International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 15, No.1 (May-24)

http://dx.doi.org/10.12785/ijcds/1501115

Phishing Website Classification using Machine Learning with Different Datasets

HABIBA BOUIJIJ[1] and AMINE BERQIA[2]

[1,2]*SS Lab, ENSIAS Mohammed V University in Rabat, Rabat, Morocco*

Received 25 Sep. 2023, Revised 25 Mar. 2024, Accepted 29 Mar. 2024, Published 1 May. 2024

Abstract: The classification of phishing websites through the analysis of their URLs is a technique used to enhance the capabilities of systems designed to detect malicious websites. However, the evolution of phishing sites has allowed them to achieve higher levels of sophistication, making proactive detection more complex. The central focus of this article revolves around the exploitation of deep learning models and machine learning techniques with lexical analysis of their URLs to facilitate the classification, detection, and preventive mitigation of phishing websites. Our study includes the evaluation of a selection of commonly castoff machine learning algorithms, specifically Random Forest, K-Nearest Neighbors, Support Vector Machines, Gradient Boosting, Decision Tree, Bagging, AdaBoost and ExtraTree, as well as the deep neural network model. To assess the effectiveness of these algorithms and models, we conduct our analysis using two distinct URL datasets, one from 2016 and the other from 2021. Through lexical analysis, we extract significant features from the URLs and then calculate the accuracy of each algorithm on both datasets. Our results reveal that some algorithms achieve remarkable accuracy scores of up to 0.99 when applied to the 2016 dataset. However, this score decreases to less than 0.91 when applied to the dataset collected in 2021.

Keywords: : Phishing, URL, Classification, Machine Learning, Deep Learning, Dataset, Accuracy metric

1. Introduction

Phishing attacks frequently involve generating fake web sites, known as phishing sites, designed to mimic legitimate websites to deceive individuals into thinking they are engaging with a credible entity. These sites are typically promoted through targeted emails or messages with links leading unsuspecting users to the counterfeit pages. Once there, users are manipulated into revealing personal and sensitive data, inadvertently handing over critical information to perpetrators for illegal purposes, including unauthorized access and identity theft. The critical need to reliably detect such fraudulent sites is vital for protecting internet users against scams, identity theft, and financial loss. As phishing schemes become more complex, imitating genuine websites to mislead both individuals and organizations, improving our detection techniques becomes crucial. This enhancement not only furthers scholarly discussions but also meets an urgent demand across several industries like finance, retail, and healthcare, where the integrity of data and the trust of consumers are paramount. Advances in research to detect phishing sites have both scientific relevance and practical implications, strengthening cybersecurity defenses, reducing the threat of cybercrime, and ensuring a safer digital environment for users [1][2].

The present day has seen the manifestation of the efficacy of algorithms of deep learning (DL) and machine learning (ML) in the fight against phishing, especially with regard to the detection and classification of these from their URLs (Uniform Resources Locator). The central part of the process underlies the fact that there exists an exhaustive dataset annotated with the marking of its legitimate or malicious nature. This dataset forms the very basis for the training and evaluation of deep learning models or machine learning algorithms. This will be meant to help the algorithm find common patterns and, hence, be in a position to correctly categorize new URLs that have not been seen before out in the wild. In other words, the marriage of deep learning and machine learning, applied to the identification of phishing websites, is an intricate dance of data, algorithmic training, and pattern recognition. Continually, as the threat landscape develops, these technologies will remain part and parcel of a better-off cyberspace environment since they identify and prevent deception strategies from being executed by bad actors. However, it has been observed that phishing construction techniques are constantly evolving. Therefore, our research aims to place emphasis on the continuity of the effectiveness of machine learning and deep learning algorithms. Thus subsequently proposing an effective solution to combat phishing attacks in real time.

In this paper, we utilize a set of established Machine Learning algorithms to construct models for detecting phishing URLs. These algorithms have been validated for their effi-

E-mail address: habiba_bouijij@um5.ac.ma, amine.berqia@ensias.um5.ac.ma

cacy in recognizing and thwarting phishing attacks [3]. We conduct a comparative analysis of their performance against a Deep Neural Network (DNN) model, as outlined in our earlier research presented at the 10th International Symposium on Digital Forensics and Security (ISDFS 2022)[4]. Moreover, our assessment involves the utilization of two distinct datasets of URLs: one compiled in 2016 [5], [6] and the other in 2021 [7], providing a comprehensive evaluation framework for our project.

2. Literature Review

A. Phishing

Phishing represents a criminal enterprise that blends social engineering strategies and technical trickery to unlawfully procure individuals' financial account credentials and personal identity data. Social engineering exploits unsuspecting victims by leading them to believe they are engaging with a credible and legitimate entity. This manipulation involves misleading messages and email addresses that guide recipients to fraudulent websites coercing them into disclosing sensitive financial particulars, such as passwords and usernames. In contrast, technical subterfuge tactics entail embedding malware into computer systems, enabling the direct theft of credentials. This often involves systems that intercept individuals' account login information or redirect them to deceptive websites, fostering a multifaceted approach to cybercrime [8], [9].

The ominous presence of phishing poses significant hazards for both individuals and organizations alike. This malicious activity takes on diverse forms, ranging from misleading emails and counterfeit webpages to SMS messages. The ultimate aim of phishing attacks is to discreetly amass valuable personal or professional information, including critical credentials, passwords, financial data, and private customer details. Regrettably, statistical data reveals an alarming uptrend in the prevalence of phishing endeavors [8]. Remarkably, the APWG announced that in 2022, it marked a record year for phishing, with the organization recording over 4.7 million attacks. Starting from 2019, the frequency of phishing attacks has been consistently increasing, experiencing an annual growth rate exceeding 150%. This alarming surge underscores the pressing need for robust and proactive measures to combat phishing's ever-expanding menace, safeguarding both organizations and individuals from the potentially devastating consequences of these insidious attacks.

Figure1, illustrates the growth in the quantity of phishing identified by the APWG [10]:

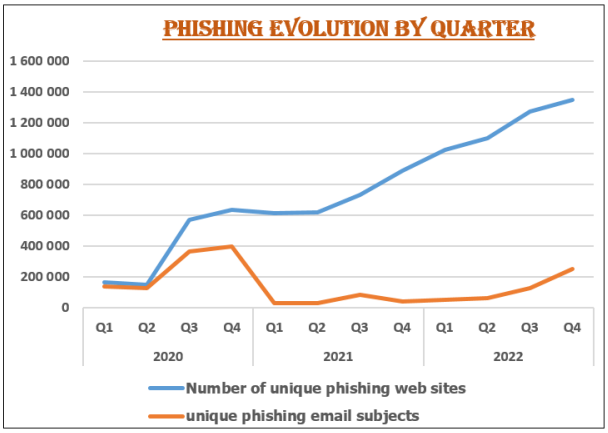

Figure 1. Phishing website activity

B. Phishing Detection: traditional methods

Numerous approaches and methods have been suggested by researchers to address the identification and classification of phishing websites [9],[11] including:

- Blacklist and whitelist approach: URLs can be categorized into two types: blacklists and whitelists. Blacklists consist of databases containing IP addresses, domain names, or URLs associated with malicious websites, serving as warnings for users to avoid such sites. Conversely, whitelists comprise URLs of websites that are known to be safe and trustworthy.

- Heuristic evaluation approach is the in-depth evaluation process where subject matter experts, adopted to measure user interface usability. It is a check of a website's user interface. This is a complete user experience analysis when visiting a website. It helps identify user problems based on predefined design rules called heuristics. Hence the name.

- Visual similarity approach: This method evaluates both suspicious and genuine websites by analyzing diverse visual attributes. As phishing websites often closely resemble their legitimate counterparts, these tools gauge similarities through comparisons involving source code, web page screenshots, text layout, CSS, website logos, and other visual components. Since these techniques rely on comparing the suspicious web page with previously accessed or stored pages, they are unable to identify zero-hour phishing attacks.

Nevertheless, these methods have major weaknesses:

- The administrator should add the website to the blacklist. Damage may have already occurred.

- Trained social professionals can be difficult to find and can be expensive.

- Multiple experts should be used and their results aggregated.

C. Phishing Detection: Machine Learning methods

In recent years, the development of models for detecting phishing URLs has shifted from traditional techniques towards advanced Artificial Intelligence approaches, specifically Deep Learning models and Machine Learning

procedures and algorithms. In other words, deep learning and machine learning have become widely adopted methods for identifying phishing websites. These techniques involve gathering typical features from both malicious and legitimate websites, such as JavaScript attributes, website architecture, URL syntax, and more, to create datasets that will be subsequently used to train and test deep learning and machine learning classifiers[9].

In our prior study titled "Machine Learning Algorithms Evaluation for Phishing URLs Classification"[3], we assessed the performance of eleven well-known Machine Learning algorithms, chosen for their contemporary development and extensive application in the field. These algorithms encompass Decision Tree, K-Nearest Neighbors, Neural Network (MLPerceptron), Logistic Regression, Support Vector Machine, Gradient Boost, AdaBoost, Random Forest, Bagging, Naïve Bayes and ExtraTree. Our findings indicated that the Extra Tree algorithm yielded the most favorable outcome. Furthermore, in a separate study titled "Phishing URL classification using Extra-Tree and DNN"[4], we demonstrated the potential for a Deep Neural Network (DNN) model to achieve an accuracy surpassing 98%.

Researchers [12] introduce a parallel neural joint model algorithm designed to analyze and identify malicious Uniform Resource Locators (URLs). The algorithm's main focus is on extracting both semantic and visual information from these URLs. Initially, a visualization algorithm transforms the URL into a gray image with distinct texture characteristics. Subsequently, the algorithm extracts character and lexical features from the URL, which are then processed using word vector technology. These extracted features are further converted into character embedding vectors and lexical embedding vectors. To achieve this, a parallel joint neural network is utilized, combining a capsule network and an independent recurrent neural network. This combination enables the simultaneous capture of multi-modal vectors encompassing both visual and semantic information. The algorithm also incorporates an attention mechanism in its final layer, which aids in filtering deep features. This integration enhances classification accuracy and facilitates the comprehensive analysis and detection of malicious URLs. They analyzed 33,498 legitimate URLs and 32,519 phishing URLs, achieving an accuracy of 99.78% through the extraction of 115 features.

Authors [13] have developed a phishing detection system engaging numerous Machine Learning algorithms, counting K-star, SMO, Adaboost, Random Forest, Naive Bayes, Decision Tree and KNN (n = 3). They utilized varying numbers and types of structures, encompassing NLP-based features, hybrid features, and word vectors. Enhancing detection accuracy evaluation metric necessitates the creation of an effective feature set, prompting the authors to categorize their features into two distinct classes: NLP-based features, predominantly composed of word vectors and human-determined attributes, which specifically effort on the utilization of words within the URL without engaging in additional operations. Their model is built using 73,575 URLs, consisting of 36,400 legitimate URLs and 37,175 phishing URLs.

Researchers [14] introduce a method called "Search, Heuristic Rule, and Logistic Regression" aimed at proficiently recognizing prevalent obfuscation techniques utilized by phishing websites and elevating the effectiveness of differentiating legitimate websites. This approach comprises three distinct phases. Initially, the title tag content of a webpage is employed as search keywords for the Baidu search engine. If the domain of the webpage matches any of the top-10 search results, it is judged legitimate; if not, further assessment is conducted. Subsequently, if the legitimacy of the web-page remains in question, a comprehensive analysis is carried out using heuristic rules based on character features. This step is designed to pinpoint potential phishing pages. The first two steps collectively contribute to a rapid real-time filtering process for web-pages. Finally, the approach utilizes a logistic regression classifier to assess the remaining pages, with the goal of improving both flexibility and precision in the detection of phishing websites. Experimental findings demonstrate that SHLR effectively identifies 22.9% of phishing web-pages and filters 61.9% of legitimate web-pages using URL lexical information. The SHLR achieves a notable accuracy of 98.9%, showcasing its strong performance in phishing detection.

Authors [15] assessed ML algorithms' effectiveness in detecting malware. They specifically utilized Gradient Boosting and Random Forest algorithms to detect phishing URLs. The experimental results obtained from their methodology underscore the impressive effectiveness of the Machine Learning algorithm (Random Forest) in efficiently managing extensive datasets and precisely determining whether a website is legitimate or malicious. The model achieves a remarkable level of accuracy, reaching 98.6

Authors [16] have developed a thorough prototype that utilizes ML techniques for the detection of Malicious URLs. They introduced an approach that leverages the AdaBoost algorithm. The authors chose AdaBoost due to its versatility, as it can be effectively combined with a range of other machine learning algorithms.

The list is quite comprehensive, as numerous authors [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30] have put forth a wide range of machine learning (ML) algorithms, deep learning (DL) models, and methodologies to identify and categorize phishing URLs. These authors have provided evidence that the accuracy of these algorithms can reach as high as 98%.

In this research, we chose eight ML algorithms known for their significant efficiency in identifying and categorizing phishing websites:

- Random Forest (RF): An ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random forests correct for decision trees' habit of overfitting to their training set.

- K-Nearest Neighbors (KNN): A simple, versatile algorithm used for classification and regression. It assigns the output based on the majority vote or average of the k-nearest neighbors to a point.
- Support Vector Machines (SVM): A powerful classifier that works by finding the hyperplane that best divides a dataset into classes, in the feature space. It is effective in high dimensional spaces and for cases where the number of dimensions exceeds the number of samples.
- Gradient Boosting (GB): An ensemble technique that builds models sequentially, each new model correcting errors made by previously trained trees. It combines weak predictive models to create a strong model.
- Decision Tree (DT): A model that uses a tree-like model of decisions and their possible consequences. It's simple to understand and to interpret but can become complex.
- Bagging (Bag): Stands for Bootstrap Aggregating. It reduces variance and helps to avoid overfitting. Essentially, it combines multiple learners in a way that improves the stability and accuracy of the model.
- AdaBoost (AdaB): Short for Adaptive Boosting, it's a technique used to boost the performance of decision trees on binary classification problems. AdaBoost changes the distribution of weights of incorrectly classified instances and thus forces the learner to focus on difficult cases.
- ExtraTrees (ET): Stands for Extremely Randomized Trees. This ensemble method works similarly to random forests but with random splits of all observations and features. It leads to more diversified trees and reduces variance.

We illustrate in table I the expressions of the classifiers used in this work [31].

TABLE I. Python scripts for ML classifiers

ML algorithm	Python Script
RF	RandomForestClassifier(max_depth=2,random_state=0)
KNN	KNeighborsClassifier(n_neighbors=5)
SVM	SVC(kernel='linear',random_state=0)
GB	GradientBoostingClassifier(random_state=0)
DT	tree.DecisionTreeClassifier()
Bag	BaggingClassifier(base_estimator=SVC(), n_estimators=10, random_state=0)
AdaB	AdaBoostClassifier(n_estimators=100, random_state=0)
ET	ExtraTreesClassifier(n_estimators=100, random_state=0)

Additionally, we included the Deep Neural Network model. A Deep Neural Network (DNN) is an advanced type of neural network that contains multiple layers between the input and output layers. These intermediate, or hidden, layers enable the network to learn complex patterns and relationships in the data by passing inputs through a series of transformations and nonlinear processing. DNNs are particularly effective for tasks such as image recognition, natural language processing, and speech recognition, due to their ability to capture hierarchical patterns in data. The "deep" aspect refers to the presence of multiple hidden layers, which can significantly enhance the model's learning capacity and accuracy, albeit at the cost of increased computational complexity and data requirements.
The DNN model architecture employed in this study features a simple, linear stack of layers, with each layer receiving one input tensor and producing one output tensor. Below is an outline of its key components [32]:

- Input Layer (Implicitly Defined): The model begins with an implicitly defined input layer. The first dense layer's input dimensionality must match the feature dimensionality of your dataset but isn't explicitly mentioned in the summary.
- First Dense Layer: This is a fully connected layer with 256 neurons (or units). It takes the input data and performs a weighted sum across the inputs for each neuron and then applies an activation function. The parameter count (5,888) comes from the weights between the input features and the neurons, plus bias terms for each neuron.
- Dropout Layer: This layer randomly sets a fraction of the input units to 0 at each update during training time, which helps prevent overfitting. The fraction rate is not provided, but there are no trainable parameters in a dropout layer (0).
- Second Dense Layer: Another fully connected layer, this time with 128 neurons. It processes the output from the previous layer (after dropout). The parameter count (32,896) includes the weights from each of the 256 inputs to the 128 neurons, plus 128 bias terms.
- Third Dense Layer: This layer has 64 neurons and functions similarly to the previous dense layers, taking the output from the second dense layer as its input. The parameter count (8,256) is derived from the 128 inputs to each of the 64 neurons, plus 64 biases.
- Fourth Dense Layer (Output Layer): The final layer is a dense layer with a single neuron (1), often used for binary classification tasks. The model output is a single value representing the prediction. The 65 parameters come from the 64 inputs plus one bias term.

3. Methodology and Evaluation

URL classification is a key strategy in the ongoing fight against malicious websites, presenting an essential binary classification task that differentiates between "phishing" and "benign" websites. This challenge has prompted the development of various techniques and approaches, with a special emphasis on those employing Deep Learning

(DL) and Machine Learning (ML) algorithms for their effectiveness in addressing complex problems.

In this research, we undertake a thorough investigation using two separate datasets as our experimental foundation. These datasets have been carefully chosen to enable a detailed evaluation of the performance of a range of Machine Learning algorithms, which have been validated for their efficacy in previous studies. The set of algorithms we examine includes notable options such as Random Forest (RF), K-Nearest Neighbors (KNN), Support Vector Machines (SVM), Gradient Boosting (GB), Decision Tree (DT), Bagging (B), AdaBoost (AB), and ExtraTrees (ET).

Our goal with this in-depth study is to dissect the performance nuances of these ML algorithms and to extract valuable insights into their capabilities for precise URL classification. This research marks a significant step forward in our ongoing effort to advance cybersecurity defenses through the application of advanced computational methods.

Achieving this goal, we choose to use Jupyter Notebook (web-based interactive computing platform), the Python programming language and Scikit-learn (open-source Machine Learning library) [31] as our tools for implementation. Additionally, we applied lexical analysis as an approach to extract URLs features to make data input. In the end, Cross-validation and data pre-processing techniques were used prior to training and testing the algorithms used in this work.

In Figure2, we will describe the process of our evaluation of the selected algorithms and the DNN model created.

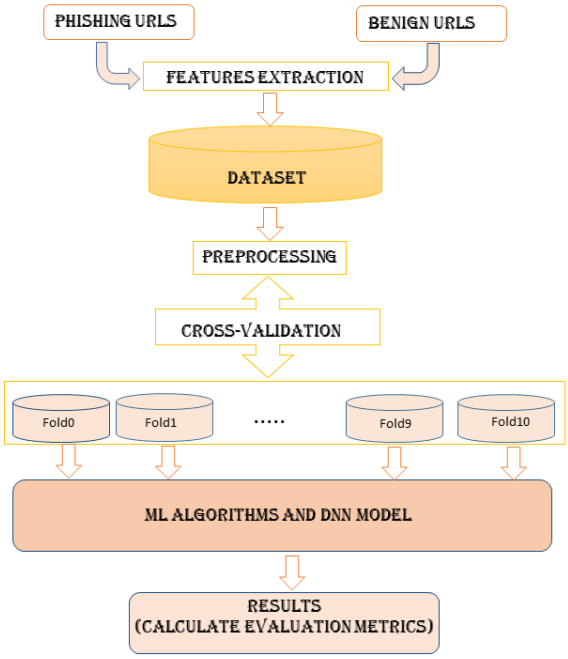

Figure 2. Evaluation Process for ML Algorithms and DNN Model

A. *Data and features extraction*

Machine learning operates on the principle of assimilating distinct attributes from one dataset and subsequently applying those insights to evaluate another dataset. Hence, the process hinges on the indispensable task of data collection, ensuring the acquisition of pertinent features essential for this cognitive process. In the context of our project, we undertake the compilation of two distinct datasets, originating from disparate sources. The first source, Mendeley Data [7], furnishes URLs amassed during the years 2016 and 2021. Concurrently, the second source, the Canadian Institute for Cybersecurity Homepage [6], contributes URLs procured in the year 2016.

Employing a lexical analysis technique, we extracted attributes from the URLs with the aim of delving into the characters and components employed in crafting these web addresses. Multiple researchers [33], [34], [35], [36] have put forth diverse methodologies to extract features specific to URLs. These distinctive attributes offer a viable means to create input data, subsequently employed in the training and assessment of algorithms engineered for the purpose of URL classification. In this paper, we utilize the functions listed in table II:

TABLE II. Functions employed for extracting URL features

Function	Definition
Protocol	check if the used protocol is https
url_len	Total of characters used
Digit_count	Total of digits used
Dot_count	Total of '.' Used
Hypen_count	Total of '-' used
Double_slash_count	Total of '//' used
Single_slash_count	Total of '/' used
Special_characters$_{count}$	Total of (':', ';', '#', '!', '%', '+', '_', '', '=', '&',', ']', '[')
At_sign count	Total of '@' used
Adress_ip	Domain Name is IP address?

B. *Data preprocessing*

To ensure that the selected data is appropriate for analysis, we applied a data preprocessing technique designed to convert it into a format that produces the best possible results and improves the accuracy of our model. Data preprocessing entails a series of activities involving cleansing, transforming, and preparing the data prior to its utilization in machine learning algorithms. These essential steps serve to ensure that the data is not only correctly structured for machine learning processes but is also devoid of errors, inconsistencies, or extraneous information. This preparatory phase holds paramount importance in the machine learning journey, as it contributes significantly to the enhancement of the model's performance and accuracy [3],[31].

Our approach specifically entails employing the Scaling technique, a process that involves the rescaling of numeric attributes with real-valued data to a standardized range of 0 to 1. This method is pivotal in ensuring that the data adheres to a consistent and manageable scale, facilitating a more effective comparison and analysis. To ensure our data adhered to a standardized format, we utilized the MinMaxScaler class, a component of the widely adopted scikit-learn Python library [31]. This careful normalization ensures that

our data is appropriately scaled, minimizing any potential discrepancies that might arise due to variations in the ranges of different attributes. As a result, our preprocessing efforts contribute significantly to creating a robust foundation for subsequent machine learning procedures, ultimately culminating in more accurate and reliable outcomes.

C. Cross validation

It is a widely employed technique in the realm of machine learning projects, offering a robust approach to evaluating the performance of a model. This method involves training the model on distinct subsets of the dataset and subsequently testing it on the remaining data. Various cross-validation strategies are utilized, with one prominent example being k-fold cross-validation. This consist to divide dataset into k subgroups, and the model is trained and tested k times. Each iteration employs a different subset as the testing data, and the overall model performance is then averaged across all iterations. The primary utility of cross-validation lies in its capacity to mitigate the risks of overfitting and aid in the selection of the optimal model from a pool of alternatives.

In the context of our study, we adopted the "Repeated Stratified K-Fold Cross-Validation" technique. This method leverages the benefits of both Stratified K-fold and repeated cross-validation. The Stratified K-fold approach ensures that the distribution of samples across different classes remains balanced across all folds, thereby averting any bias that might arise due to imbalanced class distributions. Repeated Stratified K-Fold Cross-Validation further extends this concept by repeatedly applying the Stratified K-fold technique, thereby producing multiple independent estimates of the model's performance. This iterative process aids in obtaining a more stable and trustworthy evaluation of the model's generalization capabilities, while also mitigating the risk of overfitting [3],[31]. Through the utilization of the Repeated Stratified K-Fold Cross-Validation methodology, our study adopts a rigorous and comprehensive approach to evaluating the performance of our model. This technique not only ensures a robust estimation of the model's capabilities but also enhances the overall reliability and credibility of our findings.

D. Evaluation metric

In this research paper, the evaluation metric of choice centers around the accuracy score, a pivotal measure with applicability across both Machine Learning algorithms and Deep Learning models. This metric serves as a yardstick to gauge the efficacy of these computational frameworks, quantifying the proportion of accurate predictions rendered by the model in relation to the total volume of predictions undertaken. The accuracy score, an essential performance indicator, is computed by divining the count of correct predictions by the overall count of predictions made. To illustrate, in a scenario where a model conducts 100 predictions and accurately forecasts 80 of them, the corresponding accuracy score stands at 0.8, equivalent to 80%. It is calculated using the formula:

$$accuracy = \frac{Numberof CorrectPredictions}{TotalNumberof Predictions} \quad (1)$$

The accuracy score, renowned for its widespread adoption, proves especially fitting for classification tasks involving well-balanced datasets. Within the context of this project, we operationalized the accuracy_score() function, a key component housed within the sklearn.metrics library. By leveraging this function, we were able to systematically quantify the accuracy of our model's predictions, thereby providing an objective means to assess its performance. The accuracy score, though widely employed, remains an integral tool in the arsenal of evaluation metrics, aiding in the discernment of a model's proficiency in making precise predictions within the realm of classification tasks [31],[37].

E. Algorithms Implementation

For our research, we selected a computer that operates on the 64-bit version of the Windows 10 operating system. This machine is configured with an Intel(R) Core(TM) i5-8350U CPU running at 1.70GHz, 1.90GHz, and is equipped with 16 GB of RAM. For our web-based interactive computing needs, we employed JupyterNotebook. We fully leveraged the capabilities of the Sklearn and Keras libraries [31],[32] to implement the ML classifiers and the DNN model. Figure3 illustrates the evaluation algorithm in a general manner (The Python scripts for the classifiers were defined in TableI). For DNN model, we used the script mentioned in Figure4.

- **Start** the algorithm.
- **Import** the necessary libraries:
- **ML classifier** from **sklearn**
- **accuracy_score** from **sklearn.metrics**.
- **Initialize** the **MLClassifier** with the parameters:
- **Fit** the classifier on the training data **X_train** and **y_train**.
- **Predict** the target values for the test set **X_test** using the trained classifier, storing the predictions in **y_pred**.
- **Calculate** the accuracy of the predictions by comparing **y_pred** with the true target values **y_test** using **accuracy_score**.
- **Return** the calculated accuracy.
- **End** the algorithm.

Figure 3. ML classifier implementation

4. Results and Discussion

This section offers an in-depth exploration of the outcomes achieved and the in-depth discussions sparked by the implementation of the algorithms and methodologies outlined in the preceding section. Our study placed a central emphasis on meticulously evaluating and drawing comparisons between the performances of these algorithms, with accuracy as the primary evaluation metric. The algorithms under scrutiny encompassed a diverse range, including Random Forest, K-Nearest Neighbors, Support Vector

```
def get_DNN_results():
 from tensorflow.keras.models import Sequential
 from tensorflow.keras.layers import Dense, Dropout
 from sklearn.metrics import accuracy_score
 model=Sequential()
 model.add(Dense(256,input_dim=22,activation='relu'))
 model.add(Dropout(0.2))
 model.add(Dense(128,activation='relu'))
 model.add(Dense(64,activation='relu'))
 model.add(Dense(1,activation='sigmoid'))
          model.compile(loss='binary_crossentropy',
          optimizer='adam', metrics=['accuracy'])
 model.fit(X_train, y_train, epochs=100, batch_size=100)
 y_pred = model.predict(X_test)> 0.5
 accuracy = accuracy_score(y_test,y_pred)
 return accuracy
```

Figure 4. DNN model implementation

Machines, Gradient Boosting, Decision Tree, Bagging, AdaBoost, and ExtraTree and Deep Neural Network (DNN). Our evaluation was conducted within the context of crafting an effective model to combat malicious website attacks, a critical endeavor in the realm of cybersecurity. To accomplish this, we rigorously adhered to a structured process that involved the collection of data, thorough data preprocessing, and the development of intricate models. It's noteworthy that we leveraged two distinct databases—one collected in 2016 and another in 2021—as part of our evaluation process. These databases served as critical resources, allowing us to comprehensively assess the algorithms' performance across different time-frames and potentially reveal evolving trends in phishing attack patterns.

A. *Dataset 1: URLs collected in 2016*

In Table III, we provide an expanded overview of the descriptions derived from the feature extraction process applied to URLs gathered in the year 2016. From the

TABLE III. Description of features extraction for Data2016

Features	Phishing 2016			Benign 2016		
	Mean	Min	Max	Mean	Min	Max
https_*protocol*	0.9689	0	1	0.9454	0	1
dot_count	3.9137	1	26	1.3764	1	9
url_len	96.3208	21	378	86.9450	83	92
digit_count	12.2762	0	144	9.0029	0	41
:_count	1.0929	0	7	1.0121	1	4
;_count	0.1666	0	23	0.0003	0	1
_count	0.0022	0	2	0.0000	0	0
!_count	0.0025	0	3	0.0007	0	3
%_count	1.7146	0	72	1.0660	0	23
?_count	0.3394	0	6	0.1283	0	2
=_count	0.7419	0	19	0.2293	0	11
@_count	0.0036	0	2	0.0032	0	1
double_slash	1.0707	0	3	1.0089	1	2
single_slash	3.9735	0	16	4.0842	1	15
adress_ip	0.8604	0	1	1.0000	1	1
short_url	0.0010	0	1	0.0000	0	0

findings presented in this table, it's evident that conducting a lexical analysis of URL structures provides a clear basis for distinguishing between phishing and legitimate websites. For instance, phishing websites often use URLs that are longer on average compared to those of legitimate sites. This difference suggests that phishing URLs may include additional misleading paths and parameters to disorient or deceive users. These extracted features enable us to train and test the implemented ML algorithms or DNN model.
Table IV and Figure 5 offer a comprehensive overview of the outcomes derived from the analysis of our initial dataset, encompassing URLs originating from as far back as 2016. The data for this dataset was meticulously sourced from the Canadian Institute for Cybersecurity Webpage [6] and the resource rich Mendeley Data Webpage [7]. Remarkably, the average accuracy scores attained by both the ExtraTree algorithm and the DNN model significantly transcend the ordinary threshold, eclipsing an impressive 99%.

TABLE IV. Results of Accuracy score calculated with URLs collected in 2016

ML algorithms	Fold0	Fold1	Fold2	Fold3	Fold4	Fold5	Fold6	Fold7	Fold8	Fold9	Accuracy Average
DT	98,70%	98,85%	98,63%	98,95%	98,58%	98,80%	98,85%	98,68%	99,03%	98,83%	98,79%
KNN	98,33%	98,03%	98,23%	98,13%	97,65%	97,63%	98,23%	98,20%	98,33%	98,33%	98,11%
GB	98,25%	98,30%	98,25%	98,43%	98,08%	98,18%	98,23%	98,05%	98,20%	98,35%	98,23%
RF	95,63%	95,45%	95,25%	95,38%	93,43%	95,05%	95,08%	95,25%	94,98%	95,28%	95,08%
SVM	91,85%	92,18%	92,60%	92,08%	92,03%	92,50%	92,38%	91,33%	91,95%	92,58%	92,15%
ET	99,03%	99,00%	99,33%	99,25%	98,90%	99,00%	99,10%	99,18%	99,33%	99,18%	99,13%
AdaB	98,10%	98,05%	98,13%	98,33%	98,00%	97,93%	98,10%	98,05%	98,08%	98,45%	98,12%
Bag	93,73%	93,78%	93,78%	93,80%	93,55%	93,63%	94,18%	93,00%	93,88%	93,98%	93,73%
DNN	98,72%	99,12%	99,12%	99,00%	99,25%	99,35%	99,45%	99,12%	99,62%	99,34%	99,21%

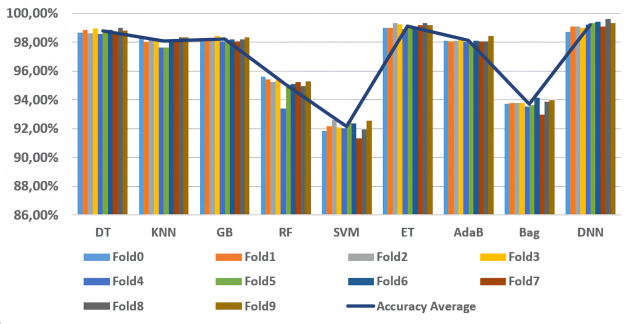

Figure 5. Plot of results(Data 2016

B. *Dataset 2: URLs collected in 2021*

In this part, we will present the findings from the analysis and classification of data collected in 2021:(see tableV) We observe that phishing attempts consistently employ a higher number of digits and a greater variety of special characters in the construction of URLs. This pattern suggests that attackers are increasingly using complex combinations of characters to mimic legitimate websites more convincingly, thereby increasing the likelihood of deceiving unsuspecting users.
The insights presented in Table VI and Figure 6 provide a succinct encapsulation of the findings stemming from the examination of the second dataset, comprising URLs generated in the year 2021 and sourced exclusively from the Mendeley Data Webpage [7]. Here, the average accuracy scores for the array of Machine Learning algorithms range commendably from 81% to 88%. However, it is noteworthy that the DNN model emerges as a standout performer,

TABLE V. Description of features extraction for Data2021

Features	Phishing 2021			Benign 2021		
	Mean	Min	Max	Mean	Min	Max
https_protocol	0.4203	0	1	0.1023	0	1
dot_count	2.5900	1	30	2.1305	1	19
url_len	72.1272	15	1200	59.4475	14	1024
digit_count	9.8159	0	689	2.3677	0	209
:_count	1.0250	1	6	1.0100	1	8
;_count	0.1765	0	30	0.0000	0	0
_count	0.0058	0	1	0.0039	0	1
!_count	0.0006	0	1	0.0010	0	2
%_count	0.1368	0	35	0.1405	0	105
?_count	0.2196	0	4	0.0636	0	2
=_count	0.4660	0	18	0.1191	0	14
@_count	0.0323	0	4	0.0003	0	1
double_slash	1.0147	1	6	1.0010	1	2
single_slash	2.5294	0	28	2.3461	0	14
adress_ip	0.5518	0	1	0.0171	0	1
short_url	0.0026	0	1	0.0076	0	1

attaining an extraordinary accuracy score of 90%. These

TABLE VI. Results of Accuracy score calculated with URLs collected in 2021

ML algorithms	Fold0	Fold1	Fold2	Fold3	Fold4	Fold5	Fold6	Fold7	Fold8	Fold9	Accuracy Average
DT	85,80%	86,68%	85,43%	86,08%	85,93%	85,15%	86,08%	87,18%	86,90%	86,18%	86,14%
KNN	87,85%	87,63%	87,25%	87,80%	87,53%	86,78%	87,45%	87,85%	87,40%	88,08%	87,56%
GB	87,70%	88,18%	88,13%	87,45%	88,03%	87,23%	87,63%	88,33%	88,43%	87,93%	87,90%
RF	81,45%	81,58%	81,80%	81,38%	81,33%	80,53%	80,68%	82,30%	81,73%	81,98%	81,47%
SVM	82,00%	81,58%	81,80%	81,50%	81,38%	80,40%	81,38%	82,15%	81,50%	83,18%	81,69%
ET	87,85%	88,33%	87,38%	88,40%	88,38%	86,63%	88,00%	89,10%	88,35%	88,08%	88,05%
AdaB	86,55%	86,58%	86,35%	86,10%	86,50%	86,18%	86,63%	87,38%	86,53%	86,65%	86,55%
Bag	84,15%	83,55%	83,23%	83,38%	83,43%	82,28%	83,38%	83,80%	83,85%	84,70%	83,58%
DNN	88,92%	89,35%	89,35%	90,35%	90,55%	89,95%	91,07%	91,77%	91,27%	91,12%	90,37%

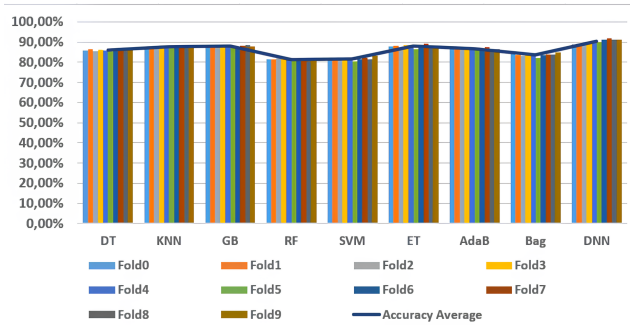

Figure 6. Plot of Results (Data 2021)

observations suggest that phishing URLs exhibit specific evolving characteristics, likely adapting as users become more aware and security technologies advance. Despite this evolution, the effectiveness of using HTTPS as a distinguishing feature seems to have diminished in recent years, presenting an intriguing area for further exploration. Additionally, outliers observed in the 2021 phishing data, such as extreme values for URL length and digit count, might represent either statistical anomalies or emerging phishing strategies, indicating important considerations for refining phishing detection models.

In our analysis, the ExtraTree algorithm and the Deep Neural Network (DNN) model have shown exceptional proficiency in identifying phishing URLs from our initial data set, achieving average accuracy rates exceeding the impressive 99% mark. This trend continues with the DNN model as we examine a second data set, where it significantly outperforms traditional Machine Learning algorithms, which recorded accuracy rates between 81% and 88%. The consistent success of the DNN model highlights its superior capability in detecting phishing URLs across different periods, confirming its effectiveness in this domain. Furthermore, the standout performance of the ExtraTree algorithm in the first dataset underscores its viability as a powerful tool in phishing detection.

As we approach the conclusion of this study, we are poised to present a condensed and cohesive overview of the outcomes we have achieved. This will be succinctly depicted in TableVII where our results will be juxtaposed with the findings of studies elucidated in the ”Literature Review” section.

TABLE VII. Comparative analysis of obtained results and selected findings from previous studies

Reference of paper	paper Date	Total URLs	URLs Date	Algorithms and methods	Features Extraction	Accuracy
[13]	2019	73,557	2017	DT	NLP features	97.02%
					Word Vector	82.48%
					Hybrid	95.14%
				Adaboost	NLP features	93.24%
					Word Vector	74.74%
					Hybrid	92.53%
				Kstar	NLP features	93.56%
					Word Vector	81.05%
					Hybrid	95.27%
				KNN(k=3)	NLP features	95.67%
					Word Vector	83.01%
					Hybrid	95.86%
				RF	NLP features	97.98%
					Word Vector	83.14%
					Hybrid	96.36%
				SMO	NLP features	94.92%
					Word Vector	82.71%
					Hybrid	94.48%
				NB	NLP features	95.67%
					Word Vector	83.01%
					Hybrid	95.86%
[17]	2023	24,000	2022	SVC	Domain Name Character	94%
[33]	2021	45,343	2016	SVC	Lexical analysis	98%
				LR		93%
				k-NN		98%
				NB		79%
				RF		99%
This paper	2023	20,000	2016	Decision Tree	Lexical analysis	98,79%
				KNN		98,11%
				GB		98,23%
				RF		81,48%
				SVM		92,15%
				ET		99,13%
				AdaB		98,12%
				Bag		93,73%
				DNN		99,21%
This paper	2023	20,000	2021	Decision Tree	Lexical analysis	86,14%
				KNN		87,56%
				GB		87,90%
				RF		81,48%
				SVM		81,69%
				ET		88,05%
				AdaB		86,55%
				Bag		83,58%
				DNN		90,37%

This comparative analysis aims to provide a holistic panorama of our research contributions and their resonance with the existing scholarly landscape. Through this systematic assessment, our intention is to spotlight the distinctive strengths and innovative dimensions inherent in our proposed methodology, contextualized within the broader tapestry of prior investigations. By drawing upon these juxtapositions, we can effectively gauge the salience of our discoveries, pinpoint avenues of progression, and underscore the potential ramifications of our endeavors

within the sphere of phishing detection and classification.

5. Conclusion and Future Work

In this research, we conducted an extensive investigation into a variety of machine learning (ML) algorithms and a deep neural network (DNN) model, aimed at enhancing the identification and classification of phishing URLs. Our goal was to create a highly effective model that could accurately distinguish even the most intricate and deceptive phishing websites. Our strategy involved a detailed lexical analysis to extract critical features from the URLs under study. By analyzing lexical elements such as domain names, subdomains, and path structures, we aimed to identify key indicators for accurately detecting and classifying phishing attempts. This method allowed us to explore deeper into the characteristics of these URLs, identifying subtle signs and patterns indicative of malicious intent. In our comprehensive investigation, we meticulously analyzed a suite of prominent machine learning (ML) algorithms, encompassing Random Forest (RF), K-Nearest Neighbors (KNN), Support Vector Machines (SVM), Gradient Boosting (GB), Decision Tree (DT), Bagging (B), AdaBoost (AdaB), and ExtraTree (ET), in conjunction with a cutting-edge Deep Neural Network (DNN) model. An integral component of our study involved the calculation of accuracy scores for these algorithms, which played a crucial role in evaluating their effectiveness in the identification and classification of phishing URLs. This metric facilitated a detailed comparison and enabled us to extract profound insights into the comparative efficacy of the traditional ML algorithms versus the advanced DNN model.

The results, detailed in the fourth section of our research, indicated a significant trend towards the escalating complexity and intricacy involved in phishing URL classification. A highlight of our findings was the superior performance of the DNN model, which showcased extraordinary accuracy levels, achieving an astounding average accuracy of 99% for the first dataset and a commendable 90% for the second. These figures not only reflect the DNN model's proficiency in recognizing the nuanced features of phishing URLs but also its potential to revolutionize the field.Moving forward, our research trajectory is strongly influenced by these key findings. We are committed to furthering our exploration and innovation in developing sophisticated models capable of handling complex URLs with greater precision, focusing on detecting the most subtle cues of phishing attempts. We plan to enrich our model with advanced techniques, including URL HTML Encoding, the WHOIS analysis method, the Tiny URL technique, and an innovative voting mechanism. By seamlessly integrating these methodologies into our existing framework, we aim to significantly refine our phishing URL detection capabilities. Our goal is to elevate the standards of accuracy and efficiency in phishing detection, paving the way for groundbreaking advancements in cybersecurity measures.

References

[1] H. Abroshan, J. Devos, G. Poels, and E. Laermans, "Phishing happens beyond technology: The effects of human behaviors and demographics on each step of a phishing process," *IEEE Access*, vol. 9, pp. 44 928–44 949, 2021.

[2] M. Jari, "An overview of phishing victimization: Human factors, training and the role of emotions." Academy and Industry Research Collaboration Center (AIRCC), 7 2022, pp. 217–228.

[3] H. Bouijij and A. Berqia, "Machine learning algorithms evaluation for phishing urls classification," 2021, pp. 1–5.

[4] H. Bouijij, A. Berqia, and H. Saliah-Hassan, "Phishing url classification using extra-tree and dnn," 2022, pp. 1–6.

[5] R. Mohammad, "Phishing websites datast," vol. 1, 2017.

[6] "Canadian institute for cybersecurity homepage." [Online]. Available: https://www.unb.ca/cic/datasets/url-2016.html

[7] "Mendeley data homepage." [Online]. Available: https://data.mendeley.com/datasets/n96ncsr5g4/1

[8] "Apwg — phishing activity trends reports." [Online]. Available: https://apwg.org/trendsreports/

[9] A. Safi and S. Singh, "A systematic literature review on phishing website detection techniques," *Journal of King Saud University - Computer and Information Sciences*, vol. 35, pp. 590–611, 2 2023.

[10] "Apwg homepage." [Online]. Available: https://apwg.org/

[11] A. Berqia and G. Nacsimento, "A distributed approach for intrusion detection systems," *Proceedings. 2004 International Conference on Information and Communication Technologies: From Theory to Applications, 2004.*, pp. 493–494, 2004.

[12] J. Yuan, G. Chen, S. Tian, and X. Pei, "Malicious url detection based on a parallel neural joint model," *IEEE Access*, vol. 9, pp. 9464–9472, 2021.

[13] O. K. Sahingoz, E. Buber, O. Demir, and B. Diri, "Machine learning based phishing detection from urls," *Expert Systems with Applications*, vol. 117, pp. 345–357, 3 2019.

[14] Y. Ding, N. Luktarhan, K. Li, and W. Slamu, "A keyword-based combination approach for detecting phishing webpages," *Computers and Security*, vol. 84, pp. 256–275, 7 2019.

[15] F. O. Catak, K. Sahinbas, and V. Dörtkardeş, "Malicious url detection using machine learning," pp. 160–180, 8 2021. [Online]. Available: http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-7998-5101-1.ch008

[16] F. Khan, J. Ahamed, S. Kadry, and L. K. Ramasamy, "Detecting malicious urls using binary classification through ada boost algorithm," *International Journal of Electrical and Computer Engineering*, vol. 10, pp. 997–1005, 2020.

[17] J. Zhou, H. Cui, X. Li, W. Yang, and X. Wu, "A novel phishing website detection model based on lightgbm and domain name features," *Symmetry*, vol. 15, p. 180, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:255635272

[18] A. K. Kassem, "Intelligent system using machine learning techniques for security assessment and cyber intrusion detection." [Online]. Available: https://theses.hal.science/tel-03522384

[19] A. Sanap, D. K. Gaikwad, S. Randhave, N. Salunke, and D. J. V. Shinde, "Smart phishing website detection using cnn algorithm," 2022.

[20] S. G. Abbas, I. Vaccari, F. Hussain, S. Zahid, U. U. Fayyaz, G. A. Shah, T. Bakhshi, and E. Cambiaso, "Identifying and mitigating phishing attack threats in iot use cases using a threat modelling approach," *Sensors*, vol. 21, 7 2021.

[21] S. Poddar, H. Salkar, P. Agarwal, D. C. Karia, M. Paraye, D. D. Ambawade, and N. Bhagat, "Phishguard - an automatic web phishing detection system," *2022 IEEE India Council International Subsections Conference (INDISCON)*, pp. 1–7, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:251773323

[22] N. Q. Do, A. Selamat, O. Krejcar, E. Herrera-Viedma, and H. Fujita, "Deep learning for phishing detection: Taxonomy, current challenges and future directions," pp. 36 429–36 463, 2022.

[23] W. Wei, Q. Ke, J. Nowak, M. Korytkowski, R. Scherer, and M. Woźniak, "Accurate and fast url phishing detector: A convolutional neural network approach," *Computer Networks*, vol. 178, 9 2020.

[24] H. Salah and H. Zuhair, "Deep learning in phishing mitigation: a uniform resource locator-based predictive model," *International Journal of Electrical and Computer Engineering*, vol. 13, pp. 3227–3243, 6 2023.

[25] A. Kumar, J. M. Chatterjee, and V. G. Díaz, "A novel hybrid approach of svm combined with nlp and probabilistic neural network for email phishing," *International Journal of Electrical and Computer Engineering*, vol. 10, pp. 486–493, 2020.

[26] R. E. Yoro, F. O. Aghware, M. I. Akazue, A. E. Ibor, and A. A. Ojugo, "Evidence of personality traits on phishing attack menace among selected university undergraduates in nigerian," *International Journal of Electrical and Computer Engineering*, vol. 13, pp. 1943–1953, 4 2023.

[27] D. J. Liu, G. G. Geng, and X. C. Zhang, "Multi-scale semantic deep fusion models for phishing website detection," *Expert Systems with Applications*, vol. 209, 12 2022.

[28] M. Sánchez-Paniagua, E. Fidalgo, E. Alegre, and R. Alaiz-Rodríguez, "Phishing websites detection using a novel multipurpose dataset and web technologies features," *Expert Systems with Applications*, vol. 207, 11 2022.

[29] P. Pavan Kumar, T. Jaya, and V. Rajendran, "Si-bba – a novel phishing website detection based on swarm intelligence with deep learning," *Materials Today: Proceedings*, vol. 80, pp. 3129–3139, 2023, sI:5 NANO 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214785321050379

[30] A. Maci, A. Santorsola, A. Coscia, and A. Iannacone, "Unbalanced web phishing classification through deep reinforcement learning," *Computers*, vol. 12, p. 118, 6 2023.

[31] "Scikit-learn homepage." [Online]. Available: https://scikit-learn.org/stable/supervised_learning.html#supervised-learning

[32] "Keras homepage." [Online]. Available: https://keras.io/guides/

[33] A. Saleem Raja, R. Vinodini, and A. Kavitha, "Lexical features based malicious url detection using machine learning techniques," *Materials Today: Proceedings*, vol. 47, pp. 163–166, 2021, nCRABE. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214785321028947

[34] B. B. Gupta, K. Yadav, I. Razzak, K. Psannis, A. Castiglione, and X. Chang, "A novel approach for phishing urls detection using lexical based machine learning in a real-time environment," vol. 175, pp. 47–57. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0140366421001675

[35] P. Drotár, M. Gazda, and L. Vokorokos, "Ensemble feature selection using election methods and ranker clustering," *Information Sciences*, vol. 480, pp. 365–380, 4 2019.

[36] A. Aljofey, Q. Jiang, A. Rasool, H. Chen, W. Liu, Q. Qu, and Y. Wang, "An effective detection approach for phishing websites using url and html features," *Scientific Reports*, vol. 12, 12 2022.

[37] "Analyticsvidhya homepage." [Online]. Available: https://www.analyticsvidhya.com/blog/2021/07/metrics-to-evaluate-your-classification-model-to-take-the-right-decisions/

Habiba BOUIJIJ is a PhD student at the Smart Systems Laboratory at ENSIAS - Mohamed V University in Rabat, Morocco. She is also a Senior high school teacher of computer science. Additionally, she serves as a Trainer in Information and Communication Technology (ICT) at the Regional Center for Education and Training Professions in Khemisset, Morocco. She has also worked as a major supervisor for on-going training in Python for secondary school computer science teachers and specialized Microsoft Office training.

AMINE BERQIA Dr. Amine BERQIA is Full Professor at ENSIAS, University Mohmmed V in Rabat, Morocco. He is Head of Communication Networks Department. He teaches computer sciences, communication networks, operating systems, security, networks security, wireless and mobile networks. He is a member of the teaching team of the engineering degree in information systems security. Dr. Amine BERQIA is member of Rabat IT Center and Smart Systems Research Lab. He holds a Ph.D. degree in Computer Sciences from the University of Dijon, France. He was Assistant Professor at University of Geneva and Coordinator of the Swiss Virtual Campus Project VITELS from 2000 to 2003. He published more than 50 papers in journals and conferences in the areas of Networks performance, Security and New Generation of learning. In 2012 Dr. Amine BERQIA was awarded by IEEE Education Society in recognition and appreciation for his valued services and contributions as 2012 IEEE Educon conference Organizing Committee Chair. In 2013 Dr. Amine BERQIA was awarded by iNEER. In 2019, Dr. Amine BERQIA was awarded by IEEE Standards Association for his valued services and contributions for the development of IEEE Standard 1876..