RV College of Engineering®

Go, change the world

Department of AIML

Real Time SMS Spam using Deep learning

Presented by

Students Name	USN
Ashrith	1RV22AI010
Jaswanth reddy M	1RV22AI020

Faculty Mentors: Somesh Nandi

Agenda

1. Agenda
2. Introduction
3. Literature Survey
4. Summary of LS
5. Requirement analysis – hardware and software specification
6. System architecture (-- ANN-DL architecture) Eg .. architecture of CNN
7. Methodology
8. Module specification –
   a.Module 1 : data collection and pre processing
      i.Input ii. Process iii output
   b.Module 2 : Implementation of ANN / DL algorithm
      i.Input ii. Process iii output
   c.Module 3 : testing and validation
      i.Input ii. Process iii output

Objective

- 1. Enhance the precision and recall of spam detection, minimizing false positives and negatives.
- 2. Provide a real-time classification experience to end-users, ensuring instant feedback on incoming messages.
- 3. Develop a scalable and efficient deployment architecture suitable for large-scale operations.

Introduction

Short Message Service (SMS) is one of the most widely used communication methods, offering fast and efficient delivery of text messages worldwide. However, this medium has also become a target for unsolicited spam messages, which include promotional content, phishing attempts, and fraudulent schemes. Spam messages not only disrupt user experience but also pose significant privacy and security threats. To address these challenges, SMS spam detection has emerged as a critical area of research in the domain of Natural Language Processing (NLP) and deep learning.

Literature Survey

Sl No	Author and Paper title	Details of Publication	Summary of the Paper
1	A. Chandra and S. K. Khatri, "Spam SMS Filtering using Recurrent Neural Network and Long Short Term Memory,"	IEEE Explore, 2021	This study proposes a method utilizing RNNs and Long Short-Term Memory (LSTM) networks to detect spam and ham messages using the 'SpamSMSCollection' dataset from the UCI Machine Learning Repository.
2.	SMS Spam Detection Based on Long Short-Term Memory and Gated Recurrent Unit	International Journal of Future Computer and Communication	This research develops classification models using LSTM and Gated Recurrent Unit (GRU) algorithms, transforming SMS text data into sequential data through Natural Language Processing (NLP) techniques

Literature Survey

Sl No	Author and Paper title	Details of Publication	Summary of the Paper
3	SMS Spam Detection using RNN Dr. K. Sree Ram Murthy, Dr.K.Kranthi Kumar	International Research Journal of Engineering and Technology,2020	This paper utilizes RNNs to distinguish between spam and ham messages, highlighting the effectiveness of RNNs in handling variable-length sequences in SMS data.
4	SMS Spam Detection using NLP and Deep Learning Recurrent Neural Network Variants	IEEE Journal, 2022	This study delves into SMS spam detection using a dataset of 5,570 samples, employing NLP and deep learning techniques with RNN variants.

Literature Survey

Sl No	Author and Paper title	Details of Publication	Summary of the Paper
5	Spam Filtering in SMS using Recurrent Neural Networks	Springer, 2023	This research explores the application of RNNs for filtering spam in SMS, demonstrating the model's capability in handling sequential data.
6	A Comparative Analysis of Recurrent Neural Network and Support Vector Machine for Binary Classification of SMS	Research gate , 2023	This paper compares the performance of RNNs and Support Vector Machines (SVM) in classifying SMS as spam or ham, providing insights into their effectiveness.

Literature Survey

Sl No	Author and Paper title	Details of Publication	Summary of the Paper
7	Improved Spam Detection Through LSTM-Based Approach	IEEE Explore, 2024	This research highlights the potential of NLP and LSTM-based models in revolutionizing spam detection, achieving promising accuracy.
8	Spam SMS Detection Based on Long Short-Term Memory and Recurrent Neural Network	Sensors, 2023	This study proposes a spam-ham classification method using RNNs and LSTM, achieving an overall accuracy of 98%.

Literature Survey

Sl No	Author and Paper title	Details of Publication	Summary of the Paper
9	Machine Intelligence-Based Hybrid Classifier for Spam Detection and Sentiment Analysis of SMS Message	MDPI, 2023	This study proposes a hybrid classifier based on SMS spam classification and sentiment analysis, using optimization algorithms to improve accuracy.
10	Advancements of SMS Spam Detection		This study proposes a hybrid classifier based on SMS spam classification and sentiment analysis, using optimization algorithms to improve accuracy.

Literature Survey

Sl No	Author and Paper title	Details of Publication	Summary of the Paper
11	A Machine Learning Approach for Efficient Spam Detection in Short Messaging System (SMS)	Researchgate, 2024	This study presents a comparative analysis of various machine learning algorithms for SMS spam detection, highlighting the performance and efficiency of different models.
12	SpaLLM-Guard: Pairing SMS Spam Detection Using Open-source and Commercial LLMs	IEEE Explore, 2024	This research evaluates the potential of LLMs, both open-source and commercial, for SMS spam detection, comparing their performance across various learning approaches.

Literature Survey

Sl No	Author and Paper title	Details of Publication	Summary of the Paper
13	SMS Spam Detection System Based on Deep Learning	MDPI Journal, 2024	This paper proposes a detection model utilizing gated recurrent units (GRU) and convolutional neural networks (CNN) as deep learning methods for identifying SMS spam
14	Detection of SMS Spam Messages Using TF-IDF Vectorizer and Deep Learning	IEEE Explore, 2023	This study aims to improve existing SMS spam detection models by employing deep learning methodologies and word embedding techniques.

Literature Survey

Sl No	Author and Paper title	Details of Publication	Summary of the Paper
15	SMS Spam Detection using Machine Learning and Deep Learning Techniques	IEEE Journal, 2024	This research applies various machine learning and deep learning techniques to SMS spam detection, utilizing a dataset from UCI to build and evaluate spam detection models.
16	Survey of SMS Spam Detection Techniques: A Taxonomy	Researchgate , 2023	This comprehensive survey categorizes existing SMS spam detection methods into five primary groups, providing a detailed analysis of each approach.

Literature Survey

Sl No	Author and Paper title	Details of Publication	Summary of the Paper
17	SMS Scam Detection Application Based on Optical Character Recognition and Machine Learning	MDPI Journal, 2023	This paper explores the development of a sophisticated model designed to identify smishing messages by understanding the complex relationships among words using machine learning techniques.
18	Deep Convolutional Forest: A Dynamic Deep Ensemble Approach for Spam Detection in Text	IEEE Explore, 2023	This study presents a dynamic deep ensemble model for spam detection that adapts its complexity and auto-extracts features using convolutional layers and ensemble learning.

Literature Survey

Sl No	Author and Paper title	Details of Publication	Summary of the Paper
19	Using BERT Encoding to Tackle the Mad-lib Attack in SMS Spam Detection	IEEE Journal, 2023	This research investigates the use of BERT language models to overcome adversarial attacks in SMS spam detection, demonstrating the model's resilience against synonym substitution strategies.
20	Spam SMS Filtering using Recurrent Neural Network and Long Short Term Memory	IEEE Explore, 2021	This study employs RNN and LSTM for SMS spam filtering, using Keras and TensorFlow to achieve high accuracy in distinguishing spam from legitimate messages.

Summary of Literature Survey

Emergence of Machine Learning in Fault Diagnosis

- Traditional rule-based and keyword-matching methods struggle with evolving spam techniques.
- Machine learning (ML) effectively detects spam by analyzing patterns, context, and linguistic structures in text messages.

Key Techniques and Approaches

- **Recurrent Neural Networks (RNN**) :Captures sequential text patterns for better spam detection.
- **Natural Language Processing (NLP):** enables text preprocessing and feature extraction techniques like tokenization, stopword removal, and word embeddings, helping machine learning models understand and classify SMS spam effectively.

Summary of Literature Survey

Applications

- **Email and Messaging Security**: Filters spam and phishing messages in emails and chat apps.
- **Cybersecurity and Fraud Prevention**: Assists organizations in preventing fraud, scams, and unwanted advertisements by automatically classifying and blocking spam messages in customer support systems, online forums, and social media platforms.

Challenges Identified

- **High False Positives**: Some legitimate messages may be mistakenly classified as spam.
- **Evolving Spam Techniques:** Spammers constantly adapt, requiring frequent model updates.

Future Trends and Opportunities

- Development of **transfer learning** approaches to address limited data availability.
- Integration of ML models with **Internet of Things (IoT)** for real-time monitoring.
- Use of **explainable AI (XAI)** techniques to improve model interpretability.

Hardware Requirements

1.Processor:

- **Minimum:** Intel i5 or equivalent
- **Recommended:** Intel i7 or higher for faster text processing

2. CPU/GPU:

- **Minimum:** NVIDIA GTX 1050 Ti
- **Recommended:** NVIDIA RTX 3060 or higher for deep learning acceleration

3. RAM:

- Minimum: 8 GB
- Recommended: 16 GB or more for handling large datasets

4. **Communication and Networking:**
   - Ethernet/Wi-Fi modules for real-time data transmission in industrial environments.
   - IoT devices for edge-level data processing (e.g., NVIDIA Jetson, Google Coral).

5. **Power Supply:**
   - Stable power sources with backup solutions (e.g., UPS) to ensure continuous operation.

Software Requirements

1. Operating System:

- Compatible OS for data acquisition and ML software:
  - Windows: Widely used for industrial and user-friendly interfaces.
  - Linux (Ubuntu/CentOS): Preferred for server-side applications and flexibility in deployment.

2. Programming Languages:

- Python 3.7 or higher for model development and implementation.

3. Data Processing & Feature Engineering:

- **Pandas & NumPy:** For handling and manipulating text datasets efficiently.
- **Scikit-learn:** For preprocessing, feature extraction, and evaluation metrics.

4. Deep Learning & NLP Frameworks:

- **TensorFlow/Keras:** For implementing and training the RNN model.
- **PyTorch (optional):** Alternative framework for deep learning-based NLP tasks.
- **NLTK & spaCy: For text preprocessing, tokenization, and linguistic analysis.**

5. Development Environment & Tools:

- **VS Code / PyCharm / Jupyter Notebook: For coding, debugging, and experimentation.**

.

6. Deployment & Frontend:

- Flask / FastAPI (optional): For deploying the trained model as an API.
- Streamlit: For building a simple, interactive UI for real-time spam detection..

System Architecture

The system architecture typically consists of three main layers:

1. Data Acquisition Layer :

- Components: Emails, chat messages, SMS logs, and social media text data.
- Functionality: Collect raw text data from various communication platforms.

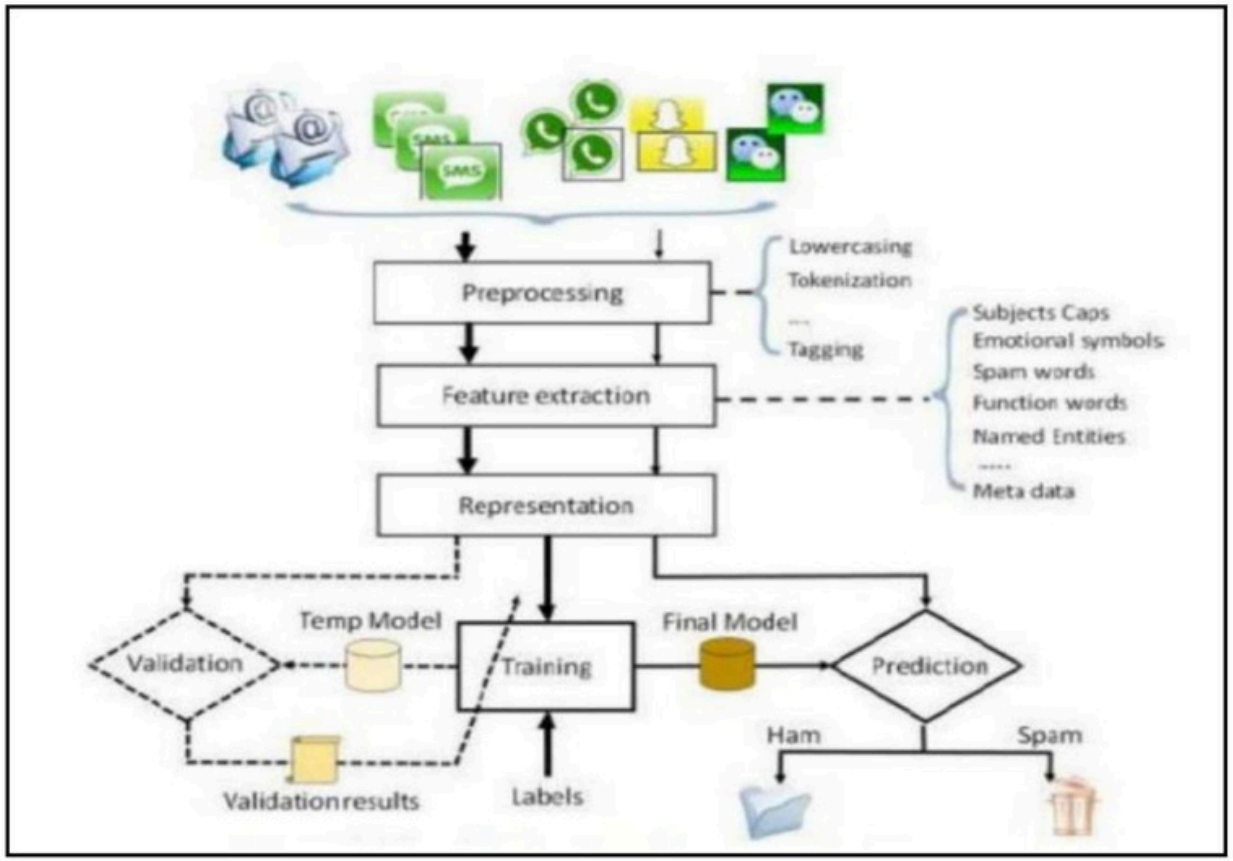
2. Processing Layer :

- Preprocessing: Tokenization, stopword removal, stemming, and vectorization (TF-IDF, Word2Vec).
- Model Processing: Spam detection using RNN/LSTM models for classification.

3. Application Layer :

- Spam Classification: Identify and categorize messages as spam or non-spam.
- Alerts & Actions: Notify users, move spam messages to a separate folder, or block suspicious messages.
- Integration: Deploy as an API or integrate with messaging platforms.

System Architecture



Methodology

1. Problem Definition

Objective:

- Detect and filter spam messages in real time across various communication platforms.
- Improve spam detection accuracy while minimizing false positives and false negatives.

Key Activities:

- Analyze different types of spam (phishing, promotional, malware links) and their impact.
- Define classification criteria based on text patterns and contextual clues.

2. Data Collection and Preparation:

Data Collection:

- Gather text data from emails, SMS, social media, and chat applications. Examples include:
- Spam and non-spam messages from public datasets (e.g., SMS Spam Collection, Enron Email Dataset).
- User-generated text data from forums, reviews, and online discussions.

Methodology

Data Labeling:

- Label the dataset with categories like "spam" and "non-spam."
- Use domain experts or crowdsourced annotations to validate labels if automated labeling isn't available.

Data Preprocessing:

- **Text Cleaning:** Remove special characters, stopwords, and unnecessary whitespace.
- **Normalization:** Convert text to lowercase and standardize formats (e.g., email addresses, URLs)**.**
- **Feature Engineering:** Extract features like word frequency, TF-IDF, and word embeddings (Word2Vec, GloVe).
- **Segmentation:** Tokenize text into words or subwords for model input.

Methodology

3.Model Development

Model Selection:

A. Supervised Models:

➜ RNNs or LSTMs for sequential text data to classify messages as spam or non-spam.

➜ Transformer-based models (e.g., BERT, GPT) for contextual understanding and improved accuracy.

B. Unsupervised Models:

➜ Autoencoders for anomaly detection in spam patterns.

➜ Clustering methods (e.g., K-Means, DBSCAN) combined with NLP techniques for spam categorization.

Methodology

Model Architecture Design:

- Use RNN/LSTM/GRU for sequential text processing.
- Apply word embeddings (Word2Vec, GloVe) for feature extraction.

Model Training:

- Train on labeled spam/non-spam datasets.
- Use data augmentation (synonym replacement, back-translation).
- Apply transfer learning if using pre-trained models (BERT, GPT).

Model Validation:

- Split data (70%-20%-10%).
- Evaluate with accuracy, precision, recall, F1-score, confusion matrix.

4. Deployment and Integration

- **Deployment Options:**
  → Deploy lightweight versions of the model for mobile apps to support offline spam detection.
  → Use cloud platforms for scalable monitoring and model updates.

Methodology

- **Integration:**
  → Seamless integration with email clients (e.g., Gmail, Outlook) and messaging platforms (e.g., Slack, WhatsApp).
  → Set up dashboards for visualization and alerts (e.g., Grafana, Power BI).

5. Real-Time Inference

→ Integrate the trained model with a real-time data pipeline.
→ Use frameworks like TensorFlow Lite or ONNX Runtime for low-latency inference.
→ Configure thresholds for anomaly scores to trigger alerts.

6. Continuous Improvement

- **Feedback Loop:**
  → Continuously collect operational data and system feedback.
  → Use new data to retrain and fine-tune models periodically.
- **Monitoring**:
  → Track system performance metrics and identify cases of false positives/negatives.
  → Update thresholds and configurations based on observed behavior.

7. Evaluation and Reporting

- Post-Deployment Metrics:
  → Test model performance using metrics like accuracy, precision, and F1-score.
  → Regularly validate the model on updated datasets to handle evolving spam patterns.

- Reporting:
  → Generate summary reports with detection rates, false positives, and false negatives.
  → Provide user-friendly reports with spam detection rates and insights via a visual analytics dashboard.

Data Collection and Preprocessing:

1.Input

- Data Sources: Text messages, emails, social media posts, or chat conversations.
- Data Types: Textual data, metadata, multimodal .

2.Process

1. Data Acquisition: Collect real-time or historical text data from messaging platforms.
2. Preprocessing: Clean text (remove noise, links), tokenize, and encode into numerical formats.
3. Feature Extraction: Identify patterns like word frequency and sequence relationships.).
4. Augmentation: Create synthetic training data by modifying or expanding existing text samples.

3.Output

- Spam Classification: Labeled messages as "Spam" or "Not Spam."
- Insights: Metadata for traceability, such as detection confidence scores and timestamps

Implementation of ANN/DL Algorithm:

1. Input

- Data TypesTextual data, metadata, multimodal data.
- Formats: Text files (TXT, CSV), JSON logs, or encoded tensors for RNN models.
- Hyperparameters: Learning rate, batch size,sequence lengths , and activation functions.

2. Process

- Data Preparation: Tokenize and clean text, split into train/validation/test, and normalize inputs.
- Model:
  - Input Layer: Sequence data (word embeddings).
  - Hidden Layers:RNN/LSTM layers to capture sequence patterns.
  - Output Layer: Softmax for classifications.
- Training: Optimize via cross-entropy loss, using Adam or SGD optimizers.
- Validation/Testing: Evaluate accuracy, precision, recall, F1-score.

3. Output

- Trained Model: Exported for deployment in formats like .h5, .pt, or .onnx
- Metrics: Accuracy, confusion matrix, other KPIs.
- Deployment-Ready: Optimized model for real-time spam detection in messaging platforms.

Testing and Validation

1. Input

- Model: Trained RNN/LSTM model (e.g., .h5, .pt).
- Validation Dataset: Text data for hyperparameter tuning and preventing overfitting..
- Test Dataset: Unseen messages to evaluate real-world performance.
- Metrics: Accuracy, precision, recall, F1-score, confusion matrix, and loss.

2. Process

- Validation: : Tune hyperparameters, monitor loss and accuracy, and address overfitting using regularization.
- Testing: Assess model on test data to compute unbiased metrics.
- Error Analysis: Examine false positives/negatives,edge cases , and refine the model.

3. Output

- Validation Results: Trends in metrics and insights into model overfitting or underfitting.
- Testing Results: Confusion matrix, precision, recall, and F1-scores.
- Deployment Decision: Model readiness based on metric thresholds.

THANK YOU