Computer Systems Science & Engineering
DOI:10.32604/csse.2022.019940
Article

Tech Science Press

Insider Attack Detection Using Deep Belief Neural Network in Cloud Computing

A. S. Anakath[1,*], R. Kannadasan[2], Niju P. Joseph[3], P. Boominathan[4] and G. R. Sreekanth[5]

[1]School of Computing, E.G.S. Pillay Engineering College, Nagapattinam, 611002, Tamilnadu, India
[2]Department of Software Systems, School of Computer Science and Engineering (SCOPE), Vellore Institute of Technology University, Vellore, 632014, Tamilnadu, India
[3]Department of Computer Science, CHRIST(Deemed to be University), Bengaluru, 560029, India
[4]Department of Information Security, School of Computer Science and Engineering (SCOPE), Vellore Institute of Technology University, Vellore, 632014, Tamilnadu, India
[5]Department of Computer Science and Engineering, Kongu Engineering College, Erode, 638060, Tamilnadu, India
*Corresponding Author: A. S. Anakath. Email: asanakathresearch@gmail.com

Received: 02 May 2021; Accepted: 17 June 2021

Abstract: Cloud computing is a high network infrastructure where users, owners, third users, authorized users, and customers can access and store their information quickly. The use of cloud computing has realized the rapid increase of information in every field and the need for a centralized location for processing efficiently. This cloud is nowadays highly affected by internal threats of the user. Sensitive applications such as banking, hospital, and business are more likely affected by real user threats. An intruder is presented as a user and set as a member of the network. After becoming an insider in the network, they will try to attack or steal sensitive data during information sharing or conversation. The major issue in today's technological development is identifying the insider threat in the cloud network. When data are lost, compromising cloud users is difficult. Privacy and security are not ensured, and then, the usage of the cloud is not trusted. Several solutions are available for the external security of the cloud network. However, insider or internal threats need to be addressed. In this research work, we focus on a solution for identifying an insider attack using the artificial intelligence technique. An insider attack is possible by using nodes of weak users' systems. They will log in using a weak user id, connect to a network, and pretend to be a trusted node. Then, they can easily attack and hack information as an insider, and identifying them is very difficult. These types of attacks need intelligent solutions. A machine learning approach is widely used for security issues. To date, the existing lags can classify the attackers accurately. This information hijacking process is very absurd, which motivates young researchers to provide a solution for internal threats. In our proposed work, we track the attackers using a user interaction behavior pattern and deep learning technique. The usage of mouse movements and clicks and keystrokes of the real user is stored in a database. The deep belief neural network is designed using a restricted Boltzmann machine (RBM) so that the layer of RBM communicates with the previous and subsequent layers. The result is evaluated using a Cooja simulator based on the cloud environment. The accuracy and F-measure are highly improved compared with when using the existing long short-term memory and support vector machine.

Keywords: Cloud computing; security; insider attack; network security; privacy; user interaction behavior; deep belief neural network

1 Introduction

The information technology (IT) world has a new revolution technology called cloud computing. A cloud infrastructure has many benefits, such as dynamic resource utilization, on-demand storage processing, and sharing unlimited resources. Cloud computing is creating a new revolution in the IT sector, but security is the major problem. The network security is highly affected by outsider and insider intruders. Providing external security is very easy, but internal security is a very difficult task. Internal intruders are pretending to be real users in the network. They tend to use authorized nodes and collapse the entire cloud network by attacking sensitive information [1]. External security has advanced firewall software to protect the system from outsider threats. When compared with an external threat, an insider attack is the most dangerous one. Insider attacks take place by following loopholes, such as using an authorized node or attacking using an authorized ID or malicious attack pretending to be a trusted node or stealing sensitive information as a user. If an intruder is present in the cloud network, then the intruder lays a way for other malicious nodes to enter the network. Recently, several studies on security threats in cloud computing exist.

In this research article, we consider abnormal behavior as a threat. However, such an abnormal behavior may have some reasons, such as a broken node that cannot work normally. Identifying malicious behavior is very challenging among insider attacks. Hackers inside the network look for the weakest node and then attack the sensitive data in the cloud server. Authorized user attackers as an insider affect the cloud network in privacy preservation. They pretend to a real user and obtain all legal services from the cloud service provider. These problems of internal attack are currently handled using various machine learning (ML) approaches.

Organizations face most damages on reputation, financial data, and enterprise property because of internal threats. As of the 2018 report, 53% of the attack happens because of insider hacking, and 28% are internal attackers adjudged as an organization origin [2,3]. The leakage of data statistics conducted states that many internal or insider threats are noticed by the media. The big solution from most organizations tends to buy or design powerful firewalls, cyber techniques, intrusion identification, and digital monitoring system to identify insider threats. Identifying the wrong or malicious nodes in the organization is possible using the threat detection technique. This technique provides mitigation measures and detection before attacking. Whatever technology precipitates, identifying and understanding insider attacks are difficult. Every existing technique has some limitations, and many solutions fail to detect exact insiders. Therefore, studying the limitation of existing internal attack algorithms and identifying the solution for limitations are necessary. Recently, ML and deep learning techniques provide a solution for most security issues in the cloud network. Among them, identification of user behavior in various perspectives is highly motivated to obtain a better solution.

The best-case study of insider threats includes one of the famous problem cases in Wiki Leaks on July 25, 2010, where a diary during the Afghan war is released as a document containing more than 90,000 reports. The diary describes the Afghan war from 2004 to 2010. This leakage is caused by an insider army of the US. He is a worker in army analytic in the intelligence department, making government communities and business organizations pay attention to insider threats. The “insider threat” among scholars is defined as an authorized access node misused by an authorized user to harm or other intentions. Many ML techniques focus on determining a negative intention.

A spammer user identification method is an ML approach used to learn the behavior of the user in the cloud network [4]. The collected behavior is classified using deep learning technology. The training layer detects the behavior and classifies them as a normal or abnormal user. To date, cloud service providers have not provided adequate datasets in real time. Considering the confidential data process, we cannot obtain real datasets. Hence, we use supervised algorithms to look for real datasets for training the model. Training the model without a proper dataset is very insufficient and hard. Moreover, unsupervised learning does not require pre-training of dataset in the training model but takes input data at a time and trains the model automatically during the process.

The traditional approach fails to take sufficient data to classify insider attacks. The techniques such as long term, short term, and support vector machine (SVM) do not meet the real-time needs of organizations. In this proposed model, we plan to train the incoming data dynamically using the user behavioral approach. The behavior pattern is collected using user mouse movements and clicks and keystroke details. Every user has a unique behavior. Based on this sensitive behavior recognition, this research contributes as follows:

1. We focused on detecting the insider using their unique character or behavior. Every person in the organization has individuality in the usage of mouse movements and clicks and keystrokes. This research collects the user behavior on mouse movements and clicks and keystrokes for the best feature extraction process.
2. A cruel insider from the organization has been identified by monitoring their interaction behavior. If changes in normal behavior exist, then our system detects and alerts the organization.
3. The ML strategy of the deep learning technique using a belief neural network is designed to train the user interaction behavior and detect abnormalities from the trained model.
4. The result of the proposed model is compared with that of the existing techniques such as SVM and long-term memory. The accuracy of the proposed research article is highly improved compared with that of the existing models.

The paper is further organized as follows. Section 2 reviews previous works to improve the present algorithm. Section 3 computes the proposed model with an architectural summary and workflow progress. Section 4 describes the result and evaluation. Section 5 concludes the study and proposes future scope.

2 Related Work

External threats are greatly influenced by internal attack detection techniques. The detection of an internal threat is fully based on the strategy of overflowing using a buffer [5]. An internal threat does not provide efficient metrics for the testing environment. Testing on real data has a very limited measure. The traditional insider attack detection can be classified as host-based, networking-based, and information-based detections. The user behavior of the host nodes is monitored in this research article. Host-based intruder detection is conducted by monitoring the user behavior pattern. First, the behavior pattern is recorded, and ML techniques are used to detect the changes in the normal behavior pattern. The supervised and unsupervised learning methods are used.

To detect an insider attack, biometric data play a major role in the authentication process nowadays [6]. The user biometric details and psychological character are stored as testing data. Psychological characteristics such as iris and fingerprint are used. Behavioral data are nothing but mouse movement and click and keystroke patterns of the users. In most detections, physiological biometric data have poor processing applicability and is as simple as password hacking. Keystroke-based internal threat detection is first proposed in 1999. Some behavior characteristics of the user attract attention for security purposes, and scholars tend to pay more attention to such characteristics for threat detection. The different anomaly detection techniques [7,8] using mouse patterns are discussed. The patterns are based on mouse movement angle, distances moved, acceleration performed, and others, and keystroke patterns include stroking interval, duration, and valued pairs.

A system based on the European standard [9] for present threat detection techniques on user behavior has a problem with high accuracy. To date, these techniques cannot meet the requirement of security concerns. In most of the present techniques, the output of the false alarm rate is very high. Notably, existing research focused on biological identification as security characteristics. Biological identification is considered an abnormal system model because of lagging in a high-threat detection strategy. These methods consider that the user behavior pattern is very difficult and less potential tasks. Authenticating the user is considered a text classification problem by processing system commands [10]. Here, they use an n-gram framework to process word sequences. The parameters of the system call include the status of processing data and values of results. The system and user calls tend to establish a relationship between the user and process for threat detection [11].

The hidden Markova model is a dynamic method [12] with the distributive static model combined to detect anomalies in the network. The process of detection is based on the system call by monitoring all the activities on the host side. During the detection process, an overhead by different threat behaviors exists, which leads to the highest false reading. To overcome the above problems, that is, large data log process and drifting problems in training data, article [13] proposed a technique based on the training data set. A user of this model tends to store its access behavior and characteristics. This behavior pattern is stored as sequences for processing the queries. This technique provides great advantages by reducing a load of data and continuous monitoring of user behavior habits. The main problem with this method is that it is applicable only to certain applications. In our research work, we use the ML model with user behavior to improve dataset and accuracy.

Unsupervised learning algorithms are mostly tried for internal threat detection in existing systems. This technique is easy because it does not require labeled or trained datasets. This algorithm directly uses unlabeled data to find abnormalities in user behavior patterns. The main significance pattern in this learning model is that it does not require training before the examination. This algorithm tends to detect irregular patterns in high-dimensional data automatically without human intervention. Threat detection using a graph is proposed using the unsupervised strategy of [14,15]. This technique keeps the data statically and checks the abnormality in the normal host nodes. Once the threat is detected, an unsupervised classifier is used dynamically for mining accurate threats [16].

The problem of data leakage in the developing technological world is addressed using the innovative data-centric model [17]. Insider threats are continuously growing and breaking the trust of online communication. Here, user behaviors are observed as keystroke details and user addressed consecutive pattern of queries. These details are considered in the training model, and finally, abnormality in this operation is detected. First [18], the above article uses mouse patterns for detecting threats from an insider. The behavior model is recognized with abnormalities using SVM, probabilistic neural network, and decision tree. The feature combination of keystroke with a biometric pattern is used [19–21] for insider threat detection. They used a statistical algorithm and ensemble classification strategy with network access control for detecting the threat. The intelligence in pattern recognizing using a user typing pattern and keystroke data is used [22–24] for accurate threat detection. Detection is performed using the outlier classifier model locally and two-class classification algorithms. Features with user command [25–27] are computed with dynamical theory on the system. Threatened data are mined using the Lempel Ziv Welch technique, incremental learning methodology, and sequence learning alignment. The stream mining algorithm with a decision graph is used as ensemble techniques [28–30] for insider threat detection earlier. This algorithm is a supervised learning approach that gives good accuracy in predicting threats. A two-way authentication protocol is proposed to authenticate the client and server anonymously [31]. This mechanism preserves the identity of the user in the cloud and fulfils the authentication. A novel RSA algorithm for secured key transmission in a centralized cloud environment is proposed [32]. This algorithm encrypts the keys used in group environment. Both these algorithms are said to be computationally efficient.

A new model for insider threat using the ML algorithm is addressed for best performance. This research article uses improvised long short-term memory (ILSTM) to learn the behavior of the user. ILSTM trains automatically with the behavioral patterns of various users. This algorithm correctly classifies the abnormality in the network from the normal user behavior. Another technique is identifying the insider using packet transactions and traffic in the network. The well cyber-using techniques are honey bot detection, and deception techniques are preferred for behavioral analysis.

3 Proposed User Interaction Behavior with Deep Belief Neural Network Methodology

The data sources in the cloud are valuable information that can be steeled by attackers because cloud security is a major issue. The security of the cloud network is already secured, although internal attackers still exist. Those internal/insider attackers obtain access to valuable information in the system. The information leakage from the cloud server through an insider is a major issue that leads to the loss of data. In the current research, ML algorithms are used widely for cloud security. ML algorithms with behavioral analysis will help detect and predict insider threats, such as employees and contractors. In this study, to detect insider threats, the combination of interaction behavioral characteristics of the insiders, such as keystroke and mouse dynamics, which are considered for feature extraction and deep learning algorithm called deep belief network (DBN), has been used to predict the abnormal behavior of the insiders in the cloud network. Fig. 1 shows the proposed system overview.

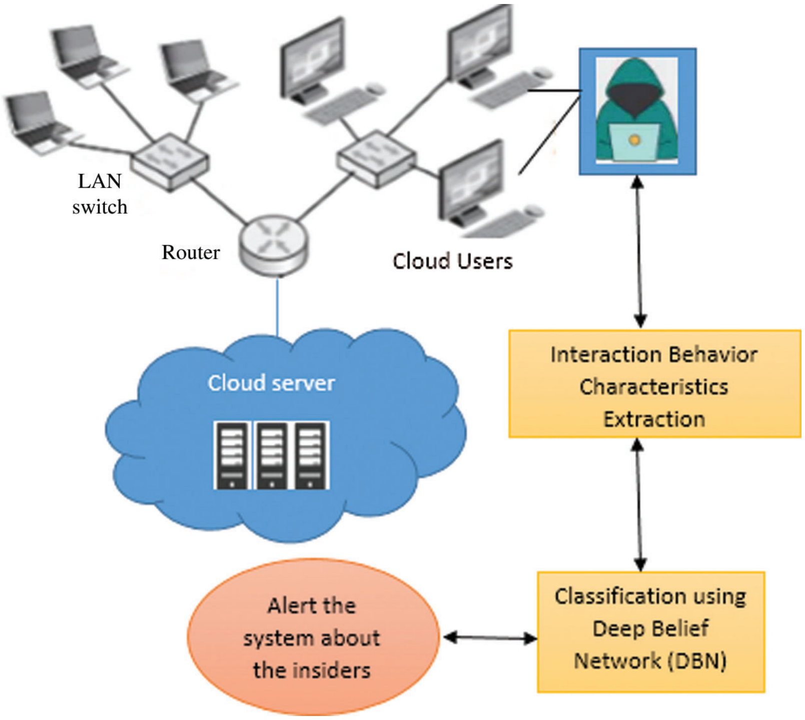

LAN
switch
Router
Cloud Users
Cloud server
Interaction Behavior Characteristics Extraction
Classification using Deep Belief Network (DBN)
Alert the system about the insiders

Figure 1: Overview of proposed cloud insiders' threat detection

3.1 Feature Extraction of Behavioral Characteristics of Cloud Users

Insider threats pertain to the security threats caused by people in the organization called employees. These illegitimate accesses of the information may negatively affect the organization's policies and leads to loss of data. These attackers are categorized into two types: traitors and masqueraders. The former are a familiar person who knows the information and assets of the organization, and their behavior fulfills the security policies of the organization. In comparison, the latter are the attackers who gain illegal access to the identity of legitimate users. Moreover, traitors have more information about the access policies, data storage location, and intellectual property details than masqueraders. In this study, the cruel insider from the organization is identified by monitoring their interaction behavior. They are working with their corresponding duties, but their activities are not normal. Based on their actions, the normal users are identified as malicious users. The interaction of the user is categorized into two, namely, rich keystroke and rich mouse operations. With these two kinds of operations, the user can interact with all kinds of data and applications in the cloud system. We assume that these two interaction operations range [0, 1]. The 0 value means key operations, whereas 1 means mouse operations.

3.1.1 Mouse Operations

The mouse movements, such as dragging, clicking, and double clicking, occur when the user wants to select or edit the application/data in the cloud system. The direction of the mouse movement is categorized into eight, and the mean and variance of distance and speed of the mouse movement are recorded. The characteristics of the mouse movements, such as mouse right click (MRclick), mouse left click (MLclick), mouse double click (MDclick), and mouse drag (MDrag), are distance and speed. Distance is calculated for any three consecutive points, that is, A, B, and C, and the length of AC from A to C. Angle is represented as ∠ABC. Tab. 1 presents the mouse operation calculation of distance and speed.

Table 1: Mouse operations

Mouse movements	Distance	Speed
MDirection0	Yes	Yes
MDirection1	Yes	Yes
MDirection2	Yes	Yes
MDirection3	Yes	Yes
MDirection4	Yes	Yes
MDirection5	Yes	Yes
MDirection6	Yes	Yes
MDirection7	Yes	Yes
Mrclick	Yes	Yes
Mlclick	No	No
Mdclick	Yes	No
Mdrag	No	No

Distance is calculated using the N grams method. For a discrete value, the query of k^{th} behavior is represented as $K = \{k_1, k_2, \ldots k_N\}$, and its frequency is defined in Eq. (1),

$$F_n(K) = \{n - gram|n - gram = \langle k_i,\ ..k_{i+1}\rangle i \in [i,\ N + 1 - n].\tag{1}$$

The distance between the two lists is defined using the Jaccard coefficient as follows:

$$D_n(K_1,\ K_2) = 1 - \frac{|F_n(K_1) \cap_n^F (K_2)|}{|F_n(K_1) \cup_n^F (K_2)|}.\tag{2}$$

3.1.2 Keystroke Operations

Keystroke events occur when we press the keys for a particular function. Tab. 2 shows the key events used in this work. They are KNum (includes the number keys 1, 2, 3, …), KAlpha (Alphabets, such as A, a, b, c, …), KShift (Shift+num, alpha), KCtrl (ctrl keys, alt, win), KDir (key such as ->, <-), KFunc (F1, F2, …), and KOhter (tab, capslocak, …). The characteristics of the key events are duration and latency. Duration is defined as the difference between the time of key press and key released, which is shown in Eq. (3) as follows:

$$duration(user,\ K) = time_p(u,\ K) - time_r(u,\ K),\tag{3}$$

where u-user, $time_p(u, K)$ pertains to the time to press the key of the user u, and $time_r(u, K)$ is the time to release the key of the user u. The latency of the keyboard not used is defined in Eq. (4).

Table 2: Key events and their characteristics

Key events	Duration	Latency
Knum	Time (up, down)	Time (key release, key press)
Kalpha	Time (up, down)	Time (key release, key press)
Kshift	Time (up, down)	Time (key release, key press)
KCtrl	Time (up, down)	Time (key release, key press)
Kdir	Time (up, down)	Time (key release, key press)
Kfunc	Time (up, down)	Time (key release, key press)
Kother	Time (up, down)	Time (key release, key press)

$$L(u,\ K) = time_r(u,\ K) - time_p(u,\ K).\tag{4}$$

The feature vector of these keystroke operations is defined in Eq. (5).

$$\{\langle duration(u,\ K_1),\ latency(u,\ K_1)\rangle,\ \ldots\langle duration(u,\ K_n),\ latency(u,\ K_n)\rangle.\tag{5}$$

With these extractions of user interaction behavior using the mouse movements and clicks and keystroke operations, the deep learning-based classification is performed to identify the malicious insider in the cloud.

3.2 Insider Threat Classification Using DBN

This study uses a deep learning-based classification algorithm called DBN to identify the threat from the cloud user. The standard neural network-based classification consists of three layers, such as input, hidden, and output layers. Many researchers modified the standard neural network structure through deep learning to improve classification accuracy. Deep learning is the extension of a standard neural network with stacked hidden layers. Deep learning can handle very large, high-dimensional data with billions of parameters. Among the various deep learning algorithms, DBN is recently popular in ML because of its promised advantages including fast implication and ability to handle larger and higher network structures. DBNs are generative models comprising multiple layers of hidden units. DBN consists of one visible layer and multiple hidden layers. A visible layer is responsible for transmitting the input data to the hidden layers to complete the ML process. This layer is based on a restricted Boltzmann machine (RBM), and the layer of RBM communicates with previous and subsequent layers. Each respective RBM contains two sub-layers, such as visible and hidden layers. The connection between the visible and hidden layers in the RBM is restricted. The transformation process of data from visible to hidden layer is activated through the sigmoid function based on the RBM learning rule. Fig. 2. shows the architecture of DBN with RBM. The architecture consists of three stacked RBMs. RBM1 consists of visible and hidden layer 1, RBM2 consists of hidden layers 1 and 2, and RBM3 consists of hidden layers 2 and 3.

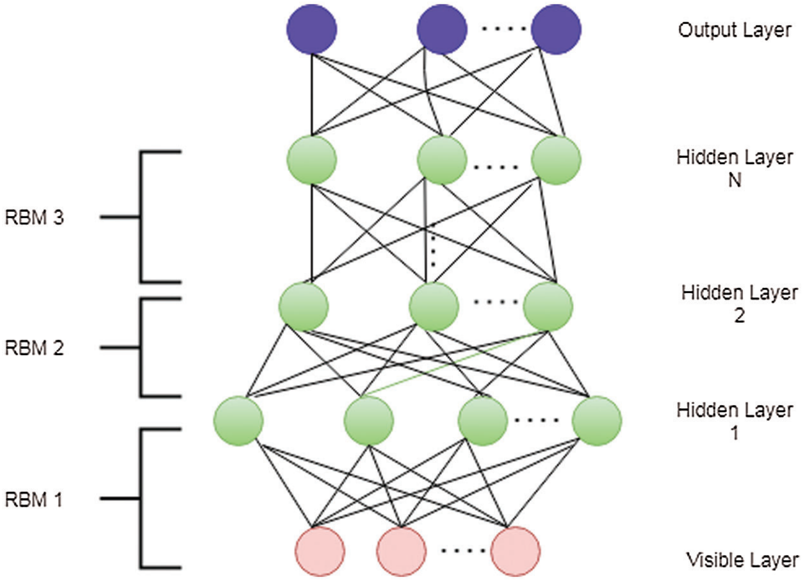


Figure 2: DBN with RBM

The DBN structure with stacked RBM, which is the training process of the DBN classifier, is based on the training of each RBM with a learning rule. The parameters for training include synaptic weight among the layers, bias, and states of the neurons. The state of each neuron in each RBM is formed by transforming the bias and state of the neuron from the previous layer weight to the next layer. The sigmoid function is used for this transformation as in Eq. (6).

$$P(s_i = 1) = \frac{1}{1 + \exp(-b_i - \sum_j s_j w_{ij})}. \tag{6}$$

Initially, the synaptic weight and bias of all the neurons in the RBM layer are initialized. Each input training data consists of two phases: positive and negative. A positive phase converts the data from the visible to the hidden layer, whereas a negative phase converts the data from the hidden to the corresponding visible layer. The individual activation of positive and negative phases is calculated using Eqs. (7) and (8), respectively.

$$P(v_i = 1|h) = sigm(-b_i - \sum_j h_j w_{ij}), \tag{7}$$

$$P(h_i = 1|v) = sigm(-c_i - \sum_j h_j w_{ij}). \tag{8}$$

Compared with the normal DBN, in this work, the weight parameters are optimized until the maximum number of epochs is reached. The training process continues, and the parameters are optimized using Eq. (9).

$$update\left(w_{ij} + \frac{\eta}{2} \times (positive(E_{ij}) - negative(E_{ij}))\right), \tag{9}$$

where, $positive(E_{ij})$-Positive statistics of edge $E_{ij} = p(h_j = 1|v)$, $negative(E_{ij})$-Positive statistics of edge $E_{ij} = p(v_j = 1|h)$, η- learning rate.

The mentioned process is for the training of one RBM, and the same process will repeat until all the RBM are trained. Fig. 3 depicts the workflow of DBN.

Algorithm 3: (improved DBN)

```
Input: keystroke and mouse features, Maxepoch, bias, synaptic weight, epoch = 1
Output: trained network with classification
Step 1: Train the first layer of the RBM with the input of the first hidden layer as $h^0$. Using Eqs. (7) and (8)
Step 2: while ( epoch<=maxepoch)
Step 3:     update weight and bias using Eq. (9).
Step 4:     the first-layer representation is used for the next layer as P ($h^1 = 1|h^0$)
Step 5:     train the consecutive layer of RBM
Step 6: end
```

In this proposed work, insider threat detection is considered the binary classification problem. DBN is used as a classifier to detect the insiders who are vulnerable to access the cloud data. The insider is classified with the features extracted from their behavioral interaction to access the application or data through the mouse and keystroke operations. The abnormal access of these keystrokes and mouse movements are calculated and used as a feature for the insider detection classification.

4 Results and Discussions

The simulation of the proposed cloud insider threat detection is experimented using a Cooja simulator based on the cloud environment. The classification model is evaluated with open-source datasets. The dataset is divided into training and test dataset. The proposed DBN provides better results than other ML algorithms, such as SVM and improved LSTM. Fig. 4 shows the simulation set with 20 nodes, where the normal and malicious user nodes are represented in green and red, respectively.

For the simulation, two-user interaction behavior is recorded for evaluation with the access of three applications in the cloud environment, namely, browser, MS-Word, and game. Tab. 3 presents the recorded results.

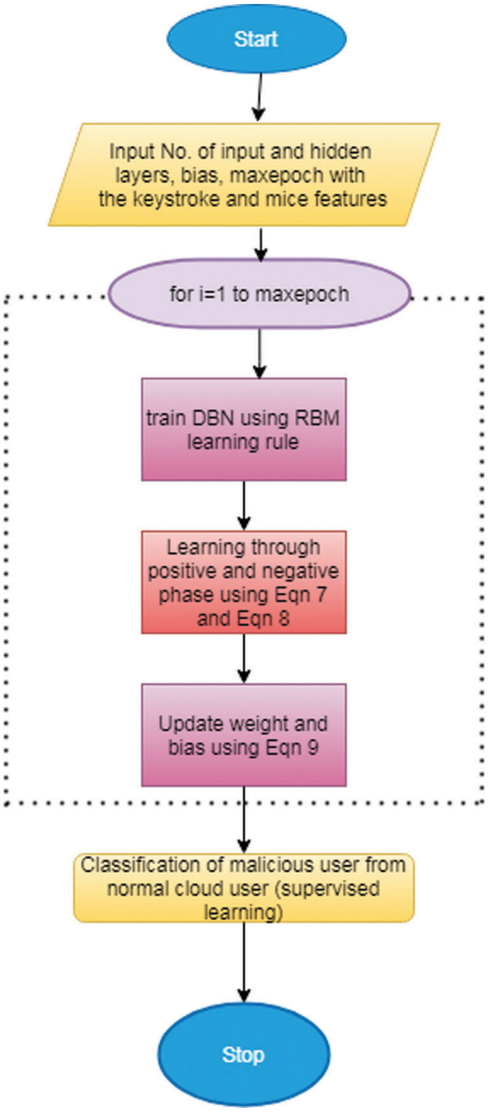

Start
Input No. of input and hidden layers, bias, maxepoch with the keystroke and mice features
for i=1 to maxepoch
train DBN using RBM learning rule
Learning through positive and negative phase using Eqn 7 and Eqn 8
Update weight and bias using Eqn 9
Classification of malicious user from normal cloud user (supervised learning)
Stop

Figure 3: Workflow of the proposed insider threat detection

The interaction in Tab. 3 is calculated as the mouse operation divided by the whole events of keystroke and mouse. With these interaction features, the classification is performed using the proposed deep learning algorithm. To evaluate the effectiveness of the proposed system, the evaluation metrics such as Accuracy and F-Measure are used. The accuracy on detecting the misbehaving user node is rated using the accuracy metrics, and the mean of precision and recall is measured using F-measure. Higher accuracy and F-measure lead to an effective performance of the detection algorithm. These metrics are calculated using Eqs. (10) and (11).

$$Accuracy = \frac{TP + TN}{N}, \tag{10}$$

where N is the total number of sample input data, TP is the rate to detect the malicious insider as a malicious insider, and TN pertains to the rate to detect the normal valid user as a normal valid user.

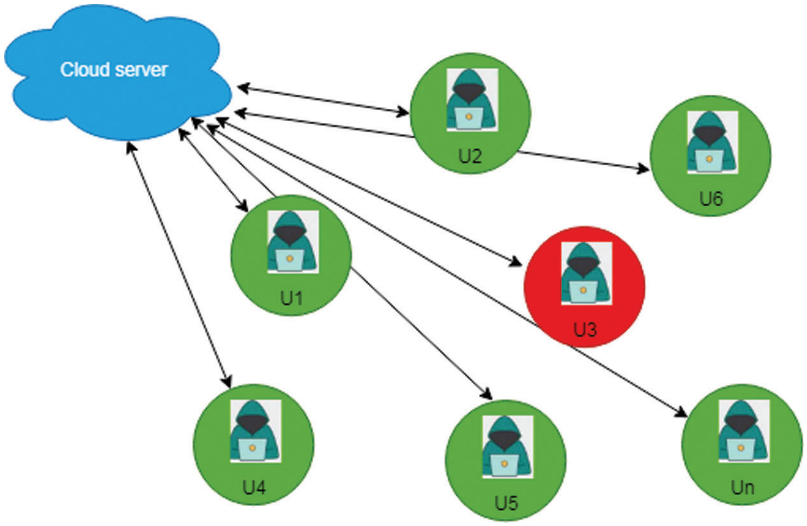


Figure 4: Cloud user simulation setup

Table 3: User interaction behavior details for a sample user

Applications	User 1		
	Keystroke operation	Mouse operation	Interaction
Browser	335 (2.24)	136057 (27.54)	0.92
MS-Word	11100 (74.96)	61500 (12.45)	0.14
Game	3300 (22.76)	29600 (59.91)	0.72

$$F - Measure = \frac{2 * precision * recall}{precison + recall}, \tag{11}$$

$$precision = \frac{TP}{TP + FP}, \tag{12}$$

$$recall = \frac{TP}{TP + FN}, \tag{13}$$

where FP is the rate to detect the valid normal user as a malicious user and FN is the rate to detect the malicious user as a valid normal user. Initially, the user interaction behavior is validated by dividing the test dataset into a block of different sizes. The statistical measurement of each user of the block is calculated. Tab. 4 and Fig. 5 show the evaluated results of proposed methods with the existing cloud insider threat detection scheme.

Table 4: F-measure comparison of cloud insider threat detection methods

No. of data in the block	F-Measure		
	Wang et al. (SVM)	Nathezhtha (ILSTM)	Proposed system (DBN)
500	0.5	0.7	0.9
1000	0.6	0.8	0.92
2000	0.65	0.82	0.95
2500	0.72	0.86	0.97
3000	0.81	0.89	0.98

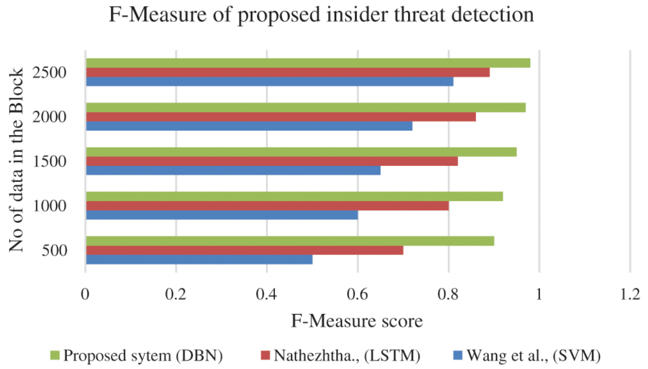
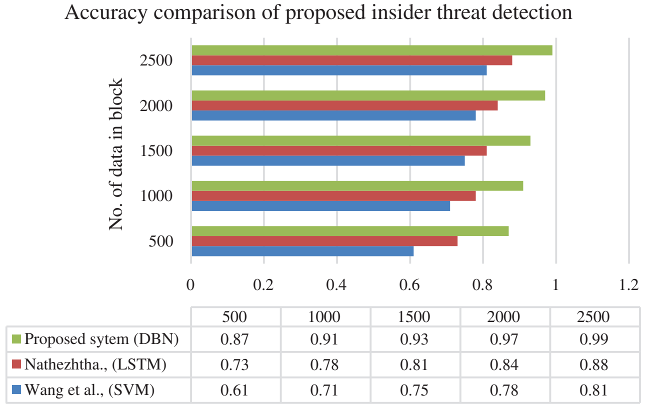


Figure 5: F-measure comparison of cloud insider threat detection

Figs. 5 and 6 show that the proposed system evaluation results are outperforming with high accuracy and F-measure score than those of the existing methods. The measurement of block size also improves the performance of the system. The smaller block size obtains a minimum accuracy compare with the higher block size. The proposed insider threat detection with deep learning network DBN obtains an accuracy of 99% and an F-score of 98%, which are higher than those of the other two approaches. The higher accuracy and F-score prove that the proposed system detection rate is highly efficient in finding malicious cloud users.



	500	1000	1500	2000	2500
🟩 Proposed sytem (DBN)	0.87	0.91	0.93	0.97	0.99
🟥 Nathezhtha., (LSTM)	0.73	0.78	0.81	0.84	0.88
🟦 Wang et al., (SVM)	0.61	0.71	0.75	0.78	0.81

Figure 6: Accuracy comparison of cloud insider threat detection

5 Conclusions

This article tends to achieve high performance in insider attack detection. Internal attackers always seem to be very intelligent to hide as an attacker and work as a trusted authority. IT makes every communication possible via networks with high-speed digital processing techniques. When the world is very happy to establish and use digital communications, others are unhappy because of insider attackers of sensitive and confidential data in the networks. In our article, we introduce the user interaction behavior pattern with a deep belief neural network. The behavior of the user is recognized through detecting mouse movements and clicks and keystrokes in their system and is collected for the training layer in DBN. This study shows the results based on trained pattern unauthorized entry. The accuracy of the proposed model is improved when compared with that of SVM and LSTM. DBN is an ML model, which acts like a neuron based on available knowledge and produces 99% outperforming results.

In future studies, the ensemble deep learning models can be used for detecting the internal attacks in most sensitive applications, such as the military, hospitals, and banking. An ensemble method proves its efficiency and performance in many applications.

Funding Statement: The authors received no specific funding for this study.

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

References

[1] I. Homoliak, F. Tofalini, J. D. Guarnizo, Y. Elovici and M. Ochoa, “Insight into insiders: 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,” *Information System Management*, vol. 1, 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] T. Qiu, H. Wang, K. Li, H. Sheng, A. Sangaiah *et al.*,"A novel machine learning algorithm for spammer identification in industrial mobile cloud computing,” *IEEE Transaction on Industrial Informatics*, vol. 15, no. 4, pp. 2349–2359, 2015.

[5] N. Nguyen, P. Reiher and G. H. Kuenning, “Detecting insider threats by monitoring system call activity,” in *Proc. ISW*, West Point, NY, USA, pp. 45–52, 2003.

[6] R. V. Yampolskiy and V. Govindaraju, “Behavioural biometrics: A survey and classification,” *International Journal of Biometrics*, vol. 1, no. 1, pp. 81–113, 2008.

[7] F. Monrose, M. K. Reiter and S. Wetzel, “Password hardening based on keystroke dynamics,” *International Journal of Information Security*, vol. 1, no. 2, pp. 69–83, 2002.

[8] K. S. Killourhy and R. A. Maxion, “Comparing anomaly-detection algorithms for keystroke dynamics,” in *Proc. DSN*, Lisbon, Portugal, pp. 125–134, 2009.

[9] CENELEC-EN 50133-1, “Alarm systems. access control systems for use in security applications. part 1:System requirements,” Belgium, 1996. [Online]. Available: https://standards.globalspec.com/std/390063/EN%2050133-1.

[10] Y. Liao and V. R. Vemuri, “Using text categorization techniques for intrusion detection,” *in Proc. USENIX Security Symposium*, vol. 12, pp. 51–59, 2002.

[11] D. Y. Yeung and Y. Ding, “Host-based intrusion detection using dynamic and static behavioral models,” *Pattern Recognition*, vol. 36, no. 1, pp. 229–243, 2003.

[12] S. Mathew, M. Petropoulos, H. Q. Ngo and S. Upadhyaya, “A datacentric approach to insider attack detection in database systems,” in *Proc. RAID*, Ottawa, Ontario, Canada, pp. 382–401, 2010.

[13] D. J. Cook and L. B. Holder, "*M Ining Graph Data*," Hoboken, NJ, United States: John Wiley & Sons, 1st Edition. 2006.

[14] W. Eberle and L. Holder, "Discovering structural anomalies in graph-based data," in *Proc. ICDM*, Omaha, NE, USA, pp. 393–398, 2007.

[15] M. M. Masud, Q. Chen, L. Khan, C. Aggarwal, J. Gao *et al.*, "Addressing concept-evolution in concept-drifting data streams," in *Proc. ICDM*, Sydney, NSW, Australia, pp. 929–934, 2010.

[16] X. Wang, Q. Tan, J. Shi, S. Su and M. Wang, "Insider threat detection using characterizing user behavior," in *Proc. DSC*, Guangzhou, China, pp. 476–482, 2018.

[17] X. Chen, J. Shi, R. Xu, S. M. Yiu, B. Fang *et al.*, "PAITS: Detecting masquerader via short-lived interventional mouse dynamics," in *Proc. ATIS*, Melbourne, VIC, Australia, pp. 231–242, 2014.

[18] J. R. Schoenher and R. Thomson, "Insider threat detection: a solution in search of a problem," in *Proc. Cyber Security*, Dubin, Ireland, pp. 1–7, 2020.

[19] L. Nkosi, P. Tarwireyi and M. O. Adigun, "Insider threat detection model for the cloud," in *Proc. Is*, South Africa, pp. 1–8, 2013.

[20] C. Xiaojun, W. Zicheng, P. Yiguo and S. A. Jinqiao, "Continuous re-authentication approach using ensemble learning, *Procedia Computer Science*, vol. 17, pp. 870–878, 2013.

[21] B. Gabrielson, "Who really did it? Controlling malicious insiders by merging biometric behavior with detection and automated responses," in *Proc. SS*, Maui, HI, USA, pp. 2441–2449, 2012.

[22] 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. ESORICS*, Vienna, Austria, pp. 396–414, 2015.

[23] N. Kanaskar, J. Bian, R. Seker, M. Nijim and N. Yilmazer, "Dynamical system approach to insider threat detection," in *Proc. ISC*, Boston, MA, USA, pp. 232–238, 2011.

[24] P. Parveen, N. McDaniel, V. S. Hariharan, B. Thuraisingham and L. Khan, "Unsupervised ensemble-based learning for insider threat detection," in *Proc. PSRT*, Amsterdam, The Netherlands, pp. 718–727, 2012.

[25] 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*, vol. 11, pp. 427–438, 2017.

[26] P. Parveen and B. Thuraisingham, "Unsupervised incremental sequence learning for insider threat detection," in *Proc. ISI*, Arlington, VA, USA, pp. 141–143, 2012.

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

[28] P. Parveen, Z. R. Weger, B. Thuraisingham, K. Hamlen and L. Khan, "Supervised learning for insider threat detection using stream mining," in *Proc. TAIS*, Bocarton, FL, USA, pp. 1032–1039, 2011.

[29] P. Parveen, N. McDaniel, Z. Weger, J. Evans, B. Thuraisingham *et al.*,"Evolving insider threat detection stream mining perspective," *Artificial Intelligence Tools,* vol. 22, no. 1360013, pp. 1–9, 2013.

[30] T. Nathezhtha and S. Vaidehi, "Cloud insider attack detection using machine learning," in *Proc. ICRTAC*, Chennai, India, pp. 60–65, 2018.

[31] A. S. Anakath, S. Rajakumar, F. Al-Turjman, R. Arun-Sekar and K. Kalai-Selvi, "Computationally efficient and secure anonymous authentication scheme for cloud users," *Personal and Ubiquitous Computing*, vol. 1, no. 1, pp. 1–11, 2021.

[32] S. Ambika, S. Rajakumar and A. S. Anakath, "A novel RSA algorithm for secured key transmission in a centralized cloud environment," *International Journal of Communication System*, vol. 33, no. 5, pp. 1–9, 2020.