An Empirical Analysis of Machine Learning Approaches for Phishing Detection

Ivan Cvitić[1](✉), Hussam Al-Hamadi[2], Tibor Mijo Kuljanić[3], and David Aleksić[1]

[1] Faculty of Transport and Traffic Sciences, University of Zagreb, Vukelićeva 4, 10000 Zagreb, Croatia
ivan.cvitic@fpz.unizg.hr

[2] University of Dubai, Academic City Emirates Road - Exit 49, Dubai, United Arab Emirates
halhammadi@ud.ac.ae

[3] HEP ODS d.o.o., Ulica grada Vukovara 37, 10 000 Zagreb, Croatia

Abstract. This research paper investigates the integration of Artificial Intelligence (AI), with a focus on Machine Learning (ML) and Deep Learning (DL), for bolstering cybersecurity defences against phishing attacks. Utilizing a comprehensive dataset of URL features, the study assesses the efficacy of various ML algorithms—namely Decision Tree, Logistic Regression, Support Vector Machine, Random Forest, and K-Nearest Neighbors—in pinpointing phishing websites. Research paper is conducted using Google Colaboratory, Python libraries and Weka tool. The research identifies the Random Forest algorithm as the most effective, demonstrating superior accuracy in detecting phishing URLs during both training and testing phases. The findings accentuate the pivotal role of AI in advancing cybersecurity measures, advocating for the incorporation of sophisticated AI technologies in the fight against cyber threats. Additionally, it outlines future research directions, including the enhancement of model precision through the integration of more comprehensive data attributes. This paper significantly contributes to the cybersecurity and AI domains by showcasing the practical applications and benefits of AI in identifying and mitigating cyber risks.

Keywords: Artificial Intelligence · Machine learning · Deep Learning · Phishing attacks

1 Introduction

With the number of devices and device interconnectivity increasing, the potential attack surface on systems is expanding. Cybersecurity in the digital age is becoming an increasingly critical factor and one of the most challenging factors to achieve, considering the growing numbers and types of attacks and attackers targeting systems. Cybersecurity is a set of procedures, measures and specific standards aimed at maintaining reliability when using products or services in the cyber domain. In the implementation of these measures, procedures and standards, the use of AI is becoming more common [1].

D. Perakovic and L. Knapcikova (Eds.): FABULOUS 2024, LNICST 596, pp. 34–53, 2025.
https://doi.org/10.1007/978-3-031-72393-3_4

The digital landscape is fraught with cybersecurity threats, among which phishing attacks are particularly pervasive and evolving. This research paper investigates the efficacy of Artificial Intelligence (AI), specifically Machine Learning (ML) and Deep Learning (DL), in the detection and mitigation of phishing threats. By systematically evaluating a range of ML algorithms—Decision Trees, Logistic Regression, Support Vector Machines, Random Forest, and K-Nearest Neighbors—applied to a dataset of URL features, the study aims to discern the most effective algorithmic approaches for phishing detection.

The research is driven by the hypothesis that advanced AI techniques can significantly outperform traditional heuristic-based detection methods in identifying phishing attempts, offering a scalable and adaptable solution to cybersecurity. The goal is to provide a comprehensive assessment of these AI methodologies' performance, thereby contributing to the strategic arsenal against cyber threats. This exploration includes an analysis of algorithmic accuracy, computational efficiency, and the potential for real-world application, underpinned by a rigorous statistical evaluation framework.

Through this endeavor, the paper seeks not only to validate the practical utility of ML and DL in cybersecurity but also to illuminate the pathway for future research in AI-driven threat detection. The implications of this research are far-reaching, offering insights into the development of more resilient digital infrastructures and informing policy and practice in cybersecurity management.

The structure of the paper is as follows: the first chapter is this introductory chapter, and after that, the second chapter of the paper provides an overview of previous research related to the application of AI in the field of cybersecurity. The reviewed papers offer an overview of ML and DL algorithms used in attack detection systems. The third chapter provides additional insights into areas of AI that can be applied in the field of cyber sciences. The fourth chapter of the paper focuses on developing a ML model for recognizing phishing URL addresses. In this chapter, the dataset and the tools used for its processing are analyzed. Following that, the process of model development using ML algorithms is presented. The fifth chapter continues from the previous chapter and involves the analysis and interpretation of the results obtained in the previous chapter. The last chapter is the concluding chapter, where the conclusions derived from the analysis of the results are presented.

2 Previous Research

The potential applications of ML algorithms and AI in the field of cybersecurity are increasing steadily, driven by technological advancements and the significant expansion of resources for data collection and processing.

The paper [2] from 2022 provides insights into events in the field of cybersecurity related to the application of machine learning. User access compromise, which includes phishing attacks, is one type of attack. According to the authors, phishing attacks are considered as tricks in which malicious users attempt to exploit the carelessness or ignorance of users to carry out malicious actions, such as stealing user data or installing malicious software on the user's computer.

The paper provides a detailed overview of research results from other authors who conducted similar research in the period from 2013 to 2018. In 2015, the focus of the

research was on the detection of Denial of Service (DoS) attacks, User-to-Root (U2R) attacks, and Remote-to-Local (R2L) attacks using ML, specifically DT algorithms. The overall detection accuracy achieved for these attacks was 92.62%.

The authors of paper [3] focus exclusively on phishing attacks and ML methods for attack detection. They analyze the results obtained for the detection of phishing attacks using different ML algorithms. In this research, the authors used labeled datasets for supervised machine learning. The goal of using ML in this study is to recognize phishing attacks. To accomplish this, a ML model must classify a URL in an incoming message as either fake (phishing) or legitimate, with no malicious intent, based on the training it has received. The paper shows that the best results were achieved using the RF algorithm. This algorithm can be used for classification and regression methods. It provided the highest efficiency in attack detection during the testing phase, with an accuracy rate of 90%, while during the training phase, it achieved a result of 81.837%.

Paper [4] provides an overview of the general use of AI and ML in the field of telecommunications. Some of the possibilities for using AI and ML in telecommunications include data collection, interpretation of collected data, cybersecurity and more. The paper explains the differences between three types of machine learning: supervised, unsupervised and reinforcement learning. For each type the most well-known methods used in developing ML models are listed.

The paper also includes examples of how ML is used in the field of telecommunications, such as in the detection of anomalies within network traffic and in distinguishing between legitimate and malicious traffic to detect potential DDoS attacks on a system. ML can also be used in Industry 4.0, which has a goal to make all devices communicate autonomously with each other by using information and communication technologies. ML can be used to predict network traffic volumes and in device maintenance, which is essential and highly advantageous when the end goal is to enable devices to communicate with each other independently.

Author Chawla A. in his work [5] conducts research on preventing phishing attacks on systems using various ML algorithms. In the paper, a dataset of 11,056 websites with 30 parameters is mentioned based on which it will be validated whether the website is genuine or fake. Some of these 30 parameters used for analyzing the URL of a website are [5]:

- Does the website have an IP address – a parametar indicating whether the website is registered to a domain or not.
- URL Length - if the URL length is shorter than 54 characters, the website can be considered reliable. However, if the length falls between 54 and 75 characters, the website is considered a potential threat or suspicious.
- Website Forwarding - this parameter indicates the number of redirections. If this number is greater than one, it may raise suspicion of a phishing URL.
- Disabled Right Click - this parameter indicates whether the right-click functionality is disabled. This feature is significant because right-clicking often enables users to check a page or its source code. If it's disabled, there's a higher likelihood that an attacker intentionally prevented these actions.

- Using Pop-up Windows - a parameter that indicates whether the page requests data, information, etc. in pop-up windows. If the page requests this, it can be classified as a phishing site.

While existing research extensively explores AI's role in cybersecurity, particularly for phishing detection using ML and DL, it often lacks a detailed comparative analysis of these algorithms against emerging phishing methodologies. This paper addresses this research gap by empirically assessing the performance of various ML algorithms in identifying sophisticated phishing attacks, thus advancing the understanding of AI's practical efficacy in contemporary cybersecurity challenges.

3 Possibilities of Applying Artificial Intelligence in Cybersecurity

AI in the field of cybersecurity is employed for the detection of attacks and intrusions by malicious code and users into systems, as well as for attack prevention. AI encompasses several scientific domains that collectively fall under the umbrella of AI. In cybersecurity, the most commonly utilized branches of AI include ML, DL and Artificial Neural Networks (ANN).

3.1 Cybersecurity

Cybersecurity is a process aimed at protecting systems, computer networks and software from cyber threats and attacks as well as unauthorized access to them. The goal of cybersecurity is to reduce or completely eliminate the possibility for malicious users to achieve their objectives through its procedures and measures, although it is impossible to entirely eliminate this possibility due to the constant evolution of attacks and technologies for malicious actions against systems [6].

Phishing attacks are web-based attacks in which malicious users deceitfully redirect the targeted user to a malicious website. Phishing attacks have long been known in the world of cybersecurity and the typical form and appearance of a phishing attack in various formats (SMS, email, phone call) are well-known. Phishing attacks that come via email are relatively easy to recognize by examining the sender's address as malicious users often pretend to be large corporations or businesses. In their email addresses, there is usually an intentional mistake made in the hope that the target of the attack won't notice it. For example, instead of "@microsoft.com," a malicious user might create an address in which they swap the positions of two letters, so that the address looks like "@mircosoft.com".

The emergence of AI significantly aids in the recognition and maintenance of cybersecurity. AI has advanced to the point where it simplifies working with large datasets and in turn it reduces the time and resource requirements, which is crucial in many cases. An advantage of AI is its ability to learn and evolve over time [7].

3.2 Machine Learning in Field of Cybersecurity

ML is a field of AI that aims to teach systems to think in a manner as close to humans as possible in order to solve predefined problems and tasks [8]. The basic categorization of ML includes supervised machine learning, unsupervised machine learning and reinforcement machine learning.

The DT algorithm is often the first choice when selecting an algorithm for intrusion detection or spam message detection. The reason for choosing the DT algorithm is that it excels in identifying rules and patterns within network traffic and system usage making it easy to spot anomalies occurring within the system and network traffic [9].

The K-means clustering algorithm is an unsupervised machine learning algorithm that works on a simple principle of grouping data based on their similarity [9]. In addition to its straightforward operation, the advantages of the k-means algorithm include its ability to work with large datasets which is even recommended to obtain a more extensive dataset for more detailed determination of data group similarities [10]. In the field of cybersecurity this algorithm is used for creating Intrusion Detection System (IDS) models, detecting malware, filtering emails for spam and legitimate messages. It can also be used for analyzing log records from proxy servers and captive portals where the collected data from the server is clustered using the K-means algorithm to analyze user trends which can then be used to detect anomalies and identify potential malicious users of the website [11].

3.3 Application of Decision Tree Algorithm for Intrusion Detection Systems

The DT algorithm is used in the process of analyzing network traffic in IDS. The DT algorithm is defined as a predictive modeling technology that arises from the fields of statistics and ML and it is used to create a tree-like structure model which is why it's called the DT algorithm. The advantage of using the DT algorithm is that it is a direct process that can be completed within a few hours or months depending on the size of the dataset and the level of detail required for investigating and understanding the type of intrusion in the system.

After the data collection process the collected data undergoes preprocessing where it is processed and transformed into the necessary format to be used in the model-building process using the DT algorithm. The next step after data preprocessing is the model training process followed by the final step of analyzing the results obtained from the model in a real environment [12].

3.4 Application of SVM Algorithm for Intrusion Detection Systems

The goal of the SVM ML algorithm is to find optimal separating hyperplanes that maximize the utilization of data during the training phase while minimizing complexity and the risk of overfitting [13]. Hyperplanes are decision boundaries used to facilitate data classification when applying the SVM algorithm [8]. Overfitting is an undesirable occurrence in the field of ML where the model provides accurate predictions for the data used in the training and evaluation phases but fails to do so for new data when the model is deployed in a real-time environment [14]. The application of the SVM algorithm is

suitable for working with small datasets during the training phase and is also suitable for analyzing extremely large datasets.

When applying the SVM algorithm for creating an IDS model, the first step is to collect data for model development. After selecting the dataset for model creation the initial training phase and classification phase are carried out. In these phases it is crucial to have a good understanding of the dataset since SVM is a supervised machine learning algorithm and accurate knowledge of the data is essential for effective training. Specifically, it is vital to know precisely which data points in the dataset represent malicious and legitimate data so that the model can learn correctly. Additionally, before starting the model creation the selected dataset must undergo a preprocessing stage to optimize the data for the SVM algorithm [15].

According to [15] SVM is one of the best ML algorithms for anomaly detection and recognizing unauthorized intrusions in network traffic. In contrast to the DT (DT) algorithm, which is also used in IDS systems, the SVM algorithm is more demanding, complex and requires more time and computational resources for model development.

3.5 K-Means Clustering for Spam and Phishing Email Detection

Phishing emails are classified as spam in electronic mail. Spam email is a form of unwanted messages that users commonly receive from various brands, stores and other senders, containing promotional content in the form of product and service advertisements. Although users receive a few spam messages daily, classifiers are used to automatically categorize this mail into one of the categories within the used email application or platform [16]. Due to the high volume of incoming promotional messages that users receive on a daily basis, malicious users disguise their phishing attacks to make their emails resemble harmless spam mails. To properly categorize incoming mail, mail sorting filters are used, utilizing the K-means algorithm for their operation.

A simplified representation of the filter architecture used for categorizing mail is shown in Fig. 1. The figure displays the email's header and body as the email architecture from which data and characteristics are extracted to filter the mail. The filter itself involves four processes [17]:

- Tokenization – the process in which key words are extracted from the context of the entire message, which for example may be located within the body of an email.
- Term Selection - a procedure in which extracted words are ranked based on their significance.
- Feature Extraction - a process in which the extracted words are further examined, reduced, and refined to obtain a smaller dataset of importance.
- Classifier - the component where, based on K significant algorithmic features, the extracted data/word is assigned to one of K data sets or groups.

Ultimately, upon exiting the filter the final result indicates whether the incoming mail is legitimate or if it's incoming spam/phishing mail.

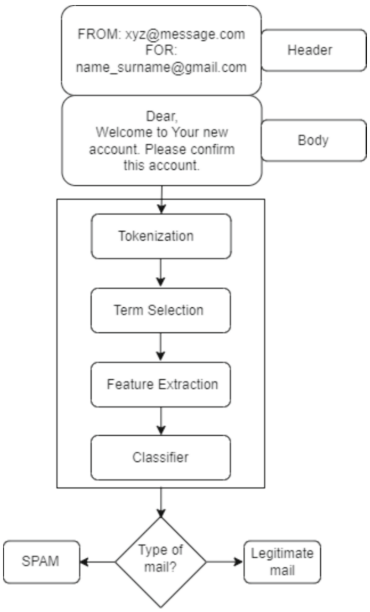

Fig. 1. Spam filter architecture [17]

3.6 Deep Learning and Artificial Neural Networks in the Field of Cybersecurity

According to [18], DL can be considered a subset of ML, although it is generally classified as a subfield of AI. DL, in conjunction with ANN, attempts to mimic human thinking and learning processes, executing various tasks based on that thinking [19].

The difference between ML and DL lies in the fact that ML models are created using algorithms trained to adapt to their environment and correctly perform their tasks regardless of minor changes in the surroundings. On the other hand, models generated through a combination of ANN and DL are specifically trained to emulate human thinking only for known situations [20].

The advantage of using DL over ML is that unstructured data can be challenging and demanding for ML algorithms, while DL algorithms do not encounter such difficulties.

DL algorithms in combination with ANN operate on a principle similar to the human brain. The human brain consists of millions of interconnected neurons that collectively learn and process incoming information. In a similar fashion, ANN consist of multiple layers of artificial neurons that work together within a computer or system to process the received data [20].

A deep ANN consists of three main layers. The first layer is the input layer, which includes a few nodes that introduce incoming data into the network. Following the input layer, there is a hidden layer, which can contain hundreds of additional hidden layers used to analyze the input data to properly train the model. At the end of the network, there is the output layer which provides output data after processing and analysis in the hidden layer. The number of nodes in this layer depends on the quantity of outputs; for

example, if the output data is only "YES" or "NO," then the output layer will contain only two nodes [20].

According to [21], the three most commonly used types of ANN are: Feedforward neural networks (FNNs), Convolutional neural networks (CNNs), and Recurrent neural networks (RNNs). In the field of cybersecurity, DL algorithms can be employed for the detection of DDoS attacks, identification of malicious software, botnet network detection, network traffic analysis and anomaly detection within it and so on [21].

The use of ANN and DL algorithms in the detection of DDoS attacks in cybersecurity systems, according to [22], can be divided into five steps. The first step, like in any DL algorithm, is data collection where datasets are gathered. In the second step the preprocessing process takes place. Preprocessing involves cleaning the selected data, identifying key features within that data and preparing the data for further processing. The next step is the division of prepared data into three groups. These three data groups are: the training data group, the validation data group and the testing data group. All these data groups are used during the training phase but at different stages. In the fourth step, the training data groups and the validation phase are utilized. This step is where the design and structuring of the deep ANN occur. During this step the process of adjusting hyper parameters is performed in conjunction with the validation process to obtain the correct and optimal network structure. After determining the optimal network structure the evaluation process of the designed model is carried out using the testing data group [22].

The input layer of an ANN is responsible for receiving data from the dataset and typically, each node within the input layer corresponds to one dimension or feature within the data. The hidden layer receives data from the input layer. This layer, along with any other possible hidden layers within it, aims to determine the appropriate number of nodes to allow the model to learn and have the ability to process complex information.

The output layer receives data after they have passed through all the preceding layers. The number of nodes in this layer represents the probability values ranging from 0.00 to 1.00. After this step the process of adjusting hyper parameters is performed to enhance the quality and effectiveness of the learning process [22].

4 The Application of Machine Learning Methods in the Field of Cybersecurity

This chapter pertains to the development of a ML model for recognizing phishing addresses from a dataset. Before the models are constructed it is necessary to collect and preprocess the data that will be used for creating the ML model. The algorithms used for building the model include LR, SVM, RF, DT, and K-Nearest Neighbors (KNN) algorithms. Figure 2 shows flowchart that describes the workflow of making a ML model using different ML algorithms.

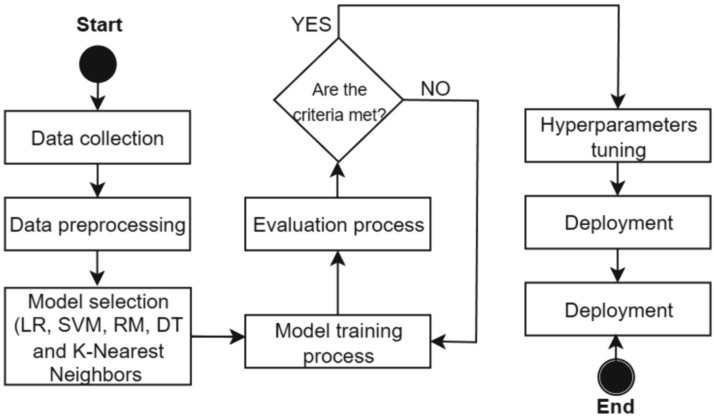

Fig. 2. Workflow of ML model making

4.1 Dataset Selection and Preprocessing

The dataset used for this research was created in May 2020 and is publicly available with A. Hannousse and S. Yahiouche listed as the authors [23]. The data was published on the "Mendeley Data" website on September 28, 2020. This dataset was created for the development of AI models for detecting phishing websites. It consists of 11,430 URL addresses, evenly divided in a 50/50 ratio. This means that 50% of the URLs are legitimate real websites, while the remaining 50% are phishing addresses. The website provides two available datasets and the second one named "dataset_B_05_2020.csv" was used [23]. The downloaded dataset has 89 features or attributes for identifying phishing addresses. Some of these attributes include URL length, IP address, the number of characters in the address, the number of dots within the URL, domain age, the amount of web traffic generated on the page and so on [23].

For data processing within the dataset, Weka software tool was used. Weka is utilized for various data mining tasks such as data preprocessing, classification, regression, data visualization and more. It provides a range of ML and data mining algorithms, making it a valuable tool for analyzing and working with datasets in research and analysis.

Data preprocessing involves cleaning, transforming, and preparing data to make it compatible for use with specific ML algorithms and to facilitate subsequent data analysis. Data preprocessing is essential because datasets often contain redundant or irrelevant information. When there is a significant amount of data with the same values it leads to redundancy, which can slow down the process and increase the risk of errors or obstruct the creation of a high-quality model. Data cleaning is one of the techniques used to remove or adjust duplicate data, incorrect values or missing values. Another technique is data transformation, which involves modifying data within the dataset to improve the efficiency and accuracy when building a machine learning model.

After loading data into the tool, it's necessary to select the appropriate filter that will enable proper data handling. By choosing a supervised learning filter like "AttributeSelection" the number of attributes within the dataset is reduced from 89 to 14 selected attributes that are suitable for working with supervised machine learning algorithms.

4.2 Model Development Using Logistic Regression Algorithm

The first step in building a model using all algorithms is to load the ML packages and libraries. The libraries that are loaded include “pandas”, “numpy” and “random”. NumPy is used for a wide range of mathematical operations performed during ML model development. Pandas is a Python library used for working with datasets, providing capabilities such as data cleaning, exploration and manipulation within datasets. Figure 3 displays the process of loading libraries and ML packages.

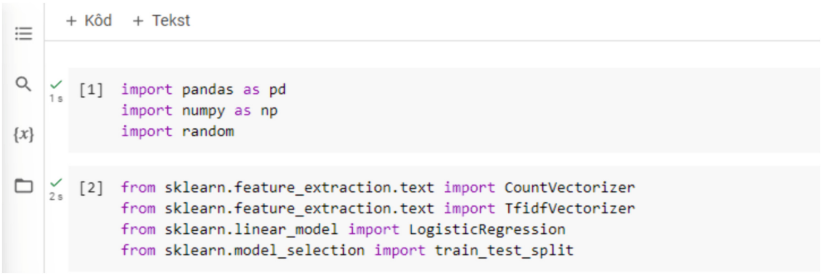

Fig. 3. Loading Python libraries and machine learning packages

After successfully loading the libraries and packages the next step in model creation is the tokenization process. Tokens are used for tokenization, which involves extracting key words or components from the data, in this case from the addresses. The first token is created after the appearance of the “/” character within the web address, the second token corresponds to the appearance of the “-” character within the address and the third token is created for the appearance of a period within the address. The second part of token creation involves removing redundant tokens and the final step is to remove potential tokens that contain “.com” since this extension frequently occurs in the dataset.

```
[60] def makeTokens(f):
         tkns_BySlash = str(f.encode('utf-8')).split('/')
         total_Tokens = []
         for i in tkns_BySlash:
             tokens = str(i).split('-')
             tkns_ByDot = []
             for j in range(0,len(tokens)):
                 temp_Tokens = str(tokens[j]).split('.')
                 tkns_ByDot = tkns_ByDot + temp_Tokens
             total_Tokens = total_Tokens + tokens + tkns_ByDot
         total_Tokens = list(set(total_Tokens))
         if 'com' in total_Tokens:
             total_Tokens.remove('com')
         return total_Tokens
```

Fig. 4. Token creation for Logistic Regression model

In Fig. 4 the process of token creation is illustrated. After successfully creating tokens the next step is the creation of labels and attributes and the utilization of the generated tokens. In the next Fig. 4 the dataset is divided into two groups in an 80/20 ratio. 80% of

the data from the dataset will be used in the training phase, while the remaining 20% of the data will be used in the testing phase. Additionally, in Fig. 5 the accuracy achieved during the model training phase is displayed and it stands at 0.88539 which is equivalent to 88.54%.

```
[94] X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

[95] logit = LogisticRegression()
     logit.fit(X_train, y_train)

     LogisticRegression()

[96] print("Accuracy ",logit.score(X_test, y_test))

     Accuracy  0.8853893263342082
```

Fig. 5. Data splitting into two groups (training and evaluation)

The final step is to conduct the testing phase using the remaining 20% of the data. The testing process is repeated twice for different web addresses and different quantities of addresses. In the first test, three addresses were used, while in the second test four were employed. In the end, the percentage accuracy of predicting phishing addresses is achieved, which stands at 0.90245 or 90.24%.

4.3 Model Development Using the SVM Algorithm

To create a model using the SVM algorithm the same dataset as used for the LR model is employed. In the first step of model development, ML libraries and packages are loaded from the Scikit-Learn program. Unlike the previous LR model, in this example, only "numpy" is loaded from the libraries. The data splitting process from the dataset into training and testing data is carried out in an 80/20 ratio.

```
✓ 1 min [11] clf = SVC(kernel='linear')
             clf.fit(X_train, y_train)

             SVC(kernel='linear')

✓ 26 s [12] predictions = clf.predict(X_test)

✓ 23 s [13] print("Training precision", clf.predict(X_test))

       Training precision ['phishing' 'legitimate' 'legitimate' ... 'phishing' 'phishing' 'phishing']
```

Fig. 6. The training phase of the SVM algorithm model

Once the data is split, the model is trained using the SVM algorithm and the process of recognizing the data status within the training dataset for model development takes place. This step is illustrated in Fig. 6.

The final step in creating a model for recognizing phishing URL addresses using the SVM algorithm is to carry out the testing phase. The testing phase resulted in an average recognition accuracy of 0.92576 which is equivalent to 92.58%. Figure 7 displays the result output for the SVM algorithm.

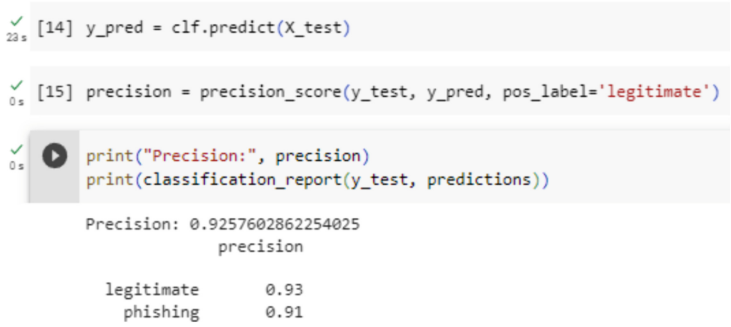

Fig. 7. Accuracy output for the recognition model using the SVM algorithm

4.4 Development Using the Random Forest Algorithm

To create a model using the RF algorithm it is necessary to import the libraries "numpy" and "pandas" as well as the ML package from Scikit Learn for working with the RF algorithm. After that, the dataset needs to be loaded again and data splitting for the training and testing phases is performed just as in all the previous examples.

Following these initial steps the model training and testing procedures are executed and the accuracy verification phases within the training and testing of the model take place. Figure 8 illustrates the process of training and testing the model.

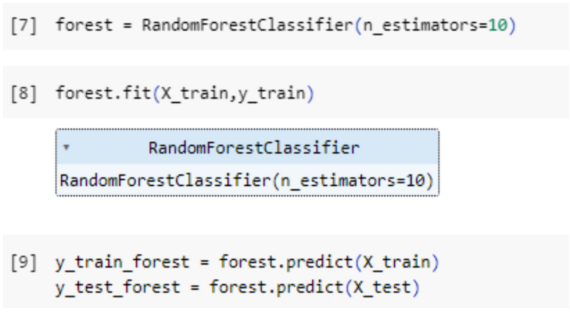

Fig. 8. The process of training and testing the Random Forest algorithm model

The process of training and testing the RF algorithm model. In this example, the "n_estimators" parameter, which typically represents the number of trees within the RF algorithm, is set to a value of 10. The value 10 is the default starting value meaning that the algorithm will create 10 different trees during model creation.

```
[10] acc_train_forest = metrics.accuracy_score(y_train,y_train_forest)
     acc_test_forest = metrics.accuracy_score(y_test,y_test_forest)
     print("Random Forest : Accuracy on training Data: {:.3f}".format(acc_train_forest))
     print("Random Forest : Accuracy on test Data: {:.3f}".format(acc_test_forest))
     print()

     Random Forest : Accuracy on training Data: 0.973
     Random Forest : Accuracy on test Data: 0.933


[11] print(metrics.classification_report(y_test, y_test_forest))

                  precision

       legitimate       0.94
         phishing       0.93
```

Fig. 9. Output of the results obtained using the Random Forest algorithm

The recognition accuracy on the training dataset is 0.973, or 97.3%. On the testing dataset the recognition accuracy is 0.933, which is 93.3%. Figure 9 displays the result output obtained using the RF algorithm.

4.5 Model Development Using the Decision Tree Algorithm

To create a model using the DT algorithm, you first need to import the ML package for the DT algorithm from Scikit-Learn.

When creating the model the code line contains a hyper parameter "max_depth = 30". This hyper parameter in the Decision Tree algorithm affects the complexity of the entire algorithm. A higher value of this hyper parameter makes model creation and data processing more complex. When selecting a value it's essential to be cautious because it

```
     Decision Tree : Accuracy on training Data: 0.979
     Decision Tree : Accuracy on test Data: 0.923


[17] print(metrics.classification_report(y_test, y_test_tree))

                  precision

       legitimate       0.93
         phishing       0.91
```

Fig. 10. Output of the results obtained using Decision Tree algorithm

can lead to underfitting and overfitting, which can impact the performance and accuracy of the created model [24]. The recognition accuracy for identifying phishing addresses is 0.91, which is 91%. In the training phase of the model the accuracy percentage is 97%. Figure 10 displays the output results obtained during the model creation. The model creation was done according to the instructions and example provided in the source [25].

4.6 Model Development Using the KNN Algorithm

KNN is a supervised machine learning algorithm that is based on determining the similarity between data points and categorizing them into separate clusters. After loading, the model creation process is executed. This model is characterized by the hyper parameter "n_neighbor = 1". This hyper parameter represents the number of neighbors that will vote on which group a data point being processed belongs to. The default initial value is 5 and it is recommended to use an odd number to avoid having an equal number of votes during distribution [26]. Figure 11 illustrates the model training process and the selection of the value "1" for the "n_neighbor" hyper parameter.

```
[16] knn = KNeighborsClassifier(n_neighbors=1)

[17] knn.fit(X_train,y_train)

     KNeighborsClassifier
     KNeighborsClassifier(n_neighbors=1)

[18] y_train_knn = knn.predict(X_train)
     y_test_knn = knn.predict(X_test)
```

Fig. 11. The process of selecting the hyperparameters value and training/testing the model

The achieved recognition results for phishing internet addresses using the model created with the KNN algorithm are as follows: training phase recognition accuracy: 0.971 (97.1%) and testing phase recognition accuracy: 85%. The result report for this model is found in Fig. 12. All models were created following the example from the source [25].

```
K-Nearest Neighbors : Accuracy on training Data: 0.971
K-Nearest Neighbors : Accuracy on test Data: 0.854

print(metrics.classification_report(y_test, y_test_knn))

              precision

  legitimate       0.88
    phishing       0.83
```

Fig. 12. Output of the results obtained using KNN algorithm

5 Research Results Analysis

All models were created using the same preprocessed dataset including the same number of attributes and addresses. The same data distribution was applied to all models with 80% of the data used for the model training phase and 20% for the testing phase.

5.1 Analysis of Recognition Accuracy Results

Table 1 provides an overview of the obtained recognition accuracy results during the model's training and testing phases.

Table 1. Summary of Resarch results

Machine learning Algorithm	Training Phase Accuracy	Testing Phase Accuracy
Logistic Regression	88.54%	90.24%
Support Vector Machine	92.60%	91.00%
Random Forest	97.30%	93.30%
Decision Tree	97.90%	91.00%
K-Nearest Neighbors	97.10%	85.00%

When looking at each model individually the algorithm with the smallest difference between the accuracy achieved during the testing and training phases is the SVM algorithm. The difference between the training and testing phases for the SVM algorithm is 1.6%. In the second place is the LR algorithm with a difference of 1.7% between the two phases. After the LR algorithm, the Random Forest algorithm has the next largest difference, which is 4%. The DT algorithm has a 6.9% difference between the results achieved in the testing and training phases, while the KNN algorithm has the largest difference of 12.1%. The DT algorithm achieved the highest recognition accuracy percentage during the training phase with a recognition accuracy of 97.90%.

During the testing phase with the remaining 20% of the dataset the model created using the RF algorithm achieved the highest recognition accuracy percentage. The accuracy percentage of that model is 93.3%. The poorest results in the testing phase were obtained using the KNN model which had a recognition accuracy of 85%.

If we consider the average accuracy when combining the results from the training and testing phases the best results are achieved by the model created using the RF algorithm with a recognition accuracy of 95.30%. Close behind the RF algorithm, the DT model has an average accuracy of 94.45% for the two phases. Figure 13 provides a graphical representation of the obtained accuracy results.

Figure 14 provides a graphical representation of how the precision of phishing URL detection changes for models created using the KNN algorithm (top) and the Random Forest algorithm (bottom) as the hyper parameters "n_neighbors" and "n_estimators" change. In the upper part of the image it is visible that with a hyper parameter value of "1" during the training phase, the precision is above 0.96, specifically 0.971, while with

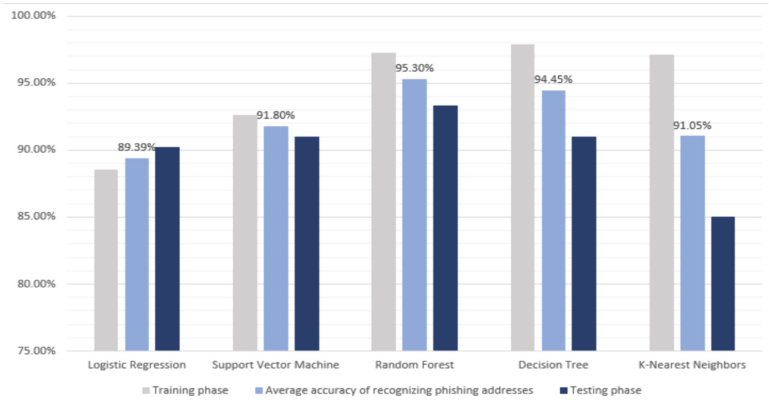

Fig. 13. Graphical representation of the accuracy results

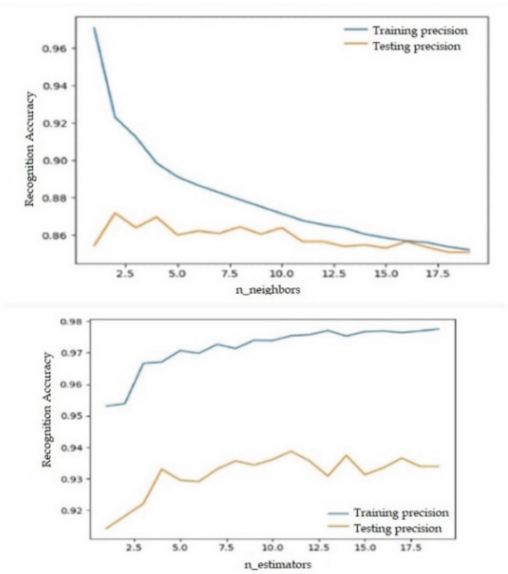

Fig. 14. Graphical representation of relationship between hyperparameters in the KNN and RF models

an increased value of "n_neighbors = 15" the precision significantly decreases to a value of 0.86. In the case of the RF algorithm (bottom), the hyper parameter "n_estimators" represents the number of trees within the forest. The default value of 10 was used during model creation, resulting in a precision of 0.973 for the training phase and 0.933 for the testing phase.

5.2 Analysis of Recall Results

The recall result is a value that indicates the ability to correctly predict positives out of all actual positives and this value is also known as sensitivity. The difference between precision and recall values is that precision measures the final percentage of correct predictions, while recall provides information about how many actual positive instances the model correctly identified as positive. A higher recall value indicates that the ML model is better because it has greater sensitivity to recognition [27]. Table 2 shows the obtained recall values for the created models.

The best sensitivity was achieved by the RF algorithm model with a recognition rate of 93% for legitimate addresses among legitimate addresses and 94% for phishing addresses.

Table 2. Display of recall value results of the created models

Machine learning Algorithm/Recall Value	Legitimate Addresses	Phishing Addresses
Logistic Regression	89%	91%
Support Vector Machine	91%	93%
Random Forest	93%	94%
Decision Tree	91%	93%
K-Nearest Neighbors	82%	89%

5.3 F1-Score Analysis

The F1-score represents the harmonic mean of precision and recall values. It is important because it provides information about the quality of the model's output. It allows insight into the model's performance and the possibility of optimizing precision or sensitivity based on the user's needs.

The mathematical expression for calculating the F1-score is as follows:

$$F1 = \frac{2 * \textbf{Precision} * \textbf{Recall}}{(\textbf{Precision} + \textbf{Recall})}$$

Table 3 shows the F1-score results obtained using the mathematical formula mentioned earlier. The results indicate that the model created using the RF algorithm has the highest and best F1-score value, which is 94.4%.

Table 3. Display of F1-score results of created models

Machine learning Algorithm	F1-score
Logistic Regression	89.6%
Support Vector Machine	91.4%
Random Forest	94.4%
Decision Tree	93.2%
K-Nearest Neighbors	87.8%

6 Conclusion

The paper provides an overview of the previous results of applying AI as a means of protection against cyber threats. By reviewing past research, it can be concluded that the potential for using AI methods such as ML and DL is wide-ranging. By utilizing ML and DL algorithms, models can be developed as part of systems like IDS for detecting unauthorized intrusions into systems. Additionally, through the analysis of previous research, various possibilities for applying algorithms in the detection of phishing attacks are evident, which can be recognized in multiple ways by using DT and ML models.

The models created in this paper were trained using five ML algorithms: DT, KNN, SVM, RF and LR. The best average recognition result for phishing URLs from the downloaded and processed dataset was achieved by the model created using the RF algorithm.

The developed models have shown high efficiency in performing the given task, but there is room for improvement. To further enhance the recognition of potential malicious addresses, the dataset used for model training could be enhanced with additional attributes. Although it currently has 89 attributes, additional attributes could be added to improve the model's accuracy. One such attribute could be domain rating. Which could indicate the rating of a web address and influence its authenticity and non-malicious nature.

References

1. CISA: What is Cybersecurity? (2021). https://www.cisa.gov/news-events/news/what-cybersecurity. Accessed Apr 2024
2. Dasgupta, D., Akhtar, Z., Sen, S.: Machine learning in cybersecurity: a comprehensive survey. J. Def. Model. Simul. Appl. Methodol. Technol. **19**(1), 57–106 (2022)
3. Kolla, J., Praneeth, S., Sameed Baig, M., Reddy Karri, G.: A comparison study of machine learning techniques for phishing detection **4**(1) (2022). Advances on business and information system (E-ISSN: 2685-2543)
4. Peraković, D., Periša, M., Cvitić, I., Zorić, P., Kuljanić, T.M., Aleksić, D.: Opportunities of using machine learning methods in telecommunications and industry 4.0 – a survey. In: Knapčíková, L., Peraković, D. (eds.) MMS 2022, pp. 211–225. EAI/SICC. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-22719-6_16

5. Chawla, A.: Phishing website analysis and detection using machine learning. Int. J. Intell. Syst. Appl. Eng. (2022). ISSN:2147- 6799
6. Springboard: What Is Cybersecurity? A Complete Overview Guide. O. Agbeleye (2023). https://www.springboard.com/blog/cybersecurity/what-is-cybersecurity/. Accessed June 2023
7. IT Governance: How to Spot a Phishing Email: With Examples. L. Irwin. https://www.itgovernance.co.uk/blog/5-ways-to-detect-a-phishing-email. Accessed June 2023
8. Aleksić, D.: Using artificial intelligence in cyber security [diploma thesis]. University of Zagreb, Faculty of Transport and Traffic Sciences, Zagreb (2023). Dostupno na. https://urn.nsk.hr/urn:nbn:hr:119:446211. Accessed May 2024
9. ENISA: Intelligence and Cybersecurity Research.2023. ENISA Research and Innovation Brief
10. k-Means Advantages and Disadvantages. https://developers.google.com/machine-learning/clustering/algorithm/advantagesdisadvantages. Accessed June 2023
11. Medium: K-Means cluster and it's use case in Cyber Security. Saha A. (2021). https://arnabsaha1.medium.com/k-means-cluster-and-its-use-case-in-cyber-security3abfaab2ec09. Accessed June 2023
12. Markey, J.: Using decision tree Analysis for Intrusion Detection: A How-To-Guide. https://sansorg.egnyte.com/dl/6edQgfwngE. Accessed June 2023
13. Ghanem, K., Aparicio-Navarro, F.J., Kyriakopoulos, K.G., Lambotharan, S., Chambers, J.A.: Support Vector Machine for Network Intrusion and Cyber-Attack Detection. https://core.ac.uk/download/288367383.pdf. Accessed June 2023
14. AWS: What IS OVerfitting? https://aws.amazon.com/what-is/overfitting/
15. Jha, J., Ragha, L.: Intrusion detection system using support vector machine. Int. J. Appl. Inf. Syst. (IJAIS). ISSN: 2449-0868. https://research.ijais.org/icwac/number3/icwac1342.pdf
16. CERT.hr. SPAM. https://www.cert.hr/19795-2/spam/
17. Sharma, A., Rastogi, V.: Spam filtering using k mean clustering with local feature selection classifier. Int. J. Comput. Appl. **108**(10) (2014). ISSN 0975-8887. https://research.ijcaonline.org/volume108/number10/pxc3900096.pd
18. Simplilearn: What is Deep Learning and How Does It Works. Reyes K. (2023). https://www.simplilearn.com/tutorials/deep-learning-tutorial/what-is-deeplearning#what_is_deep_learning. Accessed July 2023
19. AWS Amazon: What is Deep Learning? https://aws.amazon.com/whatis/deep-learning/. Accessed June 2023
20. Coursera: Deep Learning vs. Machine learning: Beginner's Guide (2023). https://www.coursera.org/articles/ai-vs-deep-learning-vs-machine-learning-beginners-guide. Accessed June 2023
21. geeksforgeeks: Introduction to Deep Learning. https://www.geeksforgeeks.org/introduction-deep-learning/. Accessed June 2023
22. Khempetch, T., Wuttidittachotti, P.: DDoS attack detection using deep learning. IAES Int. J. Artif. Intell. (IJ-AI) **10**(2), 382–338 (2021). https://ijai.iaescore.com/index.php/IJAI/article/view/20884/13116
23. MendeleyData: Web page phishing detection. https://data.mendeley.com/datasets/c2gw7fy2j4/2. Accessed June 2023
24. Scikit Learn: Importance of decision tree hyperparameters on generalization. https://inria.github.io/scikit-learn-mooc/python_scripts/trees_hyperparameters.html. Accessed May 2023
25. GitHub: Phishing-Website-Detection-by-Machine-Learning-Tehniques. https://github.com/shreyagopal/Phishing-Website-Detection-by-Machine-LearningTechniques/tree/master. Accessed May 2023

26. WordPress: Posts Tagged 'KneighborsClassifier explained'. https://ashokharnal.wordpress.com/tag/kneighborsclassifier-explained/. Accessed June 2023
27. Vitalflux: Accuracy, Precision, Recall & F1-Score – Python Examples. A. Kumar (2023). https://vitalflux.com/accuracy-precision-recall-f1-score-pythonexample/#Terminologies_%E2%80%93_True_Positive_False_Positive_True_Negative_False_Negative. Accessed Aug 2023