JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 14(2), JUNE, 2026
E-ISSN: 3093-0898, PRINT ISSN: 2277-0011; Journal homepage: www.atbufstejoste.com

Enhanced Deep Learning-Based Classification Approach for Detecting Data Leaks in Cloud Computing Environments

Abba Mohammed Yayaji, Badamasi Imam Ya'u
Abubakar Tafawa Balewa University, Bauchi Nigeria

ABSTRACT

In the realm of data security and privacy, organizations strive to safeguard sensitive information from unauthorized access and leakage. Data leakage, unintentional or malicious, can have serious implications, including financial losses, damage to reputation, and legal action. Traditional methods of data leakage detection often struggle to identify subtle patterns and anomalies. Whereas the network scenario that the study addresses needs the data leakage to be detected with highly consistent results for all the metrics, therefore, there is a need for advanced techniques that can effectively detect data leakage with high accuracy. This study suggests a deep learning approach for cloud computing that is based on data categorization and data leaking. Autoencoder (AE) architecture has been used to classify the input data. The developed AE classifier model was implemented using MATLAB R2023a. Support vector machines Convolutional neural networks, and artificial neural networks (SVM, CNN, and ANN) were used to compare the proposed model to current approaches.

ARTICLE INFO

Article History
Received: January, 2026
Received in revised form: April, 2026
Accepted: May, 2026
Published online: June, 2026

KEYWORDS

Deep Learning, Cloud Computing, Data Leakages, Autoencoder

INTRODUCTION

Data leaks are a major problem in today's business environment as they need to be protected from unauthorized access. A data breach occurs when private firm information is disclosed to unauthorized persons, either unintentionally or intentionally. It is critical to protect unauthorized individuals from accessing sensitive data, such as intellectual property rights, patents, functionality, and other essential information. This critical organizational information was frequently shared with parties outside the corporation. This can make it difficult to identify the person or organization responsible for a data breach. (Patil, Kumar, Narmadha, Suganthi, Rao & Rajesh 2022).

Data leakage may pose a security risk separate from the traditional, classical security levels of protection. Over the last two decades, businesses have increasingly relied on digital information to achieve their objectives. A considerable number of data activities on any given business day involve partners both within and beyond the organization's network borders. Much of this material is not sensitive; in most cases, it is typically labeled as "Sensitive" or "Exclusive," indicating that it should be protected from unauthorized access or display. This requirement can be driven by Data Loss/Leakage prevention programs, which secure data within the business by establishing basic rules and regulations and monitoring all types of data entering and leaving the organization (Hassan, Jincai, Iftekhar, Shehzad, & Cui, 2020).

An intrusion prevention system (IPS) is a security tool that uses a variety of security technologies to instantly stop infiltration and minimize dangerous network traffic. Model-based, signal-based, or knowledge-based fault detection techniques are used by most non-manual defect detection systems. These intelligent systems have three important components: data collecting, feature extraction, and data categorization (Singh & Ranga, 2021). To properly handle these challenges, cognitive data processing skills are necessary.

Corresponding author: Abba Mohammed Yayaji
yayajiabba@gmail.com
Abubakar Tafawa Balewa University, Bauchi Nigeria.

JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 14(2), JUNE, 2026
E-ISSN: 3093-0898, PRINT ISSN: 2277-0011; Journal homepage: www.atbufstejoste.com

RELATED WORKS

To prevent business employees or system insiders from disclosing critical information without authorization, a great deal of research has been done on data security. An automated device known as an intrusion detection system (IDS) continuously monitors network or computer activity in order to identify potential intruders. On the contrary, an intrusion prevention system (IPS) is a full-featured security solution that employs a number of security technologies to limit potentially harmful network traffic and immediately thwart invasions. Signal- based, model-based, or knowledge-based fault detection approaches are used by the majority of non-manual defect detection systems.

Defect detection systems frequently employ artificial neural networks (ANNs) and other classifiers to boost detection rates. For example, Patil et al. used deep learning classification techniques to evaluate and show data leakage detection in cloud computing systems using the CERT dataset in their work from 2022. Results from experiments show that the method is low-cost computationally, efficiently utilizes time and space, and can detect the data leaker in real time while thwarting a range of active and passive threats. The study's main shortcomings, however, are its incapacity to strengthen the system's defense against insider threats and its incapacity to adjust to an online environment where several users often access the same data item.

In a similar vein, Singh, Raj, Gupta, and Sayeed (2023) suggested using cloud computing to find and stop data breaches. The technique was created to help maintain formatted data integrity and stop unintentional unstructured material leaks during dissemination. The method includes a study plan for identifying certain data breaches in addition to a clear summary of data leak detection. It should be highlighted, though, that the approach did not specifically target certain businesses or industries, including banking and finance.

In addition, a hybrid deep learning approach-based security system for intrusion detection in a cloud computing environment was presented by Mayuranathan, Saravanan, Muthusenthil, and Samydurai (2022). For data processing, they employed the improved heap optimization (IHO) technique, which guaranteed data quality by eliminating superfluous data from the dataset. They also optimized feature selection using the Chaotic Red Deer Optimization (CRDO) technique. When working with large datasets, this method proved to be quite helpful in reducing dimensionality. Threats and intrusions into the cloud were categorized and found using the Deep Kronecker Neural Network (DKNN). The CSE-CIC-IDS2018 and DARPA IDS datasets were utilized to assess the system's performance. The outcomes were compared with intrusion detection systems that are currently in use. Moreover, Gupta, Mittal, Tiwari, Agarwal, and Singh (2022) employed a centroid document classifier technique to classify documents into the correct category, ensuring confidentiality and proper use of cluster data to avoid data leakage. The method could only gather records on a certain subject, such as government database.

The paper "Data Leakage in Notebooks: Static Detection and Better Processes" by Yang, Brower- Sinning, Lewis, and Kästner (2022) also examined notebook of data science code from 100,000 open- access notebooks and rated it using Sklearn fit and prediction algorithms. A static code analysis technique for identifying several kinds of notebook data leaks might be developed. The method was not able to identify test data that was missing from the notebook or repeated assessments that were not included in it. The study finds that the method undervalues reported leakage, produces a few false positives, and overlooks some leakages.

Also, through his research paper titled "Cloud Computing Security for Multi-Cloud Service Providers: Controls and Techniques in Our Modern Threat Landscape," Achar (2022) acquired a thorough comprehension of current cyber threats and their mitigation strategies, in addition to a thorough analysis of cloud security components and suggestions for multi-cloud service providers. The report digs into numerous aspects of cloud security, providing thorough insights and recommending best practices for multi-cloud service providers. The issues of providing security for a multi-cloud system were

Corresponding author: Abba Mohammed Yayaji
yayajiabba@gmail.com
Abubakar Tafawa Balewa University, Bauchi Nigeria.

JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 14(2), JUNE, 2026
E-ISSN: 3093-0898, PRINT ISSN: 2277-0011; Journal homepage: www.atbufstejoste.com

examined, and methods were proposed to address them. However, the approach fails to address the question of regulatory enforcement in different circumstances.

Kiranmai and Haritha (2021) used large-scale distributed query processing to create a machine learning-based method for detecting data breaches. They presented the deep machine learning-based Dynamic Inference-Based Rule Set Reduction technique. This method allows for dynamic inference- based rule set reduction and builds rule sets using a configurable threshold. This technique reduces the loss of dataset attribute information while simultaneously helping to increase temporal complexity. Its equivalence shows an accuracy of about 89%. The method is unable to provide distributed query processing that is both safe and effective.

Similarly, Zhang, Gao, Guo and Hu (2020), proposed scheme, “Improving the Leakage Rate of Ciphertext- Policy Attribute-Based Encryption for Cloud Computing (LR-CP-ABE)” to improve the rate of leakage in the prime order group, and achieve the maximum leakage rate using extension of lattice- based trapdoor. In accordance with the DLIN assumption scheme in the selective access policy model, the method has achieved anonymity, thereby enhancing users' privacy. It efficiently constructs prime-order groups, surpassing the performance of Composite order creation. However, a notable limitation of this approach is its failure to provide guidance on developing a more efficient method.

Faiz, Arshad, Alazab Shalaginov (2020) in their work collected data of historical email incident from reputable firms were used to conduct an experiment. Finally, the model was able to reliably predict approximately 95% of data loss instances, enabling the detection of actual data loss in email DLP. Furthermore, it enhanced the effectiveness of prioritizing real data loss through a human- understandable choice explanation.

Kaushik and Gandhi (2020) in their work, proposes a secure framework which resolves open research issues for secure access control, efficient revocation, assured file deletion and trust maintenance to provide a sound system for cloud. The framework was able to solved security problems by using a quorum of key managers and trusted third party.

Kumari, Game, Kawale, and Nikita's (2019) proposal, titled "Analysis of Data Leakage Detection Techniques Using Cloud Computing Environment," presents suggestions for guidelines for multi-cloud service providers to safeguard their consumers, particularly in the realm of identity and access management. However, one limitation of this strategy is that it does not address regulatory enforcement across different contexts.

Ariffin, Rahman, Darus, Awang, and Kasiran proposed "Detection of Data Leakage in Cloud Computing Environment" in 2019. They introduced an approach that involves recording packets throughout the VM replication and migration processes, as well as the online dashboard authentication procedure. A cloud testbed infrastructure based on OpenStack software and VMware vCenter was employed for emulation. When cloud management communication is not adequately secured, the platform can identify and effectively verify data leakage. However, the technique failed to uncover potential data leakage in other cloud components, such as block storage, API requests, and telemetry.

METHODOLOGY

The aim of this study is to enhance data leakage detection skills by developing a deep learning system that predicts the possibility of legitimate data loss. We conduct extensive testing using historical cloud incident data collected by an internationally renowned telecommunications firm, employing re- engineered deep learning techniques. Our research concentrates on leveraging deep learning methodologies to improve the capabilities of a conventional data leakage system, enabling proactive protection from insider threats.

We will carefully experiment and analyze real-world data given by a major telecoms operator in order to evaluate the utility of our technique in real-world systems. We measure classifier performance using many metrics such as accuracy, recall, precision, and F1 score. It is calculated using the confusion matrix, which

Corresponding author: Abba Mohammed Yayaji
yayajiabba@gmail.com
Abubakar Tafawa Balewa University, Bauchi Nigeria.

JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 14(2), JUNE, 2026
E-ISSN: 3093-0898, PRINT ISSN: 2277-0011; Journal homepage: www.atbufstejoste.com

displays the number of real and anticipated cases collected from the classifier.

Analysis of the Existing System

Patil et al. (2022) introduced a novel approach for detecting data leakage in cloud computing, based on data categorization using deep learning techniques in their study. The input data were collected as network data and were smoothed, with noise removed. Categorization relied on SVM with a Generative Regression kernel. The following metrics were used to determine the experimental results: recall, precision, accuracy, F-1 score, RMSE, and SNR. The proposed approach offers workable solutions to potential bias issues and class imbalance in order to build an insider data leak detection system. The proposed approach achieved a 97% accuracy, 92% precision, 67% recall, 66% F-1 score, 62% RMSE, and 61% SNR.

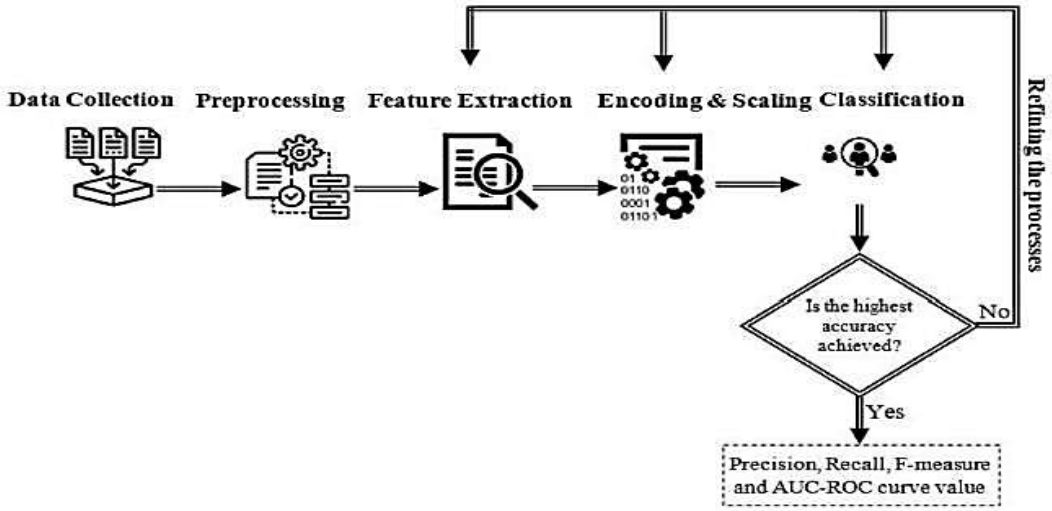

Figure 1: Architecture of the Existing System. (Patil et al., 2022)

Proposed Method

Our aim is to address and improve on the setback identified in previous work by Patil et al. (2022) where the SVM, in its training phase which is computationally extensive and largely dependent on the size of the input dataset. Larger datasets negatively impact SVM's overall performance, which affects both accuracy and computational costs. The Recall and f-score, two important measures, were too low (67% and 66%, respectively). We develop a data leakage detection system using Autoencoders, a type of artificial neural network designed for unsupervised feature learning and dimensionality reduction.

Autoencoders have shown promise in capturing intricate patterns within data, making them suitable for identifying data leakage instances. The proposed SVM address the aforementioned problems. The novel method uses an encoder encoder approach, developing an extremely fast and accurate algorithm. Accordingly, the proposed method can give more accurate and faster results than current SVM implementations.

Corresponding author: Abba Mohammed Yayaji
yayajiabba@gmail.com
Abubakar Tafawa Balewa University, Bauchi Nigeria.

JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 14(2), JUNE, 2026
E-ISSN: 3093-0898, PRINT ISSN: 2277-0011; Journal homepage: www.atbufstejoste.com

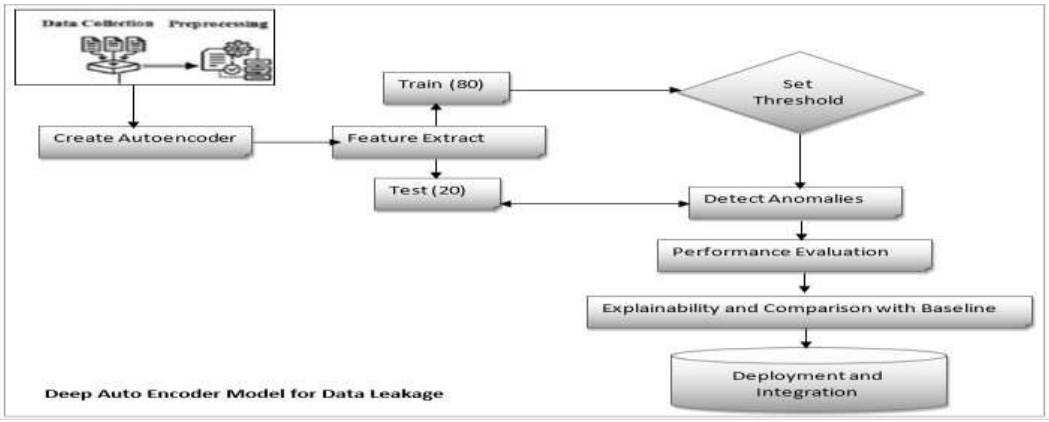

Figure 2: *Architecture of the Proposed Algorithm*

The aim here is to apply the novel method for identification of data leakage in cloud environment datasets. The detection rate of the proposed algorithms is evaluated against other state-of-the-art method using decision support accuracy such as (accuracy, recall, precision F-score, SNR and RMSE). The working principle of the proposed model is further explaining the next subsection.

Working principles

1. **Data Preprocessing:** Clean and preprocess the data, addressing issues such as missing values, outliers, and normalization. Ensure the integrity of the dataset while preparing it for input to the autoencoder.
2. **Dataset Creation:** Create a labeled dataset with both positive instances (data leakage cases) and negative instances (normal instances). Ensure that the dataset is balanced and representative of real-world scenarios.
3. **Autoencoder Architecture:** Design an autoencoder architecture comprising an encoder and a decoder. Experiment with the number of layers, nodes, and activation functions to optimize the model's performance.
4. **Training and Reconstruction:** Train the autoencoder on the dataset with normal instances. Monitor the reconstruction loss during training to ensure the model captures the underlying patterns.
5. **Anomaly Detection:** Utilize the trained autoencoder to reconstruct instances from the test set. Calculate the reconstruction error (difference between the input and the reconstructed output). Higher reconstruction errors are indicative of potential anomalies.
6. **Threshold Setting:** Set a threshold on the reconstruction error to differentiate between normal instances and potential data leakage cases. Fine-tune the threshold to balance precision and recall.
7. **Performance Evaluation:** Evaluate the autoencoder's performance using metrics such as accuracy, recall, precision, F1-score, and ROC curve area. Pay attention to false positives and false negatives.
8. **Explainability:** Analyze and interpret the learned features from the autoencoder to gain insights into the characteristics of normal and anomalous instances.
9. **Comparison with Baseline:** Compare the performance of the autoencoder-based approach with traditional methods like rule-based anomaly detection or SVM- based approaches.

Corresponding author: Abba Mohammed Yayaji
yayajiabba@gmail.com
Abubakar Tafawa Balewa University, Bauchi Nigeria.

JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 14(2), JUNE, 2026
E-ISSN: 3093-0898, PRINT ISSN: 2277-0011; Journal homepage: www.atbufstejoste.com

10. **Deployment and Integration:** Deploy the trained autoencoder model as a component of the organization's data infrastructure for real-time data leakage detection.

Table 1: Dataset Files Description

Files	Description
Logon	Serves as a log for insider activity. It includes details such as Insider IDs, logon and logoff events, PC identifiers, and relevant timestamps.
File	Files contain records of certain file operations. csv
HTTP	Stores web browsing history of insiders. It includes details such as timestamps, insider identifiers, visited URLs, PC identifiers, and a selection of keywords.
Email	Email activity is tracked in the e-mail.csv file, which includes email addresses, email sizes, timestamps, and attachments.
Device	Monitors insider behavior involving usage of removal devices. A spreadsheet file.

Testing and Training Phase

In order to verify the abnormal network leakages based on the deep autoencoder model in this research, a simulation experiment environment is built. The deep auto encoder classifier model is implemented using MATLAB R2023a. The computer to be used is a Dell laptop running on Windows 10 Operating System with 8GB RAM and Pentium ® Core i7 processor.

Evaluation Metrics

Accuracy: is concern with the prediction made correctly by the model, expressed as:

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN} \quad \ldots(1)$$

Precision: Describe the level of precision and accuracy of your model's anticipated positives, as well as the proportion of actual positives. It is a helpful indicator to determine if there is a significant cost associated with false positives. Expressed as:

$$\text{Precision} = \frac{TP}{TP+FP} \quad \ldots(2)$$

Recall attempts to determine what fraction of true positives were accurately detected. Expressed as:

$$\text{Recall} = \frac{TP}{TP+FN} \quad \ldots(3)$$

F-Measure is a function of both precision and recall, therefore when the class distribution is even and we need to strike a balance between the two, it could be preferable to utilize. Express as:

$$F1 = 2*\frac{Precision*Recall}{Precision+Recall} \quad \ldots(4)$$

TP (True Positive) refers to cases where the data point's actual and projected class are both 1 (true).
TN (True Negative) = Cases in which the data point's actual class was 0 (false) and the anticipated class was likewise 0.
FP (False Positive) = Cases in which the actual class of the data point is 0 (false) while the predicted class is 1 (true).
FN (False Negative) = Cases in which the actual class of the data point was 1 (true) while the predicted class was 0 (false).

RMSE: Root Mean Square Error (RMSE) is a popular metric in statistics and machine learning for determining the accuracy of a prediction model, notably in regression assignments. It measures the average size of mistakes or residuals between expected and actual (observed) values. The RMSE measures how closely the model's predictions match the ground truth data. It's calculated as:

Corresponding author: Abba Mohammed Yayaji
yayajiabba@gmail.com
Abubakar Tafawa Balewa University, Bauchi Nigeria.

JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 14(2), JUNE, 2026
E-ISSN: 3093-0898, PRINT ISSN: 2277-0011; Journal homepage: www.atbufstejoste.com

$$RMSE = \sqrt{\frac{\sum(y_i - \hat{y}_i)^2}{n}}$$

where n is the number of observations, y_i is the i_{th} observed value, $\acute{Y}_i$ is the corresponding predicted value, and y is the true value (or ground-truth value) **SNR: SNR** stands for Signal-to-Noise Ratio, and it is a measure used in various fields, including telecommunications, electronics and signal processing, to quantify the quality or strength of a signal relative to the background noise level. SNR is expressed as a ratio and is typically measured in decibels (dB).

The formula for calculating SNR in decibels (dB) is as follows:

SNR (dB) = 10 * log10 (Signal Power / Noise Power) …(6)

Where:
Signal Power: The power of the desired signal or the component of interest.
Noise Power: The power of the background noise or interference.

SIMULATION RESULTS AND ANALYSIS

Experimental Setup and Implementation of the System

The deep learning classifier model that was developed was implemented using MATLAB R2023a. The system used for this implementation was an HP laptop running on the Windows 10 Operating System. The laptop is equipped with 8GB RAM, a 1 TB HDD, and a Pentium® Core i7 processor. The developed data leakage detection system uses deep autoencoder architectures to classify the leak data from the normal data. In order to ascertain the best performing model, the simulation result is analyzed and compare in different phases including decision support and accuracy.

To implement the system, the datasets were provided as input to the network in CSV format, with the data transposed from columns to rows. The feature selection was conducted with an autoencoder neural network. The autoencoder was trained with a hidden layer of size 10 using linear transfer functions for the decoder. The sparsity percentage, L2 weight regularizer, and sparsity regularizer were all set. Dimensionality reduction and efficient feature selection were made possible by this arrangement. Thus, we extract the characteristics from the hidden layer. The characteristics obtained from the first autoencoder were used to train the second autoencoder. The second autoencoder's characteristics were then used to train a SoftMax layer for categorization. Finally, the encoders and SoftMax layer were layered to create a deep network. This tiered method enables more intricate and subtle patterns to be extracted from the data.

After developing the model, we setup division of data for Training, Validation, and Testing. In this case we set 80% of the data for training and 20% for testing. Hence, we detect the leakages using the deep network and the network performance was measured using the decision support and Accuracy.

Table 2: Parameters Settings for the Proposed Implementation

S/N	Parameters	Settings
1	number of hidden neurons	10
2	L2 weight regularizer	0.001
3	sparsity regularizer	4
4	sparsity proportion	0.05
5	Decoder Transfer Function	Purelin
6	Scale Data	False
7	SoftMax Layer	1

A high-level, interactive environment for technical computing is called MATLAB. It makes numerical computing, data analysis, data visualization, and algorithm building easier.

Corresponding author: Abba Mohammed Yayaji
yayajiabba@gmail.com
Abubakar Tafawa Balewa University, Bauchi Nigeria.

JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 14(2), JUNE, 2026
E-ISSN: 3093-0898, PRINT ISSN: 2277-0011; Journal homepage: www.atbufstejoste.com

MATLAB is particularly adept at addressing problems that require technical calculations, thanks to its incorporation of optimized algorithms into user-friendly commands. The primary elements of the MATLAB environment encompass the following: the Command Window, Command History, Workspace, Current Directory, Figure Window, and Edit Window.

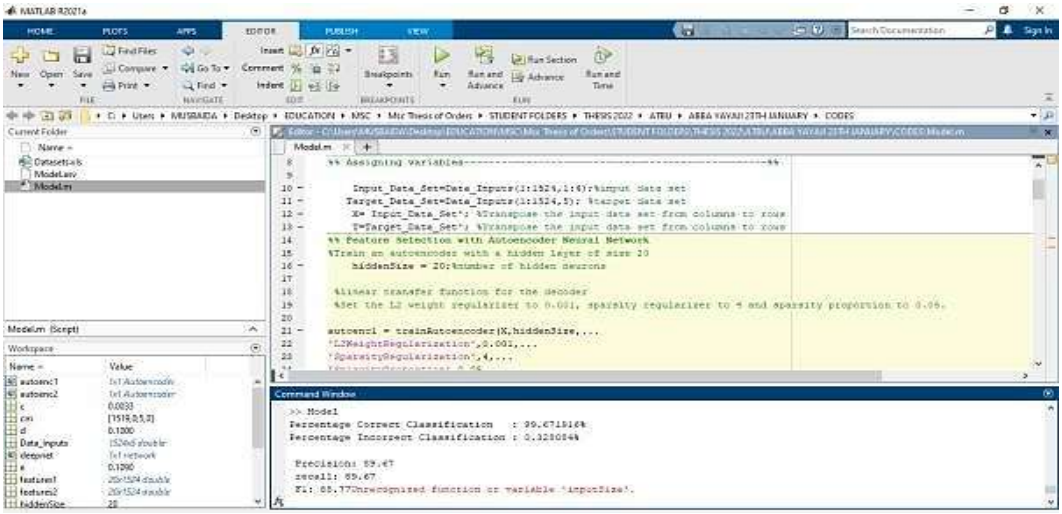

Figure 3: MATLAB Simulation Environment

Figure 4 below was obtained from the MATLAB neural network training toolbox portraying basic information including the parameters configuration, number of iterations, training progress, and epochs and validation accuracy for the proposed autoencoder network.

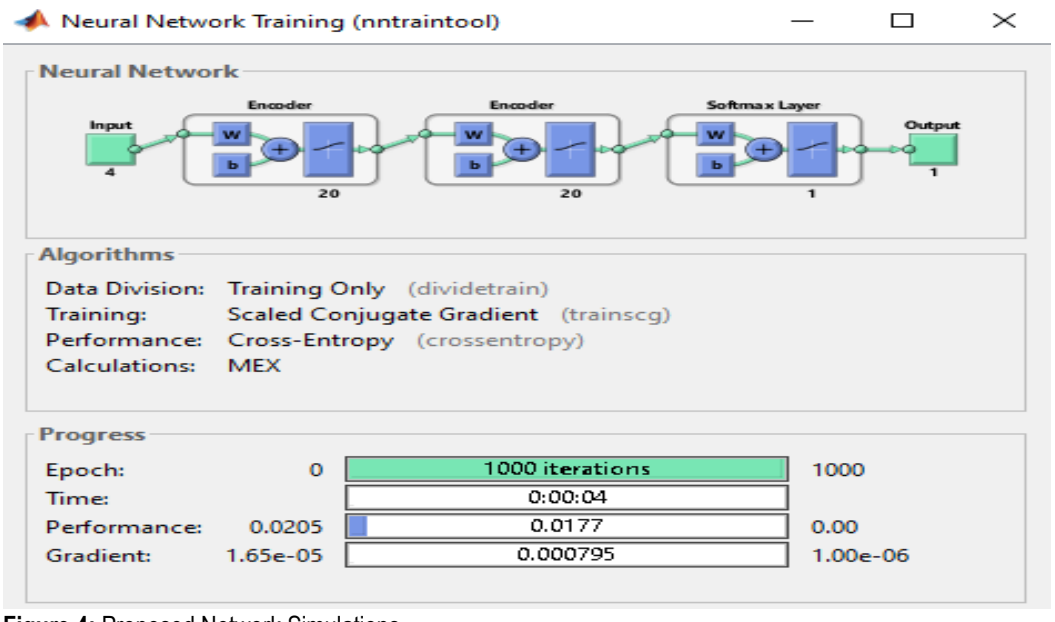

Figure 4: Proposed Network Simulations

Corresponding author: Abba Mohammed Yayaji
yayajiabba@gmail.com
Abubakar Tafawa Balewa University, Bauchi Nigeria.

JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 14(2), JUNE, 2026
E-ISSN: 3093-0898, PRINT ISSN: 2277-0011; Journal homepage: www.atbufstejoste.com

From Figure 4, it can be noticed that the scale conjugate was used as the default training algorithms for the autoencoders and the cross entropy were employed for performance measure. The model converges after 26 iterations. The proceeding sections will present the results achieved after the simulation experiment.

Results Presentation and Discussion

The proposed model's findings are shown, along with a comparison of suggested and current data leakage detection strategies in cloud computing that use data categorization and deep learning architectures. In this scenario, parametric analysis is carried out using accuracy, recall, precision, and F-score. The results presentation is separated into two sections. We use a confusion matrix to evaluate the suggested model's performance. Second, we evaluate the suggested model's performance against the baseline techniques.

Similarly, Figure 5 depicts the confusion matrix obtained after the simulationA confusion matrix serves as a comprehensive summary of the outcomes derived from a classification problem. It encapsulates the numerical counts of accurate and inaccurate predictions, further categorizing them by class. This is the primary function of the confusion matrix. It exposes the many ways in which your classification model gets confused while generating predictions. It provides information on not just the classifier's faults, but also the sorts of errors it makes. This breakdown discusses the disadvantages of depending only on categorization accuracy.

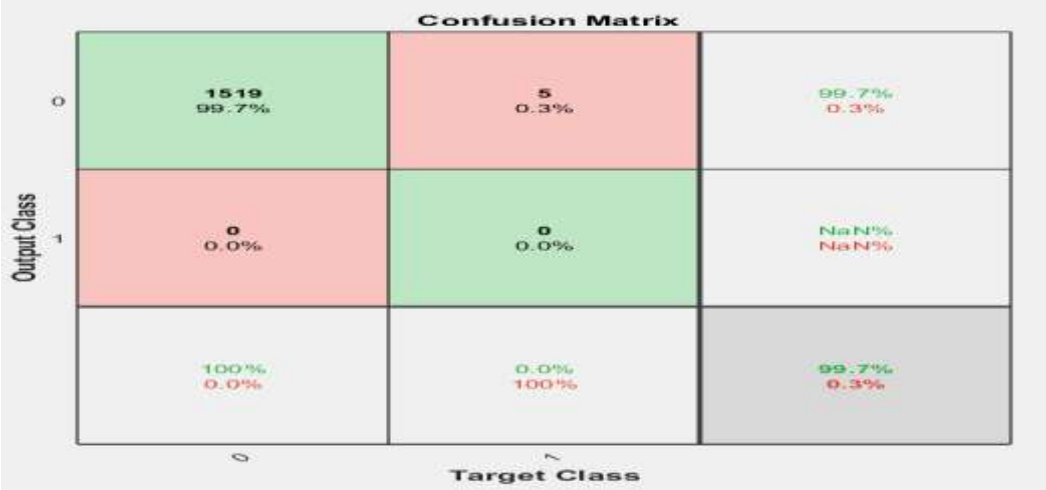

Figure 5: Confusion Matrix for the Proposed Model

As depicted in Figure 3, the cells colored in green on the diagonal represent the quantity and proportion of classifications that the trained network correctly identified. Conversely, the red cells on the diagonal symbolize the classifications that were incorrectly identified, which are zero for each target class and each output class. The square in the lower right corner, shaded in ash color, demonstrates the overall accuracy of the classifications. In this case, the proposed model attained 99.7% accuracy. Table 4 depicts the overall performance of the proposed for all decision support accuracy metrics.

Comparative analysis of results against baseline method

Table 3 compares proposed and current data leakage detection techniques in cloud computing based on data categorization using deep learning models. In this case, a parametric analysis is conducted in terms of accuracy, recall, precision, and F1 score.

Corresponding author: Abba Mohammed Yayaji
yayajiabba@gmail.com
Abubakar Tafawa Balewa University, Bauchi Nigeria.

JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 14(2), JUNE, 2026
E-ISSN: 3093-0898, PRINT ISSN: 2277-0011; Journal homepage: www.atbufstejoste.com

Table 3: Comparative Analysis against Baseline Methods for Data Leakage Detection

Algorithm	Accuracy	Precision	Recall	F-score	RMSE	SNR
Proposed	99.7	89.7	89.7	88.8	42.6	63.6
ANN	92.0	85.0	63.0	59.0	53.0	55.0
CNN	95.0	88.0	65.0	67.0	55.0	58.0
DLD_CC_DL	97.0	92.0	67.0	66.0	62.0	61.0

For decision support accuracy (Accuracy, recall, Precision, F-score, SNR and RMSE), the values were reported between 0, to 100. Values close to 100 means perfection detection performance while on the other hand, values close to zero implies poor detection rate. Hence, the higher the value, the better will be the performance of the model. With the exception of precision, where the SVM had the greatest score, table 4 shows that the suggested method performed the best in terms of accuracy, recall, and F-Measure. This illustrates how the suggested model outperforms the other current approach. The detailed explanation, analysis, and assessment based on the common evaluation measure used in this study are provided in the following subsections. These consist of f-measure, recall, accuracy, and precision.

RESULT AND ANALYSIS

Classification accuracy

This performance indicator looks at how well the model predicts the future. In machine learning, accuracy is a statistic used to assess a classification model's performance. Out of all the examples in a dataset, it calculates the percentage of correctly categorized instances, or data points. Stated differently, accuracy indicates the frequency with which the model's predictions agree with the actual class labels. The performance attained by the suggested model in comparison to cutting-edge methods is shown in Figure 8.

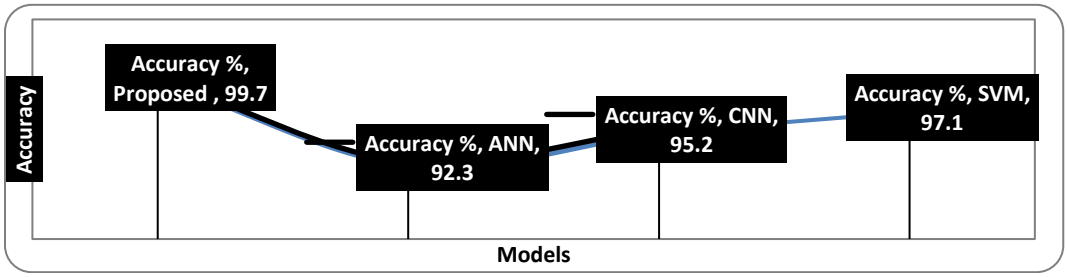

Figure 6: Classification Accuracy for all Methods

The proposed system achieved the best classification accuracy of 99.7 % followed by the SVM which achieved 97% and CNN with 95% respectively. The least performing model was the Conventional ANN model with 92%. This suggests that the proposed model predicted the correct instances of leakages with higher accuracy compare to the state of the art. Although accuracy is a fundamental evaluation metric, it may not always be the most effective approach to evaluate a model's performance, especially if the classes are not balanced. When one class predominates in the dataset, a model that forecasts the majority class at each occurrence can perform poorly yet still attain high accuracy. In such instances, various measures (such as recall, accuracy, and F1-score) are frequently utilized to offer a more thorough view of a model's performance, especially when dealing with unbalanced data. These metrics take into account issues like false positives, false negatives, and true positives to

Corresponding author: Abba Mohammed Yayaji
yayajiabba@gmail.com
Abubakar Tafawa Balewa University, Bauchi Nigeria.

JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 14(2), JUNE, 2026
E-ISSN: 3093-0898, PRINT ISSN: 2277-0011; Journal homepage: www.atbufstejoste.com

give a more comprehensive assessment of a model's capabilities. As a result, the accuracy was elaborated in the next subsection.

Precision

In certain instances, relying solely on accuracy for evaluating a model can lead to deceptive conclusions. The incorporation of Precision and Recall metrics can provide a more robust understanding of the true performance of the model, particularly in terms of its reliability across different problem scenarios. Precision is a parameter used in machine learning to assess a model's accuracy of positive predictions, especially in the setting of binary classification. "Out of all the instances that the model predicted as positive, how many were actually positive?" is the question it answers. Stated differently, precision is concerned with the caliber of accurate forecasts. It is the proportion of successfully predicted positive cases, or true positive predictions, to all instances that were correctly forecasted as positive (true positives + false positives).

Figure 9 depicts the performance achieved by the proposed method against the existing method for the precision.

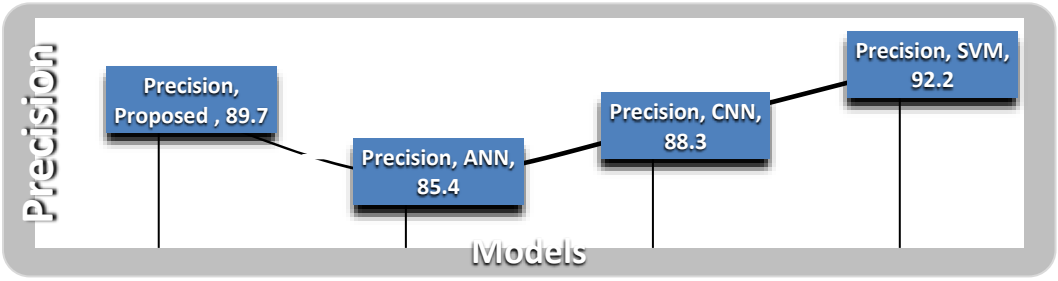

Figure **7**: Precision for all Methods

From Figure 7 above, it was observed that the SVM model outperforms the proposed model by achieving the highest precision of 92%. However, the proposed model received the second-best score, 89.7%. This outperformed the ANN (85%) and the CNN model (88%). However, accuracy does not account for cases in which the model overlooked positive samples (false negatives), which is where recall comes into play. When both false positives and false negatives have different meanings, accuracy and recall must be balanced to produce the best model performance. Thus, the recall score is analyzed in the next subsection.

Recall

As stated previously, precision and recall allow us to better comprehend how strong the accuracy displayed is for a certain situation. Figure 10 compares the performance of the suggested strategy to the existing method for recall.

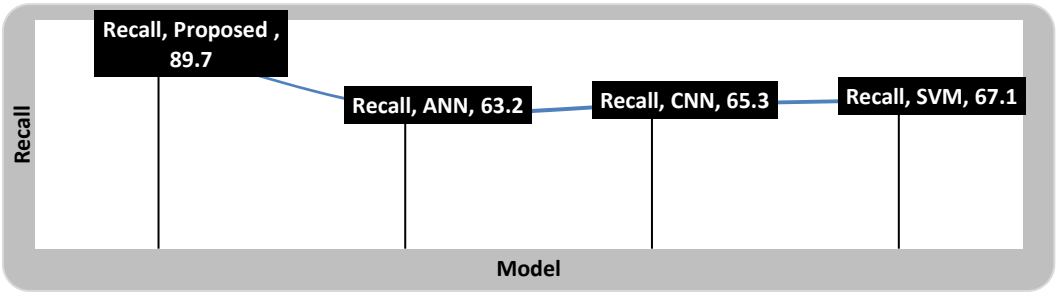

Figure 8: Recall for all Methods

Corresponding author: Abba Mohammed Yayaji
yayajiabba@gmail.com
Abubakar Tafawa Balewa University, Bauchi Nigeria.

JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 14(2), JUNE, 2026
E-ISSN: 3093-0898, PRINT ISSN: 2277-0011; Journal homepage: www.atbufstejoste.com

From Figure 8 above, the proposed system achieved the best recall of 89.7.% as against the other baseline methods. This performance was superior when compared to the recall value achieved by all the other methods (ANN 63%, CNN 65% and SVM 67%) respectively. However, precision and recall are often combined into a single metric called the F1-score, which provides a balance between the two.

F-Measure

As stated earlier, F-Measure produces a single score that addresses both accuracy and recall issues with a single number. The F1 score, also referred to as the F- measure or F-score, serves as a metric for evaluating the accuracy of a test in binary classification statistical analysis. It is derived from the test's precision and recall values. Precision, in this context, is defined as the ratio of accurately identified positive outcomes to the sum of all identified positive outcomes, including those that were inaccurately identified. Recall, on the other hand, is the number of correctly identified positive results divided by the total number of samples that should be considered as positive. In each case a higher value shows how confident the classification accuracy or performance can be relied upon.

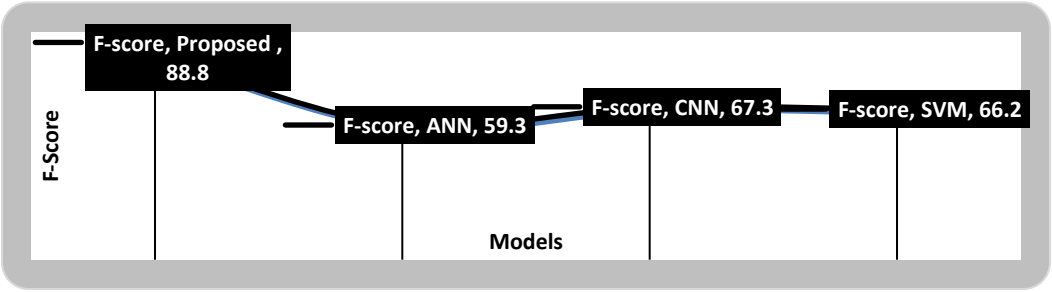

Figure 9: F-Measure for all Methods

From Figure 9 above, the proposed system achieved the highest F-score of 88.8% as against the other classical approaches. Therefore, by achieving higher values both in terms of the accuracy, recall and F- score, the proposed model has further cemented its overall superiority in classifying the cases of leakages from the network datasets as compare to the state- of-the-art methods.

Based on the analyzed results presented above, it is evident that the proposed deep learning model outperforms all others, securing the top position in all cases. The SVM model follows closely in second place, only falling short when compared to CNN and ANN among the baseline algorithms.

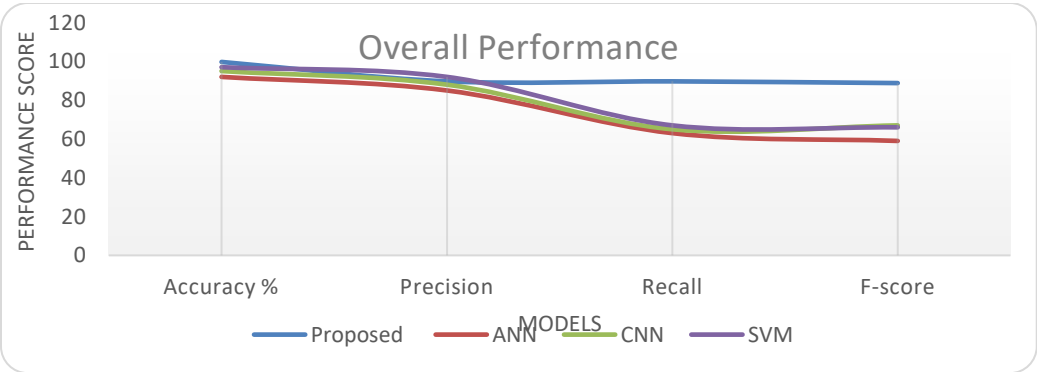

Figure 10: Overall performance for all Methods

Corresponding author: Abba Mohammed Yayaji
yayajiabba@gmail.com
Abubakar Tafawa Balewa University, Bauchi Nigeria.

JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 14(2), JUNE, 2026
E-ISSN: 3093-0898, PRINT ISSN: 2277-0011; Journal homepage: www.atbufstejoste.com

Moreover, the optimal performance of the proposed deep learning model is further highlighted by its near- perfect scores in accuracy, recall, precision, and F- measure, all of which approach 100%. However, precision, recall, and F1-score are metrics more commonly used for classification tasks where we're interested in the balance between true positives, false positives, true negatives, and false negatives. Hence, in this research, we extend our focus to prediction evaluation metrics RMSE and SNR to assess the accuracy and performance of the proposed model. These evaluations will give a more accurate picture of how well the model predicts continuous variables. Figure 12 shows the performance of the suggested model in terms of predicted accuracy.

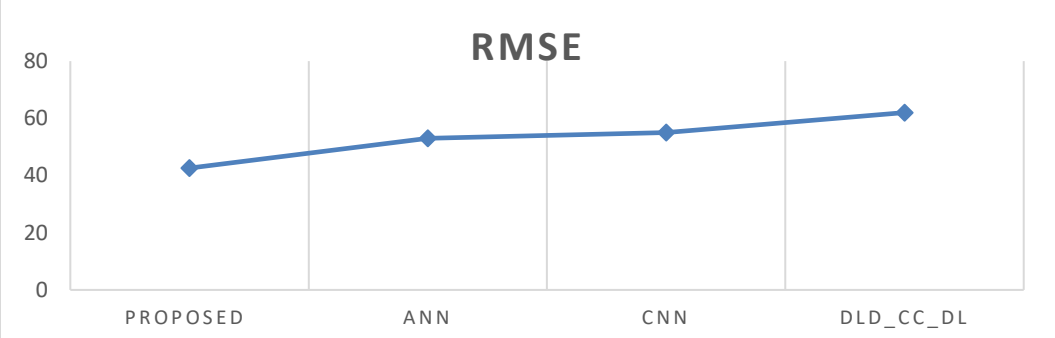

Figure 11: RMSE for all method

Root Mean Square Error (RMSE) is a popular metric in statistics and machine learning for determining the accuracy of a prediction model, especially in regression assignments. It measures the average size of mistakes or residuals between expected and actual (observed) values. The RMSE measures how closely the model's predictions match the ground truth data. The RMSE is reported in the same units as the target variable, making it easier to understand and compare between models or datasets. Since they demonstrate that the model's predictions are more in line with the actual data, lower RMSE values are indicative of better model performance. The proposed model demonstrates it superior performance by achieving the lowest RMSE score of 42.6 compare to the ANN which achieved RMSE of 53, CNN which achieved RMSE of 55.0 and DLD_CC_DL which achieved RMSE of 62 respectively.

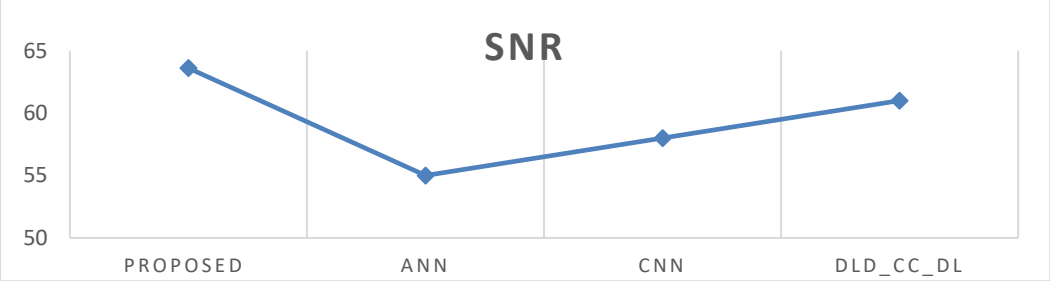

Figure 12: SNR for all method

Similarly, in machine learning, SNR can be relevant when dealing with feature selection or feature engineering. It helps determine which features or variables are most informative for a predictive model by comparing the strength of the signal (useful information) to the level of noise (uninformative or random variation). Features with a higher SNR are typically more relevant for

Corresponding author: Abba Mohammed Yayaji
yayajiabba@gmail.com
Abubakar Tafawa Balewa University, Bauchi Nigeria.

modeling. A higher SNR value indicates a greater signal quality. Thus, from Figure 13, the proposed model demonstrates its superior performance by achieving the highest SNR score of 63.6 compared to the ANN which achieved an SNR of 55, CNN which achieved an SNR of 58.0 and DLD_CC_DL which achieved an SNR of 61.0% respectively. Hence, this indicates a better signal quality.

The success of this approach relies on the autoencoder's ability to capture latent features that distinguish normal instances from data leakage instances. Additionally, the system's precision and recall trade-off were optimized to minimize false positives and false negatives, considering the critical nature of data leakage detection.

CONCLUSION, RECOMMENDATIONS AND FUTURE WORK

This study presents a deep learning approach for cloud computing that is based on data categorization with data leakage prevention techniques and employs deep learning architectures. The input data were categorized using autoencoder architecture in this case. The developed deep learning model classifier was implemented in MATLAB R2023a. Experimental results show that the proposed system achieved the best classification accuracy of 99.7 % followed by the SVM which achieved 97% and CNN with 95% respectively. The Conventional ANN model has the lowest performance rate (92%).

When various measures are used to evaluate, it is clear that the suggested deep learning approach comes out on top in all circumstances, while SVM comes in second place, behind only CNN and ANN on the baseline algorithms. Furthermore, it is obvious that the suggested deep learning achieves the maximum degree of performance by achieving accuracy values extremely near to 100%. The success of this approach relies on the autoencoder's ability to capture latent features that distinguish normal instances from data leakage instances. Additionally, the trade-off of precision and recall of the system were optimized to minimize false positives and false negatives, considering the critical nature of data leakage detection.

We recommend the integration and deployment of the developed model into organization real time network environment. In our future work, we recommend using computational time as an indicator to evaluate the quality of the models.

REFERENCES

Abdurachman, E., Gaol, F. L., & Soewito, B. (2019). Survey on threats and risks in the cloud computing environment. *Procedia Computer Science, 161*, 1325-1332.

Achar, S. (2022). Cloud Computing Security for Multi- Cloud Service Providers: Controls and Techniques in our Modern Threat Landscape. *International Journal of Computer and Systems Engineering, 16*(9), 379-384.

Aggarwal, T., & Kaur, N. (2019). Data Leakage Detection using Cloud Computing: IRJET.

Alos, A., & Dahrouj, Z. (2020). Decision tree matrix algorithm for detecting contextual faults in unmanned aerial vehicles. *Journal of Intelligent & Fuzzy Systems, 38*(4), 4929-4939.

Ariffin, M. A. M., Rahman, K. A., Darus, M. Y., Awang, N., & Kasiran, Z. (2019). Data leakage detection in cloud computing platform. *Int. J. Adv. Trends Comput. Sci. Eng, 8*(1.3), S1.

Ayyad, S. M., Saleh, A. I., & Labib, L. M. (2019). Gene expression cancer classification using modified K-Nearest Neighbors technique. *BioSystems, 176*, 41-51.

Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC genomics, 21*, 1-13.

Faiz, M. F., Arshad, J., Alazab, M., & Shalaginov, A. (2020). Predicting likelihood of legitimate data loss in email DLP. *Future Generation Computer Systems, 110*, 744-757.

Corresponding author: Abba Mohammed Yayaji
yayajiabba@gmail.com
Abubakar Tafawa Balewa University, Bauchi Nigeria.

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*.

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*.

Hassan, M., Jincai, C., Iftekhar, A., Shehzad, A., & Cui, X. (2020). Implementation of security systems for detection and prevention of data loss/leakage at organization via traffic inspection. *arXiv preprint arXiv:2012.14111*.

Kaushik, S., & Gandhi, C. (2020). Capability based outsourced data access control with assured file deletion and efficient revocation with trust factor in cloud computing. *International Journal of Cloud Applications and Computing (IJCAC), 10*(1), 64-84.

Kiranmai, M., & Haritha, D. (2021). Detection of Data Leaks through Large Scale Distributed Query Processing using Machine Learning. *International Journal of Advanced Computer Science and Applications, 12*(12).

Kong, F., Zhou, Y., Xia, B., Pan, L., & Zhu, L. (2019). A security reputation model for IoT health data using S-AlexNet and dynamic game theory in cloud computing environment. *IEEE Access, 7*, 161822-161830.

Kumari, M. A., Game, M. R., Kawale, M. S., & Nikita, M. (2019). Analysis of Data Leakage Detection Technique using Cloud Computing Environment.

Mayuranathan, M., Saravanan, S., 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.

Patil, R. C., Kumar, A., Narmadha, T., Suganthi, M., Rao, A. V. S. R., & Rajesh, A. (2022). Data Leakage Detection in Cloud Computing Environment Using Classification Based on Deep Learning Architectures. *International Journal of Intelligent Systems and Applications in Engineering, 10*(2s), 281-285.

Saleh, A. I., Shehata, S. A., & Labeeb, L. M. (2019). A fuzzy-based classification strategy (FBCS) based on brain–computer interface. *Soft Computing, 23*(7), 2343-2367.

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

Singh, V., Raj, M., Gupta, I., & Sayeed, M. A. (2023). Data Leakage Detection and Prevention Using Cloud Computing *Sustainable Computing* (pp. 159-169): Springer.

Xuming, L., Lina, C., Peng, J., Xiao, G., & Shuo, C. (2019). *Current status and future prospects of data leakage prevention technology: A brief review.* Paper presented at the Journal of Physics: Conference Series.

Yang, C., Brower-Sinning, R. A., Lewis, G. A., & Kästner, C. (2022). Data Leakage in Notebooks: Static Detection and Better Processes. *arXiv preprint arXiv:2209.03345*.

Zhang, I., Gao, X., Guo, F., & Hu, G. (2020). Improving the leakage rate of ciphertext-policy attribute-based encryptionfor cloud computing. IEEE Access, 8.94033-94042.

Corresponding author: Abba Mohammed Yayaji
yayajiabba@gmail.com
Abubakar Tafawa Balewa University, Bauchi Nigeria.