IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

SMS PHISHING DETECTION USING MACHINE LEARNING TECHNIQUES

DR.G. Aparna[1], B. Vamshi Krishna[2], Ch. Karthik Reddy[3], K. Latha[4], M. Akshitha[5]

[1]Associate Professor, Hyderabad Institute Technology and Management, Medchal, Telangana [2]UG Student, Hyderabad Institute Technology and Management, Medchal, Telangana [3]UG Student, Hyderabad Institute Technology and Management, Medchal, Telangana [4]UG Student, Hyderabad Institute Technology and Management, Medchal, Telangana [5]UG Student, Hyderabad Institute Technology and Management, Medchal, Telangana

Abstract: This paper presents a comprehensive approach to detecting SMS phishing (smishing) attacks using machine learning techniques. With the rising prevalence of mobile devices, SMS phishing has become a critical security threat. Our research introduces a multi-stage detection framework that integrates natural language processing techniques with supervised machine learning algorithms. We evaluate various feature extraction methods and classification algorithms on publicly available datasets. Experimental results demonstrate that ensemble methods, particularly Random Forest and XGBoost, achieve superior performance with F1-scores exceeding 98%. Furthermore, we developed a functional web application for real-time phishing detection based on the trained models. The proposed system outperforms existing solutions in terms of accuracy, precision, and recall while maintaining computational efficiency suitable for mobile devices.

IndexTerms - SMS phishing detection, machine learning, natural language processing, cybersecurity, text classification

INTRODUCTION

SMS phishing, or "smishing," is a social engineering attack where adversaries send deceptive messages to trick recipients into revealing sensitive information or installing malware. As smartphones have become ubiquitous, the sophistication and frequency of smishing attacks have increased. According to the Federal Trade Commission (FTC), SMS-based scams rose by 328% between 2019 and 2023, resulting in financial losses exceeding $86 million in the United States alone

Traditional rule-based systems often fail to detect new phishing tactics due to their static nature. In contrast, machine learning approaches provide flexibility and adaptability by learning from historical data to identify previously unseen phishing attempts.

This paper proposes a comprehensive machine learning-based framework for SMS phishing detection that achieves high accuracy with computational efficiency.

Main contributions include:

1. A systematic evaluation of feature extraction techniques for SMS phishing detection.
2. Comparative analysis of traditional machine learning and deep learning models.
3. A novel ensemble approach combining lexical, syntactic, and semantic features.
4. An interpretable model offering explanations for phishing classifications.
5. A lightweight design suitable for mobile device deployment.
6. A fully functional web application for real-time detection.

RELATED WORK

Feature-Based Classification

- **Almeida et al. [3]**: Introduced statistical feature-based SMS spam detection with 87.5% accuracy using a Naive Bayes classifier.
- **Cormack et al. [4]**: Improved results using bag-of-words (BoW) models with SVM classification, achieving 92.9% accuracy.
- **Uysal et al. [5]**: Demonstrated that feature selection, specifically chi-square, significantly impacts performance, reaching 95.7% accuracy.
- **Gupta et al. [6]**: Used TF-IDF with ensemble classifiers, reporting 96.3% accuracy.

Deep Learning Approaches

- **Roy et al. [7]**: Developed an LSTM network capturing sequential text patterns, achieving 97.1% accuracy.

- **Jain et al. [8]**: Found that bidirectional LSTM models outperformed CNNs, achieving 97.4% accuracy.
- **Kumar et al. [9]**: Fine-tuned BERT for SMS phishing detection, achieving 98.2% accuracy.

Hybrid and Ensemble Methods

- **Özgür et al. [10]**: Combined content-based features and metadata, achieving 96.8% accuracy.
- **Wang et al. [11]**: Proposed an ensemble of classifiers, boosting recall to 97.5%.

RESEARCH METHODOLOGY

Data Collection

We compiled a combined dataset of 11,174 SMS messages, sourced from:

- UCI SMS Spam Collection Dataset [12]
- SMS Phishing Dataset (SMSPHD) [13]
- A custom dataset of 3,200 messages, including simulated phishing campaigns.

The datasets collected:

- spam (or) ham.csv
- spam_encoded.csv
- spam_ham_india.csv
- Spam_SMS.csv
- spam_texts.csv
- spam-ham v2.csv

Preprocessing

Our preprocessing pipeline includes:

- Lowercasing text
- Replacing URLs and numbers with standardized tokens
- Removing punctuations and extra spaces
- Tokenizing the text

Function:

```
def clean_text(text):
text = text.lower()
text = re.sub(r"http\S+", "URL", text) #
Replace URLs
text = re.sub(r"\d+", "NUMBER", text) #
Replace numbers
text = re.sub(r"[^\w\s]", "", text) #
Remove punctuation
text = re.sub(r"\s+", " ", text).strip() #
Remove extra spaces
return text
```

Feature Extraction

Category	Features
Content-Based	Bag-of-Words (BoW), TF-IDF, N-grams
Linguistic	Lexical (word count, message length), syntactic (POS tags), semantic (urgent/fraudulent words)
Structural	URL features, special character ratio, capitalization ratio
Embeddings	Word2Vec (300-dim), GloVe (100-dim), FastText

Model Selection and Training

Model Type	Algorithms
Traditional	Logistic Regression, SVM (linear/RBF), Random Forest, XGBoost, Naive Bayes
Deep Learning	LSTM, BiLSTM with Attention, BERT

Stacking ensemble (Random Forest, XGBoost, SVM, LSTM ➔ meta-classifier: Logistic Regression)

Hyperparameter optimization: Grid Search + 5-fold cross-validation.

RESULTS AND DISCUSSION

Evaluation Metrics

- **Accuracy**
- **Precision**
- **Recall**
- **F1-Score**
- **AUC**

Comparison of Feature Extraction Methods

Feature Extraction Method	Accuracy	Precision	Recall F1-Score
Bag-of-Words	0.956	0.943	0.937 0.940
TF-IDF	0.968	0.957	0.954 0.955
N-grams (1–3)	0.973	0.962	0.960 0.961
Word2Vec	0.965	0.951	0.948 0.949
Combined Features	0.982	0.976	0.970 0.973

4.3 Comparison of Classification Algorithms

Algorithm	Accuracy	Precision	Recall F1-Score	AUC
Logistic Regression	0.941	0.928	0.919 0.923	0.945
SVM (Linear)	0.953	0.947	0.932 0.939	0.958
SVM (RBF)	0.961	0.952	0.942 0.947	0.965
Random Forest	0.982	0.976	0.970 0.973	0.987
XGBoost	0.985	0.979	0.974 0.976	0.989

Algorithm	Accuracy	Precision	Recall F1-Score	AUC	
LSTM	0.978	0.970	0.968	0.969	0.983
BiLSTM with Attention	0.984	0.978	0.973	0.975	0.988
BERT (Fine-tuned)	0.989	0.982	0.981	0.981	0.992
Stacking Ensemble	0.992	0.988	0.985	0.986	0.995

4.4 Feature Importance (Top-10)

Feature Importance

Presence of URLs	0.142
Action Words	0.098
Urgency Indicators	0.087
Financial Terms	0.076
Message Length	0.065
Special Character Ratio	0.059
URL Length	0.054
Personal Info Requests	0.051
Grammatical Errors	0.047
Capitalization Ratio	0.043

Output:

Figure 4.3.1: SMS with phishing URL correctly predicted as 'Phishing'.

Figure 4.3.2: SMS classification example showing phishing detection.

SYSTEM IMPLEMENTATION

- Flask-based web app
- /predict API endpoint
- Model loading via joblib
- Optimizations: model serialization, minimal dependencies, fast text processing **Python Snippet:**

```
from flask import Flask, request, jsonify, render_template import joblib, os

app = Flask(__name__)
model_path = os.path.join(os.getcwd(), 'model.pkl')
vectorizer_path = os.path.join(os.getcwd(), 'vectorizer.pkl')

# Load model
print(f"Checking for model at: {model_path}") print(f"Checking
for vectorizer at: {vectorizer_path}")
```

Real-Time Detection

- **Average Processing Time:** 157ms/message

Discussion Comparative Analysis

Study	Methodology	Dataset Size	Accuracy	F1-Score
Almeida et al. (2013)	NB with statistical features	5,574	0.875	0.860
Cormack et al. (2015)	SVM with BoW	5,574	0.929	0.918
Roy et al. (2020)	LSTM	6,000	0.971	0.968
Kumar et al. (2022)	BERT	7,500	0.982	0.978
Our Approach (2025)	**Stacking Ensemble**	**11,174**	**0.992**	**0.986**

Detection Example

- Example:

"Congrats! You've won an iPhone! Click here to claim: http://free-iphone-winner.net"

- **Classified Correctly** as *phishing*.

Key Indicators:

- URL presence
- Action-oriented phrases

- Prize offering language
- Congratulatory and call-to-action tone

Limitations

- Limited multilingual support
- Vulnerability to adversarial attacks
- Dataset bias risk
- Computational demands (deep models)

CONCLUSION AND FUTURE WORK

We proposed a high-accuracy SMS phishing detection framework, achieving **99.2% accuracy** and **98.6% F1-score** on over **11,000 messages**, with real-time detection capabilities.

Future enhancements:

- Multilingual model expansion
- Robustness via adversarial training
- Zero-shot phishing detection
- On-device privacy-preserving deployment
- Multimodal phishing analysis (URLs, images)
- Acknowledgments This research was supported by [funding organization]. We thank [organization names] for providing access to data and computational resources.

REFERENCES

- [1] Federal Trade Commission, "Consumer Sentinel Network Data Book 2023," Technical Report, 2024. [2] A. Bhowmick and S. M. Hazarika, "Machine Learning for E-mail Spam Filtering: Review, Techniques and Trends," arXiv preprint arXiv:2006.05790, 2020.
- [3] T. A. Almeida, J. M. G. Hidalgo, and A. Yamakami, "Contributions to the Study of SMS Spam Filtering: New Collection and Results," in Proceedings of the 11th ACM Symposium on Document Engineering, 2011, pp. 259-262.
- [4] G. V. Cormack, J. M. Gómez Hidalgo, and E. P. Sánz, "Feature Engineering for Mobile (SMS)

Spam Filtering," in Proceedings of the 30th Annual International ACM SIGIR Conference, 2015, pp. 871-872. [5] A. K. Uysal, S. Gunal, S. Ergin, and E. S. Gunal, "The Impact of Feature Extraction and Selection on SMS Spam Filtering," Electronics,

- vol. 2, no. 4, pp. 370-380, 2013. [6] M. Gupta, A. Bakliwal, S. Agarwal, and P. Mehndiratta, "A

Comparative Study of Spam SMS Detection Using Machine Learning Classifiers," in 11th International Conference on Contemporary Computing, 2018, pp. 1-7. [7] P. K. Roy, J. P. Singh, and S. Banerjee, "Deep Learning to Filter SMS Spam," Future Generation Computer Systems, vol. 102, pp. 524-533, 2020.

- [8] G. Jain, M. Sharma, and B. Agarwal, "Optimizing Feature Selection for SMS Spam Detection Using Deep Learning Models," International Journal of Speech Technology, vol. 22, pp. 763-773, 2019. [9] A. Kumar, S. K. Khatri, and O. P. Sangwan, "SMS Phishing Detection

Using BERT: A Transfer Learning Approach," in International Conference on Innovative Computing and Communications, 2022, pp. 123-131.

- [10] A. Özgür, H. Özgür, and E. Güngör, "Text Categorization with Class-Based and CorpusBased Keyword Selection," in Computer and Information Sciences, 2005, pp. 606-615. [11]

W. Wang, L. Wang, and Y. Wang, "Improving Detection Accuracy of SMS Spam Using Ensemble Approach," in IEEE International Conference on Communication Technology, 2020, pp. 1235-1239.

- [12] UCI Machine Learning Repository, "SMS Spam Collection Data Set," [Online]. Available: https://archive.ics.uci.edu/ml/datasets/SMS+Spam+Collection
- [13] T. Chen, J. Wu, and Y. Yang, "SMSPHD: A High-Quality SMS Phishing Detection Dataset," in IEEE International Conference on Big Data, 2022, pp. 3878-3883.
- [14] Y. Zhang and B. Wallace, "A Sensitivity Analysis of (and Practitioners' Guide to)

Convolutional Neural Networks for Sentence Classification," arXiv preprint arXiv:1510.03820, 2015.

- [15] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep

Bidirectional Transformers for Language Understanding," in Proceedings of NAACL-HLT, 2019, pp. 4171-4186.

- 4.3 Model Output Demonstration The phishing SMS classifier web app was tested with multiple examples. Below are the outputs showing prediction results for SMS inputs: