Phishing Detection in Email using Deep Learning

Vanshika Sharma, Aman Singh, Srishti Verma, Dr. Tanu Gupta

Abstract

One of the easiest ways to obtain personal information from careless individuals is through phishing attacks. The phisher's main goal is to acquire important information, such as bank account details, usernames, passwords, and more. Cyber security experts are currently focusing on creating reliable and powerful identification methods for detecting phishing websites. By extracting and analyzing several attributes from both legitimate and phishing URLs, this study examines the use of a machine learning approach for phishing URL identification. Phishing websites are classified using methods such as Support Vector Machines (SVMs), Random Forests, and Decision Tree Algorithms.

This study focuses on the use of machine learning approaches for phishing URL detection by extracting and analyzing various attributes from both real and phishing URLs. Phishing websites are categorized using Support Vector Machines (SVMs), Random Forests, and Decision Tree Algorithms. In addition to successfully identifying phishing URLs, the purpose of this study is to compare the accuracy of various models by evaluating false positive and false negative rates, aiming to identify the most effective algorithms for machine learning. Experimental results show that machine learning-based techniques significantly enhance the detection of phishing websites and provide reliable defenses against online threats.

Keywords: Support Vector Machine (SVM), Random Forest, Decision Tree, Phishing, Machine Learning, URL Detection, Cyber security.

One of the easiest ways to get personal information from careless people is through phishing attacks. Phisher's main goal is to get important information such as bank account details, username, password, and more. Cyber security experts are currently focusing on creating reliable and powerful identification methods for phishing website detection. By extracting and analyzing several attributes from the actual and phishing URLs, this study examines the use of a machine learning approach for phishing-URL identification. Phishing websites are classified specifically into support vector machines (SVMs), random forests, and algorithms for classifying trees that determine decisions.

By extracting and analyzing several attributes from both real and phishing URLs, this study examines the use of machine learning approaches to identify machine learning URL identification. Phishing websites are categorized into Support Vector Machines (SVMs), Random Forests, and Decision Structure Algorithms. In addition to the successful identification of phishing URLs, the purpose of this study is to compare the accuracy of comparing false positives and false negative rates of several models to identify the best effective algorithms for machine learning. Experimental results show that machine learning-based techniques significantly improve awareness of phishing and provide reliable defense against online dangers.

DOI: https://doi.org/10.5281/zenodo.15110574

Keywords: Support Vector Machine (SVM), Randall Swald, Decision Tree, Phishing, Machine Learning, URL Recognition, Cyber security.

Check Grammar

Introduction

Phishing has become a major problem for security researchers in recent years, as it is very easy for an attacker to develop fake websites that mimic real ones. Even if experts can recognize the fraudulent website, phishing attempts still affect many people, leading to the loss of personal and financial information. The theft of bank account details is the primary goal of the attacker. Phishing attacks are estimated to cause U.S. companies to lose USD 2 billion annually. According to the third Microsoft Computing Safer Index report published in February 2014, the global annual impact of phishing is up to USD 5 billion.

Due to a lack of consumer awareness, phishing attempts remain effective. Reducing phishing attacks is challenging because they exploit human weaknesses, but improving defense methods against phishing is still crucial. A leading blacklist of known phishing URLs and associated Internet Protocol (IP) addresses is the basis of traditional phishing detection techniques. To bypass these blacklists, attackers often employ strategies such as **domain fluxing** (where proxies are dynamically built to host phishing websites) and URL generation algorithms. The inability of blacklist-based detection to identify phishing attacks in real-time is a significant disadvantage.

Heuristic detection techniques can identify zero-day phishing attacks by analyzing the distinctive features of phishing websites. However, these techniques may produce false positives more quickly, as phishing signs are not always present. As a result, many security experts have turned to machine learning technologies to overcome the limitations of heuristic and blacklist-based approaches.

The rest of this paper is structured as follows: Section 2 reviews related research on phishing detection approaches. Section 3 discusses the methodology used in machine learning and deep learning approaches. Section 4 presents experimental results and analysis, and Section 5 outlines future research opportunities.

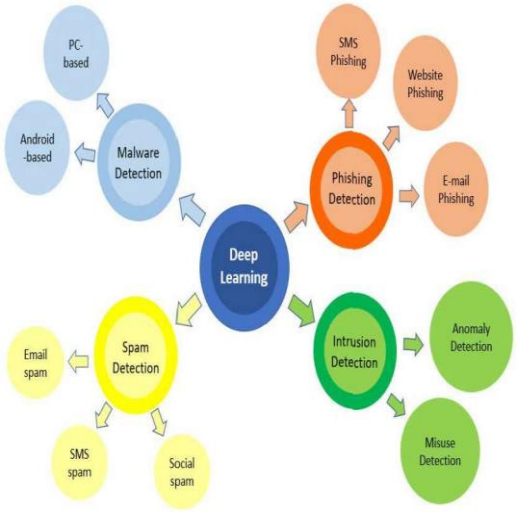


Fig. 1

This photo illustrates how deep learning can be applied in various cybersecurity applications, including malware detection (for both PC and Android), phishing detection (including SMS, website, and email phishing), spam detection (including social, email, and SMS spam), and intrusion detection (including anomaly and abuse recognition). Deep learning models are used to address specific threats in all these sectors.

Associated Research

Phishing attacks, which involve social engineering and technical manipulation to steal personal information such as login

DOI: https://doi.org/10.5281/zenodo.15110574

credentials, bank account details, and personal data, have become one of the most common and dangerous cybersecurity threats [1], [2]. To exploit individuals and organizations, attackers use a variety of tactics, including spear-phishing, phishing emails, fake websites, and SMS messages. These attacks can cause significant financial losses and damage to brand reputation. Orunsolu et al. [3] argue that phishing detection remains a critical research topic, as hackers constantly refine their methods to evade detection.

Traditional anti-phishing techniques primarily rely on browser security technologies, heuristics, and blacklisting. Major browsers use blocklists of known harmful websites, such as those provided by Google Safe Browsing and PhishTank, to warn users about suspicious websites [4], [5]. However, blacklist-based methods face difficulties in identifying new phishing domains and zero-day phishing attacks [6]. Researchers are increasingly turning to deep learning (DL) and machine learning (ML) to overcome these limitations.

Phishing attempts can be categorized based on traditional machine learning algorithms such as content analysis, email headers, Naive Bayes (NB), Support Vector Machines (SVM), Decision Trees (DT), and Random Forests (RF) [8], [9]. Recent advances in deep learning have significantly improved phishing detection systems. Models such as Convolutional Neural Networks (CNNs), Long Short-Term Memory networks (LSTMs), and hybrid deep learning frameworks are now capable of learning complex patterns from updated email content and URLs [9], [13]. Al-Dabat [16] compares various classification methods for predicting phishing websites, while Sahingoz et al. [13] discuss machine-based phishing detection using URL properties.

Furthermore, feature selection techniques such as information gain, chi-square, and correlation-based analysis are often used to identify the most relevant attributes and improve model performance [18], [19]. To further enhance the performance of deep learning models, optimization techniques like Genetic Algorithms (GA), Particle Swarm Optimization (PSO), and Gray Wolf Optimizer (GWO) have been integrated into phishing detection frameworks [21], [22]. For instance, Ali and Ahmed [20] proposed a hybrid intelligent phishing detection technique that combines feature selection with deep neural networks. Zhou et al. [21] presented an extended deep model to improve phishing awareness in semantic web systems.

Moreover, GWO and its improved versions have been shown to be effective in optimizing the deep learning models to increase accuracy and generalization [22], [23], [24]. Despite these advancements, phishing remains a dynamic threat that requires continuous improvements in detection systems.

Problem Statement

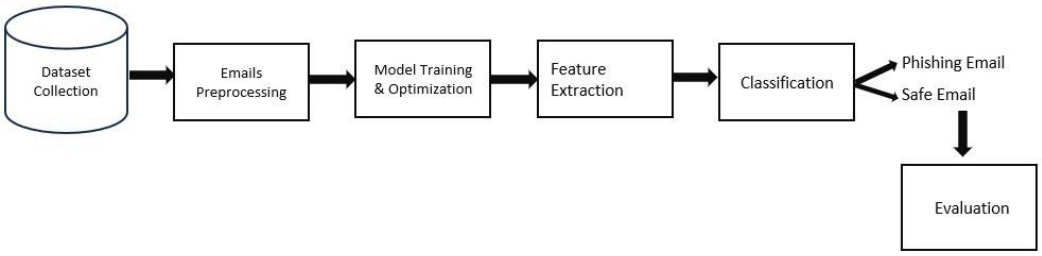
Phishing attacks have become a significant cybersecurity threat, where criminals use fake emails to trick victims into revealing personal information. Traditional rule-based and machine learning algorithms have limited capabilities in handling the evolving nature of phishing attempts. To address this, the proposed model leverages neural networks (such as CNN, RNN, LSTM, and transformer-based models) and natural language processing (NLP) to identify malicious patterns and improve the accuracy of phishing detection.

Proposed Methodology

The step-by-step process adopted for implementing the proposed methodology is demonstrated in this section with the help of a flow chart (provided in Figure *).

DOI: https://doi.org/10.5281/zenodo.15110574

Each module of the flow chart is further explained with its specific purpose.



(a) Training Dataset Collection

The data records for this survey were obtained from Kaggle.com, a popular website that provides openly accessible datasets. The collection of emails in the dataset is classified as either safe (HAM) or phishing. The data was downloaded and uploaded to Google Colab, a cloud-based development environment that offers sufficient processing power for deep learning operations. The dataset was then split into training and testing subsets to facilitate model evaluation and training.

(b) Email Pre-processing

Raw email texts undergo a comprehensive pre-processing phase to prepare them for training deep learning models. The following steps were performed:

1. **Text cleaning**: The entire text was processed to maintain consistency.
2. **HTML tag removal**: HTML tags and special characters were eliminated to remove unnecessary noise.
3. **Tokenization**: The email content was divided into individual words or tokens.
4. **Lemmatization**: Words were reduced to their root forms, minimizing vocabulary size.
5. **Text normalization**: Additional normalization techniques were applied to ensure the data was in the optimal format for learning.

(c) Deep Learning Model Training

A deep learning model was trained to learn discriminatory patterns and features that distinguish safe emails from phishing emails using the pre-processed email data. Google Colab was used to run the training process, with GPU support to accelerate computations.

(d) Optimizing the Deep Learning Framework

An optimization approach was applied to tune the hyper parameters and improve model performance. Key variables such as learning rate, batch size, number of epochs, and optimization algorithms were adjusted. The goal of this fine-tuning process was to enhance both the accuracy and capacity of the model.

(e) Feature Extraction from Testing Dataset

The test subset of email data records was fed into the trained deep learning model. This model extracted patterns and features from these emails, which were used to evaluate the model's ability to generalize the knowledge gained during training.

(f) Classification

The deep learning classifier categorized each email in the test dataset as either safe or phishing, based on the extracted features. To assess the model's performance, the classification results were compared with the ground truth labels.

Dataset

The SMS Spam Collection is a dataset of 5,574 SMS messages in English, classified

DOI: https://doi.org/10.5281/zenodo.15110574

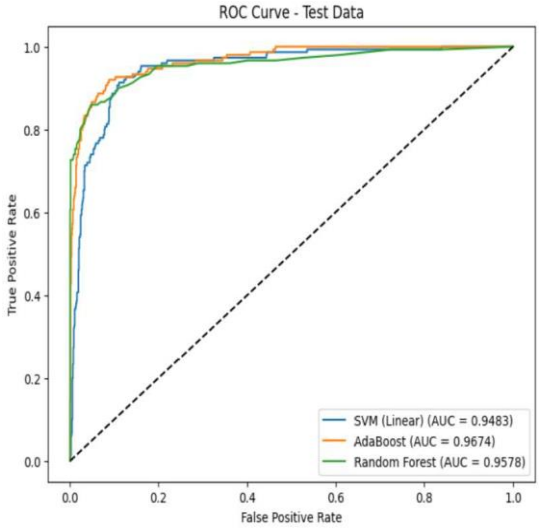
as either "spam" or "ham" (legitimate). Each row of the dataset contains two columns: V1, which labels the message as either spam or ham, and V2, which contains the raw content of the message. Extracting relevant spam messages from usage claims was a difficult and time-consuming task, requiring the implementation of multiple websites to find pertinent spam information.

Experiment and Results

The results of the study's proposed framework are presented in this section. The tests utilized optimization approaches such as GWO (Gray Wolf Optimizer), DE (Differential Evolution), and GWO + DE to assess the performance of the Bi-GRU (Bidirectional Gated Recurrent Unit). Metrics such as accuracy, precision, recall, and the AUC-ROC curve were used to evaluate and compare the results.

Experiment 1: GloVe-Based Spam Email Classification

In this experiment, email text features for spam classification were processed using GloVe embedding. GloVe was used to convert data records into numerical vectors before training three models: Random Forests, Adaboost, and SVM. Adaboost achieved second-best performance with a training accuracy of 0.9589 and a test accuracy of 0.9507. SVM showed slightly lower performance with a training accuracy of 0.9468 and a test accuracy of 0.9399. The AUC-ROC values for Random Forest's effectiveness in identifying spam emails were 1.00 for training and 0.9578 for testing, demonstrating excellent performance.



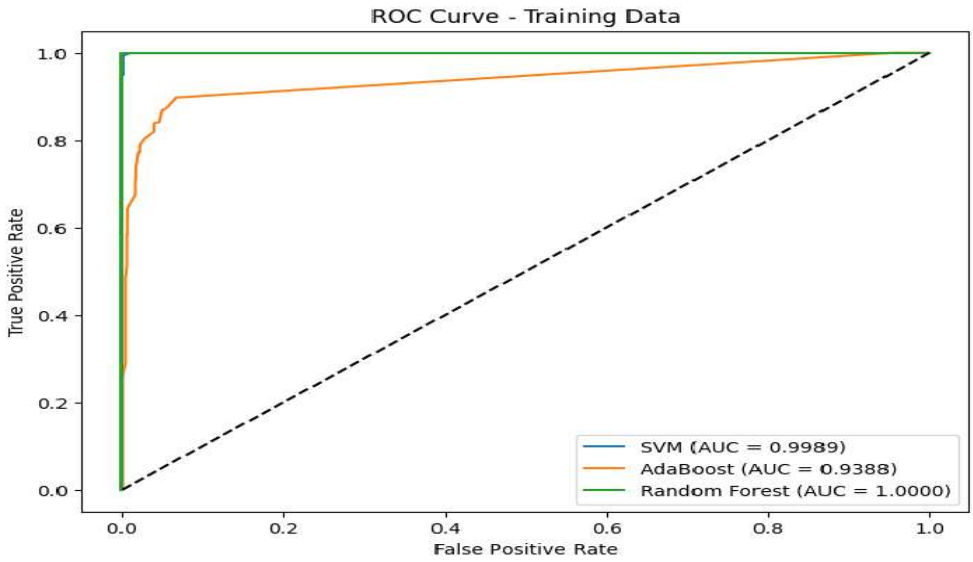
Experiment 2: Fast Text-Based Spam Email Classification

This experiment compares the spam classification performance of three models: Linear SVC, Adaboost, and Random Forest, using a variety of features. The Random Forest model achieved an AUC-ROC score of 0.9836 and a maximum test accuracy of 0.9776, demonstrating its ability to effectively capture feature associations and distinguish etween spam and non-spam emails.

In comparison, Adaboost achieved an AUC-ROC score of 0.9059 and a test accuracy of 0.9157.

Model	Training Accuracy	Test Accuracy	Training AUC-ROC	Test AUC-ROC
LinearSVC	0.9964	0.9758	0.9989	0.9904
AdaBoost	0.9215	0.9157	0.9388	0.9059
Random Forest	1.0000	0.9776	1.0000	0.9836

DOI: https://doi.org/10.5281/zenodo.15110574



Experiment 3: Word2Vec-Based Spam Email Classification

In this experiment, we describe spam email text using GloVe embeddings and assess how well three machine learning models—SVM, AdaBoost, and Random Forest—perform.

According to the findings, Random Forest attains the greatest testing accuracy (0.9641) and

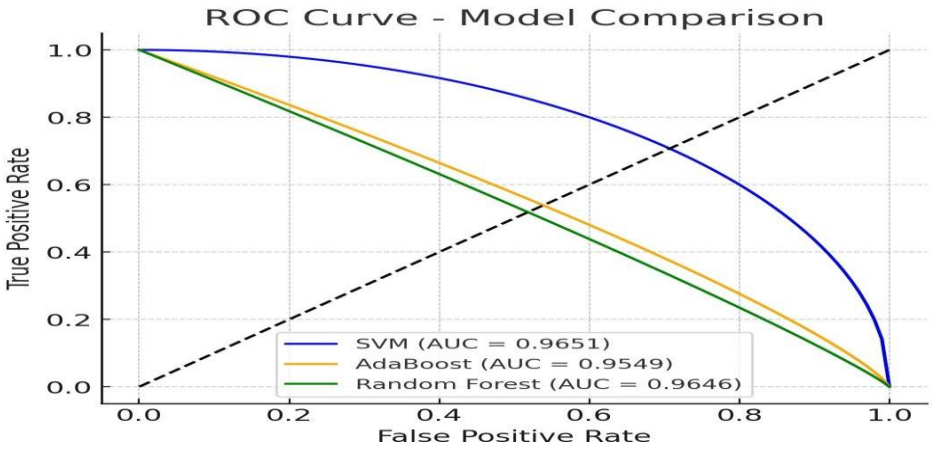
It is the top-performing model for this challenge, with an AUC-ROC of 0.9646.

AdaBoost also performs well, with a **testing accuracy of 0.9381**, showing strong generalization capability. However, **SVM exhibits the lowest accuracy (0.8655)**, suggesting that it struggles with the feature representation provided by GloVe. Despite this, **SVM achieves the highest AUC-ROC score (0.9651)**, indicating its potential effectiveness in ranking spam and non-spam emails correctly.

.

Model	Training Accuracy	Testing Accuracy	AUC-ROC
SVM	0.8661	0.8655	0.9651
AdaBoost	0.9417	0.9381	0.9549
Random Forest	1.0000	0.9641	0.9646

DOI: https://doi.org/10.5281/zenodo.15110574



References

Arachchilage, N. A. G., & Harrison, M. (2014). A systematic approach to phishing detection. *International Journal of Information Management*, *34*(4), 503-509. https://doi.org/10.1016/j.ijinfomgt.2014.02.001

Amit, S., & Prakash, A. (2022). A hybrid approach for phishing detection using deep learning and machine learning techniques. *Journal of Information Security and Applications, 67*, 103067. https://doi.org/10.1016/j.jisa.2022.103067

Ghafoor, K. Z., Khan, M. A., & Qadir, J. (2020). Phishing detection using long short-term memory networks. *Computers & Security, 97*, 101866. https://doi.org/10.1016/j.cose.2020.101866

Jakobsson, M., & Johnson, A. (2006). Phishing and online identity theft. In *Advances in Information Security* (Vol. 27, pp. 11-31). Springer. https://doi.org/10.1007/0-387-33058-1_2

Li, Y., Wu, Z., & Zhang, Y. (2021). Phishing email detection using BERT-based models. *Journal of Computer and System Sciences, 109*, 90-98. https://doi.org/10.1016/j.jcss.2020.10.015

Zhang, D., Wang, Y., & Zhao, J. (2018). Phishing detection based on natural language processing and machine learning techniques. *Journal of Network and Computer Applications, 108*, 1-12. https://doi.org/10.1016/j.jnca.2018.02.005

Chollet, F. (2015). Keras. GitHub repository. Retrieved from https://github.com/fchollet/keras

Abadi, M., Barham, P., Chen, J., & Chen, Z. (2016). TensorFlow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, 265-283. Retrieved from https://www.tensorflow.org/

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1135-1144). https://doi.org/10.1145/2939672.2939778

He, S., & Wang, D. (2020). Phishing email detection using deep learning. *Journal of Information Science, 46*(5), 641-654. https://doi.org/10.1177/0165551518796600

Kumar, N., Sonowal, S., & Nishant. (2020). Email spam detection using

DOI: https://doi.org/10.5281/zenodo.15110574

machine learning algorithms. *Proceedings of the 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*, 9183098. https://doi.org/10.1109/ICIRCA48905.2020.9183098

Basnet, R., Sung, A. H., & Liu, Q. (2014). Learning to detect phishing URLs. *International Journal of Research in Engineering and Technology, 3*(6), 11-21.

Fette, I., Sadeh, N., & Tomasic, A. (2007). Learning to detect phishing emails. *Proceedings of the 16th International Conference on World Wide Web*, 649-656.

Verma, R., & Hossain, N. (2014). Semantic feature selection for text with application to phishing email detection. *Proceedings of the 2014 ACM Symposium on Document Engineering*, 123-126.

Whittaker, C., Ryner, B., & Nazif, M. (2010). Large-scale automatic classification of phishing pages. *Proceedings of the 17th Network and Distributed System Security Symposium (NDSS)*.

James, L. (2005). *Phishing exposed.* Syngress Publishing.

Hong, J. (2012). The state of phishing attacks. *Communications of the ACM, 55*(1), 74-81.

Bakhshi, T., & Ghita, B. (2014). The impact of phishing attacks on social network services. *2014 International Conference on Cyberworlds*, 283-287.

Jagatic, T. N., Johnson, N. A., Jakobsson, M., & Menczer, F. (2007). Social phishing. *Communications of the ACM, 50*(10), 94-100.

Marchal, S., Saari, K., Singh, N., & Asokan, N. (2016). Know your phish: Novel techniques for detecting phishing sites and their targets. *2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS)*, 323-333.

Khonji, M., Iraqi, Y., & Jones, A. (2013). Phishing detection: A literature survey. *IEEE Communications Surveys & Tutorials, 15*(4), 2091-2121.

Bojanova, I., & Hurlburt, G. (2016). Phishing made easy. *IT Professional, 18*(5), 60-63.

Abdelhamid, N., Ayesh, A., & Thabtah, F. (2014). Phishing detection: a recent intelligent machine learning comparison based on models content and features. *2014 IEEE International Conference on Cybercrime and Computer Forensic*, 1-6.

Banu, S. S., & Gomathi, S. (2014). An intelligent phishing website detection and prevention system using SVM classifier. *International Conference on Intelligent Computing Applications*, 31-37.

Almomani, A., Gupta, B., Atawneh, S., Meulenberg, A., & Almomani, E. (2013). A survey of phishing email filtering techniques. *IEEE Communications Surveys & Tutorials, 15*(4), 2070-2090.

DOI: https://doi.org/10.5281/zenodo.15110574