Proceedings of the 2nd International Conference on Mathematical Physics and Computational Simulation
DOI: 10.54254/2753-8818/39/20240636

Addressing data imbalance in neural network spam detection with insights from SMS spam collection

Siyi He

School of Economics, Xiamen University, Xiamen, Fujian, 361005, China

hesiyi@stu.xmu.edu.cn

Abstract. In cybersecurity, the persistent challenge of spam detection remains paramount. Traditional methods reliant on human scrutiny or rule-based algorithms are proving inadequate against the constantly evolving tactics employed by spammers. Machine learning emerges as a promising solution, leveraging vast datasets to swiftly and objectively discern patterns and traits within spam messages. By uncovering subtle correlations among message elements, machine learning enhances the precision and efficacy of spam detection systems, offering a dependable and economical approach to combat spam. This study aims to investigate the impact of different strategies for addressing data imbalance on neural network-based spam detection performance. Using the SMS Spam Collection Dataset, four methods for mitigating data imbalance are evaluated against an untreated scenario. Notably, despite inherent data imbalance, the unprocessed scenario exhibits the highest overall performance. Stratified sampling emerges as the most effective technique for accurately identifying spam, while SMOTE excels in preserving legitimate messages (ham) while filtering out spam. These results contribute significantly to peoples' understanding of the intricate dynamics in controlling data imbalance in spam detection and offer insightful information for future studies and real-world applications.

Keywords: Spam detection, Data imbalance, Neural network, Machine learning

1. Introduction

Within cybersecurity, the ongoing issue of spam detection remains a crucial problem. Conventional approaches that depend on human examination or algorithms based on predefined rules need to be revised in dealing with the ever-changing strategies used by spammers. Machine learning provides a hopeful alternative by utilizing extensive datasets to swiftly and objectively identify patterns and characteristics present in spam messages. Machine learning improves the accuracy and effectiveness of spam detection systems by revealing subtle connections between message elements. This offers a reliable and cost-effective method to address the constant problem of spam. The study aims to examine the effects of various approaches in handling data imbalance on the performance of neural network-based spam detection. Using the SMS Spam Collection Dataset, four solutions for dealing with data imbalance are compared to a scenario without data imbalance treatment. Remarkably, even though the dataset has an intrinsic imbalance, the unprocessed scenario has the highest overall performance. Stratified sampling is the most effective strategy for accurately predicting spam, whereas SMOTE is particularly good at keeping legitimate messages (ham) while removing spam. These findings provide a

DOI: 10.54254/2753-8818/39/20240636

deeper understanding of the intricate dynamics of managing data imbalance in spam detection, delivering valuable insights for future research and practical implementation.

2. Literature review

Neural networks (NNs) pioneered spam detection, demonstrating efficacy through a NN classifier applied to an email corpus, achieving results on par with existing market solutions [1]. NN algorithms for web spam classification were also influencial in discerning complex patterns [2]. A novel machine learning method for detecting SMS spam, leveraging feature extraction and an averaged NN model, achieved notably high detection rates [3]. Comparisons between NN methodologies and alternative approaches emphasized machine learning dominance in SMS spam classification [4], with Support Vector Machine (SVM) emerging as a standout model due to its high accuracy [5]. Another study analyzed linguistic elements and stylistic features in SMS spam detection, contrasting traditional with emerging deep learning methods [6].NNs solidified their position in spam detection, leading to advancements. A deep learning architecture for spam detection in social media harnessed Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM) models, showcasing enhanced performance [7]. Another model leveraging LSTM and CNN achieved remarkable accuracy by autonomously extracting features from textual data [8]. Additionally, a deep learning model based on BiLSTM for automatic SMS spam classification outperformed traditional classifiers on various datasets [9].Researchers explored Recurrent Neural Networks (RNNs) for spam detection, proposing a method utilizing RNN and LSTM models with impressive accuracy [10]. The introduction of the Lightweight Gated Recurrent Unit (LGRU), a lightweight deep neural model, further bolstered NN applications in SMS spam detection by incorporating semantic context [11].

3. Methodology

3.1. Experimental Setup and Objective

The experimental setup involved a tailored neural network architecture for SMS message classification, optimized using the Adam optimizer with a learning rate of 0.001 and a batch size of 32. A simple neural network model with three fully connected layers was constructed, utilizing ReLU activation for the input and hidden layers and sigmoid activation for the output layer. Four strategies were used to resolve the imbalance in the data: Random Oversampling, SMOTE, Class Weighting, and Undersampling, each performed separately to improve classification performance.

Furthermore, text preprocessing steps were integrated into the setup, including tokenization, punctuation removal, lowercase conversion, stopwords removal, and text-to-vector conversion. These steps ensured appropriate formatting and optimization for machine learning algorithms.

3.2. Dataset

For SMS spam research, this study employed the SMS Spam Collection, which curated SMS messages. The 5,574 English SMS messages are carefully categorised as “ham” (legal) or “spam” Each dataset entry includes two columns: “v1” for the categorical label, “ham” or “spam,” and “v2” for the unaltered text. The statistics show that 87% of items are “ham,” signifying legal messages, and 13% are “spam,” suggesting unwanted or commercial information. Further investigation of the “v2” column shows a diverse textual corpus with 5,169 unique values. This means a large and diverse textual collection for SMS spam analysis and investigation.

4. Neural Network Model

In this section, the paper provides a theoretical overview of the neural network architecture used for spam classification. The neural network is a computational model inspired by the human brain, composed of interconnected layers of artificial neurons. This research describes the fundamental components of the neural network, including its structure, activation functions, loss function, and optimization algorithm.

DOI: 10.54254/2753-8818/39/20240636

4.1. Model Architecture

The neural network architecture consists of an input layer, one or more hidden layers, and an output layer. Let x denote the input vector representing a preprocessed text message. The input layer passes the input vector to the hidden layers, where each neuron computes a weighted sum of its inputs followed by an activation function. Mathematically, the output of neuron j in layer l is given by:

$$a_j^{(l)} = g\left(\sum_{i=1}^{n^{(l-1)}} \omega_{ij}^{(l)} a_i^{(l-1)} + b_j^{(l)}\right) \tag{1}$$

where $a_i^{(l-1)}$ is the output of neuron i in the previous layer, $\omega_{ij}^{(l)}$ is the weight associated with the connection between neuron i and neuron j in layer l, $b_j^{(l)}$ is the bias term of neuron j, and $g(\cdot)$ is the activation function.

4.2. Activation Functions

Activation functions introduce non-linearity into the neural network, enabling it to learn complex mappings between inputs and outputs. Commonly used activation functions include the rectified linear unit (ReLU), sigmoid, and softmax functions. The ReLU activation fu1nction is defined as $g(z) = \max(0, z)$, while the sigmoid function is given by $g(z) = \frac{1}{1+e^{-z}}$. The softmax function is used in the output layer to produce a probability distribution over the output classes.

4.3. Loss Function and Optimization

The choice of loss function depends on the task at hand. For binary classification tasks like spam detection, binary cross-entropy is commonly used:

$$L(y, \hat{y}) = -\frac{1}{N}\sum_{i=1}^{N}(y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)) \tag{2}$$

where N is the number of samples, y_i is the true label of sample i, and $\hat{y}_i$ is the predicted probability of sample i belonging to the positive class.

The optimization algorithm used to train the neural network is Adam, which combines adaptive learning rates with momentum. It updates the parameters of the network based on the gradients of the loss function concerning the parameters.

4.4. Training Process

During training, the neural network learns to minimize the loss function by adjusting its parameters using stochastic gradient descent. The parameters are updated iteratively using backpropagation, where the gradients of the loss function concerning each parameter are computed and used to update the parameter values.

4.5. Evaluation Metrics

To evaluate the performance of the model, several metrics are used, including accuracy, precision, recall and F1 score of both classes. These metrics provide insights into the model's ability to classify spam and non-spam messages correctly and its overall performance on the dataset.

5. Data Imbalance Handling

This section discusses strategies for handling data imbalance in the spam classification task are discussed. Data imbalance occurs when one class (e.g., spam messages) is significantly more prevalent than another (e.g. non-spam messages), leading to biased model performance. Four common techniques for addressing data imbalance are introduced: oversampling, undersampling, Synthetic Minority Over-sampling Technique (SMOTE), and stratified sampling. Uniform mathematical expressions for the

DOI: 10.54254/2753-8818/39/20240636

following methods are as follows: let $N_{majority}$ and $N_{minority}$ represent the number of instances in the majority and minority classes, respectively.

5.1. Oversampling

Oversampling involves increasing the number of instances in the minority class to balance the class distribution. One common technique is random oversampling, where instances from the minority class are randomly duplicated until the class distribution is balanced. The oversampling process can be represented as follows:where α is the oversampling ratio.

$$Oversampled\ N_{minority} = \alpha \times N_{majority} \tag{3}$$

5.2. Undersampling

Undersampling involves reducing the number of instances in the majority class to balance the class distribution. One common undersampling technique is random undersampling, where instances from the majority class are randomly removed until the class distribution is balanced. The undersampling process can be represented as follows:

$$Undersampled\ N_{majority} = \beta \times N_{minority} \tag{4}$$

where β is the undersampling ratio.

5.3. Synthetic Minority Over-sampling Technique (SMOTE)

SMOTE is a synthetic data generation technique that creates synthetic instances of the minority class to balance the class distribution. It works by selecting a random instance from the minority class and generating synthetic instances along the line segments joining its k nearest neighbors. The oversampling ratio determines the number of synthetic instances generated for each minority class instance. The synthetic instance x_{new} is generated as follows:

$$x_{new} = x_{minority} + \lambda \times \left(x_{neighbor} - x_{minority}\right) \tag{5}$$

where $x_{minority}$ is the minority class instance, $x_{neighbor}$ is a randomly selected nearest neighbor of $x_{minority}$, and λ is a random value in the range $[0, 1]$.

5.4. Stratified Sampling

Stratified sampling is a technique used to address data imbalance by sampling data in so that each class is represented in proportion to its occurrence in the original dataset. This method ensures that the class distribution in the sampled dataset reflects the distribution in the original dataset, thereby mitigating the effects of data imbalance.

The stratified sampling process involves selecting a subset of instances from majority and minority classes based on a predefined sampling ratio. Mathematically, the number of instances sampled from each class can be represented as follows:

$$Sampled\ N_{majority} = \gamma \times N_{majority} \tag{6}$$

$$Sampled\ N_{minority} = \gamma \times N_{minority} \tag{7}$$

where γ is the sampling ratio. By preserving the original class distribution in the sampled dataset, stratified sampling helps prevent the loss of valuable information associated with data imbalance.

These data imbalance handling techniques aim to improve the performance of the classification model by providing a more balanced training dataset, thereby reducing the bias towards the majority class and improving the model's ability to generalize to unseen data.

6. Results

The study investigated data imbalance handling in machine learning. Without techniques, the model fit well but had some instability. Oversampling led to jaggedness in loss lines and a rough decision

DOI: 10.54254/2753-8818/39/20240636

boundary. Undersampling had similar jaggedness with lower initial losses. SMOTE was adequate for predicting ham emails. Stratified sampling showed small losses but higher validation losses and excelled in predicting spam mail. The following table presents the classification metrics obtained without any data imbalance handling techniques:

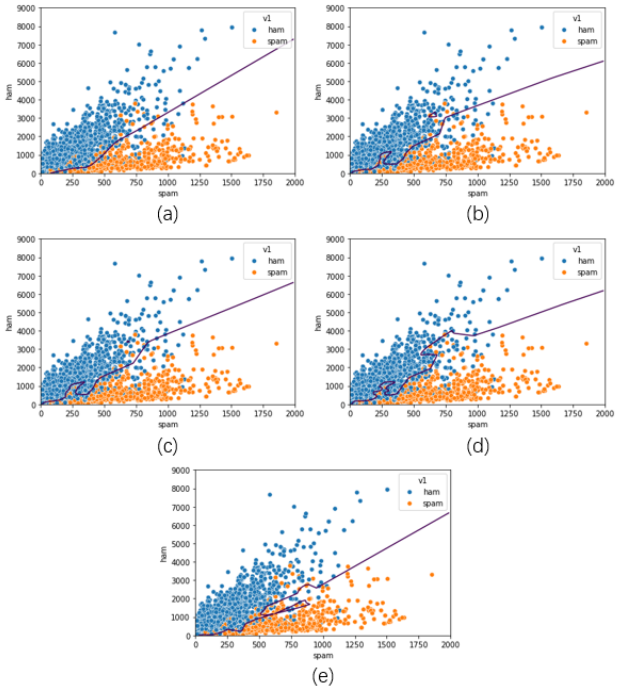

Figure 1. Boundary Curves under Different Data Imbalance Handling Strategies (a: Without Handling, b: Oversampling, c: Undersampling, d: SMOTE, e: Stratified)

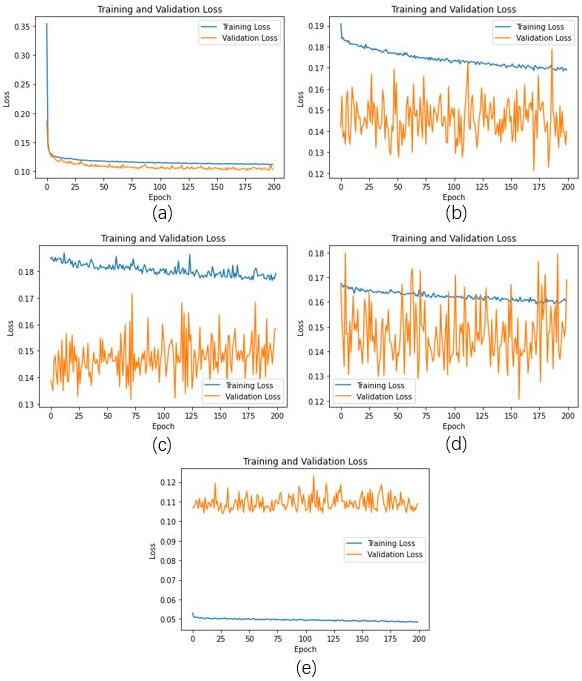

Figure 2. Training and Validation Losses under Different Data Imbalance Handling Strategies (a: Without Handling, b: Oversampling, c: Undersampling, d: SMOTE, e: Stratified)

DOI: 10.54254/2753-8818/39/20240636

Table 1. Performance Metrics of Various Data Handling Techniques in Spam Detection

	Spam precision	Spam recall	Spam F1-score	Ham precision	Ham recall	Ham F1-score
Without Handling	0.93	0.82	0.87	0.97	0.99	0.98
Oversampling	0.75	0.90	0.82	0.98	0.95	0.97
Undersampling	0.72	0.90	0.80	0.98	0.94	0.96
SMOTE	0.68	0.93	0.78	0.99	0.93	0.96
Stratified	0.95	0.82	0.88	0.97	0.99	0.98

Without any data imbalance handling strategies, the model exhibits ideal fitting characteristics. Notably, the validation loss consistently remains lower than the training loss, indicating rapid convergence. However, the validation loss line displays some instability. Conversely, the decision boundary curve appears remarkably smooth. Furthermore, the confusion matrix is provided for reference.

After applying oversampling, the validation and training loss lines exhibit increased jaggedness despite initially lower losses. Additionally, the decision boundary manifests as rougher and more erratic, with discontinuities observed in certain areas. Oversampling performs less optimally than the scenario without any data imbalance handling strategies, although the prediction accuracy for the ham category is relatively improved.

Conversely, undersampling yields similarly jagged validation and training loss lines, albeit with lower losses initially. While better than oversampling, the decision boundary remains inferior to the scenario without any imbalance handling. Analysis of the table highlights a trade-off, where overall predictive accuracy is sacrificed for improved ham prediction.

SMOTE exhibits comparable drawbacks, with an even more jagged and unstable validation loss line. However, the decision boundary displays improvement over undersampling and oversampling, maintaining at least some level of continuity. Additionally, it effectively encompasses a more significant proportion of ham samples. This method emerges as the most effective in predicting ham emails.

Stratified sampling is the only method with a validation loss higher than the training loss. Nevertheless, both losses are the smallest among all strategies. The decision boundary, though continuous, exhibits unevenness. Notably, this strategy excels in predicting spam mail.

7. Conclusion

This study investigates the impact of different strategies on neural network-based spam detection using the SMS Spam Collection Dataset, identifying effective approaches for addressing data imbalance. The conclusion drawn from the study indicates that the best overall predictive performance, method stability, and smoothest decision boundary are achieved when no data imbalance treatment is applied. Specifically, Stratified Sampling performs better in predicting spam, while SMOTE excels in predicting ham.

In conclusion, while the study sheds light on the impact of addressing data imbalance on neural network-based spam detection, certain limitations within the scope of this research deserve attention for future investigation. Although the findings indicate that specific approaches, such as stratified sampling and SMOTE, are superior, their applicability in various datasets and real-world contexts has yet to be thoroughly investigated. Furthermore, using the SMS Spam Collection Dataset may restrict the applicability of the findings to wider circumstances, underscoring the necessity for verification using other datasets. Furthermore, it is necessary to investigate further the effectiveness of neural network architectures in dealing with severely imbalanced datasets. This includes exploring alternate network structures and optimisation methodologies specifically designed to address this particular difficulty. By addressing these limitations, future research can refine the understanding and effectiveness of neural network-based spam detection data imbalance.

DOI: 10.54254/2753-8818/39/20240636

References

[1] Ndumiyana D, Magomelo M, Sakala L. Spam detection using a neural network classifier [J]. 2013.

[2] Chandra A, Suaib M, Beg D R. Web spam classification using supervised artificial neural network algorithms [J]. arXiv preprint arXiv:1502.03581, 2015.

[3] Sheikhi S, Kheirabadi M T, Bazzazi A. An effective model for SMS spam detection using content-based features and averaged neural network [J]. International Journal of Engineering, 2020, 33(2): 221-228.

[4] Abayomi-Alli O, Misra S, Abayomi-Alli A, et al. A review of soft techniques for SMS spam classification: Methods, approaches and applications [J]. Engineering Applications of Artificial Intelligence, 2019, 86: 197-212.

[5] Jain T, Garg P, Chalil N, et al. SMS spam classification using machine learning techniques [C]//2022 12th international conference on cloud computing, data science & engineering (confluence). IEEE, 2022: 273-279.

[6] Odera D, Odiaga G. A comparative analysis of recurrent neural network and support vector machine for binary classification of spam short message service [J]. World Journal of Advanced Engineering Technology and Sciences, 2023, 9(1): 127-152.

[7] Jain G, Sharma M, Agarwal B. Spam detection in social media using convolutional and long short term memory neural network [J]. Annals of Mathematics and Artificial Intelligence, 2019, 85(1): 21-44.

[8] Roy P K, Singh J P, Banerjee S. Deep learning to filter SMS Spam [J]. Future Generation Computer Systems, 2020, 102: 524-533.

[9] Abayomi-Alli O, Misra S, Abayomi-Alli A. A deep learning method for automatic SMS spam classification: Performance of learning algorithms on indigenous dataset [J]. Concurrency and Computation: Practice and Experience, 2022, 34(17): e6989.

[10] Chandra A, Khatri S K. Spam SMS filtering using recurrent neural network and long short term memory [C]//2019 4th international conference on information systems and computer networks (ISCON). IEEE, 2019: 118-122.

[11] Wei F, Nguyen T. A lightweight deep neural model for SMS spam detection [C]//2020 International Symposium on Networks, Computers and Communications (ISNCC). IEEE, 2020: 1-6.