Scientific Research Publishing

Journal of Computer and Communications, 2024, 12, 91-115
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

Artificial Intelligence in Cybersecurity to Detect Phishing

Dominique Wasso Kiseki[1], Vincent Havyarimana[2], Désiré Lumonge Zabagunda[3], Walumbuka Ilundu Wail[4], Therence Niyonsaba[5]

[1]Department of Computer Engineering, University of Burundi (UB), Bujumbura, Burundi
[2]Ecole Normale Supérieure (ENS), Bujumbura, Burundi
[3]University Research Laboratory in Modeling and Applied Statistical Engineering (LURMISTA), Nyamugerera, Bujumbura, Burundi
[4]Department of Physics and Technology, ISP (Institut Supérieur Pédagogique), TTC (Teachers' Training College), Bukavu, Democratic Republic of the Congo
[5]Department of Management Computer Sciences, ISP (Institut Supérieur Pédagogique), TTC (Teachers' Training College), Bukavu, Democratic Republic of the Congo
Email: wassokisekidom@gmail.com, havincent12@gmail.com, ilunduwail@gmail.com, desirelumonge02@gmail.com

How to cite this paper: Kiseki, D.W., Havyarimana, V., Zabagunda, D.L., Wail, W.I. and Niyonsaba, T. (2024) Artificial Intelligence in Cybersecurity to Detect Phishing. *Journal of Computer and Communications*, **12**, 91-115.
https://doi.org/10.4236/jcc.2024.1212007

Received: July 25, 2024
Accepted: December 22, 2024
Published: December 25, 2024

Abstract

Phishing is one of the most common threats on the Internet. Traditionally, detection methods have relied on blacklists and heuristic rules, but these approaches are showing their limitations in the face of rapidly evolving attack techniques. Artificial Intelligence (AI) offers promising solutions for improving phishing detection, prediction and prevention. In our study, we analyzed three supervised machine learning classifiers and one deep learning classifier for detecting and predicting phishing websites: Naive Bayes, Decision Tree, Gradient Boosting and Multi-Layer Perceptron. The results showed that the Gradient Boosting Classifier performed best, with a precision of 96.2%, a F1-score of 96.6%, recall and precision of 99.9% in all classes, and a mean absolute error (MAE) of just 0.002. Closely followed by the Gradient Boosting Classifier with a precision of 96.2% and a score of 96.6%. In contrast, Naive Bayes and the Decision Tree showed a lower accuracy rate. These results underline the importance of high accuracy in these models to reduce the risk associated with malicious attachments and reinforce security measures in this area of research.

Keywords

Artificial Intelligence, Machine Learning, Deep Learning, Cybersecurity, Phishing, Detection, Algorithm, Supervised Learning

1. Introduction

The cybersecurity sector is no exception to the technological revolution. It is

confronted with the rise of Artificial Intelligence (AI), which can analyse metadata in record time. This is enabling us to make a major leap forward in a number of areas, particularly cybersecurity [1]. Phishing is a social engineering technique designed to obtain confidential information by posing as a trusted entity [2]. Phishing attacks [3] have increased exponentially, affecting individuals and organisations alike, according to reports and studies on cyber security published by the Anti-Phishing Working Group (APWG). This APWG report for the fourth quarter of 2023, detailed in **Figure 1**, observed almost five million phishing attacks, or 4,987,809. These attacks constitute the worst phishing year on record and are still considered to be a record phishing year [4]. Piter Cassidy (published on 14 April 2024).

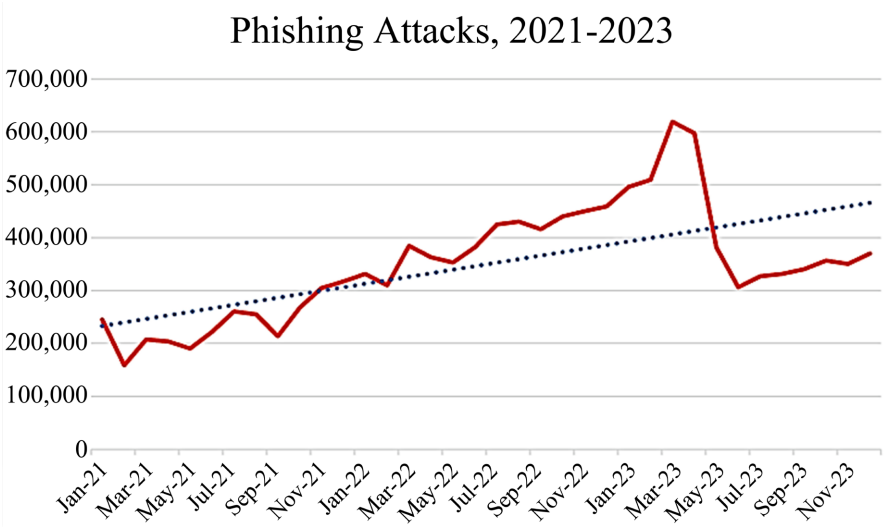

Figure 1. APWG report on trends in phishing activity in Q4 2023.

Traditional phishing detection methods, such as blacklists, are struggling to keep pace with the rapid evolution of phishing techniques. Other approaches have already been used to filter phishing sites, sometimes applicable at different stages of the attack flow. These include network protection, authentication, user education and classifiers. Artificial intelligence would be an indispensable aid in tackling new cyber-attacks in all their forms, using machine-learning algorithms [5]. Phishing is a common attack technique used by cybercriminals to trick users into divulging confidential information such as login credentials, passwords and financial information, in order to compromise their activity on the platform [6].

This study focuses on the growing adoption of artificial intelligence in the field of cybersecurity. It is conducted particularly in the context of phishing detection. It explores the various Machine Learning techniques used for phishing detection, such as supervised learning algorithms [7]. Artificial intelligence and Machine Learning (ML) techniques have shown significant potential for improving the detection and prevention of phishing attacks [8]. This capability enables proactive detection of attacks, even when phishing techniques are constantly evolving to bypass traditional defenses.

Several studies have focused on analyzing the contribution of artificial intelligence in phishing detection. Yang *et al.* [9] proposed a multidimensional phishing detection approach based on deep learning in multidimensional features by combining URL statistical features. On the other hand, authors of [10]-[12], in their studies, went with a single classifier while analysis in order to better exploit the potentiality offered by supervised machine learning with all its algorithms and models as is the case for our study. Authors of [5] [6] [9] [12]-[19], AI is playing an increasingly important role in cybersecurity, they discussed a way to identify the URL of a given website is a phishing site or not by going through two popular machine learning algorithms such as SVM and random forest. Authors of [20], [21] compared the results of several machine learning methods for predicting phishing websites and achieved very good performance by assembling the Random Forest and XGBoost classifiers, a process we considered time-consuming.

Furthermore, these aforementioned studies praise the applicability of AI in cybersecurity without particularly exploiting supervised machine learning algorithms combined with deep learning, which is our hobbyhorse. We are focusing on regression problems where the prediction involves a continuous value, and the models are trained on labelled data in order to accurately predict the results for new instances. The objective of this study is to analyse 3 classifiers used in machine learning and 1 in deep learning for the detection and prediction of phishing websites in AI such as Naive Bayes, Decision Tree, Gradient Boosting and Multi-Layer Perceptron. Compare these models after study and analysis in order to identify their limitations and strengths. Draw and find a more effective model to propose and guide future perspectives in this area of research.

After this introduction, our paper proceeds with a theoretical framework of phishing techniques, followed by phishing detection hardware and methods, description of the dataset, experimental results, limitations and future prospects, contribution and finally conclusion.

2. Theoretical Framework of Phishing Technics

2.1. Theoretical Framework

In this section, we will define and describe phishing and URL, phishing anatomy and structure, examine some known phishing techniques and even those used by criminals to deceive people. To date, numerous techniques have been introduced to eliminate phishing attacks and keep users safe online. Spoofed e-mails and false URLs remain difficult to detect and unavoidable [22].

2.2. What Is Phishing?

Phishing is a type of cybercrime in which a digital attacker impersonates a trusted entity to obtain sensitive information. Phishing can target individuals for banking credentials or credit card data, or could also target organizations through their employees. Deloitte estimates that a whopping 91% of cyber-attacks begin with a

phishing email. Phishing is a major problem for organizations trying to stay safe, as employees are often targeted as a means of infiltrating the business. With so much at stake, it really pays to understand the biggest phishing attacks[1]. Phishing is based on social engineering. Its whole premise is to build up a degree of trust with the victim so that they feel comfortable providing information. This could involve the hacker posing as a colleague, manager or technology support. Once the victim is convinced that they are speaking with a known entity, the stage is set for the attack.

2.3. Anatomy of URL Phishing

A URL (Uniform Resource Locator) is a type of uniform resource identifier (URI) used to access information from remote computers, such as a web server or cloud storage space. It contains various elements, including the network communication protocol, a sub-domain, a domain name and its extension. URL phishing is a fraudulent activity that involves diverting people to a fake website where they download malware or expose their confidential information. Before understanding how attackers proceed when creating a phishing domain, we'll introduce the structure of the URL. Figure 2 shows that a URL is made up of several distinct parts[2].

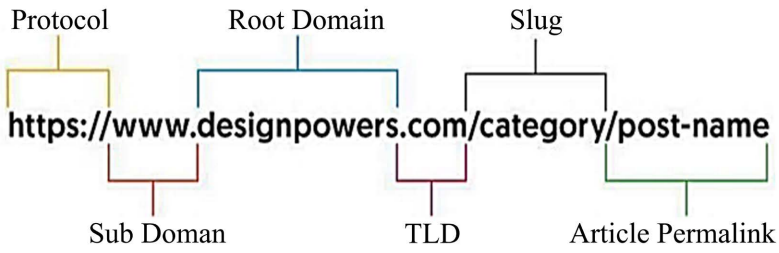

Figure 2. The URL structure [23].

2.4. Phishing Awareness Training and Software

To combat the ever-changing nature of phishing attacks, organizations and individuals can benefit from phishing awareness training programs and software solutions. These tools inform users about the latest phishing techniques, provide simulated phishing scenarios to test their awareness, and offer real-time protection against malicious e-mails. Investing in such resources can significantly improve a person's ability to identify and avoid falling victim to phishing scams.

2.5. Phishing Techniques

We take a look at some well-known phishing methods used by criminals to trick people: Fraudulent e-mails, Fraudulent websites, Fraudulent text and SMS messages, Phishing attacks on social networks, Fraudulent phone calls, Targeted phishing (spear phishing), Fake tech support fraud.

[1]https://www.webopedia.com/definitions/phishing-meaning/ [consulted June 2024].

[2]https://designpowers.com/blog/url-best-practices [viewed in June 2024].

3. Phishing Detection Materials, Method and Techniques

This section highlights the materials and methods used in this study to achieve the assigned objectives, but also goes on to describe and present the data set used for the experiments and outlines the algorithms used in our study. Machine learning as a field of Artificial Intelligence relies on mathematical and statistical approaches to provide simplified and efficient methods for data analysis [24]. It can automate repetitive operations, analyze large data sets, facilitate more informed and accurate decision-making, personalize customer experiences and optimize operational processes to increase profitability [25]. The aim of machine learning is to find flexible patterns in data and for specific tasks such as predictions, phishing detection. ML algorithms can be broadly classified into three types as summarized in **Figure 3**, namely supervised for labeled observations, unsupervised for unlabeled observations and reinforcement learning for models that learn from errors to improve accuracy [26].

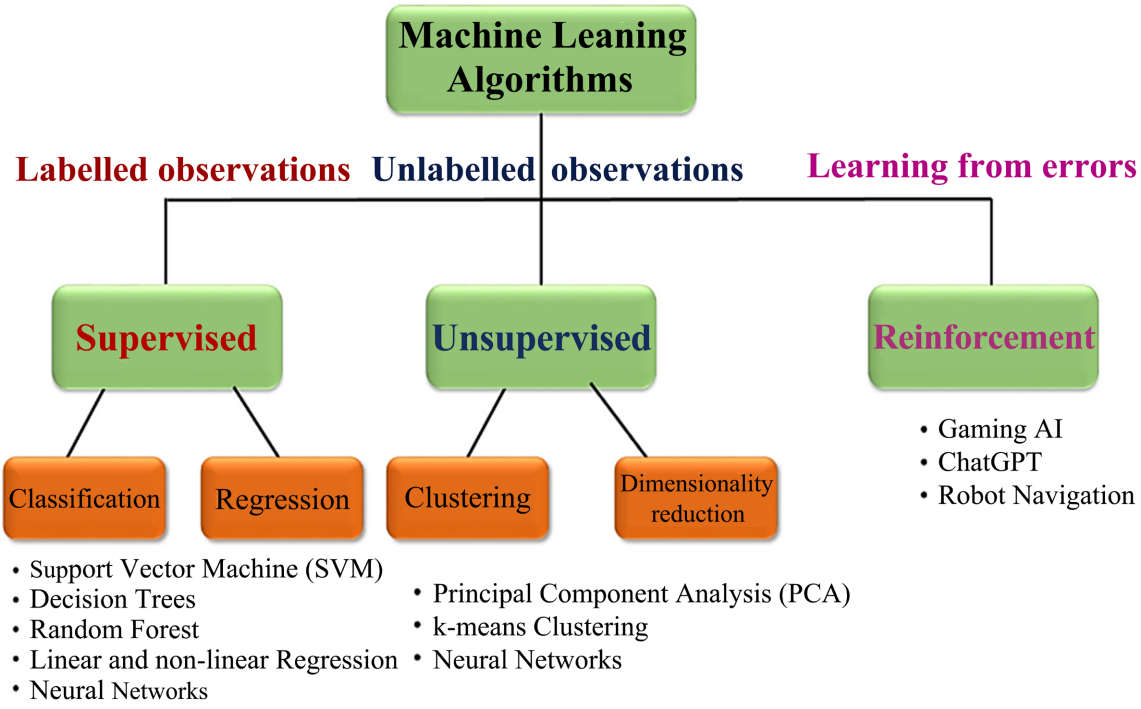

Figure 3. Flow chart illustrating the classification of different machine learning algorithms into supervised, unsupervised and reinforcement models [26].

3.1. Rationale for Model Selection

This study focuses on four algorithms and models, combining both machine and deep learning. The choice of these models is based on their respective capabilities: 1) Naïve Bayes is based on Bayes' theorem, assumes conditional independence between features, simple and fast to train, effective for classifying textual data, such as spam e-mail detection, and can work with a small training dataset. 2) Decision trees widely used for classification and regression, easy to interpret, handle

numerical and categorical data, visually understandable and help explain model decisions and adaptability to mixed data (numerical and categorical). 3) Powerful Gradient Boosting for regression and classification, combines several weak decision trees to form a robust model. With high accuracy by aggregating weak models, but also adaptable to complex data and non-linear interactions. 4) Multi-Layer Perceptron (MLP) Deep neural network, learns complex representations from data, capable of capturing non-linear patterns, models complex relationships between features. It is often used for classification, regression and other tasks.

This choice of models covers a wide spectrum of approaches, from the simplest and most interpretable (Naïve Bayes, decision trees) to the most complex and powerful (Gradient Boosting, MLP). This has enabled us to study the trade-offs between performance, interpretability and adaptation to new phishing threats.

3.2. Description and Representation on the Data Set

Most of the machine learning models examined here fall into the category of supervised machine learning. To update our dataset with new phishing sites, we also implemented code that extracts features of new phishing sites provided by the Kaggle website collected from the well-known public repository Kaggle.com[3]. In addition, each sample is tagged with a class identifier indicating whether it is categorized as a phishing website (1) or not (−1). An overview of the data set reveals a total of 11,054 samples, each composed of 32 features. Of these, 31 are independent and 1 dependent. All features are of integer type, avoiding the need for the Label Encoder transformation. No outliers are detected in the dataset [27]. There are no missing values in the dataset. The algorithm for the methodology used in this study comprises the following steps: Data acquisition and preparation: Loading of the dataset containing instances of suspected phishing websites. Performing exploratory data analysis (EDA) to better understand the structure and content of the dataset. Data visualization: Use of visualization techniques to illustrate patterns and trends within the dataset. Data pre-processing and splitting: Pre-processing of data to ensure compatibility with ML algorithms, including feature scaling and encoding of categorical variables. Partitioning of the dataset into training and test sets to facilitate model evaluation. Model training: Implementation of various supervised machine learning algorithms, and neural networks, to train predictive models. Model comparison: Evaluation of the performance of trained models using appropriate measures, such as accuracy, precision, recall and F1 score. Comparison of the effectiveness of different ML algorithms in accurately classifying phishing websites. Each website is marked as either legitimate or phishing.

3.3. The Algorithms Used

Supervised machine learning, a common technique, is used to predict outcomes

[3]https://www.kaggle.com/datasets/taruntiwarihp/phishing-site-urls [consulted on May 26, 2024 in Bukavu].

from a given set of features, using example feature-label pairs. In regression problems such as ours, where prediction involves a continuous value, models are trained on labeled data to accurately predict outcomes for new instances. The regression models selected are:

3.3.1. Decision Trees

The decision tree is a non-parametric supervised learning method used for classification and regression tasks. It divides the feature space into segments to make predictions. They are among the most versatile supervised learning models, and have the important property of being easy to interpret [28]. A decision tree is, as shown in **Figure 4**, a binary tree-like data structure that is used to make a decision. Trees are a very intuitive way of displaying and analyzing data, and are commonly used even outside the field of machine learning. A decision tree helps individuals make better decisions through a tree graph or modeling of alternatives and their possible implications, such as likely outcomes, resource costs and utility [29]. With the ability to predict both categorical and real values, as well as the ability to integrate numerical and categorical data without any normalization or creation of indicator variables, it's not hard to see why they are a popular choice for machine learning.

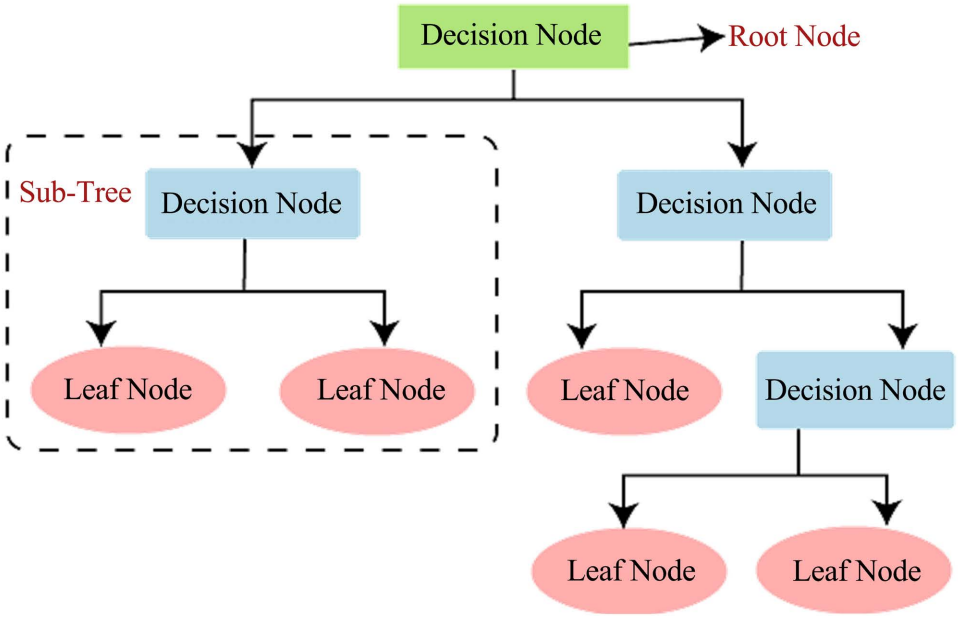

Figure 4. Presentation of the decision tree classifier[4].

3.3.2. Gradient Boosting Classifier

The Gradient Boosting Classifier is a powerful ensemble learning algorithm that combines several weak learners to create a strong predictive model. It sequentially trains a series of decision trees, each correcting the errors of its predecessors,

[4](https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm [consulted on June 12, 2024])

ultimately producing a highly accurate model. It belongs to the family of machine learning methods based on ensemble learning. It is a boosting algorithm, *i.e.* it builds a predictive model as a weighted combination of simpler models, usually decision trees [19]. Here are the main steps of the algorithm: Initialization: The model starts with a very simple decision tree, which makes poor predictions. Boosting iterations: At each iteration, a new decision tree is added to the model. This tree is trained to predict the residual errors (difference between true values and predictions) of the previous model. Model update: The model is updated by combining the new small decision tree with the previous trees, assigning greater weight to those trees that reduce the error the most [30]. Stop: The process stops when a maximum number of iterations is reached, or when a stop criterion is met (for example, when performance improvement is deemed sufficient).

3.3.3. Naive Bayes

The Naive Bayes algorithm is a supervised learning algorithm based on Bayes' theorem and used to solve classification problems. It is mainly used for text classification involving a high-dimensional training data set. The Naïve Bayes classifier is one of the simplest and most effective classification algorithms, making it possible to build fast machine learning models capable of making rapid predictions [31]. This is a probabilistic classifier, which means it predicts based on the probability of an object. Some popular examples of the Naïve Bayes algorithm are spam filtering, sentimental analysis and article classification [32].

3.3.4. Multi-Layer Perceptron Classification

The MLP is a type of artificial neural network composed of several layers of nodes, capable of learning complex patterns in data. It is a type of artificial neural network used, as shown in **Figure 5**, for classification and regression[5]. It consists of several layers of interconnected neurons: 1) Input layer: receives the problem's input variables (features), the number of neurons in this layer corresponds to the number of input variables. 2) Hidden layer(s): these intermediate layers apply non-linear transformations to the input data. The number of hidden layers and neurons per layer is chosen by the user during model design, and each neuron in a hidden layer calculates a weighted linear combination of its inputs, then applies a non-linear activation function (such as the sigmoid or ReLU function). 3) Output layer: produces the model's final predictions; the number of neurons in this layer depends on the problem, and the activation function used in this layer also depends on the problem. The main advantages of the multilayer perceptron are[6]:

- Its ability to approximate complex functions thanks to its hidden layers.
- Flexibility to handle different types of problem (classification, regression).
- Ts robustness to noise and incomplete data.

However, MLP requires careful tuning of its hyperparameters (number of layers, neurons, learning rate, etc.) to achieve good performance.

[5]https://fr.wikipedia.org/wiki/Perceptron_multicouche [consulté le 08 Juillet 2024]).
[6]Idem.

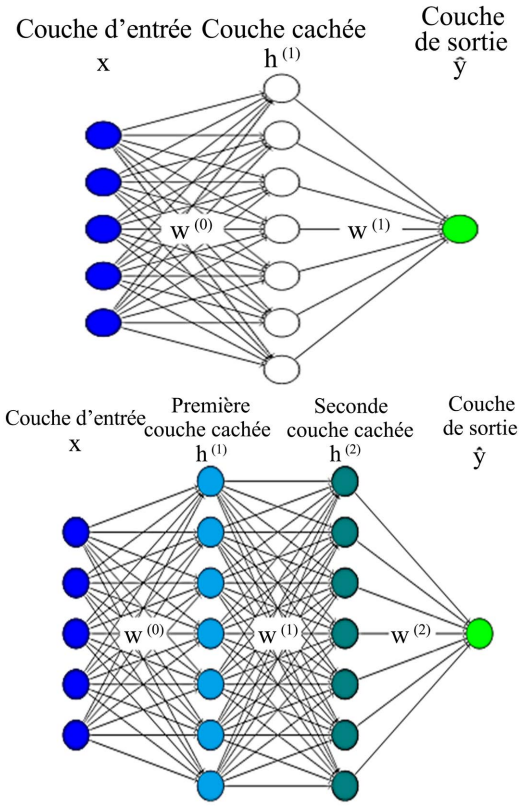

Figure 5. Multilayer perceptron classifier[7].

3.4. User Behavior Analysis

Some systems combine analysis of e-mail content with tracking of user behavior (browsing history, past interactions, etc.) to improve phishing detection. The characteristics of our data set are as follows:

1) Index: Unique identifier for each line (sample) of data;

2) UsingIP: Indicates whether the IP address is used in the URL (1 for yes, −1 for no;

3) LongURL: Indicates whether the URL is long (1 for yes, −1 for no); a fundamental measure often used to distinguish legitimate from malicious URLs. Long, convoluted URLs can indicate attempts to obscure their true intent;

4) ShortURL: Indicates whether the URL is short (1 for yes, −1 for no);

5) Symbol@: Indicates whether the "@" symbol is present in the URL (1 for yes, −1 for no);

6) Redirecting//: Indicates whether the URL contains "//" redirects (1 for yes, −1 for no);

7) PrefixSuffix-: Indicates whether the URL contains prefixes or suffixes with hyphens "-" (1 for yes, −1 for no);

8) SubDomains: Indicates the number of subdomains in the URL (1 for many, −1 for few);

9) HTTPS: Indicates whether HTTPS protocol is used (1 for yes, −1 for no);

[7]https://rtavenar.github.io/deep_book/fr/content/fr/mlp.html [consulted in July 8, 2024].

10) DomainRegLen: Length of domain record (1 for long, −1 for short);

11) Favicon: Indicates whether the favicon is loaded from the same domain (1 for yes, −1 for no);

12) NonStdPort: Indicates whether a non-standard port is used (1 for yes, −1 for no);

13) HTTPSDomainURL: Indicates whether the domain URL uses HTTPS (1 for yes, −1 for no);

14) RequestURL: Indicates whether the majority of URLs on the page point to the same domain (1 for yes, −1 for no);

15) AnchorURL: Indicates whether the URLs of anchor tags point to the same domain (1 for yes, −1 for no);

16) LinksInScriptTags: Indicates presence of links in script tags (1 for yes, −1 for no);

17) ServerFormHandler: Indicates whether the page form is managed by the server (1 for yes, −1 for no);

18) InfoEmail: Indicates whether an email address is present in the URL (1 for yes, −1 for no);

19) AbnormalURL: Indicates whether the URL is abnormal in relation to the domain (1 for yes, −1 for no);

20) WebsiteForwarding: Indicates whether the site frequently redirects to other pages (1 for yes, −1 for no);

21) StatusBarCust: Indicates whether the status bar is customized (1 for yes, −1 for no);

22) DisableRightClick: Indicates whether right-click is disabled on the page (1 for yes, −1 for no);

23) UsingPopupWindow: Indicates whether popup windows are used (1 for yes, −1 for no);

24) IframeRedirection: Indicates whether redirection iframes are used (1 for yes, −1 for no);

25) AgeofDomain: Indicates domain age (1 for old, −1 for young);

26) DNSRecording: Indicates whether DNS records are available (1 for yes, −1 for no);

27) WebsiteTraffic: Indicates website traffic level (1 for high, −1 for low);

28) PageRank: Indicates page PageRank (1 for high, −1 for low);

29) GoogleIndex: Indicates whether the page is indexed by Google (1 for yes, −1 for no);

30) LinksPointingToPage: Indicates the number of links pointing to the page (1 for many, −1 for few);

31) StatsReport: Indicates the presence of statistical reports (1 for yes, −1 for no);

32) class: The target variable, indicating the class to which each sample belongs (1 for one class, −1 for the other class, typically representing labels such as “phishing” or “legitimate”).

4. Results and Discussion

In this section, we present the results obtained from our analyses and experiments with data analysis and processing, dataset manipulation using different Machine Learning algorithms, discussion of the results, limitations and future prospects.

4.1. Presentation of Results

4.1.1. Description of the Dataset

Table 1. Dataset description.

Features	Mean	std
UsingIP	0.313914	0.949495
LongURL	−0.633345	0.765973
ShortURL	0.738737	0.674024
Symbol@	0.700561	0.713625
Redirecting//	0.741632	0.670837
PrefixSuffix-	−0.734938	0.678165
SubDomains	0.064049	0.817492
HTTPS	0.251040	0.911856
DomainRegLen	−0.336711	0.941651
Favicon	0.628551	0.777804
NonStdPort	0.728243	0.685350
HTTPSDomainURL	0.675231	0.737640
RequestURL	0.186720	0.982458
AnchorURL	−0.076443	0.715116
LinksInScriptTags	−0.118238	0.763933
ServerFormHandler	−0.595712	0.759168
InfoEmail	0.635788	0.771899
AbnormalURL	0.705446	0.708796
WebsiteForwarding	0.115705	0.319885
StatusBarCust	0.762077	0.647516
DisableRightClick	0.913877	0.406009
UsingPopupWindow	0.613353	0.789845
IframeRedirection	0.816899	0.576807
AgeofDomain	0.061335	0.998162
DNSRecording	0.377239	0.926158
WebsiteTraffic	0.287407	0.827680
PageRank	−0.483626	0.875314
GoogleIndex	0.721549	0.692395
LinksPointingToPage	0.343948	0.569936
StatsReport	0.719739	0.694276
class	0.113986	0.993527

4.1.2. The Feature Correlation Matrix

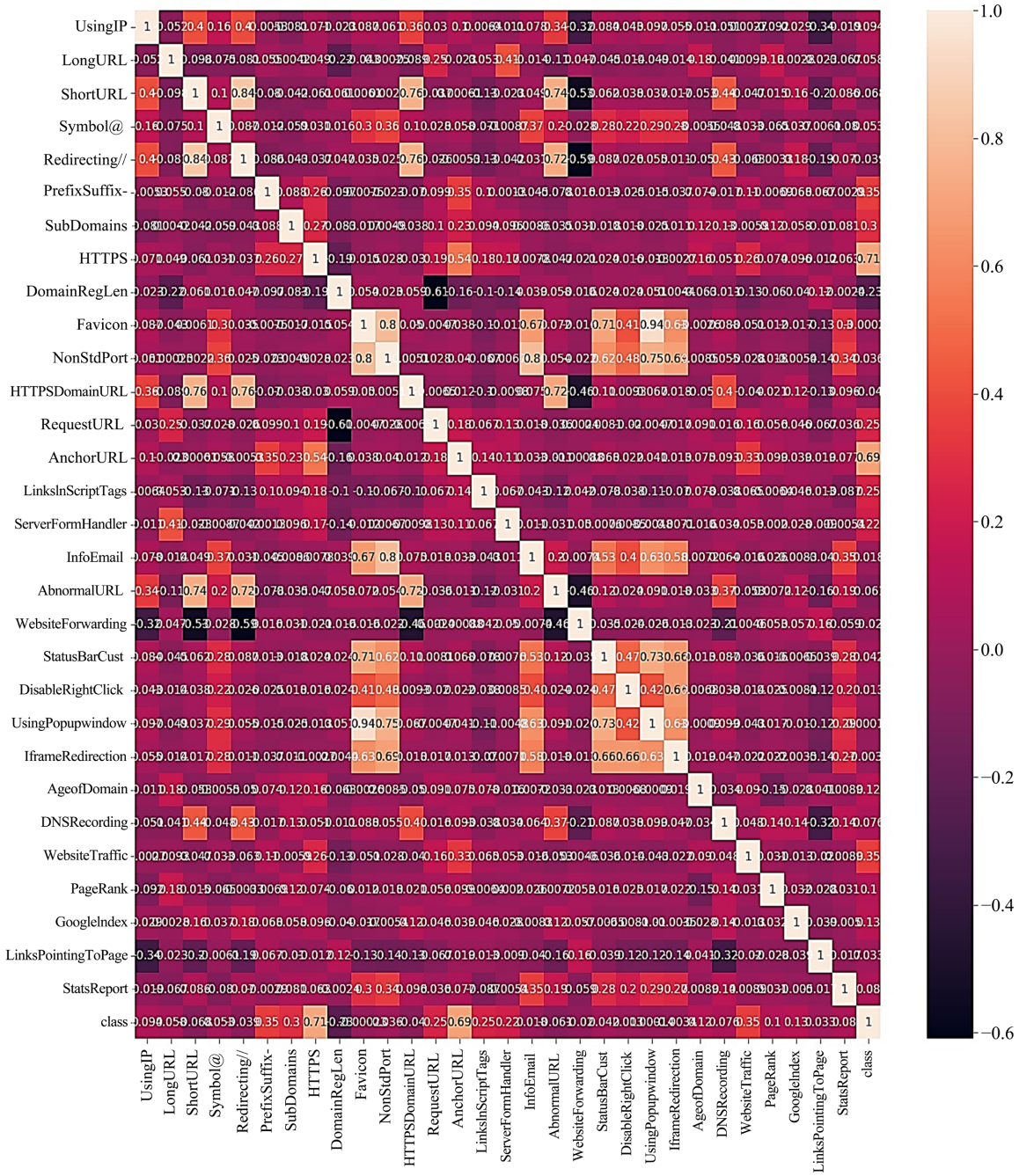

Figure 6. Feature correlation matrix.

Figure 6 is a correlation matrix between different features of a dataset of phishing URLs from our study. Correlation measures the strength and direction of the linear relationship between two variables. Values range from −1 to 1, where: 1 indicates a perfect positive correlation (when one variable increases, the other also increases). −1 indicates a perfect negative correlation (when one variable increases, the other decreases). 0 indicates no linear correlation. These correlations

were used in Figure 6 to identify and detect the most important features when detecting phishing URLs, and helped us to improve the classification models used to detect these threats.

4.1.3. Characteristic Histogram

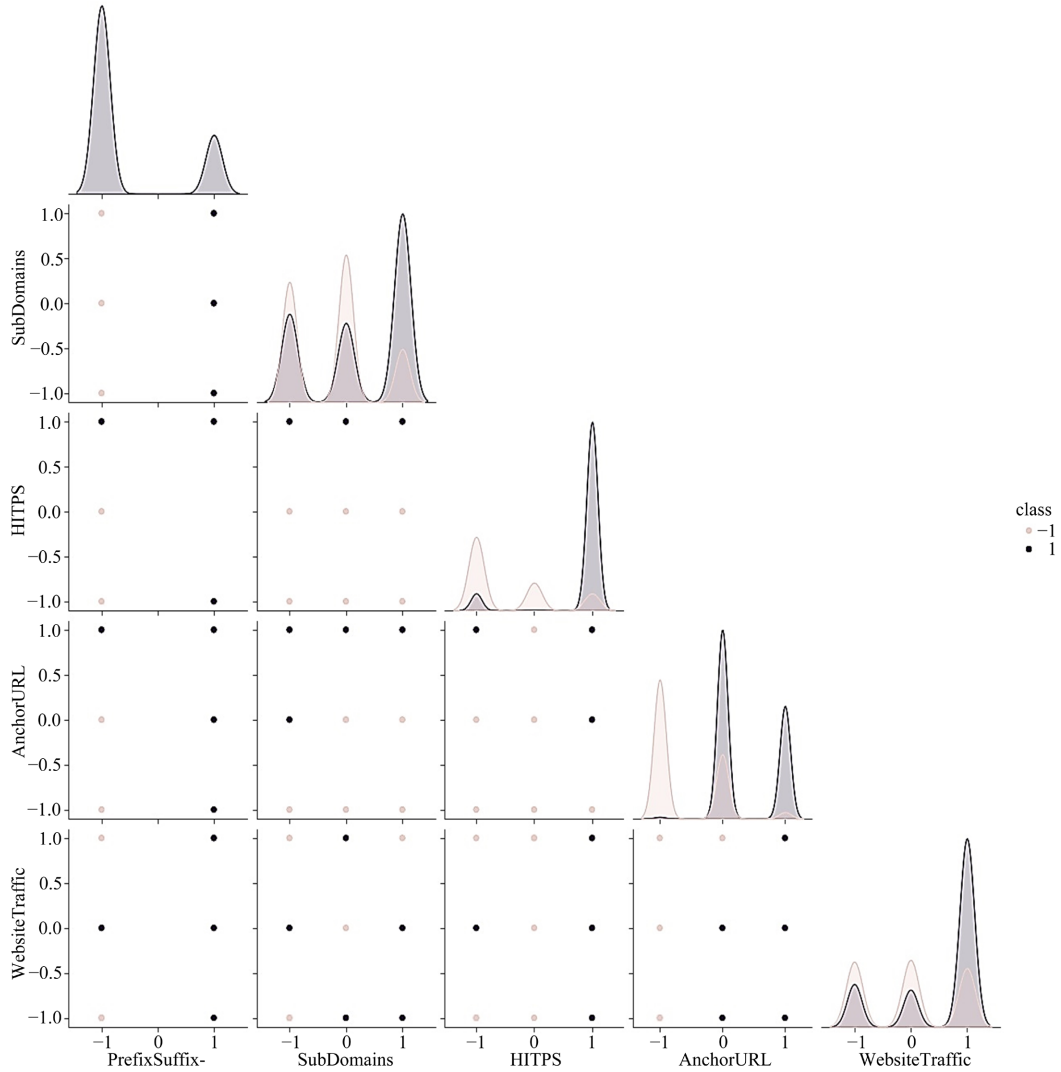

Figure 7. Feature histograms.

This Figure 7 shows histograms of various characteristics extracted from cybersecurity-related data. Each histogram shows the distribution of values for two distinct classes, represented by the blue and red dots. These histograms in Figure 7 allow us to analyze the discriminability of these different features for classification between the two classes. Features with more separated distributions (such as WebsiteTraffic and HTTPS) appear to be more informative in distinguishing the two classes. The graphs show that features related to sub-domains, HTTPS usage and web traffic are the most relevant for differentiating these two classes, while prefixes/suffixes and URL anchors seem less discriminating.

4.1.4. Diagram Representing the Number of Phishing Cases with the Values of Two Label Classes

With these source codes we have:

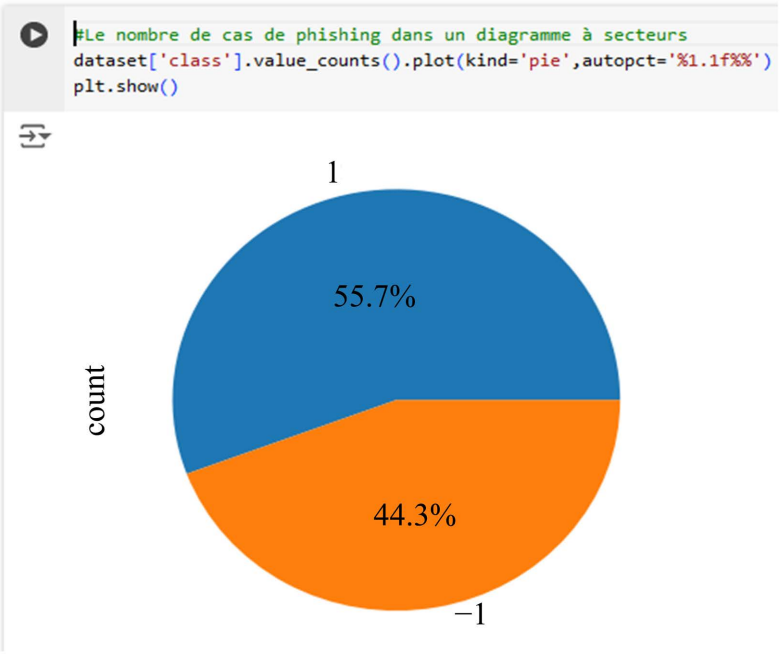

Figure 8. Diagram representing the values of two labeled classes.

This diagram is represented by Figure 8 with two parts representing the values of two different classes, *i.e.* each sample is labeled with a class identifier indicating whether it is categorized as a phishing website (1) or not (−1). Composed of 11,054 website samples. The blue part represents class 1 and occupies 55.7% of the diagram, making up 6157 phishing sites. The orange part represents class −1 and occupies 44.3% of the diagram, *i.e.* 4897 non-phishing or non-phishing sites. This diagram therefore shows that class 1 has a greater weight than class −1 in the data represented. In Figure 8, the percentage difference between the two classes is around 11.4 points (55.7% for class 1 vs. 44.3% for class −1). This diagram makes it easy to compare the proportions between the two classes of data.

4.1.5. Data Splitting

The data set is divided into training and test sets on a 70 - 30 basis, or the data has been split into a training set (70% of the data) and a test set (30% of the data). This prepares the data for training and evaluation of a machine learning model.

4.2. Algorithms Used

4.2.1. The Naïve Bayes Classifier (NBC) Algorithm

On the training data, the Naive Bayes model performed relatively well, with an Accuracy of 88.8%, an F1-score of 89.3% and an Accuracy of 94.8%. However, Recall is a little lower at 84.4%. On test data, performance remains fairly close to training data, with Accuracy at 87.6%, F1-score at 88.3% and Precision at 94.3%.

Recall fell slightly to 82.9%. The difference between performance on training and test data is relatively small (around 1 to 2 percentage points), indicating that the Naive Bayes model does not appear to be overlearned. The MAE (Mean Absolute Error) is 0.224 on training and 0.248 on test, which is relatively high compared with the other metrics. This suggests that the model still has sizeable errors in its predictions. In summary, the Naive Bayes model performs well, with accuracy and F1-score around 87% - 88% on test data. However, the Recall is a little lower, indicating that the model still has difficulty in correctly identifying certain samples. The relatively high MAE also shows that the model is still making sizeable errors in its predictions.

```
# représentation graphique de la précision de l'entraînement et du test pour différentes valeurs de var_smoothing
plt.plot(var_smoothing_values, training_accuracy, label="training accuracy")
plt.plot(var_smoothing_values, test_accuracy, label="test accuracy")
plt.ylabel("Accuracy")
plt.xlabel("var_smoothing")
plt.legend()
plt.show()
```

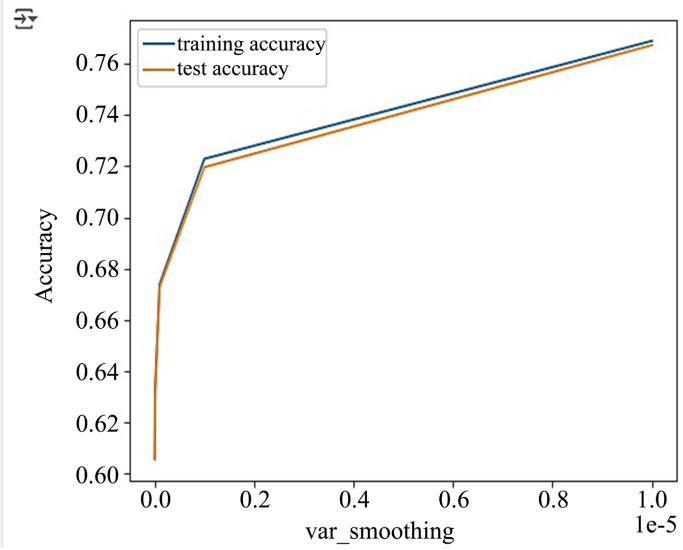

Figure 9. Graphical representation of training and test accuracy for different var_smoothing values.

Figure 9 shows the evolution of the accuracy of the machine learning model as a function of the "var_smoothing" regularization parameter. With the two curves shown: the blue curve represents the accuracy on the training data and indicates how the model performs on the data used to train it. The orange curve represents accuracy on test data. This measure indicates the model's overall performance on new data not seen during training. We can observe that as the "var_smoothing" parameter increases from 0.0 to 1.0, the accuracy on test data progressively decreases from around 0.57 to 0.55. This suggests that the model is suffering from over-training. This suggests that the model suffers from overlearning when "var_smoothing" is too low, and that this parameter needs to be carefully adjusted to find the right balance between learning on training data and generalization on

new data.

4.2.2. Decision Tree Classifier

The performance results of the decision tree model show that: Accuracy on the training set: 1.000. This indicates a very good fit to the training data. Accuracy on the test set: 0.952. This indicates that the model generalized well and achieved an accuracy of 95.2% on the test data, which is very satisfactory. F1 score on training data: 1.000. A score of 1.0 shows that the model performs perfectly on training data. F1 score on test set: 0.957, this test score also indicates very good overall model performance. These results show that the decision tree model learned the training data very well, and was able to generalize its performance well to the test set. It's a high-performance model.

Evolutionary graph of model accuracy as a function of maximum decision tree depth.

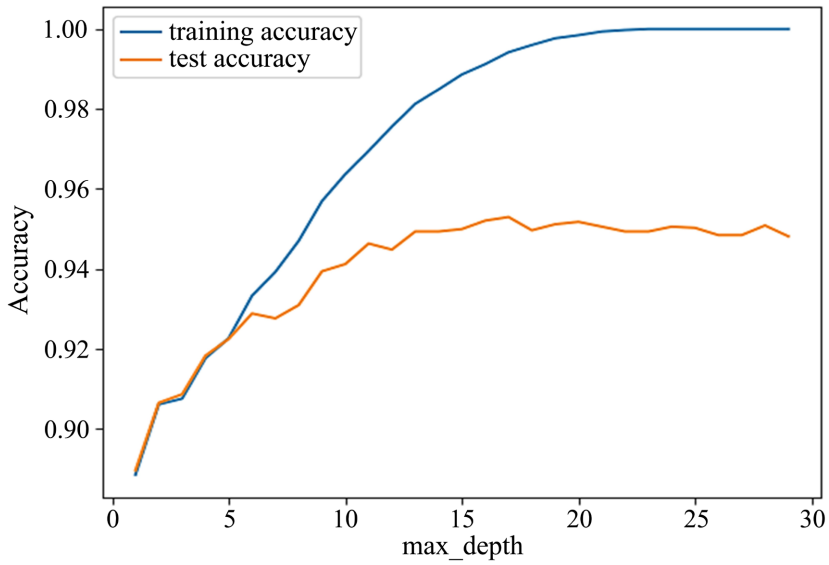

Figure 10. Graph showing the evolution of the accuracy of a machine learning model as a function of the maximum depth of the decision tree model.

Model accuracy on the training set increases monotonically with maximum depth, reaching a value close to 1 (*i.e.* 100% accuracy) for a depth greater than 25, as shown in **Figure 10**. Model accuracy on the test set follows a similar trend, but with a less steep curve. It reaches a plateau around 0.97 (*i.e.* 97% accuracy) for a depth greater than 20. The gap between training and test accuracy narrows as maximum depth increases, indicating that the model generalizes better to new data when model complexity (measured by depth) is higher. This suggests that there is a trade-off between the model's ability to fit training data well and its ability to generalize to new data.

4.2.3. Gradient Boosting Classifier

Gradient Boosting Classifier prove that the accuracy on the training data is 0.999

which means that the model has a very high, almost perfect (99.9%), accuracy on the training data. This indicates that the model fits the training data very well. The Precision on test data of 0.962 is also very high. This shows that the model generalizes well and performs well on data it didn't see during training. The F1 Score on training data of 0.999 combines precision and recall and indicates excellent overall performance on training data. The F1 Score on test data: 0.966 confirms the good generalization of the model. So, these results indicate that the Gradient Boosting Classifier performs excellently on both training and test data. The model seems well adapted to the data and capable of making accurate predictions.

Evolutionary plot of the model as a function of the learning rate used Gradient Boosting.

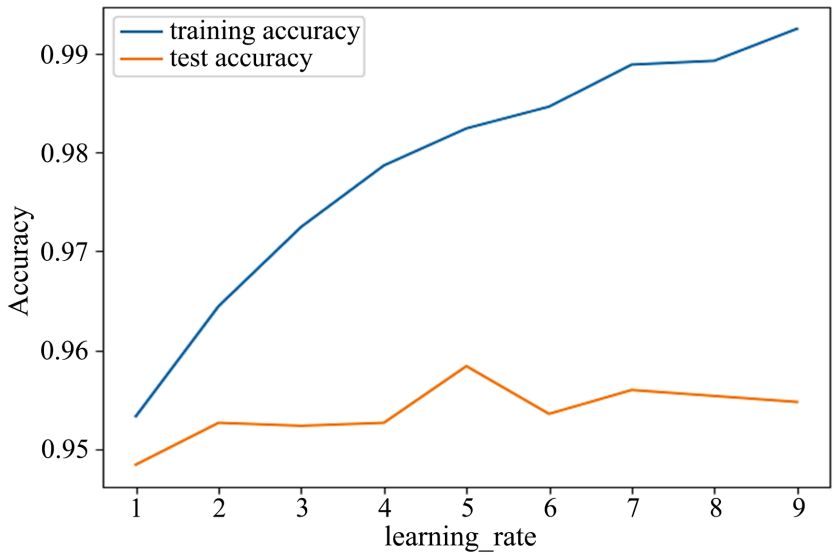

Figure 11. Graph showing the evolution of the accuracy of a machine learning model as a function of the learning rate using Gradient Boosting.

This **Figure 11** shows the evolution of the model as a function of the training rate using Gradient Boosting. It shows that model accuracy on the training set increases steadily with learning rate, reaching a value close to 0.98, or 98% accuracy at high learning rates. Model accuracy on the test set follows a similar trend, although the curve is a little less steep. It reaches a plateau around 0.965, *i.e.* 96.5% accuracy for a learning rate above 6. The gap between training and test accuracy narrows as the learning rate increases, indicating that the model generalizes better to new data with a higher learning rate. This suggests that there is a trade-off between the model's ability to fit training data well and its ability to generalize to new data. Fine-tuning the learning rate is necessary to find the best balance between these two objectives. This **Figure 11** shows the importance of the choice of learning rate in optimizing the performance of a machine learning model.

4.2.4. Multi-Layer Perceptron Classifier

The Multi-layer Perceptron model seems to perform exceptionally well in phishing detection. According to the brief summary of metrics we had: accuracy on training data is 98.8%, and on test data it is 96.4%. This means that the model makes few classification errors, with only a slight drop in performance on the test data. F1-score combines precision and recall. Our model achieves an F1-score of 98.9% on both training and test data, showing equivalent performance on both classes. Recall is high, at 98.9% on training data and 97.3% on test data. This indicates that the model correctly identifies the vast majority of samples in each class. MAE (Mean Absolute Error) our model has a MAE of 0.025 on training data and 0.071 on test data. A low MAE suggests that the model makes few absolute errors on predictions. In sum, our MLP model demonstrates excellent generalization capability, which is essential for phishing detection.

4.3. Model Comparison Table

Table 2. Model comparison.

	ML Model	Accuracy	f1_score	Recall	Precision	Mean Absolute Error
0	Naive Bayes Classifier	0.876	0.883	0.844	0.948	0.224
1	Decision Tree	0.950	0.955	1.000	1.000	0.000
2	Gradient Boosting Classifier	0.962	0.966	0.999	0.999	0.002
3	Multi-layer Perceptron	0.903	0.918	0.970	0.875	0.186

4.4. Heat Map of Model Performance Measurements

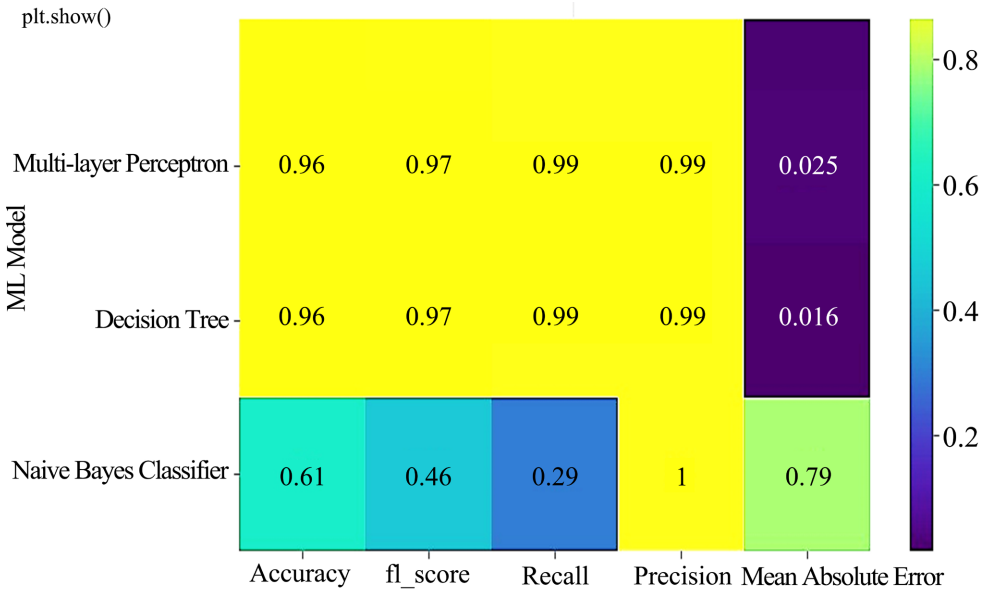

Figure 12. Heat map of performance metrics for all models.

4.5. Stacked Bar Diagram

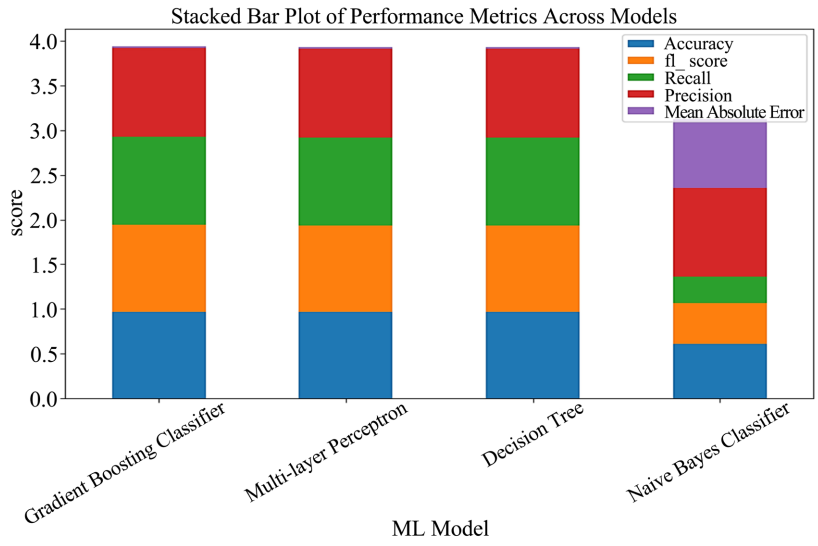

Figure 13. Stacked bar chart of performance measures for all models.

4.6. Checking the Importance of Features

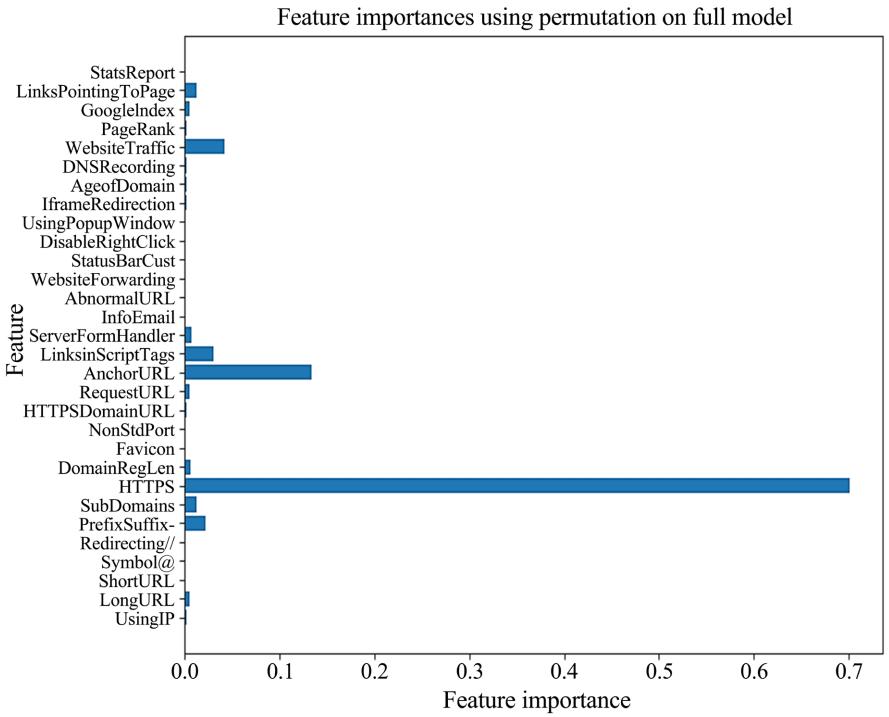

Figure 14. Checking the importance of features by permutation on the complete model.

5. Discussion of Results

5.1. Discussion of Results in Relation to Model Comparison

Different machine learning models were used in this study, and the previous sections presented the results and effects of the machine learning model on the

classification process for phishing and legitimate URLs Table 1. Comparative analyses of machine learning models are presented in this section. Table 2 and Figure 13 and Figure 14 presented the clear and significant effects of the machine learning models in this study. From the outset, the comparative analyses show that the machine learning models Naive Bayes Classifier and Multi-layer Perceptron do not perform well, Naive Bayes give very poor results and Multi-layer Perceptron satisfactory results. Gradient Boosting Classifier and Decision Tree showed very effective and significant results in classifying phishing URLs.

5.2. Comparison of Study Models

These results from Table 2 helped us to make a comparative assessment of the performance of the different machine learning models, facilitating the choice of the best model for the problem under consideration: the Naive Bayes Classifier model seems to be a good compromise in terms of simplicity and training speed, but performs slightly less well than the other models in terms of precision and recall. On the other side of Table 2, the Decision Tree performs perfectly, but as the present results show, this may indicate overtraining. The Gradient Boosting Classifier stands out for its excellent overall performance, with a precision of 96.1%, an F1-Score of 0.966 and a very low MAE of 0.002. It's a solid choice for this dataset, as it manages to balance precision and recall without showing any obvious signs of overlearning.

5.3. Discussion of Results in Relation to Metric and Stacked-Bar Performance

Figure 12 shows a heat map of performance metrics for different classification models. It is clear from this graph that Gradient Boosting Classifier, Multi-layer Perceptron and Decision Tree stand out for their high scores in terms of precision, recall, F1 score and mean absolute error. The Naïve Bayes Classifier, while having high precision, has lower recall, which affects its F1 score. In addition, its mean absolute error is higher than that of the other models. Figure 12 clearly shows that the Gradient Boosting Classifier, the Gradient Boosting Classifier, Multi-layer Perceptron and Decision Tree stand out for their high scores in terms of accuracy, F1-score, recall and precision, clearly outperforming the Naive Bayes Classifier on most metrics.

5.4. Discussion of Results versus Storing or Retaining the Best Model

The Gradient Boosting Classifier combines several weak models to form a more powerful model. The main parameters in this case are: The Learning Rate parameter: with a value of 0.7 indicates that each new tree makes a moderate contribution to the final prediction. This lower learning rate makes the model more robust to noise, but can also slow convergence. The maximum tree depth parameter or max_depth: with a depth of 4 proves that each tree can have up to 4 decision levels.

Shallow trees therefore limit the risk of overlearning, but may also reduce the model's ability to capture complex interactions. With these parameters, the GradientBoostingClassifier iteratively builds a set of shallow decision trees (max_depth = 4), updating the tree weights at each iteration according to a moderate learning rate (learning_rate = 0.7). We found this configuration to be a good starting point for a classification problem.

5.5. Experimental Results for the Most Important Characteristics

In regression problems such as ours, where prediction involves a continuous value, models are trained on labeled data in order to accurately predict results for new instances. A prediction approach to illustrate the importance of different features has been adopted to improve the results and efficiency of machine learning models, illustrated in our study by **Figure 14**. The main predictive observations remain the most important features, which are "StatsReport", followed by "LinksPointingToPage" and "GoogleIndex". According to the results presented in **Figure 14**, these three features have a high relative importance, exceeding 0.5 on the scale. This approach highlights the most influential features on the prediction model, making it possible to identify the key factors to be taken into account to improve model performance. The most important characteristics are linked to aspects such as website reputation, traffic and search engine indexing. This approach provides valuable information on the relative importance of different features in the prediction model, helping us to better understand the determining factors and direct our efforts to improve our model.

5.6. Limits and Future Prospects

Although the results of our study are satisfactory, they are not without their limitations. Although the algorithms and models used are powerful, as shown in **Table 2**, these models may have difficulty adapting to the new phishing techniques that are constantly emerging, as shown in **Figure 1**. The Naive Bayes model is limited by its simplicity and sensitivity to missing data. Gradient Boosting, on the other hand, has considerable complexity and computation time, especially in the case of our research, where the dataset to be analyzed is voluminous. The Multi-Layer Perceptron, being deep, requires a large amount of training data to avoid overlearning, so if the data is limited, the model may not generalize correctly. These models process elements from a single Kaggle dataset. However, these limitations do not diminish the scientific value of this study, nor do they remain insurmountable; to this end, our results remain scientifically valid. Therefore, to address these limitations, we are planning a future study in the first instance, on AI techniques for phishing detection using supervised learning by extending algorithms such as random drill, neural networks but also making use of other Datasets such as APWG eCrime Dataset, Phishing Dataset. In the second place on unsupervised learning and deep learning using deep neural networks for text and image analysis.

5.7. Contributions

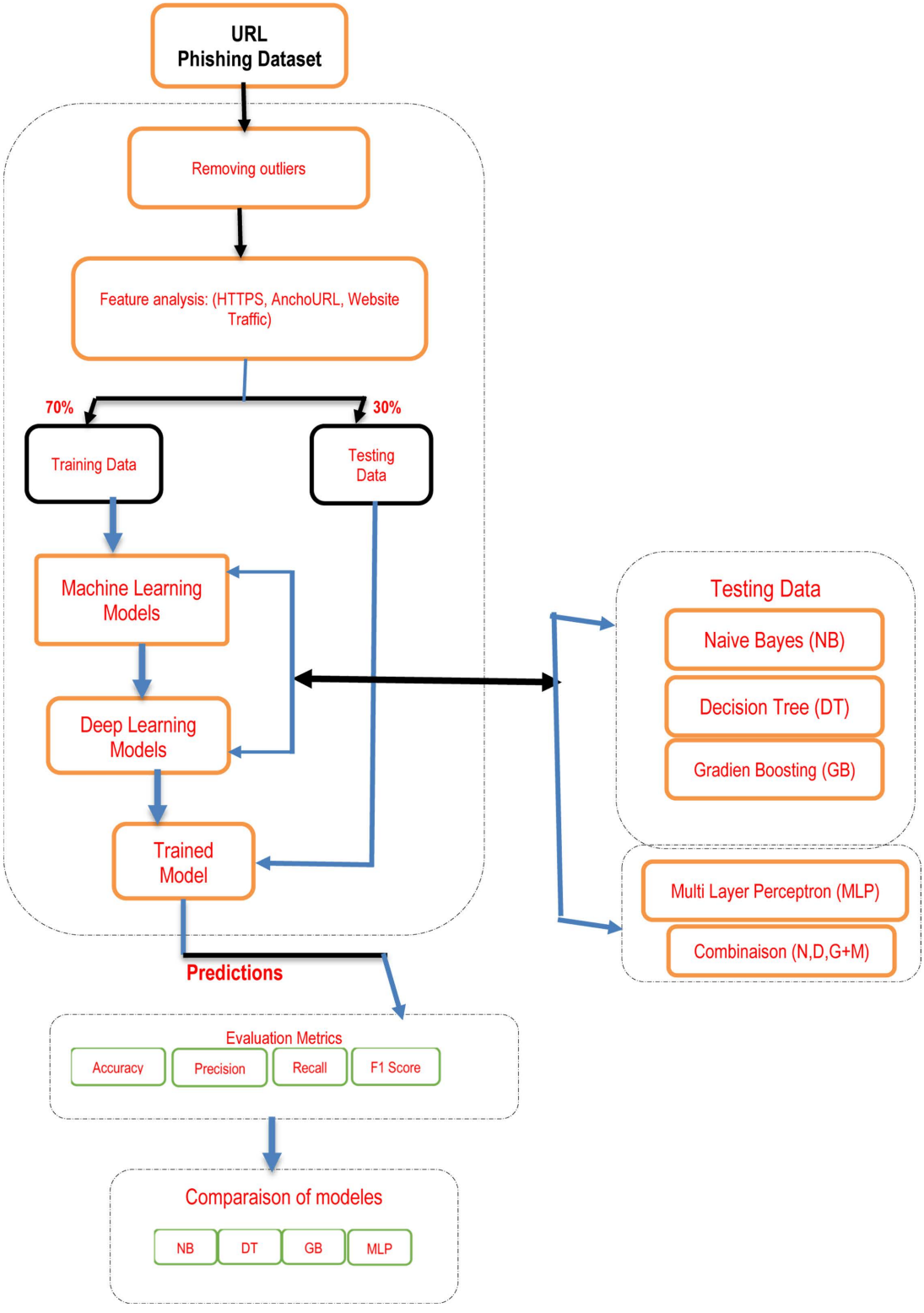

Figure 15. The NDG + M model proposed in this study.

This section presents the solution, method or model we have just implemented.

Our study makes remarkable contributions to the field of artificial intelligence by combining machine learning and deep learning. It synthesizes 3 machine learning models and 1 deep learning model. This study proposes the detection of cyber-attacks based on phishing URLs in order to prevent crime and protect people's privacy. On a global scale, the dataset includes more than 6157 phishing URL attributes or 55.7% represented by class 1 which allow phishing URLs to be classified according to these attributes. Machine learning models were applied, such as Naive Bayes (NB), Decision Tree (DT), Gradian Boosting and a deep learning model such as Multi-Layer Perceptron (MLP) giving the model (NDG + M), which accurately classified the phishing URL threats. Validation with a parameter based on the feature selection technique was used with the hybrid NDG + M model to improve prediction results. The proposed methodology was evaluated using evaluation parameters such as accuracy, precision, recall, specificity and F1 score. **Figure 15** shows the proposed model to improve the efficiency and accuracy of phishing detection.

In summary, the proposed NDG + M model aims to improve the effectiveness and accuracy of phishing detection with the predictive approach. The study focuses on phishing prevention techniques, including URL filtering, user education and training [33], real-time detection by machine learning. It also provides a detailed explanation of how 4 machine learning models work. However, it does not explore the details of the dataset are omitted, which has an impact on generalizability.

6. Conclusions

With the development of the Internet, cybercrime is on the increase on a daily basis, particularly through the use of suspicious and malicious URLs. This has a significant impact on the quality of services provided by the Internet and industrial companies. The detection of phishing is a major challenge for online security. The aim of this study was to analyze the different AI approaches used for phishing detection, and to determine the most practical types of supervised machine learning algorithms. The results we obtained with the Gradient Boosting Classifier are impressive. Precision of 96.2%, F1-score of 96.6% and recall of 99.9% across all classes are solid performances. The fact that this model balances precision and recall well is crucial to minimizing false positives while effectively detecting true positives. However, there are still aspects to be explored in future work, and the creation of a new dataset based on phishing URLs is essential to improve the robustness of detection systems. The integration of unsupervised, hybrid machine learning algorithms and elements of user behavior analysis may also bring new perspectives.

Raising staff awareness of security and phishing techniques is also a crucial step. Vigilance, source verification and ongoing training are effective ways of protecting against these growing threats. In conclusion, this study has provided valuable insights into phishing detection and practical skills in template selection, feature analysis and performance optimization.

Conflicts of Interest

The authors declare no conflicts of interest.

References

[1] C.S.E. (2023) Intelligence Artificielle et Cybersécurité Contrôler Le défi de rendre l'IA éthique.

[2] Zhang, Y., Egelman, S., Cranor, L. and Hong, J. (2007) Phinding Phish: Evaluating Anti-Phishing Tools.

[3] Park, H., Lim, K., Kim, D., Yu, D. and Koo, H. (2023) Demystifying the Regional Phishing Landscape in South Korea. *IEEE Access*, **11**, 130131-130143. https://doi.org/10.1109/access.2023.3333883

[4] Response, G. and Cybercrime, T. (2023) 4 Quarter, No. February 2024.

[5] Rosay, A. (2020) MLP4NIDS: Application pratique d'un réseau de neurones pour la détection d'intrusions réseau dans les voitures connectées Méthodologie et solution à perceptron multi-couche Travaux antérieurs. 4-11.

[6] Reddy, J.M. and Rao, K.V. (2020) An Approach for Detecting Phishing Attacks Using Machine Learning.

[7] Hermann, Y.K.J. and Frederic, O.T. (2023) Study on the Use of Artificial Intelligence for Cybersecurity in Companies: Case of Companies in Burkina Faso. *Engineering*, **15**, 798-809. https://doi.org/10.4236/eng.2023.1512056

[8] Namatherdhala, B., Mazher, N. and Sriram, G.K. (2022) Artificial Intelligence in Product Management: Systematic Review. *International Research Journal of Modernization in Engineering Technology and Science*, **4**, 2914-2917.

[9] Yang, P., Zhao, G. and Zeng, P. (2019) Phishing Website Detection Based on Multidimensional Features Driven by Deep Learning. *IEEE Access*, **7**, 15196-15209. https://doi.org/10.1109/access.2019.2892066

[10] Mohammad, R.M., Thabtah, F. and McCluskey, L. (2013) Predicting Phishing Websites Based on Self-Structuring Neural Network. *Neural Computing and Applications*, **25**, 443-458. https://doi.org/10.1007/s00521-013-1490-z

[11] Cengiz, E. and Gök, M. (2023) Reinforcement Learning Applications in Cyber Security: A Review. *Sakarya University Journal of Science*, **27**, 481-503. https://doi.org/10.16984/saufenbilder.1237742

[12] Bohacik, J., Skula, I. and Zabovsky, M. (2020). Data Mining-Based Phishing Detection. *Proceedings of the* 2020 *Federated Conference on Computer Science and Information Systems*, Vol. 21, 27-30. https://doi.org/10.15439/2020f140

[13] Zhang, Y., Lu, Y. and Liu, F. (2023) A Systematic Survey for Differential Privacy Techniques in Federated Learning. *Journal of Information Security*, **14**, 111-135. https://doi.org/10.4236/jis.2023.142008

[14] Fauvelle, J., Dey, A., Navers, S., Fauvelle, J., Dey, A. and Navers, S. (2019) Protection d'un système d'information par une intelligence artificielle: Une approche en trois phases basée sur l'analyse UEBA des comportements pour détecter un scénario hostile.

[15] Korkmaz, M., Sahingoz, O.K. and Diri, B. (2020). Detection of Phishing Websites by Using Machine Learning-Based URL Analysis. 2020 11*th International Conference on Computing, Communication and Networking Technologies* (*ICCCNT*), Kharagpur, 1-3 July 2020, 1-7. https://doi.org/10.1109/icccnt49239.2020.9225561

[16] Krichen, M. (2023) Renforcer la sécurité des contrats intelligents grâce à la puissance de l'intelligence artificielle Moez Krichen.

[17] Amine, F.M. (2020) Mémoire de Fin d'études Master Un système de dé tection d'intrusion pour la cybersécurité.

[18] Shahrivari, V. (2020) Phishing Detection Using Machine Learning Techniques.

[19] Translated, M. and Joga, S.R.K. (2023) Système de de phishing via hybride Apprentissage automatique basé sur l'URL. 36805-36822.

[20] Morucci, S., *et al.* (2019) Algorithmes d'Intelligence Artificielle en Cybersécurité & Intégration en environnements contraints.

[21] Shahrivari, V. (2018) Détection de l'hameçonnage à l'aide de techniques d'apprentissage automatique.

[22] Zamir, A., Khan, H.U. and Iqbal, T. (2020) Détection de sites Web de phishing à l'aide de divers algorithmes d'apprentissage automatique Ammara.

[23] Abdelhakim, H. (2020) Mémoire de Fin d'études Master Détection des sites d'hameçonnage pour assurer la sécurité sur Internet.

[24] Kharj, A. (2022) Machine Translated by Google électronique Article URL de intelligente pour l'apprentissage automatique profond Détection basée sur l'extraction de Muna lharbi Salut secteurs d'activité des de est le est proposé Machine Translated by Google.

[25] Rouvi, L. (2022) Apprentissage supervisé—Machine learning.

[26] Detection, B. (2023) Recent Trends in SERS-Based Plasmonic Sensors for Disease.

[27] Ogah, M.D., Essien, J., Ogharandukun, M. and Abdullahi, M. (2024) Machine Learning Models for Heterogenous Network Security Anomaly Detection. *Journal of Computer and Communications*, **12**, 38-58. https://doi.org/10.4236/jcc.2024.126004

[28] Azencott, C. (2018) Introduction au Machine Learning Préambule.

[29] Alnemari, S. (2023) Applied Sciences.

[30] Jaotombo, F. (2022) Apports des méthodes de Machine Learning et de Deep Learning dans la prédiction des durées de séjours hospitalières et des ré-hospitalisations. https://theses.hal.science/tel-04079356%0A https://theses.hal.science/tel-04079356/document

[31] Principes de l'apprentissage statistique supervisé (Supervised Machine Learning) Romain Couillet.

[32] Royer, W. (2021) Fondements du Machine Learning.

[33] Kiseki, D.W., Havyarimana, V., Niyonsaba, T., Zabagunda, D.L., Wail, W.I. and Semong, T. (2023) The Knowledge of Cyber-Security Vulnerabilities in an Institution of Higher and University Education. A Case of ISP-Bukavu (Institut Supérieur Pédagogique de Bukavu) (TTC = Teachers' Training College). *Journal of Computer and Communications*, **11**, 12-32.