An Enhanced Machine Learning with NLP Modelling Technique for Smishing Attacks Detection in Low-Resourced Languages

Aaron Zimba, Katongo Ongani Phiri
ZCAS University, Lusaka, Zambia

Abstract

Smishing, a form of phishing through SMS, has emerged as a significant cybersecurity threat, particularly on mobile money platforms in regions with limited cybersecurity awareness. This research introduces a robust machine learning model integrated with advanced natural language processing (NLP) techniques for effective smishing detection. The proposed model targets English and Bemba, a low-resourced language, addressing a critical gap in cybersecurity research for linguistically diverse, resource-constrained environments. The model incorporates pseudonymization to enhance data security by anonymizing sensitive information such as personal identifiers while retaining the contextual integrity of messages. Named Entity Recognition (NER) is employed to detect and mask sensitive entities, further safeguarding user privacy. To bolster model robustness against adversarial attacks, adversarial training is applied, exposing the model to perturbed inputs during training to improve its resilience to manipulation. Regularization techniques, specifically L1 regularization, are used to optimize the model by reducing overfitting and ensuring efficient performance. The evaluation utilized datasets in English, Bemba, and a combination of both to assess the model's adaptability to multilingual inputs. The results demonstrate superior performance, with high F1-Scores, low log loss, and AUC values exceeding 0.97 across datasets. These metrics underscore the model's capability to distinguish between smishing and legitimate messages effectively. By combining machine learning and NLP in a privacy-preserving and security-enhanced framework, this research provides a scalable, efficient solution for smishing detection in under-resourced contexts, contributing significantly to advancements in cybersecurity for low-resourced languages.

Keywords: pseudonymization, low-resourced language, adversarial training, mobile money platforms, data privacy

1. INTRODUCTION

Machine learning (ML), a subset of Artificial Intelligence (AI) is increasingly being applied in a wide variety of application domains including education, healthcare, vehicular networks, intelligent manufacturing, among others **[22].** Machine learning models have constantly been improved to scale to meet the demand while maintaining efficiency. The need to optimize machine learning models is a vital factor that must be implemented. These models rely on sensitive data that potentially create potential security and privacy risks. A major hurdle in developing ML systems is the requirement for vast amounts of training data. While it's logical that more data leads

to better model performance, gathering large datasets from diverse sources poses significant privacy risks. The collection, use, and processing of this data, along with the creation and application of ML models, can inadvertently expose sensitive or confidential information **[22].** Therefore, there's need to use techniques that will mask sensitive information such as phone numbers, email addresses, among others in a dataset. Machine learning models can be vulnerable to privacy attacks. Membership inference attacks aim to determine if a particular data point was used in training, while model inversion seeks to recover sensitive details about individuals. Since these attacks exploit similar vulnerabilities, anonymizing training data could potentially mitigate both risks **[27].**

Pseudonymization and anonymization are some of the traditional techniques that are widely used to preserve privacy. Data anonymization involves modifying raw data to remove personally identifiable information. This process aims to make the data less sensitive and reduce the risk of re-identifying individuals. Importantly, anonymized data is often exempt from strict data protection regulations like GDPR, allowing for more flexible use, analysis, sharing, and monetization **[27].** However, **[34]** stated simple anonymization or pseudonymization, are now easily circumvented by malicious actors equipped with powerful computational tools and algorithms.

Furthermore, adversarial techniques can be explored in which a network is trained on adversarial examples, is one of the few defenses against adversarial attacks that withstand strong attacks **[20].** In security-critical applications, robustness against adversarial attacks has become essential, as these attacks exploit vulnerabilities in systems to compromise their reliability and integrity. Ensuring robustness not only strengthens defense mechanisms but also enhances trust in applications where data confidentiality, integrity, and availability are paramount.

In addition, implementing security aspects on a model might reduce its performance and efficiency and thus there's need to optimize the model. Regularization is a technique used to mitigate overfitting in machine learning models. Overfitting happens when a model becomes overly complex, capturing noise in the data instead of the true underlying patterns. As a result, the model may excel on its training data but perform poorly on unseen data. Regularization addresses this by introducing a penalty term to the loss function, promoting simpler and more generalizable models. This ensures that the model not only fits the training data but also maintains strong performance

on new datasets **[35].** This paper proposes a machine learning and natural language processing model against Smishing attacks on mobile money platforms that utilizes techniques to enhance security through masking sensitive information while ensuring the machine learning model is efficient.

The rest of the paper is organized as follows: The Related Works and the comparisons thereof are presented in Section 2 while the Methodology and the developed model are presented in Section 3. The Results and Discussions come in Section 4 and the Conclusion is drawn in Section 5.

2. RELATED WORK

Our approaches to enhance security is through anonymization with NER to ensure privacy and data preservation while optimizing our machine learning model through adversarial techniques and regularization incorporates NLP techniques like Named Entity Recognition (NER) and Part-of-Speech (POS) tagging, drawing on approaches from other works. As such, we focus on related works that are based on these concepts.

[34] proposes the use of a Semantic K-Anonymity Framework that addresses the need for implementing robust privacy protection mechanism maintaining the intrinsic value of the data for analytical pursuits. A notable gap in the research is the absence of discussion on how the integration of the framework ensures that algorithm efficiency is maintained or optimized. While the framework is described as adaptable to various Θ thresholds, it does not provide clarity on its generalizability across different domains (e.g., healthcare, finance). Anonymization techniques often encounter domain-specific challenges, as different types of data may require specialized treatment. Thus, there's need to explore the proposed framework in a domain-specific setup.

[19] proposes using NER to improve the security of data. Sequence tagging, commonly employed in tasks like Named Entity Recognition (NER), can be useful for identifying private information. However, training sequence tagging models requires a substantial amount of labeled data, which poses a challenge in privacy-sensitive domains since such data cannot be shared directly. To optimize these models, Part-of-Speech (POS) tagging is a technique that could be explored, leveraging Natural Language Processing (NLP) capabilities for improved tagging accuracy.

Named Entity Recognition (NER) is a Natural Language Processing (NLP) technique that identifies and categorizes entities like names, locations, organizations, dates, and other specific

elements within text. NER can detect and anonymize sensitive information, such as personal names, addresses, or identification numbers, reducing the risk of exposing private data. This is particularly useful for compliance with data privacy regulations (e.g., GDPR) as GDPR does not consider anonymized data as personal data **[27].**

Experimental results demonstrate that models developed through machine-learning-as-a-service platforms can expose significant amounts of sensitive information about their training datasets. This vulnerability occurs because these platforms often provide query-based access to the models, which can be exploited using techniques such as membership inference or model inversion attacks. These findings highlight critical privacy and security concerns, particularly in applications that handle confidential or personal data, emphasizing the need for robust mitigation strategies to safeguard against such leaks **[18].** There's limited exploration of how membership inference attacks vary across different types of machine learning models (e.g., neural networks, decision trees, ensemble methods) and diverse datasets. The research focuses on specific commercial models and a particular dataset but does not investigate how these attacks perform on a broader range of model architectures or in different contexts. Additionally, the study does not address the application of these techniques in other industries, such as finance or social media, where privacy risks may differ. Exploring these aspects could provide a more comprehensive understanding of the privacy risks posed by machine learning models.

Determining whether a specific data record was included in a model's training dataset can reveal information leakage. If an adversary has full knowledge of a record and discovers it was used to train a model, this could indicate a breach of information through the model. In certain scenarios, this may directly result in a privacy violation. For instance, if it is known that a patient's clinical record was used to train a model related to a disease—such as one designed to determine medication dosages or uncover genetic factors—it could inadvertently disclose that the patient has the disease in question **[18].**

In specialized applications, Named Entity Recognition (NER) plays a pivotal role by extracting relevant information tailored to specific domains, such as medical terms in healthcare or financial entities in banking, to support targeted and domain-specific analysis. Similarly, in the context of smishing detection, NER can identify critical elements like sender identities, transaction references, and monetary amounts within fraudulent SMS messages. By isolating and analyzing

these entities, NER enhances the precision of smishing detection models and aids in uncovering patterns that distinguish genuine communications from malicious ones.

Moreover, incorporating NER into smishing detection workflows can serve as an added layer of defense against membership inference attacks. Since NER focuses on extracting high-level patterns and features rather than directly exposing raw data, it reduces the likelihood of adversaries deducing specific training records. This abstraction not only enhances model accuracy in detecting smishing but also strengthens privacy safeguards, ensuring that sensitive user information, such as mobile numbers or transaction details, remains protected against potential leaks.

In the field of Natural Language Processing, some researchers employ data enhancement techniques that involve simple operations to enrich datasets, ensuring the sentence's meaning and the associated target entity remain unchanged. However, in real-world applications, it is impossible to impose restrictions on the synonyms that attackers might use **[17].**

[16] proposed generating adversarial samples for deep neural networks based on text. In their approach, the objective function of the text classifier is denoted as f(x), where x = {w1, w2, ..., wL} is the original input sample, and L represents the text length. Assuming the correct classification of sample x is y, the adversarial sample x' = {w'1, w'2, ..., w'L} is created, and the model's prediction is evaluated for x'.

The attack is considered successful if the following conditions hold:

$f(x') \neq y$,

$Sim(x, x') \geq nmin$,

where Sim(x, x') represents the semantic similarity between x and x', and nmin denotes the minimum acceptable similarity between the two texts.

Adversarial examples, as introduced by **[15]** highlight the susceptibility of machine learning models, especially deep neural networks, to small, deliberate perturbations in the input data. It is argued that adversarial training, which involves training models on both original and adversarial perturbed examples, can significantly improve model generalization and make them resistant to

adversarial attacks **[15].** This approach becomes even more relevant when working with small datasets, as adversarial training helps expose the model to a more diverse set of examples without needing to expand the dataset size.

[14] extend the concept of adversarial examples to the physical world, demonstrating that machine learning models are not only vulnerable to adversarial examples in controlled environments but also to perturbations in real-world scenarios. This study emphasizes the necessity of training models on adversarial samples to ensure robustness.

A key study by **[12]** introduces FinChain-BERT, a model optimized for financial fraud detection, which combines deep learning techniques with NLP to improve the handling of financial language. The model demonstrates improved performance by employing a Keywords Loss Function and integer distillation technology to reduce its size while maintaining high accuracy. However, exploring how traditional Natural Language Processing (NLP) techniques, such as Named Entity Recognition (NER) and Part-of-Speech (POS) tagging, can be leveraged to optimize detection in low-resource languages with diverse notations in domain-specific contexts could enhance evaluation across a broader range of scenarios.

Regularization is an essential technique in machine learning aimed at preventing overfitting, ensuring that models generalize well to unseen data. It works by adding a penalty term to the loss function, which discourages the model from becoming excessively complex or fitting too closely to the noise in the training data. Common methods of regularization include **L1 regularization (Lasso)**, which induces sparsity by forcing some coefficients to zero, and **L2 regularization (Ridge)**, which penalizes large coefficients, thereby stabilizing the learning process.

There's need to explore how regularization can be effective in a diverse dataset by optimizing a model's efficiency.

The table below shows a comparison of related work:

Work	Classifier	Domain	Language	Approach	**Data Security**	**Model Security**	**Model Optimization and Performance Tuning**	**METRIC**
[33]	Random Forest	Smishing	Swahili	Machine Learning	Not Mentioned	Not Mentioned	Not Mentioned	Log-Loss, AUC & Exec. time
[3]	Decision Tree, PRISM, RIPPER	Smishing	English	Rule Based	Not Mentioned	Not Mentioned		TPR, FPR, TNR, FNR
[2]	Neural Network	Smishing	English	forward propagation and backward propagation	Not Mentioned	Not Mentioned	loss function	F1 and Accuracy
[1]	SVM, RF, LR	Smishing	English	Natural Language Processing	Not Mentioned	Not Mentioned	Not Mentioned	F1
Proposed Model	Random Forest	Smishing	English and Bemba	Machine Learning And Natural Language Processing	Pseudomization	Adversarial Training	Regularization	F1, Accuracy, AUC and MCC

Existing models in the domain of smishing detection show various limitations in the areas of data security, model security, and optimization. Many of these models, such as the ones using random forest [33], decision trees [3], and neural networks [2], do not provide any information on securing data. The absence of measures to safeguard the data used for training these models exposes them to privacy risks, particularly in sensitive contexts like smishing detection.

In addition, several of these models fail to address **model security**. For instance, while machine learning models, such as Support Vector Machines (SVM) or random forests (RF), may be effective in detection, the vulnerability of these models to adversarial attacks is not acknowledged or mitigated. This leaves these models open to manipulation, potentially undermining their reliability and trustworthiness in real-world applications.

Furthermore, there is a noticeable **lack of model optimization and performance tuning** in the existing literature. Despite using various machine learning techniques, only a few papers include strategies like hyperparameter tuning or regularization to improve model performance. Moreover, many existing models focus primarily on basic evaluation metrics such as accuracy or F1-score

without delving into other optimization techniques that can help to improve robustness and efficiency. This oversight can result in suboptimal performance, especially in challenging real-world scenarios.

Lastly, the **metrics used for model evaluation** are often limited. While common metrics such as True Positive Rate (TPR), False Positive Rate (FPR), and accuracy are considered, they may not capture the full picture of the model's performance. For instance, aspects like **model robustness** is seldom discussed, despite it's importance in practical deployment.

Our **proposed model** addresses these gaps by incorporating **pseudonymization** for data security, **adversarial training** for model security, and **regularization** to optimize performance. Additionally, we evaluate our model using a more comprehensive set of metrics, such as **F1**, **Accuracy**, **AUC**, and **MCC**, to provide a more thorough understanding of its effectiveness and robustness in smishing detection. These improvements make the proposed model a more secure, optimized, and reliable solution compared to previous efforts in the field.

3. METHODOLOGY

The framework used was designed to utilize the strengths of natural language processing to enhance the machine learning model and produce good results. The framework starts with data collection to pre-processing and performance evaluation. This is depicted in Figure 1.

The proposed detection framework for smishing attacks in low-resourced languages integrates advanced machine learning and NLP techniques, focusing on privacy preservation, robustness, and adaptability. The process begins with dataset preprocessing, including text normalization and noise removal, followed by a decision to determine whether the data is masked. If unmasked, pseudonymization and custom Named Entity Recognition (NER) masking are applied to anonymize sensitive entities. Tokenization then converts text into structured input, preparing it for machine learning. L1 regularization is employed to ensure sparsity and focus on relevant features, addressing the data scarcity challenges of low-resourced languages. Adversarial examples are generated to test and enhance the model's robustness against evasion attacks. Finally, the framework trains a detection model using the processed data and adversarial samples, enabling

accurate classification of SMS messages as benign or malicious, thereby mitigating smishing threats effectively.

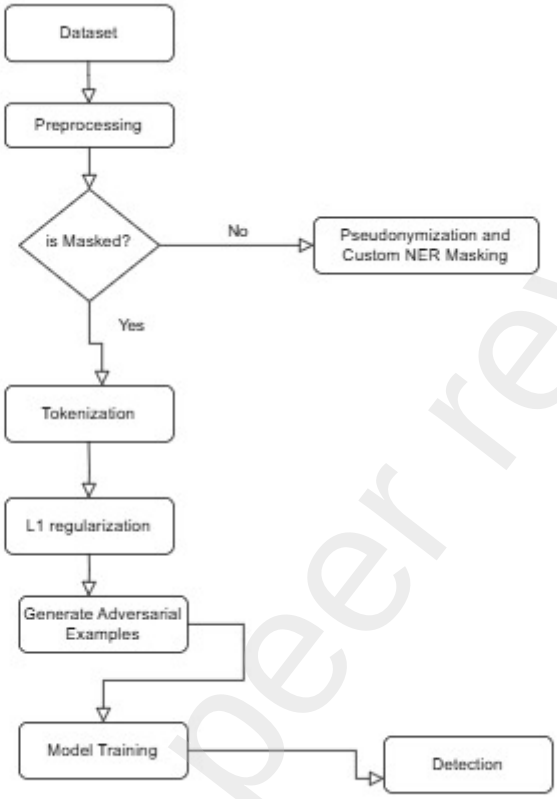

Figure 1. Smishing detection framework

A. DATA COLLECTION AND PREPROCESSING

The datasets utilized in this research consisted of texts in both Bemba and English, necessitating a two-step preprocessing approach tailored to the unique characteristics of each language. Given that Bemba is a low-resource language with limited library and tool support, additional preprocessing efforts were required, including the manual addition of Bemba stop words to a custom Named Entity Recognition (NER) dictionary. This step was essential to enhance the accuracy of the model by accounting for linguistic nuances and addressing the challenges posed by the scarcity of readily available resources for Bemba language processing. Stop words such as (“fye”, “shani”….) were used.

Furthermore, tokenization was employed as a foundational step in the preprocessing pipeline, followed by the application of Term Frequency-Inverse Document Frequency (TF-IDF) vectorization to numerically represent the text data. This process was carried out after converting all text to lowercase to ensure uniformity and eliminate variations due to capitalization. Each feature extracted during preprocessing, including tokenized terms and their corresponding TF-IDF values, plays a critical role in enabling the model to accurately distinguish between Smishing messages and legitimate communications by capturing essential patterns and linguistic differences.

B. INTEGRATION OF MODEL

Our proposed approach focuses on securing sensitive information and ensuring the efficient performance of machine learning models. In many developing countries, cybersecurity awareness is often overlooked as a critical aspect of technology adoption. This is evident from the survey results we obtained, where 70% of a sample size of 400 individuals were unaware of Smishing attacks, a significant cybersecurity threat targeting mobile money platforms. This lack of awareness is a key motivator for developing a machine learning model to detect Smishing attacks effectively.

During this study, we observed that most related works did not address securing sensitive information in datasets, which can make them vulnerable to attacks such as Membership Inference Attacks (MIA). These attacks attempt to determine whether a specific data sample was included in the training set of a machine learning model and are increasingly used to evaluate the privacy risks of language models and other machine learning systems.

To address these concerns, our approach incorporates several privacy-preserving and regularization techniques. We apply pseudonymization to replace sensitive identifiers in the dataset, ensuring that individual data points cannot be traced back to specific users, thereby enhancing privacy. In Addition, to prevent overfitting and further secure the model, we implement L1 regularization, which promotes sparsity in the model parameters, ensuring that only the most relevant features are used, thereby improving generalization and reducing the risk of memorizing sensitive information from the training data. To bolster the model's robustness, we use adversarial training. This technique generates adversarial examples by introducing small perturbations to the

input data, making the model more resilient to malicious attempts to deceive it, such as adversarial attacks **[15][7].**

By combining these techniques, we aim to build a machine learning model that not only detects Smishing attacks effectively but also ensures the privacy and security of sensitive information. This approach contributes to the broader effort to secure mobile money platforms in regions with limited cybersecurity awareness and protection.

C. PSEUDONYMIZATION

Several techniques were considered and one of them was pseudonymization. Pseudonymization, as defined by the GDPR, refers to the processing of personal data in a way that prevents it from being linked to a specific individual without the aid of additional information. This additional information is stored separately and safeguarded through technical and organizational measures to ensure that the data cannot be associated with an identifiable individual **[36].**

In simpler terms, Pseudonymization refers to the processing of personal data in a way that it can no longer be linked to a specific individual without additional information. This additional information must be stored separately and protected with technical and organizational measures to ensure that the data cannot be connected to an identifiable person **[35].**

Anonymization is another technique that was considered to secure sensitive information. Anonymization involves the permanent elimination of all information that could act as an identifier. After a dataset is anonymized, it becomes impossible to identify any individuals from the data **[36].** However, Anonymization is unsuitable for a smishing dataset because it removes critical identifiers and context necessary for detecting patterns, validating models, and maintaining real-world representativeness. Pseudonymization is preferred as it preserves essential information while protecting privacy. In our approach the Pseudonymization technique that was used was tokenization. Tokenization replaces sensitive data with non-sensitive data, called tokens, that have no meaning or value. This method doesn't change the length or type of data, so it can be processed by systems that are sensitive to those characteristics. Furthermore, NER was used to help identify entities in the text, such as names, locations, or other sensitive information. A custom NER dictionary was developed because the dataset contains Bemba, a low-resource language with

limited support for NER. The approach described in **[11]** suggests that an NER model designed for anonymization should be trained specifically to treat sensitive data as named entities.

The implemented Pseudonymization Process is presented as follows:

Algorithm 1: Pseudonymization Algorithm

Step 1: Let T denote an input text document, and $T = \{T_1, T_2, \ldots, T_m\}$ be a dataset of m such documents. The goal is to transform T into a pseudonymized representation T^* while preserving semantic and structural utility for downstream tasks.

Step 2: The preprocessing step involves text cleaning to standardize the input, defined as:

$F_{\text{clean}}(T)$ = lowercase(re.sub(non-alphanumeric characters, "", T)) Let T' represent the cleaned text: $T' = Fclean(T)$

Step 3: Named Entity Recognition (NER) Function F_{NER}. An NLP model N is employed to identify entities in T, generating a set of entities E: $E = FNER(T', N) = \{e1, e2, \ldots, en\}$

Each entity e_i is represented as a tuple: $e_i = (text_i, label_i)\ \forall i \in \{1, 2, \ldots, n\}$

Step 4: Entity Replacement Function F_{replace}. For each identified entity e_i, replace text_i with a pseudonymized placeholder $\langle \text{label}_i \rangle$ *in* T': $T* = Freplace(T', E) = T'.replace(texti, \langle labeli \rangle)\ \forall ei \in E$

Step 5: Custom NER Extensions F_{custom}. Extend N with domain-specific patterns $P = \{p_1, p_2, \ldots, p_k\}$ using an entity ruler to identify entities unique to the application context. This process is defined as: $N' = Fcustom(N, P)$

where N' represents the augmented NLP model.

Step 6: Dataset Transformation. For a dataset T, the transformation function $F_{\text{pseudonymize}}$ is applied to each document: $pseudonymize\ (Ti, N')$

Complete Pseudonymization Algorithm:

$$Fpseudonymize(T, N) = Freplace(Fclean(T), FNER(Fclean(T), N'))$$

This pseudonymization framework combines preprocessing, enhanced NER capabilities, and systematic entity replacement, enabling secure and effective handling of sensitive text data, particularly in low-resource language contexts. The methodology ensures data privacy while preserving linguistic and contextual integrity for tasks like smishing detection.

D. OPTIMIZATION

Feature selection is a machine learning technique used to identify a subset of relevant variables for model construction. The goal of feature selection is to eliminate redundant or irrelevant features, or those that are highly correlated, while preserving the essential information. This technique is commonly employed to simplify model interpretation and improve generalization by reducing variance **[8].** Shrinkage methods aim to minimize the residual sum of squares in a model using Ordinary Least Squares (OLS), while also reducing model complexity, such as the number or size of coefficients. Unlike subset selection or dimension reduction methods, shrinkage allows fitting a model with all predictors, applying regularization to the estimated coefficients. This regularization reduces variance and can also perform variable selection. Two key techniques in shrinkage are Ridge Regression and LASSO. Ridge regression applies L2 regularization by minimizing the squared sum of coefficients, while LASSO uses L1 regularization to minimize the absolute sum of coefficients. These methods are particularly effective in handling collinearity in the data, where Ordinary Least Squares would typically overfit. In our model, L1 regularization was used due to optimize it after preprocessing the dataset securely with pseudonymization due its simplicity. This approach can be reflected with this formula:

$$w^{\wedge} = \text{argmin}(\text{MSE}(w) + \lambda \|w\|_1)\ w$$

where:

$$n\ \text{MSE}\ (^{w}) = \frac{1}{n} \sum_{i=1} (y_i - \mathbf{x_i}^T w)^2 \_$$

is the Mean Squared Error (MSE) between the true output values y_i and the predicted values based on the input features $\mathbf{x_i}$, and

$$\|w\|_1 = \sum_{j=1}^{p} |w_j|$$

is the L1 norm of the weight vector w, which is the sum of the absolute values of the coefficients. The parameter λ controls the strength of the regularization.

The LASSO regression with cross-validation (LassoCV) is applied to find the optimal regularization parameter α. The optimization is done by minimizing the following objective function:

$$\hat{w} = \left(\frac{1}{n} \sum_{i=1}^{n} (y_i - \mathbf{x_i}^T w)^2 + \lambda \|w\|_1 \right) \quad \text{argmin } w$$

Where:

- $\hat{w}$ are the coefficients to be estimated.
- y_i are the true output values.
- $\mathbf{x_i}$ are the input features.
- λ is the regularization parameter that controls the strength of the L1 penalty (the LASSO term).
- $\|w\|_1 = \sum_{j=1}^{p} |w_j|$ is the L1 norm of the weight vector w.

The cross-validation process selects the best α from the set of candidate values $\{0.0001, 0.001, 0.01, 0.1, 1, 10\}$. The optimal value of α is identified based on the lowest cross-validation error. The following code describes the process:

Algorithm 2: LASSO with Cross-Validation and Feature Selection

Input: Training data X_{train}, y_{train}
Output: Selected features *X*train selected, *X*test selected
Initialize LASSO model: *lasso = LassoCV(alphas=[0.0001, 0.001, 0.01, 0.1, 1, 10], cv=5, random state=42)*
Fit LASSO model: *lasso.fit(Xtrain dense,ytrain)*
Optimal *alpha:lasso.alpha*
Non-zero coefficients: *np.sum(lasso.coef_ != 0)*
Coefficients: *lasso.coef_*

Initialize feature selector: *selector = SelectFromModel(lasso, prefit=True)*

Apply feature selection on training and testing data:

Xtrain selected = selector.transform(Xtrain dense) Xtest selected = selector.transform(Xtest dense)

E. ADVERSARIAL TRAINING

[15] demonstrated that adversarial training plays a significant role in developing resilient neural networks by defending against adversarial examples. In the process of adversarial training, gradients are derived from clean data samples to generate slight perturbations. These gradients are then constrained within a normalization ball and added to the original inputs to form adversarial examples. These adversarial inputs are incorporated into the training process to enhance the model's robustness against attacks based on gradient manipulations. Due to the discrete nature of textual data, this method is not directly applicable to NLP tasks **[9].** In the field of natural language processing (NLP), adapting gradient-based adversarial attack and training methods is challenging due to the discrete nature of the embedding space, where gradients cannot be directly applied to generate perturbations. Unlike in continuous spaces, where small adjustments can be made using gradients, NLP models typically rely on token embeddings that require different techniques for perturbation generation **[5].**

Our Proposed Generative Adversarial examples algorithm is as follow:

Require:

- Training Samples: Xtrain selected, Test Samples: Xtest selected
- Perturbation Bound: ϵ, Perturbation magnitude

Algorithm 3: Generate Adversarial Examples

1: **Input:** Training Samples X, Perturbation Bound ϵ
2: **Output:** Adversarial Examples X_{adv}
3: Generate Noise: δ = uniform$(-\epsilon, \epsilon, X.\text{shape})$
4: Generate Adversarial Example: $X_{\text{adv}} = X + \delta$
5: Clip the Values: $X_{\text{adv}} = \text{clip}(X_{\text{adv}}, 0, 1)$ (valid input range)
6: **Return:** X_{adv}

Training Process:

1. Generate Adversarial Examples for Training Data:

Xtrain adv = generate adversarial examples(Xtrain selected, $\epsilon = 0.1$)

2. Combine Adversarial and Original Training Data:

Xtrain combined = vstack(Xtrain selected,Xadv)

$y$train combined = hstack($y$train,$y$train)

3. Shuffle the Data: Xtrain combined,$y$train combined = shuffle(Xtrain combined,$y$train combined,random state = 42)

4. Generate Adversarial Examples for Test Data:

$X_{\text{test adv}}$ = generate adversarial examples($X_{\text{test selected}}, \epsilon = 0.1$)

5. Make Predictions on Adversarial Test Data:

$y$pred adv = best model.predict(Xtest adv)

This focuses on improving machine learning model robustness through adversarial training. By generating adversarial examples—data modified with small perturbations, the model learns to handle noisy, adversarial inputs. The adversarial data is combined with original training data and shuffled to prevent overfitting. The model is then evaluated using these adversarial examples to assess its resilience. This approach aims to enhance the model's ability to withstand adversarial attacks, improving its reliability in critical applications like Smishing Detection.

4. RESULTS AND DISCUSSION

In this section, we present a detailed analysis of the findings from our experiments. The model was trained on three distinct datasets using Random Forest Classifier an ensemble learning algorithm that multiple trees and introducing randomness. One of the datasets consisted solely of English text, another exclusively in Bemba, and a third combining both English and Bemba data to assess the model's capability to handle multilingual inputs. To comprehensively evaluate the model's performance, we utilized three robust metrics: the F1 score to measure the balance between precision and recall, the Area Under the Curve (AUC) to assess the model's ability to distinguish between classes across varying thresholds, and the Matthews Correlation Coefficient (MCC) to provide an unbiased metric for binary classification tasks, as well as the confusion matrix heatmaps. These metrics collectively offered a nuanced and holistic evaluation of the model's effectiveness across different linguistic and dataset configurations. The Receiver Operating Characteristic (ROC) curve, depicted in Figure 2, illustrates the True Positive Rate (TPR) against the False Positive Rate (FPR) at various classification thresholds for the Bemba dataset. The model

achieved an Area Under the Curve (AUC) of 0.9333, reflecting a high level of discriminatory power. The steep initial slope of the curve indicates the model's ability to achieve a high TPR with a minimal FPR, underscoring its effectiveness in distinguishing between positive and negative instances. The high AUC value supports the model's robustness and reliability in classification tasks.

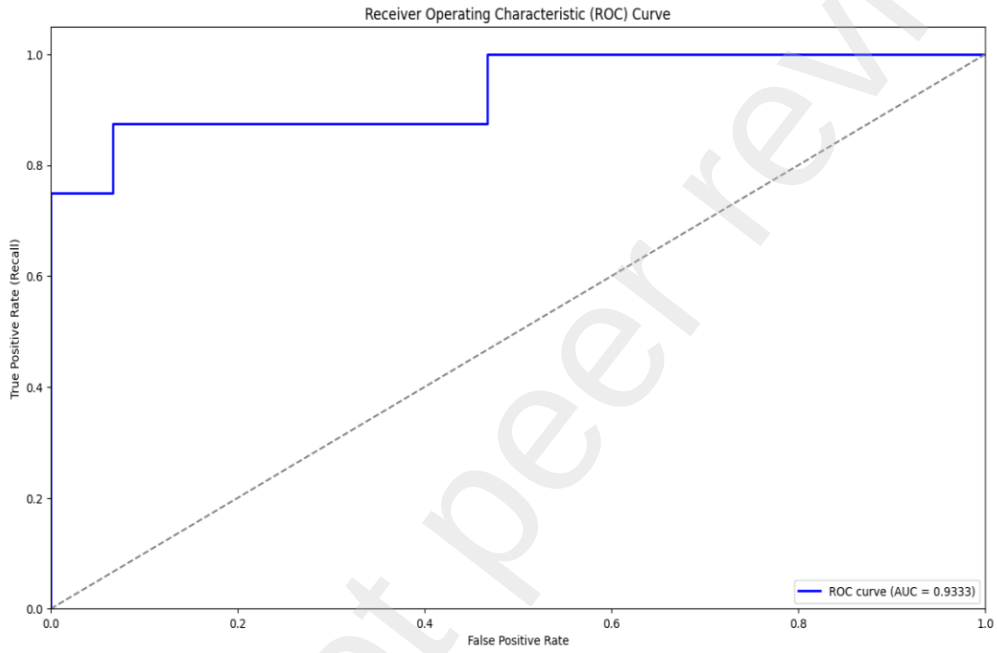

Figure 2 Bemba dataset ROC Curve

Figure 3 shows the confusion matrix heatmap for the low-resourced language, Bemba. The confusion matrix heatmap for the Bemba dataset illustrates the performance of the proposed smishing detection model. The matrix shows that out of 160 actual smishing messages, the model correctly identified 140 (true positives), with 20 being misclassified as non-smishing (false negatives). Similarly, out of 240 non-smishing messages, 200 were accurately classified as non-smishing (true negatives), while 40 were incorrectly labeled as smishing (false positives). The distribution of values in the heatmap highlights the model's high accuracy in distinguishing between smishing and non-smishing messages, particularly in a low-resourced language context, despite some room for improvement in minimizing false classifications.

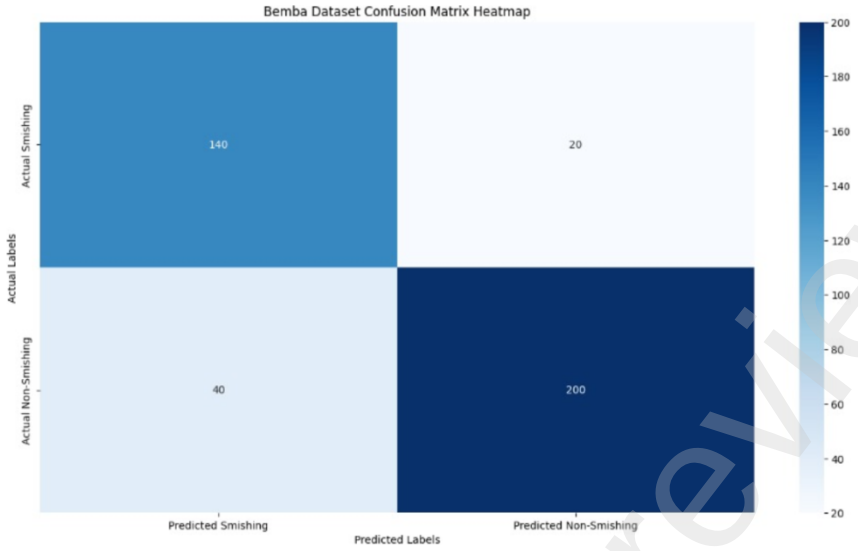

Figure 3. Confusion matrix heatmap for Bemba low-resourced language

The hyperparameter configuration optimized for the Bemba dataset included a maximum depth of 20, unrestricted features ('max_features=None'), a minimum of one sample per leaf ('min_samples_leaf=1'), a minimum of two samples per split ('min_samples_split=2'), and 200 estimators ('n_estimators=200'). This combination allowed the model to achieve an F1-Score of 0.875, Matthews Correlation Coefficient (MCC) of 0.9, Log Loss of 0.2912, and an accuracy of 0.89.

A. Performance on Combined English and Bemba Dataset

For the combined English and Bemba dataset, Figure 4 presents the ROC curve with an AUC of 0.9795. Similar to the Bemba dataset, the curve's steep initial slope highlights the model's ability to maintain a high TPR while minimizing the FPR. The model's hyperparameter configuration ('max_depth=None', 'max_features=sqrt', 'min_samples_leaf=1', 'min_samples_split=10', 'n_estimators=200') enabled superior performance metrics, including an F1-Score of 0.9171, an MCC of 0.8825, Log Loss of 0.2828, and accuracy of 0.9248. These results affirm the robustness and adaptability of the model when applied to multilingual datasets.

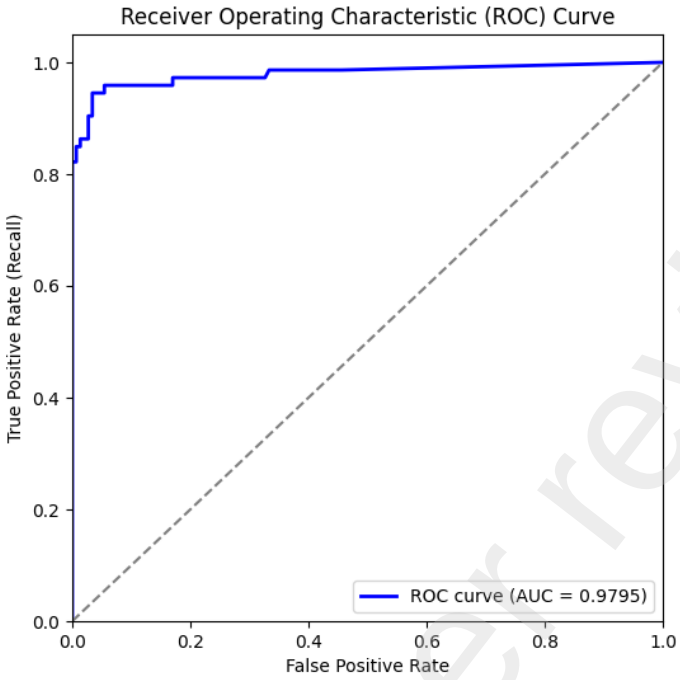

Figure 4 English and Bemba dataset ROC Curve

Figure 5 shows the confusion matrix heatmap for the multilingual dataset. The confusion matrix heatmap provides a visual representation of the model's performance on the English-Bemba dataset. The darker the color, the higher the number of correctly classified instances. The diagonal elements (1640 and 1670) represent the true positive and true negative predictions, respectively. The off-diagonal elements (185 and 145) represent the false positive and false negative predictions, respectively. The heatmap suggests that the model exhibits high accuracy in both English and Bemba, with only a small number of misclassifications.

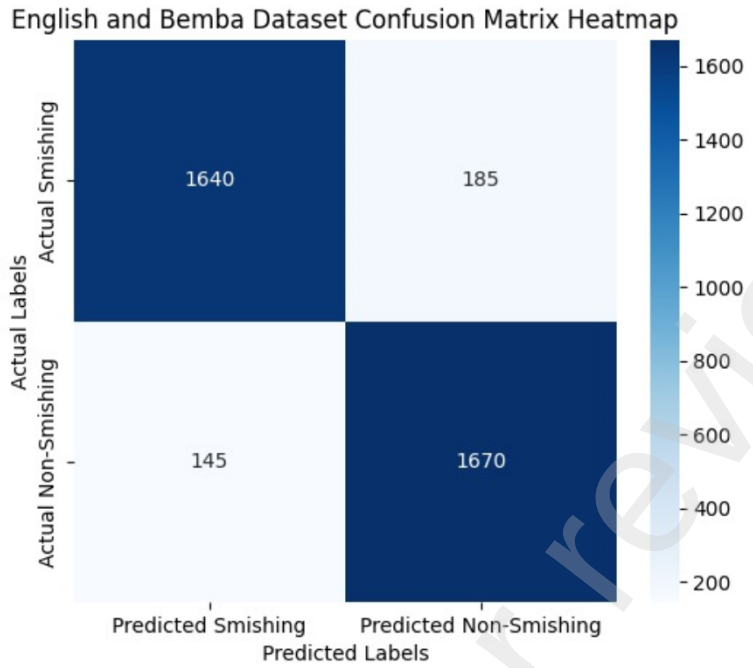

Figure 5. Confusion matrix heatmap for the multilingual dataset

B. Performance on English Dataset

Figure 6 demonstrates the ROC curve for the English dataset, where the model achieved an AUC of 0.9726. The steep slope of the ROC curve reiterates the model's efficiency in distinguishing between classes. The optimized hyperparameters ('max_depth=20', 'max_features=sqrt', 'min_samples_leaf=1', 'min_samples_split=10', 'n_estimators=200') resulted in an F1-Score of 0.924, an accuracy of 0.9362, an MCC of 0.9017, and Log Loss of 0.2001.

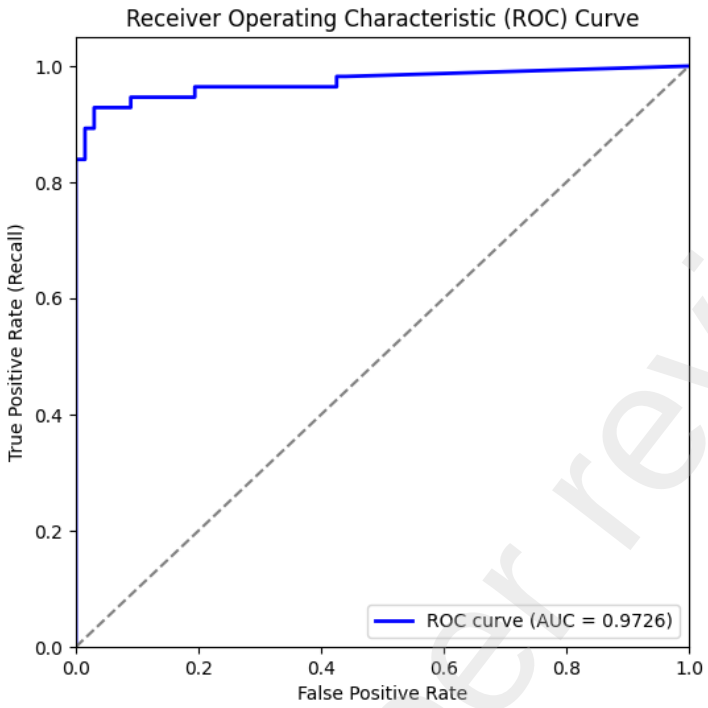

Figure 6 English Dataset ROC Curve

Figure 7 shows the confusion matrix heatmap for the English dataset The confusion matrix heatmap for the English dataset showcases the model's performance in classifying smishing and non-smishing messages. The diagonal elements (1493 and 1520) represent the true positive and true negative predictions, respectively. The off-diagonal elements (118 and 127) represent the false positive and false negative predictions, respectively. The heatmap indicates that the model performs well on the English dataset with only a small number of misclassifications.

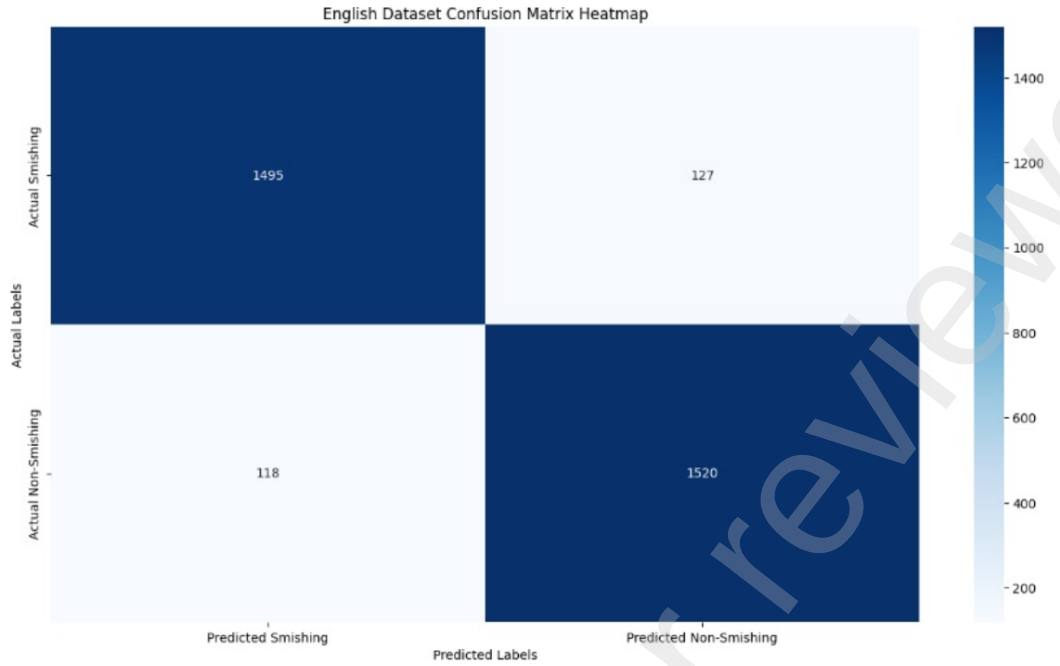

Figure 7. Confusion matrix heatmap for English dataset

C. Discussion of Hyperparameters and Performance Metrics

The selection of hyperparameters played a pivotal role in achieving these performance metrics across datasets. Notably:

- **Maximum Depth ('max_depth'):** For the Bemba dataset, a restricted depth (20) facilitated effective learning by controlling overfitting, whereas an unrestricted depth ('None') for the combined dataset allowed the model to explore more complex patterns.
- **Feature Selection ('max_features'):** Using 'sqrt' for the combined and English datasets leveraged a subset of features, enhancing generalization, while unrestricted features for the Bemba dataset captured nuanced details.
- **Minimum Samples:** The use of 'min_samples_leaf=1' and 'min_samples_split=10' ensured balanced splits, improving stability and reducing overfitting in multilingual datasets.
- **Number of Estimators ('n_estimators'):** A consistent value of 200 across datasets provided sufficient ensemble diversity without excessive computational cost.

Metrics such as the F1-Score, MCC, Log Loss, and accuracy were critical in evaluating the model's comprehensive performance. The MCC, in particular, offered a robust measure of classification performance, while Log Loss quantified the model's predictive uncertainty. The high accuracy values across all datasets demonstrated the model's overall effectiveness, whereas the low Log

Loss values highlighted its reliability in probability-based predictions. Furthermore, the model ensures the privacy of sensitive information through pseudonymization, a technique that effectively reduces the risk of exposing personal data while maintaining model utility. It achieves robustness against adversarial attacks, such as membership inference, by leveraging adversarial training, which enhances the model's ability to withstand adversarial manipulation. Additionally, L1 regularization promotes efficiency by selecting relevant features and preventing overfitting, ultimately improving the model's generalizability and performance across diverse datasets. Existing models for smishing detection often neglect aspects of data security, model robustness, and optimization. Our approach addresses these gaps by incorporating pseudonymization for data security, adversarial training for model robustness, and regularization techniques for optimization. Additionally, our model's evaluation goes beyond traditional metrics, employing AUC, MCC, and Log Loss to provide a comprehensive assessment of its performance. Compared to previous efforts, our model demonstrates superior discriminatory power and robustness, as evidenced by the consistently high AUC and MCC values across datasets.

5. CONCLUSION

This research has introduced an innovative machine learning framework, enhanced with natural language processing techniques, for detecting smishing attacks in both English and Bemba, a low-resourced language. By addressing critical gaps in data security, model robustness, and optimization, the proposed model demonstrates significant advancements over existing approaches. The integration of pseudonymization and Named Entity Recognition ensures privacy preservation while maintaining the semantic integrity of data. Adversarial training enhances the model's resilience against malicious inputs, and L1 regularization optimizes performance by mitigating overfitting. Evaluation across monolingual and multilingual datasets has highlighted the model's adaptability and effectiveness, with superior metrics such as high F1-scores, MCC values, and AUC exceeding 0.97. This study contributes to the field of cybersecurity by offering a scalable, privacy-preserving solution tailored to linguistically diverse and resource-constrained environments. Future work could focus on extending the framework to additional low-resourced languages and further refining adversarial defense mechanisms. By advancing smishing detection in these contexts, this research lays a foundation for enhancing mobile money platform security and user trust in under-protected regions.

REFERENCES

[1] C. Balim and E. S. Gunal, "Automatic detection of smishing attacks by machine learning methods," in Proc. 2019 1st Int. Informatics and Software Engineering Conf. (UBMYK), Ankara, Turkey, Nov. 2019, pp. 1-6, doi: 10.1109/UBMYK48245.2019.8965429.

[2] S. Mishra and D. Soni, "Implementation of 'Smishing Detector': An efficient model for smishing detection using neural network," SN Comput. Sci., vol. 3, no. 1, pp. 189, Mar. 2022, doi: 10.1007/s42979-022-01078-0.

[3] A. K. Jain and B. B. Gupta, "Rule-based framework for detection of smishing messages in mobile environment," Procedia Comput. Sci., vol. 115, pp. 456-463, 2017, doi: 10.1016/j.procs.2017.12.079.

[4] J. X. Morris, E. Lifland, J. Y. Yoo, J. Grigsby, D. Jin, and Y. Qi, "TextAttack: A framework for adversarial attacks, data augmentation, and adversarial training in NLP," arXiv preprint arXiv:2005.05909, 2020. [Online]. Available: https://doi.org/10.48550/arXiv.2005.05909.

[5] L. Li and X. Qiu, "TAVAT: Token-aware virtual adversarial training for language understanding," arXiv preprint arXiv:2004.14543, 2020. [Online]. Available: https://doi.org/10.48550/arXiv.2004.14543.

[6] N. Carlini and D. Wagner, "Adversarial examples are not easily detected: Bypassing ten detection methods," in Proc. 10th ACM Workshop on Artificial Intelligence and Security (AISec '17), Dallas, TX, USA, Nov. 2017, pp. 3-14, doi: 10.1145/3128572.3140444.

[7] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," arXiv preprint arXiv:1706.06083, 2017. [Online]. Available: https://doi.org/10.48550/arXiv.1706nn.06083.

[8] R. Muthukrishnan and R. Rohini, "LASSO: A feature selection technique in predictive modeling for machine learning," in Proc. 2016 IEEE Int. Conf. Advances in Computer Applications (ICACA), Coimbatore, India, Oct. 2016, pp. 1-5, doi: 10.1109/ICACA.2016.7887916.

[9] L. Pan, C. W. Hang, A. Sil, and S. Potdar, "Improved text classification via contrastive adversarial training," AAAI-22 Technical Track on Speech and Natural Language Processing, vol. 36, no. 10, pp. 21362-21370, 2022, doi: 10.1609/aaai.v36i10.21362.

[10] T. Bai, J. Luo, J. Zhao, B. Wen, and Q. Wang, "Recent advances in adversarial training for adversarial robustness," in Proc. Thirtieth Int. Joint Conf. Artificial Intelligence (IJCAI), 2021, pp. 4312-4321, doi: 10.24963/ijcai.2021/591.

[11] O. Bridal, "Named-entity recognition with BERT for anonymization of medical records," Linköping University, Dept. of Computer and Information Science, 2020. [Online]. Available: http://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1566701&dswid=3810.

[12] X. Yang, C. Zhang, Y. Sun, K. Pang, L. Jing, S. Wa, and C. Lv, "FinChain-BERT: A high-accuracy automatic fraud detection model based on NLP methods for financial scenarios," Inf., vol. 14, no. 9, p. 499, 2023, doi: 10.3390/info14090499.

[13] A. Adams, E. Aili, D. Aioanei, R. Jonsson, L. Mickelsson, D. Mikmekova, F. Roberts, J. F. Valencia, and R. Wechsler, "AnonyMate: A toolkit for anonymizing unstructured chat data," in Proc. Workshop on NLP and Pseudonymisation, Turku, Finland, 2019, pp. 1-7, Linköping Electronic Press.

[14] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," arXiv preprint arXiv:1607.02533, 2016. [Online]. Available: https://doi.org/10.48550/arXiv.1607.02533

[15] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," arXiv preprint arXiv:1412.6572, 2014. [Online]. Available: https://doi.org/10.48550/arXiv.1412.6572.

[16] R. Jia and P. Liang, "Adversarial examples for evaluating reading comprehension systems," in Proc. 2017 Conf. Empirical Methods in Natural Language Processing (EMNLP), Copenhagen, Denmark, 2017, pp. 2021-2031, Association for Computational Linguistics.

[17] X. Liu, S. Dai, G. Fiumara, and P. De Meo, "An adversarial training method for text classification," J. King Saud Univ. Comput. Inf. Sci., vol. 35, no. 1, pp. 101697, 2023, doi: [10.1016/j.jksuci.2023.101697.

[18] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," arXiv preprint arXiv:1610.05820, 2017. [Online]. Available: https://doi.org/10.48550/arXiv.1610.05820.

[19] A. Jana and C. Biemann, "An investigation towards differentially private sequence tagging in a federated framework," in Proc. Third Workshop Privacy Natural Language Processing, Online, 2021, pp. 30-35, Association for Computational Linguistics.

[20] A. Shafahi, M. Najibi, A. Ghiasi, Z. Xu, J. Dickerson, C. Studer, L. S. Davis, G. Taylor, and T. Goldstein, "Adversarial training for free!," arXiv preprint arXiv:1904.12843, 2019. [Online]. Available: https://doi.org/10.48550/arXiv.1904.12843.

[21] A. Girka, V. Terziyan, M. Gavriushenko, and A. Gontarenko, "Anonymization as homeomorphic data space transformation for privacy-preserving deep learning," Procedia Computer Science, vol. 184, pp. 363-370, 2021, doi: 10.1016/j.procs.2021.01.337.

[22] R. Xu, N. Baracaldo, and J. Joshi, "Privacy-preserving machine learning: Methods, challenges and directions," arXiv preprint arXiv:2108.04417, 2021, doi: 10.48550/arXiv.2108.04417.

[23] B. Zhou, J. Pei, and W. Luk, "A brief survey on anonymization techniques for privacy preserving publishing of social network data," ACM SIGKDD Explorations Newsletter, vol. 10, no. 2, pp. 12-22, 2008, doi: 10.1145/1540276.1540279.

[24] Z. Zuo, M. Watson, D. Budgen, R. Hall, C. Kennelly, and N. Al Moubayed, "Data Anonymization for Pervasive Health Care: Systematic Literature Mapping Study," JMIR Med. Inform., vol. 9, no. 10, p. e29871, Oct. 2021, doi: 10.2196/29871. PMID: 34652278; PMCID: PMC855664

[25] A. Gadotti, L. Rocher, F. Houssiau, A.-M. Creţu, and Y.-A. de Montjoye, "Anonymization: The imperfect science of using data while preserving privacy," Sci. Adv., vol. 10, no. 29, p. eadn7053, Jul. 2024, doi: 10.1126/sciadv.adn7053.

[26] A. Gadotti, L. Rocher, F. Houssiau, A.-M. Creţu, and Y.-A. de Montjoye, "Anonymization: The imperfect science of using data while preserving privacy," Science Advances, vol. 10, no. 29, p. eadn7053, Jul. 2024. [Online]. Available: https://doi.org/10.1126/sciadv.adn7053.

[27] N. Senavirathne and V. Torra, "On the Role of Data Anonymization in Machine Learning Privacy," 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Guangzhou, China, 2020, pp. 664-675, doi: 10.1109/TrustCom50675.2020.00093.

[28] B. Liu, M. Ding, S. Shaham, W. Rahayu, F. Farokhi, and Z. Lin, "When machine learning meets privacy: A survey and outlook," ACM Computing Surveys (CSUR), vol. 54, no. 2, Art. no. 31, pp. 1-36, 2021. [Online]. Available: https://doi.org/10.1145/3436755.

[29] D. Zhu, C. Cai, T. Yang, and X. Zhou, "A machine learning approach for air quality prediction: Model regularization and optimization," Big Data and Cognitive Computing, vol. 2, no. 1, p. 5, 2018. [Online]. Available: https://doi.org/10.3390/bdcc2010005.

[30] T. Kotsilieris, I. Anagnostopoulos, and I. E. Livieris, "Special issue: Regularization techniques for machine learning and their applications," Electronics, vol. 11, no. 4, p. 521, 2022. [Online]. Available: https://doi.org/10.3390/electronics11040521.

[31] S. Mazilu and J. Iria, "L1 vs. L2 regularization in text classification when learning from labeled features," Proceedings of the 10th International Conference on Machine Learning and Applications (ICMLA), 2011, pp. 361-366, doi: 10.1109/ICMLA.2011.85

[32] R. Muthukrishnan and R. Rohini, "LASSO: A feature selection technique in predictive modeling for machine learning," 2016 IEEE International Conference on Advances in Computer Applications (ICACA), Coimbatore, India, 2016, pp. 18-20, doi: 10.1109/ICACA.2016.7887916.

[33] I. S. Mambina, J. D. Ndibwile, and K. F. Michael, "Classifying Swahili Smishing Attacks for Mobile Money Users: A Machine-Learning Approach," IEEE Access, vol. 10, pp. 83061-83074, 2022, doi: 10.1109/ACCESS.2022.3196464

[34] A. K. Saxena, "Enhancing Data Anonymization: A Semantic K-Anonymity Framework with ML and NLP Integration," SSRAML SageScience, vol. 5, no. 1, pp. 81–92, 2022

[35] Michalsons, "Membership Inference Attacks: A New AI Security Risk," Michalsons, 2021. [Online]. Available: https://www.michalsons.com/blog/membership-inference-attacks-a-new-ai-security-risk/64440. [Accessed: Dec. 19, 2024

[35] GDPR, "General Data Protection Regulation. Art4. GDPR Definitions," available: https://gdpr-info.eu/art-4-gdpr/. Accessed: Dec. 19, 2024.

[36] S. L. Ribeiro and E. T. Nakamura, "Privacy protection with pseudonymization and anonymization in a health IoT system: Results from OCARIoT," 2019 IEEE 19th International Conference on Bioinformatics and Bioengineering (BIBE), Athens, Greece, 2019, pp. 904-908, doi: 10.1109/BIBE.2019.00169.