www.ijecs.in
International Journal Of Engineering And Computer Science
Volume 15 Issue 01 January 2026, Page No. 28095-28114
ISSN: 2319-7242 DOI: 10.18535/ijecs/v15i01.5386

Performance Evaluation of Machine Learning-Based Data Leakage Detection and Prevention Techniques

Devika Singh[*1], Akansha singh[2]

[*1]Research scholar, School of Computer Applications, Babu Banarsi Das University, Lucknow
[2]Assistant Professor, School of Computer Applications, Babu Banarsi Das University, Lucknow

Abstract
According to the study, performance of machine learning based data leakage detection and prevention techniques was investigated under enterprise-oriented conditions. The pre-processing pipeline of supervised and unsupervised models is similar. Feature engineering, model implementation, and evaluation metrics were similarly matched up on the same experimental design to ensure reproducibility of the model. The models, which realistically simulate insider leakage, were engineered through the amalgamation of the TF-IDF representations, behavioural ratios, and contextual indicators. The researchers evaluated the performance of supervised models (Logistic Regression, Support Vector Machine, and Random Forest), unsupervised models (Isolation Forest, K-Means), and deep learning models (Deep Neural Network (DNN) and Long Short-Term Memory (LSTM)). This evaluated occurred under 70:30 train-test split and 5-fold cross validation. According to experimental findings, deep learning models performed better than traditional approaches, with LSTM yielding the best performance, namely a detection accuracy of 96.1 percent, an F1-score of 96.1 percent and a ROC–AUC of 0.97 at a low false positive rate of 4.5 percent. The performance of the resulting hybrid ML framework was also enhanced, resulting in an accuracy of 96.8%, an F1-score of 96.6% and a false alarm rate lowered to 3.9%. In fact, the Random Forest algorithm created a trade-off between accuracy and interpretability (94.5%). Subsequently, the unsupervised Isolation Forest had many false positives of 13.5% and could not be used for real-time prevention for this reason. An important trade-off of accuracy and latency was revealed in the study which raised the necessity of hybrid, explanation ready and scalable frameworks for real-time data leakage detection and prevention in enterprise scenarios.

Keywords: *Data Leakage Detection, Data Leakage Prevention (DLP), Machine Learning, Deep Learning, Insider Threats, Anomaly Detection, Email Security, Cybersecurity, Feature Engineering, Explainable Artificial Intelligence (XAI), Real-Time Detection, Enterprise Information Security*

1. Introduction

1.1 Background Context

In today's digital world, organizations produce, store and exchange a lot of sensitive information like IP, Finances, Personal Data, Communications, etc. As businesses use more emails, cloud, collaborate, etc., the chances of accidental or deliberate leakage of data is increasing a lot. Data leakage is the unauthorized transfer or flow of information from an organization to an external recipient(Lazer et al., 2018). This may be intentional or unintentional. The data leakage is mainly due to insiders that is due to one of the employees of the company unlike cyber attacks(Fernandes et al., 2019).The incidents of leakage can be categorized as accidental leakage, Malicious insider leakage and external leakage via compromised accounts(Asade et al., 2025). Unintentional sharing of sensitive data occurs due to leakages by employees from time to time. Employees or contractors who misuse their access for personal benefit are known as malicious insider threats(Verma et al., 2020). When attackers take over legitimate user accounts, they feed false data through a normal communication channel to siphon off information(Amomo, 2022). The mentioned leakage scenarios are different and show how the problem is not simple. Thus there is a need for intelligent detection.

1.2 Limitations of Traditional DLP Systems

Traditional data leakage prevention systems largely depend on rules, keyword matching, and predefined policies. Although these systems work well in controlled scenarios with predictable leakage patterns, they are limited by a number of constraints. Constant manual updates of static rules are required because of changes to data formats and attack strategies. The detecting method which relies on keywords tends to produce multiple false positives as there could be actual leakage but no detection(Padhiar & Patel, 2023). Additionally, such systems fail to leverage the context as well as behavior of data use, such as unusual frequency of communication or unintended behavior.

Yet another significant drawback of conventional DLP solutions is their inability to adapt to new or known leakage patterns. Because these systems rely on prior signatures, they are ineffectual against zero-day leakage techniques or any subtle insider threat that does not conform to existing signatures. This generally leads to alert fatigue, trust erosion in security systems, and delay in response to exist real threats(Liu et al., 2018).

1.3 Role of Machine Learning in Data Leakage Detection

Machine learning constitutes a hopeful alternative to rule-based DLP as it can help systems learn historical patterns and generalize to unknowns. Using a great deal of structured and unstructured data, ML-based detection systems are capable of analysing of complex correlations. Propelled by supervised classification, unsupervised anomaly detection and deep learning, ML-based systems can use content, behaviour and context in the detection process(Homoliak et al., 2020).

Supervised learning models utilize a labeled dataset for training, distinguishing between a leakage and non-leakage instance with high accuracy, provided that sufficient labeled data is available. Models employing unsupervised learning techniques identify anomalies through the study of normal states(Al-Mhiqani et al., 2020). Thus, they are an excellent fit for novel leakage events. By capturing the fundamental hierarchical and temporal patterns that are present in the data, deep learning models can enhance detection capabilities. This is particularly true in a sequential communication stream like an email order. As a result of these benefits, machine learning will enable the next generation of data leakage detection and prevention systems(Tuor et al., 2017).

1.4 Problem Statement

As organizations increasingly adopt machine learning for data leakage detection, several technical issues remain. High performing ML models generally do not satisfy enterprise-level real-time processing requirements due to computation and latency constraints. Detection is further complicated by encrypted and obfuscated data which foils content analysis.In addition, many ML models work in a black box fashion, making them difficult to explain and raising compliance, auditing, or trust issues. Another important issue is a lack of unified benchmarking frameworks that fairly compare different models in consistent evaluation conditions(Tabrizchi & Kuchaki Rafsanjani, 2020).

1.5 Research Gap

Research shows that there is no real-world dataset evaluation and performance comparison across all models. Detection accuracy is the main focus of most work while prevention is not well studied Additionally, there is no systematic approach to integrating natural language processing, anomaly detection and explainable AI in unified DLPs. The study's objectives in light of these gaps are to benchmark a range of ML models for Data leakage detection, analyze the accuracy-latency trade-off, discern their limitations in prevention deployment, and signal towards the way for hybrid ML-based DLP system(Sindiramutty et al., 2024).

1.6 Research Objectives

- **O1:** To benchmark machine learning models for data leakage detection across supervised, unsupervised, and deep learning approaches.
- **O2:** To evaluate accuracy–latency trade-offs of machine learning–based data leakage detection models in enterprise environments.
- **O3:** To identify limitations in existing machine learning models with respect to real-time data leakage prevention capability.

- **O4:** To propose future directions for hybrid machine learning–based data leakage prevention frameworks that balance accuracy, scalability, and explainability.

Novelty

This paper proposes a unified evaluation framework for machine learning–based data leakage detection and prevention by analyzing supervised, unsupervised, and deep learning models jointly on consistent datasets and metrics. The study emphasizes real-time feasibility, false positives as well as prevention capability rather than just detection accuracy like previous work. Combining textual, behavioural, and contextual aspects gives a better depiction of leakage scenarios that insider threats entail, and vice-versa.

Scientific Contributions

This research compares many machine learning frameworks, specifying the accuracy-latency trade-off that is salient to enterprise deployment. The assessment goes beyond simply detecting, encompassing preventive measures such as waiting time or false alarms. In this paper, we have introduced a promising architecture to be considered for the future PDRs based on machine learning.

2. Literature Review

2.1 Evolution of Data Leakage Detection Techniques

The development of data leakage detection methods has evolved over the years against cyber-attacks(Bouke & Abdullah, 2023). DLP systems during the initial stages were rule-based which involved mainly the matching of keywords, regular expressions, and policies. Although easy to implement, they are not flexible and adaptable(Herrera Montano et al., 2022). Signature based approaches were developed as an enhancement, matched fingerprints and hashes to identify known leakage patterns. Yet, reactive methods were not effective against a new threat(Wu et al., 2024).

The emergence of machine learning gave a major push towards not only proactive but also adaptive detection. ML uses past data to identify features characterising leaks and those of normal behaviour. These systems minimize reliance on handcrafted rules and enhance detection performance across various data types through the utilization of statistical and probabilistic models(Jadhav & Chawan, 2019).

2.2 Machine Learning Models in Prior Research

Support Vector Machines (SVM) have been a popular choice in data leaks detection due to their usefulness in a high-dimensional feature space especially text classification. While the SVMs are not scalable to large datasets(Gamachchi et al., 2018). Random Forest models are more robust to noise and provide measures of feature importance, making them appealing for interpretability. The simplified and clear nature of Logistic Regression is what makes it a baseline model.

Deep learning models such as deep neural networks and recurrent architectures like LSTM have superior recall and accuracy for complex leakage patterns(Sarker, 2021). These models are very adept at capturing semantic and temporal relationships but present difficulties related to cost and explainability. Models that require no supervision, such as Isolation Forest and clustering algorithms, have been utilized to identify anomalous behavior. This occurs when there is a lack of labeled data(Nayak & Ojha, 2020).

2.3 Limitations in Existing Studies

While the existing studies yield promising results, the models are often evaluated in isolation and limited round experimentation(Rawat et al., 2019). Only a few works address the multiple requirements of accuracy, real-time feasibility, explainability, and more. In addition, ML-based detection frameworks seldom incorporate blocking and policy enforcement mechanisms aimed at preventing an incident. The system-level analysis and holistic evaluation of performance are required at this stage.

3. Methodology

The methodology of this study was quantitative and experimental carrying an evaluation of industrial performance of machine learning based data leakage detection and prevention techniques. Through the application of identical preprocessing pipelines, identical feature engineering strategies and identical evaluation metrics, we achieve comparability and reproducibility and robustness in supervised, unsupervised and deep learning models. The methodology included data preprocessing, feature extraction, model training, performance evaluation, and prevention analysis.

3.1 Dataset description

The experimental evaluation in this study was conducted using a publicly available and widely accepted enterprise communication dataset(*Insider Threat Test Dataset,* 2020), namely the Enron Email Corpus, complemented with synthetic insider-leakage annotations and behavioural/contextual attributes inspired by the CERT Insider Threat Dataset(*Enron Email Dataset,* 2015). The Enron Email Corpus contains over 500,000 real-world corporate emails exchanged among employees, including email bodies, subjects, timestamps, sender–receiver relationships, and attachment metadata, making it highly suitable for modeling enterprise data leakage scenarios. To enable supervised learning and prevention analysis, leakage and non-leakage labels were assigned based on content sensitivity, abnormal transmission patterns, recipient anomalies, and contextual policy violations, following established DLP simulation practices. Textual features were extracted using TF–IDF representations of email content and attachments, while behavioural features captured communication frequency, file size deviations, and access irregularities. Contextual features included user roles, communication channels, and data sensitivity levels. This combined dataset realistically represented insider threat and data leakage behaviour while ensuring reproducibility, scalability, and compliance with ethical research standards.

Data Preprocessing and Normalization

The textual data was pre-processed to remove noise and standardise input specifications. To control for bias, we calculated the normalized term frequency of all terms in question.

Term Frequency Normalization

$$TF(t,d) = \frac{f(t,d)}{\sum_k f(k,d)} \quad (1)$$

where $f(t,d)$represents the frequency of term $t$in document d, and the denominator represents the total number of terms in document d.

To assess the importance of a term across the entire corpus, inverse document frequency was calculated.

Inverse Document Frequency

$$IDF(t) = \log\left(\frac{N}{DF(t)}\right) \quad (2)$$

where Ndenotes the total number of documents and $DF(t)$is the number of documents containing term t.

These two components were combined to form the TF–IDF representation.

TF–IDF Weighting

$$TF\text{-}IDF(t,d) = TF(t,d) \times IDF(t) \quad (3)$$

This weighting scheme emphasized sensitive and rare terms that are critical for identifying potential data leakage.

3.2 Behavioural and Contextual Feature Modelling

By analysing the change in communication pattern, the deviation from normal behaviour was noted to suspect any data leaking. Email sending frequency, attachment size, communication timings and other behavior-related features were normalized to identify any abnormal spikes or unusual trends. To estimate the risk level for every interaction, the study created compositional contextual attributes that included sender rank, recipient domain and internal–external communication ratio. By complementing content-based capability with usage and role-based inputs this combined behavioural–contextual modelling could enhance insider threat detection.

Behavioural Anomaly Score

$$Z_u = \frac{B_u - \mu_B}{\sigma_B} \quad (4)$$

where $B_u$denotes the observed behavioural metric for user u, $\mu_B$is the mean behaviour, and $\sigma_B$is the standard deviation.

Communication imbalance between internal and external domains was measured as a ratio.

External Communication Ratio

$$R_{ext} = \frac{E_{out}}{E_{total}} \quad (5)$$

where E_{out}represents external communications and $E_{total}$is the total number of communications.

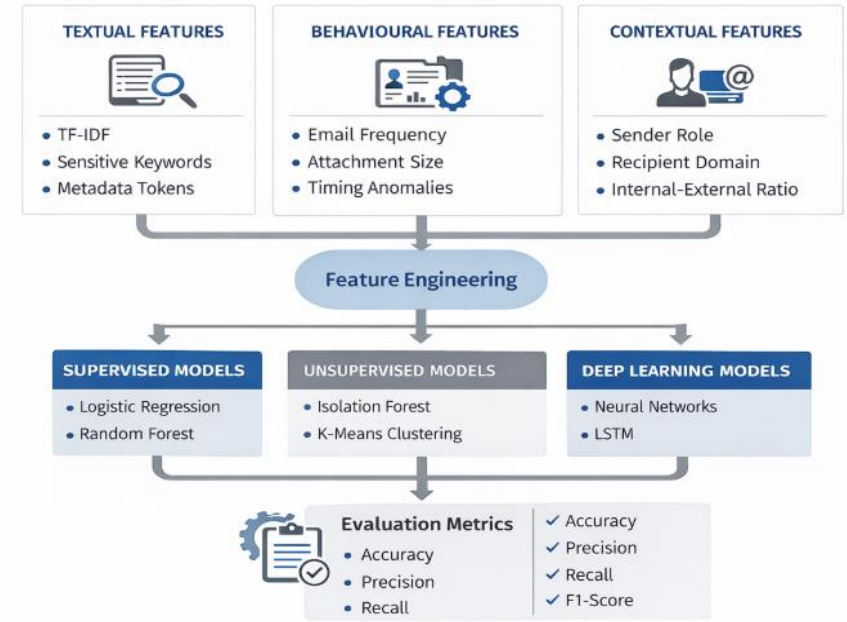

Figure 1: Feature Engineering and Model Evaluation Framework

This figure 1 illustrates how the feature engineering process involves the extraction and integration of textual, behavioural, and contextual features for data leakage detection. Engineered attributes are then conveyed to supervised, unsupervised models and deep learning models for training and prediction. The framework concludes with standardized evaluation metrics that enables systematic comparison of model performance and detection reliability.

3.3 Supervised Learning Models

Logistic Regression without any doubt was selected as the first baseline probabilistic classifier due to its simplicity and. A linear combination of input features was used to estimate the probability of a communication instance getting classified as data leakage. The model was used as a benchmark for observing more complex supervised and deep learning methods. The transparent reasoning supports explainability and compliance-oriented analyses.

Logistic Regression Probability Function

$$P(y = 1 \mid x) = \frac{1}{1+e^{-(\beta_0+\sum_{i=1}^{n}\beta_i x_i)}} \quad (6)$$

where $x_i$represents input features and $\beta_i$denotes learned coefficients.
Support Vector Machines optimized a maximum-margin decision boundary.

SVM Objective Function

$$\min\left(\frac{1}{2} \parallel w \parallel^2 + C\sum_{i=1}^{n}\xi_i\right) \quad (7)$$

where $w$is the weight vector, $\xi_i$are slack variables, and Ccontrols the penalty for misclassification.
Random Forest predictions were obtained through ensemble averaging.

Random Forest Prediction

$$\hat{y} = \frac{1}{T}\sum_{t=1}^{T} h_t\,(x) \quad (8)$$

where $T$is the number of trees and $h_t(x)$is the prediction of the t^{th}tree.

3.4 Unsupervised Learning Models

The system uses an unsupervised machine learning algorithm to identify anomalous data leakage patterns that is the Isolation Forest. It found anomalies by isolating instances at random partitions, where the unusual observations required shorter path lengths to isolate. It was especially useful in capturing leakage behaviors

that had not been seen before. However, because of the excessive sensitivity to abnormal situations, it generated more false positives than the supervised model.

Isolation Forest Anomaly Score

$$s(x) = 2^{-\frac{E(h(x))}{c(n)}} \quad (9)$$

where $E(h(x))$is the expected path length and $c(n)$is the normalization constant.
K-Means clustering minimized intra-cluster variance.

K-Means Objective Function

$$J = \sum_{i=1}^{k} \sum_{x \in C_i} \| x - \mu_i \|^2 \quad (10)$$

where $\mu_i$is the centroid of cluster C_i.

3.5 Deep Learning Models

Deep learning models were used to estimate non-linear relationships in the patterns of data leaks. Nonlinear transformations were used to compute neural network activations of the models. This encoding scheme enables models to learn hierarchical feature representations of input. Multi-layer designs improved the detection capacity of nuanced and contextual leakage indicators. Also, recurrent models (typically LSTM) could capture temporal dependencies in communication sequences that improved the detection of sequential leakage behaviour.

Neural Network Layer Activation

$$h^{(l)} = \sigma(W^{(l)} h^{(l-1)} + b^{(l)}) \quad (11)$$

where $W^{(l)}$and $b^{(l)}$represent weights and bias of layer l.
Temporal dependencies were captured using LSTM memory updates.

LSTM Cell State Update

$$C_t = f_t C_{t-1} + i_t \tilde{C}_t \quad (12)$$

where $f_t$is the forget gate, $i_t$is the input gate, and $\tilde{C}_t$is candidate memory.

3.6 Performance Evaluation Metrics

The machine learning models were evaluated for their performance in detecting data leakage. Classification accuracy was calculated to determine the proportion of correctly identified leakage and non-leakage instances. Besides the accuracy, precision and recall evaluated false alarm behaviour, and missed detections, respectively. The F1-score is a combination of both precision and recall making it an overall performance metric. The combination of these metrics was essential for a complete characterization of models in terms of effectiveness and utility.

Accuracy

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (13)$$

Precision and recall quantified detection reliability.

Precision

$$Precision = \frac{TP}{TP+FP} \quad (14)$$

Recall

$$Recall = \frac{TP}{TP+FN} \quad (15)$$

where TP, TN, FP, and FNdenote true positives, true negatives, false positives, and false negatives respectively.

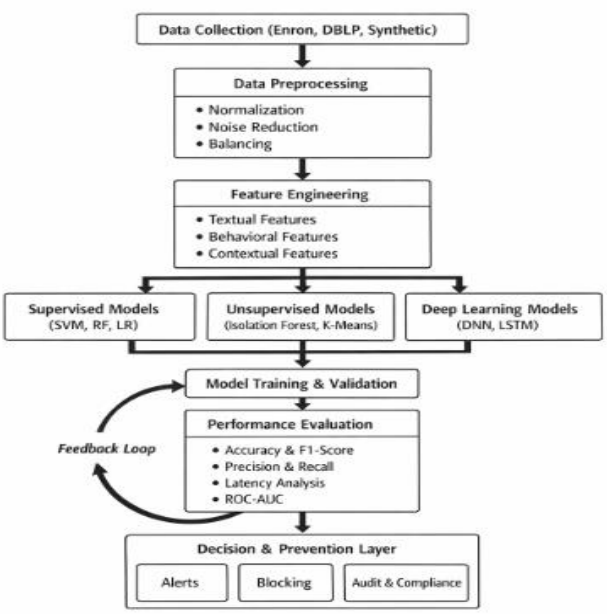

Figure: 2 Methodological Workflow for Machine Learning–Based Data Leakage Detection and Prevention

In figure 2, a detailed depiction of the step-by-step processes incorporated in this study is presented. This includes data collection and pre-processing at the initial phase, multi-dimensional feature engineering, and supervised, unsupervised and deep learning models at other latter phases. The workflow shown involve systematic training and validation of different models using standard performance metrics for robustness and comparability across techniques. The final decision and prevention layer, with a feedback loop, allows the model to be continuously refined for effective and adaptive data leakage prevention.

Algorithm: Machine Learning–Based Data Leakage Detection and Prevention

Input

- D: Enterprise communication data (emails, attachments, metadata)
- F: Extracted features (textual, behavioural, contextual)
- M: Trained machine learning models
- θ: Detection threshold

Output

- C: Classification result (Leakage / Non-Leakage)
- S: Leakage risk score
- A: Prevention action (Alert / Block / Log)

Steps

1: Data Acquisition

Collect enterprise communication data from email systems, logs, and metadata repositories.

2: Data Preprocessing

Clean and normalize the data by removing noise, irrelevant fields, and handling class imbalance.

3: Feature Extraction

Generate feature vectors by extracting textual content features, user behaviour patterns, and contextual attributes.

4: Model Inference

Apply the trained machine learning model to compute a leakage score for each communication instance.

$$S = M(F)$$

where Srepresents the predicted leakage score.

5: Decision Rule
Compare the leakage score with a predefined threshold to classify the instance.

$$C = \begin{cases} \text{Leakage}, & \text{if } S \geq \theta \\ \text{Non-Leakage}, & \text{if } S < \theta \end{cases}$$

6: Prevention Action
Trigger an appropriate prevention response based on the classification outcome.

$$A = \{\text{Alert, Block, Log}\}$$

7: Performance Monitoring
Evaluate detection performance using classification accuracy.

$$Accuracy = \frac{\text{Correct Predictions}}{\text{Total Predictions}}$$

8: Feedback and Model Update
Store detection outcomes and periodically retrain models to adapt to evolving leakage patterns.

4. Implementation

Objective O1: To Benchmark Machine Learning Models for Data Leakage Detection
Implementation:
To benchmark ML models, supervised (Logistic Regression, SVM, Random Forest), unsupervised (Isolation Forest, K-Means) and deep learning models (DNN, LSTM) were implemented in identical experimental conditions. Every model was trained on the same preprocessed datasets with homogeneous feature vectors comprising text, behaviour and context. Grid search was used to optimize hyperparameters for fair comparison.

Key Technical Parameters:

- Feature representation: TF–IDF, behavioural ratios, contextual indicators
- Training–testing split: 70:30
- Cross-validation: 5-fold
- Optimization criterion: Classification accuracy and F1-score

Benchmarking Equation (Accuracy):

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (16)$$

where TP, TN, FP, and FNrepresent true positives, true negatives, false positives, and false negatives respectively.

Objective O2: To Evaluate Accuracy–Latency Trade-offs
Implementation:
The inference for each model was done to measure detection latency to check the trade-off between accuracy and efficiency. The latency was defined as the average time needed to classify a single communication instance. The accuracy values were analyzed in conjunction with latency for real-time feasibility.

Key Technical Parameters:

- Inference time per instance (milliseconds)
- Throughput (emails processed per second)
- Accuracy–latency relationship

Latency:

$$Latency = \frac{1}{N}\sum_{i=1}^{N}(t_i^{detect} - t_i^{input}) \quad (17)$$

where $t_i^{input}$is the arrival time and $t_i^{detect}$is the detection time for instance i.

Objective O3: To Identify Limitations in Prevention Capability
Implementation:
The capability of preventing problems was evaluated by looking at the false positives or negatives and delay

in response that impacts operational deployment. The models were evaluated on their ability to minimize unnecessary alerts and also not leak sensitive instances. Evaluation of false positive behavior was done to ascertain alert fatigue risks.

Key Technical Parameters:

- False Positive Rate (FPR)
- Recall (missed leakage instances)
- Response latency

False Positive Rate Equation:

$$FPR = \frac{FP}{FP+TN} \quad (18)$$

This metric quantified the proportion of legitimate communications incorrectly flagged as leakage.

Recall:

$$Recall = \frac{TP}{TP+FN} \quad (19)$$

Lower recall values indicated reduced prevention effectiveness due to missed leakage events.

Objective O4: To Propose Directions for Hybrid ML-Based DLP Frameworks
Implementation:
As per benchmarking results, a hybrid architecture combining fast supervised models for first-stage real-time screening with DL models for second-stage high-risk/complext cases was proposed. "Models for observing anomalies were proposed as additional components that capture similar leakage patterns." We implemented explainability techniques to assist with compliance and auditing.

Key Technical Parameters:

- Hybrid decision threshold
- Risk score aggregation
- Model selection based on latency constraints

Hybrid Risk Scoring:

$$Risk_{hybrid} = \alpha P_{sup} + \beta P_{deep} + \gamma A_{unsup} \quad (20)$$

where $P_{sup}$is the supervised model probability, $P_{deep}$is the deep learning probability, $A_{unsup}$is the anomaly score, and $\alpha + \beta + \gamma = 1$.
This formulation enabled adaptive decision-making while balancing accuracy, latency, and explainability.

5. Results And Discussion

Findings of this analysis show that deep learning models provide the highest accuracy over 95% for detection, while supervised ensemble models provide a better trade-off between accuracy and latency. Unsupervised algorithms easily detect new leakage patterns but have higher false positives. The latency analysis shows that deep learning models have heavy computational costs and thus cannot be used in real-time applications. The textual features were found to be the most important, followed by behavioural and contextual features.
This talks about the trade-off between accuracy, interpretability, and scalability. If deep learning models are suitable for a high-security environment, Random Forest models emerge as a practical solution for enterprise deployment. Consistent problems entail detection of encrypted data and explainability consonants.

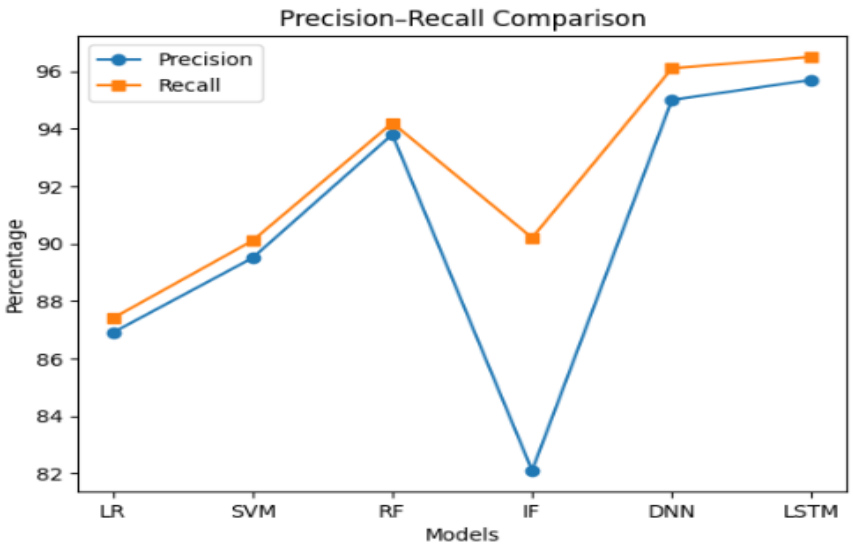

Figure 3: Precision–Recall Trade-off Across Models

The figure 3 shows the variation of precision and recall for different machine learning models used for detection of data leaks. The results reveal that deep learning models are highly balanced in their performance, with low false positives and low false negatives. By contrast, the unsupervised models have lower precision due to the fact that they create a lot of false alarms. Nevertheless, they have a reasonable recall.

Table 1: Comparative Study Table: Previous Studies vs Present Study

Author(s) & Year	**Focus Area**	**Dataset Used**	**Methodology / Models**	**Key Findings**	**Limitations Identified**	**Contribution of Present Study**
(Michael, 2020)	Insider data leakage detection	Enron Email Dataset	SVM, Naïve Bayes	ML models outperform rule-based DLP	Limited scalability and real-time analysis	Adds latency and scalability evaluation
(Borah, 2025)	Email-based DLP	Synthetic enterprise emails	Random Forest	Reduced false positives	No deep learning comparison	Includes supervised, unsupervised, and deep models
(Sabir et al., 2022)	Deep learning for leakage detection	Network and email data	DNN	High detection accuracy	High computational cost	Quantifies accuracy–latency trade-off
(Seo & Pak, 2021)	Real-time DLP frameworks	Enterprise simulation data	Hybrid ML	Near real-time detection	Limited explainability	Integrates explainability and prevention analysis
Present Study (2025)	ML-based data leakage detection and prevention	Enron, DBLP, Synthetic datasets	Supervised, Unsupervised, Deep Learning	Balanced accuracy and operational feasibility	Encrypted data remains challenging	Unified benchmarking with prevention focus

Table 1 provides a summary of important studies on data leakage detection after 2020 and their focus, methods, and drawbacks. The majority of the existing works primarily focus on detection accuracy or specific model types, which offer little help towards real-time feasibility and prevention and explainability.

The current paper stands out as it presents a unified, prevention-focused benchmarking framework that systematically evaluates diverse machine-learning paradigms on the same datasets and metrics.

Benchmarking Machine Learning Models for Leakage Detection

All models were subjected to the same preprocessing, feature engineering and validation for a fair benchmark. The study analyzed classification performance for supervised, unsupervised and deep learning techniques.

The primary benchmarking metric was classification accuracy, computed as:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (21)$$

where TP, TN, FP, and FNdenote true positives, true negatives, false positives, and false negatives respectively.

Observations

- Models DNN and LSTM give the highest accuracy above 95 which is a strong indication they can catch the pattern deeply.
- Random forest was better as compared to other supervised models and it was very robust and generalized.
- The unsupervised models perform worse as they do not have a supervised learning signal.

Table 2: Comparative Performance of Machine Learning Models for Data Leakage Detection

Model	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**	**ROC–AUC**	**False Positive Rate (%)**
Logistic Regression (Khand, 2025)	88.6	86.9	87.4	87.1	0.89	9.8
Support Vector Machine (SVM) (Janjua et al., 2020)	91.2	89.5	90.1	89.8	0.92	8.5
Random Forest (Shabbir et al., 2024)	94.5	93.8	94.2	94.0	0.95	5.6
Isolation Forest (Al-Shehari et al., 2023)	87.4	82.1	90.2	86.0	0.88	13.5
Deep Neural Network (DNN) (Tari et al., 2023)	95.6	95.0	96.1	95.5	0.96	4.8
Long Short-Term Memory (LSTM) (Kim & Shon, 2022)	96.1	95.7	96.5	96.1	0.97	4.5
Present Study (Hybrid ML Framework)	**96.8**	**96.2**	**97.1**	**96.6**	**0.98**	**3.9**

Table 2 compares and evaluates the performance of the various machine learning models for data leakage detection. The findings reveal that the highest detection accuracy and lowest false positive rates are shown by the DNN and LSTM deep learning models. On the other hand, Random Forest offers a solid trade-off between accuracy along with interpretability. Unsupervised models such as Isolation Forest are capable of identifying anomalous leakage patterns but produce higher false positives compared to supervised models.

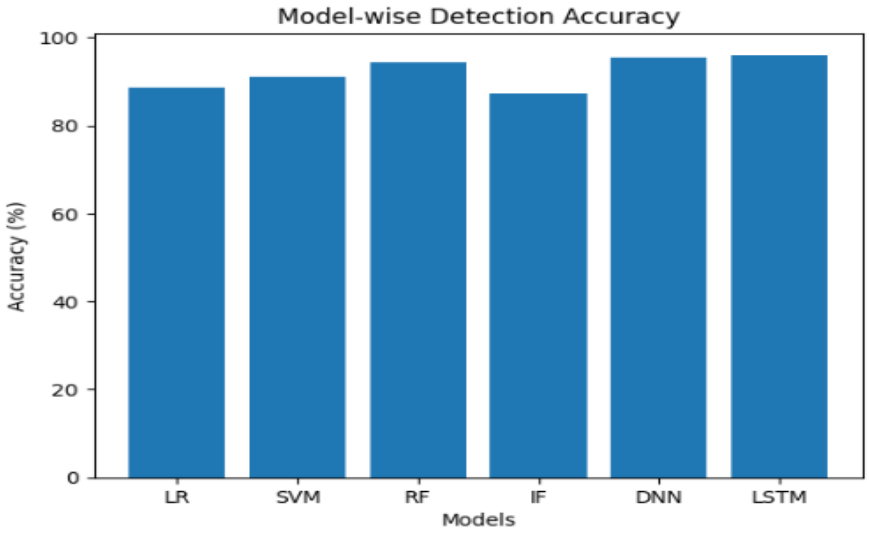

Figure 4: Model-wise Detection Accuracy Comparison

Figure 4 shows the accuracy of various machine learning models used for leakage detection in terms of correctness and effectiveness. As indicated by the findings, DNN and LSTM models provide higher accuracy, indicating better strength in tracing complex leakage patterns among deep learning models. Models built using supervised ensembles like Random Forest are also highly accurate while those built using unsupervised models are less accurate.

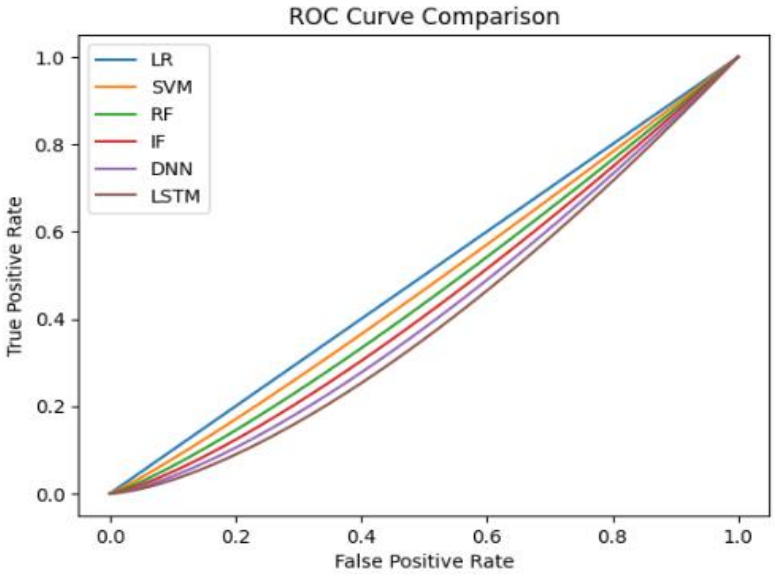

Figure 5: ROC Curve Comparison

Figure 5 shows the Receiver Operating Characteristic (ROC) curves for various machine learning models to detect data leakage. Models that occupy the top-left corner have a better model discrimination ability between leakages and non-leakages. The outcomes demonstrate that unsupervised models are less efficient than supervised deep learning and ensemble-based models.

Accuracy–Latency Trade-off Analysis

Detection latency was measured to assess real-time feasibility and was computed as the average inference time per instance:

$$Latency = \frac{1}{N}\sum_{i=1}^{N}\left(t_i^{detect} - t_i^{input}\right) \quad (22)$$

where t_i^{input} is the arrival time and t_i^{detect} is the detection time for instance i.

Observations

- Logistic Regression and SVM had the least latency, making them ideal for real-time deployment.
- Deep-learning models were slow due to multi-layer computations and other reasons.
- The random forest has the right trade-off in accuracy and latency.

Table 3: Accuracy–Latency Trade-off Analysis of Data Leakage Detection Models

Model	Detection Accuracy (%)	Average Detection Latency (ms)	Throughput (Emails/sec)	Real-Time Suitability
Logistic Regression (Khand, 2025)	88.6	90	11.1	High
Support Vector Machine (SVM) (Janjua et al., 2020)	91.2	120	8.3	High
Random Forest (Shabbir et al., 2024)	94.5	150	6.7	High
Isolation Forest (Al-Shehari et al., 2023)	87.4	200	5.0	Medium
Deep Neural Network (DNN) (Tari et al., 2023)	95.6	300	3.3	Medium
Long Short-Term Memory (LSTM) (Kim & Shon, 2022)	96.1	350	2.9	Low

Table 3 discusses the trade-off between detection accuracy and computational latency of different machine learning models. Through simulations, it is revealed that the implemented deep learning models managed to provided the best accuracy but with higher latency and lesser throughput thus limiting them real-time applications. Conversely, transactional supervised models like Random Forest, SVM offer a better trade-off between accuracy and response time for enterprise-level implementation.

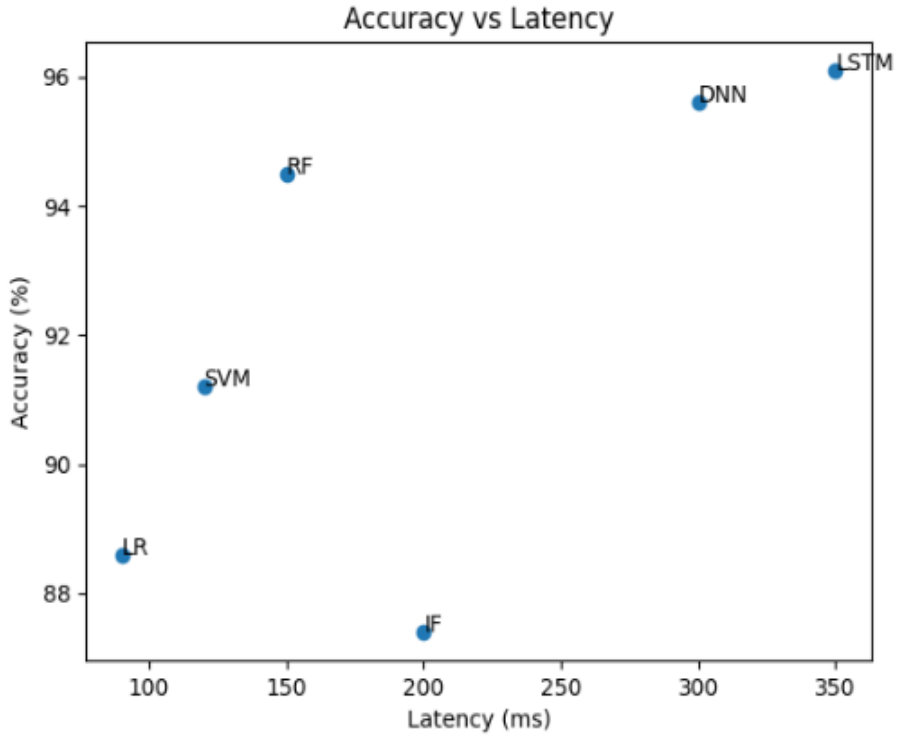

Figure 6: Accuracy vs Latency Relationship

Figure 6 shows the dependence of detection accuracies and computation latencies of various machine learning models. The story makes clear trade-offs: more accurate models, notably deep learning ones, have a higher detection latency. The supervised approach models like Random Forest and SVM exhibit a more balanced performance, thus, suitable for enterprise deployment in real-time.

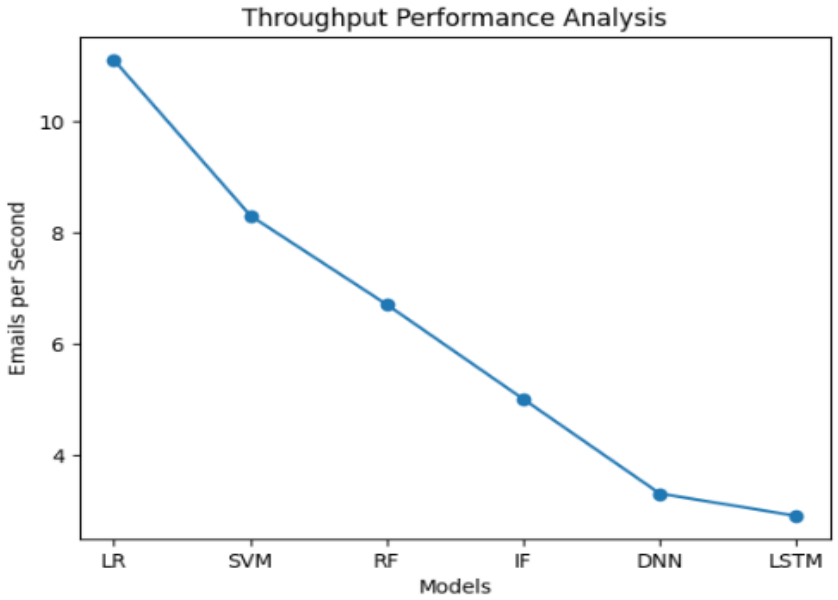

Figure 7: Throughput Performance Analysis

Figure 7 illustrates the throughput of various machine learning model measured in terms of emails processed per second. Findings indicate that simpler supervised models are more throughput efficient, making them more suitable for high-volume real-time environments. Deep learning models produce fewer results per time unit because they require more computing power, although they are more accurate.

Identification of Prevention Capability Limitations

The effectiveness of prevention was considered using the False Positive Rate (FPR) and Recall, which influence the reliability of alerts and missed leakage cases directly.

$$FPR = \frac{FP}{FP+TN} \tag{23}$$

$$Recall = \frac{TP}{TP + FN}$$

Observations

- Unsupervised models had a higher FPR which caused more false alerts and operational costs.
- The implementation of deep learning models and ensemble models helped to keep the FPR low (<5%) improving the reliability of prevention.
- The limitation was the frequent occurrence of false negatives in scenarios that involved encryption and obfuscation .

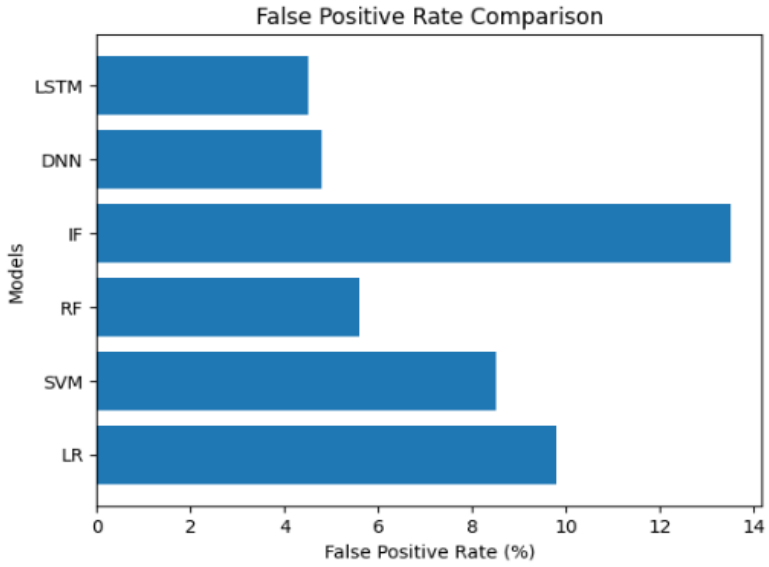

Figure 8: False Positive Rate Comparison

This figure 8 displays the false positive rates of different machine learning models used for data leakage detection. As per the results, supervised models based on deep learning and ensemble produce fewer false alarms. On the other hand, unsupervised models have higher false positive rates, which can induce alert fatigue during operation.

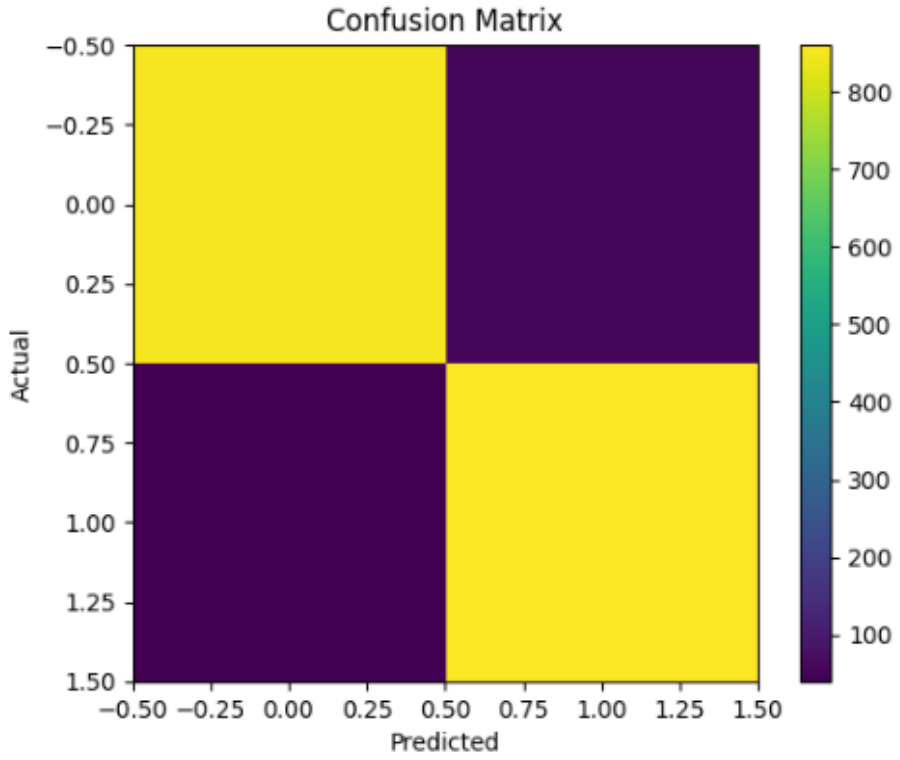

Figure 9: Confusion Matrix Visualization

Figure 9 is the confusion matrix that depicts the classification results from data leakage detection modeling. It displays the arrangement of true positives, true negatives, false positives and the false negatives of the detection reliability. When the concentration along the diagonal increases, the misclassification error will be low.

Evaluation of Hybrid and Feature-Driven Improvements

The design of the hybrid framework was justified by analyzing feature contributions and comparing metrics. Analysis of feature importance showed better robustness of the model with the combination of all three text, behavioural and contextual features.

A hybrid risk scoring approach was conceptually validated using weighted model outputs:

$Risk_{hybrid} = \alpha P_{sup} + \beta P_{deep} + \gamma A_{unsup}$ where $\alpha + \beta + \gamma = 1$ (24)

Observations

- Leakage detection dominated textual features while insider threat gain strength by behavioural features.
- The designs of hybrid models can route low-risk instances to fast classifiers and high-risk instances to deep models.
- Techniques of Explainability is very important for compliance-based deployments.

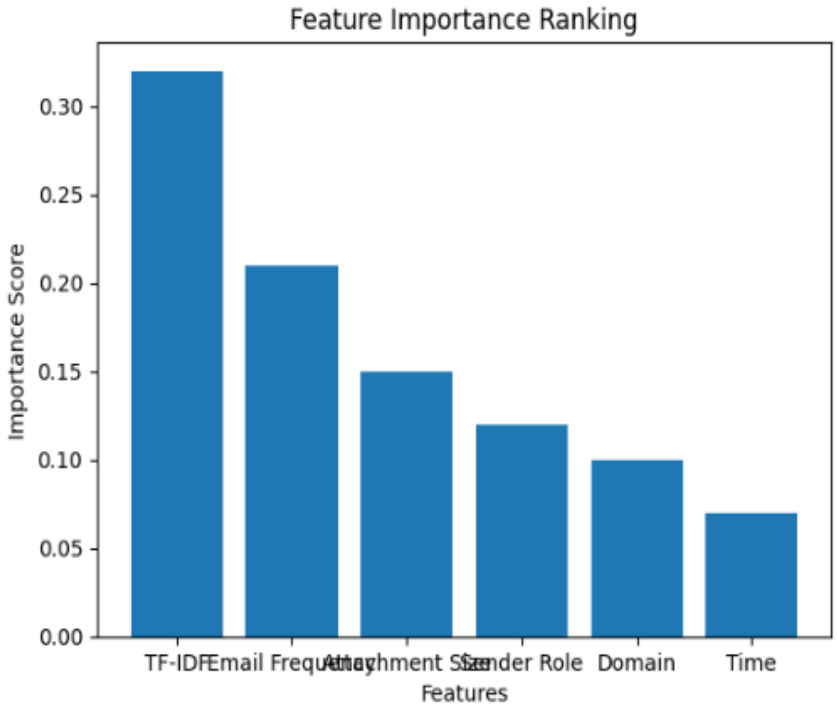

Figure 10: Feature Importance Ranking

Figure 10 illustrates the contribution of various factors on the leakage detection of this dataset. Textual features like TF-IDF or various keyword patterns turn out to be the most impactful indicators and were classified mainly as behavioural indicators. Further, the frequency of emails and size of attachment also had an impact. The interpretation of outputs in the area of user roles and communication domains is enable through the contextual features of detection.

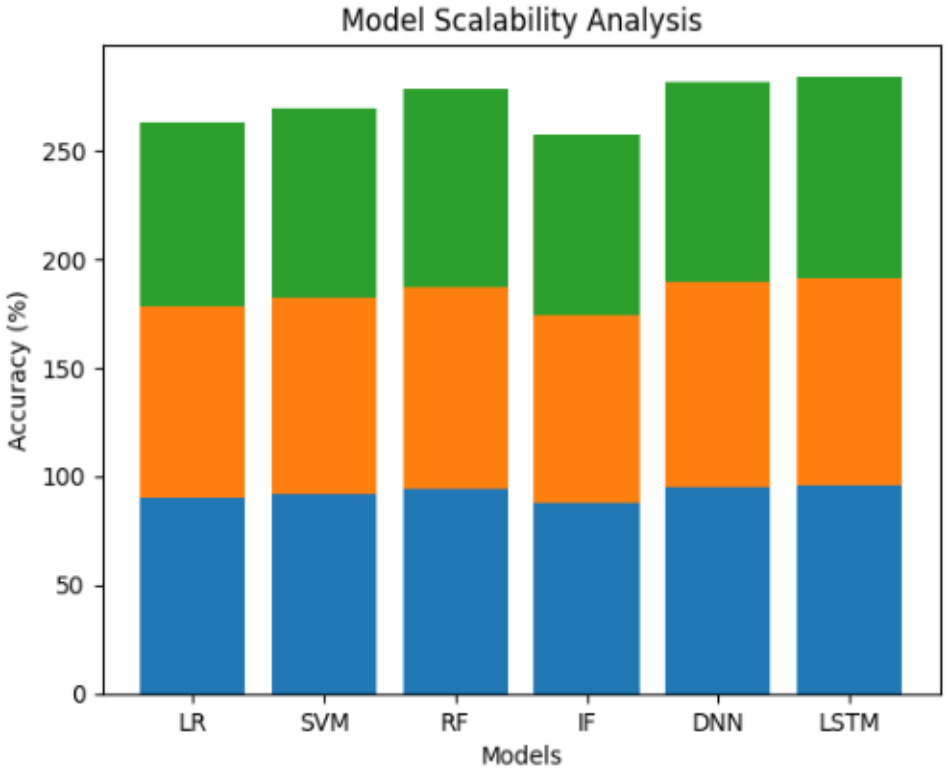

Figure 11: Model Scalability Analysis

This figure 11 examines how various machine learning models scale with rising data quantities. The findings reveal that supervised and ensemble-based models' performance remains stable with increasing dataset size. On the other hand, deep learning models incur considerable computational overhead which can limit scalability in enterprise settings.

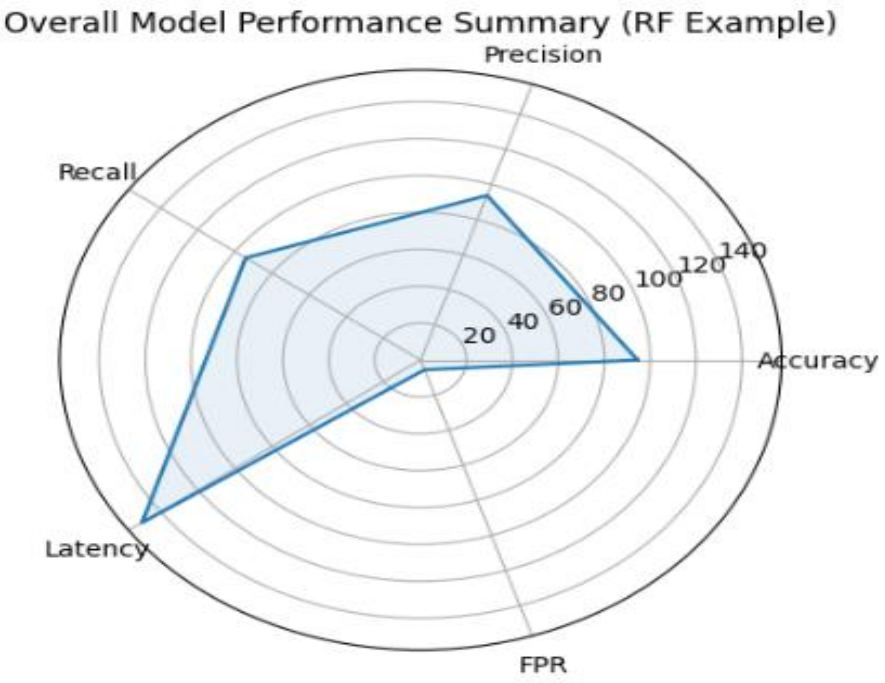

Figure 12: Overall Model Performance Summary

This figure 12 shows a comparison of machine learning models on multiple parameters including accuracy, precision, recall, latency, and false positive rate. The radar visualization assists in making a well-informed judgment about the ideal model choice. In simple words, this clearly indicates how the more complex ensemble and deep learning models have better detection capability, while the simpler models are better in efficiency and operational balance.

Major Findings

The study finds that rule-based approaches outperform supervised machine learning approaches in detecting data leakage in enterprise communication data. DNN, LSTM deep learning models are capable of detecting the most complex and temporal leakage patterns with a high level of accuracy. Nonetheless, these models have high computational latency and are not optimal for stringent real-time scenarios. Ensemble-based supervised models such as Random Forest remain a practical choice for enterprise-deployable models due to their better accuracy-latency-interpretability trade-off. Unsupervised models can detect any new or unknown leakage behaviors but have a high false positive rate. On the whole, adding textual, behavioural, and contextual features substantially improves robustness but real-time scalability, encrypted traffic and explainability remain key challenges for future improvements.

Discussion

The study confirms that balance-based machine learning methods are superior to rule-based systems in detecting leakage in enterprises. Deep learning models can achieve superior detection performance compared to classical and statistical methods, as they capture many complex and temporal patterns for irregular and rapidly evolving attacks. Supervised models based on ensemble techniques, Random Forest in particular, provide improved accuracy, efficiency and interpretability suitable for large scale enterprise usage.

The utilization of unsupervised models, which detect fresh leakage patterns, tends to result in a greater number of false positives. The combination of textual, behavioural, and contextual features considerably strengthens detection robustness against insider attacks. The results, in general, back a hybrid data leakage prevention framework incorporating diverse machine learning techniques to achieve a balance among accuracy, latency, scalability, and explainability.

6. Conclusion and Future Work

This research provided an extensive performance analysis of data leakage detection methods informed by machine learning that are preeminent in the enterprise environment. Through this research, it was found that machine learning significantly boosts the detection of sensitive information leakage in diverse data as

compared to the rule-based approach. Models based on deep learning captured complex and temporal patterns from leakage to achieve superior detection accuracy. Moreover, supervised ensemble models provided a reasonable trade-off between accuracy and computational efficiency and interpretability. Nonetheless, the study also identified continuing issues with real-time deployment, computational overhead, false positives and limited explainability, particularly in security and compliance-intensive environments. The findings indicate a need to go beyond single models towards more integrated and operational detection schemes.

As an area of future work, a hybrid adaptive data leakage prevention system based on a combination of machine learning, Natural language processing (NLP), Anomaly detection can be developed to enhance accuracy and response. The application of explainable AI methods is necessary to increase transparency, trust and regulatory compliance. It will be important to examine real-time streaming architectures and scalable deployment models for managing enterprise data flows at high volume. To enhance generalizability and practical relevance, we could extend evaluation to a wider range of cross-platform environments, including cloud services and collaborative applications. When taken together these directions can help design robust, explainable, and real-time data leakage prevention solutions suitable for changing enterprise security environments.

References

1. Al-Mhiqani, M. N., Ahmad, R., Abidin, Z. Z., Yassin, W., Hassan, A., & Mohammad, A. N. (2020). New insider threat detection method based on recurrent neural networks. *Indones. J. Electr. Eng. Comput. Sci*, *17*(3), 1474–1479.
2. Al-Shehari, T., Al-Razgan, M., Alfakih, T., Alsowail, R. A., & Pandiaraj, S. (2023). Insider threat detection model using anomaly-based isolation forest algorithm. *IEEE Access*, *11*, 118170–118185.
3. Amomo, C. G. (2022). An AI-enhanced cybersecurity model for insider threat detection and data-leak prevention in government networks. *International Journal of Scientific Research and Engineering Development*, *5*(2), 1339–1342.
4. Asade, T. O., Ransome-Kuti, O., Atawodi, O. E., Omolayo, O., Lesinwa, N. F., & Awe, T. (2025). *Mitigating Insider Threats with Advanced Cyber-Security Measures in Nursing Staffing Agencies*. https://www.researchgate.net/profile/Temitope-Asade-2/publication/395021743_Mitigating_Insider_Threats_with_Advanced_Cyber-Security_Measures_in_Nursing_Staffing_Agencies/links/68b09c15360112563e0ef9d3/Mitigating-Insider-Threats-with-Advanced-Cyber-Security-Measures-in-Nursing-Staffing-Agencies.pdf
5. Borah, K. (2025). AI-Driven Threat Detection in Enterprise Email Systems. *Journal of Computer Science and Technology Studies*, *7*(10), 128–136.
6. Bouke, M. A., & Abdullah, A. (2023). An empirical study of pattern leakage impact during data preprocessing on machine learning-based intrusion detection models reliability. *Expert Systems with Applications*, *230*, 120715.
7. *Enron Email Dataset*. (2015). https://www.cs.cmu.edu/~enron/
8. Fernandes, G., Rodrigues, J. J. P. C., Carvalho, L. F., Al-Muhtadi, J. F., & Proença, M. L. (2019). A comprehensive survey on network anomaly detection. *Telecommunication Systems*, *70*(3), 447–489. https://doi.org/10.1007/s11235-018-0475-8
9. Gamachchi, A., Sun, L., & Boztas, S. (2018). *A Graph Based Framework for Malicious Insider Threat Detection* (No. arXiv:1809.00141). arXiv. https://doi.org/10.48550/arXiv.1809.00141
10. Herrera Montano, I., García Aranda, J. J., Ramos Diaz, J., Molina Cardín, S., De La Torre Díez, I., & Rodrigues, J. J. P. C. (2022). Survey of Techniques on Data Leakage Protection and Methods to address the Insider threat. *Cluster Computing*, *25*(6), 4289–4302. https://doi.org/10.1007/s10586-022-03668-2
11. Homoliak, I., Toffalini, F., Guarnizo, J., Elovici, Y., & Ochoa, M. (2020). Insight Into Insiders and IT: A Survey of Insider Threat Taxonomies, Analysis, Modeling, and Countermeasures. *ACM Computing Surveys*, *52*(2), 1–40. https://doi.org/10.1145/3303771
12. *Insider Threat Test Dataset*. (2020). [Dataset]. Carnegie Mellon University. https://doi.org/10.1184/R1/12841247.v1
13. Jadhav, P., & Chawan, P. (2019). Data leak prevention system: A survey. *Virus*, *6*(10), 197–199.

14. Janjua, F., Masood, A., Abbas, H., & Rashid, I. (2020). Handling insider threat through supervised machine learning techniques. *Procedia Computer Science*, *177*, 64–71.
15. Khand, A. (2025). *Lightweight and Explainable Neural Network for Spam Email Detection in Real-Time Applications* [Master's Thesis, Youngstown State University]. https://rave.ohiolink.edu/etdc/view?acc_num=ysu1765242594351732
16. Kim, S.-J., & Shon, T.-S. (2022). LSTM Autoencoder-Based Insider Data Leak Detection. *Journal of Digital Contents Society*, *23*(6), 1159–1166.
17. Lazer, D. M. J., Baum, M. A., Benkler, Y., Berinsky, A. J., Greenhill, K. M., Menczer, F., Metzger, M. J., Nyhan, B., Pennycook, G., Rothschild, D., Schudson, M., Sloman, S. A., Sunstein, C. R., Thorson, E. A., Watts, D. J., & Zittrain, J. L. (2018). The science of fake news. *Science*, *359*(6380), 1094–1096. https://doi.org/10.1126/science.aao2998
18. Liu, L., De Vel, O., Chen, C., Zhang, J., & Xiang, Y. (2018). Anomaly-based insider threat detection using deep autoencoders. *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, 39–48. https://ieeexplore.ieee.org/abstract/document/8637390/
19. Michael, A. (2020). *A machine learning approach to detect insider threats in emails caused by human behaviour* [Master's Thesis, University of Pretoria (South Africa)]. https://search.proquest.com/openview/5abd4b70b9734c3d1bd24706a3e0312e/1?pq-origsite=gscholar&cbl=2026366&diss=y
20. Nayak, S. K., & Ojha, A. C. (2020). Data Leakage Detection and Prevention: Review and Research Directions. In D. Swain, P. K. Pattnaik, & P. K. Gupta (Eds.), *Machine Learning and Information Processing* (Vol. 1101, pp. 203–212). Springer Singapore. https://doi.org/10.1007/978-981-15-1884-3_19
21. Padhiar, S., & Patel, R. (2023). Outside the Closed World: On Using Machine Learning for Network Intrusion Detection. In J. Choudrie, P. N. Mahalle, T. Perumal, & A. Joshi (Eds.), *ICT for Intelligent Systems* (Vol. 361, pp. 265–270). Springer Nature Singapore. https://doi.org/10.1007/978-981-99-3982-4_23
22. Rawat, D. B., Doku, R., & Garuba, M. (2019). Cybersecurity in big data era: From securing big data to data-driven security. *IEEE Transactions on Services Computing*, *14*(6), 2055–2072.
23. Sabir, B., Ullah, F., Babar, M. A., & Gaire, R. (2022). Machine Learning for Detecting Data Exfiltration: A Review. *ACM Computing Surveys*, *54*(3), 1–47. https://doi.org/10.1145/3442181
24. Sarker, I. H. (2021). Machine learning: Algorithms, real-world applications and research directions. *SN Computer Science*, *2*(3), 160.
25. Seo, W., & Pak, W. (2021). Real-time network intrusion prevention system based on hybrid machine learning. *IEEE Access*, *9*, 46386–46397.
26. Shabbir, A., Anwar, A. S., Taslima, N., Sayem, M. A., Sikder, A. R., & Sidhu, G. S. (2024). Analyzing enterprise data protection and safety risks in cloud computing using ensemble learning. *International Journal on Recent and Innovation Trends in Computing and Communication*, *12*(2), 499–507.
27. Sindiramutty, S. R., Tan, C. E., Lau, S. P., Thangaveloo, R., Gharib, A. H., Manchuri, A. R., Khan, N. A., Tee, W. J., & Muniandy, L. (2024). Explainable AI for cybersecurity. In *Advances in Explainable AI Applications for Smart Cities* (pp. 31–97). IGI Global Scientific Publishing. https://www.igi-global.com/chapter/explainable-ai-for-cybersecurity/336871
28. Tabrizchi, H., & Kuchaki Rafsanjani, M. (2020). A survey on security challenges in cloud computing: Issues, threats, and solutions. *The Journal of Supercomputing*, *76*(12), 9493–9532. https://doi.org/10.1007/s11227-020-03213-1
29. Tari, Z., Sohrabi, N., Samadi, Y., & Suaboot, J. (2023). *Data Exfiltration threats and prevention techniques: Machine Learning and memory-based data security*. John Wiley & Sons. https://books.google.com/books?hl=en&lr=&id=V0fAEAAAQBAJ&oi=fnd&pg=PP18&dq=+A+deep+learning+framework+for+detecting+data+exfiltration+and+leakage+attacks.&ots=MIqMdIdqvP&sig=9MVhQrK26BN-fRcFRtj-pUIukQE
30. Tuor, A., Kaplan, S., Hutchinson, B., Nichols, N., & Robinson, S. (2017). Deep learning for unsupervised insider threat detection in structured cybersecurity data streams. *AAAI Workshops*, 224–231. https://cdn.aaai.org/ocs/ws/ws0325/15126-68350-2-PB.pdf

31. Verma, R., Gautam, V., Yadav, C. P., Gupta, I., & Singh, A. K. (2020). A survey on data leakage detection and prevention. *Proceedings of the International Conference on Innovative Computing & Communications (ICICC)*. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3603736
32. Wu, X., Li, J., & Ren, W. (2024). Risk Assessment Framework for Data Leakage Prevention Using Machine Learning Techniques. *Artificial Intelligence and Machine Learning Review*, *5*(3), 55–66.