malicious nodes and has better performance than similar models. The proposed approach used aggregate behavior to detect insider threats, more specific behaviors

deviated from expectancy [36]. This approach designed event-based access logs detection using the specialized network anomaly detection (SNAD).

Many researchers declared that inside threats are listed as hazardous threats [37]. However, insider threats not more attention by many originations. This approach proposed methods to identify the insiders malicious, based on the behaviors, which leads to the attacks. This approach used the data processing tool to detect the insider threat based on information-use events [38]. It has samples of max and min weight for data adjustment(DA) formulation. The principal merit of this model is that it combined gradient-boosting (XGBoost) and formulated DA to detect insider threats. This approach evaluates the high accuracy based on CERT dataset to detect insider threats. By using this model found that 8.76% better accuracy.

In this approach, authors classify insider threat data into two categories: malicious and nonmalicious classes [39]. This model used the dual-layered deep autoencoders approach with the data adjustment technique and compared some popular existing models such as RF and multilayer models. In proposed models, used the 14-GB web-browsing Insider Threat logs (CMU Dataset). We observed that the model has good precision, recall, and f-score percentage, available in Tab. 1.

Table 1: Many techniques for the detection of insider threats

SN	Method	Detection tech.	Features	Dataset	Result	Ref.
1	Unsupervised learning techniques (LZW)	Anomaly-based	Behavior of biometrics	RD	Acc = 87, TPR = 58, FPR = 0.09	[17]
2	K-means++ clustering, dts, and SVM	Anomaly-based	Cyber features	OE	HTTP = 99.6, TCP = 93, Wiki = 76	[17]
3	TF-IDF	Signature-based	Behavior of biometrics	SE	Acc = 94.29%	[18]
4	Sequence alignment	Anomaly-based	Behavior of biometrics	SY	TPR = 98.3%, FPR = 0.77%	[19]
5	Supervised learning OCSVM	Anomaly-based	Behavior of biometrics	SY	Acc.= 71% FPR = 31%, FNR = 0%	[20]
6	Multiple OCSVM models and multiple graph-based anomaly detection mode	Anomaly-based	Behavior of biometrics	SY	ACC = 0.71, SV FPR = 0.31, SV FNR = 0.0	[21]
7	Unsupervised flat data classification	Anomaly-based	Psychosocial Behaviors	RD	Accuracy: NBM = 79 SVM = 81, MLR = 80	[22]
8	local outlier factor LOF, DT, logistic regression, random forest	Anomaly-based	Behavior of biometrics	OE	AUC = 0.979	[23]
9	Markov modulated poisson process module mmppm and malicious transaction generation module (MTGM)	Signature-based	Behavior of biometrics	SE	TP = 98%, FP = 10%	[24]

(Continued)

Table 1 (continued)

SN	Method	Detection tech.	Features	Dataset	Result	Ref.
10	Dynamic approach Generalized sequential pattern mining	Anomaly-based	Behavior of biometrics	OE	Precision = 100%, Recall = 70%	[25]
11	Ensemble-based stream mining and unsupervised learning	Anomaly-based	Behavior of biometrics	SY	Accuracy = 56%, FPR = 54% FNR = 42%	[26]
12	SVM	Anomaly-based	Behavior of biometrics	OE	Accuracy approx. 98%	[27]
13	Statistical model, Markov model	Anomaly-based	Behavior of biometrics	OE	Recall = 67%, Precision = 99%	[28]
14	Tree-structured profiles and PCA	Anomaly-based	Cyber features	SY & SE	Precision = 42%, Recall = 100%	[29]
15	HMMs	Anomaly-based	Cyber features	SY	AUC: 10 state = 0.755, 20 state = 0.784, 10 state = 0.797	[30]
16	Unsupervised KNN	Anomaly-based	Cyber features	SY		
17	Bi-clustering and OCSVM	Anomaly-based	Cyber and psychosocial behaviors	OE	FP = 1%–2%	[31]
18	Hidden Markov method	Anomaly-based	Cyber features	SY	DTR = 0.39, 0.36, and 0.47 detection rate of 0.8	[32]
19	Graph-based optimization approach	Anomaly-based	Cyber features	SY	AUC, CERT = 0.9520 NATOPS = 0.7196	[33]
20	Deep belief network DBN, OCSVM	Anomaly-based	Cyber features	SY	Acc = 87.79, FP = 12.18	[34]
21	Trust management model, KNN, back-propagation neural networks (BPNN), and DT	Anomaly-based	Psychosocial behavior	SE and RD	Acc. = 60%, KNN = 96% BPNN = 89%, DT = 90%	[35]
22	Rule-learning algorithm	Signature-based	Psychosocial behavior	OE	Minsup 0.9, TPR = 0.94, TNR = 0.02	[36]
23	Analysis of logs for event-related actions	Anomaly-based	Cyber features	OE	AUC: 0.83, AVC = 0.91	[37]
24	XGBoost+DA			CERT	Acc.= 96.87% , F1-score = 98.83%, Recall = 98.16%	[38]
25	Deep autoencoder	Anomaly-based	Cyber features	SE	Acc.=98.97%	[39]
25	XGBoost (Present study)	Anomaly-based	Cyber features	RD	Precision = 0.92, recall = 1.00, f1-score = 0.95, accuracy = 0.92	(Present Study)

(Continued)

Table 1 (continued)

SN	Method	Detection tech.	Features	Dataset	Result	Ref.
26	AdaBoost	Anomaly-based	Cyber features	RD	Precision = 0.87, Recall = 1.0, F1-score = 0.93, Acc. = 0.87,	(Present Study)
27	KNN (Present study)	Anomaly-based	Cyber features	RD	Precision = 0.86, Recall = 0.92, F1-score = 0.89, Acc.= 0.80	(Present study)
28	RF (Present study)	Anomaly-based	Cyber features	RD	Precision = 0.92, Recall = 0.92, F1-score = 0.92, Acc. = 0.87	(Present study)
29	LR (Present study)	Anomaly-based	Cyber features	RD	Precision = 0.86, Recall = 0.92, f1-score = 0.89, Acc. = 0.80	(Present study)

Here, we have collated a few previous studies that have used different approaches and datasets to detect insider threats and show their accuracy. The datasets used are Case study (CS), Own environment (OE), Real data (RD), Simulation (SE), and Synthetic Dataset (SY), as presented in Tab. 1.

3 Dataset

In 2000, Enron company was among the biggest corporations in the United States. It was bankrupted in 2002 due to a corporate scam. Due to the investigation by federal agencies, confidential information was made public, with 250000 e-mail messages and detailed financial information for top employees. MIT made this dataset public for research purposes, especially related to insider threat and e-mail classification. The insider threat domain [40] was utilized in [41–43] to test their detection approaches. The detailed network diagram of the dataset is given in Fig. 1.

4 Methodology

Two top-rated pre-trained NLP models, i.e., Word2Vec and GLoVe were used in the present investigation utilizing transfer learning. The reasons to use transfer learning instead of our own embedding were the sparsity of training data and the large number of training parameters. Various libraries (Keras, TensorFlow, Numpy, Scipy, Matplotlib, genism, Seaborn, Sklearn, and e-mail) were used for word embedding based on Word2Vec and GLoVe. The pictorial representation of proposed models in Fig. 2.

4.1 Data Preprocessing

The shape of the Enron e-mail dataset was (517401, 2). The transformation of the e-mail into the correct format was performed using the e-mail library. The headers, message body, and employee names were extracted. There were 517401 and 5336 folders and unique folders. The date column was transformed for the date, month, year, hour, minutes, and seconds. The X-folder names were truncated to the last name instead of the long file name. NumPy.nan replaced the empty, missing values in the subject, and the missing value rows were dropped. It made the data shapes of (489236, 9). The columns (file, message, date, X-From, X-To, employee) were dropped.

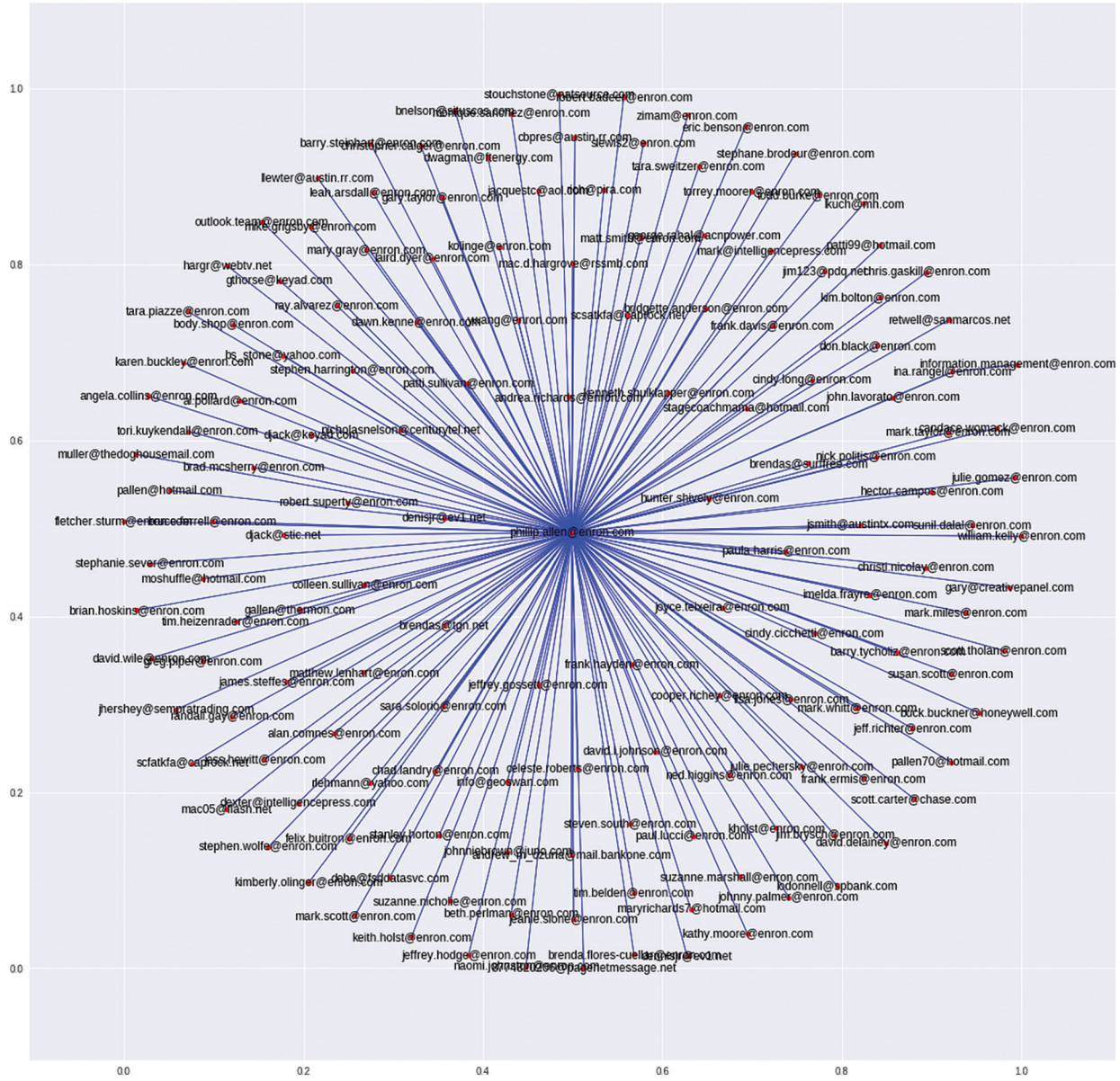

Figure 1: Network diagram of the dataset

4.2 Word2vec

As a first step, Google's pre-trained Word2Vec embeddings were loaded. It took 30.718 s to load. The Enron preprocessed data from the earlier section were loaded. The class labels were encoded using label_encoder and applied on the X-folder column. The data was split into training (90%) and testing (10%). One-hot encoding was applied for the output labels. The tokenizer was prepared and fitted on the data. The documents were padded to 150 words of maximum length and the weight matrix was created for input context in the training docs. The sequential model was defined where embedding was applied and added as a layer. Then the LSTM layer was added with 100 units and a dropout rate of 0.2, followed by a flattening layer. Then the dense layer was added with softmax function. The model was compiled with ADAM optimizer and 'categorical_crossentropy as a loss function. This model was trained to classify folders based on Word2Vec word embeddings. The summary of the model is given in Tab. 2.

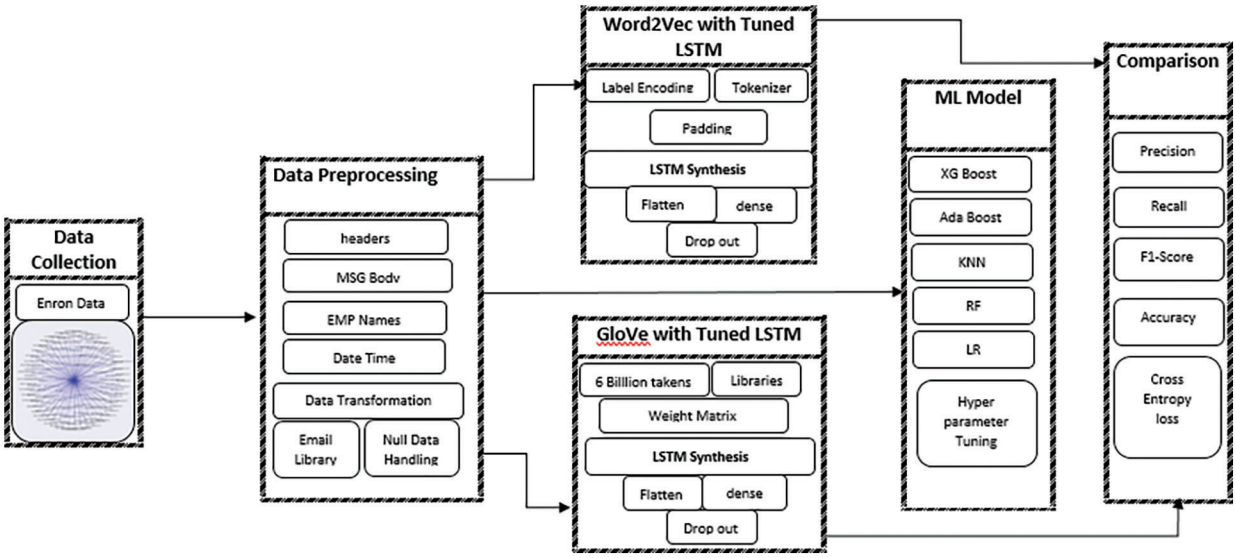

Figure 2: Two top-rated pre-trained NLP models

Table 2: Word2vecLSTM model summary Model: "sequential_2"

Layer (type)	Output shape	Param #
embedding_2 (Embedding)	(None, 150, 300)	17877600
lstm_2 (LSTM)	(None, 100)	160400
flatten_2 (Flatten)	(None, 100)	0
dense_2 (Dense)	(None, 20)	2020

Total params: 18,040,020

Trainable params: 162,420

Non-trainable params: 17,877,600

The model was trained with 60 number of epochs, verbose of 1 to know the loss and accuracy for each epoch, validation_split of 0.1 was selected.

4.3 GloVe

The pre-trained vectors trained on Wikipedia data with 6 billion tokens and a vocabulary of 400,000 words were downloaded from https://nlp.stanford.edu/projects/glove/. Then the libraries (Keras, TensorFlow, Numpy, Scipy, Matplotlib, genism and Sklearn) were imported. The glove_files, i.e., glove.6B.300d.txt and glove.6B.100d.txt.word2vec file, were loaded. The GLoVe embedding was loaded in 110 s. The weight matrix was created for input words in training docs. The GLoVe LSTM model summary is given in Tab. 3.

Table 3: GLoVe LSTM model summary Model: "sequential"

Layer (type)	Output shape	Param #
embedding (Embedding)	(None, 150, 300)	17877600
lstm (LSTM)	(None, 100)	160400
flatten (Flatten)	(None, 100)	0
dense (Dense)	(None, 20)	2020

Total params: 18,040,020

Trainable params: 162,420

Non-trainable params: 17,877,600

The model was fit and compiled using 60 epochs and verbose of 1, and validation_split of 0.1 was selected.

4.4 Machine Learning (ML) Models

In this investigation, 5 ML models were applied to the preprocessed data to detect insider threat as a person of interest (poi) identifier using financial and e-mail data by Enron. The Stratified K Fold technique for k value of 10 was applied to assess the models properly. Search CV was also applied to select the best parameters for all 5 ML algorithms. The dataset was converted from dictionary to data frame, and the insider threat person was labeled as poi and non-poi. The dataset was divided into three categories including poi labels, financial data, and e-mail data. Feature selection was performed to identify the best features and to create the feature list for train and test using the feature format function. Selecting the best parameters is an important step in selecting and optimizing the AI/ML model. We used GridSearchCV in the present investigation to select the best parameters for all the 5 ML models.

4.4.1 XGBoost

The XGBoost was initially applied with default parameters of XGBoost library and achieved the accuracy value of 0.82. To select the best parameters GridSearchCV was applied with following parameters such as Learning rate ranging from {0.05, 0.2, 0.3, 0.5, 0.6}; gamma value of {0, 0.1, 0.2, 0.5, 0.8}; max depth value of {2, 3, 4, 6, 8, 10}; and weight of minimum child between {0.25, 0.5, 1, 2, 4, 6}.

4.4.2 AdaBoost

The GridSearchCV for AdaBoost was applied with the following parameters Number of estimators ranging from {2, 4, 6, 8, 10, 12}; maximum depth ranging from {2, 4, 6, 8, 10}; min samples split of {2, 3}. The criterion of base estimator gini and entropy were given to GridSearchCV. The estimator splitter was suggested as best and random. The base estimator was given between random forest and decision tree (DT) classifier.

4.4.3 KNN

KNN dependent on the access log, which the user generates automatically; whomever users access the system. This model collects the deviation meta-information of the nearest neighbors. The selection of the number of nearest neighbors is important to get the optimized KNN model. The value of nearest neighbors was given from the grid of {3, 4, 5, 6, 7, 8, 9, 10, 11, 12}.

4.4.4 RF

Here, applied this model to detect insider threats. We applied the rule-based monitoring system to detect target scoring some background reality. The initial number of estimators was set to 100, and the range of the number of estimators was given to GridSearchCV as {1, 10, 25, 50, 100} with a max depth of {10, 20, 30, 40, 50, 60}.

4.4.5 LR

The maximum number of iterations was fixed to 5000 for RF. The learning rate is an important function of RF model. The value of learning rate was supplied as 0.01, 0.03, 0.1, 0.3, 1, 3, 10 to gridsearch. The cross validation of 10 was also selected.

4.5 Accuracy Assessment

The accuracy metrics for word embedding methods based on LSTM were computed based on accuracy and loss. The accuracy metrics for ML models were evaluated based on the precision, recall, f1-score and accuracy.

Precision is a fraction of data entries of insider threats are labeled as a truly malicious insider

$$\text{Precision} = \frac{TP}{TP + FP} \tag{13}$$

Recall is the fraction of insider data of correctly classified malicious entries.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{14}$$

f1- score is the harmonic mean between sensitivity and precision.

$$\text{f1} - \text{score} = 2 * \frac{Precision.Recall}{Precision + Recall} \tag{15}$$

Accuracy or overall classification accuracy is the fraction of all correctly classified negative and positive records

$$\text{Accuracy} = \frac{TN + TP}{TN + FN + TP + FP} \tag{16}$$

$$\text{Cross Entropy Loss} = -\frac{1}{N}\sum_{\text{o}=1}^{\text{n}} log\ \text{p}_{\ model}(y_o \epsilon\ Cy_o) \tag{17}$$

5 Results and Discussions

NLP models based on pre-trained word embeddings are important to detect the semantic and syntactic value of the word vector. The utility of pre-trained word embeddings was contributed for two significant models, i.e., Word2Vec and GloVe. In the case of Word2Vec LSTM model, the number of trainable parameters with pre-trained model was only 162,420. An attempt was made to make our model from scratch and it was observed that the number of trainable parameters were 17, 564, 879. It was a very large number of parameters to train using our model, which we built from scratch. The accuracy estimates of the pre-trained model and the model built from scratch were 0.734 and 0.675, respectively. It clearly demonstrated the need of using pre-trained model using transfer learning. Similar observation was found in the case of the GLoVe LSTM model (Fig. 3). These models can be used on the top layer as for NLP-based classifications. To sum up, all ML models have higher accuracies, precision and recall values

higher than the NLP-based word embedding models; this demonstrates that all ML models showed promising performance for e-mail classification to detect insider threats. Additionally, the best ML model was XGBoost, which achieved the accuracy of 92%.

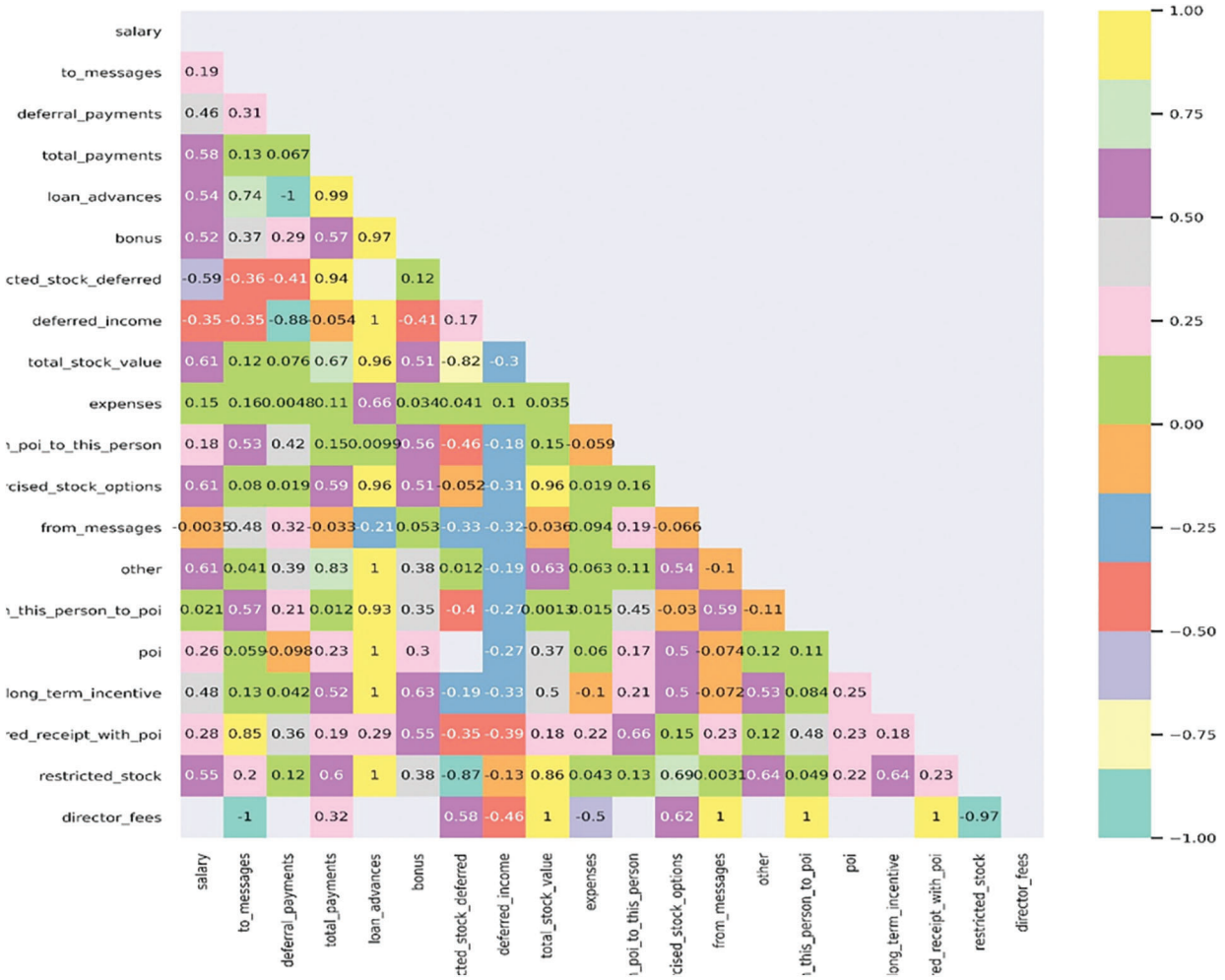

Figure 3: Heatmap depicting the correlation between all the parameters of the enron data

5.1 Word2vecLSTM

The word2vecLSTM model was evaluated based on loss value of 1.156 and accuracy value of 0.734 (Fig. 4).

5.2 GLoVeLSTM

The accuracy value of 0.748 was slightly better in the GLoVe model, with a loss of 1.167 (Fig. 5).

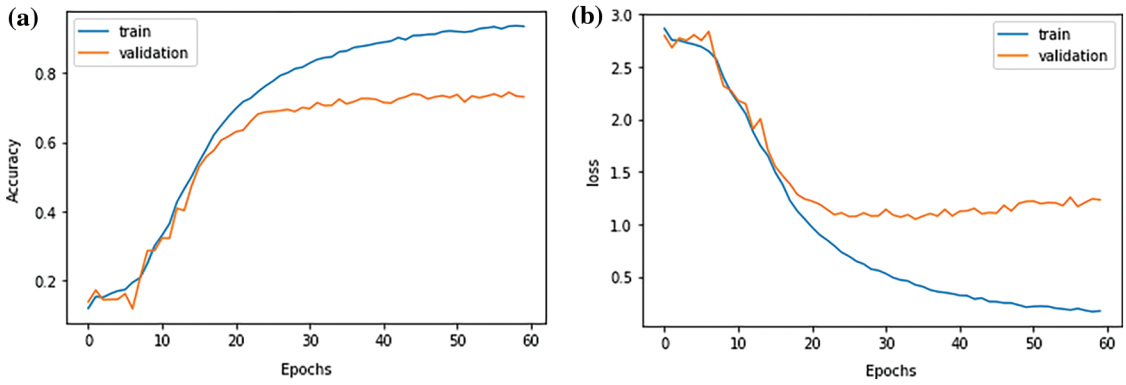

Figure 4: (a) Training and validation accuracy; and (b) Training and validation loss of Word2Vec LSTM model

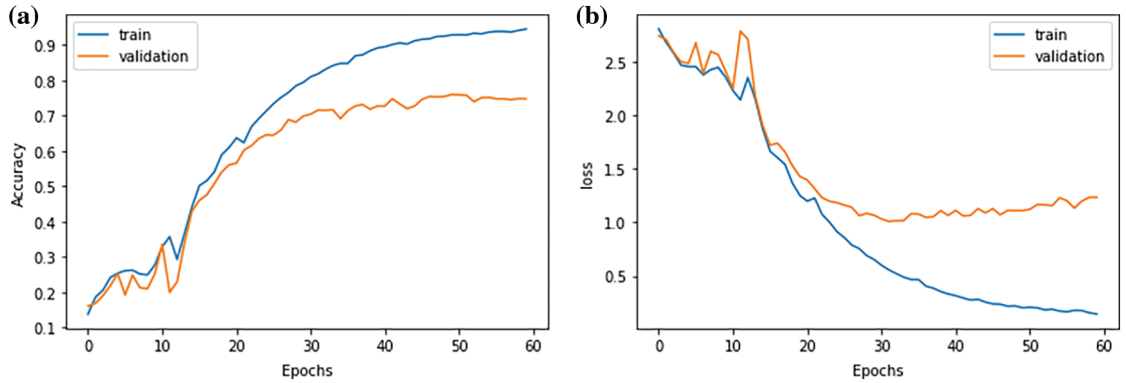

Figure 5: (a) Training and validation accuracy; and (b) Training and validation loss of GLoVe LSTM model

5.3 ML Models

The accuracy value of XGBoost reached 0.92. The best parameters of the base estimator, maximum depth, splitter, and the number of estimators were selected as DT, 2, random, and 2, respectively, in the case of AdaBoost to achieve the accuracy of 0.87. It was found that the best KNN model contained the number of neighbor values of 4 to achieve an accuracy of 0.80. The RF model contained the number of the estimator and the maximum depth of 10 and 40, respectively, to achieve an accuracy of 0.87. It was interesting to find that the classification accuracy and other LR matrices were similar to the KNN model. The best accuracy of the LR model was 0.8 for a learning rate of 0.1. The classification report (Tab. 4) found that the accuracy of the LR model was 0.8 with a good recall and f-1 score.

Table 4: Accuracy metrices for ML models

ML models	Precision	recall	f1-score	accuracy
XGBoost	0.92	1.00	0.95	0.92
AdaBoost	0.87	1.00	0.93	0.87
KNN	0.86	0.92	0.89	0.80
RF	0.92	0.92	0.92	0.87
LR	0.86	0.92	0.89	0.80

5.4 Challenges and Future Scope

Availability of a limited volume of real data was the most critical challenge against insider threat detection; however, the present investigation used the real dataset of the Enron corpus. Another critical challenge is ethical and privacy issues; the present investigation did not show any e-mail messages or names of employees. The use of e-mail communication is in billions and is rapidly increasing; similarly, the rise of insider threats is also high, although most institutions do not report such incidents to maintain their goodwill in the market. The volume of data was quite high; e.g., in the present investigation, the amount of input and output data was about 30 GB, including Google word2vec embedding for Wikipedia, GLoVe embedding, and Enron e-mail data. The huge quantity of dataset requires good computation. The current investigation used Google Colab with GPU support, making it efficient to handle computation. There are few recommendations for future scope, including detection on real-time data. The non-technical aspects of insider threat detection need to be assimilated with technical issues to complete the ecosystem for better understanding and address the issue in a better way. There are multiple accuracy assessment metrics used to evaluate the insider threat; however, no framework or standard exists in the current time to evaluate the standard for insider threat detection models or tools.

6 Conclusions

Recent studies have suggested that the cost of insider attacks is higher than the external threats, making it an important aspect of information security for organizations. This issue of insider threat detection requires state-of-the-art Artificial Intelligence models and utility. In this paper, an attempt was made to detect insider threats based on the deep learning hybrid model of Word2vecLSTM, GLoVeLSTM, and Machine learning models such as XGBoost, AdaBoost, RF (Random Forest), KNN(K-Nearest Neighbor), and LR (Logistics regression). It was found that the ML-based model XGBoost showed an accuracy of 92%, whereas DL-based word2vecLSTM and GLoVeLSTM achieved accuracy values of 73.4% and 74.00%, respectively. There are a few recommendations for future scope, including detection on real-time data. The non-technical aspects of insider threats need to be assimilated with technical issues to complete the ecosystem for better understanding and address the issue better. There are multiple accuracy assessment metrics used to evaluate insider threats; however, no framework or standard exists in the current time, which can address the evaluation of insider threat detection systems.

Funding Statement: Mohd Abdul Rahim Khan would like to thank the Deanship of Scientific Research at Majmaah University for supporting this work under Project No. R-2021-265.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] I. Homoliak, F. Toffalini, J. Guarnizo, Y. Elovici and M. Ochoa, "Insight into insiders and it: A survey of insider threat taxonomies, analysis, modeling, and countermeasures," *ACM Computing Surveys*, vol. 52, no. 2, pp. 1–40, 2019.

[2] M. N. Al-Mhiqani, R. Ahmad, Z. Z. Abidin, W. M. Yassin, A. Hassan *et al.,* "A new taxonomy of insider threats: An initial step in understanding authorised attack," *International Journal of Information Systems and Management*, vol. 1, no. 4, pp. 343–359, 2018.

[3] J. Kim, M. Park, H. Kim, S. Cho and P. Kang, "Insider threat detection based on user behavior modeling and anomaly detection algorithms," *Applied Sciences*, vol. 9, no. 19, pp. 1–21, 2019.

[4] Crowd Research Partners Insider and Cybersecurity Insiders. In Insider Threat 2018 Report; Cybersecurity Insiders: Washington, DC, USA, 2018. [Online]. Available: https://crowdresearchpartners.com/wp-content/uploads/2017/07/Insider-Threat-Report-2018.pdf (Accessed on 7 April 2021).

[5] L. L. Ko, D. M. Divakaran, Y. S. Liau and V. L. Thing, "Insider threat detection and its future directions," *International Journal of Security and Networks*, vol. 12, no. 3, pp. 168–187, 2017.

[6] Crowd Research Partners (2019): "2019 insider threat report,". Accessed: Jun.1, 2021. [Online]. Available: http://www.securonix.com/resources/2019-insider-threat-survey-report/.

[7] B News. (2014). "Edward snowden: Leaks that exposed US Spy programme," Accessed: Aug. 15, 2019. [Online]. Available: https:// www.bbc.com/news/world-us-canada-23123964.

[8] The Guardian. (2013). "Bradley manning prosecutors Say soldier `Leaked sensitive Information'. Accessed: Aug 15, 2019. [Online]. Available: https://www.theguardian.com/world/2013/jun/11/bradley-manningwikileaks-trial-prosecution.

[9] Famous Cases & Criminals: Robert Hanssen. Accessed: Aug. 15, 2019. [Online]. Available: https://www.fbi.gov/history/famouscases/robert-hanssen.

[10] D. M. Cappelli, A. P. Moore and R. F. Trzeciak, "The CERT guide to insider threats: how to prevent, detect, and respond to information technology crimes (Theft, Sabotage, Fraud)," Addison-Wesley, 2012.

[11] L. Xiao, G. Wang and Y. Zuo, "Research on patent text classification based on word2vec and lstm," in *Proc. ISCID*, Hangzhou, China, pp. 71–74, 2018.

[12] P. F. Muhammad, R. Kusumaningrum and A. Wibowo, "Sentiment analysis using word2vec and long short-term memory (lstm) for Indonesian hotel reviews," *Procedia Computer Science*, vol. 179, pp. 728–735, 2021.

[13] G. Di Gennaro, A. Buonanno and F. A. Palmieri, "Considerations about learning word2vec," *The Journal of Supercomputing*, vol. 77, pp. 1–16, 2021.

[14] G. Rao, W. Huang, Z. Feng and Q. Cong, "LSTM with sentence representations for document-level sentiment classification," *Neurocomputing*, vol. 308, pp. 49–57, 2018.

[15] K. Chen, R. J. Mahfoud, Y. Sun, D. Nan, K. Wang *et al.,* "Defect texts mining of secondary device in smart substation with GloVe and attention-based bidirectional LSTM," *Energies*, vol. 13, no. 17, pp. 1–17, 2020.

[16] J. Pennington J, R Socher and C. D. Manning, "Glove: Global vectors for word representation," in *Proc. EMNLP*, Doha, Qatar, pp. 1532–1543, 2014.

[17] P. Parveen, N. McDaniel, V. S. Hariharan, B. Thuraisingham and L. Khan, "Unsupervised ensemble based learning for insider threat detection," in *Proc. Int. Conf. on Privacy, Security, Risk and Trust*, Amsterdam, Netherlands, pp. 718–727, 2012.

[18] F. Y. Leu, K. L. Tsai, Y. T. Hsiao and C. T. Yang, "An internal intrusion detection and protection system by using data mining and forensic techniques," *IEEE Systems Journal*, vol. 11, no. 2, pp. 427–438, 2015.

[19] J. M. Vidal, A. L. S. Orozco and L. J. G. Villalba, "Online masquerade detection resistant to mimicry," *Expert Systems with Applications*, vol. 61, pp. 162–180, 2016.

[20] P. Parveen, Z. R. Weger, B. Thuraisingham, K. Hamlen and L. Khan, "Supervised learning for insider threat detection using stream mining," in *Proc. Int. Conf. on Tools with Artificial Intelligence*, Boca Raton, FL, USA, pp. 1032–1039, 2011.

[21] P. Parveen, N. Mcdaniel, Z. Weger, J. Evans, B. Thuraisingham *et al.,* "Evolving insider threat detection stream mining perspective," *International Journal on Artificial Intelligence Tools*, vol. 22, no. 5, pp. 1360013, 2013.

[22] M. Kandias, D. Gritzalis, V. Stavrou and K. Nikoloulis, "Stress level detection via OSN usage pattern and chronicity analysis: An OSINT threat intelligence module," *Computers & Security*, vol. 69, pp. 3–17, 2017.

[23] Y. Park, I. M. Molloy, S. N. Chari, Z. Xu, C. Gates *et al.,* "Learning from others: user anomaly detection using anomalous samples from other users," in *Proc. ESORCS*, Vienna, Austria, pp. 396–414, 2015.

[24] S. Panigrahi, S. Sural and A. K. Majumdar, "Two-stage database intrusion detection by combining multiple evidence and belief update," *Information Systems Frontiers*, vol. 15, no. 1, pp. 35–53, 2013.

[25] E. Yuan and S. Malek, "Mining software component interactions to detect security threats at the architectural level,". in *Proc. WICSA*, Venice, Italy, pp. 211–220, 2016.

[26] P. Parveen, J. Evans, B. Thuraisingham, K. W. Hamlen, and L. Khan, "Insider threat detection using stream mining and graph mining," in *Proc. PASSAT-SOCIALCOM*, Boston, MA, USA, pp. 1102–1110, 2011.

[27] J. White and B. Panda, "Insider threat discovery using automatic detection of mission critical data based on content,". in *Proc. Int. Conf. on Information Assurance and Security*, Atlanta, GA, USA, pp. 56–61, 2010.

[28] C. H. Hsieh, C. M. Lai, C. H. Mao, T. C. Kao and K. C. Lee, "AD2: Anomaly detection on active directory log data for insider threat monitoring," in *Proc. ICCST*, Taipei, Taiwan, pp. 287–292, 2015.

[29] Y. Hu, C. Frank, J. Walden, E. Crawford and D. Kasturiratna, "Profiling file repository access patterns for identifying data exfiltration activities". in *Proc. CICS*, Paris, France, pp. 122–128, 2011.

[30] T. Rashid, I. Agrafiotis and J. R. Nurse, "A new take on detecting insider threats: Exploring the use of hidden markov models," in *Proc. MIST*, Vienna Austria, pp. 47–56, 2016.

[31] R. Pagliari, A. Ghosh, Y. M. Gottlieb, R. Chadha, A. Vashist *et al.,* "Insider attack detection using weak indicators over network flow data," in *Proc. MILCOM Military Communications Conf.*, Tampa, FL, USA, pp. 1–6, 2015.

[32] O. Lo, W. J. Buchanan, P. Griffiths and R. Macfarlane, "Distance measurement methods for improved insider threat detection," *Security and Communication Networks*, vol. 2018, pp. 1–18, 2018.

[33] S. D. Bhattacharjee, J. Yuan, Z. Jiaqi and Y. P. Tan, "Context-aware graph-based analysis for detecting anomalous activities," in *Proc. ICME*, Hong Kong, China, pp. 1021–1026, 2017.

[34] L. Lin, S. Zhong, C. Jia and K. Chen, "Insider threat detection based on deep belief network feature representation," in *Proc. ICGI*, Fuzhou, China, pp. 54–59, 2017.

[35] W. Li, W. Meng, L. F. Kwok and H. S. 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.

[36] Y. Chen, S. Nyemba, W. Zhang and B. Malin, "Leveraging social networks to detect anomalous insider actions in collaborative environments," in *Proc. Intelligence and Security Informatics*, Boston, MA, USA, pp. 119–124, 2011.

[37] L. Nkosi, P. Tarwireyi and M. O. Adigun, "Detecting a malicious insider in the cloud environment using sequential rule mining," in *Proc. Adaptive Science and Technology*, Pretoria, South Africa, pp. 1–10, 2013.

[38] S. Zou, H. Sun, G. Xu and R. Quan, "Ensemble strategy for insider threat detection from user activity logs," *Computers Materials & Continua*, vol. 65, no. 2, pp. 1321–1334, 2020.

[39] P. Chattopadhyay, L. Wang and Y. Tan, "Scenario-based insider threat detection from cyber activities," *IEEE Transactions on Computational Social Systems*, vol. 5, no. 3, pp. 660–675, 2018.

[40] W. Li, W. Meng, L. F. Kwok and H. S. 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.

[41] 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, 2017.

[42] J. S. Okolica, G. L. Peterson and R. F. Mills, "Using PLSI-U to detect insider threats by datamining e-mail," *International Journal of Security and Networks*, vol. 3, no. 2, pp. 114–121, 2008.

[43] W. Eberle, J. Graves and L. Holder, "Insider threat detection using a graph-based approach," *Journal of Applied Security Research*, vol. 6, no. 1, pp. 32–81, 2010.