EAI Endorsed Transactions
on Scalable Information Systems
Research Article EAI.EU

OPIN-ITP: Optimized Physics Informed Network with Trimmed Score Regression Based Insider Threats Prediction in Cloud Computing

B. Gayathri[1,*]

[1]Department of Computer Science, Bishop Heber College (Autonomous), Affiliated with Bharathidasan University, Tiruchirappalli - 620024, Tamil Nadu, India

Abstract

INTRODUCTION: Insider threats are a major issue for cyber security. In contrast to external attackers, insiders have more privileges and authorized access to data and resources, which can cause an organization great harm. To completely understand an insider's activities throughout the organization, a more sophisticated method is needed.
OBJECTIVES: Based on an organization's login activity, this study proposes a novel conceptual method for insider threat detection. Behavioural activities such as HTTP, Email and Login details are collected to create a dataset which is further processed for pre-processing using data transformation and Trimmed Score Regression (TSR).
METHODS: These pre-data are given to the feature extraction process using Deep Feature Synthesis (DFS) extraction. The extracted data are fed to Physics Informed Neural Networks (PINN) for insider threat detection.
RESULTS: The prediction process of PINN was improved through optimally choosing parameters such as learning rate and weight using Hunter-prey Optimization (HPO). The proposed model offers 68% detection rate, 98.4% accuracy, 5% FDR, 95% F1_score and 0.7005 sec execution time.
CONCLUSION: Observed outcomes are compared to other traditional approaches of validation. The contrast with traditional approaches shows that the proposed model provides better outcomes than in traditional models and is therefore a good fit for real-time threat prediction.

Keywords: Insider threats, trimmed score regression, deep feature synthesis extraction, physics informed neural networks, Hunter-prey Optimization.

Received on 11 01 2024, accepted on 10 06 2024, published on 11 07 2024

doi: 10.4108/eetsis.6134

1. Introduction

Insider threats are becoming a significant security risk for many companies [1]. It is widely defined as malicious actions by insiders occurring in a secure environment, often leading to electronic fraud, theft of information, and sabotage of systems. As a result, it can be detrimental to individuals, institutions, and national security [2]. Insider threat identification has received a lot of interest lately from the academic and business communities, whilst the task of detecting insider threats has become more difficult and complex [3]. Insiders first utilize their authorized access to undertake illicit actions by ensuring their invisibility to external network security equipment. Second, there are numerous ways that insider attacks can manifest. A disgruntled employee might, for instance, plant a logic bomb to damage systems or steal intellectual property for their personal gain [4]. The multiplicity of insider attacks increases the difficulty of identifying insider threats. Last but not least, insider threats are frequently carried out by insiders over

*Corresponding author. Email: gayathiriaya@outlook.com

business hours, leading to abnormal insider behaviours sporadically observed across a significant portion of typical working behaviours [5]. Consequently, it makes identifying insider threats more challenging.

Insider threat detection relies on modelling a user's typical behaviour to spot aberrant behaviour [6]. Reducing the risks presented by insiders involves discovering hidden patterns in their malevolent behaviour and distinguishing it from benign behaviour. Typically, domain experts would specify precise criteria for feature extraction and then construct proper categorization models for detecting fraudulent behaviour [7]. The feature engineering approach frequently varies depending on the problem formulation since various field specialists have varied viewpoints on individual behaviour [8]. In parallel, machine learning approaches like Support Vector Machines (SVM) [9], Naive Bayes (NB), Logistic Regression (LR), and Artificial Neural Networks (ANNs) [10] have advanced applied detection techniques from simple statistical methods. However, applying graph intelligence approaches would slightly diminish individual qualities, for example, by placing too much focus on particular connections.

Deep learning techniques have recently been used to find insiders in cloud computing [14]. Deep learning is a major study topic and is employed in many security structures due to its many advantages. It can be applied both under supervision and without it [15]. One area of machine learning that offers many benefits is called deep learning. The algorithms perform better in terms of accuracy and performance than conventional machine learning methods. Some traditional models used for detecting insider thread like Gated Recurrent Unit (GRU) [11], Convolutional Neural Network (CNN) [12], Multi State Long Short Term Memory (MSLSTM), and Deep Belief Network (DBN) [13]. On the other hand, it might make it less likely to identify unusual activity, requires combining user data, and is unable to identify unidentified insider threats [16, 17]. As a result, a strong and effective detection technique ought to combine several independent methods. Using optimized PINN, the proposed approach can enhance insider threat prediction by more precisely identifying behavioural similarities among employees based on organizational links. The key objectives of the proposed model can be listed as follows;

- Optimized PINN with TSS approach was introduced to predict insider thread in cloud computing based on activity status.
- HTTP, email, and login status are gathered to make a dataset, which is pre-processed using data transformation and the TSR approach to fill the missing values.
- Extract the features at the pre-data using deep feature synthesis to extract the data for effective classification.
- PINN classifier predicts the insider thread in an organization using these features. It mainly works on the function of Partial Differential Equations to predict the exact outcomes.
- Prediction performance of PINN is further improved through selecting the optimal parameters, using Hunter-Prey Optimization (HPO).

The following sections make up the remainder of the manuscript: Part 2 lists some of the related work being done to detect insider threat in cloud system. Part 3 presents the recommended strategy for cloud computing and organization privacy architecture. The work's outcome and discussion are presented in Part 4, while the work's conclusion is defined in Part 5.

2. Related Work

The state of the art in recognizing anomalies in cloud computing and where our solution fits in. The following section reviews a few recent studies that used hybrid learning, machine learning, and other statistical techniques to try and perform harmful behaviour recognition.

Al-Mhiqani et al. [18] designed a multilayer structure for the detection of insider threats. The first layer of the system uses multi-criteria decision-making approaches to select the best detection of insider threat classification framework from a variety of models. The selection technique is based on the integration of the entropy-VIKOR approaches. For the second layer, a hybrid insider threat detection method has been introduced. The Misuse Insider Threat Detection (MITD) model is created by applying the random forest algorithm. The K-Nearest Neighbors method has later used to develop an abnormal insider threat detection system. This system's primary flaw is that it is unsuitable for datasets that are unbalanced and have classification issues.

Anakath et al. [19] introduce the deep belief neural network-based user interaction behaviour pattern. By identifying mouse movements, clicks, and keystrokes in their system, the user's behaviour is identified and gathered for the DBN training layer. The results of this study are based on unlawful entrance using a taught pattern. This article typically performs well when it comes to insider attack detection. Internal assailants consistently exhibit a high level of intelligence in disguising themselves as trustworthy authorities. With the use of fast digital processing processes and networks, IT enables all forms of communication. However, crucial applications like those in the military, hospitals, and banking should not use this strategy.

Wang et al. [20] provides techniques for feature extraction and selection which render it possible to summarize and generalize important features. To recognize insider attacks arriving through SAS's untrusted control devices. This detection method uses a sliding window-based sequential classification procedure to find anomalies across several devices without requiring the learning of information collected from each device. First, from Generic Object-Oriented Substation Events (GOOSE) messages, six key network attributes and seven distilled physical attributes based on the overall architecture of distribution substation essential organisms were selected and retrieved. But the cost of the system is high.

Nasir et al. [21] created a method for detecting insider attacks using deep learning, which is described. The main reason for developing this system was to deploy it on user technical data in a company with low memory and processing requirements. In addition, the developed system is simple to use, versatile, and requires little domain experience. Several insider threat scenarios are used for insider threat detection, with "LSTM-Autoencoder" being the chosen technique. However, the system's accuracy and quality are lacking.

Alsowail and Al-Shehari [22] designed technical, behavioural, psychological, and cognitive elements of the insider threat scenario are integrated into a structure. The methodology, which is based on a multi-tiered approach that encompasses pre, in, and post-countermeasures, addresses insider threats systematically. It considers several factors connected to the length of insider employment, from the moment they join an organization until their departure. Instances of insider threats in the real world exploit the architecture. However, this approach is not appropriate for huge datasets.

Al Razib et al. [23] presented an intrusion detection system (IDS) that combats growing cyber threats in the IoT by utilizing Software Defined Networking with Deep Learning assistance. In order to counter a comprehensive array of possible security threats, such as denial-of-service (DDOS), port scanning, man-in-the-middle (MITM), infiltration, botnet, brute force, and others, DNN-LSTM offers corresponding strength. But the design of the method is complicated.

Sheykhkanloo and Hall [24] utilize a prevalent balancing method called spread subsample to handle insider threat detection on a highly imbalanced dataset. The results demonstrate that while employing this strategy to balance the dataset did not enhance performance measures, it did reduce the amount of time needed for testing and model development. The authors also found that, when the imbalanced dataset is utilized, the effect of the chosen classifiers is considerable, regardless of whether balanced or unbalanced situations are affected when different parameters are employed. But the accuracy of this system needs to improve.

Haq et al. [25] suggested a two deep learning hybrid LSTM models merged with Google's Word2vec LSTM-GLoVe-LSTM were initially constructed in an effort to close these gaps by detecting insider threats using the novelties of the current work. To complete the ecosystem and improve knowledge and response, the non-technical components of insider threats must be integrated with the technological ones. Nevertheless, anomaly detection, network flows, and signature-based detection are better suited for this method.

Meng et al. [26] developed an intrusion detection system based on behavioural profiling and hierarchical trust. By evaluating any variation in Euclidean distance among two behavioural profiles, one can ascertain the dependability of MSN nodes. Through an operational healthcare facility partnership, we investigated our mechanism's performance in an actual MSN setting. However, this approach is not effective in determining a trust level in many network scenarios.

Rabbani et al. [27] introduce an approach to enhance cloud service providers capacity to simulate user behaviour. For the purpose of detection and recognition, utilized a probabilistic neural network based on particle swarm optimization. Subsequently, in the first module of the recognition process, converted the user behaviour into an understandable format. Following this used a multi-layer neural network to classify and recognize the harmful behaviour. However, huge datasets are not suited for this strategy.

Asha S et al., [28] developed Nearmiss2 (NM-2) and SVM classifier for solving data imbalancing problems and malicious threat detection in cloud. Developed model consist of two-layer architecture. First layer was used for data integration, transformation and sampling. Following that in second layer anomaly detection using SVM classifier was performed. Developed model attained 82.46% accuracy and 78.72% f-score along with reasonable false detection rate.

Muhammad Mehmood et al., [29] also developed ensembles of machine learning algorithm for escalation attack detection and solving in cloud. Four ML models such as XGBoost, Adaboost, LightGBM, and Random Forest (RF) were developed for internal attack detection. Accuracy achieved using different model included XGBoost at 88.27%, AdaBoost at 88% and RF at 86%.

Numerous systems are developed for optimal insider threat intrusion in cloud environments, according to the literature. According to the articles listed above, insider threat intrusion presents a number of important difficulties. Unsuitable for unbalanced and large datasets [18], [19], [22], [27], cost is high [20], system quality lacking [21], [26], design is intricate [23] and needs to improve [24], [25]. Thus, the proposed method relies on for effective insider threat detection in a cloud computing scenario.

3. Proposed Methodology

Insider threats provide a serious danger to cyber security. Internal attackers have more power and authorized access to information and resources than external attackers, which can cause an organization great harm. Because most businesses don't have a thorough understanding of insider behaviour patterns, they often struggle to identify insider threats when using standard cyber security tactics like intrusion detection systems. A complex procedure is needed to gain a comprehensive grasp of the insider's activities within the firm. Based on insider behaviour, this study offers a fresh conceptual method for insider threat detection. To further enhance the insider threat detector, neural network research will be done. With this method, the optimal insider behaviour pattern will be identified. The process flow of the proposed model is shown in Figure 1.

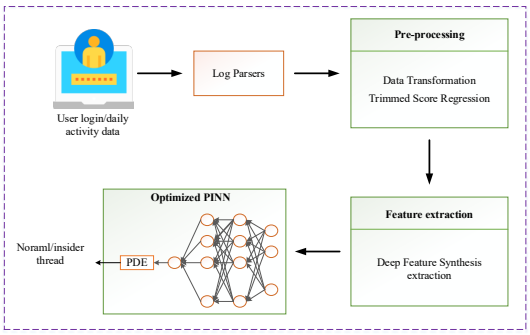

Figure 1. Architecture of proposed model.

Data are gathered from different sources such as network logs, which include email, LDAP, firewalls, VPNs, and proxies. After that, semi-structured log data is converted from gathered data into organized information via log parsing. Collected datasets are identified and missing data is replaced by using Trimmed Scores Regression (TSR). To fill in the missing data in the TSR, based on a regression equation and trimmed score matrix. Following cleaning and pre-processing of the dataset, deep feature Latent patterns which could aid the algorithm's learning are extracted by synthesis-based feature extraction and used as independent variables. It builds prediction models from initial data by automatically creating features for related datasets according to correlations in the data set to a base variable. Following that, an enhanced PINN-based insider threat detection framework analysing user behaviour and entity insights was implemented. It combines Partial Differential Equation (PDE) constraints with observational data in a smooth manner. The suggested methodology is explained in detail in the section below.

3.1. Dataset collection

The Connect, HTTP, Email, and Login databases are the main sources of user data. The background information includes the user's employment status and motivation inside the organization. It offers helpful information about typical or unusual user behaviour, usually real data that need to be updated on a regular basis, taking into account that user behaviour can change at any time.

3.2. Log Parsers

Transforming semi-structured logs into organized ones is the goal of log parsing. Attributes like event level, event template, timestamp, and dynamic variables are frequently seen in structured log data. For anomaly identification, the timestamp, the event template, and the dynamic variables are frequently utilized. The portion of the log that the log parser recognizes as static texts is known as the event template. The portion that the log parser recognizes as the variable is known as the dynamic variable.

3.3. Data pre-processing

Non-continuous or data-void records are actually deleted at this step. All features are correctly identified, and the best feature is chosen based on pertinent data. The system's "garbage in, garbage out" procedure is actually improved by it. Careful data analysis aids in the removal of inaccurate findings. Enhancing the data quality prior to conducting any analysis is beneficial. By using pre-processing, it is possible to eliminate incorrect findings that arise from datasets with missing values. Pre-processing based on trimmed score regression is used in the proposed model, improving accuracy.

Data Transformation

For additional processing, the integrated data must be converted to a category value. The integrated data's "pc," "vector," "user," and "activity" properties were transformed into a numerical number.

Trimmed Score Regression (TSR)

First, the column mean is used to assign the missing data in a matrix X_0; the imputed matrix is designated as X. Centralized column matrix X is represented by $Z = (Z_{ij})_{n\times m} \in R^{n\times m}$. The matrix Z had an i[th] row vector of $Z_i = (Z_{i1}, Z_{i2}, Z_{i3}, \dots, Z_{im}) \in R^{1\times m}$. Let us consider the matrix X_0 at the i[th] row of d_i missing value. The missing value is divided into two group vector $Z_i \overset{Q}{\rightarrow} (Z_i^{(M)}, Z_i^{(O)})$ at a projection matrix Q. Here the superscripts (o) and (M) denote known data and missing data correspondingly [30]. The vector $Z^{(i)} = (Z^{(i)(M)}, Z^{(i)(O)})$ can get the term $(Z_i^{(M)}, Z_i^{(O)})$. The performing Principal component analysis by Singular value decomposition on a matrix $Z^{(i)}$have,

$$Z^{(i)} = B^{(i)} D^{(i)} C^{(i)^T} \quad for\ i = 1,2,\dots,n \qquad (1)$$

Where $B^{(i)}$ matrix is termed as $n \times r_{Z^{(i)}}$, which is left singular vectors $(r_{Z^{(i)}} = rank(Z^{(i)}) \geq r)$; the term $C^{(i)}$ matrix is denoted as $m \times r_Z(i)$, which is right singular vectors; the $D^{(i)}$ matrix is denoted as diagonal $r_{Z^{(i)}} \times r_{Z^{(i)}}$ and the values of this matrix is arranged in descending order. $B_r^{(i)}, D_r^{(i)}$ and $C_r^{(i)}$ are the portion of matrices $B^{(i)}, C^{(i)}$ and $D^{(i)}$ correspondingly that are correspond to r components. The $F^{(i)}$ matrix is $n \times r$ score component, the $A^{(i)}$ matrix is $m \times r$ loading component. Based on $Z^{(i)}$ in (2), the $A^{(i)}$ is stated as,

$$A^{(i)} = C_r^{(i)} = \begin{pmatrix} A^{(i)(M)} \\ A^{(i)(O)} \end{pmatrix} \qquad (2)$$

The $F^{(i)}$ matrix is stated as,

$$F^{(i)} = Z^{(i)} A^{(i)} \qquad (3)$$

Thus,

$$F^{(i)(O)} = Z^{(i)(O)} A^{(i)(O)} \quad (4)$$

Where, $F^{(i)(O)}$ denote trimmed score matrix and make the subsequent regression equation,

$$Z^{(i)(M)} = F^{(i)(O)} W^{(i)} + G^{(i)} \quad (5)$$

Where, $G^{(i)}$ signifies $r \times d_i$ error matrix and $W^{(i)}$ denote $n \times r$ coefficient matrix. The estimating matrix $\hat{Z}^{(i)(M)}$ and vector $\hat{Z}_i^{(M)}$ are stated as,

$$\hat{Z}^{(i)(M)} = F^{(i)(O)} \left(F^{(i)(O)^T} F^{(i)(O)} \right)^{-1} F^{(i)(O)^T} Z^{(i)(M)} \quad (6)$$

$$\hat{Z}_i^{(M)} = f_i^{(O)} \left(F^{(i)(O)^T} F^{(i)(O)} \right)^{-1} F^{(i)(O)^T} Z^{(i)(M)} \quad (7)$$

Where $f_i^{(O)}$ represent ith row of $F^{(i)(O)}$. The vector $\hat{Z}_i$ is expressed as,

$$\hat{Z}_i \overset{Q^T}{\leftarrow} \left(\hat{Z}_i^{(M)}, Z_i^{(O)} \right), \hat{Z} = \left(\hat{Z}_1^T, \hat{Z}_2^T, \ldots \hat{Z}_3^T \right)^T \quad (8)$$

Consequently, the matrix $\hat{X}_0$ is termed as,

$$\hat{X}_0 = \hat{Z} + 1_{n \times 1} \bar{X} \quad (9)$$

Here the vector $\bar{X} = (\bar{X}_{.1}, \bar{X}_{.2}, \ldots, \bar{X}_{.m})$ and $\bar{X}_j (j = 1,2, \ldots, m)$ are the mean of the matrix X_0. The estimator matrix $\hat{X}_0$ is obtained by equation (9). The ith row of a matrix $\hat{X}_0$ is signified by $\hat{X}_{0i}$.

3.4. Feature extraction

During the feature engineering process, one of the primary issues with identifying insider threats is the extraction of features. The amount of characteristics extracted from each log source is not regulated and has been seen to vary among experiments. Every researcher uses Deep Feature Synthesis (DFS) to manually synthesize as many features as they can in order to obtain high-quality descriptive results. This program does feature design for multi-table and transaction datasets that are frequently seen in databases or logs. It collects features that are typically supported by human intuition. Following the DFS procedure, the information is collected so that each user-related event is depicted in a distinct feature vector that is based on user action sequences. Since many DFS attributes are categories and thus unreadable by the ML techniques that are using, the aggregated data must be structured properly. The most important linear combinations of the initial set of features are reduced to a smaller set of independent features [31].

Deep Feature Synthesis (DFS)

The input features that machine learning algorithms use are very important. In many situations, a machine learning algorithm requires the use of human intuition in order to make an appropriate feature selection. However, recent developments in deep learning algorithms can do away with the requirement to choose a suitable feature since the feature is learned by the network design. Still, other machine learning algorithms require human judgment and require a lengthy feature selection process. An automatic feature engineering tool was employed in this work to increase the efficacy of insider threat prediction techniques. In addition to the carefully selected variables from previous research, the DFS tool takes into account a huge variety of additional features. By automatically generating features for related datasets according to links in the data to a base field, it creates predictive models from raw data. The final features were then constructed by applying mathematical functions in a systematic manner along that path. Figure 2 explains the process of the DFS feature extraction process.

From the provided data, the algorithm creates certain specialized features by figuring out what could forecast the result. Even though the method is autonomous, it captures features that are typically enhanced by human interpretation. Due to their foundation in simple, naturally comprehensible combinations of primitives, DFS characteristics are more easily understood. The inputs of this method consist of a collection of relationships, many mathematical functions, and entity sets containing various data types, including timestamps, numerics, and categories. The mathematical processes are applied at two levels: the relational level and the entity level. DFS generates three distinct feature types: relational features (rfeat), entity features (efeat), and direct features (dfeat). There is just one entity considered while computing entity features (efeat). Rounding a number, converting a categorical string data type to predefined numeric data, and translating a previous feature into a different value type are a few examples of this technique.

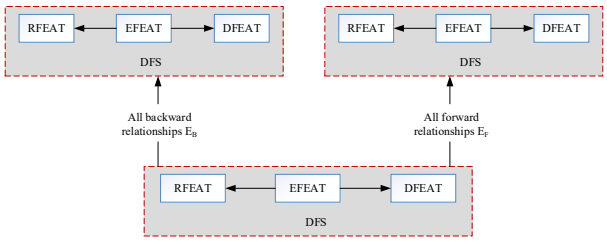

Figure 2. Structure of DFS process.

In order to create this feature, DFS first creates a function of $x_{:,j}$, after creates a new feature by evaluating the cumulative density function of $x_{i,j}$. When one case of instance m and entity E_I have a forward relationship. The forward relationship is covered by direct features. These characteristics are passed over directly as attributes for the $m \in E_I$ from the linked entity $i \in E_k$. In the case of backward relationships, all cases $m = 1 \ldots M$ in E_I which had a forward association with k are connected to instance i in E_k. As soon

as the decision tree or another structure responds to determine deep DFS activation.

3.5. Classification

Neural networks called PINNs are taught to solve broad nonlinear supervised learning tasks or partial differential equations while adhering to a set of physical laws. As a result, they can be included into intricate physical systems and offer substitute models that organically convey the underlying physical rules of the system. A set of PDE is known as the Navier-Stokes equations can be approximated by PINNs through training. The input, which consists of observed values of infected data, is first weighed and biased by the first layer of perceptrons. After that, the subsequent layer will use those inputs to make increasingly complicated judgments. This procedure is carried out up until the last decision layer, at which point the layer generates outputs that might or might not coincide with the parameter values of the analytical model, such as the recovery rates and transmission. Using the computed weights (w) and bias (b), develop a PINN in this work that makes decisions according to suitable activation functions. Using techniques similar to gradient descent, the network then seeks to decrease the regression's MSE taking into account the biases and weights [32]. Spatial (x) and temporal (t) coordinates are the input parameters, and the normal/insider thread (v) is the output. The loss function is then computed using the network's outputs, the physical law, and the starting conditions. Figure 3 demonstrates the architecture of the proposed PINN model.

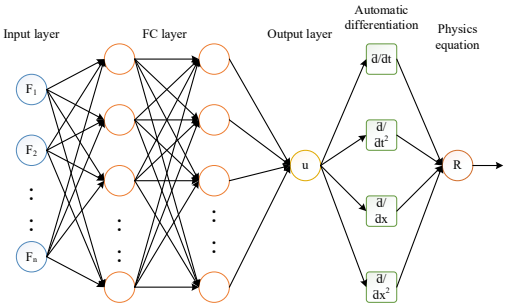

Figure 3. Structure of proposed model.

The PINN's input functions are expressed as,

$$\frac{\partial u}{\partial t} = -N[u,\lambda], x\epsilon\Omega, t\epsilon[0,T] \quad (10)$$

where, $N[u,\lambda]$ signifies a nonlinear operator relating the state parameter u at a system variable λ and $u(t,x)$ denote the solution set. Time is represented by the term t, and system input is by x. Based on past understanding of the dynamical system, the domain Ω can be bounded, and the time period $[0,T]$ represents the system's evolution. λ one of the model parameters, may be fixed or undetermined. When λ is unknown, the function approximation problem transforms into a system identification problem. Define the PINN $f(t,x)$ in order to impose the physical law defining the dynamical system.

$$f(t,x) = \frac{\partial u}{\partial t} + N[u,\lambda] \quad (11)$$

Where, the parameter λ is denoted as nonlinear operator $N[u,\lambda]$ simplifies to $N[u]$. Using t and x as inputs, a neural network is utilized to forecast $u(t,x)$. Automatic distinction between the neural network's constituent elements that forecast $u(t,x)$ yields $f(t,x)$. The necessary derivatives of $u(t,x)$ at time t and input x are therefore determined by this. Because of this, the neural network that predicts $f(t,x)$ and $u(t,x)$ differ in their activation functions yet share the same parameters. By reducing the loss function, the two neural networks' shared variables are optimized.

$$MSE = MSE_u + MSE_f \quad (12)$$

$$MSE_u = \frac{1}{N_u}\sum_i^{N_u}\left|u(t_u^i, x_u^i)\right|^2 \quad (13)$$

$$MSE_f = \frac{1}{N_f}\sum_i^{N_f}\left|f(t_f^i, x_f^i)\right|^2 \quad (14)$$

where MSE_u denotes loss corresponding to initial data, N_u represents the number of training data, N_f represents the number of collocation points, and MSE_f denotes a limited set of collocation values. The quantity of training data and collocation points affects both the computing time required to improve the loss function and the quality of the predictions. The mistake enforces the physics for the dynamical process that the condition requires MSE_f, which penalizes departures from the expected physical law, and the border situations of an liberated parameter x are enforced by the error MSE_u. To ascertain which neural network variables decline given a batch of training data and given system variables λ. If the parameters λ are unknown, we use the system parameters as supplementary variables and train for the same purpose.

Hunter-prey Optimization

A novel intelligent optimization technique called Hunter-Prey Optimization (HPO) simulates how hunters might hunt. Through illuminating the fascinating nature-prey connection, nature can assist in problem solving. This algorithm creates the illusion of an animal spotting prey by picking it apart from the herd, which frequently congregates in large numbers. While the hunter pursues and kills the target prey, the prey has the opportunity to search for a nutrient and escape the dangerous location in case the hunter approaches. To tackle this issue, a meta-heuristic method named HPO is created, which consists of initialization, exploration, and exploration phases. A straightforward meta-heuristic approach called HPO can assist in resolving natural optimization issues [33].

Step 1: Initialization

The proposed approach considers learning rate and weights as input and provides an initialization procedure in Eqn. (15) and (16).

$$F = (F1, F2, \ldots Fn) \quad (15)$$

$$L = (L1, L2, \ldots Ln) \quad (16)$$

Where F represents weights and L denote learning rate.

Step 2: Fitness function

The fitness function's ultimate objective is to maximize accuracy for optimal feature selection, which is calculated using the equation (18).

$$Fintness\ (F_i) = \max(accuracy) \quad (17)$$

$$Accuracy = 1 - error \quad (18)$$

Where, $error$ denote the value of each population set.

Step 3: Updation

The values are adjusted each time iteration to achieve the most optimal value. Updation function of HRO is given in equation (19).

$$hp_{ij}(k+1) = hp_{ij}(k) + 0.5\{[2B_pA_pP_{pr(j)} - hp_{ij}(k)] + [2(1-B_p)A_p\gamma_j - hp_{ij}(k)]\} \quad (19)$$

Where, $hp_{ij}(k+1)$ was the modified position of hunter for the next iteration, $P_{pr(j)}$ represents the position of prey, γ_j denotes the mean of all prey positions, A_p represents the adaptive parameter, B_p was the parameter used for balancing the exploration and exploitation phases.

Step 4: Termination

Once the optimal solution is obtained, the process is terminated, and the best combination is set as the PINN network.

4. Result and Discussion

This division describes the construction of a PINN with TSR imputation model to detect insider threads in a cloud system using login data. Malevolent insiders often carry out their schemes bit by bit, taking care to hide their activities and avoid being discovered. Because of this, it is very challenging to identify and eliminate these risks. Thus, in the proposed work, login credentials are collected from an organization to detect insider threads and alert PINN users to them. The recommended model was implemented using 64GB of RAM, an Intel Core i7 CPU, an NVIDIA GeForce RTX 3070 GPU, and Python software. The dataset used to analyse the proposed model was the main topic of the discussion that followed.

4.1. Dataset description

A group of test datasets for hostile actors and fictitious background information, which are used to simulate insider risks. A set of simulated insider threat test datasets was developed by the CERT Division in partnership with ExactData, LLC and DARPA I2O. These datasets provide synthetic information as well as data from dangerous agents that are purposefully created. The datasets are arranged based on the release date of the data source. A lot of datasets are included in most versions. Subsequent versions typically include a subset of the data production features found in previous versions. Each dataset file is accompanied by a readme file with comprehensive commentary regarding the characteristics of that release [34].

This dataset will be filled in by applying the TSR missing value imputation technique. After that, each filling data is extracted independently using DFS. After the data is retrieved, a PINN classifier is fed with it to predict the insider thread in a cloud system. The performance of the suggested model was discussed as follows,

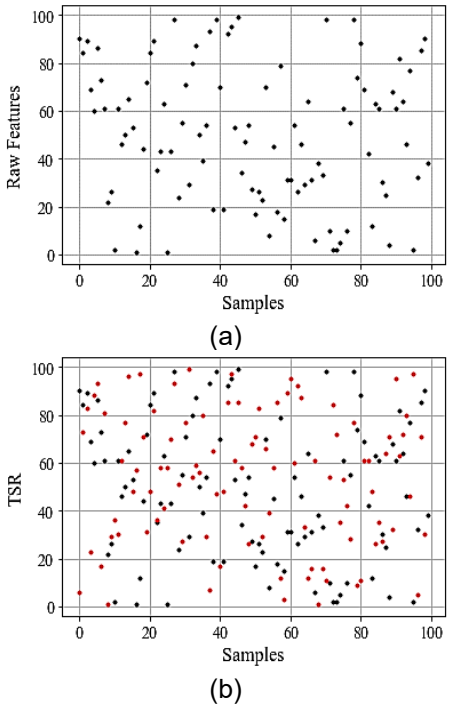

Figure 4. Evaluation of (a) raw dataset (b) After missing value replaced dataset.

The raw data before to filling in the missing values is evidently shown in Figure 4(a). Data following the replacement of missing values is displayed in Figure 4(b). The observed plot illustrates the suggested procedure for replacing missing values to increase the classifier's accuracy.

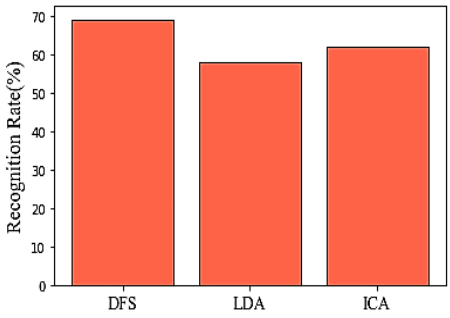

Figure 5. Evaluation of feature extraction process.

In figure 5, Independent component analysis (ICA), linear discriminant analysis (LDA), and TSR were combined with PINN to produce the desired findings. The TSR, LDA, and ICA algorithms overall performance analysis were computed using the recognition rate as a measure. It demonstrates that the ICA provides 62%, LDA provides 59%, and TSR provides 68% recognition rate. The PINN classifier combined with TSR yields the greatest results when compared to LDA, and ICA.

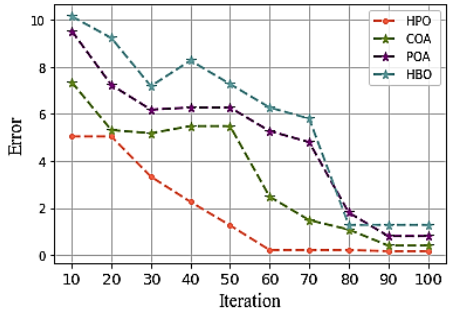

Figure 6. Convergence curve of HPO.

The HPO's convergence curve was shown in Figure 6 with a comparison of the existing model. The available coati optimization algorithm (COA) provides a solution with an error of 0.4, the pelican optimization algorithm (POA) provides an error of 1, and the honey badger optimization (HBO) provides an error of 1.4, based on the available data. By comparison, a better solution with a low error of 0.2 is provided by the proposed HPO. Thus, it can be demonstrated that in order to select the optimal PINN parameter solution, the proposed model reduced overall error values.

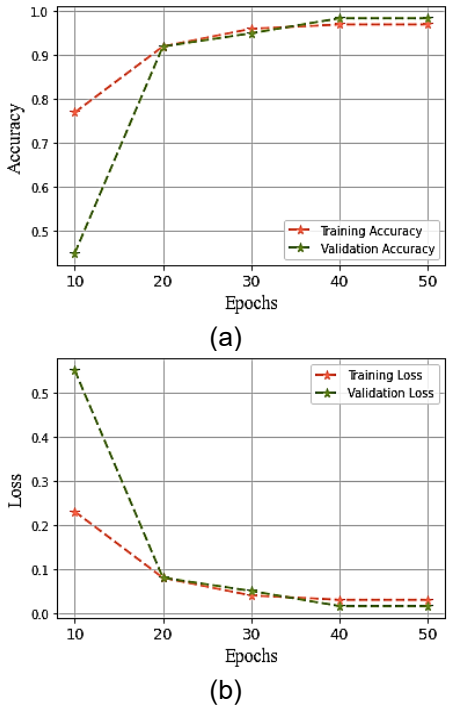

Figure 7. Evaluation of PINN's training and validation (a) accuracy (b) Loss.

A comparison of accuracy over multiple training epochs is displayed in Figure 7(a). This proposed classifier has an accuracy value of 99% since its epoch is 50. To raise the recommended model's training accuracy, enhance the epoch value. Additionally, a comparison of accuracy throughout multiple validation epochs was noted. This proposed classifier produces a 98.4% accuracy after 50 epochs. Figure 7(b) displays the training loss, which is calculated by deducting the accuracy value from 100. The suggested JNN offers superior training performance, as shown by the numbers for accuracy and loss during the training period. Based on study and observation, the validation loss for the validation period at epoch 50 is found to be 1.6%. The suggested PINN offers a better validation performance, as evidenced by both accuracy and loss in validation period data.

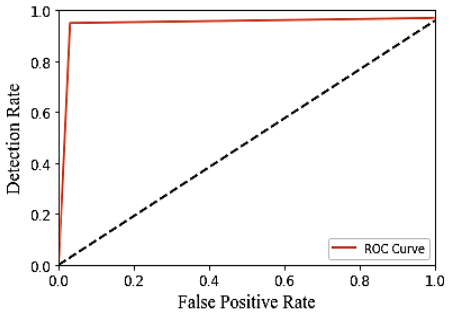

Figure 8. ROC of proposed model.

Figure 8 demonstrates the ROC characteristics of the proposed model. The probability curve compares the

detection rate at various threshold levels against the false positive rate. Consequently, it was demonstrated that in terms of insider thread and normal separation, the recommended technique performs better than others.

4.2. Performance Evaluation

The suggested PINN is contrasted with traditional prediction such as Gated Recurrent Unit (GRU), probabilistic neural network (PNN), and Multi-layer Perceptron classifier (MLP). Table 1 demonstrates the parameters with its ranges of proposed PINN and existing approaches.

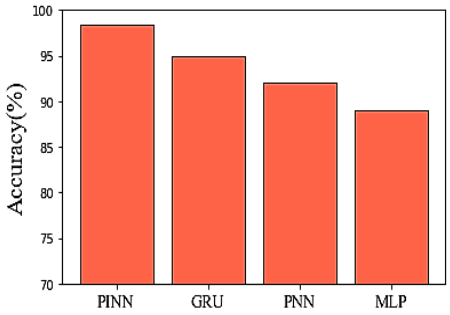

Figure 9. Evaluation of accuracy.

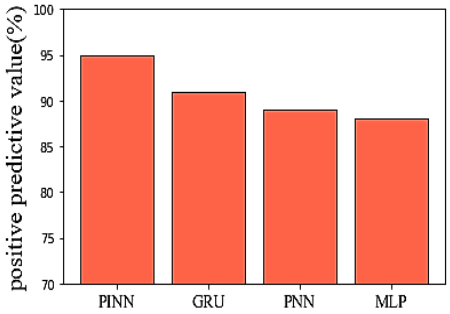

Figure 10. Evaluation of PPV.

The accuracy values that were determined using the confusion matrix are displayed in Figure 9. An accurate system is one that predicts a value with the least amount of error. The accuracy rate of the proposed method is 0.984, compared to 0.89 for MLP, 0.95 for GRU, and 0.92 for PNN. Figure 10 displays the positive prediction value (PPV) of the recommended and current approaches. What is meant to be correctly measured is the quantity of expected favorable events. In comparison to other existing algorithms such as GRU, PNN, and MLP, which had equal accuracy values of 0.91, 0.89, and 0.88, the PPV value of the suggested method was determined to be 0.95.

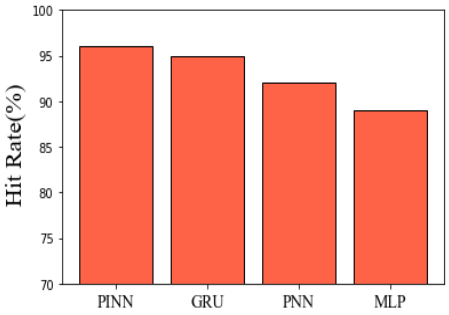

Figure 11. Evaluation of Hit rate.

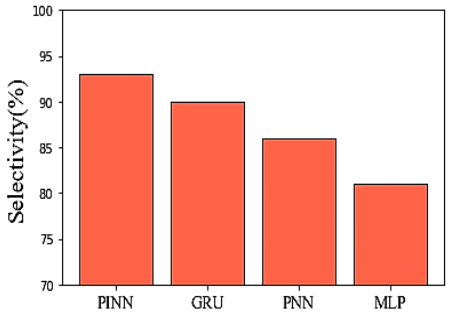

Figure 12. Evaluation of selectivity.

As shown in Figure 11, the hit rate is computed by dividing the total amount of components that are classified as positive by the total number of true positives. The proposed method's Hit rate value was found to be 0.96 when compared to other current approaches, namely GRU, PNN, and MLP, which had corresponding Hit rate values of 0.95, 0.92, and 0.89. A comparison of the selectivity of suggested and existing methods is shown in Figure 12. The specificity value of the suggested method was found to be 0.93 when compared to other existing algorithms, such as GRU, PNN, and MLP, which have respective specificity values of 0.9, 0.86, and 0.81.

Figure 13. Evaluation of NPV.

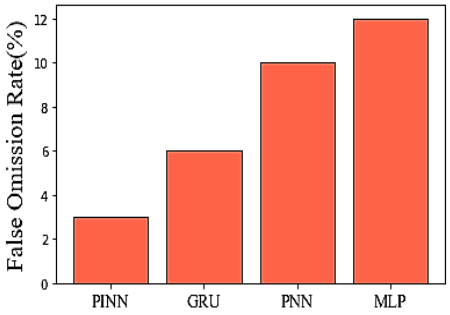

Figure 14. Evaluation of FOR.

Subsequently, the NPV of the proposed and current approaches is evaluated. The ratio of true negative predictions when taking into account all negative predictions is measured by the Negative Predictive Value, or NPV. Figure 13 displays the NPV of the suggested approach, which is 0.97, GRU of 0.94, PNN of 0.9, and MLP of 0.88. The percentage of erroneous negative activities across all transactions that had a negative consequence is shown in Figure 14 by the false Omission Rate (FOR) difference. The FOR, GRU, PNN, and MLP of the suggested approach are 0.03, 0.06, 0.1, and 0.12. It claims that compared to traditional methods, the suggested model produces better results.

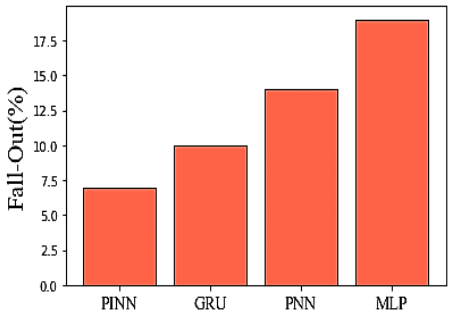

Figure 15. Evaluation of Fall-out.

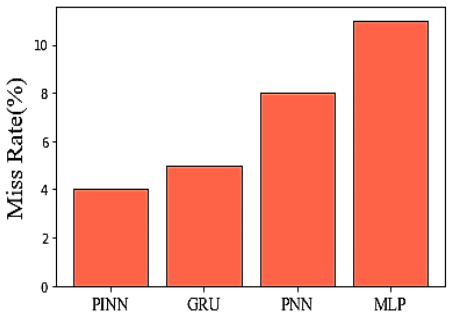

Figure 16. Evaluation of Miss Rate.

Figure 15 compares the fall-out of the recommended and present practices. The fallout approach is used to calculate the quantitative probability that any positive or negative test result will result in a positive result because of a fault. Compared to the 10% from the present GRU, the 14.2% from PNN, and the 19% from MLP, the fallout from the suggested approach is 7.4%. This shows that the overall fall-out value of the suggested model is lower than that of the current methods. The Fall-Out for the proposed and current methodologies is then looked at. The Miss rate is shown in Figure 16 for both the present and future strategies. The Miss rate of the suggested approach is 0.4, GRU is 0.5, PNN is 0.8, and MLP is 0.11. It shows that the recommended methodology produces better miss rate outcomes than traditional models.

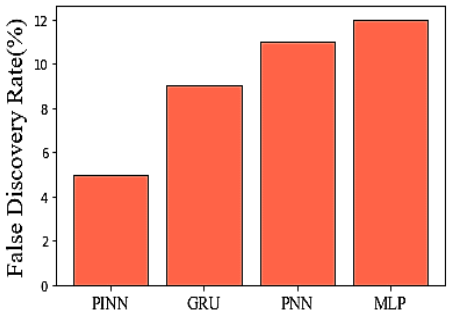

Figure 17. Evaluation of FDR.

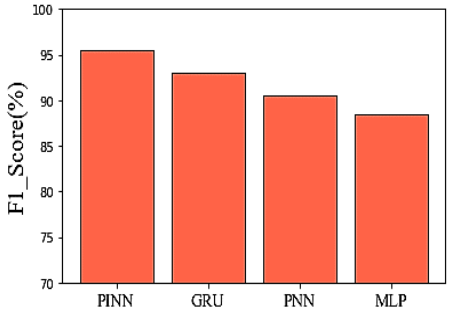

Figure 18. Evaluation of F1_score.

The application of the False Discovery Rate (FDR) to identify as many meaningful characteristics as possible while minimizing false positives is seen in Figure 17. FDR is 0.5, GRU is 0.9, PNN is 0.11, and MLP is 0.19 for the suggested approach. This demonstrates that the suggested model produces superior results than the conventional models. Next, examine the F1_Scores of the suggested and used processes. A statistical study of the F1_score reveals the binary types of the system and the level of accuracy of the data set. A contrast of the actual and predicted F1 scores is presented in Figure 18. The suggested method has an F1 Score of 0.95, while GRU, PNN, and CNN have scores of 0.95, 0.92, and 0.9. It proves that the proposed model produces better outcomes than other approaches.

1.1. Computation time comparison

The total computing time of suggested and existing methods was covered in this subsection. The amount of time needed to complete a computing process is known as the computation time. Training time, testing time, and total

computation time were reported as the compute performance analysis's output.

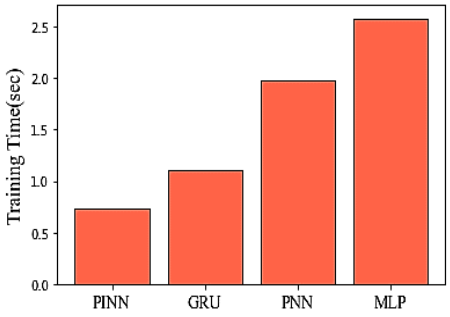

Figure 19. Evaluation of training time.

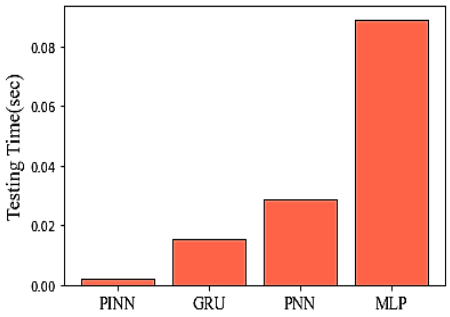

Figure 20. Evaluation of testing time.

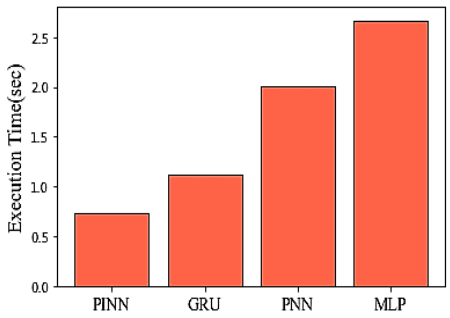

Figure 21. Evaluation of testing time.

The training period of the proposed classifier is displayed in Figure 19. While other existing techniques like GRU take 1.1 sec, PNN take 2.0 sec, and MLP take 2.51 sec for training, the suggested classifier only takes 0.7 s. The testing times for the existing and suggested approaches are displayed in Figure 20. The suggested method takes 0.0005 sec to test the classifier, which is less time compared to other existing methods like GRU, PNN, and MLP, which have testing times of 0.012 sec, 0.03 sec, and 0.09 sec, respectively. The execution times for the recommended and traditional methods are displayed in Figure 21. The suggested method's execution time of 0.7005 sec is less than that of other existing systems, such as GRU, PNN, and MLP, whose respective execution times are 2.6 sec, 2.03 sec, and 1.112 sec. This demonstrates that the proposed framework outperforms other conventional algorithms.

4.3. Comparative analysis

In this section, the overall comparative analysis of proposed and traditional models are examined. For validating the process of the proposed model a comparison was made to traditional approaches including convolutional neural network (CNN), deep neural network (DNN), long short-term memory (LSTM), Encoded rules IDS (E-IDS), and random forest (RF) [35].

Table 1. Comparative analysis of performance metrics.

Methods	Detection rate	Accuracy	F1_score	Hit Rate	Fall-out
Proposed	0.68	0.984	0.95	0.96	0.74
CNN	0.993	0.981	0.973	0.98	0.004
DNN	0.991	0.976	0.967	0.974	0.007
EIDS	0.789	0.801	0.654	0.701	0.107
RF	0.874	0.912	0.869	0.892	0.061
LSTM	0.985	0.971	0.957	0.962	0.008

An overall comparison of proposed and traditional approaches was made to validate the working process. Table 1 demonstrates the comparison of performance metrics, presenting the traditional approaches. The comparative analysis shows that the proposed model provides a better detection rate and accuracy for the prediction process.

5. Conclusion

Optimized PINN with TSR missing imputation was developed to detect insider threats in cloud computing. Insider threats are security risks that originate from within an organization, while outsider threats are risks that come from outside the organization. Malevolent insiders frequently execute their plans gradually, taking precautions to conceal their actions and evade detection. This makes it especially challenging to identify and stop these kinds of threats. Thus, in the proposed work, an organization activity data-based insider threat detection approach was developed using the PINN detection process. An organized login activity was gathered to make a dataset, which is pre-processed to improve the prediction process. The quality of data was improved through data transformation and the TSR approach. When data must be changed to match the target system's employing data transformation, and TSR model is utilized to fill the missing data in the dataset. The pre-data was further

processed for the feature extraction process using DFS extraction. After that using an optimized PINN model to predict the insider thread detection based on the extraction data. With respect to any given set of physics rules specified by general nonlinear PDE, PINN is utilized to perform supervised learning tasks, using the HPO algorithm to choose the suitable value of learning rate and weight of PINN to improve the prediction performance. The proposed model yields a 68% detection rate, 96% hit rate, 98.4% accuracy, 5% FDR, 74% fallout, 95% F1_score and 0.7005 sec execution time. This model offers a better outcome than the traditional approaches, yet with impacts which require a large number of training points, expensive computational costs, and heavy memory overheads. As a result, in future work, a novel deep learning model with advanced techniques will be introduced to overcome these impacts.

Acknowledgements.

Funding: The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.
Conflict of Interest: The authors declared that they have no conflicts of interest within this work. We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.
Availability of data and material: Not applicable
Code availability: Not applicable
Author contributions: The author claims the major contribution of the paper, including formulation, analysis and editing, providing guidance to verify the analysis result and manuscript editing.
Compliance with ethical standards: This article is a completely original work of its authors; it has not been published before and will not be sent to other publications until the journal's editorial board decides not to accept it for publication.

Reference

[1] Yuan Yuan S, Wu X. Deep learning for insider threat detection: Review, challenges and opportunities. Computers & Security. 2021;104:102221.
[2] Jeong M, Zo H. Preventing insider threats to enhance organizational security: The role of opportunity-reducing techniques. Telematics and Informatics. 2021;63:101670.
[3] Yu K, Tan L, Mumtaz S, Al-Rubaye S, Al-Dulaimi A, Bashir AK, Khan FA. Securing critical infrastructures: deep-learning-based threat detection in IIoT. IEEE Communications Magazine. 2021;59(10):76-82.
[4] Robayo TA. The Enemy Within: A Framework for Understanding the Lifecycle of the Malicious Insider Threat to Information Systems (Doctoral dissertation, Saint Leo University).
[5] Saxena N, Hayes E, Bertino E, Ojo P, Choo KK, Burnap P. Impact and key challenges of insider threats on organizations and critical businesses. Electronics. 2020;9(9):1460.
[6] Al-Shehari T, Alsowail RA. An insider data leakage detection using one-hot encoding, synthetic minority oversampling and machine learning techniques. Entropy. 2021;23(10):1258.
[7] Bao Y, Hilary G, Ke B. Artificial intelligence and fraud detection. Innovative Technology at the Interface of Finance and Operations: Volume I. 2022:223-47.
[8] Wei Y, Chow KP, Yiu SM. Insider threat prediction based on unsupervised anomaly detection scheme for proactive forensic investigation. Forensic Science International: Digital Investigation. 2021;38:301126.
[9] Chowdhury M, Ray B, Chowdhury S, Rajasegarar S. A novel insider attack and machine learning based detection for the internet of things. ACM Transactions on Internet of Things. 2021;2(4):1-23.
[10] Williams AD, Abbott SN, Shoman N, Charlton WS. Results from invoking artificial neural networks to measure insider threat detection & mitigation. Digital Threats: Research and Practice (DTRAP). 2021;3(1):1-20.
[11] Feng W, Wu Y, Fan Y. A new method for the prediction of network security situations based on recurrent neural network with gated recurrent unit. International Journal of Intelligent Computing and Cybernetics. 2020;13(1):25-39.
[12] Bu SJ, Cho SB. A convolutional neural-based learning classifier system for detecting database intrusion via insider attack. Information Sciences. 2020;512:123-36.
[13] Al-Mhiqani MN, Ahmed R, Abidin ZZ, Isnin SN. An integrated imbalanced learning and deep neural network model for insider threat detection. International Journal of Advanced Computer Science and Applications. 2021;12(1).
[14] Zeng, Y., Kang, Z., & Shi, Z. (2023). Secure data processing technology of distribution network opgw line with edge computing. EAI Endorsed Transactions on Scalable Information Systems, 10(3), e7-e7.
[15] Ahmed, S. H., & Aljuboori, A. F. (2023). Big Data Detection Utilizing Cloud Networks with Video Vision Techniques. EAI Endorsed Transactions on Scalable Information Systems, 10(5).
[16] Hong, W., Yin, J., You, M., Wang, H., Cao, J., Li, J., ... & Man, C. (2023). A graph empowered insider threat detection framework based on daily activities. ISA transactions, 141, 84-92.
[17] Yin, J., Tang, M., Cao, J., You, M., Wang, H., & Alazab, M. (2022). Knowledge-driven cybersecurity intelligence: software vulnerability coexploitation behavior discovery. IEEE transactions on industrial informatics, 19(4), 5593-5601.
[18] Al-Mhiqani MN, Ahmad R, Abidin ZZ, Abdulkareem KH, Mohammed MA, Gupta D, Shankar K. A new intelligent multilayer framework for insider threat detection. Computers & Electrical Engineering. 2022;97:107597.
[19] Anakath AS, Kannadasan R, Joseph NP, Boominathan P, Sreekanth GR. Insider Attack Detection Using Deep Belief Neural Network in Cloud Computing. Computer Systems Science & Engineering. 2022;41(2).
[20] Wang X, Fidge C, Nourbakhsh G, Foo E, Jadidi Z, Li C. Anomaly detection for insider attacks from untrusted intelligent electronic devices in substation automation systems. IEEE Access. 2022;10:6629-49.
[21] Nasir R, Afzal M, Latif R, Iqbal W. Behavioral based insider threat detection using deep learning. IEEE Access. 2021;9:143266-74.
[22] Alsowail RA, Al-Shehari T. A multi-tiered framework for insider threat prevention. Electronics. 2021;10(9):1005.
[23] Al Razib M, Javeed D, Khan MT, Alkanhel R, Muthanna MS. Cyber threats detection in smart environments using SDN-enabled DNN-LSTM hybrid framework. IEEE Access. 2022;10:53015-26.
[24] Sheykhkanloo NM, Hall A. Insider threat detection using supervised machine learning algorithms on an extremely imbalanced dataset. International Journal of Cyber Warfare and Terrorism (IJCWT). 2020;10(2):1-26.
[25] Haq MA, Khan MA, Alshehri M. Insider threat detection based on NLP word embedding and machine learning. Intell. Autom. Soft Comput. 2022;33:619-35.

[26] Meng W, Li W, Wang Y, Au MH. Detecting insider attacks in medical cyber–physical networks based on behavioral profiling. Future Generation Computer Systems. 2020;108:1258-66.
[27] Rabbani M, Wang YL, Khoshkangini R, Jelodar H, Zhao R, Hu P. A hybrid machine learning approach for malicious behaviour detection and recognition in cloud computing. Journal of Network and Computer Applications. 2020;151:102507.
[28] Asha, S., Shanmugapriya, D., & Padmavathi, G. (2023). Malicious insider threat detection using variation of sampling methods for anomaly detection in cloud environment. *Computers and Electrical Engineering*, *105*, 108519.
[29] Mehmood, M., Amin, R., Muslam, M. M. A., Xie, J., & Aldabbas, H. (2023). Privilege escalation attack detection and mitigation in cloud using machine learning. IEEE Access.
[30] Mishra P, Biancolillo A, Roger JM, Marini F, Rutledge DN. New data preprocessing trends based on ensemble of multiple preprocessing techniques. TrAC Trends in Analytical Chemistry. 2020;132:116045.
[31] Maliwat JY, Ylade PA, Regala RC, Cortez DM, Alipio AJ, Mata KE, Blanco MC. An Enhancement of Deep Feature Synthesis Algorithm Using Mean, Median, and Mode Imputation.
[32] Cai S, Wang Z, Wang S, Perdikaris P, Karniadakis GE. Physics-informed neural networks for heat transfer problems. Journal of Heat Transfer. 2021;143(6):060801.
[33] Naruei I, Keynia F, Sabbagh Molahosseini A. Hunter–prey optimization: Algorithm and applications. Soft Computing. 2022;26(3):1279-314.
[34] Dataset 1: https://kilthub.cmu.edu/articles/dataset/Insider_Threat_Test_Dataset/12841247/1
[35] Bouchama F, Kamal M. Enhancing Cyber Threat Detection through Machine Learning-Based Behavioral Modeling of Network Traffic Patterns. International Journal of Business Intelligence and Big Data Analytics. 2021;4(9):1-9.