SPAM CLASSIFICATION USING LONG SHORT-TERM MEMORY ALGORITHM RNN

#1 MAREDDY CHANDANA, #2 B. Amarnath Reddy
#1 MCA Scholar
#2 Assistant Professor
Department Of Master Of Computer Applications
QIS College of Engineering and Technology

ABSTRACT

In order to differentiate between spam and valid emails, spam classification is a crucial duty for email filtering systems. For this, traditional machine learning techniques have been employed, but they frequently fail to recognize the intricate patterns and variances found in spam emails. In this paper, we offer a unique spam categorization method based on Recurrent Neural Networks (RNNs). Because RNNs can identify word relationships in an email, they are ideal for sequence modeling jobs like this one. To determine if an email is spam or not, we employ an RNN architecture called Long Short-Term Memory (LSTM), which is renowned for its capacity to remember information across lengthy periods. To maximize the model's performance, we test several feature formats, preprocessing methods, and hyper parameters. Our tests on a publicly accessible dataset show that the suggested RNN-based method performs better for spam classification than conventional machine learning techniques, with increased accuracy and resilience to spam email variants.

I. INTRODUCTION:

Every day, billions of emails are sent and received, making email one of the most widely used forms of communication. However, email has evolved into a venue for spamming in addition to being used for legitimate communication. Spam emails, sometimes referred to as unsolicited bulk emails, are annoying to recipients and may include malware or phishing links. Email filtering systems are used to automatically categorize incoming emails as spam or valid in order to tackle the problem of spam. Conventional email filtering systems frequently use manually created rules or machine learning algorithms to categorize emails according to attributes like metadata, sender information, and email content. Recurrent neural networks (RNNs), a type of deep learning technique, have demonstrated promise in recent years for a variety of sequence modeling tasks, including natural language processing (NLP) activities like text generation, sentiment analysis, and language translation. Because RNNs are able to record word dependencies in a sequence a critical component in determining the context of an email—they are well-suited for jobs like spam classification. In this study, we suggest a unique method for classifying spam that makes use of RNNs, more especially Long Short-Term Memory (LSTM) networks. An RNN type called an LSTM network can learn long-term dependencies in sequential input, which makes it appropriate for applications requiring context across lengthy durations.

II. RELATEDWORKS

Email spam detection has evolved significantly from rule-based filters to advanced AI-powered systems. Researchers have leveraged various machine learning (ML), deep learning (DL), and hybrid approaches to enhance the accuracy, adaptability, and robustness of spam filtering techniques.

1. Traditional Rule-Based and Statistical Approaches

Earlier spam filters like **SpamAssassin** relied on heuristics, blacklists, and manually created rules.

- **Meyer &Whateley (2004)** compared rule-based filters with Naïve Bayes, showing machine learning models to be more scalable and adaptive.

Merits: Easy to implement, interpretable. **Demerits**: High maintenance and vulnerability to evolving spam tactics.

2. Machine Learning Techniques

Machine learning has been a foundational pillar for spam detection.

- **Sahami et al. (1998)** were pioneers in applying **Naïve Bayes classifiers** for spam filtering using keyword probabilities.
- **Carreras & Marquez (2001)** implemented **Support Vector Machines (SVM)** for spam detection, demonstrating higher precision than simpler models.
- **Cormack (2008)** evaluated **Boosting and Bagging** methods, concluding ensemble learning significantly boosts spam classification accuracy.

Merits: Good performance on structured feature sets.
Demerits: Requires feature engineering, may struggle with evolving language.

3. Deep Learning Models

Deep learning automates feature learning and improves accuracy on complex patterns.

- **Xiang et al. (2018)** introduced a **CNN-based model** that processed email headers and content for spam detection.
- **Hidalgo et al. (2020)** employed **RNN and LSTM networks** to capture temporal patterns and context in email sequences.
- **Zhang et al. (2021)** implemented an **attention-based Bi-LSTM** to enhance spam identification from long-text content.

Merits: High accuracy, robust to obfuscation, works on raw text. **Demerits**: Needs large datasets, higher computational resources.

4. Natural Language Processing (NLP) in Spam Filtering

NLP helps models understand context, semantics, and sentiment of emails.

- **Moustafa et al. (2019)** used **TF-IDF and Word2Vec** with classifiers for text vectorization.
- **Devlin et al. (2019)** introduced **BERT**, and subsequent research showed **BERT-based fine-tuned models** outperform traditional methods on benchmark spam datasets.

Merits: Better understanding of language nuances.
Demerits: Computationally intensive, risk of over fitting on small datasets.

5. Hybrid and Ensemble Models

To combine the strengths of different techniques:

- **Zhou et al. (2020)** proposed a hybrid model combining **Naïve Bayes and SVM**, improving detection and reducing false positives.
- **Yang et al. (2022)** developed a **deep ensemble model** using CNN, RNN, and Gradient Boosting for robust performance across varied spam types.

Merits: Balanced trade-offs, improved generalization.

Demerits: More complex training and integration.

6. Adversarial Spam Detection

Newer works address adversarial spam where attackers evade detection using tricks (e.g., Unicode obfuscation).

- **Good fellow et al. (2015)** introduced **adversarial training**, later applied in spam to improve model robustness.
- **Shen et al. (2023)** applied **GANs** (Generative Adversarial Networks) to simulate spam and retrain classifiers dynamically.

Merits: Anticipates unseen spam types. **Demerits**: Hard to train and validate.

7. Datasets Used

Common datasets for benchmarking:

- **Enron Spam Dataset**
- **SpamAssassin Corpus**
- **Ling-Spam Dataset**
- **TREC Public Spam Corpus**

Note: Most datasets are dated; recent works emphasize the need for newer, realistic corpora.

III. SYSTEMANALYSIS

Existing System

In the existing system, spam classification in email filtering systems is typically performed using traditional machine learning techniques and rule-based approaches. These methods rely on manually crafted features such as sender information, email content, and metadata to classify emails as either legitimate or spam. Common machine learning algorithms used for this purpose include decision trees, support vector machines (SVM), and naive Bayes classifiers. While these approaches have been effective to some extent, they often struggle to capture the complex patterns and variations in spam emails. Spam emails can be highly dynamic and may include obfuscation techniques to evade detection, making it challenging for traditional machine learning models to generalize well. Moreover, traditional approaches may require frequent updates and maintenance to adapt to new spamming techniques and patterns. This can be labor-intensive and time-consuming, especially as the volume and sophistication of spam emails continue to increase.

DRAW BACKS:

1. **Limited Feature Representation**: Traditional machine learning approaches often rely on manually crafted features, which may not capture all relevant information in spam emails. This can lead to lower accuracy and generalization performance.
2. **Difficulty in Handling Sequential Data**: Spam emails are often characterized by sequential patterns, such as the order of words or phrases. Traditional machine learning models may struggle to capture

these dependencies, leading to suboptimal performance.

3. **Scalability Issues**: As the volume of emails continues to increase, traditional machine learning approaches may struggle to scale efficiently. This can lead to longer processing times and reduced responsiveness in email filtering systems.

PROPOSED SYSTEM :

In the proposed system for spam classification using Recurrent Neural Networks (RNNs), we aim to address the limitations of traditional machine learning approaches by leveraging the power of deep learning for sequence modeling. RNNs, and specifically Long Short-Term Memory (LSTM) networks, are well-suited for this task as they can capture long-range dependencies in sequential data, which is crucial for understanding the context of an email.The proposed system consists of several key components. Firstly, we preprocess the email data to convert it into a format suitable for input into the neural network. This preprocessing may include tokenization, removing stop words, and converting words into numerical representations using techniques like word embedding's.Next, we train an LSTM neural network on the preprocessed email data to learn the complex patterns and relationships in spam emails. The network is trained using a large dataset of labeled emails, with the objective of minimizing a loss function that measures the difference between the predicted and actual labels.

ADVANTAGES

1. **Better Sequence Modeling**: RNNs, and specifically LSTM networks, are well-suited for modeling sequential data like email text. They can capture long-range dependencies in the data, which is crucial for understanding the context of an email and distinguishing between legitimate and spam emails.
2. **Automatic Feature Learning**: RNNs can automatically learn relevant features from the input data, reducing the need for manual feature engineering. This can lead to better performance and generalization to new and unseen spamming techniques.

IV. IMPLENTATION

Modules:

1. Email Input Module

- **Function**: Captures incoming emails from an email server or inbox (IMAP/POP3).
- **Components**:
  - Email parser (to extract headers, subject, body, attachments)
  - MIME decoder
- **Tools**: Python libraries like email, imaplib, or mail parser

2. Preprocessing Module

- **Function**: Cleans and prepares email data for analysis.
- **Tasks**:
  - Remove HTML tags, stop words, punctuation
  - Lowercasing and lemmatization
  - Extract text from attachments (optional)
- **Techniques**: Regular expressions, NLP preprocessing (NLTK, spaCy)

3. Feature Extraction Module

- **Function**: Converts text into a format suitable for machine learning models.
- **Methods**:
  - **TF-IDF / Bag-of-Words**
  - **Word2Vec / GloVeembedding's**
  - **BERT embeddings** for contextual representation

- **Additional Features**:
  - Frequency of suspicious keywords
  - URL and domain analysis
  - Email sender/receiver metadata

4. Classification Module

- **Function**: Detects spam using machine learning or deep learning models.
- **Approaches**:
  - **Traditional ML**: Naïve Bayes, SVM, Random Forest
  - **Deep Learning**: CNN, RNN, LSTM, BERT
  - **Hybrid Models**: Ensemble methods combining ML and DL
- **Output**: Labels email as **Spam** or **Ham (Not Spam)**

5. Training & Model Update Module

- **Function**: Trains and updates the AI model using labeled datasets.
- **Processes**:
  - Train-test split / cross-validation
  - Hyperparameter tuning (Grid Search, Optuna)
  - Online learning for dynamic updates
- **Datasets Used**: Enron, SpamAssassin, TREC, Ling-Spam

6. Spam Filtering & Action Module

- **Function**: Acts based on classification results.
- **Actions**:
  - Move spam emails to spam/junk folder
  - Tag subject line with [SPAM]
  - Quarantine or delete email
- **Integration**: Works with email clients or servers (e.g., Gmail, Outlook)

7. Feedback & Learning Module (Optional)

- **Function**: Takes user feedback to improve model accuracy.
- **Mechanism**:
  - User marks email as spam/not spam
  - Updates training dataset with new labels
  - Retrains or fine-tunes the model periodically

8. Logging & Reporting Module

- **Function**: Maintains logs and generates reports for administrators.
- **Includes**:
  - Number of emails processed
  - Spam detection accuracy
  - False positive/negative counts
  - Audit trail for compliance

9. Visualization Dashboard (Optional)

- **Function**: Provides real-time monitoring of system performance.
- **Tools**: Streamlit, Dash, Grafana
- **Displays**:
  - Spam statistics
  - Model confidence scores
  - User feedback trends

10. Security & Privacy Module

- **Function**: Ensures secure handling of email content and user data.
- **Practices**:
  - Data anonymization
  - Secure model APIs
  - GDPR-compliance mechanisms

Methodology:

The methodology follows a systematic pipeline consisting of data collection, preprocessing, feature extraction, model training, classification, and deployment.

1. Data Collection

- **Source**: Public datasets like:
  - **Enron Email Dataset**
  - **SpamAssassin Corpus**
  - **Ling-Spam Dataset**
- **Data Types**: Raw email content including:
  - Subject line
  - Body text
  - Header fields (sender, receiver, timestamp)
  - Links and attachments (optional)

2. Preprocessing

To ensure that raw email data is suitable for model training and classification:

- **Text Cleaning**:
  - Remove HTML tags, special characters, and punctuation.
  - Normalize text (e.g., lowercasing, removing stop words).
- **Tokenization and Lemmatization**:
  - Convert email text into tokens and reduce them to base forms using tools like NLTK or spaCy.
- **Handling Non-Text Features** (optional):
  - Extract metadata such as the number of links, sender domain, presence of attachments.

3. Feature Extraction

Transform cleaned text into numerical features suitable for AI models:

- **Traditional NLP Methods**:
  - Bag-of-Words (BoW)
  - Term Frequency–Inverse Document Frequency (TF-IDF)
- **Word Embedding Techniques**:
  - Word2Vec, GloVe
  - BERT or other transformer-based embeddings for contextual understanding
- **Custom Features** (optional):
  - Number of capital letters
  - Count of exclamation marks
  - Suspicious word frequency

4. Model Development and Training

Different algorithms are evaluated to identify the best-performing model:

- **Machine Learning Algorithms**:
  - Naïve Bayes, Logistic Regression, SVM, Random Forest
- **Deep Learning Models**:
  - CNNs for spatial pattern recognition in text
  - LSTM/GRU for sequential modeling of email text
  - BERT for deep contextual classification
- **Training Process**:
  - Split dataset into training (70%), validation (15%), and test (15%)
  - Use cross-validation and hyperparameter tuning (e.g., Grid Search)

5. Classification

- **Model Output**:
  - Classifies each email as:
    - **Spam**
    - **Ham** (non-spam)
- **Metrics for Evaluation**:
  - Accuracy

  - Precision, Recall, F1-score
  - ROC-AUC (to evaluate trade-off between false positives and false negatives)

6. Model Optimization and Updating

- **Imbalance Handling**:
  - Use SMOTE (Synthetic Minority Over-sampling Technique) or class weighting
- **Regular Updates**:
  - Retrain the model periodically with new email samples to adapt to evolving spam patterns

7. Deployment and Real-Time Detection

- **Integration with Email Clients**:
  - Through APIs or plug-ins
- **Deployment Options**:
  - On-cloud (for enterprise-level filtering)
  - On-device (for lightweight real-time classification)
- **Output Actions**:
  - Move to spam folder
  - Mark with spam tag
  - Notify user or system admin

8. Feedback Loop (Optional)

- Allow users to manually correct false positives/negatives.
- Use feedback to update the labeled dataset and retrain the model for improved accuracy over time.

V. FUTURE SCOPE AND CONCLUSION

To sum up, applying Recurrent Neural Networks (RNNs) to spam classification presents a viable strategy for raising the precision and potency of email filtering systems. Since long-range dependencies in sequential data are essential for comprehending the context of an email, RNNs—more especially, Long Short-Term Memory (LSTM) networks—are well-suited for this purpose. RNNs can automatically extract pertinent features from email data by utilizing deep learning, which eliminates the need for human feature engineering and may even boost performance. RNNs can also adjust to new and changing spamming strategies, which over time makes them more resilient and efficient. Although employing RNNs for spam classification has drawbacks, such as computational complexity and the requirement for a significant volume of labeled data, these drawbacks are outweighed by the advantages. RNNs can outperform more conventional machine learning techniques in terms of accuracy and scalability with the right training and tuning. All things considered, email filtering technology has advanced significantly with the usage of RNNs for spam classification. Email filtering systems can become more precise, flexible, and efficient in thwarting the always changing danger of spam by implementing deep learning techniques.

VI. REFERENCES

1. Hochreiter, S., &Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 9(8), 1735-1780. doi:10.1162/neco.1997.9.8.1735

2. Graves, A., Mohamed, A., & Hinton, G. (2013). Speech recognition with deep recurrent neural networks.In IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 6645-6649).IEEE.

3. Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhutdinov, R., Zemel, R., &Bengio, Y. (2015). Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In International Conference on Machine Learning (pp. 2048-2057).PMLR.

4. Singh, A., &Juneja, M. (2018). A Review on Spam Detection Techniques Using Machine Learning and Datasets.In International Conference on Advanced Computing and Communication Systems (ICACCS) (pp. 1-6).IEEE.

5. Liu, Y., Wang, D., & Zhang, D. (2019). A Review on Email Spam Filtering Techniques. In IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC) (pp. 650-655). IEEE.

6. Papadopoulos, S., Kotsiantis, S., &Pintelas, P. (2020). A Survey on Machine Learning for Spam Detection.In International Journal of Knowledge-Based Organizations (IJKBO), 10(2), 17-36. doi:10.4018/IJKBO.2020040102