

Review

Not peer-reviewed version

An Investigation into the Performances of the State-of-the-art Machine Learning Approaches for Various Cyber-attack Detection: A Survey

[Tosin Ige](#)*, Christophet Kiekintveld, Aritran Piplai

Posted Date: 27 February 2024

doi: 10.20944/preprints202402.1567.v1

Keywords: Cyberattack; SQL attack; Drive-By attack; Malware Attack; Phishing Attack; cyberattack detection; Machine Learning; Machine Learning Algorithms



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Review

An Investigation into the Performances of the State-of-the-Art Machine Learning Approaches for Various Cyber-Attack Detection: A Survey

Tosin Ige *, Christophet Kiekintveld and Aritran Piplai

dept. of Computer Science, The University of Texas at El Paso, Texas, USA; cdkiekintveld@utep.edu (C.K.);
apiplai@utep.edu (A.P.)

* Correspondence: toige@miners.utep.edu

Abstract: To secure computers and information systems from attackers taking advantage of vulnerabilities in the system to commit cybercrime, several methods have been proposed for real-time detection of vulnerabilities to improve security around information systems. Of all the proposed methods, machine learning had been the most effective method in securing a system with capabilities ranging from early detection of software vulnerabilities to real-time detection of ongoing compromise in a system. As there are different types of cyberattacks, each of the existing state-of-the-art machine learning models depends on different algorithms for training which also impact their suitability for detection of a particular type of cyberattack. In this research, we analyzed each of the current state-of-the-art machine learning models for different types of cyberattack detection from the past 10 years with a major emphasis on the most recent works for comparative study to identify the knowledge gap where work is still needed to be done with regard to detection of each category of cyberattack.

Keywords: cyberattack; SQL attack; drive-by attack; malware attack; phishing attack; cyberattack detection; machine learning; machine learning algorithms

1. Introduction

The fact that modern systems are not perfect guarantees that there will always be vulnerabilities no matter how small which could be exploited by an attacker to have unauthorized access which will enable him to violate security policy. Every modern system has vulnerabilities that could be exploited because exploitation could be developed for any system whose vulnerabilities could be described, hence, attacks are easily developed the moment such vulnerabilities are found. It is for this reason that finding vulnerabilities that are previously undiscovered as one of the proven ways to be a hacker elite is a strong cybersecurity culture. At the same time, an exploit is an attack through the vulnerability of a computer system with the purpose of causing either denial-of-service (DoS), install malware such as ransomware, Trojan horses, worms, spyware and so on. The result of a successful attack is what leads to security breach which is an unauthorized access to entity in the cyberspace, this often result in loss of confidentiality, integrity, or availability of data and information, as the attacker is able to remove or manipulate sensitive information.

On the other hand, modern attacker have developed a sophisticated social engineering technique by the use of low level of technology attack such as impersonation, bribes, lies, tricks, threats, and blackmail in order to compromise computer system. Social engineering usually relies on trickery for information gathering and the aim is to manipulate people to perform action(s) which will lead to the attacker getting confidential information of the person or organization. Phishing attack falls into this category of attack because the attacker use trick that can eventually lead the victim into divulging sensitive and personal information which attacker can use to gain access to server, compromise organization system, or commit various cyber crime which includes but not limited to business e-mail compromise (BEC), phishing, malware attack, denial of service (DDoS) attack, Eavesdropping Attacks, Ransomware attack and so on.

In order to secure computer and information systems from attacker taken advantage of vulnerabilities in system to commit cybercrime, several methods had been adopted for earlier detection of vulnerabilities as well as quick or real time detection of compromise in computer information system space to improve security around computer and information system. In all the methods machine learning had been the most effective methods in securing system with capability ranging from early detection of software vulnerabilities to real-time detection of ongoing compromise in a system. Each of the existing machine learning classifier models depends on different algorithms such as Support Vector Machine (SVM), Logistic Regression (LR), Naïve Bayes classifier, deep learning based, decision tree, random forest, XGBoost and so on, for which they are suitable for different kind of cybersecurity-related classification tasks. Having observed the under-performance of Naïve Bayes variants in comparison with another suitable classifier for the same cybersecurity-related tasks such as phishing detection, anomaly detection in network intrusion, software vulnerability detection, malware detection, and so on, several work had been done by to improve the performance of Naïve Bayes classifier by other researchers. In this review paper, we analyzed both the performance and result of various proposals from the past 10 years to address the underperformance of Naïve Bayes-based classifier to better understand the current state-of-art of Bayesian-based classifier.

2. Background Study

Technological innovation has led to different methods of securing a system, and so, an appropriate attack technique had to be applied for an attack to be successful. It is for this reason that experienced attacker takes time to study a system to understand it to determine the right approach of a successful attack on a targeted system. Hence, it is imperative to summarize attack methodologies for each of the current state-of-the-art attack techniques as it keeps evolving to understand why some state-of-the-art defense methods remain vulnerable. In this section, we summarized techniques and how some of the major attacks are being perpetrated by cybercriminals.

2.1. Distributed Denial of Service (DDoS) Attack

In a distributed denial of service attack, the resources of a targeted system are maliciously flooded from multiple systems at a time to disrupt the normal traffic flow of the targeted server, service, or network, in most cases the traffic is synthetically generated by the attacker to maliciously overwhelm the system which eventually leads to denial of service as the system is unable to deliver service to legitimate user. It uses brute force attack [1] which can be triggered through Botnet when any of the devices of the network environment are actively infected with malware. DDoS attacks can be classified into three main categories which could be (i) Traffic/fragmentation attack, (ii) Bandwidth/Volume attack and (iii) Application attack [1] depending on the nature, severity of the attack, or form of the attack.

2.2. Malware Attack

Malware attack is a type of cyberattack that are mostly introduced through phishing, social engineering, and downloads in which unauthorized actions by malicious software commonly known as a virus are executed on the victim's system. Each malicious software has different malicious activities ranging from stealing sensitive data, launching DDoS attacks [2], conducting ransomware, or pushing unwanted adverts in the case of adware attacks, these specialized activities determine the type of malware attack which in some cases might be command and control, spyware, ransomware, adware and so on.

2.3. Man in the Middle (MITM) Attack

Man-in-the-middle[a] (MITM) attack is a type of cyberattack where an attacker secretly relays or alters communications between two parties. By intercepting and relaying the messages back to the intending receiver, the attacker makes the sender and the receiver believe they are directly

communicating with each other. It is a sort of eavesdropping where the attacker connects with the victim's network to relay incoming and outgoing messages which could be altered by the attacker before relaying back to the receiver [3]. Man-in-the-browser is the most common MITM attack where the attacker could inject malicious proxy malware into the victim's device through browser infection. MITN attack is carried out by data interception involving the interception of a data transfer between client and server and the actual decryption of the messages. Malware that is involved in this type of attack is mostly introduced to the victim's system by phishing email where the goal is to steal sensitive and other personal information from the victim.

2.4. Drive-by Attack

A drive-by attack is the malicious infection of a computer with malware when a victim visits an infected website. Merely visiting an infected website is enough to get the malware downloaded and running on a system unknown to the victim. This type of attack exploits vulnerabilities in the victim's browser application to successfully infect the computer. Still, they become obsolete when a security patch for the browser application is released. The drive-by attack uses different malware distribution techniques, sophistication, and attack intensity which can go unnoticed for a long time [4] and with the ability to cause significant damage to a system, as the process involves attacking a legitimate website by injecting malicious code into the pages to compromise it so that the injected code could be loaded into the victim's system through his browser whenever a user browses the compromised legitimate website thereby initializing the drive-by attack

2.5. SQL Injection Attack

SQL injection is an application layer attack through which hackers steal sensitive data by inserting malicious SQL statements into the input field of an application for execution such as the input of SQL query to dump the database contents to the attacker. In this type of attack, the aim of the attacker is the usage of malicious SQL code for backend database manipulation to access information that is not supposed to be displayed [5] such as private customer details, company data, user's list, and so on.

2.6. Phishing Attack

Phishing is a type of cybercrime in which an individual is lured to divulging sensitive information details through text message, email, or phone conversation by someone posing either as a legitimate institution or a member of a legitimate institution, some of these commonly requested sensitive details which are social security number, password, credit, and banking card details etc are later used to access more sensitive information [6] for different type of cybercrime which often results in financial loss or identity theft as about 76% of the phishing attacks were credential-harvesting in 2022 according to Digital Information world. A California teenager was able to get sensitive information to access credit card details and withdraw money from his victim's account through his fake "America Online" website which resulted in the first lawsuit filed in 2004. Efficient phishing detection has been challenging as attackers continue to advance their tactics as technologies evolve. To defraud personnel, all an attacker needs to do is simply clone a legitimate website to create a new website (SCAM Website) which is then used to defraud computer users.

3. Category of Current State of the art Phishing Detection model

3.1. Bayesian-Based-Classifier

Naive Bayes is a family of probabilistic-based algorithms that is based on the Bayes rule. It is based on the fact that, if B has occurred, we can find the probability that A will occur. B is taken to be the evidence while the hypothesis is A and with a strong assumption that each of the features is independent. It uses the prior probability distribution to predict the posterior probability of a sample

that belongs to a class. In this process, the class with the highest probability is then selected as the final predicted class [7]. Naive Bayes updates prior belief of an event occurring given that there is new information. Hence, given the availability of new data, the probability of the selected sample occurring is given by;

$$P(class/features) = \frac{P(class) * P(features/class)}{P(features)}$$

Where

- $P(class/features)$: Posterior Probability
- $P(class)$: Class Prior Probability
- $P(features/class)$: Likelihood
- $P(features)$: Predictor Prior Probability

It has a very strong assumption of independency which affects its performance for classification tasks[8] as the strong assumption of independence among features is not always valid in most of the dataset that is used to train the current state-of-the-art model for several classification tasks. The strong assumption of the Naive Bayes classifier is one reason why it usually underperforms when compared with its peers for similar classification tasks. Naive Bayes classifier has different variants with each variant having its own individual assumption which also impacts its performance in addition to the general assumption of independence which is common to all variants of the Naive Bayes classifier, and so each variant is suitable for different classification tasks.

Multinomial Naive Bayes is a variant of Naive Bayes, It assumes multinomial distribution among features of dataset in addition to the general assumption of independency, and so its performance is affected if the actual distribution is not multinomial or partially multinomial. Multinomial Naive Bayes is the suitable variant for natural language processing classification task [9] but still underperforms when compared with non-bayesian and deep learning-based classifiers for the same NLP classification task.

Gaussian Naive Bayes is the suitable Bayesian variant for anomaly detection in network intrusion which could be used to detect Distributed Denial of Service (DDOS) attacks [8]. It assumes the normal distribution among features in dataset in addition to the general assumption of independence which is common to all variants of Naive Bayes. The probability density of the normal distribution in Gaussian Naive Bayes is such that:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \quad (1)$$

Where

- ' μ ' is the mean or expectation of the distribution,
- ' σ ' is the standard deviation, and
- ' σ^2 ' is the variance.

Despite being a suitable Naive Bayes variant for anomaly detection, it still underperforms when compared with its suitable peer for detection of Distributed Denial of Service (DDOS) attack as evident in the work done by Rajendran [10] where Gaussian Naive Bayes have the least accuracy of 78.75% compared with other non-bayesian based for attack detection classification task.

Bernoulli Naive Bayes assumes Bernoulli distribution in addition to the assumption of independence. Its main feature is that it only accepts binary values such as success or failure, true or false, and yes or no as input while complement Naive Bayes is used for imbalance datasets as no single variant of Naive Bayes can do the task of all the variants. Both the suitability and performance of each variant are determined by their individual assumption in addition to the general assumption of independence which impacts their performance when compared with their suitable peer for the same classification task.

3.2. Non-Bayesian Based Classifier

3.2.1. Decision Tree

A decision Tree is a Supervised learning technique whose operation is based on a tree-structured classifier, with features in the dataset being represented by an internal node, each decision rule is represented by the branches, while the internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the decision outcome is represented by the leaf node and so does not have further branches. It makes a decision-based graphical representation of all possible solutions to a problem. It uses the Classification and Regression Tree algorithm (CART) [11] to construct a decision tree starting with the root node whose branch keeps expanding further to construct a tree-like structure. It is a non-parametric and the ultimate goal is the creation of a machine learning model capable of making prediction by learning simple decision rules that are inferred from data features.

3.2.2. Random Forest

It is an ensemble-based learning algorithm that could be used for classification, regression task, and other similar tasks that operates based on the construction of multiple decision trees [12]. Since the algorithm works by constructing multiple decision trees during training, the output of a classification model trained with a random forest algorithm is the class selected by most of the trees, while the mean or average prediction of individual trees is returned as the output for a regression task. This system of aggregating and ensembling with multiple trees for prediction makes it possible for a random forest-trained model to outperform the decision tree-trained model and also avoid overfitting which is a peculiar problem for decision tree classifiers.

3.2.3. Logistic Regression

Logistic regression is the modeling of the probability of a discrete outcome by having the event log-odds be a linear combination of one or more independent variables given an input variable [13]. Logit transformation is applied to the bounded odds which is the division between the probability of success and probability of failure. It is a linear regression that could be used for both classification and regression tasks and since the output is a probability, the dependent variable is bounded between 0 and 1 values, it uses logistic function to model binary output for classification problems. The difference between linear regression and logistic regression is that the range in logistic regression is bounded by 0 and 1, and also that logistic regression does not require a linear relationship between input and output.

3.2.4. XGBoost

It is a supervised learning algorithm that is gradient boosting based. It is extremely efficient and highly scalable, the algorithm works by first creating a series of individual machine learning models and then combining each of the previously created models to form an overall model that is more accurate and efficient than any of the previously created individual models in the series. This system of creating a series of models and combining them to create a single model [14] makes XGBoost perform better than other state-of-the-art machine learning algorithms in many classification, ranking, several user-defined prediction problems, and regression tasks across several domains. XGboost uses gradient descent to add additional individual models to the main model for prediction, hence it is also known as stochastic gradient boosting, gradient boosting machines, or multiple additive regression trees.

3.2.5. K-Nearest Neighbor (KNN)

k-nearest neighbors (kNN) algorithm is a non-parametric supervised learning algorithm that uses the principle of similarity to predict the label or value of a new data point by considering values of its K-nearest neighbors in the training dataset based on a distance metric like Euclidean distance.

$$\text{dist}(x, z) \leq \text{dist}(x, y) + \text{dist}(y, z) \quad (2)$$

for which the distance between x and z could be calculated by

$$d(x, z) = \sqrt{\sum_{i=1}^n (x_i - z_i)^2} \quad (3)$$

The prediction of the new data point is based on the average or majority vote of its neighbor, this method allows the classifier to adapt its prediction according to the local structure of the data which ultimately helps to improve its overall accuracy and flexibility. Since KNN can be used for both classification and regression tasks, its prediction output depends on the type of task (classification or regression). In the case of a classification task, it uses class membership as the output by using the plurality vote of its neighbor to assign the input to the class that is most common among its k nearest neighbors, but when KNN is being used for a regression task, it uses the average of the values of k nearest neighbors as the prediction output, the value of k has an impact on the overall accuracy [15] of the model.

3.2.6. Support Vector Machine (SVM)

Support Vector Machine (SVM) is a supervised machine algorithm that works by looking for a hyper-plane that creates a boundary between two classes of data to solve classification and regression-related problems [16]. It uses the hyper-plane to determine the best decision boundary between different categories in the training dataset, hence they can be applied to vectors that could encode data. Two theories must hold before we can determine the suitability of SVM for certain classification or regression tasks, the first is the availability of high-dimension input space as SVM tries to prevent overfitting by using an overfitting protective measure which is independent of the number of features in the data gives SVM the potential to handle feature spaces in the dataset. The second theory is the presence of linearly separable properties of categorization in the training dataset, and this is because SVM works by finding linear separators between each of the categories to make accurate predictions.

3.3. Deep Learning Based Classifier

3.3.1. Convolutional Neural Network (CNN)

CNN is a deep learning model with a grid pattern for processing data that is designed to automatically and adaptively learn spatial hierarchies of features, from low- to high-level patterns [17,18]. It is a mathematical construct that is composed of convolution, pooling, and fully connected layers as three types of layers or building blocks responsible for different tasks for predictions. While convolution and pooling layers, perform feature extraction, the fully connected layer, maps the extracted features into the final output usually known as classification. The convolution layer is composed of mathematical operations (convolution) which plays a very crucial role in Convolutional Neural Networks as in a kind of linear operation. The CNN architecture is a combination of several building blocks like convolution layers, pooling layers, and fully connected layers, and so, a typical architecture consists of repetitions of a stack of many convolution layers and a pooling layer, and then followed by one or more fully connected layers. It stored digital images, and pixel values as a two-dimensional (2D) grid which is an array of numbers along with some parameters called the

kernel before an optimizable feature extractor is finally applied at each image position. This makes CNNs a highly efficient classifier for image processing classification tasks, since a feature may occur anywhere in the image. extracted features can hierarchically and progressively become more complex as each layer progressively feeds its output to the next layer, the main task is the minimization of differences between output and ground truth by backward propagation and gradient descent which is an optimization algorithm. This process of optimizing parameters like kernels to minimize the difference between outputs and ground truth is called training.

3.3.2. Recurrent Neural Network (RNN)

Recurrent Neural Networks (RNNs) is a type of Neural Network in which output from the previous step is fed to the current step as input, It introduce the concept of memory to neural networks through the addition of the dependency between data points. This addition of dependency between data points ensured that RNNs could be trained to remember concepts by able able to learn repeated patterns. The main difference between RNN and the traditional neural network is the concept of memory in RNN which is made possible as a result of the feedback loop in the cell. Here, it is the feedback loop that enables the possibility of passing information within a layer unlike in feedforward neural networks where information can only be passed between layers. While input and output are independent of each other in a traditional neural network, It is a different ball game in RNN where sequence information is to be remembered, this was made possible in RNN by its Hidden state also known as the memory state through which it remembers previous input to the network, and so it is safe to conclude that the most important features of RNNs is the Hidden state by which it remembers some information in a sequence. In terms of architecture, RNN architecture is the same as that of other deep neural networks, the main difference lies in how the information flows from the input to the output. While the weight across the network in RNN is the same, deep neural network has different weight matrices for each dense network. The Hidden state in the RNNs which enables them to remember sequence information makes it suitable for natural language processing tasks.

3.3.3. Long Short-Term Memory (LSTM)

Long short-term memory (LSTM) network is a recurrent neural network (RNN) that is specifically designed to handle sequential data, such as speech, text, and time series, it is aimed at solving the problem of vanishing gradient in traditional RNNs. It is insensitive to gap length which gives it an advantage over hidden Markov models, hidden Markov models, and other RNNs. It provides a short-term memory for RNN which can last thousands of timesteps thereby making it a "long short-term memory" network. A single LSTM network unit is composed of an output gate, a cell, an input gate, and a forget gate. While the three gates regulate the flow of information into and out of the cell, the cell is responsible for remembering values over arbitrary time intervals as the Forget gates decide on the information to discard from a previous state by assigning a previous state, compared to a current input which assigns a value between 0 and 1. A value of 1 means the information is to be kept, and a value of 0 means the information is to be discarded. The Input gates decide on the exact pieces of new information to store in the current state in the same way as forget gates. Output gates consider both the previous and current states to control which pieces of information in the current state are to output by assigning a value from 0 to 1 to the information. This selective outputting of relevant information from the current state allows the LSTM network to utilize both useful and long-term dependencies in making more accurate predictions in current and future time steps. The fact that they are designed to learn long-term dependencies in sequential data makes them suitable for time series forecasting, speech recognition, and language translation tasks.

4. Literature Review

Several methods have been proposed for each of the major categories of cyberattack ranging from Malware, phishing, Man In The Middle, SQL Injection, and Drive-by attack detection and

to mitigate the effect of each category of cyberattack with different results and efficiency. These methods are classified based on the different methodologies of the algorithm which can be classified as Bayesian-based, non-Bayesian-based, and deep learning-based. Each of these categories of classification has different accuracy and efficiency for phishing detection and prevention tasks with several underlying causes. In this section, we reviewed and explained existing state-of-the-art phishing detection techniques to identify weak spots where improvement is needed to increase efficiency and project future research direction.

Mayank Agarwal et al. [19] proposed IDS that adheres to the 802.11 standard which does not require any protocol modification to detect the existence of flooding-based DoS attacks in a Wi-Fi network and to quickly recover from the attack. The proposed intrusion detection system is independent of client software and only requires hardware that is capable of sniffing the wireless data to ensure cost-effectiveness.

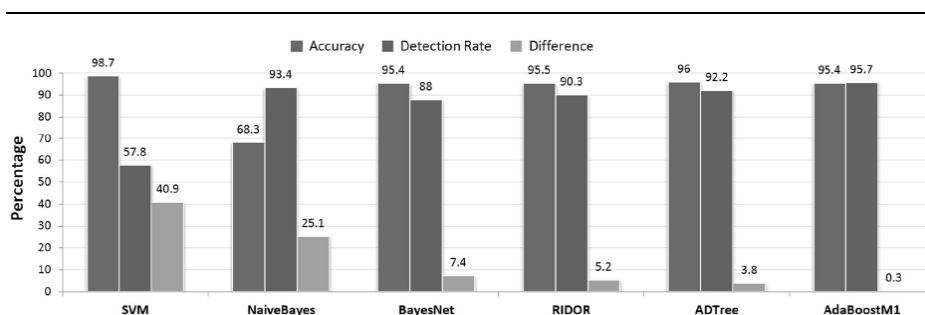


Figure 1. MITM Attack detection rate, accuracy, and differences from different classifiers from Mayank Agarwal et al [19].

For man-in-the-middle Attack Detection 1, Naive Bayes has the weakest overall performance with an accuracy of 68% [19], while another probabilistic classifier in Bayes Net has significant improvement in performance over Naive Bayes with accuracy of 95% and a detection rate of 88% while SVM algorithm has the best performance in terms of accuracy with accuracy rate of 98.7% but with abysmal detection rate of 57.8% which is extremely poor for an intrusion detection system.

Muhanna Saed and Ahamed Aljuhani [20] proposed a set of machine learning techniques both to detect and identify Man-In-The-Middle attacks on a wireless communication network. In addition, validation and evaluation were based on performance metrics, as well as performance comparison with other machine learning-based Man-In-The-Middle attack detection methods. By training traditional machine learning-based and deep learning-based models with a set of data that represents the transmission of data over a wireless network for MITM attack detection, the deep learning-based model which was trained with Long short-term memory (LSTM) network as a recurrent neural network (RNN) as an accuracy of 92%, Support vector machine (SVM) having an accuracy of 85%, while random forest with accuracy of 94%.

Ann Zeky et al [21] proposed extraction based naive bayes model for phishing detection with emphasis on the extraction of relevant features like unusual characters, spelling mistakes, domain names and URL analyzation with substantial success albeit with an imbalance dataset and susceptibility to bayesian poisoning.

Patricia Iglesias [22] used an artificial neural network for the detection of drive-by attacks on polyglot payloads in the image, however, the results obtained are only limited to the successful detection of stego images by the use of LSB and F5 steganographic methods. Hence, they propose deep learning techniques through the use of convolutional Neural Network as a suitable method of detection in a situation where both the malicious content and the images are delivered by Spatial and Steganography algorithms. The proposed CNN model was evaluated with benchmark image databases along with collections of JavaScript exploits which yielded an AUC validation score of 99.75% and an accuracy of 98.61%.

mahdi bahaghighat et al [23] proposed phishing detection method based on Logistic Regression, K-Nearest Neighbors, Naive Bayes, Random Forest, Support Vector Machine, and Extreme Gradient Boosting (XGBoost) algorithm relying solely on the attribute feature of the webpage URL. The experiment resulted in the outperformance of other algorithms by XGBoost, Random Forest, KNN with accuracy of 99.2%, 98.1% and 98.3% respectively while naive bayes has the worst performance with accuracy of 93% for phishing detection of all the algorithm echoing result obtained by Kamal Omari [24] on the performance of naive bayes algorithm for phishing detection.

Morufu Olalere et al. [25] proposed a naïve Bayes model for effective categorization and detection of SQL injection attacks, they obtained 98% and 99% for detection and categorization respectively by validating the proposed model with stratified cross validation having 1-10 random seeds which is a significant leap when compared with previous studies and performance, although we don't know the actual distribution of different categories of SQL injection attack type in the dataset, and also how the proposed Naive Bayes model will perform on KAGGLE SQL injection attack dataset.

While assuming the absence of a single solution to detect most phishing attacks, and to evaluate performances of Bayesian classifier for phishing detection tasks based on different feature selection techniques. Twana Mustafa and Murat Karabatak [26] developed 6 bayesian based models in which each model involves a single feature selection technique chosen from individual FS, forward FS, Backward FS, Plus-I takeaway-r FS, AR1, and All to compare the performance of bayesian classifier for phishing detection task based on different feature selection techniques. The experiment resulted in the Bayesian model with Plus-I takeaway-r feature selection having the best performance with an accuracy of 93.39% while the Bayesian classifier with individual feature selection technique has the least performance with an accuracy of 92.05%, and so concluded that feature selection has a direct impact on classifier accuracy.

To investigate the performance of different classifiers in detecting SQL Injection attacks, Prince Roy et al. [27] trained multiple models with Logistic Regression, AdaBoost (Adaptive Boosting), Random Forest, Naive Bayes, and XGBoost (Extreme Gradient Boosting) Classifier on Kaggle SQL Injection Dataset, and concluded with Naive Bayes as the best classifier for SQL injection attack detection with an accuracy of 98.33% echoing similar conclusion from Morufu Olalere [25]. Logistic Regression provided 92.73% accuracy, Adaboost provided 90.35% accuracy, XGBoost provided 89.64%, and Random Forest provided 92.14% accuracy.

Table 1. Comparison Analysis of Methodologies for Detection of SQL Injection Attack.

Classifier	Authors	Mean Score
Naive Bayes	[27], [28], [29], [30], [31], [32], [33]	90.4
SVM	[28], [29], [34], [30], [32], [35], [33]	87.63
Random Forest	[27], [28], [32], [35], [36], [37], [33]	93.71
Logistic Regression	[27], [30], [35], [38], [37], [33], [39]	89.68
KNN	[32], [37], [33], [40], [41], [39], [42]	87.2
Decision Tree	[35], [29], [36], [33], [40], [41], [43]	90.04

Table 2. Comparison Analysis of Methodologies for Detection of DDoS Attack.

Classifier	Authors	Mean Score
SVM	[44], [45], [46], [47], [48], [49], [50]	90.0
Naive Bayes	[44], [51], [46], [52], [53], [54], [47]	84.6
Random Forest	[55], [51], [46], [56], [52], [53], [57]	93.34
Decision Tress	[55], [44], [46], [56], [47], [49], [58]	96
XGBoost	[55], [56], [50], [57], [59], [60]	96.2
KNN	[44], [56], [47], [61], [62], [63], [58]	96.5

Table 3. Comparison Analysis of Methodologies for Detection of Phishing Attack.

Classifier	Authors	Mean Score
Naive Bayes	[23], [64], [65], [66], [24], [67], [68], [69], [70], [71], [72], [73], [74], [75], [76]	80.431
SVM	[23], [64], [66], [24], [77], [67], [78], [79], [80], [81], [82], [75]	89.429
Random Forest	[23], [64], [65], [66], [24], [83], [77], [67], [80], [84], [78], [85], [82]	97.065
Decision Tree	[64], [66], [24], [77], [67], [78], [80], [86], [82], [70], [71], [75]	95.248
Logistic Regression	[23], [64], [24], [79], [80], [87], [85], [82], [88], [69], [70]	92.589
KNN	[23], [64], [66], [24], [67], [84], [79], [80], [87], [85], [89], [82], [71], [75]	90.479

Table 4. Limitations of current state-of-the-art Approaches for cyberattack detection.

Author	Dataset	Research Summary	Method/Algorithm	Limitation
Mayank Agarwal et al.[19]	Internally generated dataset	Proposal of IDS to detect the existence of flooding-based DoS attacks in a Wi-Fi network	Naive Bayes, Bayes Net, Ridor, ADTree	1. The dataset used is not publicly available 2. An unknown number of records in the generated dataset
Ali Bin et al. (2022) [90]	Linux distribution having 3855 attacks	evaluation of machine learning classifiers for the detection of man-in-the-middle attacks	Decision Tree Naive Bayes K-Nearest Neighbor(KNN) Artificial Neural Network XGBoost	1. The effect of feature variation on the classifier was not investigated 2. The result obtained is limited to only one dataset
Muhanna Saed and Ahamed Aljuhani (2022) [20]	Wi-Fi network benchmark dataset	Proposal machine learning techniques both to detect and identify Man-In-The-Middle attacks on a wireless communication network	LSTM, Support Vector Machine, Random Forest	1. proposed model likely to be bias due to imbalance dataset 2. Imbalance dataset as attack 2 packet is above 12000, attack 1 is less than 500, while normal packet is about 2000.
Kamal Omari (2023) [24]	UCI phishing domains dataset	an investigation into the performances and efficiency of Logistic Regression, KNN, SVM, Naive Bayes, Decision Tree, Random Forest, and Gradient Boosting for phishing detection task	naive bayes KNN SVM Random Forest Gradient Boost Decision Tree Logistic Regression	1. While Random Forest have a good accuracy of 97.1%, its complexity of re-generating tree still remains 2. Heavy reliance on URL attributes which means the model remains vulnerable
Prince Roy et al. (2022) [90]	Kaggle SQL Injection attack dataset	evaluation of machine learning classifiers for the detection of SQL Injection attacks	Logistic Regression Naive Bayes XGBoost Random Forest Adaboost	1. Limited only to SQL injection, does not address other web application vulnerabilities
Ann Zeky et al (2023) [21]	Internally generated dataset	proposal of extraction based naive Bayes robust model for phishing detection with emphasis on the combination of webpage content and URL feature analysis	naive Bayes URL analyzation webpage content extraction	1. problem of bayesian poisoning was not addressed and so the model remains vulnerable to bayesian poisoning
Patricia Iglesias et al. [22]	Coco and ILSVR dataset	proposal of deep learning based CNN model for detecting drive-by road attack	Convolutional Neural Network	limited to drive-by road attack detection in stego images
mahdi bahaghighat (2023) [23]	Grega Vrbančić phishing dataset	performance comparison of phishing detection method based on several six different algorithm	naive bayes KNN SVM Random Forest Gradient Boost Logistic Regression	1. complexity of re-generating tree for every output in random forest remains 2. sole reliance on URL feature attributes means the model remains vulnerable to friendly URL
Nishitha U et al (2023) [91]	Kaggle Phishing Dataset	performance comparison of machine learning and deep learning based algorithm for phishing detection	CNN RNN KNN Random Forest Decision Tree Logistic Regression	1. 5000 records is too small to train a CNN model and so no confidence here 2. imbalance in the dataset will lead to bias sole reliance on URL feature
Prince Roy et al. (2022) [90]	Kaggle SQL Injection attack dataset	evaluation of machine learning classifiers for the detection of SQL Injection attacks	Logistic Regression Naive Bayes XGBoost Random Forest Adaboost	1. Limited only to SQL injection, does not address other web application vulnerabilities
Twana Mustafa and Murat Karabatak (2023) [26]	UCL Phishing Dataset	Performance Comparison of different Bayesian classifier based on different Feature Selection Algorithm	naive bayes individual FS forward FS backward FS Plus-I takeaway-r FS ARI FS	1. each of the Bayesian model remains vulnerable to Bayesian poisoning no hint on why Plus-I takeaway-r FS works better best for naive bayes
Morufu Olalere	dataset was internally generated from dmoz	proposal of Naive Bayes model for detecting SQL injection attack	Naive Bayes	SQL syntax distribution in the dataset was not known, It is difficult to know how the proposed model will perform on Kaggle SQL injection dataset
Jaya T et al (2023) [92]	UK-2011, SpamBase, and Spam Assassin datasets	usage of frequency weightage of the words for unsupervised clustering of mail into spam and ham messages	naive bayes random forest logistic regression random tree LTSM	1. while random forest performed well, its complexity of generating tree for every output remains a problem possible reason for the poor performance of Bayesian classifier remains to be investigated
Santhosh Raminedi et al (2023) [64]	Internally collected phishing dataset	evaluation of several machine learning and deep learning based algorithms for phishing detection using URL features	ANN SVM KNN and Naive Bayes Random Forest Decision Tree Logistic Regression	1. complexity of generating tree for every output in random forest was not addressed 2. sole reliance on URL feature

5. Analysis and Discussion

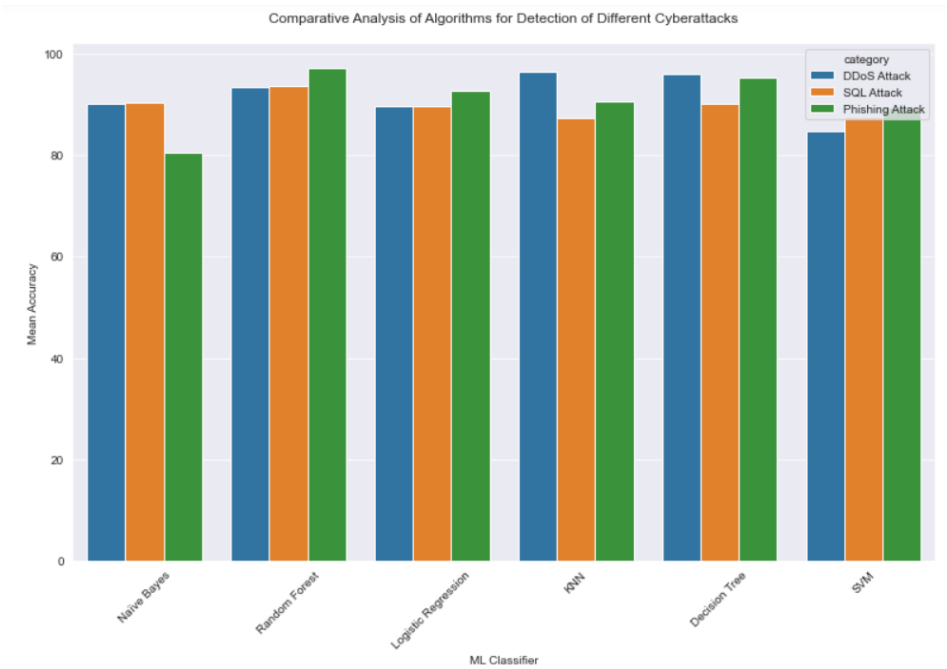


Figure 2. Comparative Analysis of Algorithms for Detection of Different Cyberattacks.

1. Insufficient research on the capability of machine learning algorithms for the detection of drive-by downloads, man-in-the-middle, and Malware attacks.

We got very few research papers where a machine learning algorithm was used to train a model for the detection of drive-by downloads, man-in-the-middle, and Malware attacks or any of their combination. The few research papers where machine learning algorithms were used to train models for the detection of drive-by download or man-in-the-middle were so few that comparing the result with the performance of machine learning algorithms for the detection of other categories of cyberattack will lead to severe bias on the result by tilting it against the performance of ML models in the detection of other categories of cyberattacks. Research papers where machine learning algorithms were used to detect drive-by download attacks were much more scanty than the other two. Hence, we chose to remove them from the relative comparison table as more research work where various machine learning algorithms are used to detect drive-by download attack are still required. We don't know why the detection of drive-by download attacks using machine learning algorithms is so scanty, and so this is open for investigation and further research.

2. Mix performance by Naive Bayes

An observation of the performance of Naive Bayes across different categories of cyberattack is very interesting knowing fully well that Naive Bayes is a parametric-based machine learning algorithm whose prediction is based on (i)the assumption of independence between features and (ii) the distribution of features in a dataset which might be Multinomial, Bernoulli, or Normal distribution. We expect Naive Bayes to have relatively consistent performance across different categories of cyberattacks based on the assumption but it was surprising to see Naive Bayes having a very strong performance in the detection of SQL Injection attacks while having the weakest performance in the detection of phishing attacks in relative comparison to other machine learning classifier. Understanding why Naive Bayes algorithm performs extremely well in the detection of SQL injection

attacks but relatively poor performance in the detection of phishing attacks requires further research.

3. Limitation of Current Approach to SQL Injection Attack Detection

The current approach to the detection of Structured Query Language Injection (SQLi) attacks is solely based on the presence of SQL statements in user input, and while this approach had been successful in finding the presence of SQL statements in user input to protect backend database against SQL injection attack, the approach cannot detect an already compromised database. Naive Bayes and Random Forest are the best-performing machine learning algorithms with mean accuracies of 90.4% and 93.71% respectively for the detection of SQLi attacks based on the current approach of finding the presence of SQL statements from user input. Hence, with a failure of 9.6% for Naive Bayes and 6.3% for Random Forest algorithms, a database can still be compromised, hence machine learning model needs to have the capability to predict a compromised database immediately an SQLi attack scale through. There is no single study on the detection of compromise of a database by a machine learning model, hence, this is an interesting area that requires further research.

6. Conclusion and Research Direction

In this research, we did a comprehensive survey of current state-of-the-art machine learning algorithms to investigate their effectiveness and suitability for the detection of different categories of cyberattacks, to ensure that our research reflects the latest advancement at the intersection of artificial intelligence and cybersecurity, we categorized and discussed various methodologies, techniques, and approaches from research papers that are from the past 10 years but predominantly from the last 5 years. We also reviewed the effectiveness and limitations of recent proposals and novel frameworks in the detection of cyberattacks. Our finding shows the need for; further research and exploration on the use of a machine learning approach for the detection of drive-by download attacks, an investigation into the mix performance of Naive Bayes to identify possible research direction on improvement to existing state-of-the-art Naive Bayes classifier, we also identify the need for an improvement to the current machine learning approach to the detection of SQLi attack because existing machine learning approach cannot detect an already compromised database with SQLi attack.

References

1. Sambangi, S.; Gondi, L. A machine learning approach for ddos (distributed denial of service) attack detection using multiple linear regression. *Proceedings. MDPI*, 2020, Vol. 63, p. 51.
2. Gopinath, M.; Sethuraman, S.C. A comprehensive survey on deep learning based malware detection techniques. *Computer Science Review* **2023**, *47*, 100529.
3. Michelena, Á.; Aveleira-Mata, J.; Jove, E.; Bayón-Gutiérrez, M.; Novais, P.; Romero, O.F.; Calvo-Rolle, J.L.; Aláiz-Moretón, H. A novel intelligent approach for man-in-the-middle attacks detection over internet of things environments based on message queuing telemetry transport. *Expert Systems* **2023**, p. e13263.
4. Smailes, J.; Salkield, E.; Birnbach, S.; Strohmeier, M.; Martinovic, I. Dishing out DoS: How to Disable and Secure the Starlink User Terminal. *arXiv preprint arXiv:2303.00582* **2023**.
5. Crespo-Martínez, I.S.; Campazas-Vega, A.; Guerrero-Higueras, Á.M.; Riego-DelCastillo, V.; Álvarez-Aparicio, C.; Fernández-Llamas, C. SQL injection attack detection in network flow data. *Computers & Security* **2023**, *127*, 103093.
6. Okomayin, A.; Ige, T.; Kolade, A. Data Mining in the Context of Legality, Privacy, and Ethics **2023**.
7. Wang, Z.; Yao, L.; Shao, X.; Wang, H. A combination of TEXTCNN model and Bayesian classifier for microblog sentiment analysis. *Journal of Combinatorial Optimization* **2023**, *45*, 109.
8. Ige, T.; Kiekintveld, C. Performance Comparison and Implementation of Bayesian Variants for Network Intrusion Detection. *arXiv preprint arXiv:2308.11834* **2023**.
9. Ige, T.; Adewale, S. AI powered anti-cyber bullying system using machine learning algorithm of multinomial naïve Bayes and optimized linear support vector machine. *arXiv preprint arXiv:2207.11897* **2022**.

10. Rajendran, T.; Abishekraj, E.; Dhanush, U. Improved Intrusion Detection System That Uses Machine Learning Techniques to Proactively Defend DDoS Attack. *ITM Web of Conferences*. EDP Sciences, 2023, Vol. 56, p. 05011.
11. Zhu, E.; Ju, Y.; Chen, Z.; Liu, F.; Fang, X. DTOF-ANN: an artificial neural network phishing detection model based on decision tree and optimal features. *Applied Soft Computing* **2020**, *95*, 106505.
12. HR, M.G.; MV, A.; others. Development of anti-phishing browser based on random forest and rule of extraction framework. *Cybersecurity* **2020**, *3*, 1–14.
13. Edgar, T.W.; Manz, D.O. Chapter 4 - Exploratory Study. In *Research Methods for Cyber Security*; Edgar, T.W.; Manz, D.O., Eds.; Syngress, 2017; pp. 95–130. doi:https://doi.org/10.1016/B978-0-12-805349-2.00004-2.
14. Gu, J.; Xu, H. An ensemble method for phishing websites detection based on XGBoost. 2022 14th international conference on computer research and development (ICCRD). IEEE, 2022, pp. 214–219.
15. Assegie, T.A. K-nearest neighbor based URL identification model for phishing attack detection. *Indian Journal of Artificial Intelligence and Neural Networking* **2021**, *1*, 18–21.
16. Guo, B.; Zhang, C.; Liu, J.; Ma, X. Improving text classification with weighted word embeddings via a multi-channel TextCNN model. *Neurocomputing* **2019**, *363*, 366–374.
17. Yamashita, R.; Nishio, M.; Do, R.K.G.; Togashi, K. Convolutional neural networks: an overview and application in radiology. *Insights into imaging* **2018**, *9*, 611–629.
18. Ige, T.; Marfo, W.; Tonkinson, J.; Adewale, S.; Matti, B.H. Adversarial Sampling for Fairness Testing in Deep Neural Network. *arXiv preprint arXiv:2303.02874* **2023**.
19. Agarwal, M.; Pasumarthi, D.; Biswas, S.; Nandi, S. Machine learning approach for detection of flooding DoS attacks in 802.11 networks and attacker localization. *International Journal of Machine Learning and Cybernetics* **2016**, *7*, 1035–1051.
20. Saed, M.; Aljuhani, A. Detection of man in the middle attack using machine learning. 2022 2nd International Conference on Computing and Information Technology (ICCIT). IEEE, 2022, pp. 388–393.
21. Magdady Jerjes, A.Z.A.; Dawod, A.Y.; Abdulqader, M.F. Detect Malicious Web Pages Using Naive Bayesian Algorithm to Detect Cyber Threats. *Wireless Personal Communications* **2023**, pp. 1–13.
22. Iglesias, P.; Sicilia, M.A.; García-Barriocanal, E. Detecting browser drive-by exploits in images using deep learning. *Electronics* **2023**, *12*, 473.
23. Bahaghighat, M.; Ghasemi, M.; Ozen, F. A high-accuracy phishing website detection method based on machine learning. *Journal of Information Security and Applications* **2023**, *77*, 103553.
24. Omari, K. Comparative Study of Machine Learning Algorithms for Phishing Website Detection. *International Journal of Advanced Computer Science and Applications* **2023**, *14*. doi:10.14569/IJACSA.2023.0140945.
25. Olalere, M.; Egigogo, R.A.; Ojeniyi, J.A.; Ismaila, I.; Jimoh, R.G. A Naïve Bayes Based Pattern Recognition Model for Detection and Categorization of Structured Query Language Injection Attack **2018**.
26. Mustafa, T.; Karabatak, M. Feature Selection for Phishing Website by Using Naive Bayes Classifier. 2023 11th International Symposium on Digital Forensics and Security (ISDFS). IEEE, 2023, pp. 1–4.
27. Roy, P.; Kumar, R.; Rani, P. SQL injection attack detection by machine learning classifier. 2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC). IEEE, 2022, pp. 394–400.
28. Xie, X.; Ren, C.; Fu, Y.; Xu, J.; Guo, J. Sql injection detection for web applications based on elastic-pooling cnn. *IEEE Access* **2019**, *7*, 151475–151481.
29. Deriba, F.; Salau, A.O.; Mohammed, S.H.; Kassa, T.M.; Demilie, W.B. Development of a compressive framework using machine learning approaches for SQL injection attacks **2022**. *1*, 183–189.
30. Krishnan, S.A.; Sabu, A.N.; Sajan, P.P.; Sreedeeep, A. SQL injection detection using machine learning. *vol* **2021**, *11*, 11.
31. Pattewar, T.; Patil, H.; Patil, H.; Patil, N.; Taneja, M.; Wadile, T. Detection of SQL injection using machine learning: a survey. *Int. Res. J. Eng. Technol.(IRJET)* **2019**, *6*, 239–246.
32. Sivasangari, A.; Jyotsna, J.; Pravalika, K. SQL injection attack detection using machine learning algorithm. 2021 5th International Conference on Trends in Electronics and Informatics (ICOEI). IEEE, 2021, pp. 1166–1169.
33. AL-Maliki, M.H.A.; Jasim, M.N. Comparison study for NLP using machine learning techniques to detecting SQL injection vulnerabilities. *International Journal of Nonlinear Analysis and Applications* **2023**.

34. Hasan, M.; Balbahaith, Z.; Tarique, M. Detection of SQL injection attacks: a machine learning approach. 2019 International Conference on Electrical and Computing Technologies and Applications (ICECTA). IEEE, 2019, pp. 1–6.
35. Hosam, E.; Hosny, H.; Ashraf, W.; Kaseb, A.S. Sql injection detection using machine learning techniques. 2021 8th International Conference on Soft Computing & Machine Intelligence (ISCMI). IEEE, 2021, pp. 15–20.
36. Tripathy, D.; Gohil, R.; Halabi, T. Detecting SQL injection attacks in cloud SaaS using machine learning. 2020 IEEE 6th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing,(HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS). IEEE, 2020, pp. 145–150.
37. Alam, A.; Tahreen, M.; Alam, M.M.; Mohammad, S.A.; Rana, S. SCAMM: Detection and prevention of SQL injection attacks using a machine learning approach. PhD thesis, Brac University, 2021.
38. Arumugam, C.; Dwarakanathan, V.B.; Gnanamary, S.; Neyveli, V.N.; Ramesh, R.K.; Kandhavel, Y.; Balakrishnan, S. Prediction of SQL Injection Attacks in Web Applications. Computational Science and Its Applications–ICCSA 2019: 19th International Conference, Saint Petersburg, Russia, July 1–4, 2019, Proceedings, Part IV 19. Springer, 2019, pp. 496–505.
39. Bhardwaj, A.; Chandok, S.S.; Bagnawar, A.; Mishra, S.; Uplaonkar, D. Detection of cyber attacks: Xss, sqli, phishing attacks and detecting intrusion using machine learning algorithms. 2022 IEEE Global Conference on Computing, Power and Communication Technologies (GlobConPT). IEEE, 2022, pp. 1–6.
40. Adebisi, M.O.; Arowolo, M.O.; Archibong, G.I.; Mshelia, M.D.; Adebisi, A.A. An SQL injection detection model using chi-square with classification techniques. 2021 International Conference on Electrical, Computer and Energy Technologies (ICECET). IEEE, 2021, pp. 1–8.
41. Hashem, I.; Islam, M.; Haque, S.M.; Javed, Z.I.; Sakib, N. A proposed technique for simultaneously detecting DDoS and SQL injection attacks. *Int. J. Comput. Appl* **2021**, *183*, 50–57.
42. Irungu, J.; Graham, S.; Girma, A.; Kacem, T. Artificial Intelligence Techniques for SQL Injection Attack Detection. Proceedings of the 2023 8th International Conference on Intelligent Information Technology, 2023, pp. 38–45.
43. Ingre, B.; Yadav, A.; Soni, A.K. Decision tree based intrusion detection system for NSL-KDD dataset. Information and Communication Technology for Intelligent Systems (ICTIS 2017)-Volume 2 2. Springer, 2018, pp. 207–218.
44. Suresh, M.; Anitha, R. Evaluating machine learning algorithms for detecting DDoS attacks. Advances in Network Security and Applications: 4th International Conference, CNSA 2011, Chennai, India, July 15-17, 2011 4. Springer, 2011, pp. 441–452.
45. Kyaw, A.T.; Oo, M.Z.; Khin, C.S. Machine-learning based DDOS attack classifier in software defined network. 2020 17th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON). IEEE, 2020, pp. 431–434.
46. He, Z.; Zhang, T.; Lee, R.B. Machine learning based DDoS attack detection from source side in cloud. 2017 IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud). IEEE, 2017, pp. 114–120.
47. Alzahrani, R.J.; Alzahrani, A. Security analysis of ddos attacks using machine learning algorithms in networks traffic. *Electronics* **2021**, *10*, 2919.
48. Pei, J.; Chen, Y.; Ji, W. A DDoS attack detection method based on machine learning. Journal of Physics: Conference Series. IOP Publishing, 2019, Vol. 1237, p. 032040.
49. Tuan, T.A.; Long, H.V.; Son, L.H.; Kumar, R.; Priyadarshini, I.; Son, N.T.K. Performance evaluation of Botnet DDoS attack detection using machine learning. *Evolutionary Intelligence* **2020**, *13*, 283–294.
50. Mohmand, M.I.; Hussain, H.; Khan, A.A.; Ullah, U.; Zakarya, M.; Ahmed, A.; Raza, M.; Rahman, I.U.; Haleem, M.; others. A machine learning-based classification and prediction technique for DDoS attacks. *IEEE Access* **2022**, *10*, 21443–21454.
51. Saini, P.S.; Behal, S.; Bhatia, S. Detection of DDoS attacks using machine learning algorithms. 2020 7th International Conference on Computing for Sustainable Global Development (INDIACom). IEEE, 2020, pp. 16–21.
52. Ajeetha, G.; Priya, G.M. Machine learning based DDOS attack detection. 2019 Innovations in Power and Advanced Computing Technologies (i-PACT). IEEE, 2019, Vol. 1, pp. 1–5.

53. Robinson, R.R.; Thomas, C. Ranking of machine learning algorithms based on the performance in classifying DDoS attacks. 2015 IEEE Recent Advances in Intelligent Computational Systems (RAICS). IEEE, 2015, pp. 185–190.
54. Zekri, M.; El Kafhali, S.; Aboutabit, N.; Saadi, Y. DDoS attack detection using machine learning techniques in cloud computing environments. 2017 3rd international conference of cloud computing technologies and applications (CloudTech). IEEE, 2017, pp. 1–7.
55. Al-Juboori, S.A.M.; Hazzaa, F.; Jabbar, Z.S.; Salih, S.; Gheni, H.M. Man-in-the-middle and denial of service attacks detection using machine learning algorithms. *Bulletin of Electrical Engineering and Informatics* **2023**, *12*, 418–426.
56. Gaur, V.; Kumar, R. Analysis of machine learning classifiers for early detection of DDoS attacks on IoT devices. *Arabian Journal for Science and Engineering* **2022**, *47*, 1353–1374.
57. Chen, Z.; Jiang, F.; Cheng, Y.; Gu, X.; Liu, W.; Peng, J. XGBoost classifier for DDoS attack detection and analysis in SDN-based cloud. 2018 IEEE international conference on big data and smart computing (bigcomp). IEEE, 2018, pp. 251–256.
58. Ramadhan, I.; Sukarno, P.; Nugroho, M.A. Comparative analysis of K-nearest neighbor and decision tree in detecting distributed denial of service. 2020 8th International Conference on Information and Communication Technology (ICoICT). IEEE, 2020, pp. 1–4.
59. Rozam, N.F.; Riasetiawan, M. XGBoost Classifier for DDOS Attack Detection in Software Defined Network Using sFlow Protocol. *International Journal on Advanced Science, Engineering & Information Technology* **2023**, *13*.
60. Dhaliwal, S.S.; Nahid, A.A.; Abbas, R. Effective intrusion detection system using XGBoost. *Information* **2018**, *9*, 149.
61. Doshi, R.; Aphorpe, N.; Feamster, N. Machine learning ddos detection for consumer internet of things devices. 2018 IEEE Security and Privacy Workshops (SPW). IEEE, 2018, pp. 29–35.
62. Yusof, A.R.; Udzir, N.I.; Selamat, A. An evaluation on KNN-SVM algorithm for detection and prediction of DDoS attack. Trends in Applied Knowledge-Based Systems and Data Science: 29th International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2016, Morioka, Japan, August 2-4, 2016, Proceedings 29. Springer, 2016, pp. 95–102.
63. Nguyen, H.V.; Choi, Y. Proactive detection of DDoS attacks utilizing k-NN classifier in an anti-DDoS framework. *International Journal of Computer and Information Engineering* **2010**, *4*, 537–542.
64. Raminedi, S.; Pandey, T.N.; Woonna, V.A.; Mascarenhas, S.C.; Bharani, A. Classification of Phishing Websites using Machine Learning Models. 2023 3rd International conference on Artificial Intelligence and Signal Processing (AISP). IEEE, 2023, pp. 1–5.
65. Yaswanth, P.; Nagaraju, V. Prediction of Phishing Sites in Network using Naive Bayes compared over Random Forest with improved Accuracy. 2023 Eighth International Conference on Science Technology Engineering and Mathematics (ICONSTEM). IEEE, 2023, pp. 1–5.
66. Karim, A.; Shahroz, M.; Mustofa, K.; Belhaouari, S.B.; Joga, S.R.K. Phishing Detection System Through Hybrid Machine Learning Based on URL. *IEEE Access* **2023**, *11*, 36805–36822.
67. Al Ahasan, M.A.; Hu, M.; Shahriar, N. OFMCDM/IRF: A Phishing Website Detection Model based on Optimized Fuzzy Multi-Criteria Decision-Making and Improved Random Forest. 2023 Silicon Valley Cybersecurity Conference (SVCC). IEEE, 2023, pp. 1–8.
68. Al Fayoumi, M.; Odeh, A.; Keshta, I.; Aboshgifa, A.; AlHajahjeh, T.; Abdulraheem, R. Email phishing detection based on naïve Bayes, Random Forests, and SVM classifications: A comparative study. 2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC). IEEE, 2022, pp. 0007–0011.
69. Ab Razak, M.F.; Jaya, M.I.; Ernawan, F.; Firdaus, A.; Nugroho, F.A. Comparative Analysis of Machine Learning Classifiers for Phishing Detection. 2022 6th International Conference on Informatics and Computational Sciences (ICICoS). IEEE, 2022, pp. 84–88.
70. Ozker, U.; Sahingoz, O.K. Content based phishing detection with machine learning. 2020 International Conference on Electrical Engineering (ICEE). IEEE, 2020, pp. 1–6.
71. Uddin, M.M.; Islam, K.A.; Mamun, M.; Tiwari, V.K.; Park, J. A Comparative Analysis of Machine Learning-Based Website Phishing Detection Using URL Information. 2022 5th International Conference on Pattern Recognition and Artificial Intelligence (PRAI). IEEE, 2022, pp. 220–224.

72. Rodríguez, J.E.R.; García, V.H.M.; Castillo, N.P. Webpages classification with phishing content using naive Bayes algorithm. *Knowledge Management in Organizations: 14th International Conference, KMO 2019, Zamora, Spain, July 15–18, 2019, Proceedings 14*. Springer, 2019, pp. 249–258.
73. Shabudin, S.; Sani, N.S.; Ariffin, K.A.Z.; Aliff, M. Feature selection for phishing website classification. *International Journal of Advanced Computer Science and Applications* **2020**, *11*.
74. Sadaf, K. Phishing Website Detection using XGBoost and Catboost Classifiers. *2023 International Conference on Smart Computing and Application (ICSCA)*. IEEE, 2023, pp. 1–6.
75. Alrefaai, S.; Özdemir, G.; Mohamed, A. Detecting Phishing Websites Using Machine Learning. *2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*. IEEE, 2022, pp. 1–6.
76. Korkmaz, M.; Sahingoz, O.K.; Diri, B. Detection of phishing websites by using machine learning-based URL analysis. *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*. IEEE, 2020, pp. 1–7.
77. Alnemari, S.; Alshammari, M. Detecting phishing domains using machine learning. *Applied Sciences* **2023**, *13*, 4649.
78. Rashid, S.H.; Abdullah, W.D. Enhanced website phishing detection based on the cyber kill chain and cloud computing. *Indonesian Journal of Electrical Engineering and Computer Science* **2023**, *32*, 517–529.
79. Arivukarasi, M.; Manju, A.; Kaladevi, R.; Hariharan, S.; Mahasree, M.; Prasad, A.B. Efficient Phishing Detection and Prevention Using Support Vector Machine (SVM) Algorithm. *2023 IEEE 12th International Conference on Communication Systems and Network Technologies (CSNT)*. IEEE, 2023, pp. 545–548.
80. Pandey, P.; Mishra, N. Phish-Sight: a new approach for phishing detection using dominant colors on web pages and machine learning. *International Journal of Information Security* **2023**, pp. 1–11.
81. Vallepu, R.; Karunakaran, M. An innovative method to improve performance analysis in classification with accuracy of phishing websites using random forest algorithm by comparing with support vector machine algorithm. *AIP Conference Proceedings*. AIP Publishing, 2023, Vol. 2655.
82. Khan, M.F.; Tiwari, R.K.; Saroj, S.K.; Tripathi, T. A Comparative Study of Machine Learning Techniques for Phishing Website Detection. In *Role of Data-Intensive Distributed Computing Systems in Designing Data Solutions*; Springer, 2023; pp. 97–109.
83. Almseidin, M.; Zuraiq, A.A.; Al-Kasassbeh, M.; Alnidami, N. Phishing detection based on machine learning and feature selection methods **2019**.
84. Aldakheel, E.A.; Zakariah, M.; Gashgari, G.A.; Almarshad, F.A.; Alzahrani, A.I. A Deep Learning-Based Innovative Technique for Phishing Detection in Modern Security with Uniform Resource Locators. *Sensors* **2023**, *23*, 4403.
85. Rugangazi, B.; Okeyo, G. Detecting Phishing Attacks Using Feature Importance-Based Machine Learning Approach. *2023 IEEE AFRICON*. IEEE, 2023, pp. 1–6.
86. Muliono, Y.; Ma'ruf, M.A.; Azzahra, Z.M. Phishing Site Detection Classification Model Using Machine Learning Approach. *Engineering, Mathematics and Computer Science (EMACS) Journal* **2023**, *5*, 63–67.
87. Sunday, S.M. Phishing Website Detection Using Machine Learning: Model Development and Django Integration. *Journal of Electrical Engineering, Electronics, Control and Computer Science* **2023**, *9*, 39–54.
88. Kumar, K.V.; Ramamoorthy, M. Machine Learning-based spam detection using Naïve Bayes Classifier in comparison with Logistic Regression for improving accuracy. *Journal of Pharmaceutical Negative Results* **2022**, pp. 548–554.
89. Borra, S.R.; Gayathri, B.; Rekha, B.; Akshitha, B.; Hafeeza, B. K-NEAREST NEIGHBOUR CLASSIFIER FOR URL-BASED PHISHING DETECTION MECHANISM. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)* **2023**, *14*, 34–40.
90. Sultan, A.B.M.; Mehmood, S.; Zahid, H. Man in the Middle Attack Detection for MQTT based IoT devices using different Machine Learning Algorithms. *2022 2nd International Conference on Artificial Intelligence (ICAI)*. IEEE, 2022, pp. 118–121.

91. Nishitha, U.; Kandimalla, R.; Vardhan, R.M.M.; Kumaran, U. Phishing Detection Using Machine Learning Techniques. 2023 3rd Asian Conference on Innovation in Technology (ASIANCON). IEEE, 2023, pp. 1–6.
92. Jaya, T.; Kanyaharini, R.; Navaneesh, B. Appropriate Detection of HAM and Spam Emails Using Machine Learning Algorithm. 2023 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI). IEEE, 2023, pp. 1–5.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.