A Study on Detecting and Preventing Phishing Attacks Using Machine Learning Techniques

Ana-Maria BROȘTIC[1], Bianca-Andreea MĂRGĂRIT[1], Diana-Sînziana MIHAI[1], Andreea NICOLESCU[1], Emil SIMION[2]
[1]Faculty of Applied Sciences,
National University of Science and Technology POLITEHNICA Bucharest
[2] Department of Mathematical Methods and Models,
Centre for Research and Training in Innovative Techniques of Applied Mathematics in Engineering,
National University of Science and Technology POLITEHNICA Bucharest
anabrostic@yahoo.com, bianca.andreea_margarit@yahoo.com, dumitrassinziana@yahoo.com, nicolescuandreea123@gmail.com, emil.simion@upb.ro

Abstract: Phishing is a major cybersecurity threat that targets users and organizations by exploiting deceptive e-mail tactics to steal sensitive data. Because the traditional methods often fail against the evolving phishing attacks, the authors explore how machine learning approaches can improve phishing detection. They evaluate the supervised learning models, including Support Vector Machine, Logistic Regression, and Random Forest, comparing their accuracy, precision, and sensitivity. The study also employs feature selection, data preprocessing, and ensemble learning to enhance the results. The goal is to improve user protection by identifying the most effective machine learning-based solution for detecting phishing attacks and advancing cybersecurity defense strategies.
Keywords: Phishing, Machine Learning, Support Vector Machine, Logistic Regression, Random Forest, e-mail, cybersecurity, user protection, phishing attacks.

INTRODUCTION

To better understand and effectively combat phishing attacks, it is essential to define the concept, analyze how it operates, and identify ways to prevent it. Phishing is a fraudulent method in which attackers try to obtain sensitive information, such as login or bank details, through misleading messages. This section will provide a detailed explanation of the mechanisms of this type of attack and defense strategies against it.

Next, some existing research in the field of phishing detection using machine learning (ML) techniques will be analyzed. This review will include an overview of the main works, highlighting the methods used, the datasets examined, and the conclusions on the performance of the models applied. This information will provide a solid basis

https://doi.org/10.54851/v7i2y202505

for improving the algorithms that will be implemented, contributing to increasing the efficiency and accuracy of detection solutions.

A phishing attack is a method of cyber fraud in which attackers attempt to obtain sensitive information, such as login credentials, banking information, or personal data, by deception. They pose as trusted entities such as banks, companies, or institutions and mislead users into providing confidential data (Ahmad et al., 2024).

The stages of a phishing attack:

According to (Ahmad et al., 2024), there are four stages of a phishing attack:

- *Initiating the attack:* The hacker creates a fake email or message that appears to come from a legitimate source (e.g., bank, online service, or known company). The message usually contains a link to a fake website, presented as necessary to verify an account or update security information.
- *Victim trickery:* The target person, believing the message to be genuine, clicks on the link provided. This redirects them to a spoofed website that almost perfectly imitates the original page of the entity the attacker claims to represent.
- *Data collection:* On the fake website, the user is invited to enter sensitive information such as username, password, or bank details. Once this is completed, the data is passed directly to the attacker.
- *Exploiting stolen data:* After obtaining the information, the attacker can use it to access the victim's account or sell it on the black market. In this way, the user may suffer financial losses or compromise other online accounts.

The phishing process, as well as the four stages mentioned above, are illustrated in Figure 1.

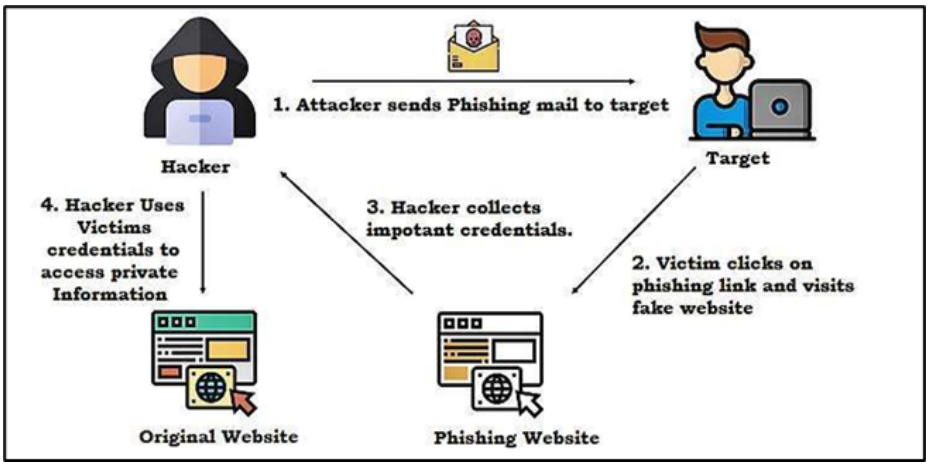


Figure 1. *The process of a phishing attack (Ahmad et al., 2024: p.1168)*

How do you recognize a phishing attack?

Phishing attacks are designed to mislead victims, but there are a few clues that can help identify them:

- *Messages suggesting urgent action:* Phishing emails try to rush the user to act quickly, citing security concerns or limited time opportunities.
- *Grammatical mistakes and unprofessional style:* Some messages may contain spelling mistakes or an unusual tone that does not match the tone of the actual institution.
- *Suspicious web addresses:* The links provided often lead to sites with domain names that do not exactly correspond to the official ones (e.g., „bank-security-update.com" instead of „bank.com").
- *Unusual requests:* A legitimate bank or company will never ask for passwords, PINs, or other sensitive data via e-mail.

How do we protect ourselves from phishing attacks?

According to (ENISA, 2020), there are several measures one can take to protect against phishing attacks:

- *Education and awareness:* Employees and ordinary users need to be trained to recognize suspicious emails.
- *Email security solutions:* Security filters can block dangerous messages before they reach users.
- *Device and networking monitoring:* Cybersecurity solutions can detect and stop phishing attempts.
- *Phishing attack simulations:* Organizations can test employee readiness through phishing recognition exercises.
- *Minimum necessary access principle:* Limited access to sensitive information reduces the risk of exploitation in the event of a successful attack.

Implementing these measures can greatly reduce the risk of users falling victim to phishing attacks, protecting both personal and organizational data.

To explore this topic deeper, several scientific papers investigating advanced phishing detection and prevention methods will be analyzed. These studies provide valuable insights into the technologies used, the performance of the security algorithms, and some effective defense strategies against cyber threats.

STATE OF THE ART

(Ahmad et al., 2024) explores the use of ML techniques to detect phishing attacks. The algorithms analyzed include Decision Tree (DT), Random Forest (RF), Logistic Regression (LR), Support Vector Machine (SVM), and XGBoost. The authors use an extensive dataset collected from various sources to test and compare the performance of these models.

According to the results, which are presented in Figure 2, RF demonstrated the highest accuracy, achieving a value of 96.4%. In contrast, the LR model performed the worst, with an accuracy of 93.94%. XGBoost achieved the highest K-fold score of 97.16%, but RF dominated in feature selection and hyperparameter optimization. The study also highlights the importance of data preprocessing, emphasizing that the removal of irrelevant features and the use of normalization techniques contribute significantly to improving model performance.

The performance analysis was made using metrics such as accuracy, ROC AUC score, and K-fold reliability. In particular, hyperparameter tuning and feature selection had a major impact on the performance of the models, optimizing phishing detection. The results suggest that the use of RF in combination with feature optimization and hyperparameter tuning is an effective solution for phishing detection at a high level of accuracy.

MODEL	K-FOLD	FEATURE SELECTION	HYPERPARAMETER TUNING
SVM	95.00	95.95	96.99
XBoost	97.16	97.99	98.11
RF	97.11	98.15	98.50
DT	97.12	97.67	97.32
LR	93.53	93.66	93.80

Figure 2. *Algorithms' results: comparing the five models, the conclusion is that RF performed the best (Ahmad et al., 2024: p.1172)*

(Omari, 2023) compares the performance of several ML algorithms in detecting phishing websites. Among the models analyzed are SVM, DT, LR, Neural Networks (NN), and RF. The author uses a dataset of over 10,000 URLs to evaluate the performance of each model in terms of accuracy, sensitivity, and specificity.

The experimental results highlight that the RF model performed best in terms of accuracy and sensitivity, outperforming the SVM-based models. The RF demonstrated superior performance due to its ability to correctly handle unbalanced data and efficiently extract features relevant for phishing detection.

SVM also performed well in terms of accuracy and sensitivity due to its ability to separate classes by an optimal hyperplane. DT showed a higher rate of misclassifications compared to other more advanced models, being more prone to overfitting. LR achieved acceptable results but performed worse compared to SVM and NN. Its accuracy was affected by the model's limitation in capturing the complex relationships between the features used for classification. Although NN has a high potential due to its ability to detect complex patterns, its performance was significantly influenced by the hyperparameter setting and the availability of a large dataset. An optimization of this model requires considerable computational power.

The article also emphasizes the importance of feature selection, demonstrating that the use of advanced techniques such as Principal Component Analysis (PCA) can significantly improve model performance by reducing dimensionality and eliminating redundant features.

In conclusion, the study highlights that the RF is the best-performing model for detecting phishing websites, with high accuracy and good generalizability, as it is depicted in Figure 3. The author also recommends applying advanced feature selection techniques to optimize the performance of the machine learning models.

Classifier	Accuracy	F1 score	Recall	Precision
Gradient Boost	97.2%	96.9%	97%	96.8%
Random Forest	97.1%	97.3%	97.4%	97.2%
Decision Tree	96.3%	96.7%	96.7%	96.6%
K-Nearest Neighbors	95.6%	96.2%	96.8%	95.7%
Support Vector Machine	93.9%	95%	96.4%	93.7%
Logistic Regression	92.7%	93.8%	95%	92.7%
Naive Bayes Classifier	60.1%	45.3%	29.3%	99.2%

Figure 3. *Results of the seven algorithms chosen for the research, Gradient Boost and Random Forest being the best-performing in terms of accuracy (Omari, 2023: p.423)*

(Kavya & Sumathi, 2025) present the latest methods and advances in detecting phishing attacks, highlighting emerging trends in the field. The authors analyze the use of the convolutional neural networks (CNNs) and recurrent neural networks (RNNs) for classifying phishing emails, demonstrating that these models can outperform the traditional machine learning-based methods. The techniques discussed include the use of transfer learning and pre-training of the models on large datasets to improve attack detection.

The paper also explores the use of the hybrid models, which combine deep learning techniques with conventional methods to obtain more robust results. The results of the

study indicate that CNN and RNN models can detect subtle patterns in phishing emails, with an accuracy of over 98% in the best cases. The study highlights that while deep learning (DL) models offer high accuracy, they are computationally more expensive and require large datasets for training.

Another important aspect discussed in the paper is the optimization of the hyperparameters and the use of data augmentation techniques to improve model performance.

In terms of ML algorithms, the study shows that the RF model performed the best in terms of accuracy (95.3%), precision (94.5%), and F1-score (94.1%). In comparison, the SVM model had an accuracy of 94.1% and LR performed the worst with an accuracy of 93.5%. These results highlight the effectiveness of the RF in detecting phishing attacks due to its ability to handle complex data and minimize the false-positive rate (FPR of 5.1%). The comparison between models can be observed in Figure 4.

The authors conclude that although DL models are promising for phishing detection, their large-scale implementation requires significant computational resources and a well-defined strategy for continuous model updating.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	FPR (%)	AUC	Training Time (s)
Logistic Regression	93.5	92.1	90.4	91.2	7.2	0.93	12
Random Forest	95.3	94.5	93.8	94.1	5.1	0.95	20
Support Vector Machine	94.1	93.0	92.4	92.7	6.4	0.94	34

Figure 4. *The three models have similar accuracies; RF is promising for phishing detection (Kavya & Sumathi, 2025: p.34)*

(Shombot et al., 2024) presents the development of an application based on SVM for detecting phishing attacks. The authors implement a classification model using a dataset consisting of legitimate and phishing URLs, evaluating the performance of the model by metrics such as accuracy, sensitivity, and F1 score.

The results are presented in Figure 5 and show that the Radial Kernel SVM (RBF) performed highly, but did not outperform the Polynomial Kernel SVM in terms of accuracy (84%). The latter seems to be better adapted to learn the complex data bounds of the phishing set, which explains why the Polynomial Kernel SVM outperformed.

The authors also emphasize the importance of selecting relevant features. This is a crucial step as ML models can be very sensitive to the selected features. Choosing the right set of features (such as URL structure, domain signature, or other features specific to phishing messages) can significantly influence the accuracy of the classification models.

The paper also discusses the challenges encountered in training the model, including the dataset imbalance. In conclusion, the authors propose improving the model by using semi-supervised learning techniques and integrating it into a real-time protection system against phishing attacks.

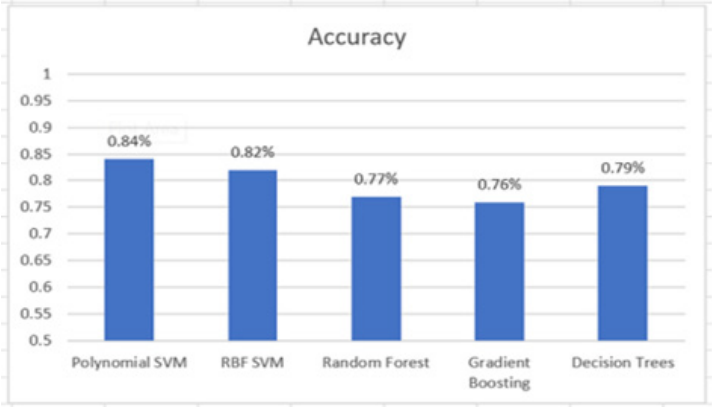


Figure 5. *The bar graph illustrates the accuracies obtained for each algorithm, concluding that the Polynomial SVM performed best (Shombot et al., 2024: p.5)*

CASE STUDY

The assumed purpose is to develop an efficient model for identifying phishing emails, based on advanced ML techniques.

This case study will analyze the detection of phishing attacks by applying ML algorithms on the **Phishing Email Dataset** (Al-Subaiey et al., 2024) available on the **Kaggle** platform. This dataset combines information from multiple sources to provide a comprehensive analysis resource.

The initial data come from several sets:

- **Enron** and **Ling,** including the content of the emails (subject, message body, and spam/ legitimate label).
- **CEAS, Nazario, Nigerian Fraud** and **Spam Assassin,** which provide contextual information such as sender, recipient, and the date of the message.

The final dataset integrates this information into a unified resource containing approximately **82,500 emails**, of which **42,891** are **spam** (phishing) messages and **39,595** are l**egitimate messages.**

The analysis of this dataset will allow the identification of the specific characteristics of the phishing messages and the development of effective detection models, thus contributing to the improvement of the cybersecurity strategies. Regarding ML methods, the study will focus on the use of SVM, RF, and LR algorithms.

SVM is an ML algorithm used for classification and regression, with high performance in identifying complex patterns. It works by finding an optimal hyperplane that separates data into distinct categories. In the case of phishing detection, SVM can classify emails or web pages as legitimate or fraudulent based on features extracted from the content. As it is shown in Figure 6, the maximum margin hyperplane and the two support vectors are the key concepts in separating the two categories, which in our case are legitimate emails and phishing emails.

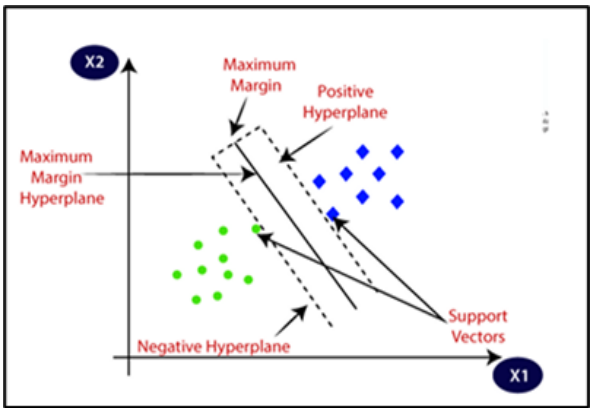


Figure 6. *Algorithm - SVM (Omari, 2023: p.418)*

RF is a decision tree-based learning method and works by combining multiple trees to improve the classification accuracy. Each tree contributes to the decision, and the result is determined by a majority vote, the process being highlighted in Figure 7. RF is effective in detecting phishing attacks because it can analyze a large number of features (Koehrsen, 2018).

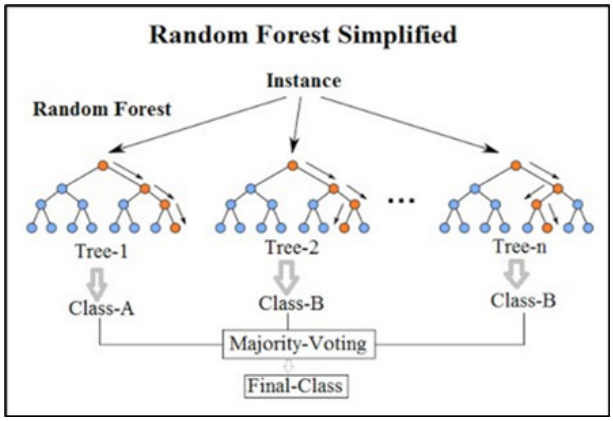


Figure 7. *Algorithm - Random Forest (Koehrsen, 2018)*

LR is a statistical model used for binary classification. The algorithm estimates the probability that a particular observation belongs to a specific class using a logistic function, as it is shown in Figure 8. In the context of phishing, it can help determine the likelihood that an email or website is fraudulent, based on predefined factors.

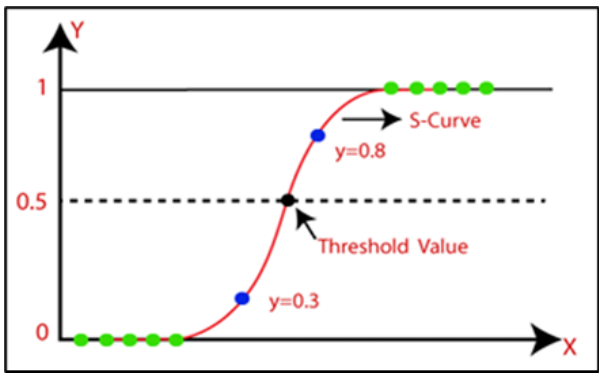


Figure 8. *Algorithm - Logistic Regression (Omari, 2023, p.418)*

To clearly illustrate the objectives pursued, a flowchart highlighting the main stages of the study has been developed and is presented in Figure 9.

Figure 9. *Flowchart - the main steps of the data set analysis*

Thus, on the dataset, the following steps will be performed: data collection and preprocessing, extraction and selection of relevant features, division of the dataset into training and test sets, followed by testing and evaluation of the applied models. It will also involve measuring the performance of each model, comparing them to identify the most efficient approach, and finally drawing conclusions based on the results obtained.

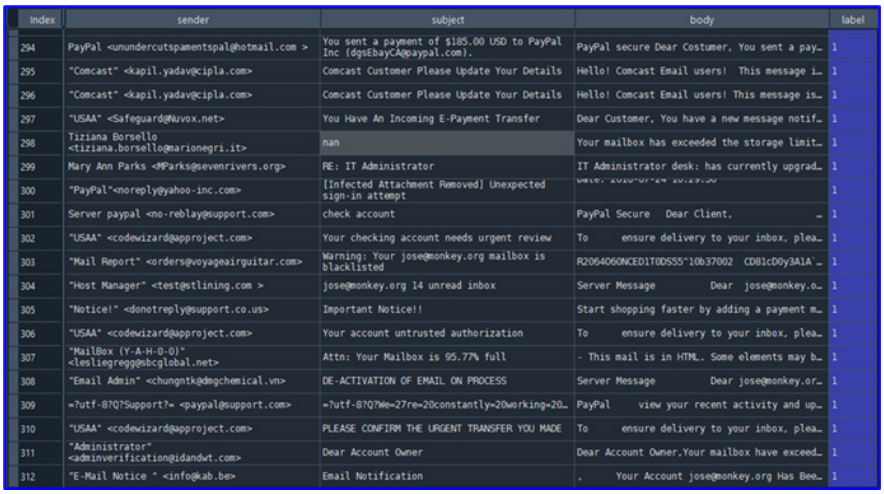
The codes used to pre-process the database and obtain the results from the training and testing processes corresponding to each algorithm can be accessed on the GitHub platform. LLM models were used for the preliminary generation of code fragments, which were subsequently reviewed for accuracy and functionality.

In order to build a solid foundation in training and testing ML models for phishing detection, we went through several key steps on the dataset. Initially, we selected four relevant datasets from public sources (Kaggle) - CEAS08, Nazario, Nigerian Fraud, and Spam Assassin - that had similar structures and included the columns „sender", „receiver", „data", „subject", „body", „urls", and „label". These sources were chosen because of their diversity of emails and valuable features for phishing analysis. Subsequently, after unifying these sets, we performed data preprocessing to optimize them for model training and evaluation.

- **Reading the data:** Each data set was loaded into the corresponding variables using pd.read_csv().
- **Merging data:** pd.concat() was used to merge the datasets into a single DataFrame.
- **Data cleaning:** It was observed that some columns, such as „receiver", „data", and „urls", do not have a significant impact on the model performance, so they were removed from the final dataset.
- **Selection of relevant characteristics:** the columns „sender", „subject", „body", and „label" were kept, indicating whether the email was spam or legitimate. These characteristics are essential for phishing detection and pattern training.
- **Exporting the final set:** The processed dataset was saved in a .csv file for further use in the model analysis.

Thus, the saved file contains the preprocessed data needed to train the three algorithms analyzed: SVM, RF, and LR. This file will be further used for testing and evaluating the performance of these models in detecting phishing attacks.

A preliminary visual analysis revealed the presence of missing values (NaN) in the dataset, which emphasized the need for additional treatment before the training process started. It can be observed in Figure 10 that for the entry at the index of 298, 'subject' has a value of 'NaN'.



Index	sender	subject	body	label
294	PayPal <ununderсutspamentspal@hotmail.com >	You sent a payment of $185.00 USD to PayPal Inc (dgsEbayCA@paypal.com).	PayPal secure Dear Costumer, You sent a pay…	1
295	"Comcast" <kapil.yadav@cipla.com>	Comcast Customer Please Update Your Details	Hello! Comcast Email users! This message i…	1
296	"Comcast" <kapil.yadav@cipla.com>	Comcast Customer Please Update Your Details	Hello! Comcast Email users! This message is…	1
297	"USAA" <Safeguard@Nuvox.net>	You Have An Incoming E-Payment Transfer	Dear Customer, You have a new message notif…	1
298	Tiziana Borsello <tiziana.borsello@marionegri.it>	nan	Your mailbox has exceeded the storage limit…	1
299	Mary Ann Parks <MParks@sevenrivers.org>	RE: IT Administrator	IT Administrator desk: has currently upgrad…	1
300	"PayPal"<noreply@yahoo-inc.com>	[Infected Attachment Removed] Unexpected sign-in attempt		1
301	Server paypal <no-reblay@support.com>	check account	PayPal Secure Dear Client, …	1
302	"USAA" <codewizard@approject.com>	Your checking account needs urgent review	To ensure delivery to your inbox, plea…	1
303	"Mail Report" <orders@voyageairguitar.com>	Warning: Your jose@monkey.org mailbox is blacklisted	R2064O6ONCED1T0DSS5^10b37002 CD81cD0y3A1A`…	1
304	"Host Manager" <test@stlining.com >	jose@monkey.org 14 unread inbox	Server Message Dear jose@monkey.o…	1
305	"Notice!" <donotreply@support.co.us>	Important Notice!!	Start shopping faster by adding a payment m…	1
306	"USAA" <codewizard@approject.com>	Your account untrusted authorization	To ensure delivery to your inbox, plea…	1
307	"MailBox (Y-A-H-0-0)" <lesliegregg@sbcglobal.net>	Attn: Your Mailbox is 95.77% full	- This mail is in HTML. Some elements may b…	1
308	"Email Admin" <chungntk@dmgchemical.vn>	DE-ACTIVATION OF EMAIL ON PROCESS	Server Message Dear jose@monkey.or…	1
309	=?utf-8?Q?Support?= <paypal@support.com>	=?utf-8?Q?We=27re=20constantly=20working=20…	PayPal view your recent activity and up…	1
310	"USAA" <codewizard@approject.com>	PLEASE CONFIRM THE URGENT TRANSFER YOU MADE	To ensure delivery to your inbox, plea…	1
311	"Administrator" <admnverification@idandwt.com>	Dear Account Owner	Dear Account Owner,Your mailbox have exceed…	1
312	"E-Mail Notice " <info@kab.be>	Email Notification	, Your Account jose@monkey.org Has Bee…	1

Figure 10. *Sample of the dataset before preprocessing and removing the NaN entries*

In order not to influence the training process, it was decided to remove these lines, and a database with 49442 lines and 4 columns was obtained. The next step was to process the data contained in the „subject", „body", and „sender" columns. For this purpose, there was implemented a function that transforms all letters into lowercase letters, removes special characters, digits, URLs, and some stop words, which are illustrated in Figure 11.

Type	Size	Value
str	2	in
str	4	this
str	6	shan't
str	4	than
str	5	under
str	8	mightn't
str	4	didn
str	4	they
str	5	where
str	3	off
str	3	did
str	7	against
str	3	own
str	2	am

Figure 11. *Examples of stop words that will be removed from all texts*

This function was then applied to the „text_cleaned" column, which was added to the database, and which represents the concatenation of the three columns, „subject", „body", and „sender". This was followed by the step of defining feature (X) and label (y) vectors, where X is a column vector containing all the entries in the „text_cleaned" column, and y is a column vector containing the labels (the entries in the „label" column).

As the three proposed models cannot receive string data as input, it was necessary to implement a transformation into numerical data using the Term Frequency - Inverse Document Frequency (TF-IDF) Vectorizer. It calculates a score for each word, so that

common words have a lower score while rare and important words (keywords) have a higher score. TF gives the frequency of the word in a document, and IDF suggests how rare the word is in all documents, in this case, in all emails.

The parameters in the TfidfVectorizer function have the following meanings:

- ***Stop_words*** – remove common words;
- ***Max_features*** – store the 5000 most frequent terms;
- ***Ngram_range*** – consider individual words (1,) or consecutive word pairs (,2);
- ***Min_df*** – ignore words that appear in too few texts.

Using the Pickle library, the vectorizer was saved so that it could later be applied to the test database. The next step, once the data processing was completed, was to split the data into training and testing data, using the train_test_split function, in a proportion of 80% train and 20% test.

Training and testing the SVM model

Regarding the SVM model training, initially, the Grid Search method was used to identify the best combination of parameters to return the best performing model, and it was concluded that an SVM model with kernel „rbf", regularization parameter C=10, and gamma = „scale" (automatic computation) would be the best. However, the test did not achieve more than 80% accuracy, so the regularization parameter was varied and set to C=1.

The SVM model was trained on the training data (X_train - y_train pairs) with the fit function, predictions were made on the test data (X_test), and evaluated by calculating the accuracy and confusion matrix, which is based on comparing the vector y_test (the actual label values) with the vector y_pred (the values predicted by the previously trained model). A training accuracy of 99.53% was obtained.

The model was saved using the Pickle library, and another script was created to test the SVM. A new dataset was chosen for testing from public sources, named TREC_05 (Champa, Rabbi & Zibran, 2024), from which 1,000 samples were randomly selected. In order for the testing process to be correct, it was necessary to perform the same database processing steps as for training. Thus, after loading the database, the „subject", „body", „sender", and „label" columns were selected, and the texts were processed to remove linking words, digits, special characters, urls. Then, the feature vector X and the label vector y were created. The previously saved vectorizer was loaded and applied to the vector X using the transform function.

The last step was loading the trained SVM model, which made predictions on the test feature set. The obtained labels were compared with the actual labels from the test database, the evaluation obtained 82.3% accuracy and the confusion matrix from Table 1 was generated.

Table 1. *Confusion Matrix after testing the SVM model*

266	144
33	557

Training and testing the LR model

Following the analysis process, an LR model was trained, one of the most commonly used methods in binary classification, especially for problems such as phishing attack detection. The model was instantiated with a specific set of parameters that influence the learning process: the hyperparameter C was set to 0.3, which indicates stronger regularization to prevent overlearning, class_weight was set to "balanced" so that the model gives proportional importance to both classes (phishing and legitimate) even under conditions of imbalance in the data distribution, and max_iter was set to 1,000 to allow the algorithm a sufficient number of iterations to converge. After model setup, the model was trained using the previously prepared dataset - X_train (feature vector) and y_train (corresponding label vector).

Following the training step, the model was tested on the test data, and the results

were evaluated by calculating the accuracy score and generating a confusion matrix. The training accuracy of 98.63% was displayed in the console, together with the confusion matrix, giving a detailed picture of the model's performance. Finally, the trained model was saved, allowing its later use in predictions on a new dataset to evaluate the ability of the model to generalize to unknown data.

For the testing phase, a new dataset from public sources, known as TREC_05, was used, from which 1,000 representative samples were selected. In order to guarantee a correct and relevant evaluation of the trained models, the same preprocessing procedures previously used in the training phase were applied to this dataset. After loading the file, only the columns essential for the analysis - ‚subject', ‚body', ‚sender', and ‚label' - which provide information on the content and nature of the emails, were kept. The texts were then cleaned by removing irrelevant elements such as linking words, special characters, and web links (URLs) in order to obtain standardized and coherent content. After this step, the feature vector (X) and the label vector (y) were constructed and served as the basis for testing. To keep consistency with the training process, the previously saved vectorizer was used, which was applied to the vector X by the `transform` method, thus ensuring the same numerical representation of the data and allowing an accurate evaluation of the models' performance. This process was used to test each model in order to make a valid comparison of the results.

The model performance was evaluated by calculating the accuracy and the confusion matrix. The accuracy obtained, 83.83%, gives a general picture of the accuracy of the predictions, while the confusion matrix (shown in Table 2) allows a more detailed analysis of the cases classified correctly or incorrectly in each class. This external testing process confirms not only the validity of the model but also its ability to perform efficiently in real-life contexts, thus contributing to a robust automated detection of phishing emails.

Table 2. *Confusion Matrix after testing the LR model*

283	117
41	536

Training and testing the RF model

In the classification experiment, we also used the RF algorithm, a decision tree-based method that offers high robustness and accuracy in binary classification problems such as phishing email detection. The model was trained on the previously processed dataset using the fit function on the training pairs X_train and y_train. After preprocessing, the text was transformed numerically by TF-IDF vectorization using a TfidfVectorizer.

For performance optimization, the RandomizedSearchCV technique was applied, which searches for the best combinations of hyperparameters in a predefined space. Among the parameters included are: number of trees, maximum depth, minimum number of examples for splitting, and class weighting.

The initialization of the model parameters was performed as shown below:

- n_estimators =100 (number of decision trees);
- class_weight =’balanced’ – to compensate for possible imbalances between classes;
- max_depth = None – avoid small values that can cause overfitting;
- min_samples_split = 2,
- min_samples_leaf =2,
- random_state = 42 for reproducibility.

After training, the model's performance was evaluated on the test set, yielding an accuracy of 99.12%, together with a detailed confusion matrix reflecting a very low misclassification rate, with only 38 negative and 49 positive samples misclassified.

The trained model, together with the TF-IDF vectorizer, were saved locally using the Pickle library to be reused in the external testing phase.

To validate the model's ability to generalize to new data, an external set, called TREC_05,

containing 1,000 samples, was used. The database was subjected to the preprocessing steps as in the training phase: selection of the essential columns (sender, subject, body, label), textual cleaning, and concatenation into a single text_cleaned field.

After preprocessing, the text was transformed using the same previously saved TF-IDF vectorizer to ensure consistency of the numerical representation. The trained RF model was loaded and applied to the new data, generating the y_ext_pred predictions.

The performance evaluation on this set was performed by overall accuracy and a new confusion matrix (shown in Table 3), indicating the model's true ability to detect phishing emails in unknown contexts. The results confirmed a modest performance with 75.8% accuracy.

Table 3. *Confusion Matrix after testing the RF model*

255	155
87	503

RESULTS

For the automatic detection of phishing messages, three classification models were tested and compared: LR, SVM, and RF. The performance evaluation of each model was conducted through a detailed analysis of the following evaluation metrics: accuracy, precision, sensitivity, F1 score, and specificity - each providing key insights into how well the model addresses the specific challenges of phishing detection.

The **LR model** was characterized by a solid balance between all metrics analyzed. It achieved an **accuracy of 83.83%**, signaling a high proportion of correct predictions overall. It is important to note that accuracy, although a useful indicator of overall performance, can be affected by imbalances in data, which is why it is complemented by other metrics that are more sensitive to misclassification.

In terms of **precision (82.07%)**, the model demonstrated a high ability to issue correct alerts, i.e., to reduce the number of false positive alarms, legitimate emails misclassified as phishing. This feature is vital in practice to maintain user confidence in the detection system.

At the same time, the **sensitivity** (recall) of 92.89% reflects the model's ability to correctly detect the majority of phishing messages, minimizing false negative errors - i.e., malicious emails that would go undetected. The **F1 score** of 87.15% balances accuracy and sensitivity and confirms the robustness of the model in both directions. In addition, the **specificity** of 70.75% shows that almost three-quarters of the legitimate emails are correctly recognized, an important result for reducing interference in the normal communication flow.

The **SVM model** achieved a **superior sensitivity of 94.41%,** which makes it extremely effective in fully detecting phishing messages. This high value is essential in scenarios where the top priority is not to miss any real threats. However, the **79.46% precision** suggests a slight increase in the number of false alarms - a trade-off commonly found in very high sensitivity models. The overall accuracy was 82.30% and the **F1 score** was 86.29%, both very close to those of the logistic regression. The **specificity** of 64.88% indicates, however, a more pronounced tendency to label legitimate emails as phishing, which may influence the user experience.

In contrast, **RF** performed the most modest of the three. Its **accuracy of 75.80%** and **precision of 76.44%** suggest a lower capability in both correctly detecting phishing and maintaining a low level of false alarms. Its **sensitivity** of **85.25%** is significantly lower than the other models, meaning it is more likely to fail to identify all malicious messages. The **F1 score** of **80.61%** and **specificity** of **62.19%** reinforce the idea that the RF model needs further fine-tuning to achieve the performance of the other two models.

In conclusion, **LR provides the best compromise** between complete threat detection and false alarm reduction, thus being the most balanced and applicable model in real contexts. **SVM** is particularly valuable in environments where no phishing messages are allowed to escape,

even at the risk of increasing false alarms. **RF**, although promising, requires optimization to become competitive with the other two models in this application. For a better comparison of the three models that were analyzed during this study, Table 4 was created.

Table 4. *Comparative table of model performance*

Model	Accuracy	Precision	Sensitivity	F1 score	Specificity
Logistic Regression	83.83%	82.07%	92.89%	87.15%	70.75%
SVM	82.30%	79.46%	94.41%	86.29%	64.88%
Random Forest	75.80%	76.44%	85.25%	80.61%	62.20%

CONCLUSIONS AND FUTURE WORK

This paper discusses how recent research has highlighted the applicability and effectiveness of the machine learning techniques in the detection of phishing messages - an increasingly sophisticated form of cyber-attack, with a major impact on the personal and organizational data security. By designing, training, and benchmarking three distinct models, Support Vector Machine, Logistic Regression, and Random Forest, significant differences in their ability to meet the challenges posed by the automated detection of malicious messages were observed.

The best performing model turned out to be Logistic Regression, which achieved a high level of accuracy and an optimal balance between sensitivity and precision, indicating a solid ability to identify most phishing messages while maintaining a reasonable level of false alarms. In contrast, the SVM and Random Forest models, although showing promising results in the training phase, were more affected by overfitting and had a weaker generalization ability on the real data.

Although Random Forest obtained very good results on the training set, its performance decreased significantly when it was applied to the external data set. This difference indicated that there was a tendency for overfitting, the model learning very well the specific characteristics of the initial set, but failed to generalize when the data came from a different distribution. In the particular case of phishing detection, where messages can vary greatly in structure, vocabulary, and context, this limitation represents an operational risk. Unlike Random Forest, Logistic Regression and SVM models showed a better ability to maintain performance in the face of these variations, so they may be more suitable for practical scenarios.

To avoid conclusions based only on accuracy or sensitivity values, it would be useful to perform a comparative statistical analysis between the models. Such an approach would allow determining the degree to which the performance differences are significant and not just the result of the random variations in the data. In addition, integrating statistical tests would strengthen the arguments regarding the choice of the most appropriate algorithm and validate the results obtained.

The integration of a Logistic Regression model into an information security system offers concrete advantages:

- Automatic and fast detection of malicious emails, reducing the response time to a threat;
- Limits users' exposure to dangerous content by effective filtering before messages reach the inbox;
- Reduced human effort to verify alerts due to high model accuracy;
- Scalability and ease of implementation, being a model with low complexity and minimal computational costs.

In an ever-changing digital landscape, where attackers are constantly refining their methods, the use of robust, balanced, and validated machine learning models becomes a fundamental component of the cybersecurity architecture. The present study confirms that, by selecting a well-calibrated algorithm tailored to the specific application, a high level of proactive security can be achieved, with a direct impact on reducing risks and effectively protecting users and digital infrastructures.

REFERENCE LIST

Ahmad, S.K., Dapshima, B., Essa, Y., Dapshima, B.A. & Chuupa, Y. (2024) *Detection of phishing attacks using machine learning techniques*. doi:10.56726/IRJMETS60054.

Al-Subaiey, A., Al-Thani, M., Alam, N.A., Antora, K.F., Khandakar, A. & Zaman, S.A.U. (2024) N*ovel Interpretable and Robust Web-based AI Platform for Phishing Email Detection*. doi: https://doi.org/10.1016/j.compeleceng.2024.109625

Champa, A.I., Rabbi, M.F., & Zibran, M.F. (2024) Curated datasets and feature analysis for phishing email detection with machine learning. *3rd IEEE International Conference on Computing and Machine Intelligence (ICMI)*. 1–7. doi: 10.1109/ICMI60790.2024.10585821

ENISA Phishing. (2020) - *Raportul ENISA privind situația amenințărilor RO.*

Kavya, S. & Sumathi, D. (2025) Staying ahead of phishers: a review of recent advances and emerging methodologies in phishing detection. *Artificial Intelligence Review*. 58 (2). doi:10.1007/s10462-024-11055-z.

Koehrsen, W. (2018) Random Forest Simple Explanation. *Medium.*

Margarit, B. (2025) Detecting Phishing Attacks using ML. *GitHub repository*. [Accessed 28th August 2025]. https://github.com/biia0115/Detecting_Phishing_Attacks_using_ML

Omari, K. (2023) Comparative Study of Machine Learning Algorithms for Phishing Website Detection. *IJACSA - International Journal of Advanced Computer Science and Applications*. doi:10.14569/IJACSA.2023.0140945

Shombot, E.S., Dusserre, G., Bestak, R. & Ahmed, N.B. (2024) An application for predicting phishing attacks: A case of implementing a support vector machine learning model. *Cyber Security and Applications*. 2. doi:10.1016/j.csa.2024.100036.