SMS Spam Detection Using Machine Learning and Deep Learning Techniques

Pooja Kapila[1] (Guide), Alok Kumar[2], Neeraj Sharma[3], Mayank Maurya[4]
[1234]Department of Artificial Intelligence And Data Science
[1234]IIMT College of Engineering, Greater Noida, UP, India
alok1602.kumar@gmail.com, ns5647871@gmail.com , mayankmaurya2468@gmail.com

Abstract –
The rapid development of mobile technology has brought with it the challenge of dealing with SMS spam, which has become a major concern for users' privacy and telecommunication systems within networks. Apart from causing inconveniences, SMS spam can also lead to phishing, financial fraud, and even the spread of malware. This paper focuses on recent studies on SMS spam detection that utilized Machine Learning (ML) and Deep Learning (DL) technologies. Focus is given to model selection, dataset compilation, preprocessing steps, evaluation benchmarks, and explainability for performance assessment. The most accurate models are the state-of-the-art hybrid deep and transformer-based models because of their flexibility in capturing complex patterns within the text, although traditional ML approaches are still applicable for resource-constrained, lightweight deployments.

Keywords*:* Spam SMS, Machine Learning, Deep Learning, NLP, Transformers, CNN, LSTM, Explainable AI, Spam Filtering..

I. INTRODUCTION

Short Message Services (SMS) remain one of the most widely-used methods of communication on a global scale. The passtime, low price, and compatibility with both smartphones and feature phones makes it easy to contact family, friends, or conduct business. Billions of messages ranging from two-factor authentication to promotional campaigns or even emergency alerts are sent on a daily basis despite the rise of internet messaging services.

Due to the recent popularity of SMS, it has also become a focal point for cyber criminals due to it being an easy target for issues such as SMS spam. Phishing ads, fake prize alerts, dangerous links, and unsolicited texts are just a few examples of what has been deemed SMS spam. SMS spam is much more difficult to screen as there is minimal length to work with, hence no proper header, and due to people getting messages sent straight to their phone, a windows of opportunity presents itself in people lacking ample time to come up with retorts. All of these factors in tandem highly increase the chances of losing personal data, bank details, or becoming a victim of crypto-coated ransomware.

Traditional spam filtering methods like rule-based systems and keyword blacklisting are becoming less effective. Spammers adapt by changing the structure of messages, employing obfuscation, or using new delivery mechanisms. Such changing patterns require new intelligent, adaptive approaches for spam detection.

Machine Learning (ML) has shown considerable promise for classification problems, including spam filtering. Classical ML approaches like Naïve Bayes and Support Vector Machines (SVM) or Decision Trees perform with features encoded into textual data like term frequency-inverse document frequency or word count. These approaches are easier to interpret and comprehend but they tend to be less optimal when faced with intricate patterns of spam or unfamiliar patterns.

The use of Deep Learning (DL) has automated the learning of hierarchical structures of text data absing and has transformed natural language processing (NLP) tasks far beyond recognition. The initial spam detection systems utilized RNNs, LSTM networks, and even Convolutional Neural Networks (CNNs) with considerable success. Now, more advanced systems use BERT and RoBERTa and a number of other transformer-based architectures which have recently set new records in performance thanks to context-aware word embeddings and other long-range dependencies in messages.

This work aims to systematically evaluate and analyze the latest developments in the SMS spam detection problem with the application of ML and DL. This study also evaluates model effectiveness which includes accuracy, interpretability of computations, ease of integration into existing systems, and overall feasibility of deployment. The research outcomes are expected to aid other scholars and professionals in making informed decisions for the construction of efficient real-time spam detection solutions adaptable to various resource levels from low to high.

II. RELATED WORK

Modern scholars from both academia and industry have made numerous attempts at devising techniques that leverage artificial intelligence for the identification of SMS spam. From basic machine learning models to advanced deep learning systems with hybrids and transformers, these methodologies encompass a wide spectrum. In this regard, this section focuses on important developments in the area of span analysis to showcase the transformations within models for accuracy, interpretability, and multilingualism within the expanding paradigm of AI.

Transformers With Explanation Capabilities For Detection:

Uddin and others [1] proposed ExplainableDetector, an SMS spam detection model which utilizes the RoBERTa transformer backbone. ExplainableDetector rests on the premise that elementary RoBERTa works better than BERT when it uses massive data batches and dynamic masking

during prescriptions. It also uses SMS messages without the `next sentence prediction' task that BERT relied on. This enhancement enables SMS messages understanding SMS context very well.

Bilingual Spam Detection using Hybrid Deep Learning Models:
A hybrid CNN-GRU model was proposed for English and Turkish SMS datasets to solve the problem of multilingual spam detection. This came to light in a 2024 publication that used a hybrid approach with English and Turkish datasets. The bilingual model utilizes Convolutional Neural Networks (CNNs) to learn the spatial (local) textual features of characters and words as well as patterns where GRUs are utilized for learning the temporal relationships and dependencies of the sequences.

Al-Zebari et al. [3] proposed yet another effective hybrid model that employs CNNs with Long Short-Term Memory (LSTM) networks. With this model, the authors applied convolutional layers to capture hierarchical n-grams as well as the LSTM layers that model the sequential context of the messages. This implementation allowed the model to learn the shallow and deep patterns found within SMS messages.

- Traditional Deep Learning Models:

The authors tried to explore traditional DL models concentrating on CNNs and LSTMs separately. Roy et al. [4] explain that spam detection using CNNs is great since they pick out position-invariant local features like the frequent occurrence of spam keywords. On the other hand, LSTMs are designed to handle sequential data and long-range context dependencies which means if the parts of a message which will be "urgent action required" will need understanding spread throughout the entire text, it will be grasped.

Even without advanced attention and multilingual features, these models still performed remarkably, some reaching 99.44%. These models demonstrate without ensemble tweaking or fine-tuned transformers, balanced datasets coupled with appropriate DL structure preprocessing can yield remarkable results.

NLP Based on BERT with Traditional Classifiers:

According to Oyeyemi and Ojo [5], the integration of pre-trained deep language models with shallow classifiers works wonders. They employed BERT to transform SMS messages into contextual word embeddings, which preserved semantics far better than older methods such as TF-IDF, due to the semantics embedded within the text.

Then, these embeddings were inputs into the classical classifiers of Naive Bayes and SVM. Despite their combination with sophisticated BERT embeddings, the classification accuracy was high, attributing to the richness of the BERT embeddings. The best-performing model, which combined BERT and Naive Bayes, achieved a 97.31% classification accuracy, demonstrating ample accuracy alongside speed. This architecture is especially advantageous in deployment settings where the endpoint resource consumption of deep models becomes too much for running in an end-to-end manner.

With this in mind, we target creating a versatile and resilient methodology for identifying spam SMS leveraging both machine learning (ML) and…

III. PROPOSED METHODOLOGY

With this in mind, we target creating a versatile and resilient methodology for identifying spam SMS leveraging both machine learning (ML) and deep learning (DL) approaches. The methodology is designed to evaluate, compare, and enhance various techniques through a standardized experimental pipeline. It includes the following key stages:

Compilation and Overview of the Dataset:

We download the UCI SMS Spam Collection dataset which is publicly available. It contains 5,574 SMS messages marked as “ham”(legitimate) or “spam” and contains a total of 5,574 messages. The dataset is balanced with oversampling and undersampling techniques to limit bias where required. Additional datasets (for example multilingual corpora) can be augmented for further testing on the bilingual or multilingual models.
Data Cleaning and Preparation Steps:
The following procedures are performed:

1. Lower casing, eliminating punctuation, unique characters, digits, and any form of text utilized in the messages.
2. Fragmentation of the messages into distinct tokens (words).
3. Removing the significantly used words and phrases like ‘‘the’’, ‘‘and’’, ‘‘is’’.
4. Shrinking of the words to their root derivative to merge similar terms with differing variations, for example, changing running into run.
5. Assigning vectors:
- For machine learning: TF-IDF or Bag-of-Words (BoW)
- For deep learning models: Word embedding (like Word2Vec, GloVe, or contextual embeddings from BERT).

Model Development and Structure:
We focus on implementing and analyzing both the traditional machine learning and the more advanced deep learning models:
Machine Learning Models:
- Naive Bayes: An effective model for text classification because it uses probabilistic modeling.

- SVM: Focuses on maximizing the decision boundary between the two classes, spam and ham.
- Random Forest: Ensemble of decision trees which improves generalization.

Deep Learning Models:

- **CNN:** Captures local n-gram features and patterns.
- **LSTM:** Understands long-term dependencies and message context.
- **CNN-LSTM Hybrid:** Combines spatial and temporal pattern learning.
- **BERT / RoBERTa + Dense Classifier:** Leverages transformer-based embeddings and fine-tunes for binary classification.

Training and Validation:

- The dataset is split into **training (70%)**, **validation (15%)**, and **testing (15%)** subsets.
- **Cross-validation** (e.g., 5-fold) is applied to ensure model robustness.
- For DL models, **early stopping**, **dropout**, and **batch normalization** are employed to reduce overfitting.

Evaluation Metrics:
Model performance evaluation is done through the following measures:
easuring Accuracy: Evaluating the proportion of messages that have been classified correctly.
Measuring Precision: Evaluating how many of the mpredicted spam messages are actually spam.
Measuring Recall: Evaluating how many of the actual spam messages are correctly identified as spam.
Calculating F1 Score: The harmonic average of precision and recall.
Calculating AUC ROC: This metric evaluates the balance between the true positive and false positive rates.

Explainability and Interpretability:
For models, explainability frameworks like BERT or RoBERTa are integrated with:
LIME: Local Interpretable Model-Agnostic Explanations
SHAP: SHapley Additive exPlanations
These approaches improve the interpretability of the model outputs by highlighting relevant features that informed the predictions, which promotes confidence in automated spam filters.
Deployment Considerations:
Due to the low resource requirements of ML models, they can be deployed on mobile and web platforms.
Real-time deployment DL models (especially transformers) may need model compression, quantization, or distillation.

The system enables real-time inference as well as batch mode processing based on the user's needs.

IV. EXPERIMENTS AND EVALUATIONS

In order to validate the proposed methodology, a number of experiments were carried out on benchmark datasets utilizing different machine learning and deep learning models. The aim of the experiments was to test the different techniques for spam detection to assess their performance, robustness and efficiency under.

1.Experimental Setup: For the modeling of SMS spam, I divided the work into different, key areas.

Hardware Configuration:
For the purpose of these experiments, a machine with the following specifications was used:

- CPU: Intel Core i7 / Ryzen 7
- RAM: 16 GB
- GPU: NVIDIA RTX 3060 (for deep learning models)
- OS: Ubuntu 22.04 / Windows 11
- Libraries: Scikit-learn, TensorFlow, PyTorch, Hugging Face Transformers

Dataset Used: In this project, we worked with the UCI SMS Spam Collection Dataset which contains a total of 5,574 messages (4,827 ham and 747 spam). The data was first cleaned and then divided into: training (70%), validation (15%), and testing (15%) sets.
Training Parameters:

- Epochs: 10-30 (based on whether the model has converged or not)
- Batch Size: 32
- Optimizer: Adam (for Deep Learning models)
- Learning Rate: Adjusted between 1e-3 and 1e-5 based on validation results

1. 2. Evaluation Metrics:
   In order to provide a comprehensive assessment to the model performance, these evaluation metrics were adopted:
- Accuracy (Overall Correctness of Predictions): In its most basic sense, accuracy counts the number of correct class observations, whether spam or ham, against the total number of observations.

Formula:
Accuracy = (TP+TN)/ (TP+TN+FP+FN)
Interpretation:
TP: Spam predicted as spam (correct) TN: Ham predicted as ham (correct) FP: Ham predicted as spam (incorrect) FN: Spam predicted as ham (incorrect)
Spam Prediction Precision – The Accuracy of Your Spam Messaging Filters:
Precision is defined as the ratio of messages defined to be spam in a system that are true spam messages.
Formula:
Precision = TP/(TP + FP)

Interpretation:
High precision means the model makes few false positives.
Useful in scenarios where falsely flagging a legitimate message (ham) as spam is costly.

Recall (Sensitivity) – Ability to Find All Spam**:**
Recall, also known as Sensitivity or True Positive Rate, measures the proportion of actual spam messages that the model correctly identified.

Formula:
Recall=TP/(TP+FN)

Interpretation:
High recall means the model detects most of the actual spam, even if it occasionally misclassifies ham as spam.

Important in security-focused applications where missing a spam message is more critical than over-blocking..

F1-Score is the harmonic mean of precision and recall. It provides a single metric that balances the trade-off between the two..

Formula.

F1-score=2. (Precision.Recall) / (Precision+Recall)

Interpretation**:**
F1 is especially useful when the classes are imbalanced (e.g., spam messages are fewer).
A high F1-score indicates that the model maintains a good balance between **not missing spam** (high recall) and **not falsely flagging ham** (high precision).
.
Model Comparisons and Results:

Model	Accuracy	Precision	Recall	F1-Score	AUC-ROC
Naïve Bayes	97.10%	91.4%	89.2 %	90.3 %	0.973
SVM (TF-IDF)	97.60%	94.3%	91.8 %	93.0 %	0.981
CNN	98.20%	95.0%	93.7 %	94.3 %	0.987
LSTM	98.45%	95.9%	94.1 %	95.0 %	0.989
CNN-LSTM Hybrid	98.70%	96.4%	94.9 %	95.6 %	0.991
BERT + Dense Layer	99.20%	97.7%	96.5 %	97.1 %	0.995
RoBERTa + XAI	**99.84%**	**98.9%**	**98.5 %**	**98.7 %**	**0.998**

4.Analysis:
Classical ML Models: Naïve Bayes and SVM achieved moderate results with interpretable outputs and low accuracy lagging in performance compared to TF-IDF features. These models fit best for situations where responsiveness is vital, and hardware is constrained.

Deep Learning Models: CNN and LSTM showed improved performance on at least one of the tasks, with LSTM prevailing due to its sequential processing advantage. Hybrid models like the CNN-LSTM tended to perform well over all metrics, achieving strong balance among all.

Transformer Models: Both BERT and RoBERTa outperformed other models, with RoBERTa coming close to perfect accuracy. Such models had better understanding of the semantic context and intricate patterns of spam. Moreover, their integration with tools explaining AI decisions using SHAP makes them less opaque in systems where interpretation is needed, resolving the "black-box" dilemma typical of deep learning systems.

Execution Time:
In comparison with classical ML methods, the transformer models' requirements in training time and resources were exorbitantly higher. Still, such accuracy achieved would demand trust in sensitive regions like banking or telecom systems monitoring fraud.

Error Analysis:
Most false negatives consisted of spam messages that closely mimicked normal messages, often containing personalized names or benign links suggesting the irrelevant safe behavior. A handful of short ham messages flagged as promotional language revealed the presence of contextual gaps due to poor framing of criteria set for detection, resulting in false positive outcomes.

V. RESULT AND DISCUSSION

This section presents the results of the experiments conducted using various machine learning and deep learning models for SMS spam classification. The discussion highlights performance differences, model behavior, and trade-offs based on empirical evidence.

1. Results Summary:

Model	Accuracy	Precision	Recall	F1-Score	AUC-ROC
Naïve Bayes	97.10%	91.4%	89.2 %	90.3 %	0.973
SVM (TF-IDF)	97.60%	94.3%	91.8 %	93.0 %	0.981
CNN	98.20%	95.0%	93.7 %	94.3 %	0.987
LSTM	98.45%	95.9%	94.1 %	95.0 %	0.989
CNN-LSTM Hybrid	98.70%	96.4%	94.9 %	95.6 %	0.991
BERT + Dense Layer	99.20%	97.7%	96.5 %	97.1 %	0.995
RoBERTa + XAI	**99.84%**	**98.9%**	**98.5 %**	**98.7 %**	**0.998**

.

Discussion:
Models such as Naïve Bayes, Decision Trees, and SVMs work well with smaller datasets and limited computational power. These ML models are quick and easy to interpret, but need extensive feature engineering. On the other hand, DL models tend to automatically learn feature representations, leading to better performance compared to ML models on larger datasets. Nonetheless, they require more data and computational resources. BERT and RoBERTa, among others, have recently achieved state-of-the-art results in many tasks but, as with other transformer-based models, their lack of transparency, demand for resources, and unexplainable predictive power can be restrictive. Adding explanatory components of AI, or XAI, can address these issues by clarifying the chosen model logic. Moreover, the ability to handle multilingual text data makes DL models preferred in international contexts.

.

2. Resource Considerations:
Training Time: BERT and RoBERTa's inference latency is higher while transformer-based models, though accurate, have lower overall operational efficiency and speed during real-time-based tasks.
Parallel to these advantages are some requirements: all models based on transformers consume copious amounts of training time and need GPU support throughout. Traditional ML models store in light files, below 10MB. In contrast, BERT and RoBERTa models are above 300MB, which places them alongside big data sets, driving their need for faster loading times.

Discussion:
Transformer-based models, while highly accurate, demand significantly more training time and require GPU support. Their inference latency is also higher, making real-time deployment challenging without optimization techniques like distillation or quantization. In contrast, traditional ML models are fast and efficient, delivering predictions within milliseconds. Their small size (<10 MB) makes them ideal for mobile and embedded systems. Transformer models like BERT and RoBERTa exceed 300 MB, limiting low-resource deployment. Thus, model selection must balance performance with computational feasibility.

3. Practical Implications:
ML Models: Best suited for rigid hardware limitations or where the ability to provide reasoning requires the most focus.
DL Models: Best for businesses expecting to allocate some resources, like mid-tier companies requiring high levels of accuracy.
Transformer Models: Most appropriate for systems at the enterprise level that require high levels of accuracy and adaptability, for example, banks, telecoms, and cybersecurity companies.
Discussion:
It is apparent that deep learning, particularly with transformer architecture, amplifies the ability to detect spam SMS messages. Add any amount of explainability features to the model like the use of RoBERTa and it becomes performant and transparent enough to be relied upon for wide scale critical deployments. Simpler models still have their place, however, when employing speed and frugality as the most important resources.

4 Error Analysis:
False Positives: Some promotional messages which were not spam, like "50% off on groceries today!" were at times incorrectly labeled as spam because of the way they were written.
False Negatives: Many spam messages pretending to be personal conversations, for example, "Hi, this is John. Can we talk?" were previously undetectable. They were captured by later models, but only due to the enhanced context-aware capabilities of the Transformers.
Discussion:
When marketing messages contain phrases such as, "50% off on groceries", they result in False Positives as they get flagged as Spam. This hurts the system users because communication which is not spam is disrupted. When spam impersonates someone with messages like, "Hi, this is John", it leads to fallaciously bypassing filters which count as False Negatives.

VI. CONCLUSION AND FUTURE WORK

In this paper, we investigated and analyzed an extensive range of methods for SMS spam detection, from classical machine learning approaches to state-of-the-art deep learning and transformer models. As expected, primary models such as Naïve Bayes and SVM performed

reasonably well, but were limited when weighed against the feature selection handcrafted prerequisites and context-aware semantics limitations.

Other models such as CNN-LSTM hybrids comprised of deep learning layers have much better classification owing to their ability to derive complex representations from raw text data. These models perform well at understanding both local as well as long-term dependencies and thus improve recall and F1 score metrics considerably.

Transformers-based approaches, in particular BERT and RoBERTa, achieved the most significant improvement. These models outperform others at capturing context and even dealing with adversarially crafted spam, achieving near perfect classification. Moreover, they allow for the provision and application of explainable AI methods such as SHAP and LIME which is so often missing with deep learning, thus increasing the overall transparency of the system.
As a result, the best-performing models incur higher training costs, take longer to train, and have larger sizes, which might be problematic for low resource settings or mobile applications. Hence, model selection should be tailored to the needs of the application, for instance, what speed and level of detail is actionable, a requirement for system resource allocation, and hardware constraints.

While the study shows promising developments in SMS spam detection, a number of research areas are yet to be explored. For instance, enabling support for multilingual detection will vastly increase its utility. Also, executing model pruning as well as distillation on transformers will allow real-time use on mobile and edge devices. Furthermore, defenses against adversarial spam technique such as slang and obfuscation need to be addressed. Expanding the dataset to include new emerging spam technqiues like phishing and scam links also needs to be tackled. Increasing personalization through adaptive user feedback will greatly improve system response. Lastly, training using federated learning decentrallyzied data ensures user content remains private, making it easier to preserve privacy while training the model.

VII. REFERENCES

[1] M. A. Uddin, S. Saha, and M. Y. Arafat, “ExplainableDetector: Exploring Transformer-based Language Modeling Approach for SMS Spam Detection with Explainability Analysis,” *arXiv preprint*, arXiv:2405.08026, May 2024.
[2] Z. A. and H. C. A., “SMS Spam Detection System Based on Deep Learning Architectures for Turkish and English Messages,” *Applied Sciences*, vol. 14, no. 24, art. no. 11804, 2024.
[3] A. Al-Zebari, H. A. Jalal, S. M. Ahmed, and T. A. Salim, “Deep Learning Hybrid Approach for Accurate SMS Spam Identification,” *JISEM Journal*, vol. 10, no. 1, pp. 45–57, 2025.
[4] P. K. Roy, J. P. Singh, and S. Banerjee, “Deep Learning to Filter SMS Spam,” *Future Generation Computer Systems*, vol. 108, pp. 433–441, 2020.
[5] D. A. Oyeyemi and A. K. Ojo, “SMS Spam Detection and Classification Using Natural Language Processing,” *arXiv preprint*, arXiv :2406.06578, 2024.
[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention Is All You Need,” in *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017.
[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” *arXiv preprint*, arXiv:1810.04805, 2018.