Approach:

Data preprocessing:

* Random Sampling of URLs: The dataset from <http://data.phishtank.com/data/online-valid.csv> was considered which had thousands of phishing URLs. A margin value of 10,000 phishing URLs and 5000 legitimate URLs was considered to avoid the risk of data imbalance. 5000 samples were picked up randomly from the dataframe.
* Tokenizing the URLs: The 'URLs' column was tokenized from the dataset. 5000 legitimate URLs were saved and used to extract features for training the model.
* Feature Extraction: The following functions were used to extract features from the URLs:
* Domain Extraction: This feature extraction function extracts the domain of the URL. It provides information about the source of the URL.
* Check for IP address (Have\_IP): This function checks if an IP address is present in the URL. This can help identify if the URL is directly linking to an IP address rather than a domain name.
* Check for '@' symbol (Have\_At): This function checks if the '@' symbol is present in the URL. The presence of '@' symbol in the URL can indicate that the URL is not a typical URL and can be suspicious.
* URL Length Categorization (URL\_Length): This function finds the length of the URL and categorizes it. The length of the URL can provide information about the complexity and structure of the URL.
* Number of '/' in URL (URL\_Depth): This function gives the number of '/' present in the URL. The number of '/' in the URL can give an idea about the depth and structure of the URL.
* Check for Redirection '//' (Redirection): This function checks if the redirection '//' is present in the URL. The presence of redirection in the URL can indicate that the user is being redirected to a different website, which can be suspicious.
* Presence of HTTPS Token in Domain Part of URL (https\_Domain): This function checks if the 'HTTPS' token is present in the domain part of the URL. The presence of 'HTTPS' can indicate that the website is secure, but it does not guarantee the authenticity of the website.
* Check for Shortening Services (Tiny\_URL): This function checks if the URL is using a shortening service. The use of shortening services can hide the actual URL and make it difficult to determine the authenticity of the website.
* Prefix/Suffix Separation in Domain (Prefix/Suffix): This function checks if the domain name has a prefix or suffix separated by a '-'. This can indicate that the domain name has been created for a specific purpose and may not be authentic.
* Web Traffic (Web\_Traffic): This feature provides information about the amount of traffic a website is receiving. High traffic to a website can indicate that it is a legitimate website.
* Domain Age (Survival time of domain): This function provides information about the survival time of the domain. It calculates the difference between the termination time and creation time. A domain that has been active for a long time can be considered more authentic.
* Domain End Time (Domain\_End): This function calculates the difference between the termination time and the current time. A domain that is close to termination can be considered less authentic.
* IFrame Redirection (iFrame): This function checks if the URL is using iFrame redirection. The use of iFrame redirection can indicate that the user is being redirected to a different website, which can be suspicious.
* Effect of Mouse Over on Status Bar (Mouse\_Over): This function checks the effect of mouse over on the status bar. The behavior of the status bar can provide information about the authenticity of the website.
* Right Click Attribute Status (Right\_Click): This function checks the status of the right click attribute. The behavior of the right click attribute can provide information about the authenticity of the website.
* Number of Forwardings (Web\_Forwards): This function checks the number of forwardings inTop of Form
* Legitimate Dataset Creation: The features were extracted from the 5000 legitimate URLs using the above-mentioned functions. The returned values of either 1 or 0 were saved in the legitimate dataset.
* Phishing Dataset Creation: The same feature extraction functions were used to extract features from the phishing URL dataset. The returned values of either 1 or 0 were saved in the phishing dataset.
* URL Data Creation: The legitimate and phishing datasets were concatenated to create the urldata.csv file. This file was then used for training the model.
* Significance: The extracted features provided important information about the structure and behavior of the URLs which was crucial in detecting malicious URLs. The feature extraction functions helped in understanding the presence of IP addresses, length of URL, depth of URL, redirection, presence of HTTPS token, and other important parameters.

Machine learning algorithms used:

1. Decision Tree Classifier: Decision tree classifier is a tree-based supervised learning algorithm that builds a decision tree to make predictions. It is significant in the malicious URL detection problem because it can handle complex data relationships and provide interpretable results. The algorithm can be used to determine whether a URL is malicious or not by building a decision tree that predicts the class of a URL based on its features.
2. Random Forest Classifier: Random Forest Classifier is an ensemble learning algorithm that creates multiple decision trees and aggregates their predictions to improve the accuracy and stability of the model. It is significant in the malicious URL detection problem because it can reduce the overfitting problem and provide improved accuracy in comparison to a single decision tree.
3. Multilayer Perceptrons (MLPs): MLPs are feed-forward artificial neural networks that consist of multiple hidden layers between the input and output layers. They are significant in the malicious URL detection problem because they can handle complex non-linear relationships between the input features and the output class. MLPs can be used to learn the complex relationships between the URL features and their class.
4. XGBoost Classifier: XGBoost is a gradient boosting tree algorithm that builds a tree-based model to improve the accuracy of predictions. It is significant in the malicious URL detection problem because it can handle large datasets, provide interpretable results, and improve accuracy by combining the predictions of multiple trees.
5. Autoencoder Neural Network: Autoencoder Neural Networks are unsupervised deep learning algorithms that learn to reconstruct the input data. They are significant in the malicious URL detection problem because they can extract important features from the URL data that can be used to build a predictive model. Autoencoders can be used to pre-process the URL data to reduce the dimensionality of the data and improve the performance of the predictive model.
6. Support Vector Machines: Support Vector Machines are linear classifiers that find the maximum margin hyperplane to separate the data into different classes. They are significant in the malicious URL detection problem because they can handle high-dimensional data and provide robust results in the presence of outliers. SVM can be used to build a predictive model for the malicious URL detection problem by finding the maximum margin hyperplane that separates the URL data into different classes.

