International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

Data Leakage Detection in Cloud Computing Environment Using Classification Based on Deep Learning Architectures

Dr. Rajashekhargouda C. Patil[1], Ajay Kumar[2], Narmadha T[3], M. Suganthi[4]
Dr Akula VS Siva Rama Rao[5], Dr. Rajesh A [6]

Submitted: 16/08/2022 **Accepted**: 20/11/2022

Abstract: Insider threats are hostile actions that a legitimate employee of a company could commit. For both commercial and governmental enterprises, insider threats pose a significant cybersecurity risk since they have a considerably greater potential to harm an organization's assets than external attacks. The majority of currently utilised insider threat methodologies concentrated on identifying common insider attack scenarios. This research propose novel technique in data leakage detection in cloud computing based on data classification using deep learning architectures. Here the input data has been collected as network data and processed for noise removal, smoothening. The classification has been done based on Generative Regression kernel SVM. The experimental findings have been calculated in terms of RMSE, SNR, F-1 score, recall, accuracy, and precision. The proposed model offers practical approaches to deal with potential bias and class imbalance issues in order to design a system that effectively detects insider data leaking. Proposed technique attained accuracy of 97%, precision of 92%, recall of 67%, F-1 score of 66%, RMSE 62% and SNR of 61%.

Keywords: *cybersecurity, data leakage detection, cloud computing, data classification, deep learning*

1. Introduction

Data leakage is a major problem in the contemporary business environment since it must be protected against unwanted access. The unintentional or deliberate release of confidential organisational information to unapproved parties is known as data leakage. It is crucial to guard against unauthorised users misusing the crucial data. Information about intellectual property rights, patents, functionality, and other relevant facts are essential. This important organisational information has frequently been distributed to stakeholders outside the boundaries of the company. As a result, it is challenging to find the person or entity responsible for the data leak[1]. In the proposed effort, our objective is to pinpoint the responsible user when organisational data has been compromised by an outsider. The Bell-La Padula security model, which offers secure computer system analysis and design, has been applied in the proposed study. Data confidentiality model is the name of this model. The Bell-LaPadula paradigm offers regulated access to classified material and primarily focuses on data confidentiality issues. Bell-LaPadula model is based on the idea of a state machine in a computer system with a set of permissible states. According to the categorization level of object O, clearance level of a subject S is compared to determine whether the subject is approved for the particular access mode [3]. When these are disclosed, the business is no longer protected and is no longer under the organization's control. Businesses are exposed as a result of this unchecked data leaking. The business is in grave danger once this data is no longer within the domain. These days, a single attack on one organisation can affect hundreds of thousands or even millions of individual customers, as well as even more individual records.

Contribution of this research is as follows:

1. To propose novel method in data leakage detection in cloud computing based on data classification utilizing DL architectures
2. Here input data has been collected as network data and processed for noise removal, smoothening. The classification has been done based on Generative Regression kernel support vector machine.

2. Literature Review

Numerous studies have remained concerned with safeguarding data to stop malicious data leakage by employees of the organisation or by insiders authorised by the system. Intrusion detection and prevention systems are common data protection techniques. An automated system known as an intrusion detection system watches network or computer activity to look for intrusions. A security measure known as an intrusion prevention system employs a number of security technologies to block hazardous network traffic while preventing intrusion in real time [5]. Model-based, signal-based, and knowledge-based fault detection methods comprise the majority of non-manual defect detection strategies. Artificial neural networks (ANNs) or other classifiers are frequently used by fault detection systems to improve detection rates. The three basic components of these intelligent systems are data collecting, feature extraction, and data classification. Four key signal processing classes are employed for feature extraction: time-domain, frequency domain, enhanced

[1] Associate Professor, Electronics and Communication Engineering Visvesvaraya Technological University, patilrajuc@gmail.com
2 Assistant Professor, Bharati Vidyapeeth (Deemed to be University) Institute of Management and Research, New Delhi, India ajay.kumar@bharatividyapeeth.edu
3 Assistant Professor, Department of Computer Science and Engineering Jain(Deemed-to-be University), Bangalore, India, naramadhat2021@gmail.com
4 Assistant Professor, Computer Science and Engineering, Thamirabharani Engineering College, Tamil Nadu/India sugi.mp@gmail.com
5 Associate Professor, Dept of CSE, Sasi Institute of Technology & Engineering, Tadepalligudem, shiva.akula@gmail.com
6Professor, Department of CSE, Faculty of Engineering and Technology JAIN (Deemed-to-be University), Karnataka, a.rajesh@jainuniversity.ac.in

frequency, and time-frequency analysis [6]. In the areas of image classification, computer vision, and flaw detection, deep learning techniques excel. A subset of deep neural networks is the convolution neural network (CNN) structure [7]. Additionally, the most widely used method for finding a water leak is to install a pressure transducer on the surface of a pipe or utilise sound sensors [8]. Three general categories can be used to categorise insider threat studies. The first area focuses on creating detecting systems using rules. They are founded on established guidelines for spotting insiders' malevolent behaviour. Rules are established by a panel of specialists, after which all insider activity is recorded and compared to the established rules. [9] discusses the different kinds of insider threats as well as the expertise needed to stop and identify them. The drawbacks of rule-based detection methods include the requirement to update the rules frequently using the expertise of domain experts and the high likelihood that the rules will be evaded [10]. Such a strict technique can therefore produce undesirable detection performance [11]. The second part is concerned with creating a network graph, where the graph's structure is watched for any potential deviations that would indicate harmful actions [12].

1. Proposed Model

This section discuss novel technique in data leakage detection in cloud computing based on data classification using deep learning architectures. Here the input data has been collected as network data and processed for noise removal, smoothening. The classification has been done based on Generative Regression kernel support vector machine. An overview of the model is illustrated in Figure 1.

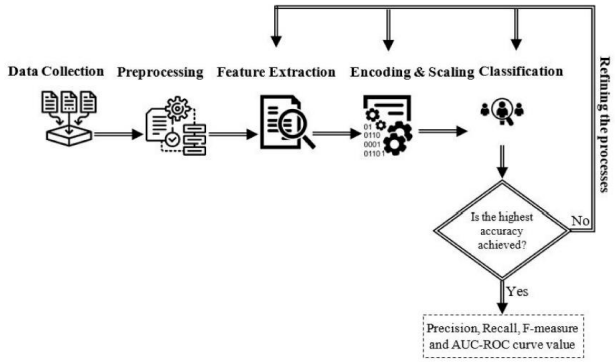

Figure-1 An overview of proposed system

Generative Regression kernel support vector machine:

Global learning model, network parameters at iteration t, and loss function, respectively, are denoted by F (•), t, and L(•). At client node individually, the local gradient дi t is determined using the standard SGD. Equation (2), where C is the number of customers in eq, demonstrates how server aggregates local gradient дi t before updating global model weights t+1 (1).

$$g_t^i = \frac{\partial L\left(F\left(< x_t^i, y_t^i >, \theta_t\right)\right)}{\partial \theta_t}$$

$$\theta_{t+1} = \theta_t - \frac{1}{C}\sum_{i=1}^{c} g_i \quad (1)$$

It outputs the bogus label that corresponds to a latent space vector that has been randomly sampled as input. We assume the input vector's elements are unrelated to one another and are distributed according to a typical Gaussian distribution. Label set in GRNN and the FL system are both identical. Formally, the creation of false picture (x j t) and fake label (y j t) data can be expressed as in Equation (2), where and vt are trainable GRNN parameters and input random vector is drawn from a unit Gaussian distribution.

$$\left(x\hat{}_t^j, y\hat{}_t^j\right) = G(v_t \mid \theta\hat{}_t) \quad (2)$$

$$arg\ min_{\theta\cdot} \parallel g_t^j - g\hat{}_t^f \parallel^2 \Longrightarrow arg\ min_{\theta\cdot} \parallel \frac{\partial L\left(F\left(< x_t^j, y_t^f >, \theta_t\right)\right)}{\partial \theta_t} - \frac{\partial L\left(F\left(< x\hat{}_t^j, y\hat{}_t^j >, \theta_t\right)\right)}{\partial \theta_t} \parallel^2 \quad (3)$$

$$L\hat{}(g, g\hat{}, x\hat{}) = MSE(g, g\hat{}) + WD(g, g\hat{}) + \alpha + TVLoss\ (x\hat{}) \quad (4)$$

$$w^*,\ b^*, \xi_i^* = arg\ min_{w,b}\left(\frac{1}{2} \parallel w \parallel^2 + C\sum_{i=1}^{N}\xi_i\right)$$

subject to $y_i \cdot f\left(\Phi(x_i)\right) \geq 1 - \xi;;\&0;C > 0; w \in R^n; i = 1,2,\cdots,N$ (5)

where the smoothness regularization's weighting parameter, and the MSE loss and WD are both equally weighted. NN that is entirely differentiable and capable of being jointly trained end-to-end is used to parameterize both branches of proposed GRNN. The margin's width in SVM is equal to 2/‖w‖. The following optimization equation (eq.) (5) represents the objective of maximising the margin width as follows:

$$\alpha^* = arg\ max_\alpha L(\alpha) = \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\left[\alpha_i\alpha_j y_i y_j K\left(x_i, x_j\right)\right]$$

(6)subject to $\sum_{i=1}^{N}(y_i\alpha_i) = 0; 0 \leq \alpha_i \leq C; \alpha - \{\alpha_i\}^N; i = 1,2,\cdots,N$

$$\{w^* = \sum_{i=1}^{N}[\alpha_i^* y_i \Phi(x_i)] = \sum_{i=1}^{p}[\alpha_t^* y_i\ b^* = \frac{1}{p}\sum_{i=1}^{p}[y_t - \alpha_t^* y_t \kappa(x_t, x_t)] \quad (7)$$

$$y\hat{} = f(x) = Sign\ \left(\sum_{i=1}^{k}[\alpha_i^* y_i \kappa(x_r, x)] + \frac{1}{p}\sum_{i=1}^{p}\left[y_f - \alpha_i^* y_i \kappa(x_i, x_t)\right]\right) \quad (8)$$

The most crucial element of kernel-based approaches like SVM is the kernel function. The Mercer condition and Hilbert-Schmidt theory are used to generate a number of kernel functions. Below is a list of three frequently used kernel functions: (1) Equation 1's Gaussian radial basis function (9)

$$\kappa\left(x_i, x_j\right) = exp\ \left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right), \sigma \in (0, +\infty) \quad (9)$$

$$\parallel \Phi(x_i) - \Phi\left(x_j\right) \parallel = \kappa(x_i, x_i) + \kappa\left(x_j, x_j\right) - 2\kappa\left(x_i, x_j\right)$$

$$cos\ \theta\left(\Phi(x_i), \Phi\left(x_j\right)\right) = \frac{\kappa(x_i, x_j)}{\sqrt{\kappa(x_i, x_i)\kappa(x_j, x_j)}} \quad (11)$$

We use cosine similarity in the suggested strategy since distance similarity is less effective in high-dimensional kernel space. The equation (eq) defines the cosine similarity matrix (M) in a kernel space (12)

$$M = [K_U' \cdots K_{il}' \vdots \ddots \vdots K_{L\wedge 1}' \cdots K_{uL}'],$$

$$K_{jj}' = \left[cos\ \theta\left(\Phi\left(x_1^{(i)}\right), \Phi\left(x_1^{(j)}\right)\right) \cdots cos\ \theta\left(\Phi\left(x_1^{(i)}\right), \Phi\left(x_{N_f}^{(j)}\right)\right) \vdots \ddots \vdots cos\ \theta\left(\Phi\left(x_{N_j}^{(i)}\right), \Phi\left(x_1^{(j)}\right)\right) \cdots cos\ \theta\left(\Phi\left(x_{N_1}^{(i)}\right), \Phi\left(x_{N_j}^{(j)}\right)\right)\right] \quad (12)$$

In this study, eq is used to estimate W and B. (13)

$$W = -Avg\ ([K_{11}' \quad \ddots \quad K_{UL}']) \quad (13)$$

$$= -\frac{1}{\sum_{i=1}^{L}N_i^2}\sum_{i=1}^{L}\sum_{i=1}^{N}\sum_{k=1}^{N}\frac{\kappa\left(x^{(i)}, x_i^{(i)}\right)}{\sqrt{\kappa\left(x_*^{(i)}, x_t^{(i)}\right)\kappa\left(x_i^{(i)}, x_t^{(i)}\right)}}$$

$$= -\frac{1}{\sum_{i=1}^{L}\sum_{j=1}^{L}N_i N_j}\sum_{i=1}^{L}\sum_{j=1}^{L}\sum_{j=1}^{N}\sum_{i=1}^{N}\frac{\kappa\left(x^{(i)}, x_i^{(j)}\right)}{\sqrt{\kappa(x_{,}^{(i)}, x^{(i)})\kappa\left(x_{,}^{(j)}, x_{,}^{(j)}\right)}}$$

Since Gaussian RBF kernel is twice differentiable, Newton's method could be used to get the ideal s. Equation (14) gives derivatives of W and B:

$$DW(\sigma) = -\frac{1}{\sum_{i=1}^{L}N_i^2}\sum_{i=1}^{L}\sum_{i=1}^{N}\sum_{i=1}^{N}\left[\kappa\left(x_+^{(i)}, x_+^{(i)}\right) \parallel x^{(i)} - x_+^{(i)} \parallel^2/\sigma^3\right].$$

$$D\ B(\sigma) = -\frac{1}{\sum_{i=1}^{L}\sum_{j=1}^{L}N_i N_j}\sum_{i=1}^{b}\sum_{j=1}^{b}\sum_{i=1}^{N}\sum_{k=1}^{N}\left[\kappa\left(x_i^{(i)}, x_+^{(j)}\right) \parallel x^{(i)} - x_+^{(i)} \parallel^2/\sigma^3\right],$$

$$D^2\ W(\sigma) =$$

$$-\frac{1}{\sum_{i=1}^{L} N_i^2}\sum_{i=1}^{L}\sum_{i=1}^{N}\sum_{i=1}^{N}\left[K\left(x_i^{(i)}, x_i^{(i)}\right)\left(\| x_i^{(i)} - x_i^{(i)} \|^4 - 3\sigma^2 \| x^{(i)} - x_i^{(i)} \|^2\right)/\sigma^6\right] \quad (14) D^2\, B(\sigma) = -\frac{1}{\sum_{i=1}^{L}\sum_{j=1}^{L} N_i N_j}\sum_{i=1}^{L}\sum_{j=1}^{L}\sum_{i=1}^{N_i}\sum_{i=1}^{N_j}\left[\kappa\left(x_*^{(i)}, x_+^{(j)}\right)\left(\| x^{(i)} - x_+^{(j)} \|^{[} - 3\sigma^2 \| x_+^{(i)} - x^{(j)} \|^2\right)/\sigma^6\right]$$

3. Performance Analysis

The AES model was selected for this model because it is quicker at both encrypting and decrypting data. It would take 1 billion years to use a well-known brute force assault to break the 128-bit AES key. The standard symmetric encryption method used by US federal institutions has replaced DES with AES. AES employs 128-bit blocks (so there's no problem there), supports keys of 128, 192, or 256 bits (128 bits is already quite difficult to crack), and is effective in both software and hardware. It was chosen after several years and hundreds of cryptographers participated in an open competition. When comparing these algorithms' encryption and decryption times, we find that AES takes less time than DES and RSA to encrypt messages of various sizes.

Dataset description: The CERT dataset was produced by the Software Engineering Institute at Carnegie Mellon University and is extensively used by the insider threat research community. It is "free of restrictions and limitations on privacy." As a result, the CERT dataset is used to verify the effectiveness of our model. It includes insider activity logs that were produced from a network simulation of a real firm using complex models. The "dense needle" version of the dataset, "R4.2.tar.bz," has been used in this study since it has a sufficient number of red team scenarios. It consists of 1000 activity logs kept by insiders for a period of more than 17 months. Logon/off, file operations, HTTP, email, and removable device are among the log files that are included. Table 1 offers a brief summary of the dataset files.

Table 1. Description of dataset files

Files	Description
Logon	This file keeps track of when insiders log on and log out to a system. It includes insider identifiers, logon/off events, PC identifiers, and related timestamps.
File	Files include records of the specifics of file operations. csv
HTTP	The web browsing histories of insiders are contained in the http file. It includes timestamps, insider ids, PCS ids, visited URLs, and a few keywords.
Email	The e-mail.csv file contains information about insiders' email activity, including their email addresses, timestamps, email sizes, and attachments.
Device	The device keeps track of insiders' activity related to the use of removal devices. A csv file

Table-2 Comparative analysis between proposed and existing technique for data leakage detection

Parameters	ANN	CNN	DLD_CC_DLA
Accuracy	92	95	97
Precision	85	88	92
Recall	63	65	67
F1_Score	59	61	66
RMSE	53	55	62
SNR	55	58	61

The table-2 shows comparative analysis between proposed and existing technique data leakage detection in cloud computing based on data classification using deep learning architectures. Here parametric analysis is carried out in terms of accuracy, precision, recall, F-1 score, RMSE and SNR.

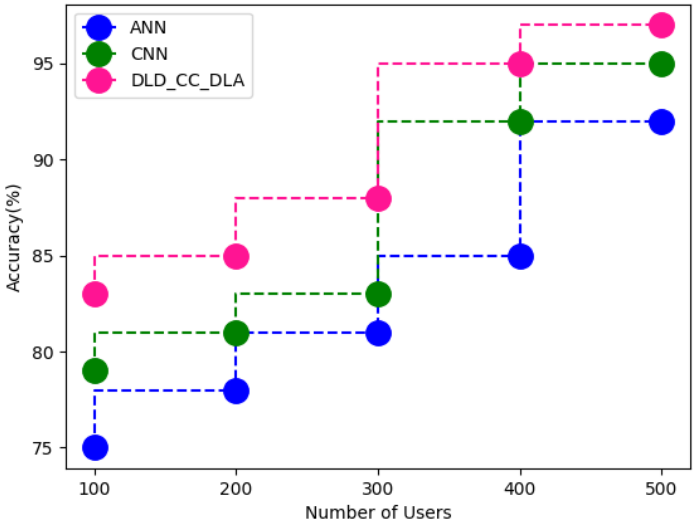

Figure-2 Comparison of accuracy

Above figure-2 shows comparative analysis between proposed and existing technique in terms of accuracy. Comparison has been carried out based on number of users and here the proposed technique has attained accuracy of 97%, existing ANN attained 92% and CNN attained 95%.

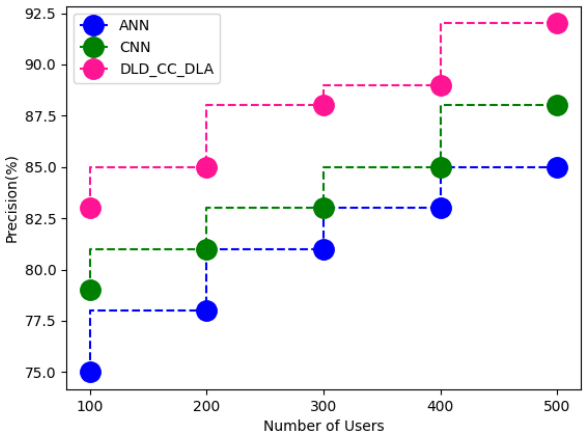

Figure-3 Comparison of precision

Above figure-3 shows comparison of precision between proposed and existing technique based on data classification for number users. Proposed technique attained precision of 92%, existing ANN attained 85% and CNN attained 88%.

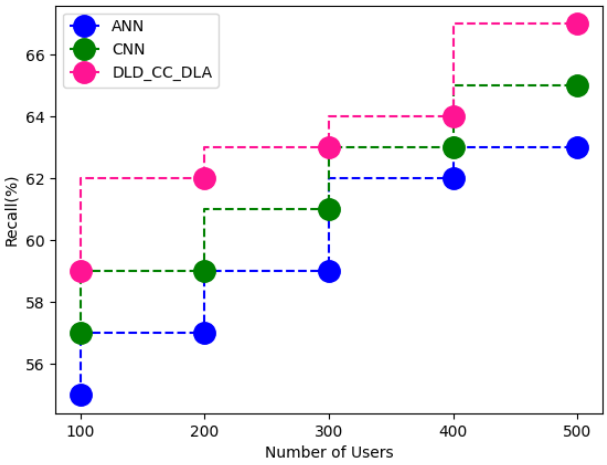

Figure-4 Comparison of recall

Based on number of users, recall between the proposed and existing techniques is compared in figure 4 above. The proposed method achieved a recall of 67%, compared to 63% for the current ANN and 65% for CNN.

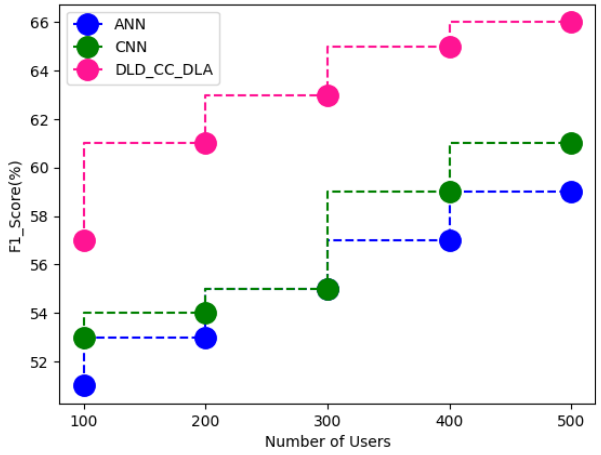

Figure-5 Comparison of F-1 score

From above figure-5 the comparison of F-1 score between proposed and existing technique. Proposed technique attained F-1 score of 66%, existing ANN attained 59% and CNN attained 61%.

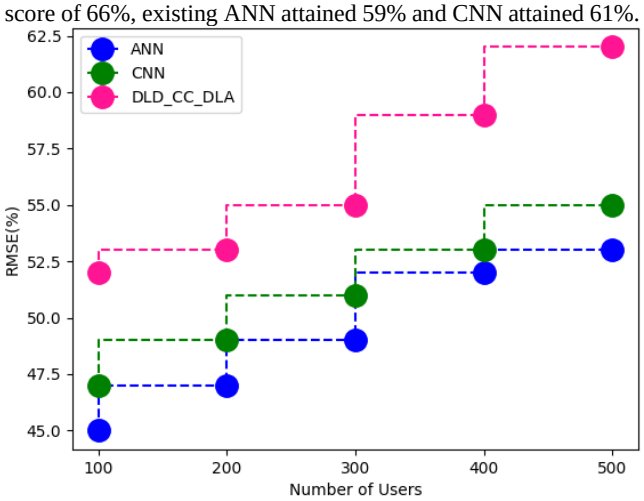

Figure-6 Comparison of RMSE

From above figure-6 the comparison of RMSE between proposed and existing technique. Proposed technique attained RMSE of 62%, existing ANN attained 53% and CNN attained 55%.

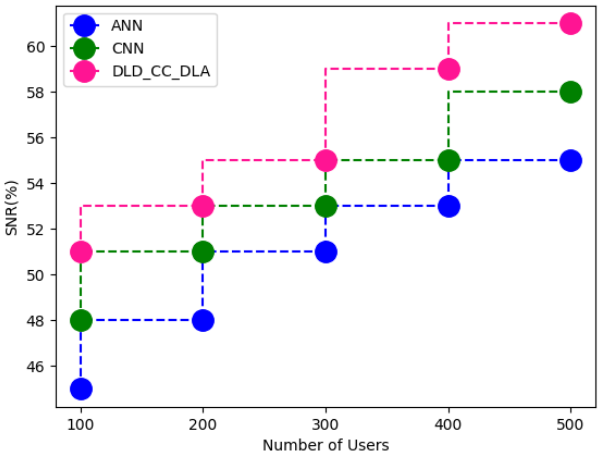

Figure-7 Comparative analysis of SNR

The above figure-3 shows comparison of SNR between proposed and existing technique based on data classification for number users. Deep metric learning uses the Signal-to-Noise Ratio (SNR) as a similarity metric. SNR is typically employed in signal processing to assess the ratio of a desired signal to noise level; a higher SNR value indicates a greater signal quality. Proposed technique attained SNR of 61%, existing ANN attained 55% and CNN attained 58%.

4. Conclusion

This research proposes novel technique in cloud computing based on data classification with data leakage using deep learning architectures. Here the input data has been classified has been done based on Generative Regression kernel support vector machine. Using this technique, we can identify the data leaker in real time. Additionally, it defends against several aggressive and passive assaults. The proposed method uses time and space efficiently while being computationally affordable. In order to prevent data leakage in a distributed computing environment, this can be useful. Since the suggested method is based on a symmetric algorithm, it is impossible to adapt it to a web context where numerous users frequently access the same data object. Parametric analysis is carried out in terms of accuracy, precision, recall, F-1 score, RMSE and SNR. The proposed technique attained accuracy of 97%, precision of 92%, recall of 67%, F-1 score of 66%, RMSE 62% and SNR of 61%. By lowering the confidence level placed in third party agents in the cryptographic services, our future work can be furthered. This will improve the system's ability to counter insider threat. Additionally, it examines the agent guilt model, which aids in stopping more leakage cases.

References

[1] Okochi, P. I., Okolie, S. A., & Odii, J. N. (2021). An improved data leakage detection system in a cloud computing environment. *World Journal of Advanced Research and Reviews*, *11*(2), 321-328.

[2] Gupta, I., Mittal, S., Tiwari, A., Agarwal, P., & Singh, A. K. (2022). TIDF-DLPM: Term and Inverse Document Frequency based Data Leakage Prevention Model. *arXiv preprint arXiv:2203.05367*.

[3] Gupta, I., & Singh, A. K. (2022). A Holistic View on Data Protection for Sharing, Communicating, and Computing Environments: Taxonomy and Future Directions. *arXiv preprint arXiv:2202.11965*.

[4] Mayuranathan, M., Saravanan, S. K., Muthusenthil, B., & Samydurai, A. (2022). An efficient optimal security system for intrusion detection in cloud computing environment using hybrid deep learning technique. *Advances in Engineering Software*, *173*, 103236.

[5] Singh, P., & Ranga, V. (2021). Attack and intrusion detection in cloud computing using an ensemble learning approach. *International Journal of Information Technology*, *13*(2), 565-571.

[6] Gupta, R., Saxena, D., & Singh, A. K. (2021). Data security and privacy in cloud computing: concepts and emerging trends. *arXiv preprint arXiv:2108.09508*.

[7] Chhabra, S., & Singh, A. K. (2022). A Comprehensive Vision on Cloud Computing Environment: Emerging Challenges and Future Research Directions. *arXiv preprint arXiv:2207.07955*.

[8] Al-Shehari, T., & Alsowail, R. A. (2021). An insider data leakage detection using one-hot encoding, synthetic minority oversampling and machine learning techniques. *Entropy*, *23*(10), 1258.

[9] Alshammari, S. T., & Alsubhi, K. (2021). Building a reputation attack detector for effective trust evaluation in a cloud services environment. *Applied Sciences*, *11*(18), 8496.

[10] Chaudhary, H., Chaudhary, H., & Sharma, A. K. (2022). Optimized Genetic Algorithm and Extended Diffie Hellman as an Effectual Approach for DOS-Attack Detection in Cloud. *International Journal of Software Engineering and Computer Systems*, *8*(1), 69-78.

[11] Wang, X., Pan, Z., Zhang, J., & Huang, J. (2021). Detection and elimination of project engineering security risks from the perspective of cloud computing. *International Journal of System Assurance Engineering and Management*, 1-9.

[12] Sharma, A., Singh, U. K., Upreti, K., & Yadav, D. S. (2021, October). An investigation of security risk & taxonomy of Cloud Computing environment. In *2021 2nd International Conference on Smart Electronics and Communication (ICOSEC)* (pp. 1056-1063). IEEE.