# Phishing Detection Using Machine Learning

**Abstract**

Phishing is an evident cybersecurity threat. Cybercriminals often pose as reputable organizations and send emails with links to phishing websites to unsuspecting individuals. Individuals who enter phishing websites, expose themselves to data breaches and malware. Companies spend time and money investing in cybersecurity training to protect their employees and customers from phishing attempts. Nevertheless, phishing is still prevalent in today's society. In an October 2022 study, conducted by security provider SlashNext, found more than 255 million phishing attempts in email, mobile, and browser channels. SlashNext reports that there has been a 61 percent increase in phishing attempts since 2021. To solve this problem, we propose to create a phishing detection system based on machine learning algorithms. The problem is essentially a classification task of whether a link is legitimate or phishing. We will train the model to detect a phishing website based on a subset of features.

## Introduction

Phishing has become a major concern in the past few years due to the ease of creating websites that look similar to legitimate websites and has become an evident cybersecurity threat. Cybercriminals often pose as reputable organizations and send emails with links to phishing websites to unsuspecting individuals. Individuals who enter phishing websites, expose themselves to data breaches and malware. Companies spend time and money investing in cybersecurity training to protect their employees and customers from phishing attempts. Nevertheless, phishing is still prevalent in today's society. In an October 2022 study, conducted by security provider SlashNext, found more than 255 million phishing attempts in email, mobile, and browser channels. SlashNext reports that there has been a 61 percent increase in phishing attempts since 2021. To solve this problem, we propose to create a phishing detection system based on machine learning algorithms. The problem is essentially a classification task of whether a link is legitimate or phishing.

The three algorithms that we used to implement a phishing detection system are: KNN Classifier, Logistic Regression, and Decision trees. KNN classifier is a supervised learning method that classifies new data based on similarity with existing classified data. Logistic regression is another supervised learning

method used to classify data into two classes based on probabilities determined by a subset of features. A decision tree is a decision-making aid that employs a model of potential outcomes based on certain conditions. These outcomes are determined by previous decisions in the training data. All three algorithms can be used in a classification task such as fraud detection. The purpose of this project is to determine the algorithm that will generate the most accurate phishing detection model.

Our work greatly differs from other projects utilizing this dataset because we use two new algorithms not previously used: KNN classifiers and decision trees. We chose to utilize logistic regression since this is commonly used for detecting spam emails. Though one existing project utilizes logistic regression, the project fails to include any relevant plots that describe the accuracy of the machine learning models. Therefore, our project offers a better comparison of machine learning algorithms for a classification task.

# Related Work

On Kaggle.com, there are 19 different projects utilizing this dataset. Most of these projects utilize different algorithms to train a model that classifies phishing URLs. The algorithms used include the following: neural networks, support vector machines, naive bayes, logistic regression, and random forest.In general, convoluted, recurrent and artificial neural networks have been extensively used in past projects. One project of note is Phishing Detection with Bait by Tyler Sullivan and Matthew Franglen. This project implements three algorithms: naive bayes, support vector machines, and logistic regression. The main advantage of this project is that it uniquely transforms the dataset and produces the machine learning models using vectorization and pipelines. The main disadvantage of this project is that it lacks any pertinent plots to describe accuracy of the machine learning models.

# Our Solution

## Description of Dataset

The first step in building the phishing detection model website is to choose an appropriate dataset that will consist of both phishing and legitimate websites, which will be used for training a model for predicting phishing URLs. The chosen dataset can be found in Kaggle, and is called “Web page Phishing Detection Dataset”, by Shashwat Tiwari. The dataset consists of 11430 rows of records containing URLs, and 87 columns that make up the features. This is a good dataset based on the fact that there are a large number of records and features to allow for a good breakdown of training and testing data. This will help us avoid either bias or variance while training the model.

This machine learning model will place an emphasis on the special char-

acters, punctuation and https token of the URLs. In order to achieve this, all of the records will be used. The features that will be extracted to train the model are the following: "https_token", "ratio_digits_url", "nb_hyphens", "nb_dots", "nb_underscore", "nb_slash". Some of these features are discrete (0 or 1) and some are continuous values. Features that hold discrete values help describe whether something is present or not in the URL (i.e hyphens), while continuous values quantify a feature of the URL (i.e ratio of digits). The values "phishing" and "legitimate" will be encoded to 1 and 0, respectively. The data is split into a training size of 75 percent, and a dataset for testing of size 25 percent of the entire dataset.

| | https_token | ratio_digits_url | nb_hyphens | nb_dots | nb_underscore | nb_slash | status |
|---|---|---|---|---|---|---|---|
| **0** | 1 | 0.000000 | 0 | 3 | 0 | 3 | legitimate |
| **1** | 1 | 0.220779 | 0 | 1 | 0 | 5 | phishing |
| **2** | 0 | 0.150794 | 1 | 4 | 2 | 5 | phishing |
| **3** | 1 | 0.000000 | 0 | 2 | 0 | 2 | legitimate |
| **4** | 1 | 0.000000 | 2 | 2 | 0 | 5 | legitimate |

## Machine Learning Algorithms

Algorithm #1: KNN-classifier is a popular and easy algorithm to understand in machine learning. KNN-classifier is used for both classification and regression, and this problem involves classification in supervised learning. The k in KNN classifier represents the nearest neighbors. In this algorithm, a data point uses the neighbors' classificaiton to predict its own. This k can be tuned, allowing for greater accuracy in the model. Something like 3 to 5 is usually a good place to start since odds values are recommended. The best of k value will be determined by comparing the trained model's accuracy values per each k. The model will use X as a group of columns/attributes extracted from an url, and a Y column/array that will be the target value [legitimate, phishing].

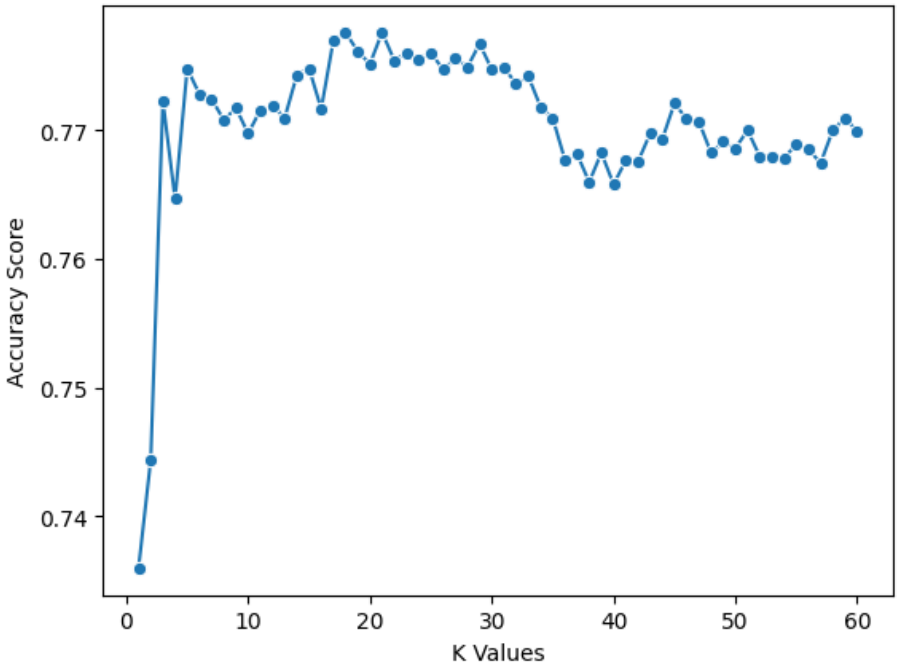
Algorithm #2: Logistic Regression is commonly used for classification tasks such as fraud detection. Logistic regression estimates the probability an instance belongs to a certain binary class. In this project, the logistic regression algorithm will estimate the probability a URL is phishing or legitimate. If the probability that the URL is phishing exceeds 50 percent, the URL will be classified phishing. If this probability is below 50 percent, the URL will be classified as legitimate. In logistic regression, the weighted sum of the input features and bias term is computed and processed through a logistic function to determine the probabilities for a binary classifcation. For the implementation of this algorithm, the following input features will be used to train the data: hyphens, dots, slashes, underscores, ratio of digits, status, and https token. The dataset will be split 75 percent for training and 25 percent for testing. Built-in libraries will be used from sklearn.linear_model to implement logistic regression. Precision, f1-score, and

recall will be calculated so that logistic regression can be compared to other algorithms for classifying spam URLs.

Algorithm #3: Decision Trees are frequently used prediction models in domains such as machine learning, statistics, and data mining. A tree structure also known as a predictive model, where Leaf nodes reflect projected classes and inside nodes represent decisions. The algorithm constructs a decision tree from the top down. The algorithm begins at the root node and makes the best choice recursively and remaining instances are divided into child nodes until there are no more features to select. In this project, the decision tree will distinguish phishing urls from legitimate urls so that the receiver is not harmed by the attackers. For the implementation, the following set of features will be used to train models such as hyphens, dots, slashes, underscores, ratio of digits. For example, the slashes in benign URLs are found to be a 5; if the number of slashes in URLs is greater than 5 then the feature is set to 1 else to 0.

## Implementation Details

One model that will be used is KNN-classifier. To optimize this algorithm, The "best k" value is 18, which is how many neighbors are used in the sklearn.neighbors.KNeighborsClassifier python function. We are collecting the best value of k "best_k", by computing the cross_val_score accuracy for k in range (1, 61) which is shown in the graph below:




Training this model yields: accuracy: 0.79, precision: 0.81, and recall: 0.76.

| URL | Status | Correct Prediction |
|---|---|---|
| https://www.coursera.org/learn/python-machine-learning/lecture/MwsUM/k-nearest-neighbors-classification | Legitimate | Yes |
| https://app.simplenote.com/publish/XhRdVc | Phishing | Yes |
| https://stackoverflow.com/questions/24878174/how-to-count-digits-letters-spaces-for-a-string-in-python | Legitimate | Yes |
| https://www.mjwaldman.com/ | Phishing | Yes |
| https://www.simplilearn.com/tutorials/python-tutorial/python-if-else-statement | Legitimate | Yes |
| https://tubitv.com/category/peliculas_en_espanol | Legitimate | Yes |

The predictions from the manual tests are very satisfiying and outperform expectations considering the accuracy, precision, and recall results.

## Comparison

This section includes the following: 1) comparing the performance of different machine learning algorithms that you used, and 2) comparing the performance of your algorithms with existing solutions if any. Please provide insights to reason about why this algorithm is better/worse than another one.

## Future Directions

This section lays out some potential directions for further improving the performance. You can image what you may do if you were given extra 3-6 months.

## Conclusion

This section summarizes this project, i.e., by the extensive experiments and analysis, do you think the problem is solved well? which algorithm(s) might be better suitable for this problem? Which technique(s) may help further improve the performance?
Last but not the least, don't forget to include references to any work you mentioned in the report.