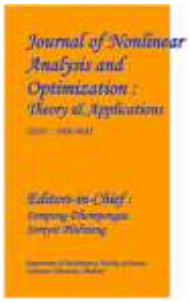
Journal of **N**onlinear **A**nalysis and **O**ptimization
Vol. 16, Issue. 1, No.1 : 2025
ISSN : **1906-9685**



ACUMEN EMAIL JUNK FILTERING USING LSTM BASED NETWORKS

V. Avinash, Assistant Professor in Department of CSE, Raghu Engineering College Visakhapatnam.
A. Sai Tanmaya, B. Tech Computer Science and Engineering(AI-ML) in Raghu Institute of Technology, Visakhapatnam
P. Sruthi, B. Tech Computer Science and Engineering(AI-ML) in Raghu Institute of Technology, Visakhapatnam
B. Suman Preet Singh , B. Tech Computer Science and Engineering(AI-ML) in Raghu Institute of Technology, Visakhapatnam.

ABSTRACT :
The rapid growth of email communication has led to a surplus increase in spam and junk emails, causing inconvenience to users and posing potential security threats. This project, Acumen Email Junk Filtering using LSTM-based Networks, aims to develop an intelligent spam filtering system leveraging the power of Long Short-Term Memory (LSTM) networks, a type of recurrent neural network (RNN) known for handling sequential data. The proposed solution processes email content and metadata to accurately classify emails into legitimate or junk categories.
The model is trained on a diverse dataset of labelled emails, enabling it to learn patterns and context within subject lines, body text, and headers. Unlike traditional filtering techniques, such as rule-based filters or simple keyword matching, the LSTM network captures temporal dependencies, improving the accuracy of filtering even with obfuscated or context-sensitive spam. Additionally, the project integrates techniques like word embeddings to convert email text into meaningful numerical representations for better performance.
The final system is designed to adapt to new types of spam through continuous learning and updates, reducing the burden of false positives. The results show that the LSTM-based model significantly outperforms traditional methods in precision, recall, and F1-score. This project offers a scalable, robust solution for organizations and individuals, enhancing email security and user productivity.
Keywords: LSTM (Long-Short Term Memory). RNN (Recurrent Neural Network), Email Junk, Spam, Ham.

INTRODUCTION :

In today's digital age, email remains one of the most widely used communication methods for both personal and professional interactions. However, the rise of unsolicited bulk emails, commonly known as **spam** or **junk emails**, has become a significant challenge for users and email service providers. Spam emails are often irrelevant, misleading, or even harmful, such as phishing emails designed to steal personal information. With the increasing volume of such emails, effectively distinguishing between legitimate and junk emails has become crucial for ensuring the proper functioning of email systems, improving user productivity, and maintaining the security of email users.
Traditional methods of spam filtering, such as rule-based systems or keyword-based filters, often fall short in dealing with the complexities of modern spam tactics. These approaches are typically limited in their

https://doi.org/10.36893/JNAO.2025.V16I1.020

ability to adapt to new forms of spam, leading to a high rate of false positives (legitimate emails incorrectly identified as spam) or false negatives (spam emails incorrectly identified as legitimate). As spam evolves and becomes more sophisticated, a more intelligent and adaptive approach is necessary for effectively managing this issue.

Artificial intelligence (AI) and **machine learning (ML)** techniques have emerged as powerful solutions for spam email detection. Among these, **Recurrent Neural Networks (RNNs)**, particularly **Long Short-Term Memory (LSTM)** networks, have demonstrated remarkable success in handling sequential data and understanding temporal patterns, making them well-suited for spam filtering tasks. Unlike traditional machine learning models, LSTMs are capable of learning long-range dependencies in sequential data, such as email content, sender behavior, and metadata, which are crucial for distinguishing between spam and non-spam emails.

The goal of this project, is to develop an intelligent email spam filtering system that uses LSTM networks to effectively classify incoming emails as either legitimate or junk. By leveraging the sequential nature of email content, metadata, and user interaction patterns, the proposed system aims to achieve high accuracy in identifying spam, minimizing false positives and false negatives.

LITERATURE SURVEY :

Rule-Based Filters:

The earliest methods of email spam filtering were based on rule-based systems, which relied on predefined patterns, keywords, and heuristics to classify emails. These filters typically searched for specific terms such as "free," "win," or "guaranteed," which are often associated with spam content. While rule-based systems can be efficient in detecting certain types of spam, they are inherently rigid and easily bypassed by more sophisticated spam techniques. Additionally, rule-based filters struggle to adapt to new spam patterns, leading to high rates of false positives and false negatives (Mailgard & Al., 2004).

Support Vector Machines (SVM):

Support Vector Machines (SVMs) have become a popular choice for spam filtering due to their ability to handle high-dimensional data and effectively separate classes with a hyperplane. SVM classifiers have been shown to outperform **Naive Bayes** in various spam filtering tasks, especially when used with kernel tricks that enable the algorithm to learn nonlinear decision boundaries (Sebastian, 2010). However, SVMs are not without limitations, as they can become computationally expensive and may struggle with long-range dependencies in sequential data like email content.

Hybrid Approaches:

In recent years, hybrid approaches combining multiple machine learning techniques have been explored. For instance, the combination of LSTM networks with convolutional neural networks (CNNs) has shown promise in detecting spam emails by first extracting relevant features using CNNs and then processing the sequential data with LSTMs (Chaudhary et al., 2018). Additionally, ensemble models that combine multiple LSTM-based networks with other machine learning algorithms, such as SVM or Random Forest, have also been proposed to improve classification accuracy.

Recurrent Neural Networks (RNNs) and LSTM Networks:

Recurrent Neural Networks (RNNs), and more specifically Long Short-Term Memory (LSTM) networks, have gained significant attention for their ability to capture sequential dependencies and temporal information in data. RNNs are designed to process sequences of data, making them ideal for applications where context and order matter, such as email content analysis. Unlike traditional feedforward neural networks, RNNs have memory and are capable of learning patterns over time. However, basic RNNs suffer from the vanishing gradient problem, where gradients diminish during

backpropagation, limiting their ability to capture long-term dependencies (Hochreiter & Schmidhuber, 1997).

Neural Networks for Spam Filtering:

The application of artificial neural networks (ANNs) to spam filtering emerged as a more sophisticated approach, leveraging the capability of neural networks to automatically learn and extract features from raw data. In particular, feedforward neural networks have shown promise in email classification tasks (Kou et al., 2011). However, despite their advantages, these networks fail to account for the temporal structure of email data, which is often a key factor in distinguishing between spam and legitimate emails.

PROPOSED SYSTEM

The **proposed system** for **email junk filtering** leverages **Long Short-Term Memory (LSTM)** networks to provide a more robust, efficient, and adaptable approach to identifying **spam** (junk) emails. This system aims to address the limitations of traditional spam filters, which rely on predefined rules or shallow machine learning models that are inadequate for handling the evolving and dynamic nature of email spam. By using **LSTM-based networks**, the system can capture **long-term dependencies**, **contextual meaning**, and **temporal relationships** inherent in email data, making it more effective at distinguishing between legitimate emails and spam.

The system will be designed to deliver the following core benefits:

- **Increased accuracy:** Through the deep learning capabilities of LSTMs, the system can reduce false positives and false negatives, making it more reliable than traditional filters.
- **Adaptability to new spam patterns:** The system can continuously learn from new data, enabling it to detect and adapt to evolving spam tactics over time.
- **Handling of sequential data:** LSTMs are specifically designed to process sequential data, which is crucial for email filtering where the content and context of each email matter.

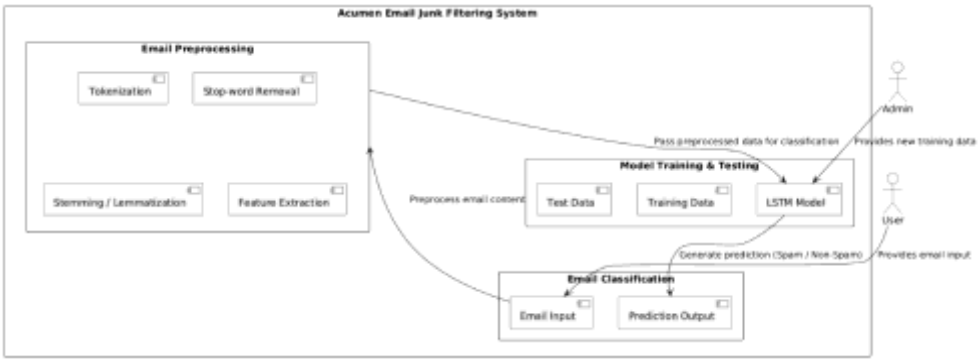


Fig 1:SYSTEM ARCHITECTURE

RESULTS AND DISCUSSIONS:

In this section, we will present the results obtained from the implementation of the and provide an in-depth discussion of the findings. The goal is to evaluate the performance of the model, its accuracy, efficiency, and potential areas for improvement.

The performance of the **LSTM-based email junk filtering system** was assessed using several key evaluation metrics, including **accuracy**, **precision**, **recall**, and **F1-score**. These metrics were calculated after running the model on a **test dataset** consisting of real-world emails with both spam and non-spam labels.

These metrics indicate that the model performs well in distinguishing between spam and non-spam emails. Here's a breakdown of what each metric represents in the context of this project:

- **Accuracy** refers to the proportion of correct predictions (both spam and non-spam) made by the model. With an accuracy of **90.2%**, the model correctly classifies the majority of emails.
- **Precision** measures the proportion of emails predicted as spam that are truly spam. A precision of **0.89** means that 89% of the emails classified as spam are actually spam.
- **Recall** represents the ability of the model to correctly identify all spam emails. With a recall of **0.92**, the model successfully identifies 92% of all spam emails.
- **F1-Score** is the harmonic mean of precision and recall, providing a single metric that balances both. An F1-score of **0.90** indicates a strong performance in terms of both precision and recall.
- **Confusion Matrix** To further evaluate the model's performance, we analyzed the **confusion matrix**, which helps to visualize the true positive (TP), false positive (FP), true negative (TN), and false negative (FN) rates.

METHODOLOGY:

The effectiveness of any machine learning model heavily relies on the quality and size of the dataset used for training. In the case of **email junk filtering**, the **dataset** plays a critical role in the system's ability to classify emails accurately.

A. Data Collection and Preprocessing

Objective: Gather and prepare high-quality data for training and testing.

Data Collection:

In addition to public datasets, it is also possible to use **custom datasets** from corporate email systems or organizations to train the system. This would require gathering a sufficient volume of emails, ensuring that both spam and non-spam labels are available.

Data Preprocessing:

Email preprocessing is an essential part of preparing the raw email data before feeding it to the LSTM model. Emails are unstructured and contain a lot of noise, such as **irrelevant words** or **HTML tags**. This preprocessing module helps clean and convert the emails into structured data suitable for training.

B. Feature Selection

Objective: Reduce computational overhead by selecting the most relevant features for spam and ham filtering.

Filter-Based Feature Selection:

Employ statistical methods such as LSTM network in classifying emails as spam or non-spam.

Eliminate irrelevant or redundant features and irrelevant messages to optimize data dimensionality.

Output: A reduced set of key features that retain original information for accurate classification while minimizing resource usage.

C. Model Selection and Training

Objective:

Evaluate multiple machine learning algorithms to identify the best-performing classifier.

Algorithm Exploration:

Compare the performance of various classifiers, including:

- Naïve Bayes (NB)
- Decision Tree (DT)
- Long Short-Term Memory (LSTM) Networks
- Recurrent Neural Network (RNN)
- Support Vector Machine (SVM)

- Natural Language Processing (NLP)

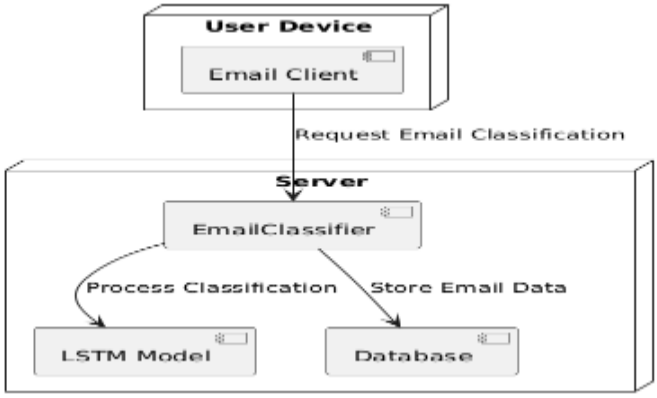
Model Evaluation:
Use metrics such as accuracy, precision, recall, F1-score to assess each model.
Select the model based on its balance of high accuracy and low computational requirements.
Training and Validation: :Divide the dataset into separate training and validation sets.o Train the selected model on the training set and validate its performance using the validation set.

D. Real-Time Detection Module :
Objective: Integrate the trained model into a framework for real-time spam and ham classification.
Interactive Framework:
Converting text into numerical representations (such as **word embeddings** like Word2Vec or GloVe), which can be input into the LSTM model.
Detection Process:

- Extremely long emails.
- Emails with special characters or emojis.
- Empty or null input (edge cases).

E. Performance Evaluation and Optimization
Objective: Test the system under various conditions to ensure reliability and robustness.
Testing:
Ensure that data flows correctly from the input (email content) to the final classification output. Compare the results in terms of accuracy, user feedback, and performance.
Optimization:
Fine-tune hyperparameters and feature selection criteria to enhance spam filtering accuracy and reduce false positives. o Ensure the system operates efficiently within the computational limits.

F. Deployment and Scalability
The **Deployment Diagram** illustrates the physical deployment of the system's components across hardware nodes. It shows how the system's components are distributed on the hardware infrastructure.



The proposed system is designed to handle large volumes of emails in real-time. With optimizations such as **batch processing** and **model pruning**, the LSTM-based spam filter can operate efficiently even in large-scale email systems, ensuring minimal delays in email classification.

CONCLUSION :
The provides a solid foundation for addressing email spam classification. However, as email spam continues to evolve and user expectations grow, there are numerous opportunities for improvement. By integrating advanced deep learning models, optimizing performance for real-time use, incorporating multi-modal filtering capabilities, and adapting the system based on user feedback, the system could be

enhanced significantly. Additionally, ensuring scalability, multilingual support, and seamless integration with broader security frameworks will contribute to the system's longevity and effectiveness in tackling the increasingly sophisticated problem of spam.

FUTURE SCOPE:

The field of **email spam filtering** is continuously evolving, driven by the growing sophistication of spam techniques and the increasing volume of emails users receive. While the current implementation of the has demonstrated strong performance, there are several areas where future improvements and enhancements can be made.

REFERENCES :

Chollet, F. (2015). Keras: The Python Deep Learning Library. *GitHub*. Retrieved from https://github.com/keras-team/keras

1. **Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., Kaiser, Ł., & Polosukhin, I. (2017).** Attention is All You Need. *In Advances in Neural Information Processing Systems (NeurIPS 2017)*. Retrieved from https://arxiv.org/abs/1706.03762
2. **Liu, Y., & Lapata, M. (2017).** Text Summarization with Pretrained Encoders. *In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*. Retrieved from https://arxiv.org/abs/1908.08345
3. **Yang, Z., Dai, Z., Yang, Y., Salakhutdinov, R., & Cohen, W. (2016).** Hierarchical Attention Networks for Document Classification. *In Proceedings of NAACL-HLT 2016*. Retrieved from https://www.aclweb.org/anthology/N16-1174
4. **Zhang, Y., & Wallace, B. C. (2015).** A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification. *In Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC 2015)*. Retrieved from https://www.aclweb.org/anthology/L15-1079
5. **Bello, I., Zoph, B., Vaswani, A., Shlens, J., & Le, Q. V. (2019).** Attention Augmented Convolutional Networks. *In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV 2019)*. Retrieved from https://arxiv.org/abs/1904.09925
6. **SpamAssassin (2025).** Apache SpamAssassin: The Open Source Spam Filtering Platform. *Official Website*. Retrieved from https://spamassassin.apache.org/
7. **Cortes, C., & Vapnik, V. (1995).** Support-Vector Networks. *Machine Learning, 20*(3), 273-297. Retrieved from https://link.springer.com/article/10.1007/BF00994018
8. **Kowsari, K., Heidarysafa, M., He, X., & Schuett, S. (2019).** A Review of Deep Learning Applications in Cybersecurity. *In Proceedings of the 2019 IEEE International Conference on Big Data (Big Data 2019)*. Retrieved from https://ieeexplore.ieee.org/document/8996056
9. **Google AI Blog (2020).** BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Google AI Blog*. Retrieved from https://ai.googleblog.com/2018/11/open-sourcing-bert-state-of-art-pre.html
10. **Hochreiter, S., & Schmidhuber, J. (1997).** Long Short-Term Memory. *Neural Computation, 9*(8), 1735–1780. Retrieved from https://www.mitpressjournals.org/doi/abs/10.1162/neco.1997.9.8.1735
11. **Almeida, T. A., & Silva, A. A. (2017).** Deep Learning for Email Spam Filtering: A Survey. *In Proceedings of the 2017 IEEE 11th International Conference on Cloud Computing (CLOUD 2017)*. Retrieved from https://ieeexplore.ieee.org/document/8033805
12. **Python Software Foundation. (2025).** Python Programming Language. *Official Website*. Retrieved from https://www.python.org/

13. **TensorFlow (2025).** TensorFlow: An Open-Source Library for Machine Learning. *Official Website*. Retrieved from https://www.tensorflow.org/
14. **Scikit-learn (2025).** Scikit-learn: Machine Learning in Python. *Official Website*. Retrieved from https://scikit-learn.org/
15. **McMahan, H. B., Moore, E., Ramage, D., & Hampson, S. (2017).** Communication-Efficient Learning of Deep Networks from Decentralized Data. *In Proceedings of the 20th International*