Enhancing URL Phishing Detection using GAN with LSTM and BERT Models

Adarsh Regulapati
Masters Student of CS Dept.,
The Pennsylvania State University
Harrisburg, PA, USA
abr6174@psu.edu

Dr. Bimal Ghimire
Assistant Professor of CS Dept.,
The Pennsylvania State University
Harrisburg, PA, USA
bxg5417@psu.edu

Dr. Jeremy Blum
Associate Professor of CS Dept.,
The Pennsylvania State University
Harrisburg, PA, USA
jjb24@psu.edu

Abstract — A URL (Uniform Resource Locator) is a web address that directs users to specific resources on the internet. A phishing attack is a type of cyber-attack that uses social engineering tactics to steal sensitive information from victims. Phishing URLs are fraudulent web addresses that mimic legitimate websites to deceive users into divulging sensitive information, such as usernames, passwords, and financial details. These attacks are harmful as they can lead to financial loss, identity theft, and the compromise of personal and corporate data. As attackers continue to develop new patterns of phishing URLs, balancing the dataset with active phishing URLs is challenging because active phishing URLs are fewer than the available legitimate URLs.

This paper proposes an approach using Generative Adversarial Networks (GANs) to generate synthetic URLs from available active phishing URLs. Using these generated synthetic URLs, we balance the dataset and train the phishing detection model. This approach prevents the model from overfitting the majority class, and by using active phishing URLs to generate synthetic URLs, it ensures the model remains updated. The GAN architecture consists of two components: a generator and a discriminator. We used an LSTM model for the generator and BERT model for the discriminator. This model provides a solution to dataset imbalance problem in phishing URL detection.

Keywords: Generative Adversarial Network (GAN), Phishing URL Detection, LSTM Generator, BERT Discriminator, Uniform Resource Locator(URL), Synthetic URLs Generation, Imbalanced Dataset, Adversarial Learning, URL Classification.

I. Introduction

Phishing attacks continue to pose a significant threat to individuals and organizations worldwide. Phishing is a deceptive practice where attackers trick users into revealing sensitive information, such as login credentials and financial details, by impersonating legitimate websites through fraudulent URLs. These attacks have evolved in complexity and scale, making them increasingly difficult to detect with traditional methods. In 2023, phishing attacks surged by 58.2% compared to the previous year [1], reflecting the growing sophistication of cybercriminals. According to the Anti-Phishing Working Group (APWG), nearly 964,000 phishing attacks were recorded in the first quarter of 2024 alone [2]. Furthermore, According to the report of Kaspersky, 85% of detected web threats are caused by malicious URLs [3]. This underscores the need for more advanced detection systems to combat the rising tide of phishing threats.

Despite the seriousness of URL phishing as a cyber-attack, no perfect model exists to fully address the problem. The primary challenges in detecting phishing URLs include balancing datasets and the dynamic nature of phishing URLs. Phishing URLs are significantly fewer in number compared to legitimate ones, making it difficult to create balanced datasets for training machine learning models. This imbalance can lead to overfitting, where models become too focused on the majority class (legitimate URLs) and perform poorly at detecting phishing attempts. Additionally, phishing URLs are highly dynamic, with new and unpredictable patterns emerging rapidly. Blacklists, often relied upon in traditional systems, become obsolete quickly and fail to detect evolving phishing threats

Traditional phishing URL detection methods often rely on blacklists or heuristic-based systems [4]. Blacklists match URLs against a predefined list of known threats, which is effective for familiar attacks but fails to detect new phishing URLs or evolving attack patterns. Similarly, rule-based systems can be bypassed by attackers who can easily modify their URLs to evade detection once the system identifies a pattern. These limitations highlight the need for a model capable of generating more synthetic URLs using the limited set of newly detected phishing URLs.

In this research, we propose the use of Generative Adversarial Networks (GANs) for generating new phishing URLs. GANs consist of two components: a generator and a discriminator. The generator in our approach is based on a Long Short-Term Memory (LSTM) network, which is well-suited for modeling sequential data, such as URLs. LSTMs excel at capturing the structural patterns of URLs and can generate realistic, diverse phishing URLs, even with varying lengths and formats. The discriminator, implemented with BERT

(Bidirectional Encoder Representations from Transformers), is designed to classify URLs as either real or synthetic. BERT's ability to understand contextual relationships within the text makes it highly effective at distinguishing between real or synthetic URLs.

By combining LSTM and BERT in a GAN framework, we offer a powerful solution to address the data imbalance issue in phishing URL detection. The LSTM generator learns and mimics the sequential structure of URLs, while the BERT discriminator leverages its contextual understanding to evaluate the quality of generated data. Together, they work to produce high-quality synthetic phishing URLs that can balance the dataset effectively.

The synthetic URLs generated by the GAN are then used to train another pre-trained BERT model for phishing and legitimate URL classification. When trained on the balanced dataset, this BERT model significantly outperformed its counterpart trained on an imbalanced dataset. The results demonstrate that our approach not only improves phishing detection but also adapts dynamically to emerging threats, offering a more resilient solution for cybersecurity defenses.

II. Related Work

Phishing attacks have grown in frequency and sophistication, posing significant security threats to internet users by targeting sensitive information. As conventional detection techniques such as blacklisting and heuristics fall short in identifying emerging threats, researchers have turned to machine learning (ML), deep learning (DL), and generative adversarial networks (GANs) for more robust phishing detection solutions. This section discusses key advances in phishing detection, focusing on GAN-based methods, transformer-based architectures, ensemble models, and hybrid approaches.

A. GAN-Based Approaches for Phishing Detection

Generative Adversarial Networks (GANs) are increasingly used to address class imbalance and generate synthetic phishing URLs for model training. Pham *et al.* [5] introduced a WGAN-GP-based method to generate phishing URLs, which were combined with the original dataset to train LSTM and GRU models. Their experiments demonstrated that GAN-generated samples improved detection accuracy when the dataset was small but provided diminishing returns with larger datasets.

Jafari and Aghaee-Maybodi [6] proposed a four-step approach involving dataset balancing using Conditional GAN (CGAN) and feature extraction via TF-IDF. They utilized a hybrid WSO-WOA optimization algorithm for feature selection, mapping selected features as RGB images for ResNet50 input. Their results showed that CGAN outperformed traditional GANs, achieving higher accuracy than deep learning architectures like VGG19 and DNN-BiLSTM.

Trevisan and Drago [7] applied GANs for URL classification, focusing on learning URL patterns to distinguish between legitimate and malicious URLs without needing extensive training data. Their results highlighted GAN's potential for phishing detection, though scaling the model to larger datasets remains a challenge.

Anand *et al.* [8] also utilized GANs for oversampling imbalanced phishing datasets. Their method, which generated synthetic phishing URLs based on textual patterns, demonstrated improvements in detection performance, emphasizing the need for continuous updates as phishing strategies evolve.

B. Transformer-Based Models and Hybrid Architectures

Transformer-based models, known for their ability to extract complex patterns and semantic meaning, have been highly effective for phishing detection. Sasi and Balakrishnan [9] introduced a GAN architecture with a VAE as the generator and a transformer with self-attention as the discriminator. Their method achieved 97.75% accuracy on a dataset of one million URLs, showing the efficacy of combining VAE's generation capabilities with a transformer's classification strength.

Su *et al.* [10] employed BERT for phishing detection, utilizing it for both raw URL strings and feature-engineered datasets. Their BERT-based approach achieved remarkable performance across multiple datasets. The study demonstrated that BERT could effectively handle non-natural language tasks by extracting meaningful patterns from URLs.

Jishnu and Arthi [11] further enhanced phishing detection using RoBERTa for feature extraction and LSTM for sequential classification. Their hybrid model achieved 97.14% accuracy, leveraging RoBERTa's semantic understanding alongside LSTM's sequential processing capabilities.

Jishnu and Arthi [12] also proposed a comprehensive phishing detection system using BERT tokenization and additional URL feature extraction. Their method, evaluated on a large dataset of 200,000 URLs, achieved 97.32% accuracy.

C. Ensemble Models and Comparative Evaluations

Ensemble models have been widely explored for phishing detection due to their ability to combine multiple weak learners for improved performance. Mankar *et al.* [4] conducted a comparative evaluation of machine learning models for phishing detection, demonstrating that Random Forest and Extra Trees achieved the highest accuracy (over 91%). However, these models struggled with precision and recall when detecting minority phishing samples, highlighting the importance of data balancing techniques.

In a similar vein, Albahadili *et al.* [13] integrated GAN-generated samples with a CNN-LSTM architecture and optimized feature selection using the white shark optimizer (WSO) algorithm. Their model achieved high accuracy across multiple datasets, including ISCX 2016 and PhishTank, outperforming traditional deep learning methods such as CNN-LSTM. However, they noted that model complexity and computational demands posed challenges for real-time implementation.

D. Impact of Dataset Balancing and Feature Engineering

Effective feature engineering and dataset balancing are crucial for phishing detection models. Jafari and Aghaee-Maybodi [6] emphasized the importance of balancing datasets using

CGAN to reduce classification errors. They combined handcrafted features with TF-IDF and mapped selected features as RGB images for input into ResNet50. Their experiments demonstrated that hybrid optimization methods significantly improved phishing detection performance.

Anand *et al.* [8] demonstrated that GAN-based oversampling provided better results than traditional methods by generating synthetic URLs that addressed dataset imbalance. Their research stressed the need for frequent updates to datasets, given the dynamic nature of phishing threats.

Albahadili *et al.* [13] further explored feature selection through the WSO algorithm, adapting features for CNN and LSTM models. Their system, which converted URLs into numerical matrices, achieved high performance on multiple datasets, demonstrating the effectiveness of intelligent feature selection and GAN-based balancing techniques.

The reviewed literature highlights the growing role of GANs, transformers, and hybrid deep learning models in phishing detection. GAN-based methods effectively address class imbalance by generating synthetic phishing URLs, while transformers like BERT and RoBERTa excel in extracting complex patterns from URLs. Ensemble models such as Random Forest and CNN-LSTM provide robust performance but face challenges with class imbalance and computational overhead. This research builds on these insights by integrating GANs with transformer-based architectures to develop a comprehensive phishing detection framework capable of handling zero-day attacks and real-time deployment.

III. Methodology

The proposed model workflow begins with an imbalanced dataset, which is first visualized to analyze its distribution. The data set is then divided into training, validation, and testing sets. Using the minority class of the training set, a GAN is utilized to generate high-quality synthetic phishing URLs. These synthetic URLs are used to balance the dataset, which is subsequently used to train a BERT model for phishing URL classification. The effectiveness of the GAN-generated synthetic URLs is evaluated by training the BERT model twice: first with the imbalanced dataset and then with the balanced dataset. The results are compared to demonstrate the benefits of using the synthetic data to balance the imbalanced dataset. Figure 1 illustrates the workflow of the proposed model.

A. GAN Architecture

The proposed methodology employs a GAN architecture with an LSTM model as the generator and a BERT model as the discriminator. This system is designed to generate high-quality synthetic phishing URLs, addressing dataset imbalance and improving the performance of phishing URL detection models. Figure 2 illustrates the architecture of the GAN model, and the following paragraphs outline the key components and steps of the GAN architecture used in this project.

B. LSTM Model as Generator

The generator in a GAN is responsible for creating synthetic data that closely mimics real data, aiming to fool the discriminator into classifying it as real. In our proposed model generator generate real looking synthetic URL's [14]. An LSTM model is used for the generator.

LSTM networks excel at processing sequential and time-series data due to their ability to retain information over extended periods, which makes them particularly suitable for generating synthetic phishing URLs [15]. Unlike other types of neural networks, LSTMs are composed of memory blocks, called cells, that store and manage information through three key gates: the forget gate, the input gate, and the output gate. These gates enable the selective storage, update, and retrieval of relevant information, allowing LSTMs to effectively handle long sequences.

Phishing URLs inherently have a sequential structure where the order of characters and tokens is critical [16]. LSTMs are well-suited for such tasks as they excel in processing sequential data, enabling the generation of realistic URL patterns. Additionally, LSTMs are designed to capture long-term dependencies within sequences, allowing the generator to maintain contextual relationships in URLs. This ensures that the generated URLs mimic not only the syntax of real phishing URLs but also their meaningful patterns. Moreover, LSTMs are flexible and can handle variable-length sequences, making them ideal for generating URLs of varying lengths and complexities. This flexibility is crucial for accurately mimicking the diverse structures found in phishing URLs, enhancing the quality of the synthetic data.

C. BERT Model as Discriminator

The discriminator in a GAN is used to distinguishing between real data (phishing URLs from the dataset) and synthetic data (URLs generated by the generator). Its primary role is to evaluate the quality of the generator's outputs and provide feedback, helping the generator improve over time. In the GAN framework, the discriminator and generator are adversaries—the discriminator aims to correctly classify URLs as real or synthetic, while the generator seeks to produce data so realistic that it fools the discriminator [17]. BERT model is used as the discriminator in this research.

BERT model as the discriminator in the GAN framework for phishing URL generation and classification offers significant advantages due to its state-of-the-art natural language understanding capabilities. BERT (Bidirectional Encoder Representations from Transformers) uses a transformer-based architecture with self-attention mechanisms to understand the context of words and tokens. Phishing URLs often contain subtle patterns, such as typo squatting, domain spoofing, or misleading subdomains, which require contextual awareness to detect [10].

Unlike traditional models, BERT reads text bidirectionally, meaning it considers the entire context of a sequence from both left-to-right and right-to-left. This bidirectional approach is particularly useful for analyzing URLs, where the significance

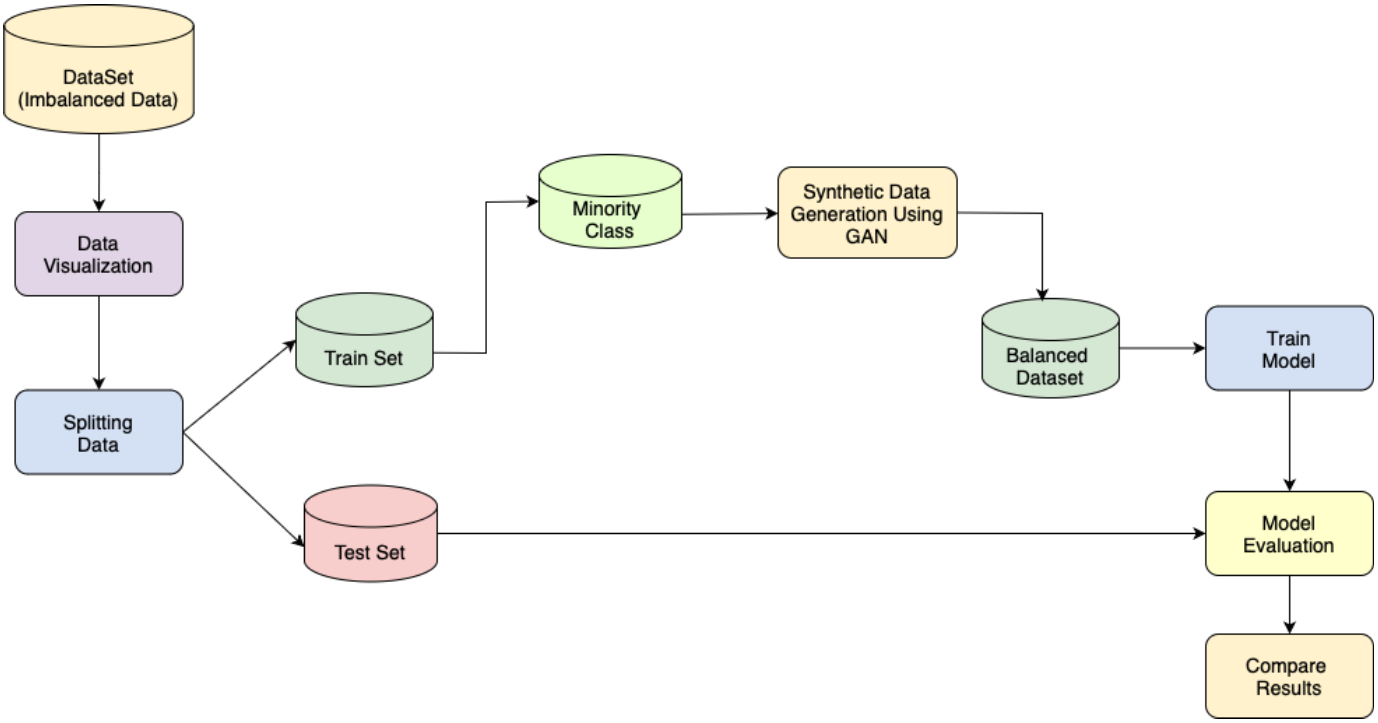


Fig. 1. Workflow of the proposed model

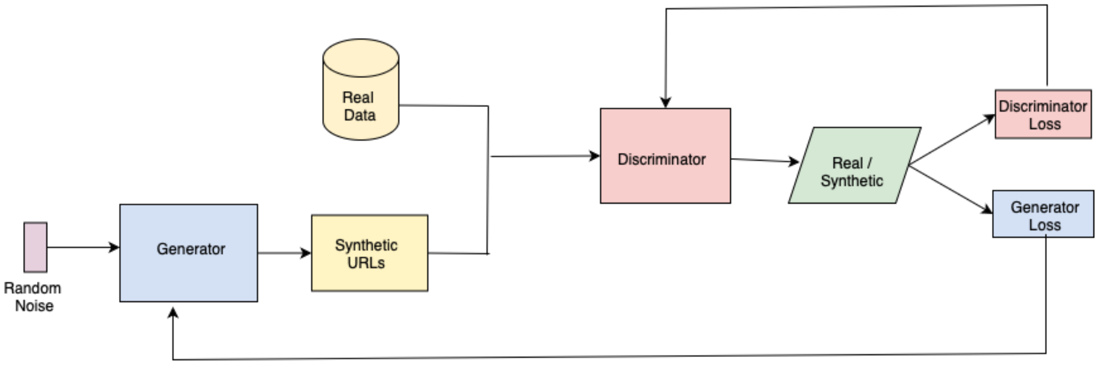


Fig. 2. Generative Adversarial Network Architecture

of tokens often depends on their surrounding context. For example, subdomains, directory paths, or query parameters might indicate malicious behavior only when considered in conjunction with the full URL [12].

URLs vary significantly in length, ranging from short domains to long, complex query strings. BERT's ability to process variable-length sequences (up to a maximum of 512 tokens) makes it ideal for handling this diversity in URL structures.

As a discriminator in the GAN, BERT not only evaluates real and synthetic URLs but also provides feedback to the generator. Its highly accurate classification helps the generator improve its ability to produce realistic synthetic URLs that are indistinguishable from real phishing URLs.By incorporating BERT as the discriminator, the GAN framework benefits from the model's superior language understanding, resulting in high-quality synthetic data generation.

D. Adversarial Learning

Adversarial learning is the core mechanism driving the GAN framework in this project. It involves a competitive process between two models: the generator and the discriminator. The generator aims to create synthetic phishing URLs that resemble real ones, while the discriminator attempts to differentiate between real and synthetic URLs. This interplay enables both models to improve iteratively, leading to the generation of high-quality synthetic data.

1) Loss Function: Binary Cross-Entropy Loss: The binary cross-entropy loss function is used for both the generator and the discriminator:

- *For the Discriminator*: The discriminator is trained to assign a label of 1 to real phishing URLs and 0 to synthetic URLs. Its loss function is calculated as:

$$\text{Loss}_D = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)\right]$$

  where y_i is the true label (1 for real, 0 for synthetic), and $\hat{y}_i$ is the discriminator's predicted probability.
- *For the Generator*: The generator is trained to fool the discriminator, aiming for the discriminator to classify its outputs as 1 (real). Its loss is calculated as:

$$\text{Loss}_G = -\frac{1}{N}\sum_{i=1}^{N}\log(\hat{y}_i)$$

  where $\hat{y}_i$ is the discriminator's predicted probability of the synthetic URL being real.

2) Optimizer: Adam Optimizer: The Adam optimizer is used for both the generator and the discriminator due to its efficiency and ability to handle sparse gradients. Adam combines the benefits of momentum and adaptive learning rates, making it well-suited for the dynamic nature of GAN training.

- The optimizer adjusts model parameters (weights and biases) based on the calculated gradients of the loss function with respect to these parameters.
- Adam uses moving averages of the gradients and squared gradients to adapt the learning rate during training.

3) Feedback Loop:

- *Discriminator Feedback*: During each iteration, the discriminator processes a batch of real URLs and a batch of synthetic URLs from the generator. Based on its performance (loss), the discriminator receives feedback in the form of gradients computed using binary cross-entropy loss. The Adam optimizer updates the discriminator's weights to improve its ability to correctly classify real and synthetic URLs.
- *Generator Feedback*: The generator uses the discriminator's predictions as feedback. If the discriminator successfully identifies synthetic URLs as fake, the generator incurs a higher loss. Gradients are computed from the generator's loss, indicating how its parameters should change to make its outputs more realistic. The Adam optimizer updates the generator's weights to reduce the loss, enabling it to produce better-quality synthetic URLs in subsequent iterations.

4) Weight and Parameter Adjustment: The weights of the generator and discriminator are adjusted through backpropagation. Gradients are calculated using the chain rule, propagating the loss from the output layer to the earlier layers. The Adam optimizer uses these gradients to update each parameter:

$$\theta_t = \theta_{t-1} - \alpha \cdot \frac{m_t}{\sqrt{v_t} + \epsilon}$$

where m_t and v_t are the first and second moment estimates of the gradients, α is the learning rate, and ϵ is a small constant for numerical stability.

5) Iterative Improvement: Over successive iterations, the discriminator becomes better at distinguishing real from synthetic URLs, while the generator improves its ability to produce realistic phishing URLs. This adversarial process continues until the generator produces synthetic data indistinguishable from real data, effectively balancing the dataset for BERT model training. Adversial

E. BERT Model for Classification

To evaluate the effectiveness of the synthetic data generated by the GAN, we used a separate BERT model for phishing URL classification. This BERT model, pre-trained on a vast corpus of text data, serves as a strong foundation for fine-tuning on the specific task of classifying phishing and legitimate URLs [18]. Its pre-trained knowledge equips it with a deep understanding of linguistic structures and patterns, which can be effectively leveraged to detect phishing tactics hidden within URLs. We trained this model first with an imbalanced dataset and then with a balanced dataset, comparing the results to analyze the effectiveness of the synthetic data in addressing the dataset imbalance.

IV. Results

In this section we discuss about the dataset, data preprocessing techniques and the experimental results to demonstrate the effectiveness of generated synthetic data.

A. Dataset

The main objective of this research is to balance the imbalanced dataset using GAN-generated synthetic data. For this purpose, 100 thousand commonly used URLs were obtained from the DomCop website, and thousand recently reported phishing URLs were collected from the URLhaus website. The prepared dataset consists of two columns: URL and label (legitimate or phishing). The dataset was split into training, validation, and testing sets in an 80:10:10 ratio.

B. Data Pre-Processing

Each URL in the dataset is converted into a sequence of integer indices representing its characters using the `char2idx` mapping. This character-level encoding transforms textual URLs into a numerical format that can be processed by the LSTM network. Since URLs vary in length, shorter sequences are padded with the Padding token to ensure all sequences have the same length as the longest URL in the dataset. Padding is applied using a pad sequence function.

For the BERT discriminator and BERT classifier, URLs are tokenized using the `BertTokenizer`. URLs are preprocessed by removing the `http://`, `https://`, and `www.` prefixes to focus on their core structure. An attention mask is used to indicate which tokens are padding (`0`) and which are actual tokens (`1`).

To prevent the discriminator from becoming overconfident, label smoothing is applied. Real labels are smoothed to `0.9` (instead of `1`), and fake labels are smoothed to `0.1` (instead of `0`).

C. Evaluation Metrics

To evaluate the performance of the model, the following metrics are used: Accuracy, Precision, Recall, and F1 Score. These metrics provide a comprehensive understanding of the model's ability to classify phishing and legitimate URLs.

1) Accuracy: Accuracy measures the proportion of correctly classified samples out of the total number of samples. It reflects how often the model makes correct predictions.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

2) Precision: Precision measures the proportion of correctly predicted phishing URLs out of all samples predicted as phishing. It indicates how reliable the model is when predicting a URL as phishing.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

A high precision means the model has a low false positive rate.

3) Recall: Recall, also known as sensitivity or the true positive rate, measures the proportion of correctly identified phishing URLs out of all actual phishing URLs. It evaluates the model's ability to detect phishing URLs.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

A high recall indicates the model successfully captures most phishing URLs, even if it occasionally misclassifies legitimate URLs.

4) F1 Score: The F1 Score is the harmonic mean of Precision and Recall. It provides a balanced metric that considers both false positives and false negatives, making it particularly useful for imbalanced datasets.

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

The F1 Score ranges from 0 to 1, with higher values indicating better performance.

D. Hyper Parameters

These are the hyper parameters used in the research

Category	Hyperparameter Value/Description
General Settings	
Device Selection	`'cuda'` or `'cpu'` based on availability
Batch Size	32
Dataset Parameters	
Vocabulary Size	Number of unique characters (including special characters)
Maximum Word Length	Maximum URL length in the dataset
Real Label Smoothing	0.9
Fake Label Smoothing	0.1
Generator	
Noise Dimension	100
Hidden Dimension	128
Number of LSTM Layers	2
Dropout Rate	0.5
Discriminator	
Number of Labels	1 (binary classification)
Tokenizer Max Length	20
Training Parameters	
Learning Rate	2×10^{-4}
Betas for Adam Optimizer	(0.5, 0.999)
Epochs	10
Loss Function	BCEWithLogitsLoss
Freeze Discriminator Layers	Base layers of BERT frozen

TABLE I
HYPERPARAMETRS USED IN ENHANCING URL PHISHING DETECTION USING GAN WITH LSTM AND BERT MODELS

E. Experiment Results

The proposed model with above mentioned. Hyper parameters performed well, the discriminator loss and generator loss curve of GAN model is shown in figure-3

Figure 4 shows a sample of synthetic URLs generated by the GAN, utilizing an LSTM model as the generator and a BERT model as the discriminator. The generator effectively captures the structure of a URL, correctly differentiating the domain name and the path using a '/'. The generated URLs are promising, and with further fine-tuning of the GAN, it is possible to produce even more realistic URLs.

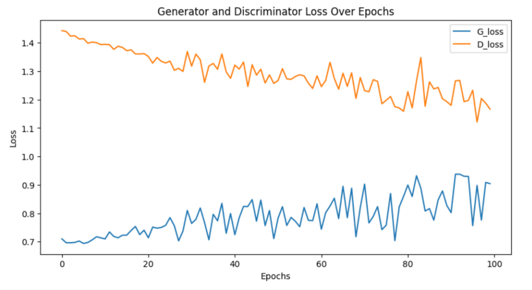


Fig. 3. GAN Generated loss and Discriminator loss

Classification BERT model Training and validation loss curves are shown in figure-5, the discriminator loss is decreasing gradually but the generator cure is not decreasing this means in the adversarial leaning process the discriminator is dominating the generator.

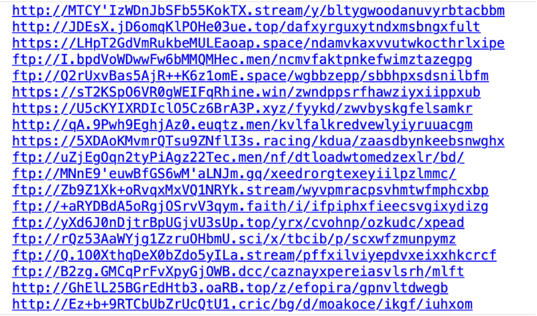
Classification BERT model Training and validation Accuracy curves are shown in figure-6, the training accuracy is 0.99 while the validation accuracy is 0.94.

In this experiment, we first trained the BERT model using an imbalanced dataset consisting of 100,000 legitimate URLs and 30,000 phishing URLs. The results of the BERT model trained with imbalanced dataset are as shown in table-2

Metric	Value
Testing Accuracy	91.1%
Precision	0.97
Recall	0.84
F1 Score	0.80

TABLE II
RESULTS OF THE BERT MODEL ON THE IMBALANCED DATASET.

Next, we balanced the imbalanced dataset using synthetic URLs and trained the BERT model with the same parameters.



```
http://MTCY'IzWDnJbSFb55KokTX.stream/y/bltygwoodanuvyrbtacbbm
http://JDEsX.jD6omqKlPOHe03ue.top/dafxyrguxytndxmsbngxfult
https://LHpT2GdVmRukbeMULEaoap.space/ndamvkaxvvutwkocthrlxipe
ftp://I.bpdVoWDwwFw6bMMQMHec.men/ncmvfaktpnkefwimztazegpg
ftp://Q2rUxvBas5AjR++K6z1omE.space/wgbbzepp/sbbhpxsdsnilbfm
https://sT2KSpO6VR0gWEIFqRhine.win/zwndppsrfhawziyxiippxub
https://U5cKYIXRDIclO5Cz6BrA3P.xyz/fyykd/zwvbyskgfelsamkr
http://qA.9Pwh9EghjAz0.euqtz.men/kvlfalkredvewlyiyruuacgm
https://5XDAoKMvmrQTsu9ZNflI3s.racing/kdua/zaasdbynkeebsnwghx
ftp://uZjEgOqn2tyPiAgz22Tec.men/nf/dtloadwtomedzexlr/bd/
ftp://MNnE9'euwBfGS6wM'aLNJm.gq/xeedrorgtexeyiilpzlmmc/
ftp://Zb9Z1Xk+oRvqxMxVQ1NRYk.stream/wyvpmracpsvhmtwfmphcxbp
ftp://+aRYDBdA5oRgjOSrvV3qym.faith/i/ifpiphxfieecsvgixydizg
ftp://yXd6J0nDjtrBpUGjvU3sUp.top/yrx/cvohnp/ozkudc/xpead
ftp://rQz53AaWYjg1ZzruOHbmU.sci/x/tbcib/p/scxwfzmunpymz
ftp://Q.1O0XthqDeX0bZdo5yILa.stream/pffxilviyepdvxeixxhkcrcf
ftp://B2zg.GMCqPrFvXpyGjOWB.dcc/caznayxpereiasvlsrh/mlft
http://GhElL25BGrEdHtb3.oaRB.top/z/efopira/gpnvltdwegb
http://Ez+b+9RTCbUbZrUcQtU1.cric/bg/d/moakoce/ikgf/iuhxom
```

Fig. 4. GAN Generated URLs

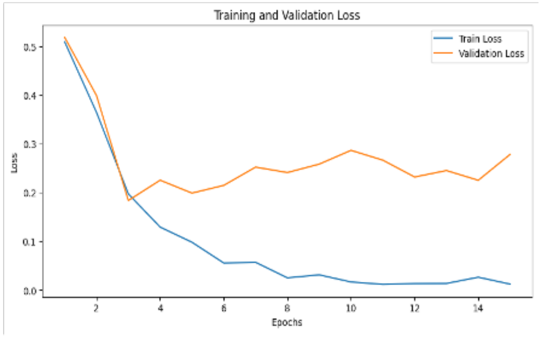


Fig. 5. BERT Classifier Loss Curves

We observed an improvement in performance metrics when the model was trained with the balanced dataset. Notably, there was a significant improvement in recall and F1 score, indicating fewer false positives and false negatives. The metrics are as shown in table-3.

Metric	**Value**
Testing Accuracy	94.62%
Precision	0.98
Recall	0.91
F1 Score	0.94

TABLE III
RESULTS OF THE BERT MODEL AFTER BALANCING THE DATASET WITH SYNTHETIC URLS.

If we compare the metrics, the accuracy increased by 3.52%. In the imbalanced dataset, we had 100,000 legitimate URLs and 30,000 phishing URLs. Due to this imbalance, the model was biased toward legitimate URLs, which accounted for approximately 76.92% of the dataset. As a result, the model primarily classified legitimate URLs correctly, leading to a seemingly high accuracy. However, this came at the cost of a significant number of false positives, where phishing URLs were misclassified as legitimate.

With the balanced dataset, we observed the following improvements: precision increased by 1%, recall improved by 7%, and the F1-score rose by 14%. These changes indicate a significant reduction in false positives and false negatives, showcasing a substantial improvement in the model's performance.

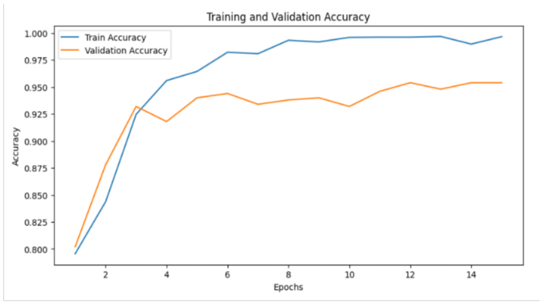


Fig. 6. BERT Classifier Accuracy Curves

V. CONCLUSION AND FUTURE WORK

In this research, we addressed the challenge of imbalanced datasets in phishing URL detection by generating synthetic phishing URLs using a Generative Adversarial Network (GAN). The GAN framework, consisting of an LSTM-based generator and a BERT-based discriminator, demonstrated its capability to produce realistic synthetic phishing URLs that effectively balanced the dataset. This balanced dataset significantly improved the performance of the phishing detection model, as evidenced by increases in accuracy, precision, recall, and F1 score.

There is no perfect model to detect phishing URLs, we need to keep our detection model up to date with available phishing data. Leveraging recently discovered suspicious URLs to generate synthetic URLs and using these synthetic URLs to train detection models ensures that the model remains updated against evolving phishing tactics.

The existing GAN-based data augmentation models have used different generators and discriminators. In this research, I used LSTM and BERT models for the generator and discriminator, respectively. The results highlight the potential of the proposed GAN-based approach to address the dataset imbalance problem.

The future directions of this research include enhancing the efficiency and stability of the adversarial process to generate even higher-quality synthetic URLs. Additionally, the approach can be expanded to other domains beyond phishing detection, such as fraud detection or malware classification, to address imbalanced datasets in various fields. Another potential development is the creation of a user-friendly web-based application for generating synthetic data on demand, making the solution more accessible and practical for real-world applications.

REFERENCES

[1] Zscaler, “Phishing attacks rise 58

[2] Anti-Phishing Working Group (APWG), “Apwg phishing activity trends report,” 2024, accessed: 2024-12-16. [Online]. Available: https://apwg.org/trendsreports/

[3] Kaspersky, “Malware variety grows by 137% in 2019 due to web skimmers,” 2019, accessed: 2024-12-16. [Online]. Available: https://www.kaspersky.co.uk/about/press-releases/malware-variety-grows-by-137-in-2019-due-to-web-skimmers

[4] N. Mankar, P. Sakunde, S. Zurange, A. Date, V. Borate, and Y. Mali, “Comparative evaluation of machine learning models for malicious url detection,” 04 2024, pp. 1–7.

[5] T. Pham, T. Sy, T. Hoang, T. Thanh, T. Phm, and V. Cuong, “Exploring efficiency of gan-based generated urls for phishing url detection,” 11 2021.

[6] S. Jafari and N. Aghaee-Maybodi, “Detection of phishing addresses and pages with a data set balancing approach by generative adversarial network (gan) and convolutional neural network (cnn) optimized with swarm intelligence,” *Concurrency and Computation: Practice and Experience*, vol. 36, no. 11, p. e8033, 2024.

[7] M. Trevisan and I. Drago, “Robust url classification with generative adversarial networks,” *ACM SIGMETRICS Performance Evaluation Review*, vol. 46, no. 3, pp. 143–146, 2019.

[8] A. Anand, K. Gorde, J. R. A. Moniz, N. Park, T. Chakraborty, and B.-T. Chu, “Phishing url detection with oversampling based on text generative adversarial networks,” in *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 2018, pp. 1168–1177.

[9] J. K S and B. Arthi, “Generative adversarial network-based phishing url detection with variational autoencoder and transformer,” *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 13, pp. 2165–2172, 06 2024.

[10] M.-Y. Su and K.-L. Su, “Bert-based approaches to identifying malicious urls,” *Sensors*, vol. 23, no. 20, p. 8499, 2023.

[11] K. Jishnu and B. Arthi, “Phishing url detection by leveraging roberta for feature extraction and lstm for classification,” in *2023 Second International Conference on Augmented Intelligence and Sustainable Systems (ICAISS)*. IEEE, 2023, pp. 972–977.

[12] ——, “Enhanced phishing url detection using leveraging bert with additional url feature extraction,” in *2023 5th International Conference on Inventive Research in Computing Applications (ICIRCA)*. IEEE, 2023, pp. 1745–1750.

[13] A. J. S. Albahadili, A. Akbas, and J. Rahebi, “Detection of phishing urls with deep learning based on gan-cnn-lstm network and swarm intelligence algorithms,” *Signal, Image and Video Processing*, pp. 1–17, 2024.

[14] M. Durgadevi *et al.*, “Generative adversarial network (gan): A general review on different variants of gan and applications,” in *2021 6th International Conference on Communication and Electronics Systems (ICCES)*. IEEE, 2021, pp. 1–8.

[15] S. Al-Ahmadi, A. Alotaibi, and O. Alsaleh, “Pdgan: Phishing detection with generative adversarial networks,” *IEEE Access*, vol. 10, pp. 42 459–42 468, 2022.

[16] E. S. Aung and H. Yamana, “Malicious url detection : A survey,” 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:235620351

[17] S. Li, J. Wu, X. Xiao, F. Chao, X. Mao, and R. Ji, “Revisiting discriminator in gan compression: A generator-discriminator cooperative compression scheme,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 28 560–28 572, 2021.

[18] A. Filali, M. Merras *et al.*, “Enhancing spam detection with gans and bert embeddings: A novel approach to imbalanced datasets,” *Procedia Computer Science*, vol. 236, pp. 420–427, 2024.