Deep Learning Approaches for Multi-Class Classification of Phishing Text Messages

Miriam L. Munoz * and Muhammad F. Islam *

Posted Date: 25 August 2025

doi: 10.20944/preprints202508.1703.v1

Keywords: smishing attacks; short message service; phishing; deep learning

 doi:10.20944/preprints202508.1703.v1

Article

Deep Learning Approaches for Multi-Class Classification of Phishing Text Messages

Miriam L. Munoz and Muhammad F. Islam *

Department of Engineering Management and Systems Engineering, The George Washington University, Washington, DC

* Correspondence: mfi@gwu.edu

Abstract

Phishing attacks, particularly Smishing (SMS phishing), have become a major cybersecurity threat, with attackers using social engineering tactics to take advantage of human vulnerabilities. Traditional detection models often struggle to keep up with the evolving sophistication of these attacks, especially on devices with constrained computational resources. This research introduces a chain transformer model that integrates GPT-2 for synthetic data generation and BERT for embeddings to detect Smishing within a multiclass dataset, including minority smishing variants. By utilizing small, open-source models optimized for resource-limited environments, this approach improves both accuracy and efficiency in detecting a variety of phishing threats. Experimental results demonstrate a precision rate exceeding 97% in detecting phishing attacks across multiple categories, showcasing the model's potential for deployment on resource-constrained devices.

Keywords: smishing attacks; short message service; phishing; deep learning

1. Introduction

Phishing attacks, particularly Smishing (SMS phishing), exploit human psychology by manipulating victims into compromising security measures, leading to the exposure of sensitive data [1,2]. These attacks are particularly effective because they rely on trust and emotional triggers, often bypassing traditional cybersecurity defenses [3]. Alarmingly, reports indicate that Americans lost over $800 million to Smishing scams in 2022, underscoring the urgent need for effective detection strategies [4,5]. Given the increasing prevalence of Smishing attacks, it is crucial to develop robust detection mechanisms. Current systems often struggle to identify and classify the diverse range of phishing attempts, especially those that are less frequent but equally harmful [6,7]. This research aims to explore how various deep learning architectures can impact the precision and effectiveness of Smishing detection models for multiclass classification, ensuring high accuracy while remaining functional on mobile hardware.

Despite advancements in machine learning and cybersecurity, conventional Smishing detection systems struggle to keep pace with the continuously evolving tactics of cybercriminals. This challenge is particularly significant when identifying Smishing on mobile devices, which often have limited computational resources [8]. Many existing models focus solely on binary classification, neglecting the need to detect and classify the diverse variants of phishing within multiclass datasets [9]. Additionally, minority classes, representing less frequent but equally dangerous Smishing attempts, can go undetected due to the limitations of current models [10]. To address these gaps, this research builds upon the work of Mishra & Soni [8] by introducing a deep learning-based detection system designed to differentiate between legitimate messages, spam, and Smishing within a multiclass dataset. Utilizing the 'SMS Phishing Dataset for Machine Learning and Pattern Recognition,' this study highlights key features, such as URLs and email addresses, which are crucial for Smishing

 doi:10.20944/preprints202508.1703.v1

classification [8]. A new chain transformer model is proposed, integrating GPT-2 for synthetic data generation and BERT for embeddings to enhance model performance, particularly for minority classes [11,12].

Through this research, the aim is to contribute to the field by developing a deep learning-based Smishing detection system that effectively classifies SMS messages into legitimate, spam, and phishing categories. This approach includes introducing a chain transformer model that leverages synthetic data generation and advanced embeddings to improve detection rates, especially for minority classes. Additionally, the performance of various deep learning architectures are evaluated to identify the most effective model for deployment on devices with limited computational resources [13]. While there has been some exploration of deep learning for Smishing detection, there is limited focus on how the choice of architecture impacts efficiency for multiclass classification [6]. This research intends to fill this gap by assessing how different deep learning models can enhance detection capabilities, particularly for underrepresented phishing types. The originality of this research lies in its comprehensive evaluation of ensemble models that combine deep learning transformer embeddings with traditional machine learning techniques, all with the goal of achieving both accuracy and efficiency.

2. Related Works

The Smishing Detector model presented by Mishra & Soni [14] integrates URL behavior with SMS content analysis to enhance detection accuracy, making it highly effective in identifying Smishing attempts. The dual-analysis approach can be resource-intensive, posing challenges for performance on mobile devices. Its effectiveness depends on the accuracy of the URL behavior analysis component. The model uses binary classification and achieves notable performance, with URL features being the most accurate at 94%. A neural network-based variant of the model [15] improved detection further, achieving a 97.93% accuracy by utilizing a backpropagation algorithm, though it remains computationally demanding. Other implementations focus on verifying URL authenticity and SMS content, achieving accuracy rates as high as 96.2%.

DSmishSMS [16] is a system designed to detect Smishing SMS messages by combining content analysis and machine learning. It focuses on extracting five features from SMS texts to classify messages, with phases for checking URL authenticity and analyzing SMS content. Although it is adaptive and can improve with new data, its reliance on machine learning requires regular updates and large datasets, which can lead to overfitting. Unlike neural network-based systems, DSmishSMS uses traditional classifiers to compare results, achieving an accuracy of 97.9% (for both Smishing and Spam combined). Despite its effectiveness, Smishing detection remains a challenge due to the limited information available in SMS messages.

SmiDCA [17] uses machine learning to detect Smishing attacks and can adapt as threats change over time. The model extracts features from Smishing messages and applies dimensionality reduction to select the 20 most relevant ones for classification. It employs correlation algorithms and machine learning techniques, achieving a 96.4% accuracy using a Random Forest classifier. The model does need large datasets for training, which may limit its effectiveness in real-world scenarios. SmiDCA uses binary classification.

The S-Detector model [18] detects Smishing messages by analyzing both SMS content and URL behavior. If a URL is included, it checks for Android package file downloads to identify potential Smishing attempts. If no URL is present, it uses keyword classification using the Naive Bayes algorithm to check for suspicious patterns. The model is lightweight and well-suited for mobile devices, but it may have difficulty handling advanced evasion techniques used by attackers and might not perform well with unseen Smishing patterns. It uses binary classification to differentiate between Smishing and legitimate messages, offering reliable detection while focusing on efficiency.

The Compact On-device Pipeline Smishing detection [19] model is optimized for mobile devices, providing real-time detection of Smishing attacks with minimal impact on performance. It uses a Disentangled Variational Autoencoder to analyze both SMS content and URL features without the

 doi:10.20944/preprints202508.1703.v1

need for large URL databases. This is important for detecting short-lived malicious URLs. Its lightweight design makes it ideal for mobile device environments with limited computational resources. However, its compact structure may limit its effectiveness against more complex or evolving Smishing tactics. The model's architecture creates a balance between performance efficiency and accuracy, addressing key challenges in mobile security.

The Detection of Phishing in Mobile Instant Messaging model [20] analyzes message content using natural language processing, improving its ability to detect phishing attempts in mobile instant messaging. The integration of machine learning increases its adaptability. However, NLP models can be resource-heavy and typically require significant processing power [21]. The model's accuracy relies on the quality of the training data and the effectiveness of the NLP techniques applied [22].

The paper on Investigating Evasive Techniques in SMS Spam Filtering [23] compares different machine learning models, focusing on their strengths and weaknesses in filtering SMS Spam. It provides useful insights into the effectiveness of various approaches. Because it is a comparative study, it does not offer a clear-cut solution but instead gives an overview of the different models. The performance of each model may vary based on the specific implementation and dataset used [24].

ExplainableDetector [25] uses transformer-based language models, which are highly effective at understanding and analyzing text. It also focuses on explainability, offering insights into how decisions are made. Transformer models are resource-intensive and may not be suitable for all mobile devices [26]. The emphasis on explainability can increase complexity to the model, which may affect its performance [7]. ExplainableDetector uses binary classification.

Privacy BERT-LSTM [27] combines BERT and LSTM to detect sensitive information in text, focusing on high accuracy in identifying privacy-related content. The combination of BERT and LSTM can be computationally intensive, requiring substantial processing power [28]. Privacy BERT-LSTM uses binary classification. In a recent study, researchers applied a Bidirectional LSTM within a federated learning framework to detect Smishing attacks, achieving an accuracy of 88.78% [29].

3. Phishing Dataset Specifications

Smishing detection faces notable challenges due to the absence of standardized benchmark datasets. This research utilizes the 'SMS Phishing Dataset for Machine Learning and Pattern Recognition' (SMSPD) by Mishra & Soni [8], available on Mendeley Data. Building upon the foundational work of Almeida et al. [30], the dataset classifies SMS messages into three categories: Ham (81.1%), Smishing (10.7%), and Spam (8.2%), highlighting a significant class imbalance.

Figure 1 illustrates the distribution of messages across these three categories, emphasizing the dominant presence of legitimate messages (Ham) in comparison to the much smaller proportions of Smishing and Spam. This imbalance plays a critical role in training detection models, as underrepresented classes may require resampling techniques to improve detection performance.

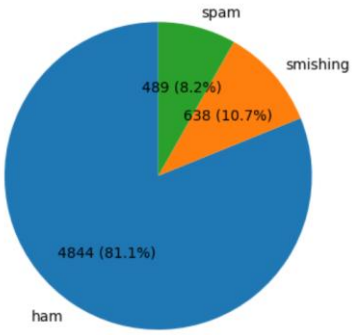

Figure 1. Dataset Distribution by Class.

Figure 2 illustrates the word distribution across text messages, revealing that Ham messages typically contain more words than Smishing and Spam. This difference indicates that word count

 doi:10.20944/preprints202508.1703.v1

may serve as a key feature in classification models, as Smishing and Spam messages tend to be shorter and more direct.

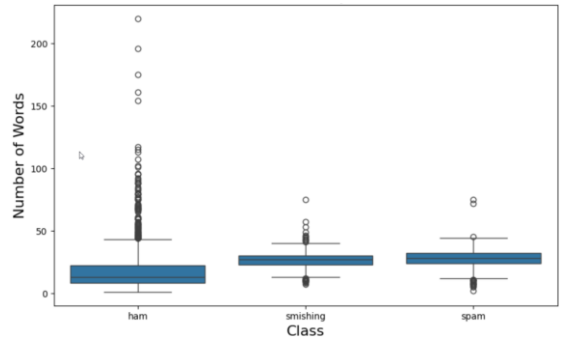

Figure 2. Distribution of Words per Text Data.

Figure 3 presents Kernel Density Estimation (KDE) to examine word frequency distribution, highlighting areas with higher or lower word concentration. This visualization helps identify anomalies within the dataset by uncovering patterns in word usage that may distinguish Smishing attempts from legitimate messages. Understanding these patterns plays a crucial role in refining classification models by leveraging linguistic variations across different message types.

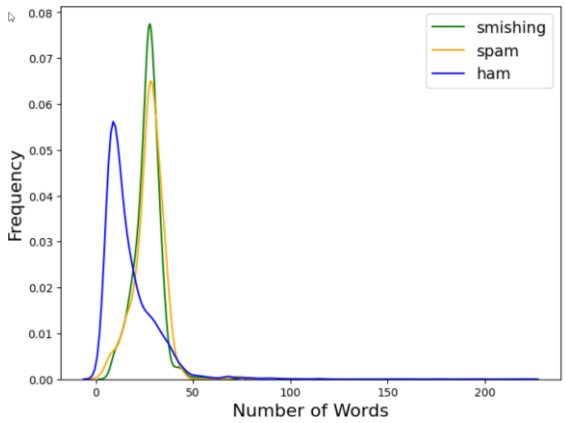

Figure 3. Kernel Density Estimation of Word Concentration.

4. Design and Implementation

This research explores the effectiveness of deep learning models, particularly transformers, alongside traditional machine learning algorithms for detecting Smishing and Spam in text messages. By leveraging Python for generating synthetic data, fine-tuning transformer models, and conducting analysis and visualizations, this study aims to enhance the accuracy and efficiency of Smishing detection systems. The choice to employ scikit-learn version 1.5.2, rather than the newer 1.6.1 release from January 2025, ensures compatibility with existing methodologies while allowing for a focused evaluation of model performance. To enhance model performance, extensive preprocessing is applied to the dataset, including cleaning, tokenization, and handling missing values. Additionally, Exploratory Data Analysis (EDA) is conducted to assess data distribution, class imbalances, and key challenges, providing essential insights for model development and fine-tuning. The originality of this research lies in its comprehensive approach to integrating deep learning and traditional machine learning techniques, as well as its focus on addressing the unique challenges posed by minority classes in Smishing detection. By systematically evaluating the interplay between different model architectures and preprocessing techniques, this study contributes valuable insights to the field of cybersecurity, particularly in the context of mobile phishing threats.

The dataset is divided into training and testing subsets, with data balancing techniques implemented on the training subset to address class imbalances. Feature selection plays a key role in

 doi:10.20944/preprints202508.1703.v1

optimizing efficiency by filtering out irrelevant features. Hyperparameter tuning and validation are performed on the training subset, while the testing subset is used as a benchmark to evaluate model performance and determine the most effective approach for Smishing detection.

To improve Smishing detection, synthetic data generation is employed to balance the dataset. The GPT-2 Medium model, fine-tuned for minority classes, is used to create synthetic text messages that capture distinct linguistic patterns. GPT-2's pre-trained language capabilities ensure that generated messages align with real-world patterns while maintaining the original training data's word count distribution, leading to enhanced model training and detection accuracy. To identify the best synthetic data generation method, SMOTE, GPT-2 Medium, and GPT-2 were evaluated using statistical metrics such as mean, standard deviation, minimum, and maximum values. GPT-2 Medium was selected for its ability to produce high-quality synthetic samples that accurately represent minority-class characteristics.

This study enhances Smishing detection by generating synthetic data to create a more balanced dataset. The synthetic training data is generated using the GPT-2 Medium model and fine-tuned for each minority class. By using the model's pre-training linguistic abilities, synthetic text messages are generated to capture the unique characteristics and semantic details of the minority class. This ensures that the generated messages not only align with the intended meaning but also maintain the word count distribution of the training data.

Figure 4 presents the stats of the dataset using the SMOTE technique.

	SMOTE (num_characters)	SMOTE (num_words)	SMOTE (num_sentences)
count	11607	11607	11607
mean	139.90523	24.919531	1
std	75.330427	12.849239	0
min	2	1	1
25%	72	14	1
50%	145	25	1
75%	199	35	1
max	473	88	1

Figure 4. SMOTE Generated Dataset Statistics.

Figure 5 displays the statistics of the GPT-2 Medium generated dataset. This is the dataset chosen for this research.

	GPT-2 Medium (num_characters)	GPT-2 Medium (num_words)	GPT-2 Medium (num_sentences)
count	11610	11610	11610
mean	115.905599	23.594401	2.346598
std	53.974767	11.006827	1.447662
min	2	1	1
25%	72	16	1
50%	132	25	2
75%	154	30	3
max	790	220	38

Figure 5. GPT-2 Medium Generated Dataset Statistics.

 doi:10.20944/preprints202508.1703.v1

Figure 6 shows the statistics of the GPT-2 generated dataset. There was no significant difference between datasets generated by GPT-2 medium and GPT-2, so the smaller model was chosen.

	GPT-2 (num_characters)	GPT-2 (num_words)	GPT-2 (num_sentences)
count	11610	11610	11610
mean	138.177347	26.591387	3.499139
std	73.747095	16.487527	3.167564
min	0	0	0
25%	80	14	1
50%	142	25	2
75%	183	35	5
max	790	220	38

Figure 6. GPT-2 Generated Dataset Statistics.

Shapley values are applied for feature selection due to their model-agnostic nature, allowing effective estimation of feature importance across different machine learning models. A major advantage of Shapley values is their ability to handle correlated features by assigning lower importance to redundant variables, ensuring that only unique contributions enhance model performance. In this study, Shapley values helped prioritize features while maintaining model accuracy and reducing classification time. The analysis revealed that URLs were the most critical for classification, followed by EMAIL, TEXT, and PHONE. To simplify interpretation, the mean absolute SHAP value was used, providing clear insights into the most influential attributes in the dataset.

Following data preparation and splitting, text embeddings are generated using pre-trained transformer models, such as BERT, DistilBERT (Distilled BERT), and ELECTRA, from Hugging Face's Transformers library. These embeddings are extracted as high-dimensional feature vectors and serve as inputs for traditional machine learning algorithms such as logistic regression, random forest, and support vector machines [31]. To enhance model robustness, k-fold cross-validation is employed, ensuring comprehensive performance evaluation while mitigating overfitting through iterative training on different data subsets [32]. Hyperparameter tuning is conducted using grid search, optimizing parameters such as learning rate, batch size, and regularization strength. The final model is evaluated using F1 score, precision, accuracy, recall, and a confusion matrix to ensure its ability to generalize effectively to new data [33].

Overfitting happens when a model memorizes training data instead of learning patterns that generalize to new data, reducing effectiveness [34]. To prevent this, a small learning rate is adopted during training, allowing gradual updates and preventing drastic weight changes that can lead to overfitting [35].

Different synthetic data generation methods are combined with BERT-based language models and traditional machine learning algorithms to identify the best-performing model. The model development process begins with generating contextual embeddings from a pre-trained transformer model, effectively capturing the semantic meaning of input text. These embeddings serve as input features for traditional machine learning models such as logistic regression, random forest, and support vector machines [36]. By integrating the deep contextual understanding of transformers with the efficiency of traditional machine learning models, this hybrid approach enhances classification accuracy and interpretability [12]. The study incorporates synthetic data generation alongside BERT-based models and machine learning algorithms to optimize multi-class SMS classification.

This research evaluates 47 model variations by testing combinations of three transformer-based embeddings (BERT, Google ELECTRA, and DistilBERT) with three traditional machine learning models: Random Forest Classifier, Logistic Regression, and Support Vector Classifier (SVC). Each

 doi:10.20944/preprints202508.1703.v1

combination undergoes testing with various hyperparameter configurations, including batch size, epochs, and learning rates, to determine the most effective settings for improving overall model accuracy.

To identify the most suitable deep learning (DL) architecture for achieving the research objective, the following tables present a summary comparison of 47 evaluated architectures, highlighting their F1-Score, Precision, and Recall. The results indicate that the DL architecture integrating GPT-2 with BERT-uncased achieves the highest overall performance.

Figure 7 presents the performance of models utilizing BERT embeddings alone, alongside those paired with ML algorithms such as SVC, Logistic Regression, and Random Forest. The results indicate that the BERT embedding without an ML algorithm, trained for 3 epochs, achieves the highest performance.

	ML	Accuracy	ham Precision	ham Recall	ham F1-score	smishing Precision	smishing Recall	smishing F1-score	spam Precision	spam Recall	spam F1-score
BERT	no ML - 3E,	3 0.97	0.99	1	1	0.9	0.91	0.91	0.83	0.78	0.81
BERT	no ML - 1E,	0.97	0.99	0.99	0.99	0.89	0.91	0.9	0.81	0.72	0.76
BERT	SVC	0.95	0.99	0.99	0.99	0.83	0.82	0.82	0.75	0.75	0.75
BERT	LR	0.95	0.99	0.99	0.99	0.8	0.83	0.82	0.74	0.72	0.73
BERT	RM	0.95	0.99	0.99	0.99	0.78	0.84	0.81	0.79	0.69	0.73

Figure 7. BERT Embedding Model Results.

Figure 8 presents the performance of models using DistilBERT embeddings, both without an ML algorithm and in combination with SVC, Random Forest, and Logistic Regression algorithms.

	ML	Accuracy	ham Precision	ham Recall	ham F1-score	smishing Precision	smishing Recall	smishing F1-score	spam Precision	spam Recall	spam F1-score
DistilBERT	no ML	0.96	0.99	0.99	0.99	0.90	0.89	0.90	0.77	0.77	0.77
DistilBERT	SVC	0.95	0.99	0.99	0.99	0.82	0.82	0.82	0.76	0.75	0.76
DistilBERT	LR	0.95	0.99	0.99	0.99	0.79	0.81	0.80	0.73	0.72	0.72
DistilBERT	RF	0.95	0.99	0.99	0.99	0.80	0.82	0.80	0.76	0.70	0.72

Figure 8. DistilBERT Embedding Model Results.

Figure 9 presents the performance of models combining Google ELECTRA embeddings with no ML, Random Forest, Logistic Regression, and SVC ML algorithms.

Google ELECTRA models ran faster but unfortunately the results were not as accurate.

	ML	Accuracy	ham Precision	ham Recall	ham F1-score	smishing Precision	smishing Recall	smishing F1-score	spam Precision	spam Recall	spam F1-score
ELECTRA	no ML	0.96	0.99	0.99	0.99	0.88	0.88	0.88	0.73	0.73	0.73
ELECTRA	SVC	0.94	0.99	0.98	0.98	0.80	0.82	0.81	0.68	0.71	0.69
ELECTRA	LR	0.94	0.99	0.98	0.98	0.86	0.82	0.84	0.62	0.70	0.66
ELECTRA	RF	0.94	0.99	0.98	0.98	0.79	0.89	0.84	0.71	0.65	0.68

Figure 9. Google ELECTRA Embedding Model Results.

Finally, Figure 10 shows the results of models with no Embeddings and using SVC, Logistic Regression, and Random Forest ML algorithms. This shows an improvement in the models' performance which included an Embedding.

	ML	Accuracy	ham Precision	ham Recall	ham F1-score	smishing Precision	smishing Recall	smishing F1-score	spam Precision	spam Recall	spam F1-score
None	SVC	0.96	0.97	1.00	0.99	0.93	0.86	0.89	0.78	0.63	0.70
None	LR	0.95	0.98	0.99	0.99	0.89	0.84	0.86	0.69	0.67	0.68
None	RF	0.95	0.97	1.00	0.98	0.90	0.84	0.87	0.73	0.55	0.63

Figure 10. No Embedding Model Results.

 doi:10.20944/preprints202508.1703.v1

Based on the results presented in Figures 7, 8, 9, and 10, the deep learning architecture combining GPT-2 with BERT-uncased achieves better precision compared to models utilizing DistilBERT or Google ELECTRA. This architecture benefits from GPT-2's ability to generate coherent and contextually relevant text. Also, it utilizes BERT-uncased's proficiency in understanding the nuances of language through bidirectional context. By integrating these two transformer models, the architecture enhances the model's capacity to capture complex linguistic patterns and improve classification accuracy in detecting Smishing and Spam messages. A full list of experiments can be provided upon request.

Establishing a clear threat model that distinguishes between Smishing, Spam, and Ham is essential for developing targeted detection strategies and improving user awareness, particularly in operational scenarios such as financial institutions or e-commerce platforms where users are frequently targeted by Smishing attacks. Smishing messages are specifically designed to deceive users into revealing sensitive information, while Spam consists of unsolicited advertisements that do not pose a direct threat. For instance, in a banking application, identifying Smishing messages can enable real-time alerts that warn users about potential phishing attempts, enhancing user trust and security. A precise definition of each class eliminates ambiguity in the classification logic, grounding the classification scheme in practical use cases, such as customer support systems that filter harmful messages while allowing legitimate communications. This strategic approach to multiclass framing provides actionable insights for improving detection systems, enabling organizations to tailor their responses based on the specific threat posed by each class and effectively address the unique challenges posed by minority classes in Smishing detection.

5. Results

The objective of this research is to develop a deep learning-based Smishing detection system using an ensemble model for multiclass classification. The primary goal is to enhance classification accuracy, with a particular focus on improving detection of minority phishing types by integrating multiple deep learning models in a unified framework.

The findings supported three key hypotheses. First, feature identification analysis demonstrated that URLs and email addresses play a key role in classifying Smishing attacks, as illustrated in the Mean Absolute SHAP values chart below. This conclusion was validated through Shapley value analysis, presented in Figure 11, which highlights their importance in distinguishing minority phishing types from legitimate messages.

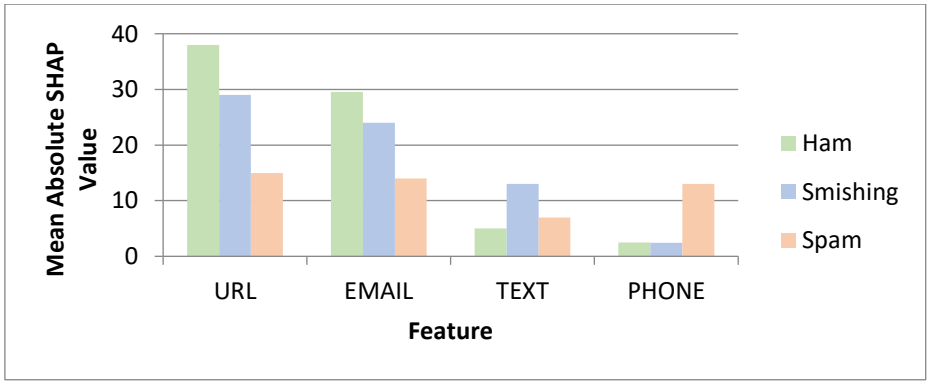

Figure 11. Mean Absolute Shapley Values of Features.

Second, Figure 12 presents the ROC curve for each class, highlighting the deep learning model's effectiveness, which achieved over 97% validation accuracy. The model demonstrated exceptional performance, especially in detecting minority phishing attacks, which are frequently misclassified by traditional methods.

 doi:10.20944/preprints202508.1703.v1

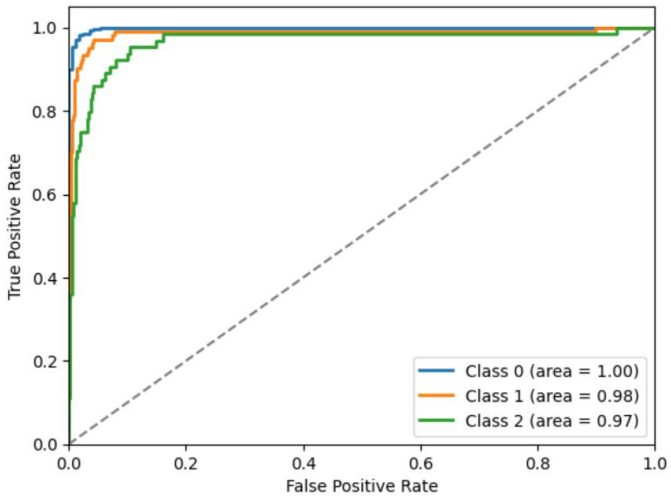

Figure 12. ROC Curve for Ham (Class 0), Phishing (Class 1) and Spam (Class 2).

Lastly, the study examined model synergies by incorporating multiple deep learning models within a chained transformer architecture. This integration notably improved classification accuracy and overall performance, creating a highly effective approach for Smishing detection in a multiclass setting. Figure 13 presents the results of the chained transformer model that achieved the best performance.

	Chained Transformer Model
Accuracy	0.97
ham Precision	0.99
ham Recall	1.00
ham F1-score	1.00
smishing Precision	0.90
smishing Recall	0.91
smishing F1-score	0.91
spam Precision	0.83
spam Recall	0.78
spam F1-score	0.81

Figure 13. Chained Transformer Model Results.

This best model enhances the Baseline model developed by Houston [37] by offering improved effectiveness in identifying minority phishing types. Figure 14 compares the findings from the best model in this research with those of the Baseline model.

	Baseline Model	Chained Transformer Model
Accuracy	0.96	0.97
ham Precision	0.99	0.99
ham Recall	1.00	1.00
ham F1-score	0.99	1.00
smishing Precision	0.90	0.90
smishing Recall	0.84	0.91
smishing F1-score	0.87	0.91
spam Precision	0.77	0.83
spam Recall	0.77	0.78
spam F1-score	0.77	0.81

Figure 14. Baseline vs. Chained Transformer Model.

This study did not find a baseline model from published journals using the same dataset or focusing on text phishing multiclass detection. The closest approximation was the work in

 doi:10.20944/preprints202508.1703.v1

DSmishSMS [16], which proposed a Smishing detection system achieving a binary classification accuracy of 0.979 and referenced as a paper offering an experimental study of the dataset used in this research. Similar accuracy was achieved for multiclass classification. No additional analysis was conducted to compare binary vs. multiclass classification models beyond their accuracy metrics.

6. Conclusion and Future Work

This research provides valuable insights into Smishing detection within multiclass datasets. Through Shapley value analysis, URLs and email addresses emerged as critical features for classification. The deep learning model achieved over 97% validation accuracy, demonstrating strong performance in detecting minority phishing types. The implementation of a chained transformer model effectively balanced complexity and accuracy, establishing a new benchmark for phishing detection. Additionally, integrating multiple deep learning models enhanced the identification of both Smishing and Spam, addressing a gap in existing methods.

Future advancements in Smishing detection could focus on incorporating ensemble techniques to further improve accuracy, particularly for minority classes. Optimizing real-time deployment on mobile devices will be essential for balancing performance and efficiency while enabling continuous updates. Implementing the model directly on mobile devices could help reduce latency, conserve bandwidth, and ensure compatibility with existing security features. Adversarial testing can reinforce model resilience against evolving phishing tactics. Expanding datasets to include a wider range of up-to-date phishing messages will enhance adaptability, creating a more robust detection framework. Introducing techniques such as dropout or early stopping to mitigate overfitting, along with demonstrating the model's ability to generalize to unseen data, would strengthen the overall effectiveness.

Author Contributions: Conceptualization, M.L.M and M.F.I.; methodology, M.L.M and M.F.I.; validation, M.L.M and M.F.I.; formal analysis, M.L.M and M.F.I.; investigation, M.L.M and M.F.I.; resources, M.L.M and M.F.I.; data curation, M.L.M and M.F.I.; writing original draft preparation, M.L.M and M.F.I.; writing, review and editing, M.L.M and M.F.I.; visualization, M.L.M and M.F.I.; supervision, M.L.M and M.F.I.. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data used to support the findings of this study are available upon reasonable request to the corresponding author.

Acknowledgments: The authors acknowledge the use of AI tools for enhancing the quality of the paper, particularly for grammar checking.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

SMS	Short Message Service
GPT-2	Generative Pre-trained Transformer 2
BERT	Bidirectional Encoder Representations from Transformers
URL	Uniform Resource Locator
LSTM	Long Short-Term Memory
SMSPD	SMS Phishing Dataset for Machine Learning and Pattern Recognition
KDE	Kernel Density Estimation
EDA	Exploratory Data Analysis
SHAP	Shapley Additive Explanations
DistilBERT	Distilled BERT
ELECTRA	Efficiently Learning and Encoder that Classifies Token Replacements Accurately

 doi:10.20944/preprints202508.1703.v1

ML	Machine Learning
SMOTE	Synthetic Minority Oversampling Technique
SVC	Support Vector Classifier
ROC	Receiver Operating Characteristic

References

1. Alkhalil, Z., Hewage, C., Nawaf, L., & Khan, I. (2021). Phishing attacks: A recent comprehensive study and a new anatomy. *Frontiers in Computer Science, 3*, 563060. https://doi.org/10.3389/fcomp.2021.563060
2. Gupta, M., Bakliwal, A., Agarwal, S., & Mehndiratta, P. (2018, August 2-4). A comparative study of spam SMS detection using machine learning classifiers. *2018 Eleventh International Conference on Contemporary Computing (IC3)*, 1–7. Noida, India. https://doi.org/10.1109/IC3.2018.8530469
3. Pant, V. K., Pant, J., Singh, R. K., & Srivastava, S. (2024). *Social Engineering in the Digital Age: A Critical Examination of Attack Techniques, Consequences, and Preventative Measures*. In *Effective Strategies for Combatting Social Engineering in Cybersecurity* (pp. 61–76). IGI Global. https://doi.org/10.4018/979-8-3693-6665-3.ch003
4. Chan-Tin, E., & J. Stalans, L. (2023). Phishing for profit. In D. Hummer & J. Byrne (Eds.), *Handbook on Crime and Technology* (pp. 54–71). Gloucestershire, UK:Edward Elgar Publishing. https://doi.org/10.4337/9781800886643.00011
5. FTC. (2023, June 8). New FTC data analysis shows bank impersonation is most-reported text message scam. *Federal Trade Commission. h*ttps://www.ftc.gov/newsevents/news/press-releases/2023/06/new-ftc-data-analysis-shows-bankimpersonation-most-reported-text-message-scam
6. Orunsolu, A. A., Sodiya, A. S., & Akinwale, A. T. (2022). A predictive model for phishing detection. *Journal of King Saud University - Computer and Information Sciences, 34*(2), 232–247. https://doi.org/10.1016/j.jksuci.2019.12.005
7. Sengupta, P., Zhang, Y., Maharjan, S., & Eliassen, F. (2023). Balancing explainability- Accuracy of complex models. *arXiv:2305.14098[cs.LG]*. http://arxiv.org/abs/2305.14098
8. Mishra, S., & Soni, D. (2023b). SMS phishing dataset for machine learning and pattern recognition. In A. Abraham, T. Hanne, N. Gandhi, P. Manghirmalani Mishra, A. Bajaj, & P. Siarry (Eds.), Proceedings of the 14th International Conference on Soft Computing and Pattern Recognition (SoCPaR 2022) (pp. 597–604). Cham, Switzerland: Springer. https://doi.org/10.1007/978-3-031-27524-1_57
9. Brownlee, J. (2021a, January 5). Random oversampling and undersampling for imbalanced classification. *Machine Learning Mastery*. https://machinelearningmastery.com/random-oversampling-and-undersampling-forimbalanced-classification/ (Accessed: 17 October 2024)
10. Qiu, S., Hu, W., Wu, J., Liu, W., Du, B., & Jia, X. (2020), *Temporal network embedding with high-order nonlinear information* [Conference session]. Proceedings of the AAAI Conference on Artificial Intelligence, Vancouver, Canada. https://doi.org/10.1609/aaai.v34i04.5993
11. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv:1810.04805[cs.CL*]. http://arxiv.org/abs/1810.04805
12. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. arXiv:1706.03762[cs.CL]. https://arxiv.org/abs/1706.03762
13. Indurkhya, N., Damerau, F. J. (2010). *Handbook of natural language processing*. (2nd ed.) Chapman and Hall/CRC. https://doi.org/10.1201/9781420085938
14. Mishra, S., & Soni, D. (2020). Smishing detector: A security model to detect Smishing through SMS content analysis and URL behavior analysis. Future Generation Computer Systems, 108, 803–815. https://doi.org/10.1016/j.future.2020.03.021
15. Mishra, S., & Soni, D. (2022). Implementation of 'Smishing Detector': An efficient model for Smishing detection using neural network. *SN Computer Science*, *3*(189), 189. https://doi.org/10.1007/s42979-022-01078-0
16. Mishra, S., & Soni, D. (2023a). DSmishSMS-A system to detect Smishing SMS. Neural Computing and Applications, 35(7), 4975–4992. https://doi.org/10.1007/s00521-021-06305-y
17. Sonowal, G., & Kuppusamy, K. S. (2018). SmiDCA: An anti-Smishing model with machine learning approach. The Computer Journal, 61(8), 1143–1157. https://doi.org/10.1093/comjnl/bxy039

 doi:10.20944/preprints202508.1703.v1

18. Joo, J. W., Moon, S. Y., Singh, S., & Park, J. H. (2017). S-Detector: An enhanced security model for detecting Smishing attack for mobile computing. Telecommunication Systems, 66(1), 29–38. https://doi.org/10.1007/s11235-016-0269-9
19. Harichandana, B. S. S., Kumar, S., Ujjinakoppa, M. B., & Raja, B. R. K. (2024). COPS: A compact on-device pipeline for real-time Smishing detection. arXiv:2402.04173[cs.CR]. http://arxiv.org/abs/2402.04173
20. Verma, S., Ayala-Rivera, V., & Portillo-Dominguez, A. O. (2023, November 6-10). Detection of phishing in mobile instant messaging using natural language processing and machine learning. 2023 11th International Conference in Software Engineering Research and Innovation (CONISOFT),Guanajuato, Mexico. https://doi.org/10.1109/CONISOFT58849.2023.00029
21. Min, B., Ross, H., Sulem, E., Veyseh, A. P. B., Nguyen, T. H., Sainz, O., Agirre, E., Heintz, I., & Roth, D. (2024). Recent advances in natural language processing via large pre-trained language models: A survey. *ACM Computing Surveys*, *56*(2), 1–40. https://doi.org/10.1145/3605943
22. Treviso, M., Lee, J.-U., Ji, T., Aken, B. van, Cao, Q., Ciosici, M. R., Hassid, M., Heafield, K., Hooker, S., & Raffel, C. (2023). Efficient methods for natural language processing: A survey. *Transactions of the Association for Computational Linguistics*, *11*, 826–860. DOI:10.1162/tacl_a_00577
23. Salman, M., Ikram, M., & Kaafar, M. A. (2024). Investigating evasive techniques in SMS spam filtering: A comparative analysis of machine learning models. *IEEE Access*, *12*, 24306–24324. https://doi.org/10.1109/ACCESS.2024.3364671
24. Ma, S. (2023, May 12). *Enhancing NLP model performance through data filtering* (Technical Report No. UCB/EECS-2023-170). Electrical Engineering and Computer Sciences, University of California, Berkeley.
25. Uddin, M. A., Islam, M. N., Maglaras, L., Janicke, H., & Sarker, I. H. (2024). ExplainableDetector: Exploring transformer-based language modeling approach for SMS spam detection with explainability analysis. arXiv:2405.08026[cs.LG] http://arxiv.org/abs/2405.08026
26. Tabani, H., Balasubramaniam, A., Marzban, S., Arani, E., & Zonooz, B. (2021, September 1-3). Improving the efficiency of transformers for resource-constrained devices [Conference session]. *2021 24th Euromicro Conference on Digital System Design (DSD)*, Palermo, Italy.
27. Muralitharan, J., & Arumugam, C. (2024). Privacy BERT-LSTM: A novel NLP algorithm for sensitive information detection in textual documents. Neural Computing and Applications, 36(25), 15439–15454. https://doi.org/10.1007/s00521-024-09707-w
28. Khan, M. A., Huang, Y., Feng, J., Prasad, B. K., Ali, Z., Ullah, I., & Kefalas, P. (2023). A multi-attention approach using BERT and stacked bidirectional LSTM for improved dialogue state tracking. *Applied Sciences*, *13*(3), 1775. https://www.mdpi.com/2076-3417/13/3/1775
29. Remmide, M. A., Boumahdi, F., Ilhem, B., & Boustia, N. (2025). A privacy-preserving approach for detecting smishing attacks using federated deep learning. *InternationalJournal of Information Technology*, *17*(1), 547-553. https://doi.org/10.1007/s41870-024-02144-x
30. Almeida, T. A., Hidalgo, J. M. G., & Yamakami, A. (2011, September 19-22). Contributions to the study of SMS Spam filtering: New collection and results. DocEng '11:Proceedings of the 11th ACM Symposium on Document Engineering, Mountain View, CA, USA, 259–262. https://doi.org/10.1145/2034691.2034742
31. Alpaydin, E. (2020. Introduction to machine learning (4th ed.) The MIT Press. https://mitpress.mit.edu/9780262043793/introduction-to-machine-learning/
32. Bishop, C. M., (2006). Pattern recognition and machine learning (Vol. 4). Springer. https://link.springer.com/book/9780387310732
33. Géron, A. (2022). Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow. (3rd ed.). O'Reilly Media, Inc. https://www.oreilly.com/library/view/hands-on-machine-learning/9781492032632/
34. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research, 15(56), 1929–1958. https://www.cs.toronto.edu/~rsalakhu/papers/srivastava14a.pdf
35. Bengio, Y. (2012). Practical recommendations for gradient-based training of deep architectures. In G. Montavon, G. B. Orr, & K.-R. Müller (Eds.), Neural Networks: Tricks of the Trade: Second Edition (pp. 437–478). Berlin, Germay: Springer. https://doi.org/10.1007/978-3-642-35289-8_26

 doi:10.20944/preprints202508.1703.v1

36. Johnson, R., & Zhang, T. (2015). Semi-supervised convolutional neural networks for text categorization via region embedding [Conference presentation]. NIPS '15: Proceedings of the 28th International Conference on Neural Information Processing Systems – Volume 1, Cambridge, MA, USA. doi/10.5555/2969239.2969342 https://proceedings.neurips.cc/paper/2015/file/acc3e0404646c57502b480dc052c4fe1-Paper.pdf
37. Houston, R. A. (2024). *Transformer-enhanced text classification in cybersecurity: GPTAugmented synthetic data generation, BERT-based semantic encoding, and multiclass analysis* [Unpublished PhD Thesis] The George Washington University. https://search.proquest.com/openview/fe2a7d3fb1e4ac4426755c3237663c7c/1?pqorigsite=gscholar&cbl=18750&diss=y