WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Spam Email Detection Using Machine Learning

Dr. Akhil Pandey[1], Dr. Ashok Kumar Kajla[2], Dr. Vishal Shrivastava[3], Charitra samadiya[4]

Department of Information Technology Student of Information Technology, Arya College of Engineering and IT, Kukas, Jaipur

ABSTRACT:-

The following ML models are examined and evaluated in this study: Naive Bayes is a probabilistic classifier that is frequently utilized in spam filtering for the classification of text. Support Vector Machines (SVM) are a high-performance classifier that use hyperplane-based learning to distinguish between legitimate and spam emails. Random Forest and Decision Trees are two ensemble learning techniques that combine multiple decision trees to enhance classification accuracy. Long Short-Term Memory (LSTM) Networks is a deep learning model that is better than traditional machine learning models at processing sequential data, understanding context, and identifying spam emails. The Enron Spam Dataset and the SpamAssassin Dataset, two publicly accessible datasets, are utilized in the study to train and evaluate machine learning models. Some of the performance metrics that are used in the research to evaluate these models are the F1-score, accuracy, precision, recall, and ROC- AUC score. Deep learning models (LSTMs) outperform conventional machine learning models in terms of accuracy and robustness when it comes to spam classification. However, despite ML-based spam detection's success, there are still a few obstacles and limitations: Spam attacks by adversaries in which the content of emails is altered to evade detection. high rates of false positives, which can cause legitimate emails to be mistaken for spam. overhead for computation, particularly in deep learning models that call for a lot of processing power. Dataset bias and class imbalance, which can affect the generalization of spam detection models.

Real-time adaptive learning, transformer-based NLP models like BERT and GPT, and hybrid spam detection systems that combine rule-based methods with AI-driven classifiers are among the study's suggested future enhancements. These advancements can further enhance the efficiency of spam detection mechanisms, making them more effective against emerging cyber threats.

Machine learning, spam detection, Naive Bayes, SVM, Random Forest, LSTM, deep learning, phishing, cybersecurity, and spam filtering are all key terms.

INTRODUCTION

With billions of emails being exchanged each day for personal, professional, and business reasons, email communication is an essential component of modern digital communication. However, a significant portion of this email traffic is made up of spam emails, which are unsolicited, irrelevant, or malicious messages that frequently pose security risks. Spam emails can be used for advertising, spreading malware, phishing attacks, and financial fraud, making spam detection a crucial aspect of cybersecurity.

Traditional rule-based spam filters, which rely on manually defined rules like keyword matching and blacklisting, are becoming less effective as a result of the ever-changing nature of spam emails. Cybercriminals constantly develop new methods to circumvent these filters, including sophisticated phishing techniques, spam content embedded within images, and obfuscation. As a result, machine learning (ML) has emerged as a powerful solution for spam detection, offering the ability to learn patterns from data and improve filtering accuracy over time.

Email content, sender behaviour, and metadata can all be used by machine learning models to distinguish between spam and legitimate (ham) emails. ML-based systems can adapt to new spam trends, making them more robust and effective at filtering out malicious emails than traditional spam detection methods. In order to evaluate their efficacy in detecting spam email, this study looks into various machine learning models like Naive Bayes, Support Vector Machines (SVM), Decision Trees, Random Forest, and Long Short-Term Memory (LSTM) networks.

Literature Review

From rule-based filtering to advanced machine learning (ML) and deep learning methods, spam detection has advanced significantly over time. This section of the review of existing research on these approaches highlights the advantages of ML- based techniques as well as the advantages and disadvantages of conventional spam filtering methods.

1.1 Traditional Techniques for Filtering Spam

Traditional Techniques for Filtering Spam Pre- defined rules and hand-crafted methods were used in traditional spam filtering to identify unwanted emails. These techniques were effective for early spam detection, but they were unable to keep up with the evolution of spam over time.

Keyword-based filtering, which flagged emails containing phrases like "win a lottery" and "free money" as spam, was one popular method. Although simple, spammers easily bypassed these filters using obfuscation techniques like modifying text (e.g., “Fr€€ MoN€Y”) or embedding spam content within images.

Another common strategy was the blacklist and whitelist approach, which allowed emails from trusted contacts but blocked those from known spammers. However, this tactic was ineffective against brand-new spammers who frequently spoof their email addresses.
Rule-based filters were also used a lot, which found spam based on things like too many links, suspicious sender domains, and weird email structures. These filters could be changed, but in order for them to be effective, they needed to be updated on a regular basis and manually manipulated.
The following are the primary drawbacks of conventional spam filtering techniques:
Static and non-adaptive: Requires manual rule updates to detect new spam patterns. High Rate of False Positives: Numerous legitimate emails are erroneously marked as spam. fails to withstand image-based spam, phishing, and text obfuscation. Possibility of Being Attacked by More Expensive Spam

1.2 Machine Learning Approaches

Machine learning (ML) has revolutionized spam detection by introducing data-driven models that automatically learn from email datasets and classify messages based on patterns and statistical relationships. In contrast to conventional filters, ML-based systems can adapt to shifting spamming tactics without much human intervention. Supervised learning models like Naive Bayes, Support Vector Machines (SVM), and Decision Trees have been extensively utilized in the field of spam detection. Naïve Bayes applies a probabilistic approach based on word frequencies, making it fast and computationally efficient. However, the accuracy of real-world spam detection is limited by its
assumption of feature independence. SVM, on the other hand, uses a hyperplane to distinguish
between spam and legitimate emails, providing high accuracy in text classification tasks. Despite its excellent performance, SVM is computationally expensive for large datasets. By combining multiple decision paths, Decision Tree and Random Forest models reduce the risk of overfitting and improve classification accuracy. Large labeled datasets are required for training, despite these models' effectiveness and robustness. Spam detection has also been studied using unsupervised learning methods like clustering algorithms like K-Means and DBSCAN. These models do not rely on labeled data and attempt to group similar emails together. However, when it comes to classifying individual emails as spam or ham, they lack precision. The accuracy of spam detection has been significantly improved by the most recent developments in deep learning, particularly Long Short-Term Memory (LSTM) networks. Because LSTMs are able to interpret contextual meaning in email text, they are effective against advanced spam and phishing techniques. Additionally, Convolutional Neural Networks (CNNs) have been applied to detect image-based spam, a challenge that traditional methods fail to address.

1.3 Recent Advances in Spam Detection

Latest Developments in Spam Detection Through the use of hybrid models, adversarial learning, and transformer-based NLP techniques, recent research has focused on improving spam detection. Hybrid spam detection systems combine the advantages of rule-based filtering and machine learning models. To increase detection accuracy, Gmail's spam filter, for instance, incorporates AI-based classifiers with manual user flagging. Adversarial machine learning, in which spam detection models are trained to identify manipulated emails intended to circumvent filters, is another significant development. Spam techniques are constantly improved by cybercriminals, necessitating dynamic adaptation of ML models to new threats. The introduction of transformer-based NLP models (BERT, GPT) has further improved spam filtering capabilities. Unlike traditional ML models that rely on word frequency analysis, transformers understand the context and semantic meaning of email text. Because of this, they are able to precisely identify obfuscated phishing attempts and spam messages. However, despite these advances, several challenges remain:

Costly Computing: Models based on deep learning require a lot of processing power. Emails are actively altered by adversarial spammers in order to evade detection. False Positives: Even highly developed models occasionally mistake legitimate emails for spam.

Methodology

The method used to implement and evaluate machine learning-based spam detection is described in the methodology section. This includes information about the datasets that were used, the methods for extracting features, the model that was chosen, and evaluation metrics. The study's scientific and reproducible approach to accurate and dependable spam detection is ensured by a well-structured methodology.

1.4 Dataset Selection

Choosing a Dataset To develop and test machine learning models for spam detection, publicly available datasets were used. These datasets provide real-world examples of spam and non- spam (ham) emails, allowing models to learn patterns and classify emails effectively.
One of the most widely used datasets for email classification research is the Enron Spam Dataset. Six Enron employees' emails were gathered as part of the Enron corporate email investigation. It contains over 35,000 emails, including both spam and ham messages. These emails are highly representative of actual email traffic because they were extracted from actual workplace communication, as opposed to synthetic datasets. However, this dataset has a drawback in that it is time-sensitive because it was created in the early 2000s, so it may not reflect current spam trends. The Spam Assassin Dataset is another widely used dataset, containing over 6,000 labeled spam and ham emails collected from various sources, including honeypots and mailing lists. One of its primary advantages is that this dataset contains a variety of spam, including phishing emails, advertisements, and scam messages. However, it is smaller in size compared to the Enron dataset, which can be a limitation when training deep learning models that require large amounts of data.
Before the models were trained on the datasets, they were pre-processed to get rid of duplicate emails, normalize text, get rid of stop-words, and tokenize words. Additionally, numerical features were used to analyse categorical data like email timestamps and sender domains.

1.5 Engineering of Features

Feature engineering is the process of extracting meaningful patterns from email data Machine learning models rely on these features to classify emails as spam or ham accurately.
Methods like Bag of Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF), and N-Grams were used to extract text-based features. These techniques helped convert email text into numerical vectors that machine learning models can process. Additionally, misleading or fraudulent language that is frequently used in spam emails was identified using sentiment analysis. Email length, subject line characteristics, and sender reputation were among the metadata features extracted from email headers. Emails from unknown or suspicious domains had a higher likelihood of being classified as spam. Subject lines with a lot of capitalization or phrases that are urgent (like "Act NOW!") were also considered strong spam indicators.
HTML content, embedded hyperlinks, and attachments were structural features. The hidden tracking pixels, excessive links, and large attachments found in many spam emails helped differentiate them from regular business emails. These structural attributes played a crucial role in spam detection models.

1.6 Model Training and Evaluation

Machine learning models were trained to classify spam emails after the dataset had been pre- processed and features had been extracted. This study utilized a number of classification algorithms, including: Naïve Bayes, a probabilistic classifier that works well for text-based spam detection due to its efficiency in handling large email datasets.
Support Vector Machines (SVM), which employ hyperplanes in a high-dimensional space to distinguish between spam and legitimate emails. Random Forest is a model of ensemble learning that improves decision tree generalization and accuracy. Long-Term and Short-Term
Because they look at sequential dependencies in text, memory networks are especially useful for detecting spam in modern email systems. The accuracy, precision, recall, F1-score, and ROC- AUC score were utilized to assess the performance of the model. Accuracy measured the overall correctness of the model, while precision and recall assessed the model's ability to detect spam while minimizing false positives and false negatives. The model's ability to distinguish ham from spam emails was evaluated using the ROC-AUC score, and the F1-score was used to strike a balance between recall and precision

Data Validation and Testing

Data validation and testing play a critical role in ensuring that machine learning models for spam detection generalize well to new, unseen emails. A model may perform well on training data but fail when applied to real-world spam detection if it lacks proper validation. The methods used to validate and test the spam detection models are described in this section to guarantee their dependability, accuracy, and robustness.

1.7 Experimental Setup

Design of the Experiment To evaluate the spam detection models fairly, the dataset was divided into training and testing sets using a structured approach. The objective was to ensure that the model did not memorize particular non- generalizable patterns in the training data and to prevent data leakage. The experimental setup that was used was as follows:

Train-Test Split

The dataset was split into 80% training data and 20% testing data to ensure the model learned patterns from a majority of emails while being tested on new, unseen emails.
This split helps prevent overfitting, where the model memorizes specific spam patterns instead of learning general spam characteristics.

K-Fold Cross-Validation

K-fold cross-validation (K=10) was used to ensure robust evaluation.

The dataset was divided into 10 subsets, where the model was trained on 9 subsets and tested on 1 subset, repeating this process 10 times to ensure fairness. A more comprehensive model emerges as a result of this strategy, which ensures that all data points are utilized for both training and validation.

Avoiding Overfitting and Underfitting

Overfitting occurs when the model performs well with training data but poorly with new data, indicating that it has learned spam pattern patterns rather than general trends. Underfitting happens when the model is too simple to capture spam patterns, leading to poor accuracy.

To deal with these problems:

To stop models from giving certain words too much weight, regularization methods like L1 and L2 penalties were used. For deep learning models, early stopping was used to prevent unnecessary training after performance stopped improving.
To get rid of noisy, irrelevant features that could lead the model astray, feature selection methods were used.

id: The question identifier; helpful for understanding the request in which tasks are performed.
Select type: demonstrates the type of question (such as SUBQUERY, Straightforward, or Essential).
table: The table that will be accessed during this execution step.
type: The type of the join, which can be any one of a number of options like "ALL" (full table sweep), "record" (file examine), and so on.
possible keys: The records that MySQL might actually use to accelerate the question.
key: The actual record used in the question execution.
Key length: The length of the key utilized.
ref: Shows which segments or constants are contrasted and the key.
lines: The estimated number of columns that MySQL ought to analyse in order to carry out the inquiry. Extra: Provides additional information, such as whether a brief table or a file type is being used (the two are generally less effective).
Extra: Gives extra data, like whether a brief table or file sort is being utilized (the two of which are by and large less effective).

1.8 Evaluation Metrics

Criteria for Evaluation A variety of performance metrics were used to assess the spam detection models' efficacy. Each metric provides insights into how well the model distinguishes between spam and ham emails.

Accuracy

Measures how well the model classifies emails correctly.
A high accuracy may be misleading if the dataset is imbalanced (for example, there are more spam emails than ham emails).

Precision

Evaluates how many emails that were predicted as spam were actually spam.
A high precision means fewer false positives (legitimate emails incorrectly classified as spam).

Recall

Measures how many actual spam emails were correctly detected by the model.
A high recall indicates that the model is successful in identifying the majority of spam emails, though it may also identify some legitimate ones as spam.
F1-Score
Provides a balance between precision and recall.
Ensures that the model is not biased toward detecting only one class.

ROC-AUC Score

identifies the model's capacity to differentiate between legitimate and spam emails.
A higher AUC score indicates a more reliable spam detection model.

Full Table Outputs: A full table sweep is displayed if the "type" is "ALL," and this typically takes longer for large tables. In such cases, including a list the segments utilized in the WHERE statement might accelerate the question.
Utilizing Records: The key section shows whether MySQL is utilizing lists. On the off chance that it isn't utilizing a file (i.e., the key segment is Invalid), then making the right records or changing the question can assist with upgrading execution.

Join Request: The request for joins can have an impact on the entire execution. If Make sense of shows that MySQL is getting to tables in a less productive request, think about changing the question or adding fitting records to guarantee the enhancer can choose the best arrangement.

Transitory Tables and Filesort: Assuming the Additional segment shows "Utilizing brief" or "Utilizing filesort", it demonstrates that MySQL is playing out extra tasks like arranging or putting away moderate outcomes in an impermanent table. Because these tasks can be expensive, it's worth looking into whether different question designs or files could avoid using these methods.

1.9 Extended Validation Techniques and Real- World Testing

Real-World Testing and Additional Validation Methods To ensure that the spam detection models work in real-world situations, additional validation methods were used. Real-time testing, human verification, and simulated adversarial attacks were used in addition to standard training and testing to verify the robustness of the models. Testing in real time with live email data The trained models were tested with a set of live emails from a variety of sources, including personal inboxes and corporate accounts. The objective was to determine the accuracy with which the models classified emails that were not included in the initial dataset. This made it possible to find potential flaws, like how the training data couldn't find spam from more recent years. Spam Classification Verification by Humans During a manual review process, emails that were deemed spam were evaluated by human evaluators. The team

was able to assess how well the model was able to distinguish between legitimate emails and spam thanks to this. The model's learning process was enhanced by flagging and analysing any emails that had been misclassified. Testing Against Adversarial Spam Attacks
In order to evade spam filters, cybercriminals frequently employ adversarial methods like modifying words, inserting hidden characters, or using image-based spam. The models were tested against intentionally manipulated spam emails to see how easily they could be bypassed.
This testing highlighted areas where the models needed improvement, especially in handling obfuscated text and phishing emails with deceptive sender addresses.

System Performance Evaluation

Evaluation of the System's Performance and Evaluating the performance of machine learning models is crucial to ensure that spam detection systems function efficiently in real- world email filtering environments. The accuracy, computational efficiency, and adaptability of the various models examined in this section are compared. Additionally, the implications of deploying these models in practical applications, such as enterprise security and personal email services, are discussed.

Spam Detection Models for Business Use

Enterprise email security relies heavily on spam detection because businesses need highly accurate and effective filtering mechanisms to protect themselves from cyber threats. Some key applications include:
Protection Against Phishing Attacks:
Spam emails are the leading cause of corporate data breaches, often used for phishing attacks targeting employees.
Machine learning-based spam filters can detect phishing attempts based on text analysis and sender behaviour.
Scalability for Large Organizations:
Spam detection solutions that are scalable are required by large corporations, which receive millions of emails daily. Cloud-based email services (e.g., Google Workspace,
Microsoft 365) utilize Random Forest and LSTM models for enhanced filtering.

Impact of Spam Detection on Personal Email Filtering.

Beyond enterprise security, spam detection enhances the email experience for individual users. Among the notable advantages are:
User Experience and Organization of Emails:
Spam filters keep inboxes tidy and well- organized, easing user frustration. ML-based filters improve filtering accuracy by learning over time and adjusting to user preferences.
Customizable Spam Filters for Personal Emails:
Users can adjust personalized spam detection settings based on filtering sensitivity.
Users can train spam filters using adaptive ML models by marking messages as spam or ham.
Security against Identity and Financial Theft Scams:
Fake lottery winnings, job scams, and investment frauds frequently target personal email accounts. ML-powered spam detection prevents users from falling victim to email- based financial frauds.

Challenges and Future Improvements in Spam Detection

Obstacles and Potential Solutions for Spam Detection Even though ML-based spam detection has progressed, there are still a few challenges to overcome:

Spam's ever-evolving tactics and adversarial strategies:
By altering the content of messages, attackers use adversarial techniques to get around spam filters. Future research must focus on real-time adaptive spam detection models that can counter evolving threats.
False Advantages and False Disadvantages: Spam filters sometimes classify important emails as spam, leading to missed communication.
Developing hybrid AI-based filtering techniques can help balance high spam detection with minimal false positives.
Overhead for Computing in Large-Scale Email Systems:
Since deep learning models like LSTMs require a lot of processing power, they won't work well in environments with few resources.

Results and Discussion

1.10 Comparative Analysis of Models

The evaluation of various ML models showed the following:
The best choice for spam detection is LSTM, which achieved the highest accuracy (96 percent). Although Naive Bayes detected complex spam with lower accuracy, it was the quickest. Random Forest provided a balance between accuracy and efficiency, making it suitable for large-scale spam filtering.
SVM performed well in accuracy but struggled with large datasets due to computational demands.

1.11 The Study's Restrictions Dataset Limitations:

The datasets used may not fully represent modern spam trends, requiring continuous updates.
Computational Complexity: Because they require a lot of processing power, deep learning models like LSTMs are difficult to implement on all devices.
Adversarial Spam Attacks: Some manipulated emails bypassed detection, indicating the need for more robust filtering techniques.
False Positives: Better balancing strategies are required because some legitimate emails were mistakenly classified as spam.

1.12 Future Scope and Improvements

Utilization of Transformer Models (BERT, GPT) to improve the accuracy of spam detection. Self- learning models are used in real-time adaptive spam filtering to combat changing spam strategies.
Hybrid Spam Detection Systems that combine rule-based filtering and machine learning models to improve precision. Optimizing for Low-Power Devices to make it possible for deep learning models to work well on all platforms.

Conclusion

In order to safeguard users from phishing, fraud, and malware, spam emails necessitate robust filtering mechanisms because they pose a significant threat to cybersecurity, productivity, and email management. Traditional rule-based spam filters have failed to keep up with changing spam techniques, so models based on machine learning have been adopted. This study evaluated multiple ML approaches, including Naïve Bayes, SVM, Random Forest, and LSTM networks, to determine the most effective spam detection system. The LSTM models had the highest accuracy (96 percent), indicating that they performed better when dealing with sequential text data and intricate spam patterns. However, adversarial spam attacks, computational costs, and false positives remain obstacles.

REFERENCES

1. J. Devlin, et al. "BERT: Deep Bidirectional Transformer Pre-Training for Language Understanding." https://arxiv.org/abs/1810.04805
2. M. S. Labonne and Moran "Spam-T5: Benchmarking Large Language Models for Few-Shot Email Spam Detection." https://arxiv.org/abs/2304.01238
3. Sultana, T., et al. "Email based Spam Detection." International Journal of Engineering Research (2020). https://www.researchgate.net/publication/342 113653_Email_based_Spam_Detection.