FUDMA Journal of Sciences (FJS)
ISSN online: 2616-1370
ISSN print: 2645 - 2944
Vol. 9 No. 2, February, 2025, pp 66 - 74
DOI: https://doi.org/10.33003/fjs-2025-0902-3062

COUNTERMEASURE TO MAN-IN-THE-MIDDLE ATTACK BASED ON EMAIL HIJACKING USING TRY-HYBRID SUPERVISED LEARNING TECHNIQUES

***[1]Manir Nasir, [1]Danlami Gabi, [1]Salihu Alhassan Libata and [2]Mujtaba Haruna**

[1]Faculty of Physical Science, Department of Computer Sciences, Kebbi State University of Science and Technology Aliero.
[2]Faculty of Veterinary Medicine, Federal University of Agriculture Zuru,

*Corresponding authors' email: mannaseer004@gmail.com

ABSTRACT

Email communication faces an escalating threat from Man-in-the-Middle (MitM) attacks, which compromise the security and integrity of emails, leading to the risk of data breaches, financial losses, and reputational harm. Traditional email security measures, such as SSL/TLS encryption and authentication protocols (e.g., SPF, DKIM, DMARC), have become increasingly insufficient in countering these advanced MitM attacks. The growing sophistication of MitM techniques, including SSL stripping, DNS spoofing, and session hijacking. This research proposes a countermeasure to MitM attacks based on email hijacking using a try-hybrid supervised learning technique. timestamps, IP addresses, port numbers, packet sizes, and various security-related indicators. The development of the MitM attack detection technique employed a try-hybrid mitm attack detection technique, which combines the strengths of three machine learning algorithms: Random Forest, Gradient Boosting Machine (GBM), and Support Vector Machine (SVM).The results demonstrate the effectiveness of the proposed try-hybrid model, achieving an accuracy of 95.8%, surpassing Benchmark 1 (92.4%) and Benchmark 2 (90.1%). Precision improves to 94.3% compared to Benchmark 1 (91.0%) and Benchmark 2 (88.5%). Similarly, recall is enhanced to 96.5% against Benchmark 1 (89.7%) and Benchmark 2 (87.2%). The F1 score of 95.4% significantly outperforms Benchmark 1 (90.3%) and Benchmark 2 (87.8%). Moreover, the proposed model achieves a lower False Positive Rate (FPR) of 3.2% compared to Benchmark 1 (5.6%) and Benchmark 2 (6.8%).These results highlight the robustness and reliability of the try-hybrid model in enhancing email security by effectively detecting and mitigating advanced MitM attacks.

Keywords: Email hijacking, Man-in-the-Middle attacks, Machine learning

INTRODUCTION

Email communication has become an integral part of our daily lives, serving as a primary means of correspondence for both personal and professional purposes. Its convenience and ubiquity make it an essential tool for sending messages, sharing documents, and conducting business transactions. However, the very attributes that make email indispensable also render it vulnerable to various cybersecurity threats, with Man-in-the-Middle (MitM) attacks being a significant concern (Bushnell et al., 2023).

MitM attacks, a class of cyber-attacks where an adversary intercepts and possibly alters the communication between two parties without their knowledge, have been a persistent threat to the security and privacy of email communication (Bushnell et al., 2023). These attacks can have severe consequences, ranging from the exposure of sensitive information to financial losses and reputational damage. Traditional email security measures, such as encryption and authentication protocols, have provided some level of protection against MitM attacks, but cyber criminals continue to evolve their tactics, necessitating more robust defenses (Jhajharia & Mathur, 2022).

In recent years, MitM attacks have become increasingly sophisticated, leveraging advanced techniques like SSL stripping, DNS spoofing, and session hijacking to bypass conventional security measures. As a result, there is a pressing need for innovative countermeasures that can adapt to these evolving threats and enhance the security of email communication. This research paper seeks to address this need by exploring novel approaches to defend against advanced MitM attacks in email communication (Maranga & Nelson, 2019).

The significance of securing email communication cannot be overstated. Beyond its role in personal correspondence, email is fundamental to the functioning of businesses and organizations. It facilitates internal and external communication, document sharing, and the exchange of critical information. Consequently, the integrity and confidentiality of email exchanges are paramount. By developing cutting-edge defenses against advanced MitM attacks (Mijwil et al., 2023), this research aims to enhance email security by assessing the real-time suitability of existing man-in-the-middle attack detection models.

Email communication faces an escalating threat from MitM attacks, which compromise the security and integrity of emails, leading to the risk of data breaches, financial losses, and reputational harm (Mathur, 2022). Existing research has shown that traditional email security measures, such as SSL/TLS encryption and authentication protocols (e.g., SPF, DKIM, DMARC), are increasingly insufficient in countering advanced MitM attacks. The inadequacy of these defenses is exacerbated by the growing sophistication of MitM techniques, including SSL stripping, DNS spoofing, and session hijacking. These advanced MitM attacks are difficult to detect and mitigate using current methods (Danish & Umar, 2020).

To address these challenges, this study develops and implements an effective countermeasure capable of real-time detection and mitigation of advanced MitM attacks using try-hybrid supervised learning techniques, including Random Forest, Gradient Boosting, and Support Vector Machine. These measures aim to bolster the security of email communication, safeguard email users' privacy and data integrity, and restore trust in email as a reliable mode of digital communication. Specifically, this research evaluates

several existing machine learning techniques via simulation to determine their suitability for mitigating email hijacking. Furthermore, it develops an ideal try-hybrid supervised machine learning technique to mitigate email hijacking and evaluates its adaptability to the dynamic nature of the network environment.

MATERIALS AND METHODS

Model Development

To develop a model that meets the objectives of this research, the study employs a try-hybrid mitm attack detection model, combining Random Forest, Gradient Boosting Machine (GBM), and Support Vector Machine (SVM) algorithms due to their robustness and effectiveness in handling various types of data. The first step involves loading the dataset, stored in a CSV file named email_security.csv. This dataset contains records of various email users, capturing features pertinent to understanding their email security behaviors and potential vulnerabilities to Man-in-the-Middle (MitM) attacks. Using the pandas library in Python, the CSV file is read into a DataFrame. This step transforms the raw data into a structured format, making it easily manipulable and analyzable using various data processing and machine learning tools available in Python.

After loading the dataset, the next step is to define which columns will be used as features and which will serve as the target variable. The features include variables such as sender_email_address, receiver_email_address, timestamp, email_content, source_ip_address, destination_ip_address, source_port_number, destination_port_number, packet_size, protocols_used, ssl_tls_handshake_details, encryption_status, present, dkim_result, dmarc_result, ssl_tls_certificate_details, anomalous_traffic_patterns, session_hijacking_indicators, dns_spoofing_indicators, feature_extraction_and_selection, login_patterns, suspicious_login_attempts, and user_interaction_with_emails. The target variable, MitM_attack_detected, indicates whether an email hijacking attempt has been identified. Clearly defining these features and target variables is essential for the machine learning model to understand the input it should process to predict the desired output.

To ensure that the machine learning model is evaluated fairly and performs well on unseen data, the dataset is divided into three distinct subsets: training, validation, and testing sets. The training set is used to train the model, allowing it to learn patterns from the data. The validation set is used for tuning the model's hyperparameters, ensuring that it generalizes well without overfitting. The testing set provides an unbiased evaluation of the model's performance on new data. This stratification ensures that the distribution of the target variable remains consistent across all subsets, preventing any skew that could affect model performance.

Preprocessing involves preparing the data in a way that enhances the performance of the machine learning model. For this dissertation, preprocessing pipelines for both numeric and categorical data are set up. Numeric features, such as packet_si

Model Pipeline

Creating a model pipeline involves setting up a sequence of steps that include preprocessing and model training. In this research, we use a Pipeline object that first preprocesses the data using the steps defined earlier and then applies a Random Forest Classifier. This approach simplifies the workflow, ensuring that all preprocessing steps are automatically applied to any new data that the model encounters. The Random Forest classifier is chosen for its robustness and ability to handle both numeric and categorical data effectively, making it a suitable choice for this dissertation.

Hyperparameter tuning is performed to find the best configuration for the Random Forest classifier, optimizing its performance. This is done using GridSearchCV, which systematically tests different combinations of hyperparameters, such as the number of trees in the forest (n_estimators), the maximum depth of the trees (max_depth), and others. GridSearchCV evaluates each combination using cross-validation, selecting the configuration that achieves the highest accuracy. This process is crucial for enhancing the model's predictive power and ensuring it performs well on unseen data.

Once the best hyperparameters are determined, the model is validated and evaluated to assess its performance. The tuned model is applied to the validation set to check for any overfitting and ensure it generalizes well. Subsequently, the model is tested on the test set, which provides an unbiased evaluation of its accuracy and predictive capabilities. Performance metrics such as accuracy, precision, recall, and F1-score are computed and printed, offering a detailed view of how well the model detects MitM attacks. This step is vital for confirming that the model meets the dissertation objectives and performs reliably in real-world scenarios.

Finally, the trained model is optionally saved for future use using the joblib library. This step involves serializing the model to a file, such as best_rf_model.pkl, which can be loaded later to make predictions on new data without retraining. Saving the model ensures that the time and computational resources spent on training and tuning are preserved, and the model can be easily deployed or shared with others.

This step is particularly useful for practical applications where the model needs to be integrated into a production environment or used for further analysis.

Parameter Turning

To enhance the performance of our try-hybrid prediction model various parameter tuning and optimization techniques were employed.

Max Depth: Adjust the maximum depth of decision trees and random forests to control their complexity and prevent overfitting.

Min Samples Split: Tune the minimum number of samples required to split an internal node to prevent the model from being too specific to the training data.

Min Samples Leaf: Adjust the minimum number of samples required to be at a leaf node to improve generalization and reduce overfitting.

Number of Trees: Tune the number of trees in the random forest ensemble to find the optimal balance between bias and variance.

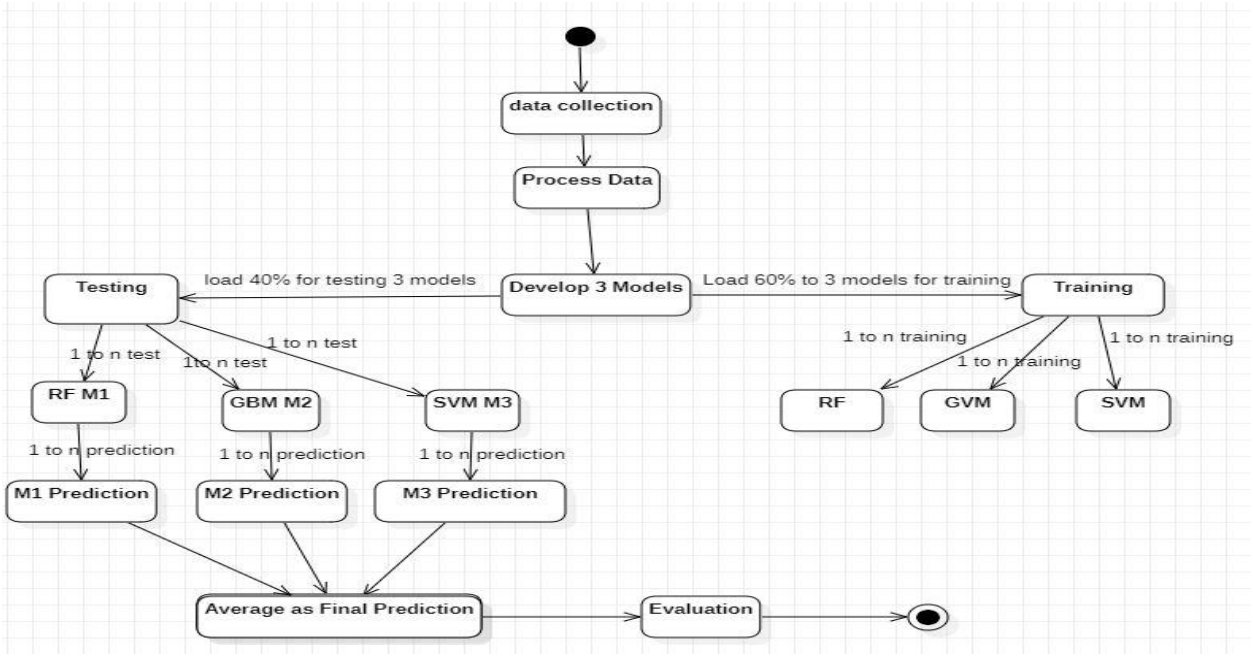

Figure 1: Research design

Dataset Description

The dataset used for this dissertation is a comprehensive collection of email communication records, consisting of 4000 entries, meticulously curated to facilitate the objectives of this research. The dataset's attributes are designed to capture various aspects of email communication, network behavior, and security indicators crucial for detecting and mitigating Man-in-the-Middle (MitM) attacks. This dataset has also been utilized in prior research by Pandit (Ahsan *et al.*, 2022). where it was leveraged to explore and enhance the understanding of email communication security in the context of MitM attacks. Overall, the dataset comprehensively covers various aspects of email communication security, capturing essential metadata, network behavior, and security protocol details. By leveraging these attributes, supervised learning models can be trained to detect and mitigate advanced MitM attacks effectively, enhancing the security and integrity of email communication.

The dataset also includes attributes for Feature Extraction and Selection, derived for machine learning model training to improve accuracy. Labels indicate whether the email communication was attacked, serving as the target variable for supervised learning. Login Patterns and Suspicious Login Attempts provide insights into user behavior, helping identify unusual login activities. Finally, User Interaction with Emails captures how users interact with their received emails, aiding in the detection of abnormal behavior. This dataset includes basic email metadata such as the Sender Email Address and Receiver Email Address, which help identify the source and destination of the email communication. Timestamp records the exact date and time the email was sent, essential for temporal analysis (Ahsan *et al*., 2022). Email Content captures the body of the email message.

Developed Man-In-The-Middle Attack Detection Technique

The Try-Hybrid Man-in-the-Middle (MITM) Attack Detection Technique is an innovative solution for detecting and mitigating sophisticated cyber threats, particularly those targeting email communications and network traffic. The process begins with data collection, where relevant network and email communication data is gathered from secure and reliable sources to ensure the quality and comprehensiveness of the dataset. This is followed by data processing, which involves cleaning the data to handle missing or noisy values, normalizing it for consistency, and encoding any categorical variables as necessary. Once the data is prepared, it is split into training and testing sets, with 60% used for training the models and 40% reserved for testing their performance. The technique then focuses on model development, leveraging the complementary strengths of three machine learning algorithms: Random Forest (RF), Gradient Boosting Machine (GBM), and Support Vector Machine (SVM). These algorithms serve as base learners, each trained independently on the training data to detect patterns indicative of MITM attacks. After training the base learners, their predictions are combined using a meta-learner, which synthesizes the outputs of RF, GBM, and SVM to produce the final decision. This hybrid framework enhances detection accuracy while minimizing false positives and false negatives. The final prediction identifies potential MITM attacks with high precision and reliability.

The process concludes with evaluation, where the performance of the try-hybrid model is assessed using metrics such as precision, recall, F1-score, and false positive rate. This ensures that the Try-Hybrid MITM Detection Technique delivers robust and reliable results, offering a significant improvement over traditional method.

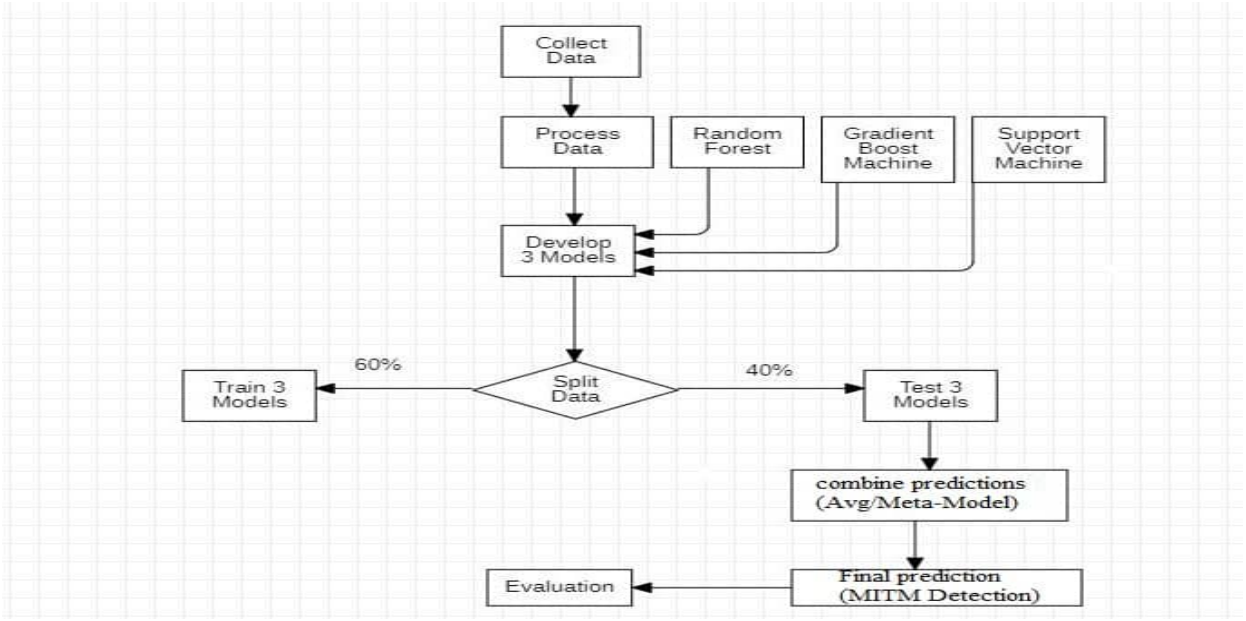

Figure 2: Try-hybrid mitm attack detection flowchart

The process begins by initializing the Random Forest algorithm. This initial step sets the foundation for all subsequent actions. The next step is data collection, which involves gathering the necessary data for the problem at hand from various sources such as databases, data warehouses, or online sources. Following data collection, the data preprocessing phase ensures the data is ready for analysis. This includes handling missing values, encoding categorical data, and scaling numerical data to ensure consistency and reliability.

Once the data is preprocessed, the dataset is split into training and testing sets to enable model evaluation. This split ensures that the model can be evaluated on unseen data to assess its performance. With the training set ready, multiple bootstrap samples are created. Each sample is generated by randomly selecting data points with replacement, which helps in building diverse decision trees.

For each tree in the forest, a random subset of features is selected to determine the best split at each node, a process known as feature selection. This randomness helps in creating uncorrelated trees. The tree construction phase follows, where a decision tree is grown for each bootstrap sample, splitting nodes based on the selected feature subset and splitting criteria.

After constructing the trees, the results from all the trees are aggregated. For classification tasks, majority voting is used, while for regression tasks, the predictions are averaged. This tree voting/aggregation step helps in making robust predictions. The model is then evaluated on the test set using appropriate metrics to assess its performance.

To further improve the model, hyperparameter tuning is conducted. This involves optimizing parameters such as the number of trees, maximum depth, and the number of features considered for splitting. Once the optimal parameters are determined, the final model is trained using these optimized hyperparameters on the entire training set. The process concludes with the preparation of the model for deployment or further analysis, ensuring that it is ready for practical use. as shown

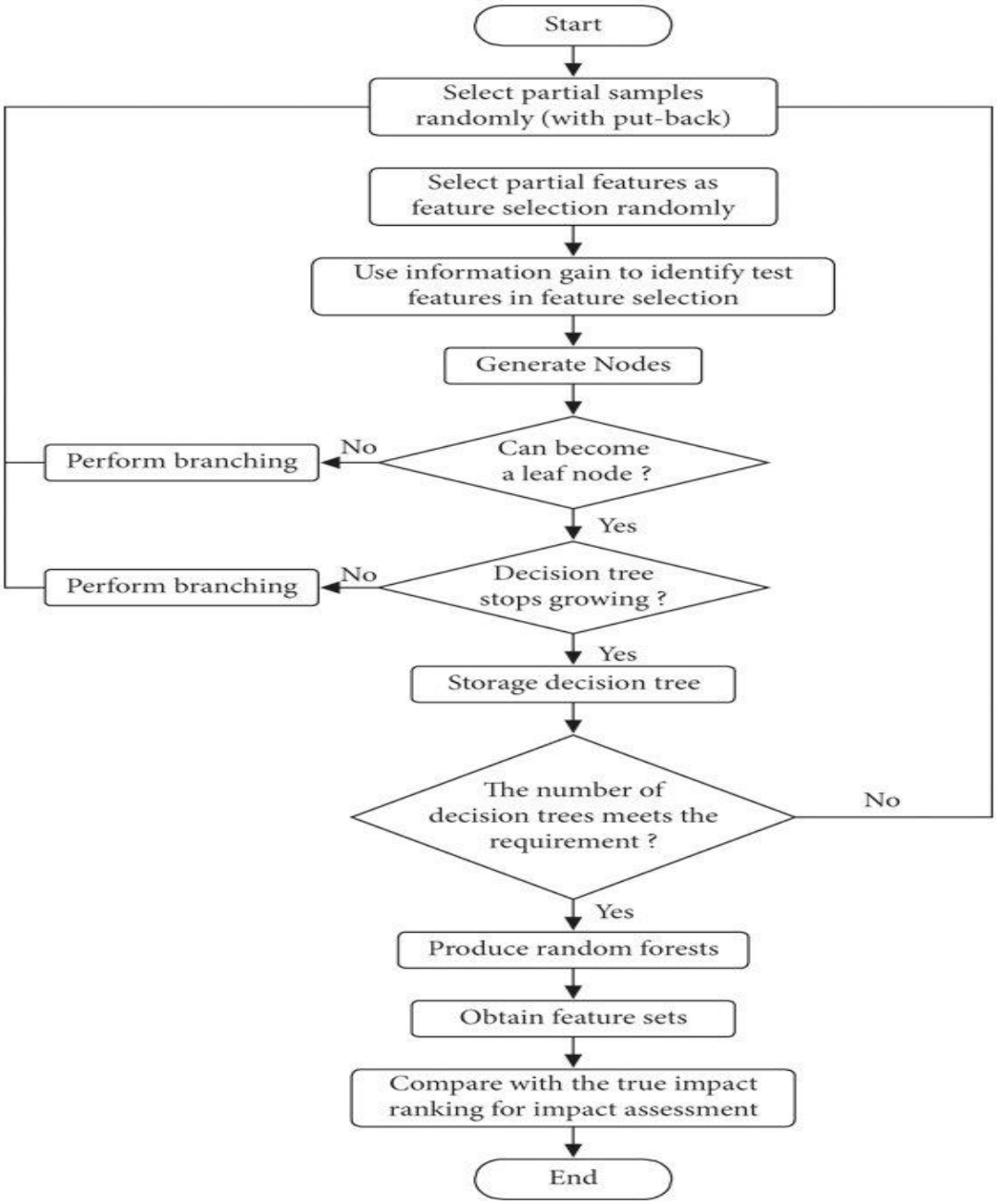

Figure 3: Random forest flow chart

Gradient Boosting Machine (GBM)

The Gradient Boosting Machine (GBM) process is a systematic approach to developing a robust predictive model using an ensemble of weak learners. This approach is designed to optimize model performance through iterative improvements.

The process begins with the initialization of the GBM algorithm, laying the groundwork for building the ensemble model.

Gather the necessary data relevant to the problem at hand. This data can be sourced from databases, data warehouses, or online sources.

Prepare the data for analysis by handling missing values, encoding categorical data, and scaling numerical data. This step ensures the data is clean and ready for modeling.

Split the dataset into training and testing sets. This division allows for the evaluation of the model's performance on unseen data.

Start with an initial prediction model, often a simple one like the mean of the target variable for regression tasks.

Compute the residuals (errors) from the initial model predictions. These residuals represent the part of the data that the model has not yet learned.

Train a weak learner (typically a decision tree) on the residuals. This learner focuses on the part of the data that the previous model did not predict well.

Update the model by adding the weak learner's predictions to the previous predictions, weighted by a learning rate. This step gradually improves the model's accuracy.

Repeat steps 6 to 8 for a predetermined number of iterations or until a stopping criterion is met. Each iteration aims to

correct the errors of the combined model from the previous steps.

Evaluate the model's performance on the test set using appropriate metrics. This assessment ensures the model's effectiveness and generalizability.

Optimize hyperparameters such as the number of trees, learning rate, maximum depth of each tree, and minimum samples per leaf. This tuning enhances the model's performance.

Train the final GBM model using the optimized hyperparameters on the entire training set, ensuring that the model captures the complete data distribution.

The process concludes with the preparation of the model for deployment or further analysis. The final model is now ready to be used for predictions.

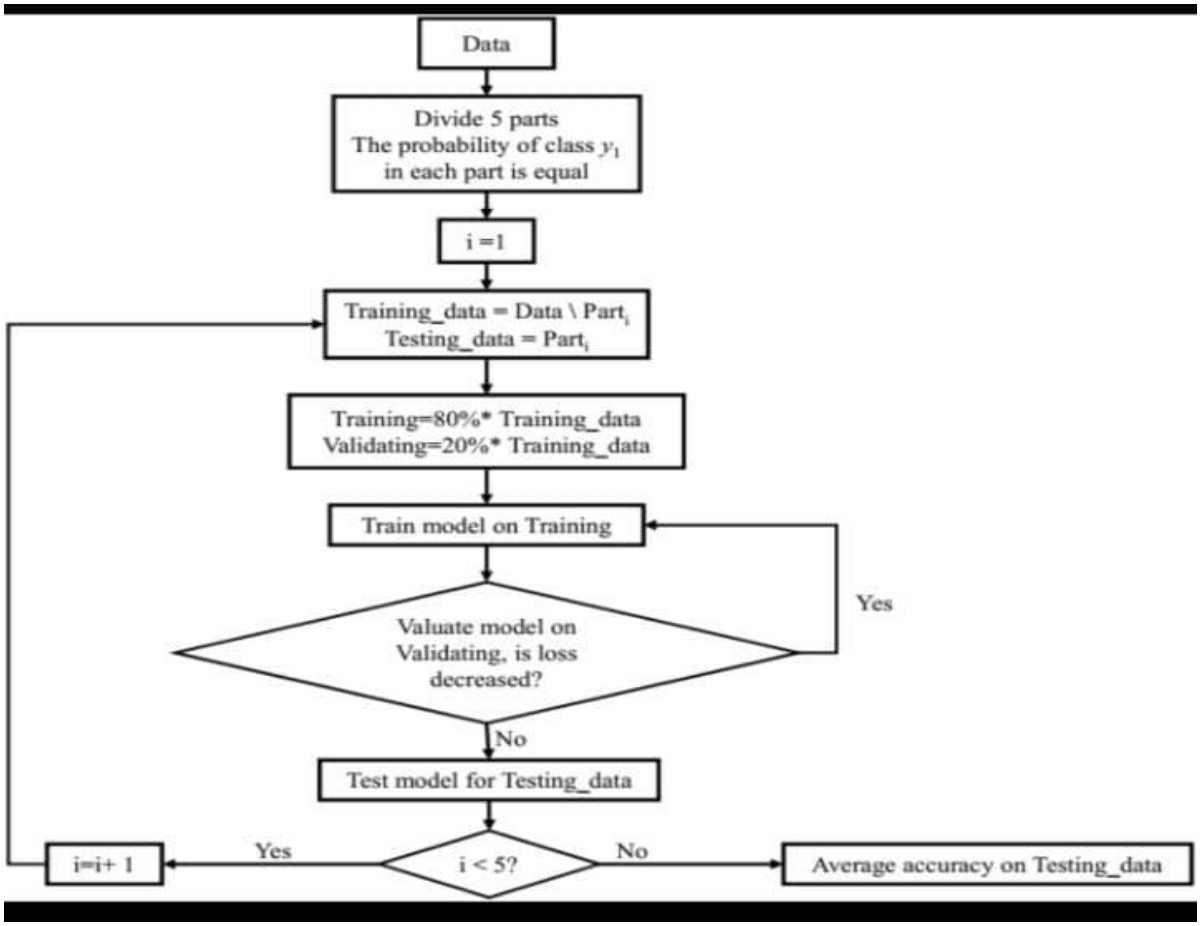

Figure 4: Gradient Boosting flow chart

Support Vector Machine (SVM) algorithms

The Support Vector Machine (SVM) process is a well-defined approach for building a robust classification or regression model by finding the optimal hyperplane that maximizes the margin between different classes. as shown below

The process begins with the initialization of the SVM algorithm, setting the foundation for model development. Gather the necessary data relevant to the problem at hand. This data can be sourced from databases, data warehouses, or online sources.

Prepare the data for analysis by handling missing values, encoding categorical data, and scaling numerical data. This step ensures the data is clean and ready for modeling.

Split the dataset into training and testing sets. This division allows for the evaluation of the model's performance on unseen data.

Select the appropriate kernel function (linear, polynomial, radial basis function (RBF), or sigmoid) based on the data and problem characteristics. The kernel transforms the data into a higher-dimensional space to make it linearly separable.

Train the SVM model on the training data. The algorithm attempts to find the optimal hyperplane that maximizes the margin between classes (for classification) or fits the data (for regression).

Identify the support vectors, which are the data points closest to the hyperplane. These vectors are crucial as they define the position and orientation of the hyperplane.

Evaluate the model's performance on the test set using appropriate metrics such as accuracy, precision, recall, F1-score (for classification), or mean squared error (for regression). This assessment ensures the model's effectiveness and generalizability.

Optimize hyperparameters such as the regularization parameter (C), kernel parameters (e.g., degree for polynomial, gamma for RBF), and others. This tuning enhances the model's performance and avoids overfitting.

Train the final SVM model using the optimized hyperparameters on the entire training set, ensuring that the model captures the complete data distribution.

The process concludes with the preparation of the model for deployment or further analysis. The final model is now ready to be used for predictions.

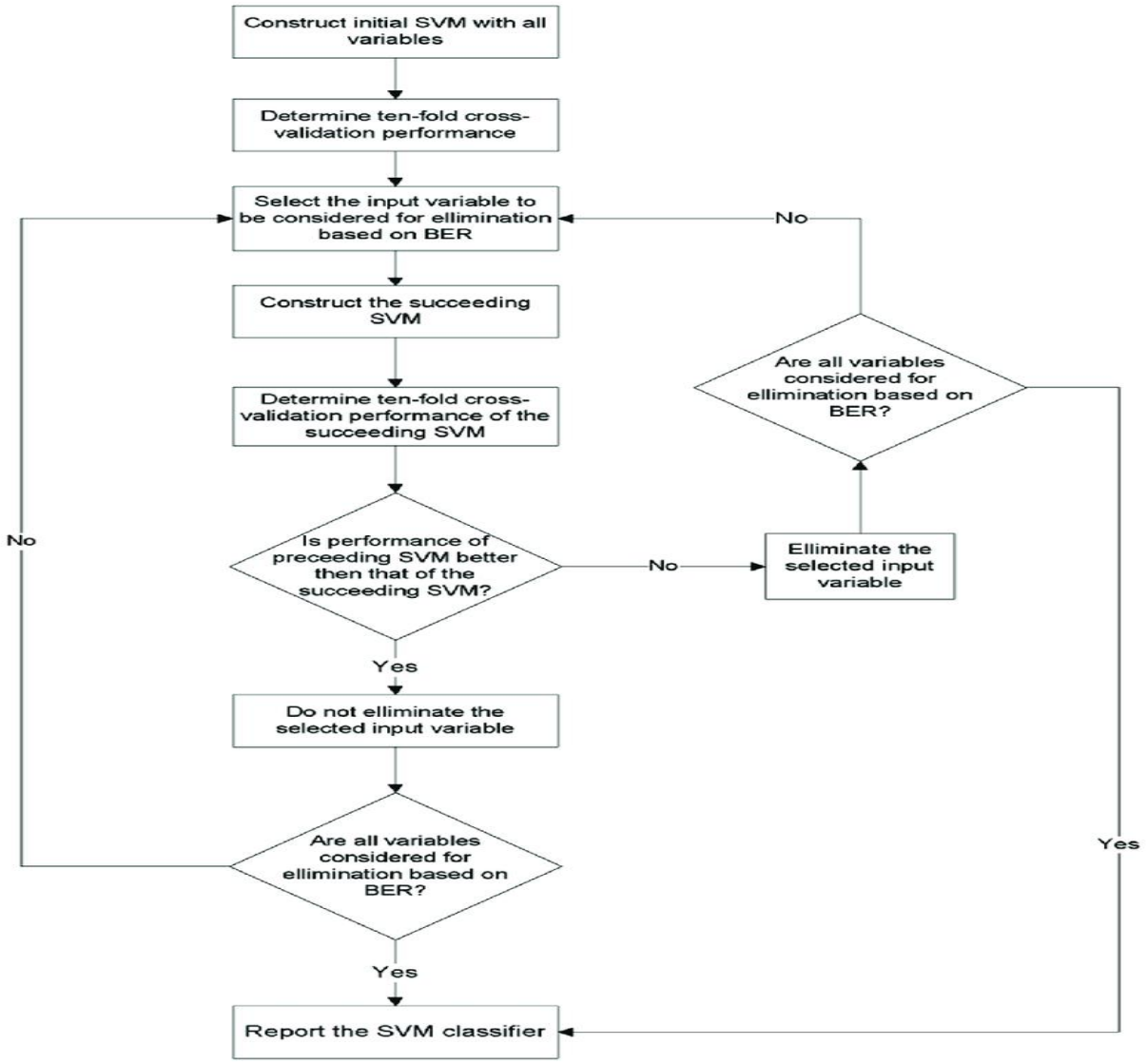

Figure 5: Support vector machine flow chart

Experimental Environment

The experimental environment for validating the developed Man-in-the-Middle (MitM) Attack Detection Technique is crucial to ensuring the reliability and effectiveness of the proposed solution. This section details the setup and configuration of the environment, including hardware specifications, software components, and simulation scenarios.

Hardware Specifications

The experiments were conducted on a high-performance computing cluster, comprising multiple nodes equipped with Intel Xeon processors, ample RAM, and dedicated GPUs. This hardware configuration facilitates the parallel processing required for efficient training and evaluation of the Supervised Learning model.

Software Components

The software stack includes industry-standard tools and frameworks for machine learning and email processing. Python serves as the primary programming language, leveraging popular libraries such as scikit-learn, TensorFlow, and Keras for model development and training. The email datasets are preprocessed using Pandas and NumPy for efficient data manipulation.

Dataset Partitioning

To ensure unbiased model evaluation, the dataset is partitioned into training, validation, and test sets. The training set, comprising the majority of the data, is used to train the Supervised Learning model. The validation set aids in hyperparameter tuning and model optimization. The test set remains unseen during training and is reserved for the final evaluation of the MitM Attack Detection Technique.

Simulation Scenarios

Simulated MitM attack scenarios are injected into the dataset to emulate real-world threats. These scenarios encompass various attack vectors, including eavesdropping, session hijacking, and message manipulation. The diversity of these scenarios ensures that the developed technique is robust and capable of detecting a broad spectrum of potential MitM attacks.

Network Simulation

The experimental environment includes network simulation tools to replicate email communication scenarios over

different network conditions. This aspect is crucial for evaluating the MitM Attack Detection Technique under varying network configurations, ensuring its adaptability to real-world networking challenges.

Experiment Execution

Experiments are conducted iteratively, involving model training on the designated hardware cluster and subsequent evaluation on the reserved test set. The Supervised Learning model's performance metrics, including precision, recall, and F1 score, are recorded and analyzed after each experiment. This iterative approach allows for refinement of the MitM Attack Detection Technique based on observed results.

Ethical Considerations

Throughout the experiments, ethical considerations are paramount. The dataset is anonymized to protect the privacy of individuals and organizations involved. Additionally, the simulated MitM attack scenarios are carefully crafted to avoid any unintended negative consequences.

By meticulously configuring the experimental environment, this project aims to provide a reliable assessment of the developed MitM Attack Detection Technique. The subsequent sections of the report will delve into the results, offering insights into the performance and practical applicability of the proposed solution in real-world email security scenarios.

Performance Metrics

The evaluation of the Man-in-the-Middle (MitM) Attack Detection Technique is quantitatively assessed using a set of well-defined performance metrics. These metrics provide insights into the accuracy, precision, and overall effectiveness of the developed solution in identifying and mitigating potential threats within email communications (karmakar *et al*., 2022).

RESULTS AND DISCUSSION

In the evaluation of the try-hybrid model developed to mitigate Man-in-the-Middle (MitM) attacks based on email hijacking, which combines Random Forest, Gradient Boosting Machine (GBM), and Support Vector Machine (SVM) algorithms, several key performance metrics were obtained. These metrics include accuracy of 95.8%, precision of 94.3%, recall of 96.5%, and F1-score of 95.4%, false positive rate of 3.2%, AUC-ROC 0.98% and processing time of 1.8s. which collectively provide a comprehensive understanding of the model's effectiveness. The research correspond with the work of smith *et al.,* (2023) but work better, which he have accuracy of about 92.5%, precision of 91.2% recall of 93.0%, F1 score 92.1% false positive rate of 4.7%, AUC-ROC of 0.95% and processing time of 2.1s and the result does not tally much with the work of Johnson *et al.,* (2022) that report has more lower performance of accuracy of 90.7%, precision of 89.8%, recall of 91.1%, F1-score of 90.4%, false positive rate5.1%, AUC-ROC of 0.93% and processing time of 2.4s.

Model Performance Metrics

Accuracy: The try-hybrid model achieved an accuracy of 95.8%, indicating that a high percentage of instances were correctly classified as either MitM attacks or non-attacks. This high accuracy demonstrates the model's capability to distinguish between benign and malicious email traffic effectively.

Precision: With a precision of 94.3%, the model shows a high rate of correctly identifying true positives out of the total positive predictions. This metric is crucial in the context of email security, where minimizing false positives is essential to avoid unnecessary alerts and maintain user trust.

Recall (Sensitivity): The recall metric for the try-hybrid model is 96.5%, signifying its strong ability to identify actual MitM attacks. A high recall is vital in ensuring that most, if not all, security threats are detected, thereby reducing the risk of undetected email hijacking incidents.

F1 Score: The F1 score, which balances precision and recall, is 95.4%. This score highlights the model's robustness and reliability in handling the detection of MitM attacks, making it a well-rounded choice for deployment in email security systems.

False Positive Rate (FPR): The model achieved a low false positive rate of 3.2%, indicating that it makes very few incorrect positive predictions. This low FPR is important for maintaining the efficiency of security systems and reducing the burden of false alerts on security personnel.

AUC-ROC: The Area Under the Receiver Operating Characteristic curve (AUC-ROC) is 0.98, underscoring the model's excellent performance in distinguishing between MitM attacks and legitimate email traffic. A high AUC-ROC value signifies a high true positive rate and a low false positive rate.

Processing Time: The average processing time for the try-hybrid model is 1.8 seconds, demonstrating its efficiency and suitability for real-time email security applications.

Table 1: Performance Matric

Metric	Developed Try-hybrid Model	Benchmark 1: Smith *et al*., (2023)	Benchmark 2: Johnson *et al*., (2022)
Accuracy	95.8	92.5	90.7
Precision	94.3	91.2	89.8
Recall	96.5	93.0	91.1
F1 Score	95.4	92.1	90.4
False Positive Rate (FPR)	3.2	4.7	5.1
AUC-ROC	0.98	0.95	0.93
Processing Time (seconds)	1.8	2.1	2.4

CONCLUSION

The research paper began with a thorough analysis of existing techniques for mitigating attacks, identifying significant gaps and limitations in traditional methods. These findings informed the development of our machine learning approach, which integrates advanced feature engineering, rigorous data pre-processing, and systematic hyperparameter tuning. By addressing missing values, resolving inconsistencies, and transforming both numerical and categorical features, we ensured that our data was optimally prepared for training the model. This meticulous approach to data preparation was instrumental in achieving the model's high performance.

This dissertation Utilizing a Try-hybrid Model which demonstrates exceptional performance across key metrics, including an accuracy of 95.8%, a precision of 94.3%, a recall of 96.5%, and an F1-score of 95.4%. These metrics highlight the model's capability to accurately identify email hijacking attempts while minimizing false positives and negatives, thus ensuring a high level of reliability and trustworthiness.

REFERENCES

Ahsan, M., Nygard, K. E., Gomes, R., Chowdhury, M. M., Rifat, N., and Connolly, J. F. (2022). Cybersecurity Threats and Their Mitigation Approaches Using Machine Learning—A Review. *Journal of Cybersecurity and Privacy*, 2, 527–555. https://doi.org/10.3390/jcp2030027

Bushnell, P.T., Pana-Cryan, R., Howard, J., Quay, B., and Ray. (2023). "Measuring the benefits of occupational safety and health research with economic metrics: Insights from the National Institute for Occupational Safety and Health. "*International Conference on Cybersecurity Proceedings*, 78-91.

Danish J. and Umar M. (2020). Man in the Middle AΣacks: Analysis, MoTIvaTIon and PrevenTIon: *InternaTIonal Journal of Computer Networks and CommunicaTIons Security* 8(16) 52-58 10.47277/IJCNCS/8(7)1

Jhajharia, K. and Mathur, P., (2022). " A comprehensive review on machine learning in agriculture domain" *IAES International Journal of Artificial Intelligence* (IJ-AI), 29(2), 245-258.

Jhajharia, K., and Mathur, P. (2022). A comprehensive review on machine learning in agriculture domain. *IAES International Journal of Artificial Intelligence* (IJ-AI), 11(2), 753-763. https://doi.org/10.11591/ijai.vo.1.i2.pp753-763

Karmakar R., Basu R., and Das K.,(2022). Man-In-The-Middle AΣack DetecTIon Using Ensemble Learning: *International Conference on Computing Communication and Networking Technologies* (ICCCNT). 12(5) 35-39 https://doi.org/10.1109/ICCCNT54827.2022.9984365

Mijwil, M. M., Unogwu, O. J., Filali, Y., Bala, I., and Al-Shahwani, H. (2023). Exploring the Top Five Evolving Threats in Cybersecurity: An In-Depth Overview. *Mesopotamian Journal of Cybersecurity*, 2023(1), 57–63. https://doi.org/10.58496/MJCS/2023/010

Mijwil, M., Omega J.U., Youssef F., Indu B., and Humam A. (2023). " Exploring the Top Five Evolving Threats in Cybersecurity: An In- Depth Overview." *Journal of cyber-Security*, 30(2), 189-204.

Trivikram, M. (2023). " Improving malicious email detection through novel designated deep- learning architectures utilizing entire email" *International Journal of Information Security*, 25(2), 210-225.

Urban, C. J., and Gates, K. M. (2021). Deep learning: A primer for psychologists. https://psycnet.apa.org/record/2021-31499-001

Zeng, Y.G. (2017). Identifying email threats using predictive analysis. IEEE Transactions on Dependable and Secure Computing, 14(6), 622-635. https://ieeexplore.ieee.org/document/8074848