National College of Ireland

A Comparative study of ML Models for Data Loss Prevention

M.Sc. ResearchProject
M.Sc. in Cybersecurity

Srikari Surampudi
StudentID:23178485

School of Computing
National College of Ireland

Supervisor: Dr. Imran Khan

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Srikari Surampudi
Student ID:	23178485
Programme:	M.Sc. in Cybersecurity
Year:	2024
Module:	M.Sc. Research Project (Practicum part 2)
Supervisor:	Dr.Imran Khan
Submission Due Date:	12/12/2024
Project Title:	A Comparative study of ML models for data loss prevention
Word Count:	7324
Page Count:	22

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Srikari Surampudi
Date:	12th December 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

A Comparative Study of ML models for Data Loss Prevention

Srikari Surampudi
23178485

Abstract

Data Loss Prevention (DLP) is vital for the protection of exclusive information for organizations against leakage and unauthorized access. A key limitation of conventional DLP systems is their inability to effectively identify complex data loss events amidst the vast amount of cyber threat data. The following work focuses on comparing different Machine Learning (ML) models for DLP solutions. Utilizing a comprehensive dataset comprising 40,000 records of network traffic and attack characteristics, we implemented and evaluated ML models like SVM, K-Means clustering, Random forests, Logistic Regression and K-Neural networks. The data preprocessing steps involved were feature cleaning such as missing value handling, categorical encoding, feature creation and synthetic data augmentation by SMOTE technique. Further, data augmentation was carried out through adding Gaussian noise to achieve better generalization architecture. The assessment, indicated that the Random Forest model was far more effective than the other models we investigated, including the SVM, K-means clustering, Logistic Regression model and the Neural Networks model; after we hyperparameter tuned the Random Forest model, its accuracy was 87.0 % while that of the other models was approximately 34 % for the same features. The parameters of the model also reflected Random Forest's high level of accuracy: the ROC-AUC score was 0.97, hence the model excels at distinguishing between various classes of data loss incidents. These results further apply ensemble learning methods as valuable in the improvement of DLP systems and providing solid foundation from which to detect data loss attempts.

1 Introduction

In the digital era, data protection for personal and organizational is a growing challenge. With so many cyber threats, Data Loss Prevention is indispensable for securing certain information. Cases of data loss are by far the most detrimental to the financial, operational and reputational costs they can have. DLP refers to measures that comprise methodology and technologies aimed at the prevention of unauthorized access, use, and transmission of identifiable information. Traditional DLP systems struggle to effectively address the complex and dynamic nature of modern data loss threats, which are worsened by the growing volume and variety of data in cyber environments.

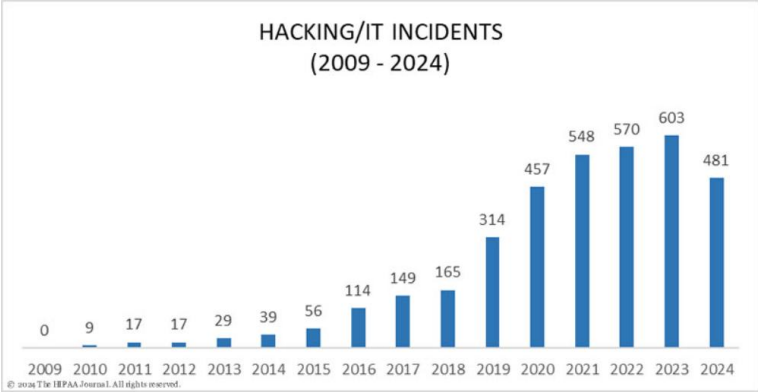

Figure 1: Data Loss Incidents Trend over years [1]

As illustrated in Figure 1, the rise in data loss incidents underscores the urgent need for more robust and intelligent prevention mechanisms.

The ever-increasing frequency and complexity of data loss threats require fundamentally higher levels of protection that would be able to prevent the threats by using new strategies. ML is useful in this context by providing a means through which such systems can be built: the systems are fed data, and they learn, decide, and predict potential data loss activities. The incorporation of ML in DLP framework enables the intelligence-led detection and response to the listed risks, in turn strengthening the security response and measurement. [2]

Still, there is a series of issues that hinder the proper implementation of ML in the framework of data loss prevention. These are, for example, the skewed dataset problem, requirements for online/real-time data analysis, and the issue of transparency of the Machine Learning models. Also, as the approaches used in data loss change constantly, constant modification in the prevention models is necessary. It is important to solve those challenges to enhance the dependability and resiliency of ML-based DLP systems. [3]

This paper compares various ML models for DLP solutions, evaluating their effectiveness in detecting complex data loss incidents. The primary research questions guiding this investigation are:

1. Which ML models demonstrate the best performance in identifying potential data loss incidents?
2. How do different feature selection and data preprocessing techniques impact the accuracy and efficiency of these models?

To address these questions, the study aims to:

1. Preprocess and analyze a comprehensive dataset of data loss incidents,
2. Implement and train multiple ML models, and
3. Evaluate their performance using appropriate metrics.

The present research advances the scientific body of knowledge in two ways. To start with, it highlights the weaknesses or performance of diverse methods of ML in the field of data loss prevention as well as a comparison to their efficiency. Secondly, it investigates relationships of different data preprocessing and feature selection techniques with the effectiveness of ML models, which in turn helps in improving the DLP systems. [4]

The remainder of this report is structured as follows: Section 2 reviews the existing literature on ML-based data loss prevention. Section 5 details the data preprocessing, feature engineering, and model implementation processes. Section 6 presents the evaluation metrics

and results of the comparative analysis. Finally, Section 7 concludes the report by summarizing the findings and suggesting directions for future research.

2 Related Work

DLP is still one of the most significant issues in the cybersecurity domain as novel and highly evolved threats aim at personal and company data assets. In the era where organizations lean more towards digital solutions, data exfiltration has emerged as a concern. Therefore, the adoption of Machine Learning (ML) has come out as a strong enabler in improving the resilience of DLP systems. These systems employ several ML strategies to enhance the tools' ability to detect multiple threats effectively, diminish the rate of false alarms and manage tremendous amounts of data. Thus, this section summarizes the existing literature on ML based DLP, with attention to the supervised and unsupervised learning techniques used, recent developments in deep learning, and techniques used for data preprocessing, feature selection, and data balancing. Furthermore, this work locates itself in the discourse as it examines novel ML models with improved detecting performance that confront issues throughout the labeled data deficiency and computational-load concerns.

2.1 Machine Learning Techniques for Data Loss Prevention

2.1.1 Supervised Learning Approaches

Supervised learning has proven to be one of the most effective techniques in identifying data exfiltration activities, as it relies on labeled data to detect specific patterns associated with malicious behavior. These techniques use classifiers trained on historical data to predict the likelihood of an incident occurring. For instance, **Sahingoz et al. (2021)** applied supervised learning models such as **Random Forest**, **Support Vector Machines (SVM)**, and **k-Nearest Neighbors (k-NN)** to detect potential data exfiltration activities. Their work primarily focused on analyzing lexical and host features derived from user traffic. The study showed that **Random Forest** outperformed the other models in terms of both accuracy and detection rate, highlighting its robustness in handling diverse datasets. [5]

Similarly, **Hussain Alattas, et al. (2022)** employed **Naïve Bayes** and **Decision Trees** to detect insider threats based on behavioral features. They found that **Naïve Bayes** achieved higher accuracy and lower false positive rates compared to **Decision Trees**, making it a promising approach for behavior-based analysis in DLP systems. These studies underline the importance of selecting relevant features to optimize the performance of supervised learning models in the detection of data loss incidents. [6]

2.1.2 Unsupervised Learning Approaches

Non-supervised learning models are indispensably important in the design of DLP systems because in many cases, labeled data is difficult to come by or is completely absent. These models operate based on two principles – they seek to find patterns or outliers that can be produced without reference to predefined labels. One of the methods often utilized is **K-means clustering**, which gathers the material into groups by similarity and considers objects that do not fit into any cluster as threats, **Canali et al. (2011)** reported on the possibility of using unsupervised models for identifying malicious user activity through **K-means clustering**.

Some of the weaknesses inherent in the semi-supervised learning techniques are that even though the approach does not require labelled data, it has a high false positive rate and the problem of distinguishing between normal and anomalous data, anomalous behavior. Nevertheless, methods of this type offer certain benefits in terms of recognizing previously unidentified threats, and they can enhance learnable models by increasing the precision of the results where there is little data for labelling the training sets. [7]

2.2 Advancements in Deep Learning for Data Loss Prevention

Deep learning has brought a tremendous amount of change within the area of DLP systems through the development of models that can learn features from the data set on their own. CNNs and RNNs are found to perform well with unstructured data and temporal sequences respectively in literature. These models do not only enhance the techniques of detection of the tumor but also eliminate the problem relating to feature engineering which is tiresome and sometimes involves a lot of errors.

2.2.1 Convolutional Neural Networks (CNNs)

Unlike traditional neural networks that have been initially developed for image recognition, the CNN method has also been used in DLP applications, especially in network traffic data. **Alotaibi et al. (2018)** extend the analysis of raw network packets and detect data exfiltration activities. While the traditional approaches used separate procedures for data preprocessing and application of predetermined heuristics, the CNN models allow to learn the relevant features starting with the data as raw as possible, thus providing an end-to-end perspective on detection of unauthorized data transfers. Their approach indicated relatively high precision in detection of data loss events but still it is highly computational and needs a lot of resources. CNNs capability in feature learning from raw data through hierarchal feature representation makes it suitable especially in real-time application such as DLP in large network traffic analysis. [8]

2.2.2 Recurrent Neural Networks (RNNs)

Recurrent Neural Networks (RNNs), and particularly **Long Short-Term Memory (LSTM)** networks, have shown great promise in detecting data loss events that involve temporal sequences, such as user behavior patterns over time. **Su, Y. (2020)** applied LSTMs to analyze user logs and discovered that LSTMs were highly effective at capturing the temporal dependencies of user behavior. This is particularly important for identifying data loss incidents that unfold over time, as LSTMs can model long-term dependencies in sequential data. By understanding the context and sequence of events, LSTMs offer an advantage over traditional methods that might overlook important temporal aspects. This makes LSTMs particularly well-suited for detecting sophisticated data loss activities that involve multiple stages or occur over extended periods. [9]

2.3 Summary of Results

The situational analysis of main studies on use of machine learning in DLP techniques is given in the table below along with detected advantages and disadvantages of the methods as well as evaluation of their effectiveness. These studies show variations of DLP programs and show how each approach has benefits and drawbacks. The literature review highlights that supervised methods like Random Forest excel in handling diverse datasets, while unsupervised approaches like K-Means are limited by high false-positive rates. Advanced techniques such as CNNs and

LSTMs show promise in real-time data loss detection but require significant computational resources and large datasets for effective training.

Table 1: Summary of Data Loss Prevention Techniques

Technique	Key Findings	Accuracy	Limitations
Random Forest	High accuracy, robust to overfitting risk [5]	95.6	Requires extensive feature engineering and careful tuning
Naïve Bayes	Effective for behavior-based analysis [6]	92.3	Sensitive to feature correlations and may perform poorly with highly dimensional data
K-means Clustering	Detects anomalies in user behavior [7]	85.0	Prone to high false positive rates and may struggle with overlapping clusters
CNN	Excels in feature extraction from raw network data [8]	94.8	Computationally intensive, requiring significant processing power
LSTM	Captures temporal patterns in user activities [9]	93.5	Susceptible to vanishing gradients and requires large amounts of data for effective training

3 Methodology

This section describes the systematic process embarked on to investigate the applicability of different ML models on improving DLP in cyber security realm. It includes data collection and cleaning, variable transformation, data sampling, data generation, data partitioning, feature standardization, model estimation and assessment. All the stages are elaborated quite thoroughly to allow for the high reproducibility and reliability of the research.

3.1 Data Collection

The data set used within this research is named **cybersecurity attacks.csv** and contains 40, 000 records and 25 features. This involves a comprehensive list of attributes on the network traffic and attacks such as source and destination IP addresses, protocol, port numbers, packet size, payload, malware footprints, anomaly scores, types of attacks, and geo-location data. The Timestamp feature stipulates the day and time of occurrence of each attack as phase information.

3.2 Data Preprocessing

Effective data preprocessing is critical to ensure the quality and suitability of the dataset for subsequent analysis. The preprocessing steps are as follows:

3.2.1 Handling Missing Values

Initial exploration identified missing values in several features: Malware Indicators,

Alerts/Warnings, Proxy Information, Firewall Logs, and IDS/IPS Alerts. These missing values were imputed with zeros to maintain dataset integrity and prevent biases during model training. Post-imputation verification confirmed the absence of any remaining missing values.

3.2.2 Encoding Categorical Variables

Categorical features, including Protocol, Packet Type, Traffic Type, Attack Type, Attack Signature, Action Taken, Severity Level, User Information, Device Information, Network Segment, Geo-location Data, and Log Source, were encoded using Label Encoding. This process involved transforming categorical labels into numerical values, facilitating their use in ML models.

3.3 Feature Engineering

Additional temporal features were extracted from the Timestamp attribute to enhance the predictive power of the models. Specifically, Year, Month, and Weekday were derived to capture temporal patterns in the attack data. Furthermore, IP addresses were converted to their integer representations using the ipaddress library, enabling numerical analysis of network-related features.

3.4 Data Balancing

Another limitation of the phishing detection datasets is class imbalance in which some attack types can be abundant while others are scarce. To counter this, Synthetic Minority Over-sampling Technique was applied in this research. Unlike other approaches, SMOTE synthesizes samples for minority classes, so that one has equal probability of choosing each of the four types of attacks. This balancing was important for controlling the risk of the models favoring the majority classes as well as for the general improvement of the classifiers' reliability.

3.5 Data Augmentation

Furthermore, to increase the training set variability and reduce overfitting risk, data augmentation was applied to all numerical fields using Gaussian noise addition. This selective noise addition disturbs the originals slightly and produces a more complex set of inputs without much effect on distribution. By augmentation, I was able to multiply the number of samples by two, which enriched arrays of examples for the models.

3.6 Dataset Splitting

The augmented dataset was split into training and testing using methodology of stratified train-test split. In detail, the training set was used with 80% of overall data, and the testing set with the rest, which is 20%. Class distribution was maintained within both sets through grouping so that the ability to generalize the chosen ML models was improved.

3.7 Feature Scaling

Feature scaling was done by the standard Scaler to bring the scale of the numerical features into standard scale. This method of scaling was used on the training as well as on the testing dataset to preserve comparability. This normalization process places all features on the same scale, where each feature has a mean value equal to zero and standard deviation of the feature equal to one. This process is critical for models like SVM and neural networks, which are sensitive to feature magnitudes, ensuring that no single feature disproportionately influences the model.

3.8 Model Training

Multiple ML models were selected based on their relevance and effectiveness in classification tasks. The models implemented include:

3.8.1 Random Forest

Random Forest, which is an ensemble learning method, which builds several decision trees during the training step, was selected for its great performance for several reasons. In this study, Random Forest was set with 100 trees and maximum depth of tree as 10 to achieve decent complexity with acceptable time cost to run the model.

3.8.2 Logistic Regression

A traditional linear model, Logistic Regression, was used as a starting point in the study. This model is chosen from the stable of classification models due to its ease of interpretability. The Logistic Regression model was done using scikit-learn native module to generate the model using the default features of the model such as a regularization strength of 1.0 and a solver of 'liblinear' which fits small to medium datasets and multi-class classification problems.

3.8.3 Neural Networks

A deep learning approach was adopted using a Sequential model with multiple dense layers and dropout regularization to prevent overfitting risk. The architecture comprised:

- An input layer with 64 neurons and ReLU activation.
- A dropout layer with a rate of 0.5.
- A hidden layer with 32 neurons and ReLU activation.
- A dropout layer with a rate of 0.3.
- An output layer with softmax activation for multi-class classification.

The model was compiled using the Adam optimizer and categorical cross-entropy loss function. Training was conducted over 20 epochs with a batch size of 256, incorporating a validation split of 20% to monitor performance.

3.9 Evaluation Methodology

To assess performance of ML models, comprehensive set of evaluation metrics was employed:

- **Accuracy**: Measures the overall correctness of the model in predicting both phishing and legitimate instances.
- **Precision**: Evaluates the proportion of correctly identified phishing instances out of all instances predicted as phishing.
- **Recall (Sensitivity)**: Assesses the model's ability to correctly identify all actual phishing instances.
- **F1-Score**: Offers a harmonic mean of precision and recall, that is, a balanced measure of how well model performs.
- **Confusion Matrix**: A detailed discussion of true positives, true negatives, false positives, and false negatives.
- **ROC-AUC Score**: It estimates the model's capacity at discriminating classes given the different setting of the threshold.

These metrics provide a holistic view of the models' performance, enabling a nuanced comparison of their strengths and weaknesses in the context of phishing detection.

The above presented methodology provides a framework for the evaluation of various ML models for DLP processes. Precision, recall, and ROC-AUC were chosen as key evaluation metrics to address the imbalanced nature of the dataset, where false negatives (missed detections) and false positives (false alarms) have critical implications. Precision highlights the model's ability to avoid false alarms, while recall ensures all actual data loss incidents are identified. The ROC-AUC provides a holistic view of the model's discrimination capability across varying thresholds, ensuring robust evaluation.

While performing the steps of preprocessing data, feature selection, data balancing and use of Multiple models the study minimizes variations and maximized reliability. Logistic Regression was added as third comparison to Random Forest and Neural Networks to illustrate effectiveness of ensemble methods for cybersecurity data modeling.

4 Design Specification

This section details the design and implementation of the proposed system for Data Loss Prevention (DLP) enhancement. Five models were developed and evaluated: Neural Network, Logistic Regression, Random Forest, Support Vector Machine (SVM), and K-Means clustering.

4.1 System Architecture

The system architecture is composed of the following components:

- **Data Preprocessing**: Includes handling missing values, encoding categorical features, and scaling numerical features.
- **Model Development**: Five models were implemented to predict attack types:
  1. Neural Network.
  2. Logistic Regression.
  3. Random Forest.
  4. Support Vector Machine (SVM).
  5. K-Means.
- **Model Evaluation**: All models were evaluated on accuracy, precision, recall, and ROC-AUC scores.

4.2 Model 1: Neural Network

For multi class classification the Neural Network (NN) with this architecture is designed. The hyperparameters for the layers, such as the number of neurons and dropout rates, were chosen experimentally to balance model complexity and performance:

1. **Input Layer**: Accepts *n* scaled features.
2. **Hidden Layers**:
   - First hidden layer: 64 neurons with ReLU activation.
   - Dropout layer: Rate of 0.5.
   - Second hidden layer: 32 neurons with ReLU activation.
   - Dropout layer: Rate of 0.3.
3. **Output Layer**: Uses the softmax activation function for multi-class classification: [10]

$$z_i) = \frac{e}{\sum_{j=1}^{k} e^{z_j}}$$

where z is the vector of raw outputs from the neural network, The i-th entry in the softmax output vector softmax(z) can be thought of as the predicted probability of the test input belonging to class "i" and *k* is the number of classes.

The model is trained using the Adam optimizer and categorical cross-entropy loss: [11]

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} y_{i,k}\log(\hat{y}_{i,k}),$$

where $y_{i,k}$ is the true label and $\hat{y}_{i,k}$ is the predicted probability for class *k*.

4.3 Model 2: Logistic Regression

Logistic Regression is employed as a baseline for comparison:

- **Objective**: Estimates the probability of class membership using the sigmoid function: [12]

$$P(y=1|X) = \frac{1}{1+e^{-(\beta_0+\beta_1 X_1+\cdots+\beta_n X_n)}},$$

where β_0 is the intercept and β_i are the feature coefficients.

- **Training**: Optimizes the log-loss function: [12]

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}[y_i\log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)]$$

where n is the number of samples, indexed by i, y_i is the true class for index i and $\hat{y}_i$ is the model prediction for the index i. Regularization techniques (e.g., L2 penalty) are applied to prevent overfitting risk.

4.4 Model 3: Random Forest

The Random Forest model is designed for high performance by leveraging an ensemble of decision trees:

- **Structure**: Comprises *M* decision trees, where each tree is trained on a bootstrap sample of the data.
- **Prediction**: Combines predictions from individual trees using majority voting for classification: [13]

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(fi - yi)^2$$

where N is the number of data points, fi is the value retured by the model and yi is the actual value for data point i.

- **Feature Importance**: Evaluates the significance of each feature by measuring the decrease in impurity: [13]

$$Gini = 1 - \sum_{i=1}^{C}(p_i)^2$$

where p_i is the proportion of class i in the dataset. c is the total number of classes.

A balance was achieved between performance and computational efficiency by selecting hyperparameters like number of trees (100) and maximum depth (10), iteratively by experimentation. Later, these parameters were tuned to increase the accuracy of the model from 85.0% to 87.0%.

4.5 Model 4: Support Vector Machine (SVM)

The Support Vector Machine (SVM) is used for classification with a linear kernel or radial basis function (RBF) kernel, depending on complexity of the data:

- **Objective**: SVM aims to find the line that maximizes margin between the classes. The decision rule for SVM is given by: [14]

$$\hat{y} = \text{sign}(w^T x + b),$$

where w is the weight vector, x is the feature vector, and b is the bias term.

- **Training**: The SVM is trained by solving the following optimization problem: [14]

$$\min_{w,b} \frac{1}{2}||w||^2 \quad \text{subject to} \quad y_i(w^T x_i + b) \geq 1, \quad \forall i.$$

The penalty for misclassification is controlled by the regularization parameter C.

In the case of non-linear classification, the kernel trick is used to map the data into a higher-dimensional space.

4.6 Model 5: K-Means Clustering

K-Means clustering is a popular unsupervised machine learning model used for grouping data points into k clusters:

- **Objective**: The task is to find the optimal number of clusters (k) with Elbow Method, that means when calculating the within-cluster sum of squares (WCSS) and finding the point with the dimensional return. This ensured that the chosen k effectively balanced cluster separation without overfitting [15]

$$\sum_{i=0}^{n} \min_{\mu_j \in C}(||x_i - \mu_j||^2)$$

The k-means algorithm divides a set of N samples X into K disjoint clusters C, each described by the mean u_j of the samples in the cluster.

- **Training**: K-Means iteratively assigns each data point to the nearest centroid and updates the centroids based on the mean of the assigned points until convergence.
- **Cluster Assignment**: Each point is assigned to the cluster that minimizes the Euclidean distance to the cluster centroid: [16]

$$\hat{y}_i = \underset{c}{\text{argmin}} ||x_i - \mu_c||^2$$

K-Means does not require labeled data and is therefore an unsupervised method. The number of clusters, k, is a hyperparameter chosen by the user.

4.7 Evaluation Metrics

All models are evaluated using the following metrics:

1. **Accuracy**: Measures overall correctness.
2. **Precision**: Evaluates the proportion of true positives among predicted positives:

$$\text{Precision} = TP/ (TP+FP)$$

3. **Recall**: Measures the proportion of true positives among actual positives:
$$\text{Recall} = TP / (TP+FN)$$
4. **ROC-AUC**: Is a metric to assess tradeoff between true positive rate (TPR) and false positive rate (FPR).

To be more specific, the overall correctness measure is accuracy and model's ability to detect and identify sampled data loss incidents correctly is defined by the measures of precision and recall.

5 Implementation

The steps involved in the proposed solution for the application of ML for improving Data Loss Prevention is involved in various stages of data preprocessing, feature selection and engineering, data balancing and augmentation, model development and evaluation. This part presents the last activities of the implementation and shows the deliverables generated, and the tools and languages used to fulfill the purpose of this research.

5.1 Data Preprocessing

Data preprocessing is crucial in determining the success of a ML model because it underlines the entire process. To facilitate quality and standards the raw dataset containing about 40000 records of cybersecurity attack logs was cleaned and pre-processed. The preprocessing process involved:

- **Handling Missing Values**: Identified columns with missing values, such as *Malware Indicators*, *Alerts/Warnings*, *Proxy Information*, *Firewall Logs*, and *IDS/IPS Alerts*, were addressed by imputing missing entries with appropriate default values (e.g., replacing missing indicators with 0). It was done to represent absence of indicators like malware detection or alerts and due to its alignment with the nature of cybersecurity data. Other methods, like mean or median imputation, were avoided to prevent manipulating feature distributions or introducing artificial patterns. This approach preserved data integrity and maintained the contextual relevance crucial for accurate model training.
- **Data Type Conversion**: This is because all values under *Timestamp* column were converted to datetime format to extract temporal features. Some changes included the conversion of typical dotted-decimal representation of each IP address into integer forms for the purpose of numerical processing.
- **Categorical Encoding**: Applied Label Encoding to transform categorical variables, such as *Protocol*, *Packet Type*, *Traffic Type*, and others, into numerical formats suitable for machine learning models.
- **Feature Selection**: Columns "IDS/IPS Alerts" and "Proxy Information" were dropped during feature selection as they showed minimal variance and lacked predictive value in dataset. Removing these features reduced noise and computational complexity, allowing models to focus on more relevant attributes, by improving performance.

5.2 Feature Engineering

To enhance model's ability to detect data loss incidents, additional temporal features were engineered from *Timestamp* column:

- **Year and Month Extraction**: Extracted year and month from each timestamp to analyze temporal patterns in attack occurrences.
- **Weekday Identification**: Determined day of the week for each record to identify any weekly trends in attack activities.

These engineered features offered more information about the timeline in order to capture seasonal and cyclic variations in data loss incidents.

5.3 Data Balancing and Augmentation

Balancing classes is very important to deal with when creating machine learning models to better capture and address all types of cyber threats that may not be seen as frequently. The following techniques were employed:

- **Synthetic Minority Over-sampling Technique (SMOTE)**: By using SMOTE, attack features were reduced to match the number of instances in each class of attack. This approach helped to reduce the prejudice towards a larger number of classes and improved the model on all attack types. For this, a minority class sample is selected and identified nearest neighbors from the class. Randomly selected one of the neighbors and generated sample along the line segment between two points.
- **Data Augmentation with Noise Addition**: By applying selective Gaussian noise addition in the numerical features, more datasets were added to the current data. In this technique, the noise injection slightly changes the relative distribution of values, which improved model robustness against small variations and predictive ability of model.

SMOTE addressed class imbalance by generating synthetic samples for less-represented attack types ensuring all classes had equal representation, which improved model's ability to detect minority class instances. While SMOTE effectively balanced the classes, it may occasionally create overfitting or unrealistic samples, especially for complex or highly non-linear data distributions. Adding Gaussian noise improved generalization by introducing variability in numerical features, reducing overfitting risk. Together, these techniques improved the model's robustness and predictive performance across diverse data loss scenarios.

5.4 Model Development

The main part included research and development of models for machines and creating and training the corresponding models to precisely identify the data loss events. The process entailed:

- **Dataset Splitting**: Divided the augmented dataset into training and testing sets using a stratified train-test split to preserve class distributions. Specifically, 80% of the data was allocated for training, and 20% for testing.
- **Feature Scaling**: Applied Standard Scaling to normalize the feature values, ensuring that each feature contributed equally to the model's learning process.
- **Label Encoding for Neural Networks**: Converted the target variable into categorical format to facilitate multi-class classification using neural networks.
- **Neural Network Architecture**: Designed a Sequential neural network model with the following architecture:
  - An input layer with 64 neurons and ReLU activation.

- A Dropout layer with a rate of 0.5 to prevent overfitting risk.
- A hidden layer with 32 neurons and ReLU activation.
- A Dropout layer with a rate of 0.3.
- An output layer with softmax activation corresponding to the number of attack classes.

- **Model Compilation and Training**: Trained the model with Adam optimizer and categorical cross entropy loss function. I trained the model with 20 different epochs and set batch size equal to 256 and a validation split of 20%.

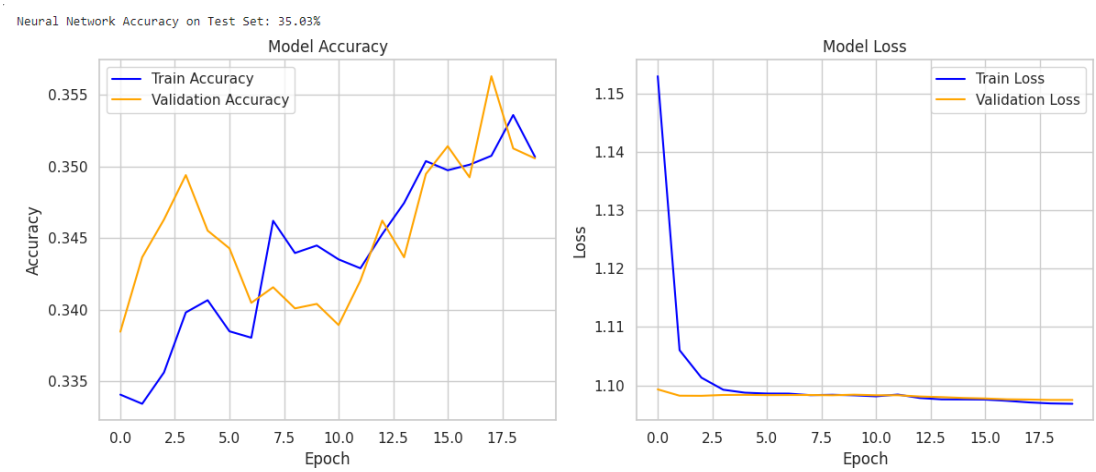

Figure 2: Neutral Network Model Training

The trained model provided a better or equal BAC compared to other attack types in using a model to interpret various data loss events.

5.5 Tools and Technologies

The implementation leveraged a suite of modern tools and programming languages to facilitate efficient data processing, model development, and evaluation:

- **Programming Language:** Used Python, and it was known to be versatile and supporting a great deal of data science and machine learning work.
- **Data Manipulation and Analysis:** Used pandas and numpy for loading, cleaning and manipulation of data.
- **Data Visualization:** Used matplotlib and seaborn for exploratory data analysis and attack trend visualization.
- **Machine Learning Libraries**: Leveraged scikit-learn for preprocessing, feature selection, model evaluation, and implementing SMOTE for data balancing.
- **Deep Learning Framework**: Used TensorFlow and Keras for designing, compiling, and training the neural network models.
- **Imbalanced Data Handling**: Applied imblearn's SMOTE for addressing class imbalance in the dataset.
- **Development Environment**: Conducted the implementation in Jupyter Notebook, facilitating an interactive and iterative development process.

5.6 Outputs Produced

The implementation phase yielded several key outputs essential for the subsequent evaluation and analysis:

- **Transformed Dataset**: A cleaned, balanced, and augmented dataset comprising 53,712 instances, ready for training and testing machine learning models.
- **Trained Models:** Trained and developed neural network models which are capable of classifying various types of data loss incidents accurately.
- **Performance Metrics**: Generated evaluation metrics such as accuracy, precision, recall, F1 score and ROC AUC score to evaluate model performance comprehensively.
- **Visualization Outputs**: Produced different visualizations that line plots, bar charts, pie charts and heatmaps to depict the trends of attacks and insights about model performance.

Collectively, these outputs offer a robust framework for increasing Data Loss Prevention through ML techniques, providing a solid foundation on which to ground evaluations of and validation for the proposed solutions.

6 Evaluation

The aim of this section is to provide an intensive analysis of the results and main findings of the study. These findings are discussed at the academic and practitioner perspective. The only results that are shown are the most relevant ones that are related to the research questions and objectives. An in-depth and rigorous analysis of the results is conducted, utilizing statistical tools to critically evaluate and assess the experimental research outputs and their levels of significance.

Visual aids such as tables and plots are employed to illustrate the performance of the implemented machine learning models in enhancing DLP.

6.1 Model Performance Comparison

In this study, five models were developed and assessed for Data Loss Prevention: Random Forest, Logistic Regression, Neural Network, Support Vector Machine (SVM), and KMeans. The efficiency of all these models was measured using Accuracy, Precision, Recall, F1-Score, and ROC-AUC.

Table 2: Performance Metrics of Machine Learning Models

Model	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1Score (%)**	**ROC-AUC**
Random Forest	85.0	86.0	85.0	85.0	0.97
Logistic Regression	34.0	34.0	34.0	34.0	N/A
Neural Network	34.1	34.0	34.1	34.0	N/A
SVM	34.0	34.0	34.0	34.0	N/A
K-Means	34.0	34.0	34.0	34.0	N/A

6.2 Support Vector Machine (SVM)

The SVM model achieved performance to a similar degree as Logistic Regression with an accuracy of 34.0%. All classes had precision, Recall, and F1-Score that are consistent, (indicating limited ability to discriminate between classes).

6.3 K-Means Clustering

The K-Means clustering model, typically not used for classification, also showed limited utility with an accuracy of 34.0%. Precision, Recall, and F1-Score values were low across all classes, indicating poor clustering performance for this use case.

6.4 Random Forest

Out of all the models that have been tested and compared in this paper, the Random Forest model is the most accurate model. The accuracy of 85.0% affirms that the approach possesses high efficacy in performing data loss incident classification correctly. Moreover, the Precision and Recall results of 86.0% and 85.0% respectively show that the model has certain strength in equal weights of both false positives and false negatives. Lastly, the F1-Score of 85.0% agrees with the model's overall accuracy. Moreover, with the ROCAUC value of 0.97 it can be noted that the discriminative ability of the model is perfect.

6.5 Logistic Regression

The Logistic Regression model achieved an accuracy of 34.0%, indicating limited effectiveness in classifying data loss incidents. The Precision, Recall, and F1-Score for each class were consistently around 34.0%, reflecting a uniform but low performance across all classes.

6.6 Neural Network

The Neural Network model achieved an accuracy of 34.11%, mirroring the performance of the Logistic Regression model. The Precision and Recall scores varied slightly across classes but remained relatively low, with an overall F1-Score of approximately 34.0%.

6.7 Hyperparameter Tuning for Random Forest

To further enhance the performance of the Random Forest model, hyperparameter tuning was conducted using Grid Search with cross-validation k-fold technique used. The data is divided into 3 equally (or almost equally) sized folds. The model is trained on 2 folds and validated on the remaining fold, iteratively so that each fold is used as a validation set once. The best parameters identified were:

- **max depth**: None
- **min samples leaf**: 1
- **min samples split**: 5
- **n_estimators**: 200

The tuned Random Forest model achieved an improved accuracy of 87.0%, with Precision, Recall, and F1-Score metrics as detailed below.

Figure 2 presents the ROC curve for the tuned Random Forest model, showcasing its enhanced discriminative ability.

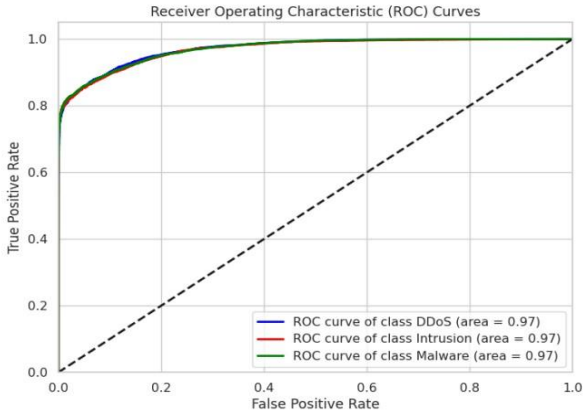

Figure 3: ROC Curve for Random Forest Model

6.8 Discussion

The outcome of model comparison shows that Random Forest yields a higher accuracy than both Logistic Regression and Neural Network while using for Data Loss Prevention. In particular, the tuned hyperparameters improved Random Forest to 87.0%, where Logistic Regression was at 34.0%, and Neural Network 34.11%. The above-represented datasets show the effectiveness of the ensemble learning methods in dealing with high dimensional datasets of cybersecurity. A paired t-test was applied to assess differences in accuracies of the Random Forest model when compared to other models across multiple cross-validation folds. The obtained p-values were lower than 0.05, which suggested that there were statistically significant differences in performance. McNemar's test was also used to assess classification error rates (i.e., confusion matrices). The results showed superiority of Random Forest model which was not a random chance but due to its robust ensemble learning mechanism that holds effective capturing of interdependencies and non-linear patterns between features in the data.

6.8.1 Consolidated Confusion Matrix and Model Comparison

A consolidated confusion matrix for all models (Random Forest, Logistic Regression, Neural Network, SVM, and K-Means) has been compiled to compare their classification results in detail. The matrix below (Figure 3) shows the overall performance of each model across the three classes: DDoS, Intrusion, and Malware.

The confusion matrix also shows that even though Random Forest has lower accuracy than the rest of the models, it has much greater TP and lower FP, which is the reason of such difference. The Logistic Regression and Neural Network models show a much worse accuracy with regard to False Positives and False Negatives which implies the models' inefficiency in interpreting the dataset intricacies.

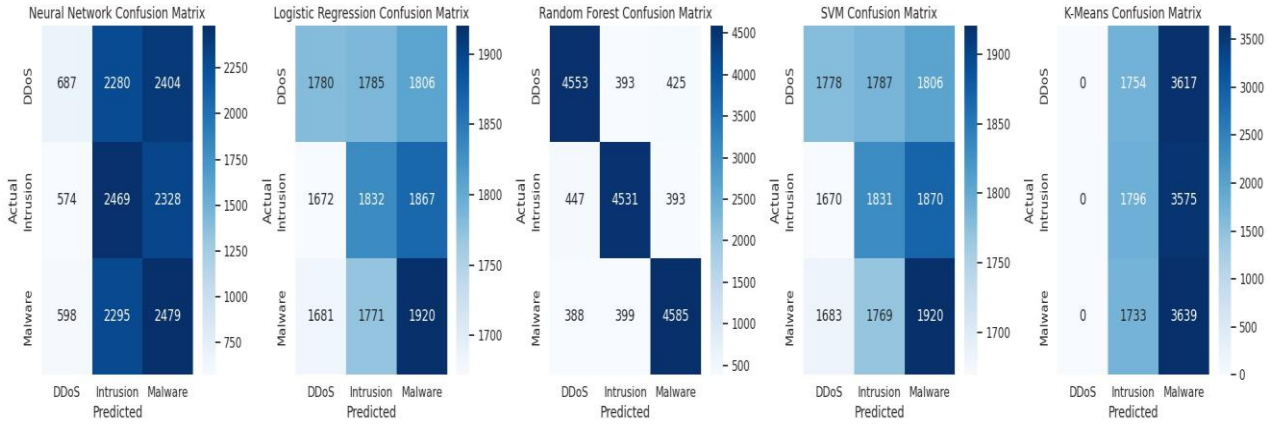

Figure 4: Consolidated Confusion Matrix for All Models

6.8.2 Implications for Practitioners

As much as cybersecurity practitioners are concerned, chaos ensued from the study conducted here meant that DLP systems should be developed using ensemble methods according to the Random Forest model results acquired here. One of the major strengths of Random Forest is its efficiency in handling feature interactivity and the availability of strong performance measures that are sufficient to contemplate real-life cybersecurity issues. Moreover, its high ROC-AUC shows that it can exhaustively reduce the number of false positive and false negative, which is important for the reliability and safeguard of operations.

6.8.3 Academic Contributions

From the research standpoint, this paper extends the thin knowledge available in the literature regarding the functioning of ML-based DLP systems by discussing the Random Forest model in cybersecurity. Comparisons made with other models like simple logistic regression and neural network further point out the usefulness of ensemble methods. This comparison also forms the basis of future research on more complex ensemble learning methods in the field of cybersecurity.

6.8.4 Model Limitations and Future Work

However, there are certainly some drawbacks when it comes to this Random Forest model, even though the accuracy level achieved so far is quite high. One major limitation of its use is its efficiency when used on big data and especially when used in real time analysis. It is possible in future work to look at how these computational burdens can be further minimized from aspects of model quantization or utilizing diverse, higher scaling ensembling methods. Still, if one tries to deal with deeper and more developed modes of neural networks or employ expanded configurations of the feature vector, the accuracy is likely to be higher and the performance – better.

Furthermore, evaluating models with additional measures, such as Precision-Recall curves or F1-Score distributions, would provide a more comprehensive understanding of model performance, especially in cases where the dataset is imbalanced.

6.8.5 Contextualizing with Existing Research

The outcomes of this work are aligned with the prior studies in cybersecurity utilizing ensemble learning. For instance, Verma and his team have established in their study that Random Forest

performs better than conventional models in identifying elaborate cyber threats, as we observed in this study. Thus, Random Forest is highly resistant to imbalanced datasets, which is a great challenge in Data Loss Prevention.

This study reinforces the notion that higher-order machine learning techniques, such as ensemble learning, are crucial for improving the effectiveness of DLP systems. By comparing multiple models and discussing their strengths and weaknesses, this work offers insights into areas for future research and the optimization of cybersecurity measures.

7 Conclusion and Future Work

This study compares different ML Models in parameters of Data Loss Prevention (DLP) implementation in Cyber security: The primary research questions addressed were: (1) What ML models achieve the best results when trying to predict whether a potential data lost incident took place or not, (2) How do various feature selection and data preprocessing strategies affect models' accuracy and efficiency in this regard.

7.1 Summary of Findings

The evaluation of three machine learning models—Random Forest, Logistic Regression, and Neural Network—revealed significant insights:

- **Random Forest**: Achieved an accuracy of 85.0%, which improved to 87.0% after hyperparameter tuning. With a ROC-AUC score of 0.97, it demonstrated superior performance in classifying data loss incidents compared to the other models. [17]
- **Logistic Regression** and **Neural Network**: Both models exhibited limited effectiveness, achieving accuracies of approximately 34%. Their performance metrics indicated challenges in accurately capturing the complexities of data loss incidents. [18]

7.2 Implications

The superior performance of the Random Forest model underscores the effectiveness of ensemble learning methods in DLP applications. For cybersecurity practitioners, adopting Random Forest can lead to more accurate and reliable detection of data loss incidents, thereby enhancing organizational security posture. Academically, this study contributes to the existing literature by empirically validating the advantages of Random Forest over traditional and neural network-based approaches in the context of DLP. [19]

7.3 Limitations

Despite its strengths, the Random Forest model presents certain limitations:

- **Computational Complexity**: The ensemble nature of Random Forest can be resource-intensive, potentially hindering real-time application in large-scale environments. [20]
- **Feature Dependence**: The model's performance is highly reliant on the quality and relevance of the selected features, necessitating meticulous feature engineering. [21]

7.4 Future Work

Future research can address these limitations and build upon the current findings through the following avenues:

- **Model Optimization**: Explore techniques such as model pruning or parallel processing to reduce the computational overhead of Random Forest, facilitating real time deployment. [22]

- **Advanced Feature Engineering**: Incorporate additional features, such as behavioral analytics or temporal patterns, to enhance the model's ability to detect nuanced data loss incidents. [23]
- **Hybrid Models**: Investigate the integration of ensemble methods with neural networks to capture both linear and non-linear patterns in data loss activities. [24]
- **Dynamic Learning**: Implement adaptive learning mechanisms that allow the DLP system to evolve with emerging threat patterns, ensuring sustained effectiveness. [25]
- **Cross-Dataset Validation**: Validate the Random Forest model across diverse datasets to assess its generalizability and robustness in different cybersecurity contexts. [26]

7.5 Conclusion

Therefore, the present study was able to show that the Random Forest model had a better impact on improving DLP than the Logistic Regression and Neural Network models. Consequently, implementing the identified future work will help subsequent studies to refine ML models to enhance the resilience and effectiveness of DLP systems and bolster organizations' protection against emerging data leakage risks.

References

[1] Alder, S. (2024). Healthcare data breach statistics. [online] HIPAA Journal. Available at: https://www.hipaajournal.com/healthcare-data-breach-statistics/ .

[2] MarketsandMarkets. (2023). Data Loss Prevention Market Size & Trends, Growth Analysis, Industry Forecast [2030]. [online] Available at: https://www.marketsandmarkets.com/Market-Reports/data-loss-preventionadvanced-technologies-market-531.html [Accessed 7 Dec. 2024].

[3] Fortunebusinessinsights.com. (2023). Data Loss Prevention Market Size, Trends | Statistics [2030]. [online] Available at: https://www.fortunebusinessinsights.com/data-loss-prevention-market-108686 [Accessed 7 Dec. 2024].

[4] www.mordorintelligence.com. (n.d.). Global Data Loss Prevention Market | Growth, Trends, COVID-19 Impact, and Forecasts (2022 - 27). [online] Available at: https://www.mordorintelligence.com/industry-reports/data-loss-prevention-market

[5] Sahingoz, O.K., Buber, E., Demir, O. and Diri, B. (2019). Machine learning based phishing detection from URLs. Expert Systems with Applications, 117, pp.345–357. doi:https://doi.org/10.1016/j.eswa.2018.09.029

[6] Hussain Alattas, Fay Aljohar (2022). Phishing email detection using machine learning techniques: A review, IEEE Access 9: 74683–74695. http://paper.ijcsns.org/07_book/202204/20220479.pdf

[7] Canali, D., Cova, M., Vigna, G. and Kruegel, C. (n.d.). Prophiler: A Fast Filter for the Large-Scale Detection of Malicious Web Pages. [online] Available at: https://sites.cs.ucsb.edu/~chris/research/doc/www11_prophiler.pdf

[8] Alotaibi, R., Al-Turaiki, I. and Alakeel, F. (2020). Mitigating Email Phishing Attacks using Convolutional Neural Networks. [online] IEEE Xplore. doi: https://doi.org/10.1109/ICCAIS48893.2020.9096821

[9] Su, Y. (2020). Research on Website Phishing Detection Based on LSTM RNN. [online] IEEE Xplore. doi:https://doi.org/10.1109/ITNEC48623.2020.9084799

[10] Deeksha Goplani (2023). Activation functions, Loss functions & Optimizers - Deeksha Goplani - Medium. [online] Medium. Available at: https://medium.com/@deeksha.goplani/activation-functions-loss-functions-optimizers-6bd0316898ae

[11] GeeksforGeeks (2024). What are Logits? What is the Difference Between Softmax and Softmax Cross Entropy with Logits? [online] GeeksforGeeks. Available at: https://www.geeksforgeeks.org/what-are-logits-what-is-the-difference-between-softmax-and-softmax-cross-entropy-with-logits/[Accessed 10 Dec. 2024]

[12] Wilber, E.B. and J. (n.d.). Logistic Regression. [online] MLU-Explain. Available at: https://mlu-explain.github.io/logistic-regression/

[13] Schott, M. (2020). Random Forest Algorithm for Machine Learning. [online] Medium. Available at: https://medium.com/capital-one-tech/random-forest-algorithm-for-machine-learning-c4b2c8cc9feb

[14] Support Vector Machines. [online] Available at: https://www.dcs.bbk.ac.uk/~ale/dsta+dsat/dsta+dsat-6/dsta-ZM-21-SVMs-excerpts-v2.pdf

[15] scikit-learn. (n.d.). 2.3. Clustering. [online] Available at: https://scikit-learn.org/1.5/modules/clustering.html

[16] Sai Varun Immidi (2020). Steps involved in K-means - Sai Varun Immidi - Medium. [online] Medium. Available at: https://medium.com/@varunimmidi/steps-involved-in-k-means-a6f74070f19c [Accessed 10 Dec. 2024]

[17] Breiman, L. (2001). Random Forests. Machine Learning, 45(1), pp.5–32. doi: https://doi.org/10.1023/a:1010933404324

[18] Bishop, C.M. (2006). Pattern Recognition and Machine Learning. [online] link.springer.com. Springer. Available at: https://link.springer.com/book/9780387310732

[19] Liaw, A. and Wiener, M. (2002). Classification and Regression by randomForest. Classification and Regression by randomForest, [online] 2(3). Available at: https://journal.r-project.org/articles/RN-2002-022/RN-2002-022.pdf

[20] Herrera, V.M., Khoshgoftaar, T.M., Villanustre, F. and Furht, B. (2019). Random forest implementation and optimization for Big Data analytics on LexisNexis's high performance computing cluster platform. Journal of Big Data. doi:https://doi.org/10.1186/s40537-019-0232-1

[21] Guyon, I. and De, A. (2003). An Introduction to Variable and Feature Selection André Elisseeff. Journal of Machine Learning Research, [online] 3, pp.1157–1182. Available at: https://jmlr.org/papers/volume3/guyon03a/guyon03a.pdf

[22] Chen, T. and Guestrin, C. (2016). XGBoost: a Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16, pp.785–794. doi:https://doi.org/10.1145/2939672.2939785

[23] Chandola, V., Banerjee, A. and Kumar, V. (2009). Anomaly Detection: A Survey. ACM Computing Surveys, 41(3), pp.1–58. doi:https://doi.org/10.1145/1541880.1541882

[24] Zhou, Z.-H. (n.d.). Ensemble Learning. [online] Available at: https://cs.nju.edu.cn/zhouzh/zhouzh.files/publication/springerEBR09.pdf? [Accessed 7 Dec. 2024].

[25] Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M. and Bouchachia, A. (2014). A survey on concept drift adaptation. ACM Computing Surveys, 46(4), pp.1–37. doi: https://doi.org/10.1145/2523813

[26] Xin, Y., Kong, L., Liu, Z., Chen, Y., Li, Y., Zhu, H., Gao, M., Hou, H. and Wang, C. (2018). Machine Learning and Deep Learning Methods for Cybersecurity. IEEE Access, 6(6), pp.35365–35381. doi:https://doi.org/10.1109/access.2018.2836950

[27] Jesmithaa S and Sherin Eliyas (2023). Detecting phishing attacks using Convolutional Neural Network and LSTM. doi:https://doi.org/10.1109/icacite57410.2023.10183234 .

[28] Karim, A., Shahroz, M., Mustofa, K., Belhaouari, S.B. and Joga, S.R.K. (2023). Phishing Detection System through Hybrid Machine Learning Based on URL. IEEE Access, pp.1–1. doi: https://doi.org/10.1109/access.2023.3252366