information

Article

Detection of Korean Phishing Messages Using Biased Discriminant Analysis under Extreme Class Imbalance Problem

Siyoon Kim [1], Jeongmin Park [1], Hyun Ahn [2] and Yonggeol Lee [2,*]

[1] Department of Information Science and Telecommunication, Hanshin University, Osan 18101, Republic of Korea; kimshiyoun1@hs.ac.kr (S.K.); wjdals93@hs.ac.kr (J.P.)
[2] School of Computing and Artificial Intelligence, Hanshin University, Osan 18101, Republic of Korea; hyunahn@hs.ac.kr
* Correspondence: pattern@hs.ac.kr; Tel.: +82-31-379-0656

Abstract: In South Korea, the rapid proliferation of smartphones has led to an uptick in messenger phishing attacks associated with electronic communication financial scams. In response to this, various phishing detection algorithms have been proposed. However, collecting messenger phishing data poses challenges due to concerns about its potential use in criminal activities. Consequently, a Korean phishing dataset can be composed of imbalanced data, where the number of general messages might outnumber the phishing ones. This class imbalance problem and data scarcity can lead to overfitting issues, making it difficult to achieve high performance. To solve this problem, this paper proposes a phishing messages classification method using Biased Discriminant Analysis without resorting to data augmentation techniques. In this paper, by optimizing the parameters for BDA, we achieved exceptionally high performances in the phishing messages classification experiment, with 95.45% for Recall and 96.85% for the BA metric. Moreover, when compared with other algorithms, the proposed method demonstrated robustness against overfitting due to the class imbalance problem and exhibited minimal performance disparity between training and testing datasets.

Keywords: machine learning; messenger phishing attack; biased discriminant analysis; phishing messages classification; class imbalanced problem

Citation: Kim, S.; Park, J.; Ahn, H.; Lee, Y. Detection of Korean Phishing Messages Using Biased Discriminant Analysis under Extreme Class Imbalance Problem. *Information* **2024**, *15*, 265. https://doi.org/10.3390/info15050265

Academic Editors: Barbara Pes and Andrea Loddo

Received: 12 April 2024
Revised: 3 May 2024
Accepted: 3 May 2024
Published: 7 May 2024

1. Introduction

Financial fraud criminals access their victims through mobile devices, such as smartphones, which are widely used by many people [1,2]. Specifically, scams that deceive victims and exploit them for personal gain through messages or messenger conversations are commonly referred to as messenger phishing or messenger phishing attacks [3]. Recent observations indicate a substantial rise in global messenger usage, from 2.56 billion users in 2019 to 2.91 billion users in 2020, with a projected increase to approximately 3.3 billion users by 2023 [4]. Consequently, the prevalence of phishing attacks through Social Network Services (SNS) has exponentially escalated. In the context of South Korea, which boasts the highest smartphone penetration rate, damages caused by messenger phishing reached 57.64 billion KRW (12,402 cases) in 2020, reflecting an increase of approximately 201.6% compared to the previous year. As the global adoption rate of smartphones, which serve as a medium for messenger phishing crimes, continues to increase, the incidence and impact of phishing attacks are expected to grow persistently [5,6].

Messenger phishing criminals utilize phishing messages to target their victims. Phishing messages can be defined as web links, promotional messages, or unrelated text messages that are regularly sent to a large number of recipients for advertising purposes [7]. Phishing messages can be sent indiscriminately to a broad audience based on predefined templates, requiring minimal effort in comparison to voice phishing crimes, thus making them actively exploited in criminal activities [1]. Proactive classification of phishing messages by telecommunications providers can serve as an effective preventive measure against

phishing attempts. However, this approach may raise concerns regarding privacy invasion and the potential for creating a 'Big Brother' problem. Therefore, a practical alternative lies in post-delivery phishing messages classification methods implemented at the recipient's end, such as on mobile devices, as a means of filtering phishing content.

The detection of phishing messages has been attempted with various algorithms by classifying phishing and non-phishing messages in text datasets. Traditional detection algorithms include Naive Bayes (NB), Support Vector Machine (SVM), and Logistic Regression (LR) [8–11]. Moreover, tree-based methods are represented by Decision Tree (DT) and Random Forest (RF), while boosting methods include Adaptive Boosting (AdaBoost), Extreme Gradient Boosting (XGBoost), and Light Gradient Boosting Model (LGBM) [12–17]. Additionally, detection techniques using neural networks involve Stochastic Gradient Descent (SGD), Artificial Neural Network (ANN), Convolutional Neural Network (CNN), and Long Short-Term Memory (LSTM), with the inclusion of Bidirectional Long Short-Term Memory (BiLSTM) to learn bidirectional long-term dependencies by adding backward LSTM [18–21]. Meanwhile, deep learning neural networks have been combined to design new architectures to enhance detection performance. Deep learning methods can learn practical features representing data for better classification through temporal and spatial data correlations across all domains [22]. Consequently, typical forms include combinations of spatial correlation-utilizing CNNs and temporal correlation-utilizing neural networks, such as Convolutional Neural Network + Long Short-Term Memory (CNN+LSTM) and Convolutional Neural Network + Gated Recurrent Unit (CNN+GRU) [23,24].

While these methods demonstrate relatively high classification performance, there are still several limitations that remain. First, there are morphological challenges in language processing. Agglutinative languages such as Korean, Japanese, Chinese, German, Russian, and Spanish exhibit diverse ways of expressing messages, and similar words can collide with each other, resulting in lower performance in morphological analysis [25–27]. Particularly, Korean, as an agglutinative language, combines nouns and verbs with particles and suffixes, leading to a significant increase in the number of derived word units, which in turn drastically increases the number of features [28]. Therefore, alternative approaches are required when dealing with agglutinative languages. Secondly, in the training process for phishing messages classification, non-phishing messages are generally much more abundant than phishing messages. Unless it involves legal authorities, collecting phishing messages is highly restricted. Consequently, there is a imbalanced data problem where the non-target class (non-phishing) that is not the focus of classification has a large number of samples, while the target class (phishing) has a significantly smaller number of collected samples [29,30]. The imbalanced data problem becomes more severe when the class of interest is relatively rare and has a small number of samples compared to the non-target class [31]. In machine learning modeling, when the size of the non-target class greatly outweighs the target class, biased learning outcomes towards the non-target class can occur, ultimately leading to an inability to effectively address various target classes that exist in real-world scenarios [30,32]. Moreover, the cost of misclassifying the target class is much higher than the cost of misclassifying the non-target class.

In this paper, we propose a method for classifying phishing messages among messages written in Korean. In the data collection phase, we assume an extreme class imbalance problem and collect data in such a way that the dataset size of the non-phishing class is more than 70 times larger than the phishing class. From the text-based phishing data, we use KoNLPy's MeCab, a Korean morphological analyzer, to extract lemma keywords targeting verbs and nouns through lemmatization, which are then used as features. Based on the extracted features, we define the data structure by creating a Bag of Words (BoW) for the entire dataset, including phishing and non-phishing. To address the class imbalance problem, we employ Biased Discriminant Analysis (BDA) [33]. The primary focus of biased learning is to distinguish a specific class of interest (e.g., phishing) from other classes (e.g., non-phishing). BDA is designed to resolve the asymmetry between these target and non-target classes, and it is utilized to enhance the robustness, especially when dealing

with small training samples [33]. Additionally, in this process, the optimal parameters are selected to resolve the asymmetry between classes. Experimental results show high classification performance in the class imbalance problem, with Recall and Balanced Accuracy (BA) reaching 95.45% and 96.85%, respectively.

This paper is structured as follows. In Section 3, we construct a BDA feature space for classification and propose a data classification method. In Section 4, we analyze the proposed method by selecting optimal parameters to address the class imbalance problem and evaluating performance through comparison experiments with various models. In Section 5, we conclude the paper.

2. Previous Work

A range of machine learning algorithms have been applied to spam detection [8–16,16–18,18–21,23,24,34]. These include traditional methods such as SGD, SVM, and NB, alongside deep learning-based techniques like ANN, CNN, LSTM, BiLSTM, and GRU. In addition, gradient boosting methods, including AdaBoost, XGBoost, and LGBM, have also been used. Moreover, non-parametric supervised learning algorithms such as DT and RF and the clustering method *k*-means have been utilized. The target languages for spam detection research include English [8–10,12–15,17,19–21,23,34], Indonesian [18], Bengali [16], Arabic [24], and Turkish [11]. These studies converted text data into embedding vectors using techniques such as TF-IDF, BoW, Bidirectional Encoder Representations from Transformers (BERT), and Continuous Bag of Words (CBoW), making them suitable for training. Subsequently, methods that incorporate Word2Vec into existing approaches have also emerged. Among the diverse performance evaluation metrics used in these studies, Recall was used as the benchmark for comparison.

2.1. Spam Detection in Balanced Dataset

Ref. [20] experimented with balanced and imbalanced class datasets. They compared traditional machine learning techniques with deep learning methods for phishing messages detection. Traditional machine learning techniques included SVM, NB, DT, LR, RF, and AdaBoost, while deep learning methods employed ANN and CNN. The performance experiments compared results in imbalanced (4827 phishing and 747 non-phishing messages) and balanced datasets (1000 spam and 1000 non-spam messages). The experimental results showed that CNN performed best in both datasets, with results of 96.4% and 97.5%, respectively. The detection results in imbalanced datasets appeared relatively lower than those in balanced datasets. Consequently, spam detection performance in imbalanced datasets proved to be relatively lower than in balanced datasets.

Ref. [18] generated embedding vectors using the TF-IDF method and detected phishing classes through SGD. The dataset consisted of 1143 entries, with 574 phishing and 569 non-phishing instances. The phishing detection performance using SGD was indicated to be 97.2%. Ref. [16] enhanced the performance of a spam messages detection model by employing XGBoost. The total dataset consisted of 550 entries, using data collected directly, with the spam and non-spam datasets comprising 300 and 250 instances, respectively. In their experiments, XGBoost showed the highest result at 82.6%.

However, phishing messages detection often involves class imbalance issues, making the application of these studies to real-world settings challenging.

2.2. Spam Detection in Imbalanced Dataset

In the majority of the previous studies, various machine learning algorithms were employed to improve the performance of phishing messages detection in imbalanced datasets [8–10,12–15,17,21,23,34].

2.2.1. Traditional Methods

In [10], methods such as NB, SVM, and Maximum Entropy classifier were employed to perform smishing classification. The dataset consisted of 4827 non-spam messages and

747 spam messages. The experimental results showed classification accuracies of 90.9%, 96.4%, and 85.9%, respectively. Similarly, ref. [9] conducted a comparative analysis of various machine learning algorithms to find a suitable spam classification model for biased datasets. Five machine learning classifiers, namely *k*NN, Linear Support Vector Machine, RBF Support Vector Machine, RF, and DT, were applied to classify spam SMS messages. The dataset comprised 4827 non-spam messages and 747 spam messages. The experimental results indicated that Linear Support Vector Machine achieved the highest accuracy of 92.3% on the imbalanced dataset based on Hashing.

In [34], machine learning classifiers such as NB, SVM, LR, *k*-Nearest neighbor (*k*NN), DT, and AdaBoost, as well as hybrid models like k-means+NB, k-means+SVM, and k-means+LR, were used to classify spam messages. This study combined the unsupervised learning-based k-means algorithm for clustering with classification models to enhance performance. The dataset comprised 4825 non-spam messages and 747 spam messages. The experimental results showed that k-means+SVM achieved the highest classification accuracy of 92%. Ref. [8] proposed an SMS spam detection and classification model using the NB machine learning method. The dataset contains 747 spam messages and 4778 non-spam messages. The NB classification methodology achieved a performance of 97.3% on this dataset.

2.2.2. Deep Learning-Based Methods

In [21], the BiLSTM model was employed for phishing detection. The training dataset consisted of 6792 non-spam messages and 3200 spam messages. Using Word to Vector (Word2Vec) as the embedding model, the proposed method achieved a phishing detection performance of 91.7%. Furthermore, ref. [23] introduced a phishing-detection approach based on a hybrid model that combines CNN and GRU within a hybrid framework. The dataset consists of 5572 text messages, including 747 phishing messages and 4825 non-phishing messages. When compared to CNN, Gated Recurrent Unit (GRU), Multi-Layer Perceptron (MLP), SVM, and XGBoost, the proposed hybrid model exhibited the highest performance at 96.5%.

2.2.3. Gradient Boosting Methods

In [17], four rank correlation algorithms, namely Pearson, Spearman's, Kendall rank, and Point biserial, were used to determine the most suitable feature set for phishing SMS detection. The dataset consisted of 4831 non-phishing messages and 747 phishing messages. For performance evaluation, classifiers including RF, DT classifier, AdaBoost classifier, and SVM were compared, and AdaBoost Classifier achieved the highest accuracy of 98.7%. The phishing SMS detection performance results using Pearson, Spearman's, Kendall rank, and Point biserial were 90.2%, 91.0%, 91.4%, and 90.2%, respectively. Consequently, the Kendall rank correlation algorithm showed the highest accuracy at 91.4%. In [15], spam SMS messages were detected using XGBoost, LGBM, and Bernoulli Naive Bayes. The dataset consisted of 5574 text messages, including 747 spam messages and 4827 non-spam messages. To address the class imbalance issue, down sampling was employed to equalize the number of spam and non-spam messages. LGBM exhibited a classification delay of 1.703 s and achieved high performance with an accuracy of 95.6%.

2.2.4. Non-Parametric Supervised Learning Methods

In [12], the performance of phishing detection was compared using machine learning algorithms *k*NN, and DT. The dataset consisted of 747 phishing data and 4827 non-phishing data. Among the three algorithms, DT-based phishing detection showed the highest performance at 93.1%. Ref. [13] analyzed better vectorization methods for feature extraction to detect phishing via SMS. They applied vectorization methods such as Bag of Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF), and Word2Vec to preprocessed data. The dataset contained 638 phishing messages and 5333 non-phishing messages. Performance evaluation was conducted using RF, LR, and Gaussian Naïve Bayes classifiers.

The experimental results showed that the combination of TF-IDF vectorization and RF Classifier achieved the highest classification performance at 85.0%. Ref. [14] compared and analyzed machine learning classification algorithms for detecting spam SMS. The dataset contained 429 spam messages and 2179 non-spam messages. The algorithms used for analysis were NB, LR, DT, and RF, with RF showing the highest performance at 96.5%.

Most languages, excluding English, commonly used for phishing detection, present challenges in direct phishing messages dataset collection. Difficulty in data collection can lead to the formation of extremely imbalanced datasets, potentially resulting in algorithmic overfitting issues.

2.3. Spam Detection in Extremely Imbalanced Datasets

Several studies have performed spam detection experiments in environments with extreme imbalance problems [11,19,24]. In this paper, an extremely imbalanced dataset is defined as one where non-phishing messages outnumber phishing messages by a ratio of more than 10 to 1.

In [11], phishing messages written in Turkish were converted into embedding vectors from BoW and TF-IDF, and phishing detection was conducted using machine learning algorithms RF, LR, AdaBoost, and SVM. The dataset consisted of 119 phishing messages and 3526 non-phishing messages. In the TF-IDF-based dataset, RF and LR had the highest performance with 92.5%. Meanwhile, in the frequency-based dataset, RF, LR, and SVM delivered a performance of 90.0%. In [19], an efficient smishing detection system was developed using an Artificial Neural Network (ANN). The dataset comprised 5858 text messages, including 538 phishing and 5320 non-phishing messages. The detection performance of smishing using ANN was reported as 92.4%. In [24], a hybrid model combining CNN and LSTM was employed for classifying phishing messages in Arabic. This model achieved a notable classification accuracy of 87.9% on a dataset that included 7579 non-phishing and 785 phishing messages (Table 1).

Table 1. Overview of previous studies on the classification performance of phishing messages.

Data Balance	Algorithm	Embedding	Year	Language	Dataset		Recall
					# Pos	# Neg	
Balanced	CNN [20]	TF-IDF	2018	English	1000	1000	97.5%
	SGD [18]	TF-IDF	2021	Indonesian	574	569	97.2%
	XGBoost [16]	TF-IDF	2023	Bengali	300	250	82.6%
Imbalanced	RF [14]	BoW	2017	English	429	2179	96.5%
	CNN [20]	TF-IDF	2018	English	747	4827	96.4%
	SVM [10]	BoW	2018	English	747	4827	96.4%
	AdaBoost+Kendall [17]	BoW	2020	English	747	4831	91.4%
	k-means+SVM [34]	TF-IDF	2020	English	747	4825	92.0%
	LGBM [15]	BoW, TF-IDF	2020	English	747	4827	96.5%
	CNN+GRU [23]	BERT	2021	English	747	4825	96.5%
	NB [8]	BoW	2021	English	747	4778	97.3%
	BiLSTM [21]	CBoW+Word2Vec	2022	English	3200	6792	91.7%
	SVM [9]	TF-IDF	2022	English	747	4827	92.3%
	DT [12]	Word Embedding	2023	English	747	4827	93.1%
	RF [13]	TF-IDF+Word2Vec	2023	English	638	5333	85.0%

Table 1. *Cont.*

Data Balance	Algorithm	Embedding	Year	Language	Dataset		Recall
					# Pos	# Neg	
Extremely Imbalanced	CNN+LSTM [24]	Word Embedding	2020	Arabic	785	7579	87.9%
	RF, LR [11]	TF-IDF, BoW	2021	Turkish	119	3526	92.5%
	ANN [19]	TF-IDF	2022	English	538	5320	92.4%

3. Proposed Method

In this paper, we propose a method for classifying phishing messages in a Korean dataset with a class imbalance problem. The proposed method consists of three stages: data conversion, feature engineering, and decision. In the data conversion stage, phishing messages ($\mathcal{M}_s$) and non-phishing messages ($\mathcal{M}_{ns}$) were assigned as the target class and the non-target class, respectively. Additionally, we extracted keywords of verbs and nouns from the collected phishing messages using a morphological analyzer. Then, we created a numerical BoW composed of the frequency of each keyword. In the parameter estimation stage, we generated the BDA feature space ($\mathbf{W}_{BDA}$) from the training dataset, setting the optimal parameters. These included the regularization parameter, the number of BDA feature vectors, and the threshold needed to construct the space. Finally, in the decision stage, we measured the distance between the projected test data and the mean vector of the training data that belongs to the phishing class within the BDA feature space. Based on this distance and a specified threshold, we classified the messages as either phishing or non-phishing. The overall procedure of the proposed method is shown in Figure 1.

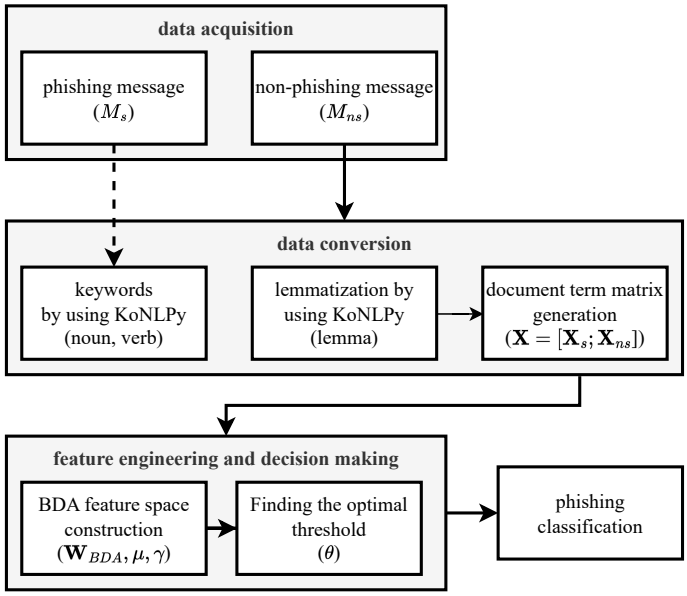


Figure 1. Overall procedure of the proposed method.

To classify phishing messages written in Korean, the following considerations need to be taken into account.

- **Data Conversion**: Typically, messages are in the form of text, and they need to be converted into a format that can be understood by machines.
- **Curse of Dimensionality**: Like all languages, using all morphemes can lead to excessively high dimensionality in the data.

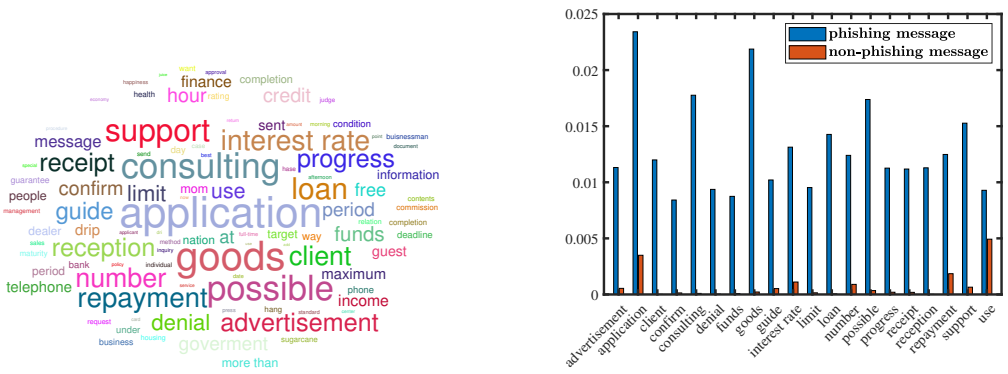
- **Morphology**: Korean, an agglutinative language, combines nouns and verbs with particles, suffixes, and endings, resulting in a large number of derived word units and a significant increase in the number of features.
- **Intention of Writing**: Since phishing messages are written with similar intentions, the text often includes a multitude of similar keywords.
- **Class Imbalance Problem**: The number of phishing messages is extremely small compared to non-phishing messages. Similar to previous studies [10,11,24,34], we assume the class imbalance problem.

3.1. Data Conversion

A message, which includes letters and symbols in text format, needs to be converted into a numerical format so that it can be understood by machines. Generally, text data is converted and used in the form of a BoW or through TF-IDF conversion [10,11,24,34]. To generate the BoW ($\mathbf{X} \in \mathbb{R}^{d \times N}$), we define the d extracted keywords from messages written in Korean through morphological analysis as features and set the frequency of each keyword as an attribute.

Generally, texts are composed of various parts of speech, so if all the morphemes included in the collected data are used, the number of features (d) could become excessively large. Moreover, when analyzing messages written in Korean, an agglutinative language, consideration of the language's morphological elements is necessary. Since nouns and verbs in Korean often combine with particles and endings, the number of derived word units increases significantly, thereby also drastically increasing the number of features [28]. To address this issue, only verbs and nouns were targeted for morpheme extraction during the feature creation phase, and all words were converted to their lemmas through lemmatization. Ultimately, this approach leads to a reduction in data dimensionality.

In the feature selection stage, when non-phishing messages are defined as "everyday conversations" the freedom of text data increases, resulting in a larger number of features. Fortunately, phishing messages exhibit a characteristic of being written with similar content and randomly sent to others, regardless of the author. Therefore, when focusing solely on phishing messages, the frequency of specific words can appear high. Figure 2 presents the extraction of major keywords with high proportions in phishing messages and the frequency of major keywords appearing in $\mathcal{M}_{ns}$. Figure 2a visualizes the distribution of major keywords (verbs or nouns) that appear in both $\mathcal{M}_s$ and $\mathcal{M}_{ns}$ using a word cloud. Overall, keywords such as "application", "goods", "consulting", and "repayment" dominate a significant portion. Figure 2b shows a histogram of the top 20 keywords with the highest frequency in $\mathcal{M}_s$. For the same keywords, the frequency in $\mathcal{M}_{ns}$ differs significantly from that in $\mathcal{M}_s$. Specifically, despite the approximately 70-fold volume difference between the non-phishing and phishing classes, there is a distinct difference in the frequency of specific keywords between the two. Therefore, we construct the features of the BoW using only the features extracted from $\mathcal{M}_s$.



(**a**) Word cloud of phishing messages (**b**) TF in phishing messages

Figure 2. Visualization of common words in phishing or non-phishing messages.

3.2. Feature Engineering and Decision Making

In phishing messages classification, there is a significant class imbalance problem where the target class, which consists of $\mathcal{M}_s$, has a considerably lower number of instances compared to the non-target class of $\mathcal{M}_{ns}$. Fisher's Discriminant Analysis (FDA) [35] is one of the widely used methods in classification problems. FDA aims to create a feature space that maximizes the separability between classes. However, when a specific class has a significantly larger number of instances, it becomes challenging to reflect the distribution of data from the target class. As a result, FDA generally exhibits poor performance in class imbalance problems [36].

In this paper, BDA, a generalization of FDA, is employed to address the class imbalance problem. BDA can effectively handle data from both the non-target class and the target class, which exhibit asymmetric nonlinear densities [33]. In other words, it focuses on enhancing the robustness of the distribution represented by the narrow target class data, avoiding bias towards the non-target class that contains a larger amount of data. Additionally, BDA aims to find a linear transformation that minimizes the distribution of phishing data while maximizing the separation between non-phishing data and phishing data.

The data matrix $\mathbf{X} = [\mathbf{X}_s, \mathbf{X}_{ns}] \in \mathbb{R}^{d \times (N_s + N_{ns})}$ is composed of phishing datasets $\mathbf{X}_s = \left[\mathbf{x}_s^0, \cdots, \mathbf{x}_s^{N_s}\right]$ and non-phishing datasets $\mathbf{X}_{ns} = \left[\mathbf{x}_{ns}^0, \cdots, \mathbf{x}_{ns}^{N_{ns}}\right]$. Accordingly, the BDA objective function $\mathbf{W}_{BDA}$ is defined as follows .

$$\mathbf{W}_{BDA} = \arg\max_{\mathbf{W}} \frac{|\mathbf{W}^T \mathbf{C}_{ns} \mathbf{W}|}{|\mathbf{W}^T \mathbf{C}_s \mathbf{W}|} \tag{1}$$

In Equation (1), the matrices $\mathbf{C}_s$ and $\mathbf{C}_{ns}$ represent the scatter matrices of the phishing data and non-phishing data, respectively, and can be defined as follows.

$$\begin{aligned} \mathbf{C}_s &= \sum_{i=1}^{N_s} (\mathbf{x}_s^i - \mathbf{m}_s)(\mathbf{x}_s^i - \mathbf{m}_s)^T \\ \mathbf{C}_{ns} &= \sum_{i=1}^{N_{ns}} (\mathbf{x}_{ns}^i - \mathbf{m}_s)(\mathbf{x}_{ns}^i - \mathbf{m}_s)^T \end{aligned} \tag{2}$$

where $\mathbf{m}_s \left(= \sum_{i=1}^{N_s} \mathbf{x}_s^i\right)$ is the mean of all the samples belonging to the phishing class, $\mathbf{W}_{BDA}$ aims to find the optimal transformation that maximizes the variance of $\mathbf{W}^T \mathbf{C}_{ns} \mathbf{W}$ and minimizes the variance of $\mathbf{W}^T \mathbf{C}_s \mathbf{W}$, resulting in the maximum ratio. As a result, BDA extracts features that densely represent $\mathcal{M}_s$ close to $\mathbf{m}_s$ and at the same time, separates $\mathcal{M}_{ns}$ far from $\mathbf{m}_s$, according to the objective function. Additionally, the effective dimensions of the BDA feature space, denoted as γ, provide a higher information density capacity than FDA, which has only one effective dimension, with $\gamma = \min(N_s, N_{ns})$ [33]. In the context of phishing messages classification, where $N_{ns} \gg N_s$, the effective dimensions correspond to $\mathbf{C}_s$. The column vectors of $\mathbf{W} = \left[\mathbf{w}_1, .., \mathbf{w}'_{N_s}\right]$ are the generalized eigenvectors associated with the generalized eigenvalues, satisfying

$$\mathbf{C}_{ns} \mathbf{w}_t = \lambda \mathbf{C}_s \mathbf{w}_t \tag{3}$$

where $t = 1, \cdots, N_s - 1$. They can be obtained by the simultaneous diagonalization of $\mathbf{C}_{ns}$ and $\mathbf{C}_s$ if $\mathbf{C}_s$ is nonsingular. However, N_s is significantly lower than d in phishing messages classification models, $\mathbf{C}_s$ becomes singular, leading to the Small Sample Size Problem (SSSP) [37]. To address this problem, Principal Component Analysis (PCA) [38] can be employed. PCA generates $N_s + N_{ns} - 1$ feature dimensions that maximize the variance of the data based on the covariance of $\mathbf{X}$. By selecting only N'_s or fewer eigenvectors with

the largest eigenvalues, the SSSP problem can be resolved. Consequently, Equation (1) is redefined as follows.

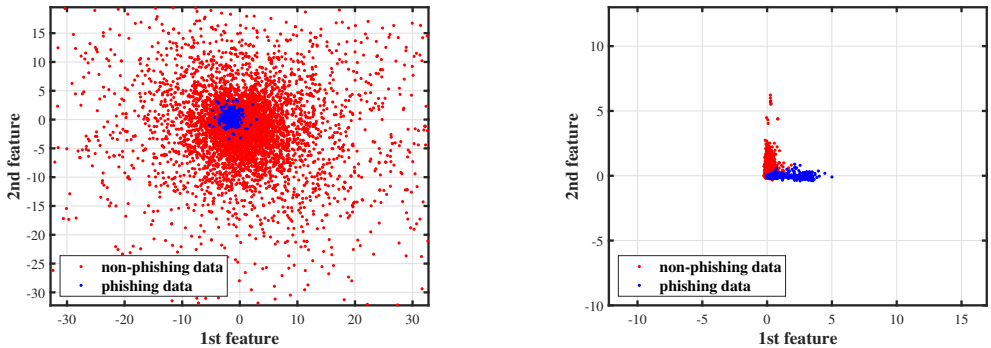
$$\mathbf{W}_{BDA} = \arg\max_{\mathbf{W}} \frac{|\mathbf{W}^T \mathbf{W}_{PCA}^T \mathbf{C}_{ns} \mathbf{W}_{PCA} \mathbf{W}|}{|\mathbf{W}^T \mathbf{W}_{PCA}^T \mathbf{C}_s \mathbf{W}_{PCA} \mathbf{W}|} = \arg\max_{\mathbf{W}} \frac{|\mathbf{W}^T \tilde{\mathbf{C}}_{ns} \mathbf{W}|}{|\mathbf{W}^T \tilde{\mathbf{C}}_s \mathbf{W}|} \tag{4}$$

$\mathbf{C}_s$ becomes a full-rank nonsingular matrix, and as a result, diagonalization can be performed. We applied whitening to ensure that $\mathbf{W}^T \tilde{\mathbf{C}}_s \mathbf{W}$ in Equation (4) becomes the identity matrix ($= \mathbf{I}$). Consequently, we select γ eigenvectors of $\tilde{\mathbf{C}}_{ns}$ that maximize $\mathbf{W}^T \tilde{\mathbf{C}}_{ns} \mathbf{W}$ when $\mathbf{W}^T \tilde{\mathbf{C}}_s \mathbf{W} = \mathbf{I}$.

Nevertheless, the class imbalance problem still remains unresolved. Unfortunately, reflecting the distribution of phishing data in the phishing messages classification model is challenging due to the very limited number of collectible $\mathcal{M}_s$. Regularization is one method that can augment the distribution of a class with constrained data. In this paper, we addressed the class imbalance problem by balancing the asymmetrical scales between classes, achieved by increasing the variance of $\mathbf{C}_s$ through the addition of a small value μ.

$$\mathbf{C}_s^r = (1 - \mu)\mathbf{C}_s + \frac{\mu}{N_s'} trace[\mathbf{C}_s]\mathbf{I} \tag{5}$$

In Equation (5), μ is the parameter controlling the variance of $\mathbf{C}_s$. μ serves to solve the asymmetry between the target class and non-target class. Generally, classification models are biased towards the distribution of the non-target class, relative to the target class, which holds a small amount of data. To address this problem, regularization simply extends the scope of the target class by adding a small value to the diagonal elements (variance) of the covariance matrix. When the value of μ is 0, there is no change in the distribution of the target class. As the value of μ increases, the variance of the target class gradually expands, ultimately leading to a more robust distribution of the target class. If $\mu = 1$ to the extreme, the target class loses the characteristics of the distribution it has in the BDA feature space. Figure 3 shows the results comparing non-regularized BDA and regularized BDA. In Figure 3a, the conventional BDA without regularization recognizes the distribution of the phishing class (target class) as part of the non-phishing class (non-target class). On the other hand, in Figure 3b, it can be seen that the asymmetrical structure of a small-scale target class and a large-scale non-target class has been mitigated when $\mu = 0.6$.



(**a**)Non-regularized BDA (**b**)Regularized BDA(μ=0.6)

Figure 3. Distribution difference of scatter matrix due to regularization.

The final phase of phishing messages classification is determining whether the query data ($\mathbf{x}_q$), converted into a vector, is phishing or not. $\mathbf{x}_q$ is projected onto the BDA feature space, and then the Euclidean distance is measured between the projected data $\mathbf{W}_{BDA}^T \mathbf{x}_q$ and the mean vector of phishing data $\mathbf{W}_{BDA}^T \mathbf{m}_s$ from the training data. The phishing status is determined based on the calculated distance.

$$dec = \begin{cases} \text{phishing}, & \left\|\mathbf{W}_{BDA}^{T}(\mathbf{x}_q - \mathbf{m}_s)\right\|_2^2 < \theta \\ \text{non-phishing}, & \text{otherwise} \end{cases} \tag{6}$$

In Equation (6), based on the distance threshold θ, if $\left|\mathbf{W}_{BDA}^{T}\mathbf{x}_q - \mathbf{W}_{BDA}^{T}\mathbf{m}_s\right|_2^2$ is smaller than θ, the model classifies it as phishing; otherwise, it is classified as non-phishing (Figure 4).

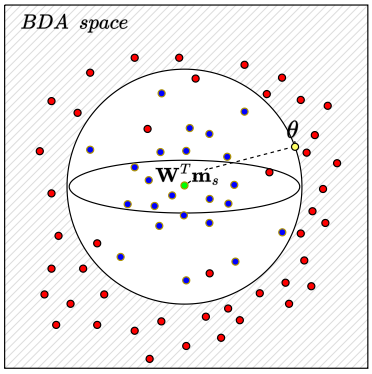


Figure 4. Distribution of ideal data in the proposed method (red circle: phishing; blue circle: non-phishing).

4. Experimental Results

We configured the experimental environment with an NVIDIA GeForce RTX 4080 GPU, Intel(R) Core(TM) i5-10210U CPU @ 1.60GHz, AMD Ryzen 9 5950X 16-Core CPU, and 32 GB DDR4 RAM. For data conversion, we utilized Python version 3.7.6 and Java version 1.8.0. Additionally, we employed KoNLPy version 0.5.2 for morphological analysis of the Korean language. Specifically, we selected MeCab from various morphological analyzers available in KoNLPy, such as HanNanum, Kkma, KOMORAN, and OKT, based on both computational speed and performance.

In the experiment, we directly collected phishing messages from the web and gathered non-phishing messages from open datasets featuring everyday conversation patterns. We explored combinations of optimal parameters that exhibit high classification performance in the BDA feature space. Finally, in order to objectively evaluate the performance of the proposed method, we conducted comparative experiments with multiple machine learning-based algorithms using Distribution-Optimally-Balanced Stratified Cross-Validation (DOB-SCV) [39]. Regarding performance metrics for classification, Recall and BA were employed.

$$\begin{aligned} Recall &= \frac{TP}{TP + FN} \\ BA &= \frac{1}{2}\left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP}\right) \end{aligned} \tag{7}$$

In the class imbalance problem, there is a challenge where the cost of misclassifying the target class becomes significantly higher compared to misclassifying the non-target class [31]. As a result, even when focusing solely on the non-phishing class without considering the phishing class, the accuracy performance can approach nearly 100%. On the other hand, Recall emphasizes how effectively phishing messages are correctly identified as phishing. Therefore, as the classification performance improves, the Recall results also increase accordingly. Additionally, BA, which represents the average accuracy obtained from both classes [40], provides a comprehensive evaluation of correctly identifying actual phishing data as phishing and actual non-phishing data as non-phishing. Consequently, if an algorithm demonstrates high results in both metrics, it can be interpreted as having good phishing message classification performance.

4.1. Dataset

We collected 615 phishing messages and 42,594 non-phishing messages for phishing messages classification. Phishing messages associated with actual messenger phishing crimes, which involve personal information and social issues, are not disclosed due to privacy and societal concerns. In this paper, we collected a total of 615 images directly uploaded by messenger phishing victims from 2013 to 2021, converted them into text, and used them as a phishing dataset. The collected phishing dataset includes various types of crimes, such as cryptocurrency scams, advertising scams, loan scams, impersonation of public institutions, and impersonation of acquaintances. Figure 5 presents examples of the directly collected dataset. Figure 5a illustrates an example related to cryptocurrency scams, where criminals masquerade as cryptocurrency exchanges and send scam messages claiming that the victims' assets are at risk. Figure 5b showcases an example of an advertising scam, where phishing attempts are made through advertisement-like messages, such as job postings or delivery notifications. Furthermore, Figure 5c depicts a form of loan scam where criminals impersonate loan providers and deceive victims in need of money. Figure 5d demonstrates a type of fraud where criminals impersonate public institutions and induce victims to click on specific URLs, often involving scams related to COVID-19 relief funds. Lastly, Figure 5e presents an example of impersonating an acquaintance, where criminals pose as the victim's acquaintances and exploit them for financial gain.

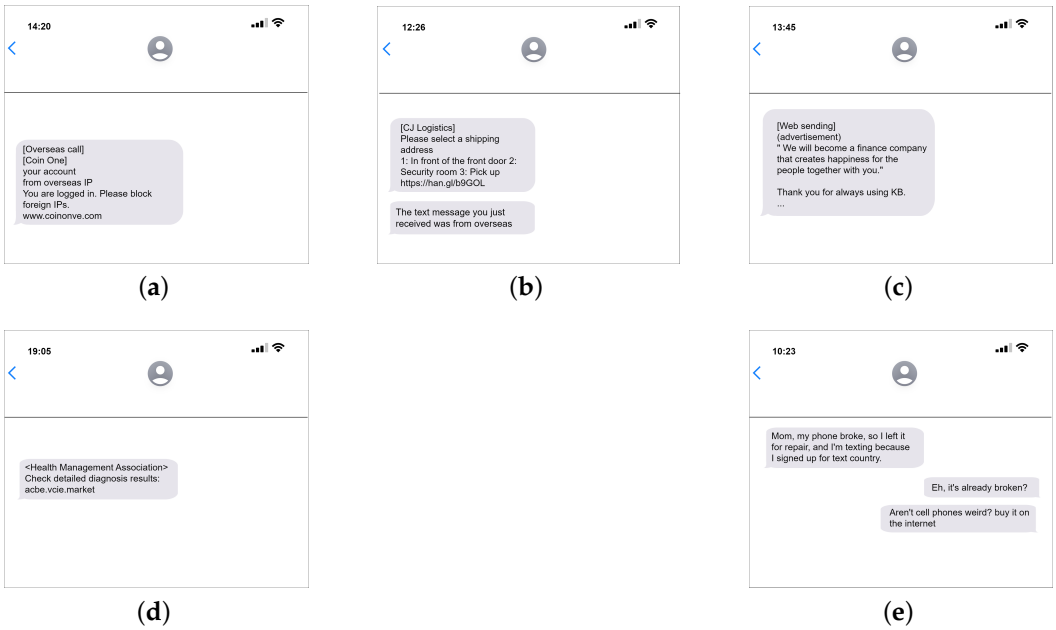


Figure 5. Examples of phishing message types collected directly. (**a**) Cryptocurrency scam. (**b**) Advertising scam. (**c**) Loan scam. (**d**) Impersonation of public institution. (**e**) Impersonation of an acquaintance.

Non-phishing messages include a Twitter conversation-based dataset [41] with everyday conversational patterns, a one-shot conversation dataset [42], and a chatbot conversation dataset (2021) [43]. The Twitter conversation-based dataset includes everyday conversations between two or more speakers, with 2000 messages ranging from a minimum of 1 to a maximum of 17 turns. The one-shot conversation dataset is comprised of 38,594 messages through web crawling of SNS posts and online comments. Lastly, the chatbot dataset is divided into three classes: general conversations, farewells, and love-related conversations. From the 3040 conversation data in the 'general conversation' class, we selected 2000 messages for the experiments, excluding duplicates.

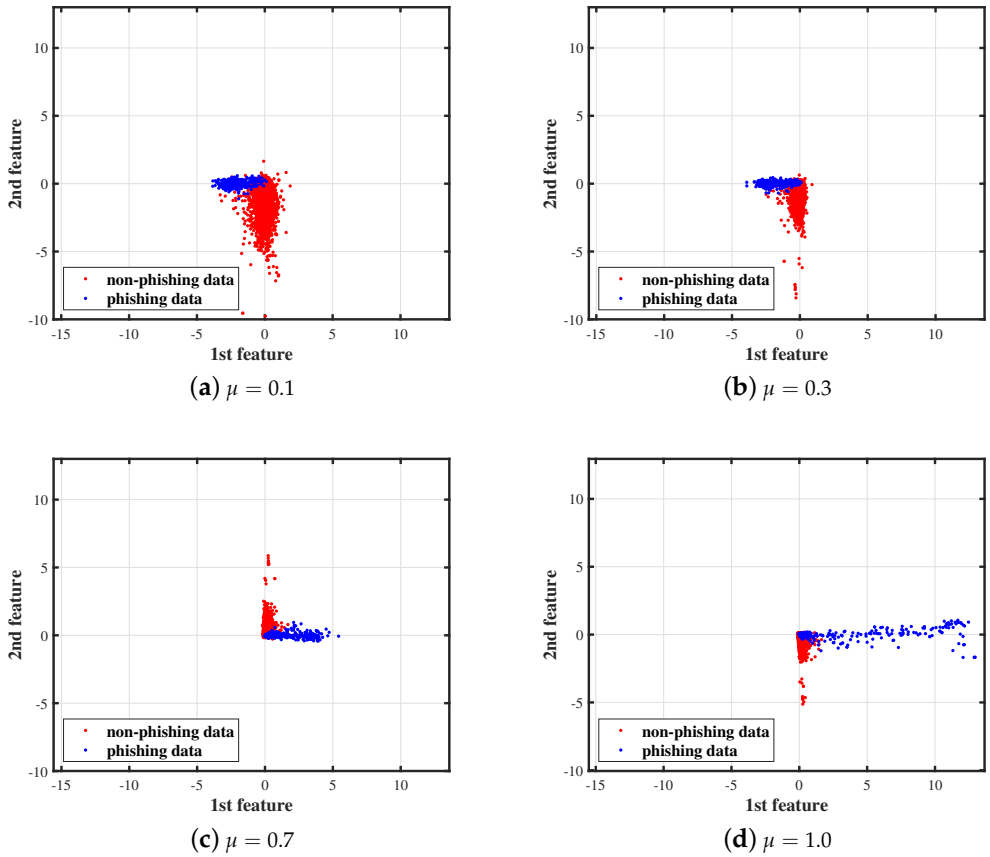
To transform the collected messages into a machine-understandable format, we created a BoW by tallying the frequencies of words extracted through morphological analysis. Since BoW can exponentially increase data dimensionality by including all words as features, we removed symbols, numbers, and words with fewer than two characters irrelevant to

messenger phishing crimes during the preprocessing stage. Additionally, we controlled data dimensionality by employing lemmatization to use the base form of all morphemes, extracting 1533 keywords from the dataset.

4.2. Parameters Estimation

By evaluating the performance in terms of Recall and BA for various combinations of parameters on the training dataset, the optimal parameters μ, γ, and θ for phishing messages classification can be estimated. First, selecting an appropriate μ addresses the issue of overfitting in machine learning modeling while preserving the characteristics related to the distribution of the phishing class. Increasing the value of μ leads to an increase in the variance of the relatively small amount of phishing class, ultimately making the distribution of the phishing class robust. Figure 6 illustrates the distribution of data in each BDA feature space according to different μ values. In Figure 6a–c, we can observe the improvement in the asymmetric structure between the phishing and non-phishing classes as μ increases. Particularly in Figure 6d, when μ approaches its maximum value of 1.0, the distribution of the phishing class expands beyond the non-phishing class, losing its distinctive characteristics. In [44], optimal values of μ between 0.1 and 0.2 were chosen for datasets that did not exhibit a relatively symmetric structure in class imbalance problem. However, in this paper, we are dealing with an extreme class imbalance problem, so we set μ to be at least 0.3.

Secondly, it is necessary to consider the number of BDA feature vectors, denoted as γ. Generally, eigenvalues close to 0 correspond to noise and should be excluded from selection. On the other hand, selecting eigenvectors corresponding to higher eigenvalues ensures higher classification performance. In this paper, to address the SSSP problem, we set $N_s' = 881$ in the PCA step and choose the optimal γ among the maximum of 881 feature vectors that exhibit the best performance. Lastly, given μ and γ, we determine the threshold θ to classify phishing and non-phishing within the BDA feature space. θ is set to the value in the BDA feature space that achieves the highest classification performance.

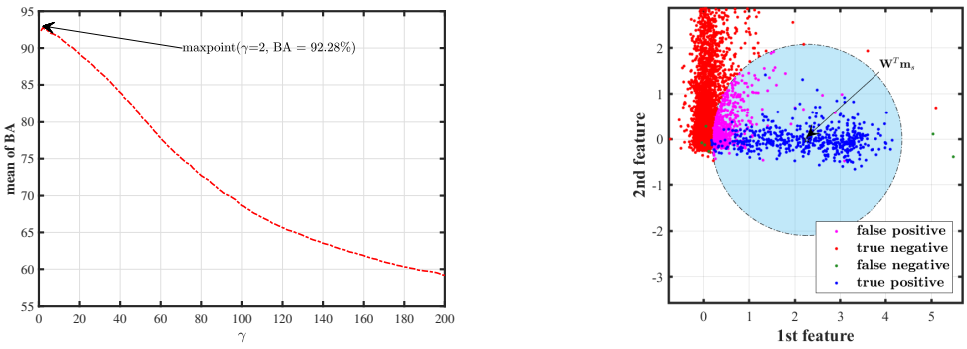


(**a**) $\mu = 0.1$ (**b**) $\mu = 0.3$

(**c**) $\mu = 0.7$ (**d**) $\mu = 1.0$

Figure 6. Change in the distribution of the scatter matrix by μ.

The optimal parameter combination (μ, γ, θ) for phishing messages classification was set to $(0.65, 2, 2.4130)$. Figure 7 represents the results of estimating the optimal parameters. Figure 7a illustrates the difference in classification performance according to γ in the BDA feature space. In extreme class imbalance problems, a relatively small number of γ can exhibit higher performance by focusing on the structure of the non-target class. Figure 7b depicts the distribution of data in the BDA feature space when applying the optimal parameters. The distribution of the phishing dataset, which contains significantly less data in the BDA feature space, maintains its distinctive characteristics without losing them. The classification is well-executed based on θ, indicating the appropriate selection of parameters.



(**a**) (**b**)

Figure 7. Evaluation of the BA metric based on parameter settings. (**a**) Mean of BA by γ in the training phase. (**b**) Optimal BDA feature space projected with training data using BA (when, μ = 0.6; γ = 2; θ = 2.4130).

4.3. Phishing Messages Classification Results

In this paper, we conducted a comprehensive evaluation by comparing the objective performance of our proposed method with specific machine learning-based algorithms. The algorithms used in this paper are as follows:

- Stochastic Gradient Descent (SGD) [45]
- Decision Tree (DT) [46]
- Random Forest (RF) [47]
- Naive Bayes (NB) [48]
- Logistic Regression (LR) [49]
- *k*-Nearest neighbor (*k*NN) [50]
- Support Vector Machine (SVM) [51]
- One-Class Support Vector Machine (OCSVM) [52,53]
- Adaptive Boosting (AdaBoost) [54]
- Random Under-Sampling Boosting (RUSBoost) [55]
- Extreme Gradient Boosting (XGBoost) [56]
- Light Gradient Boosting Model (LGBM) [57]
- Convolutional Neural Network (CNN) [24]
- Convolutional Neural Network (CNN) and Gated Recurrent Unit (GRU) [23]
- Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) [24]
- Bidirectional Long Short-Term Memory (BiLSTM) [21]
- Synthetic Minority Over-sampling TEchnique (SMOTE) [58]

This paper applied a range of machine learning algorithms to address class imbalance problems and assess their phishing message classification performance. We evaluated the phishing message classification performance at the algorithm level by utilizing a 5-fold DOB-SCV approach suitable for class-imbalanced datasets. Table 2 presents the results comparing the classification performance of the proposed method with the methods utilized in previous studies using the same BoW generated from the dataset. Regarding the

performance metric Recall in the training phase, the classification performance ranked as follows: 2NN > SMOTE > DT = RF > SVM > LGBM > CNN+LSTM > Proposed Method > BiLSTM > SGD > XGBoost > LR = RUSBoost > CNN+GRU > NB > AdaBoost > CNN > OCSVM = 3NN. Additionally, for the BA performance metric, the classification performance ranked as follows: DT = RF > 2NN > SVM > LGBM > SMOTE > CNN+LSTM > Proposed Method > SGD > BiLSTM > XGBoost > LR > RUSBoost > CNN+GRU > NB > AdaBoost > 3NN > CNN > OCSVM.

In the testing phase, using the optimal parameters, the data that did not overlap with the training dataset was classified to determine whether it was phishing. When considering the Recall metric, the classification performance for the test dataset ranked as follows: Proposed Method > BiLSTM > SMOTE > CNN+GRU > CNN+LSTM > LGBM > SVM > SGD > RUSBoost > XGBoost > 2NN > OCSVM > CNN > LR > AdaBoost > DT > RF > 3NN > NB. Similarly, when considering the BA metric, the classification performance ranked as follows: Proposed Method > SMOTE > BiLSTM > LGBM > SVM > CNN+LSTM > SGD > CNN+GRU > XGBoost > 2NN > RUSBoost > LR > AdaBoost > DT > RF > CNN > 3NN > OCSVM > NB. The proposed method exhibited high classification performance in the dataset with a class imbalance problem, achieving 95.45% in Recall and 96.85% in BA metrics.

The detailed analysis of these results is as follows. It was found that traditional methods, including the probability-based NB and regression-based LR, generally lacked robustness against class imbalance issues. In the *k*NN approach, the parameter choice greatly influenced classification performance, establishing the number of neighbors as a critical determinant of classification efficacy. Both DT and RF demonstrated high classification performance during the training phase. However, there was a significant decrease in classification performance during the testing phase, indicating a potential overfitting issue within non-parametric supervised learning methods. SVM was effective in binary classification problems, exhibiting strong performance in the training and testing phases. Nevertheless, OCSVM showed inferior performance compared to SVM in both the Recall and BA metrics.

Among gradient boosting methods, LGBM displayed relatively high classification performance, yet there was a noticeable gap between training and testing performance. Conversely, AdaBoost, which combines several weak learners to create a strong learner, showed weak classification performance with unknown data and in class imbalance problems. XGBoost, designed to prevent overfitting, did not solve the class imbalance problem effectively. Regarding deep learning-based methods, traditional SGD-utilized neural networks did not exhibit good classification performance in class imbalance problems. RUSBoost, employing the Random Under-Sampling Boosting technique to tackle class imbalance issues, still demonstrated low classification performance throughout both the training and testing phases.

The CNN series generally showed low classification performance, but they had the advantage of producing generalized results due to the small gap in classification performance between training and testing. CNN combined with GRU showed low classification performance in the training phase but outperformed other algorithms in the testing phase, indicating its practical application potential. LSTM is employed in language recognition tasks. However, it has been observed that, under conditions of extreme class imbalance, the performance of models combining LSTM with CNN in classification tasks deteriorates compared to that of BiLSTM models.

During the training phase, the oversampling technique SMOTE was employed to adjust the spam to non-spam data ratio to 1:1. Using SVM as the classifier, this method nearly achieved perfect classification performance for both metrics, approaching 100%. However, in the testing phase, the model experienced a significant decrease in classification performance due to overfitting.

Finally, the method proposed in this study demonstrated lower classification performance during the training phase compared to other algorithms. However, it exhibited high

classification performance in the testing phase and created a more generalized model due to the small gap in outcomes between training and testing.

Table 2. Experimental results of phishing messages classification performance compared with existing methods.

Algorithm	Training		Test		Gap	
	Recall	BA	Recall	BA	Recall	BA
SGD [1]	96.75%	98.37%	91.38%	95.67%	5.37%	2.70%
DT [2]	99.84%	**99.92%**	88.46%	94.17%	11.38%	5.75%
RF [3]	99.84%	**99.92%**	87.64%	93.82%	12.20%	6.10%
NB [4]	93.01%	95.56%	73.66%	85.12%	19.35%	10.44%
LR [5]	94.11%	97.05%	88.78%	94.38%	5.33%	2.67%
2NN [6]	**100.00%**	99.89%	90.24%	94.99%	9.76%	4.90%
3NN [6]	89.51%	94.75%	82.76%	91.37%	6.75%	3.38%
SVM [7]	99.51%	99.76%	91.87%	95.89%	7.64%	3.87%
OCSVM [8]	89.51%	90.33%	90.08%	89.40%	−0.57%	0.93%
AdaBoost [9]	91.14%	95.55%	88.62%	94.27%	2.52%	1.28%
RUSBoost [10]	94.11%	96.32%	91.22%	94.93%	2.89%	1.39%
XGBoostSVM [11]	94.43%	97.22%	90.41%	95.18%	4.02%	2.04%
LGBM [12]	99.39%	99.70%	92.03%	96.00%	7.36%	3.70%
CNN [13]	90.28%	93.56%	88.94%	92.58%	1.34%	0.98%
CNN+GRU [14]	94.00%	96.09%	93.50%	95.37%	0.50%	0.72%
CNN+LSTM [15]	98.62%	99.12%	92.36%	95.86%	6.26%	3.26%
BiLSTM [16]	97.89%	98.17%	94.96%	96.38%	2.93%	1.79%
SMOTE [17]	99.96%	99.67%	94.31%	96.66%	5.65%	3.01%
Proposed Method [18]	98.58%	98.56%	**95.45%**	**96.85%**	3.13%	1.71%

Summary of parameters used during the training phase for each algorithm. [1] loss function = hinge, max iterations = 1000, penalty = euclidean, regularization term = 0.0001, tolerance for stopping criteria = 0.001, width of the insensitive region = 0.1. [2] cost-complexity pruning = 0, criterion = gini index, min impurity decrease = 0, min sample leaf = 1, min sample split = 2, min weight fraction leaf = 0. [3] cost-complexity pruning = 0, criterion = gini index, min impurity decrease = 0, min sample leaf = 1, min sample split = 2, min weight fraction leaf = 0, # of trees in the forest = 100. [4] variance smoothing = 0.000000001. [5] intercept scaling = 1, inverse of regularization strength = 1, max iter = 100, penalty = euclidean, solver = L-BFGS, tolerance for stopping criteria = 0.0001. [6] leaf size = 30, metric = euclidean, weights = uniform. [7] loss function = squared hinge, penalty = euclidean. [8] kernel = radial basis function, gamma = scale, nu = 0.1. [9] learning rate = 0.7, # of estimators = 300. [10] learning rate = 1.0, # of estimators = 50, algorithm = samme real. [11] learning rate = 0.3, maximum tree depth = 6. [12] euclidean regularization term on weights = 0, learning rate = 0.1, Manhattan regularization term on weights = 0, maximum tree leaves = 31, min child samples = 20, min child weight = 0.001, # of boosted trees = 100. [13] batch size = 50, epochs = 30, loss function = binary crossentropy, metric = accuracy, optimizer = adam. [14] batch size = 32, epochs = 30, loss function = binary crossentropy, metric = BA, optimizer = adam. [15] batch size = 100, epochs = 200, loss function = binary crossentropy, metric = BA, optimizer = adam. [16] batch size = 32, epochs = 30, loss function = binary crossentropy, metric = BA, optimizer = adam. [17] random state = 42, k neighbor = 5, loss function = squared hinge, penalty = euclidean. [18] BDA regularization term = 0.5, metric = euclidean, # of PCA feature vectors = 510, # of BDA feature vectors = 2.

5. Conclusions

Globally, there is a growing trend of messenger phishing crimes [5,6]. Particularly in South Korea, with its notably high smartphone penetration rate, messenger phishing is emerging as a significant societal issue [59]. These crimes efficiently exploit phishing messages, allowing culprits to target a broad, unspecified group with minimal effort [1]. To reduce the potential damage from these phishing endeavors, proactive detection and filtering of phishing messages are crucial.

In this paper, we conducted research on classifying phishing in messages received on mobile devices in Korean. During the data conversion phase, morphological analysis (using

MeCab) was carried out on all collected messages to extract features based on verbs and nouns. By measuring the frequency of each feature across all messages, a BoW of numerical data was generated. In the feature engineering phase, we employed the BDA technique, a robust biased learning method, to effectively function under severe class imbalance conditions. In this process, we estimated parameters of BDA such as the regularization parameter ($\mu = 0.65$) and the number of BDA feature vectors ($\gamma = 2$). Importantly, the regularization parameter mitigates the asymmetrical structure between the target and non-target classes and concurrently prevents overfitting, addressing the class imbalance problem. Lastly, in the decision phase, we measure the Euclidean distance between an arbitrary data point and the average vector of phishing data, classifying the message as phishing or non-phishing based on the threshold. For the experiment, we constructed a dataset comprising 615 phishing messages and 42,594 non-phishing messages.

In an experiment involving the classification of Korean phishing messages, characterized by a data scale difference of over tenfold (commonly referred to as the class imbalance problem), our proposed method exhibited performance improvements of at least 0.49% in Recall and 0.19% in BA metrics when compared to machine learning algorithms used in prior studies, such as traditional methods, deep learning-based methods, gradient boosting methods, and non-parametric supervised learning methods. The proposed method effectively utilized the BDA algorithm to classify phishing by analyzing the linguistic differences between phishing and non-phishing messages. In particular, we addressed the class imbalance issue through a normalization strategy due to the significantly lower occurrence of phishing messages than non-phishing messages. When utilized for crime prevention, the proposed phishing message filtering method is expected to ideally lead to a reduction in the damages caused by crime. Furthermore, we anticipate that this approach could be applied to text-transcribed voice phishing-related messages, potentially enhancing its effectiveness in combating voice phishing crimes.

However, the method proposed in this paper presents several limitations. Firstly, finding an optimal combination of parameters for the objective function is challenging. While the appropriate parameter settings can lead to enhanced performance, they also increase the complexity of the algorithm, potentially hindering its practical application. Secondly, the BoW utilized to convert text data into numeric data has its drawbacks. Since the BoW employs every word in the training data as a dimension, it results in a high-dimensional dataset. Moreover, the BoW forms the dataset based solely on word frequencies, neglecting the relative importance of words. Lastly, although our approach demonstrates robustness in the face of the class imbalance problem, it does not guarantee the highest performance under general conditions. Consequently, when a sufficient amount of data is secured, traditional algorithms might achieve superior performance.

In future works, we aim to explore classification algorithms that can be utilized in extreme class imbalance situations without the need for manual parameter tuning. In addition, in this paper, we chose the BoW approach to validate the phishing messages classification performance at the algorithmic level. However, in the process of converting textual data into numerical data, we plan to incorporate state-of-the-art NLP techniques such as BERT [60] and self-attention [61] to not only measure word frequency but also consider the context of sentences. Additionally, we anticipate achieving higher classification performance by combining sampling strategies like the Synthetic Minority Over-sampling TEchnique (SMOTE) [58] with classification algorithms. If possible, with the cooperation of law enforcement agencies, we intend to collect data originating from actual crimes, as opposed to using data directly gathered for phishing messages classification, to validate our classification issues.

Author Contributions: Conceptualization, Y.L.; methodology, H.A. and Y.L.; software, S.K. and J.P.; formal analysis, S.K.; investigation, H.A. and Y.L.; resources, Y.L.; writing—original draft preparation, S.K. and J.P.; writing—review and editing, H.A. and Y.L.; visualization, S.K. and J.P.; supervision, Y.L.; funding acquisition, Y.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Korea Electric Power Corporation (Grant number: R22XO05-07).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used our paper are available on its Github repository: https://github.com/Ez-Sy01/KOR_phishing_Detect-Dataset, accessed on 11 April 2024.

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

Abbreviations

The following abbreviations are used in this manuscript:

ANN	Artificial Neural Network
AdaBoost	Adaptive Boosting
BA	Balanced Accuracy
BDA	Biased Discriminant Analysis
BERT	Bidirectional Encoder Representations from Transformers
BiLSTM	Bidirectional Long Short-Term Memory
BOW	Bag of Words
CBoW	Continuous Bag of Words
CNN	Convolutional Neural Network
DOB-SCV	Distribution-Optimally-Balanced Stratified Cross-Validation
DT	Decision Tree
FDA	Fisher's Discriminant Analysis
GRU	Gated Recurrent Unit
*k*NN	*k*-Nearest Neighbor
KoNLPy	Korean NLP in Python
LGBM	Light Gradient Boosting Model
LR	Logistic Regression
LSTM	Long Short Term Memory
MLP	Multi-Layer Perceptron
NB	Naive Base
NLP	Natural Language Processing
NN	Nearest Neighbor
OCSVM	One-Class Support Vector Machine
PCA	Principal Component Analysis
RF	Random Forest
RusBoost	Random Under-Sampling Boosting
SGD	Stochastic Gradient Descent
SMOTE	Synthetic Minority Over-sampling TEchnique
SNS	Social Network Service
SSSP	Small Sample Size Problem
SVM	Support Vector Machine
Word2Vec	Word to Vector
XGBoost	Extreme Gradient Boosting

References

1. Kim, S.; Lee, Y.; Lee, B. A Study on Countermeasure against Telecommunication Financial Fraud. *Police Sci. Inst.* **2022**, *36*, 343–378. [CrossRef]
2. Lee, H.J. A study on the newtypes of crime using smart phone and the police counter measurements *J. Korean Police Stud.* **2012**, *11*, 319–344.
3. Choi, Y.; Choi, S. Messenger Phishing Modus Operandi in South Korea. *J. Korean Public Police Secur. Stud.* **2021**, *18*, 241–258.
4. Clement, J. Global Number of Mobile Messaging Users 2018–2022. 2019. Available online: https://www.statista.com/ (accessed on 11 April 2024).
5. A critical analysis of cyber threats and their global impact. In *Computational Intelligent Security in Wireless Communications*; CRC Press: Boca Raton, FL, USA, 2023; pp. 201–220.

6. Phishing evolves: Analyzing the enduring cybercrime. In *The New Technology of Financial Crime*; Routledge: London, UK, 2022; pp. 35–61.
7. Annareddy, S.; Tammina, S. A comparative study of deep learning methods for spam detection. In Proceedings of the 2019 Third International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 12–14 December 2019; pp. 66–72.
8. Asaju, C.B.; Nkorabon, E.J.; Orah, R.O. Short message service (sms) spam detection and classification using naïve bayes. *Int. J. Mechatron. Electr. Comput. Technol. (IJMEC)* **2021**, *11*, 4931–4936.
9. Gautam, S. Comparison of Feature Representation Schemes to Classify SMS Text using Data Balancing. *Int. J. Mech. Eng.* **2022**, *7*, 198–209.
10. Navaney, P.; Dubey, G.; Rana, A. SMS spam filtering using supervised machine learning algorithms. In Proceedings of the 2018 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 11–12 January 2018; pp. 43–48.
11. Turhanlar, M.; Acartürk, C. Detecting Turkish Phishing Attack with Machine Learning Algorithm. In Proceedings of the WEBIST, Online, 26–28 October 2021; pp. 577–584.
12. Saeed, V.A. A Method for SMS Spam Message Detection Using Machine Learning. *Artif. Intell. Robot. Dev. J.* **2023**, *3*, 214–228. [CrossRef]
13. Verma, S. Detection of Phishing in Mobile Instant Messaging Using Natural Language Processing and Machine Learning. Ph.D. Thesis, National College of Ireland, Dublin, Ireland, 2023.
14. Choudhary, N.; Jain, A.K. Towards filtering of SMS spam messages using machine learning based technique. In Proceedings of the Advanced Informatics for Computing Research: First International Conference, ICAICR 2017, Jalandhar, India, 17–18 March 2017; Revised Selected Papers; Springer: Berlin/Heidelberg, Germany, 2017; pp. 18–30.
15. Ora, A. Spam Detection in Short Message Service Using Natural Language Processing and Machine Learning Techniques. Ph.D. Thesis, National College of Ireland, Dublin, Ireland, 2020.
16. Al Maruf, A.; Al Numan, A.; Haque, M.M.; Jidney, T.T.; Aung, Z. Ensemble Approach to Classify Spam SMS from Bengali Text. In Proceedings of the International Conference on Advances in Computing and Data Sciences, Kolkata, India, 27–28 April 2023; Springer: Berlin/Heidelberg, Germany, 2023; pp. 440–453.
17. Sonowal, G. Detecting phishing SMS based on multiple correlation algorithms. *SN Comput. Sci.* **2020**, *1*, 361. [CrossRef]
18. Dwiyansaputra, R.; Nugraha, G.S.; Bimantoro, F.; Aranta, A. Deteksi SMS Spam Berbahasa Indonesia menggunakan TF-IDF dan Stochastic Gradient Descent Classifier. *J. Teknol. Inf. Komput. Apl. (JTIKA)* **2021**, *3*, 200–207.
19. Mishra, S.; Soni, D. Implementation of 'smishing detector': An efficient model for smishing detection using neural network. *SN Comput. Sci.* **2022**, *3*, 189. [CrossRef]
20. Gupta, M.; Bakliwal, A.; Agarwal, S.; Mehndiratta, P. A Comparative Study of Spam SMS Detection Using Machine Learning Classifiers. In Proceedings of the 2018 Eleventh International Conference on Contemporary Computing (IC3), Noida, India, 2–4 August 2018; pp. 1–7. [CrossRef]
21. Abayomi-Alli, O.; Misra, S.; Abayomi-Alli, A. A deep learning method for automatic SMS spam classification: Performance of learning algorithms on indigenous dataset. *Concurr. Comput. Pract. Exp.* **2022**, *34*, e6989. [CrossRef]
22. Han, H.; Li, Y.; Zhu, X. Convolutional neural network learning for generic data classification. *Inf. Sci.* **2019**, *477*, 448–465. [CrossRef]
23. Ulfath, R.E.; Alqahtani, H.; Hammoudeh, M.; Sarker, I.H. Hybrid CNN-GRU framework with integrated pre-trained language transformer for SMS phishing detection. In Proceedings of the 5th International Conference on Future Networks & Distributed Systems, Dubai, United Arab Emirates, 15–16 December 2021; pp. 244–251.
24. Ghourabi, A.; Mahmood, M.A.; Alzubi, Q.M. A hybrid CNN-LSTM model for SMS spam detection in Arabic and english messages. *Future Internet* **2020**, *12*, 156. [CrossRef]
25. Kim, Y.B.; Chae, H.; Snyder, B.; Kim, Y.S. Training a korean srl system with rich morphological features. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Baltimore, MD, USA, 22–27 June 2014; pp. 637–642.
26. Park, H.j.; Song, M.c.; Shin, K.s. Sentiment analysis of korean reviews using cnn: Focusing on morpheme embedding. *J. Intell. Inf. Syst.* **2018**, *24*, 59–83.
27. Lim, J.S.; Kim, J.M. An empirical comparison of machine learning models for classifying emotions in Korean Twitter. *J. Korea Multimed. Soc.* **2014**, *17*, 232–239. [CrossRef]
28. Shim, K.S. Syllable-based pos tagging without korean morphological analysis. *Korean J. Cogn. Sci.* **2011**, *22*, 327–345. [CrossRef]
29. Thabtah, F.; Hammoud, S.; Kamalov, F.; Gonsalves, A. Data imbalance in classification: Experimental evaluation. *Inf. Sci.* **2020**, *513*, 429–441. [CrossRef]
30. Ali, H.; Salleh, M.N.M.; Saedudin, R.; Hussain, K.; Mushtaq, M.F. Imbalance class problems in data mining: A review. *Indones. J. Electr. Eng. Comput. Sci.* **2019**, *14*, 1560–1571. [CrossRef]
31. Abd Elrahman, S.M.; Abraham, A. A review of class imbalance problem. *J. Netw. Innov. Comput.* **2013**, *1*, 332–340.
32. Zheng, Z.; Cai, Y.; Li, Y. Oversampling method for imbalanced classification. *Comput. Inform.* **2015**, *34*, 1017–1037.

33. Zhou, X.S.; Huang, T.S. Small sample learning during multimedia retrieval using biasmap. In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2001, Kauai, HI, USA , 8–14 December 2001; Volume 1, p. I–I.
34. Baaqeel, H.; Zagrouba, R. Hybrid SMS Spam Filtering System Using Machine Learning Techniques. In Proceedings of the 2020 21st International Arab Conference on Information Technology (ACIT), Giza, Egypt, 28–30 November 2020; pp. 1–8.
35. Martinez, A.; Kak, A. PCA versus LDA. *IEEE Trans. Pattern Anal. Mach. Intell.* **2001**, *23*, 228–233. [CrossRef]
36. Choubineh, A.; Wood, D.A.; Choubineh, Z. Applying separately cost-sensitive learning and Fisher's discriminant analysis to address the class imbalance problem: A case study involving a virtual gas pipeline SCADA system. *Int. J. Crit. Infrastruct. Prot.* **2020**, *29*, 100357. [CrossRef]
37. Fukunaga, K. *Introduction to Statistical Pattern Recognition*, 2nd ed.; Academic Press: Cambridge, MA, USA, 1990.
38. Turk, M.; Pentland, A. Eigenfaces for recognition. *J. Cogn. Neurosci.* **1991**, *3*, 71–86. [CrossRef]
39. Moreno-Torres, J.G.; Sáez, J.A.; Herrera, F. Study on the impact of partition-induced dataset shift on k-fold cross-validation. *IEEE Trans. Neural Netw. Learn. Syst.* **2012**, *23*, 1304–1312. [CrossRef]
40. Brodersen, K.H.; Ong, C.S.; Stephan, K.E.; Buhmann, J.M. The balanced accuracy and its posterior distribution. In Proceedings of the 2010 20th International Conference on Pattern Recognition, Istanbul, Turkey, 23–26 August 2010; pp. 3121–3124.
41. Conversational Scenarios Collected and Refined from Twitter. 2023. Available online: http://www.aihub.or.kr/aihubdata/data/view.do?currMenu=115&topMenu=100/ (accessed on 8 May 2023).
42. One-Shot Conversation Dataset Containing Korean Emotion Information. Available online: http://aicompanion.or.kr/nanum/tech/data_introduce.php?idx=47,2023 (accessed on 9 May 2023).
43. Chatbot Data for Korean. 2018. Available online: https://github.com/songys/Chatbot_data (accessed on 9 May 2023).
44. Choi, S.I.; Lee, Y.; Kim, C. Confidence Measure Using Composite Features for Eye Detection in a Face Recognition System. *IEEE Signal Process. Lett.* **2015**, *22*, 225–228. [CrossRef]
45. Robbins, H.; Monro, S. A stochastic approximation method. *Ann. Math. Stat.* **1951**, *22*, 400–407. [CrossRef]
46. Safavian, S.R.; Landgrebe, D. A survey of decision tree classifier methodology. *IEEE Trans. Syst. Man, Cybern.* **1991**, *21*, 660–674. [CrossRef]
47. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
48. Rish, I. An empirical study of the naive Bayes classifier. In Proceedings of the IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence, Washington, DC, USA, 2–5 August 2001; Volume 3, pp. 41–46.
49. Menard, S. *Applied Logistic Regression Analysis*; Number 106 in Sage university papers; Quantitative applications in the social sciences; Sage: Thousand Oaks, CA, USA, 1995.
50. Zhang, M.L.; Zhou, Z.H. A k-nearest neighbor based algorithm for multi-label classification. In Proceedings of the 2005 IEEE International Conference on Granular Computing, Beijing, China, 25–27 July 2005; Volume 2, pp. 718–721.
51. Noble, W.S. What is a support vector machine? *Nat. Biotechnol.* **2006**, *24*, 1565–1567. [CrossRef]
52. Schölkopf, B.; Platt, J.C.; Shawe-Taylor, J.; Smola, A.J.; Williamson, R.C. Estimating the support of a high-dimensional distribution. *Neural Comput.* **2001**, *13*, 1443–1471. [CrossRef] [PubMed]
53. Amer, M.; Goldstein, M.; Abdennadher, S. Enhancing one-class support vector machines for unsupervised anomaly detection. In Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description, Chicago, IL, USA, 11–14 August 2013; pp. 8–15.
54. Freund, Y.; Schapire, R.; Abe, N. A short introduction to boosting. *J.-Jpn. Soc. Artif. Intell.* **1999**, *14*, 1612.
55. Seiffert, C.; Khoshgoftaar, T.M.; Van Hulse, J.; Napolitano, A. RUSBoost: A hybrid approach to alleviating class imbalance. *IEEE Trans. Syst. Man,-Cybern.-Part A Syst. Hum.* **2009**, *40*, 185–197.
56. Chen, T.; Guestrin, C. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd ACM Sigkdd International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794.
57. Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.Y. Lightgbm: A highly efficient gradient boosting decision tree. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 3149–3157.
58. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [CrossRef]
59. Wike, R.; Silver, L.; Fetterolf, J.; Huang, C.; Austin, S.; Clancy, L.; Gubbala, S. *Social Media Seen as Mostly Good for Democracy Across Many Nations, but US is a Major Outlier*; Pew Research Center: Washington, DC, USA, 2022; Volume 6.
60. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
61. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 6000–6010.