electronics

Article

Efficient Dynamic Phishing Safeguard System Using Neural Boost Phishing Protection

Abdul Quadir Md [1], Dibyanshu Jaiswal [1], Jay Daftari [1], Sabireen Haneef [1], Celestine Iwendi [2,*] and Sanjiv Kumar Jain [3]

[1] School of Computer Science and Engineering, Vellore Institute of Technology, Chennai 600127, India
[2] School of Creative Technologies, University of Bolton, Bolton BL3 5AB, UK
[3] Electrical Engineering Department, Medi-Caps University, Indore 453331, India
* Correspondence: c.iwendi@bolton.ac.uk

Abstract: The instances of privacy and security have reached the point where they cannot be ignored. There has been a rise in data breaches and fraud, particularly in banks, healthcare, and government sectors. In today's world, many organizations offer their security specialists bug report programs that help them find flaws in their applications. The breach of data on its own does not necessarily constitute a threat or attack. Cyber-attacks allow cyberpunks to gain access to machines and networks and steal financial data and esoteric information as a result of a data breach. In this context, this paper proposes an innovative approach to help users to avoid online subterfuge by implementing a Dynamic Phishing Safeguard System (DPSS) using neural boost phishing protection algorithm that focuses on phishing, fraud, and optimizes the problem of data breaches. Dynamic phishing safeguard utilizes 30 different features to predict whether or not a website is a phishing website. In addition, the neural boost phishing protection algorithm uses an Anti-Phishing Neural Algorithm (APNA) and an Anti-Phishing Boosting Algorithm (APBA) to generate output that is mapped to various other components, such as IP finder, geolocation, and location mapper, in order to pinpoint the location of vulnerable sites that the user can view, which makes the system more secure. The system also offers a website blocker, and a tracker auditor to give the user the authority to control the system. Based on the results, the anti-phishing neural algorithm achieved an accuracy level of 97.10%, while the anti-phishing boosting algorithm yielded 97.82%. According to the evaluation results, dynamic phishing safeguard systems tend to perform better than other models in terms of uniform resource locator detection and security.

Keywords: data breach; email phishing; internet protocol address; anti phishing neural algorithm; anti phishing boosting algorithm; swish; tracker auditor; attack detection

Citation: Md, A.Q.; Jaiswal, D.; Daftari, J.; Haneef, S.; Iwendi, C.; Jain, S.K. Efficient Dynamic Phishing Safeguard System Using Neural Boost Phishing Protection. *Electronics* **2022**, *11*, 3133. https://doi.org/10.3390/electronics11193133

Academic Editor: Aryya Gangopadhyay

Received: 30 August 2022
Accepted: 26 September 2022
Published: 29 September 2022

1. Introduction

The term phishing originated in 1966 by a faction of fraudsters who were striving to steal accounts and passwords in America. Using email, they enticed people and used their emails as hooks to fish for passwords and other information from the sea of internet users. Phishing is a cybercrime where the target user is contacted through email, telephone, or short message service (SMS) where they pose or show themselves as a legitimate organization, or members of those organizations and deceive the user to provide sensitive or private data [1] such as passwords, bank details, personal details or card details which can be used against the will or information of the user. According to the 2019 phishing and email fraud statistics: 90 percent of data breaches are caused by phishing. Within a year, 15% of those who were successfully phished will be targeted at least once more. Phishing attempts have increased by 65 % in the last year. Each month, around 1.5m new phishing sites are generated. The average financial cost of a data breach is $3.86m (IBM) and these are the statistics for just 2019; with the pandemic ongoing, the cases of phishing in the year 2020 increased exponentially.

1.1. Need for Phishing Security

In today's world, the internet is more popular than ever. It has given the platform to express, influence, and even earn by connecting to people around the world. In the case of internet security, there are many benefits of using the internet and Internet of Things (IoT) devices, however, demerits such as the compromise of user privacy [1], IP spoofing [2], etc., have affected people in various aspects. To deal with these securities, privacy, and other cyber issues, companies are working constantly to protect the rights of the people. For instance, using organizational units in Gcentr, users can adjust security settings according to the preferences they want. All suspicious content can either go to the Spam folder, or it can be left in their inbox with a warning. In addition to endeavours by private companies, legal regulations such as the General Data Protection Regulation (GDPR) had been transmitted by the European Union (EU) to ensure data privacy and security [3].

The proposed paper concentrates on the phishing attack that has helped cyber miscreants to trick people into getting their credentials or esoteric data. This paper aims to proposed an architecture that protects against phishing attacks through a cyber-bulwark. In addition to it, other factors that may ascribe data breach entails permissions to access details such as location, gallery, contacts, or messages that users may give to certain applications or websites. These permissions can be tracked and can be used against the user.

1.2. Data Breaching Decoded

An attack or threat is not caused by a breach of data alone. In reality, a data breach is the result of a cyber-attack, which provides cybercriminals with access to financial and esoteric data [4]. Some of the most common cyber-attacks used to compromise data are spyware, phishing, broken or misconfigured access controls, etc. Cybercriminals are continuously looking for data they may sell, use to break into other accounts, steal identities, or make fraudulent purchases. Typically, most targeted organizations are businesses and medical institutions because they contain enormous amounts of personal and sensitive information [5,6].

1.3. Motivation for Security over Data Breaching

The companies collect the data from different websites or mobile applications using trackers or tracers. Data is collected when the users visit their page so that companies can track that particular user. If data leakage were to happen then the privacy and security of the data that companies have provided us can vanish and this would destroy the private space in the digital world. Stolen data can sometimes end up on the dark web.

To transcend the challenges above and to summarize, the contribution of this paper is as follows:

- To combat the stealing of personal data from major apps and websites, this paper proposes a dynamic phishing safeguard system that gives users the ability to check where their data goes, thereby safeguarding users' privacy. In this way, the users' are assured that their information is being used with their permission, rather than believing that companies are gathering it for their own benefit.
- We propose an algorithm that predicts whether the given Uniform Resource Locator (URL) is phishing or not, using the APNA and APBA.
- Furthermore, a method name tracker auditor is proposed in the paper in which all trackers used by a specific website are visible to the user and the user has the authority to block the website from collecting their data. As a result, they will also avoid clicking on phishing links found on that application.

The rest of this paper is organized as follows: The background and related efforts to detect phishing websites are presented in Section 2. The problem statement is presented in Section 3, and the proposed architecture for dynamic phishing safeguard is described in Section 4. In Section 5, we discuss the experimental setup of DPSS for this study. A final summary and conclusions are provided in Section 6 of this paper.

2. Related Works

Data breaches are becoming prosaic in terms of range and frequency. Data breaches in the US public sector over the last five years were analyzed by the authors in [7]. Only threats to the US public sector which is a global issue are discussed. Cyber miscreants are canny hackers, who are always looking to pivot from one system to another in order to achieve their goals. This involves gaining access to sensitive information by exploiting security weaknesses. An in-depth look into this research [8] reveals common faults that are often linked to data breaches, targeted sectors by cyber criminals according to their relevance, and the actions that need to be taken to improve business cyber security. Moreover, it shows that 45% of security breaches occur through hacking and 22% occur due to targeted members of a company's workforce. The social attack represents the highest percentage followed by theft breaches due to weak credentials and phishing attacks. Email phishing has been a huge impact and with the growing number of attacks there is an approach called "Email Sender Centric Approach" which suggests that while the phishers do a lot in hiding the information they send on the email, they cannot fully hide their information, and sender information can be clearly seen in the emails; people can use this as an advantage to figure out the phishing emails. They have used this approach and have succeeded in gaining a high amount of accuracy in detecting the phishing emails. In study [9], a comprehensive review of the strategies for detecting phishing websites has been given. A comparative analysis of anti-phishing tools in use was completed, and their precincts were approved.

In [10], the author uses URL-based definitions to enhance their definitions. Anti-phishing methods and technologies have always relied on passive methods to gather user submissions and identify phishing URLs. Normally, they are unable to detect and eliminate phishing assaults in a timely and efficient manner. They showed phishing reports in the study and offered a hybrid strategy established on query logs of Domain Name Service (DNS) and recognized phishing URLs to actively detect phishing attacks. They developed and deployed their method to describe incarnate phishing URLs spontaneously to APAC on a daily basis. In the study [11], the authors have proposed a three-level attack recognition system called Web Crawler-based Phishing Attack Detector (WC-PAD). This paper uses web traffic, web content, and the URL as entry parameters to classify phishing and non-phishing websites. Although this research reached a contract on the decisive features that must be utilized in phishing detection, they left out other potential features that can be used for prediction. In paper [12], the author applies Fuzzy Rough Set (FRS) theory as an instrument to choose the most suitable attributes from the three sets of data. The features selected are given to three used phishing detection classifiers. The classifiers are trained using a distinct sample set data of 14,000 website samples to rate the FRS feature selection in developing a generalized phishing detection. Although this paper has achieved good accuracy, there is a scope for further increasing the accuracy by applying different algorithms. In [13], the author uses a polynomial neural network to build a phishing website classifier. Further optimization techniques such as genetic algorithm, gradient descent, and particle swarm optimization are applied; however, it was identified that the genetic algorithm outperforms the former and it is also computationally efficient. Genetic algorithms are algorithms that encode solutions to specific problems by using data structures and simple chromosomal recombination operators to maintain crucial information.

Established on a built neuro-fuzzy framework, [14] employs URL attributes and online traffic attributes to identify phishing websites (dubbed Fi-NFN). The study's findings are based on novel technologies, such as fog computing, as promoted by Cisco. They built an anti-phishing prototype to observe and guard users of fog from phishing assaults which can be practically implemented; however, this is not enough to enable users to feel safe and secure. In [15], the author analyzes the availability of mobile phones that give rise to tools that collect data on users by using algorithms based on data mining. Basically, their key objective is to assure privacy protection in the course of the request of the sociometer. A sociometer is a tool for computing official statistics. They have designed a framework named as PRIMULE which is based on the methodology of prudence, from

the detailed study of data breaches which have occurred since 2005. Moreover, this study shows the vulnerable company and type of attack so the security manager can review it and strengthen their company's security. A data breach can affect both company and user adversely. The uninterrupted impact on the organization is economical and the indirect impact is represented by the formation of a side accountable for getting in touch with the victims and investigating system's breaches. Along with this it involves loss of capital and customers. Taking note after a data breach is common practice. The study [16] shows the voyager tool which tracks the user by storing the IP address along with the browser information of the user who visits the page. The tool will inevitably track users, whether they realize it or not. This tool is made of a Hyper Text Markup Language (HTML) snippet which automatically loads when the URL is accessed. Then this information is used as web analysis or online marketing. In study [17], the author shows the network forensic tool which collects the legal evidence of cybercrime by capturing the packet. It can detect even when the system is formatted or modified. It can identify the source of a cyber-attack by reaching beyond the Internet Service Provider (ISP). This is deployed on Amazon Web Services (AWS).

3. Problem Statement

The importance of information has grown so much that at present, analyzing information provided by clients on websites and other electronic gadgets that gather data can lead to the discovery of whole client profiles, including behavior, exercise, premium, segment, monetary objectives, angles, etc. In order to increase profits, large corporations use this type of data to provide the user with information that they are more interested in. As a result, these firms obtain data that is private and secure, ensuring that it is not used by others for their own gains to blackmail or threaten the user. In recent years, data breaches or leaks have increased, so if this data falls into the wrong hands, there could be a lot of difficulty and a great deal of controversy, since personal data can be utilized and managed without consent. There is no such efficient and robust approach to detect whether a website is a phishing website or not in real-time, along with the tracking ability; where the data is going.

Let us presume a set W consisting of all URLs.

$$W = \{\, w \mid w = x_i \quad , \quad x_i \quad \epsilon \; url,\; i \,\epsilon\, N+\},\; |W| = n$$

Let K_p as a set indicating phishing,

$$K_p = \{\, k \mid k = f,\; f \,\epsilon\, phishing\}$$

K_{np} as a set indicating non-phishing,

$$K_{np} = \{\, k \mid k = np,\; np \,\epsilon\, non - phishing\},\; \text{and } K = K_p \cup K_{np},$$

t_i is an apprehensive URL. Technically, the detection problem of phishing website can be distinct as follows:

Definition 1. *(Phishing Detection of t_i). Suppose P(K) is a power set of K. Defining function $m : W \rightarrow P(K)$, m as a mapping correlation desires to be set up to execute the detection of w_i. Let K_i' as the computed outcome by w, and $K' = \cup_i \epsilon\, nK_i'$, K_i as the classification to the URL w_i, $K_i \epsilon K$,*

$$K_i = \{1\; if \; not \; a \; phishing \; site \; and - 1 \; phishing \; site\}.$$

Detection of phishing website can be termed as:

$$\forall w_i \, \epsilon \, D, \; K_i' \; = m(w_i \,) \tag{1}$$

The objective function is:

$$O(u) = max(\frac{n - \sum_{i=1}^{n} (m(w_i \,) \oplus \; K_i \,)}{(\sum_{i=1}^{n} m(k_i \,))}) \tag{2}$$

The core of unraveling the detection of phishing website problem is to identify an appropriate function w that can find the maximum value for the objective function.

For this reason, we propose a dynamic phishing safeguard system and create an advanced classification approach to identify phishing websites.

4. Proposed Architecture of Dynamic Phishing Safeguard System

A Dynamic Phishing Safeguard System (DPSS) illustrated in Figure 1 has been proposed in this paper to prevent users from phishing attacks. The DPSS architecture helps users to know the following; e.g., is this site legitimate or phishing, where their data is going, what trackers are used on that URL and is their data going to a trusted source or not. The evaluation of the system is based on the accuracy of the incorporated model. Precision, recall, F1-score and accuracy (correct classification rate). The incorporated models in DPSS are compared with other related work on the basis of correct classification rate of other models.

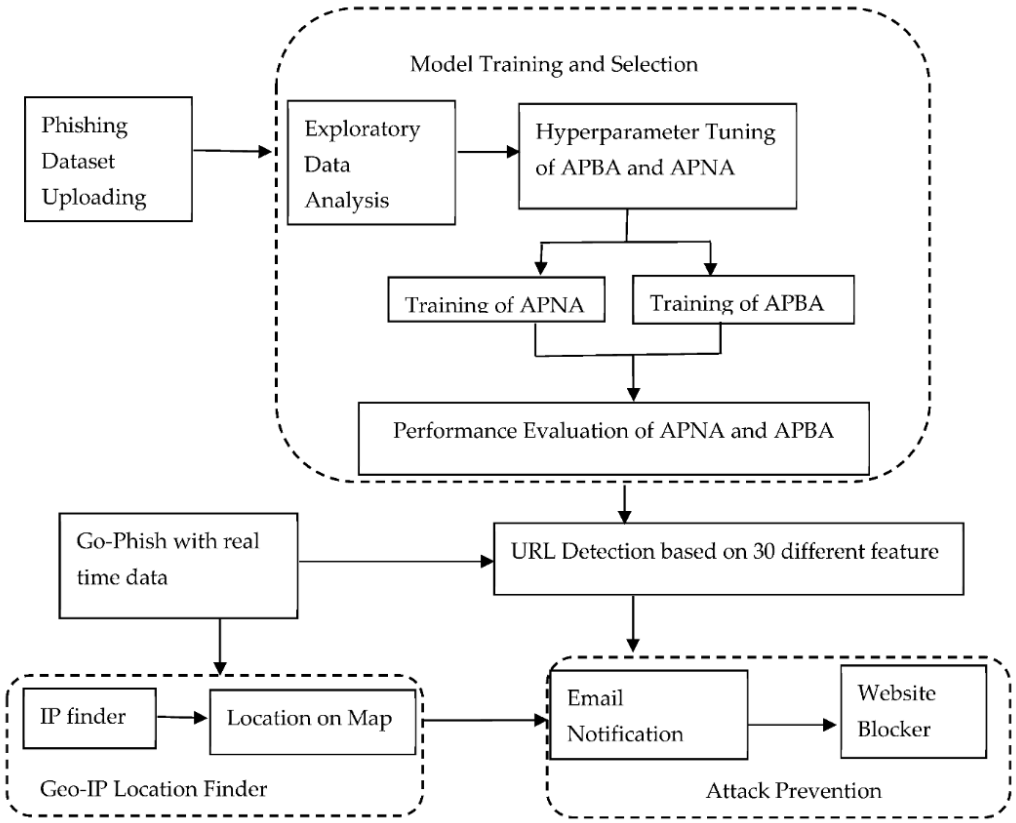

Figure 1. Proposed dynamic phishing safeguard system.

4.1. Attack Generation

Go-Phish is used in real-time to illustrate a process of generating phishing attacks for checking our model. It is an open toolkit that is used to create a phishing email and a website that links to the email. First, email accounts are entered on which a phishing link is sent and that is known as a sending profile page. After setting up the email account there is a connection success message on the email. Then the system moves to make a landing page. A landing page is the HTML page that the user will be redirected to when the user clicks the link. A fake landing page has been created, which looks legitimate so that the user types the credentials which are saved for the admin to access. This landing page helps the admin store the data that users will enter.

Further, the warning shows that the password stored is in data format and not in an encrypted format. Go-Phish can even import a site and make it look the same. Then, the admin moves to generate the convincing subject and body of the email. Go-Phish can also be used for other emails as a reference here to create more likely emails. Now, the admin creates a group of people to send the email. In the end, the campaign is started and the emails are sent to every individual and all the actions by the user regarding email can be tracked. Email movements can be tracked. Further, the system can also keep track of how many people have reported the email or how many have neglected that and entered the data. This is how an email phishing attack is produced.

4.2. Geo-Ip Location Finder

Geo-IP location finder consists of three components, namely: IP finder, location finder and tracker auditor. The IP finder outputs IP address of the URL, by looking at its domain name [18–20]. The IP address is then passed to the location finder which detects location of the server of the given IP. Subsequently, an attachment containing the location is sent via email notification so that user can see the whereabouts of the server on map. Tracker auditor tracks the trackers which are collecting data for analytical purposes using network and code signature. This effectively gives users an all-round explanation of the URL and a sense of credence.

4.3. Model Design for Dynamic Phishing Safeguard System

For this model, the dataset came from the UCI machine learning repository and performed EDA (exploratory data analysis) on that particular dataset. The dataset consists of 30 different features of URLs expounded in paper [21] which can be used to detect whether a site is a phishing site or not. For training the Neural Boost Phishing Protection Algorithm, various supervised machine learning approaches were used. After some experimentation with different supervised machine learning algorithms APNA and APBA was considered for this kind of dataset.

Dynamic phishing safeguard system (DPSS) uses the neural boost phishing protection algorithm in the background [22–26]. This algorithm uses the APNA and APBA for the detection of phishing websites. These algorithms are best suited for DPSS in terms of accuracy, efficiency, performance, and predictive power. This algorithm uses two approaches, in the first approach uses a Geo-IP location tracker to track the location of the URL then through the location parameter user has the flexibility to check or block a particular site by analyzing the content of the site and then evaluating whether that information can originate from that particular location [27–30]. The second approach takes 30 different features of the URL, then actively analyses the websites and classifies whether it is a phishing website or not. Algorithm 1 is shown as following. To declare whether an URL is a phishing website or not it uses Algorithm 2 and Algorithm 3, it assigns -1 if the algorithm predicts the site is in the phishing category, otherwise it assigns 1 to the non-phishing category.

Algorithm 1: Neural Boost Phishing Protection Algorithm

Input:
(W)→any URL (In this paper Go-Phish URL)
(K)→ phishing dataset (from UCI machine learning repository)
(K_p)→phishing URL in UCI machine learning repository
(K_{np})→non-phishing URL in UCI machine learning repository
Output:
K_i = 1 if non-phishing site
$K_i = -1$ if phishing site
IP-address and location is shown on map
trackers and tracers present on that URL is shown
Generating algorithm begin:
Step 1 Initialization
Start Detection
Dynamic Phishing Safeguard System (DPSS) intercepts entered URL
Step 2 Location tracing and tracking
Geo-Ip location inspects location using API calls
Tracker auditor inspects trackers and tracers using web-scraping and by identifying code and network signature
Step 3 Training Machine learning model
URL (W) is intercepted by Algorithm 2 and Algorithm 3
Hyperparameter tuning conducted by Randomsearchcv
Train the model using K_p and K_{np}
Step 4 Detection of URL
Detecting whether K_i = 1 or -1 using Anti Phishing Boosting Algorithm (APBA) or Anti Phishing Neural Algorithm (APNA)
Step 5 Attack prevention
After this the URL gets directed to the attack prevention layer where DPSS gives users the authority to stop the website
Stop the program

4.4. Underscoring Anti Phishing Boosting Algorithm

The Anti-Phishing Boosting Algorithm (Algorithm 2) is used for training the model with ensemble learning techniques. This algorithm has the base architecture of extreme gradient boosting with some different parameters from traditional XG Boost. Tree boosting is a highly effective and widely used AI approach. Bagging trains multiple classifiers autonomously on bootstrap data, while boosting trains various powerless classifiers sequentially on contrastingly weighted variations of training samples. In other ensemble algorithms, trees are generated using Gini impurity or entropy. In terms of the node selection and splitting, XGBoost introduces a new metric known as the similarity score.

4.5. Underscoring Anti Phishing Neural Algorithm

The Anti-Phishing Neural Algorithm (Algorithm 3) can also be used for training the model. Algorithm 3 has the base architecture of an artificial neural network with some different parameters. It uses a rectifier linear unit in Equation (3) as the first hidden layer, swish activation function in Equation (4) as a second hidden layer and sigmoid in Equation (5) as an output layer. DPSS uses Adam as an optimizer.

$$\text{Relu}: \ f(x) = max(0, x) \tag{3}$$

$$\text{Sigmoid}: \ S(x) = \frac{1}{(\,1 + e^{-x})} \tag{4}$$

$$\text{Swish}: \ f(x) = x \cdot sigmoid(x) \tag{5}$$

Algorithm 2: Anti Phishing Boosting Algorithm (APBA)

Step 1 Initialization

X←Training the dataset of size s*z
Y←labels for feature in X(Result)
Let APBA loss function be $Lg(y, G(x))$
Number of estimators (iterations) = K
S = $\{(x_1, y_1, \ldots\ldots, x_s, y_s)\}$ where $x_i \epsilon X$ and $y_i \epsilon Y = \{-1, +1\}$
Model's initial value: $G_o(x) = argmin \sum_{i=1}^{s} L_g(y_i, \gamma)$.

Step 2 Computation

For k = 1, 2, ... K do
For i = 1,2, ... s do
Compute Pseudo-residuals: $p_{ik} = -\left[\frac{\delta Lg(y_i, G(x_i))}{\delta G(x_i)}\right]_{G(x)=G_{k-1}(x)}$
Fit a tree (base learner) $t_k(x)$ to pseudo-residuals (training it with by adding the training sets $\{(x_i, p_{ik})\}$ for every i)
$\gamma_k = argmin \sum_{i=1}^{s} L_g(y_i, G_{k-1}(x_i) + \gamma t_k(x_i))$

Update model
$G_k(x) = G_{k-1}(x_i) + y_k t_k(x_i)$
End for

Step 3 Generation

Output $G_k(x)$
End for

Algorithm 3: Anti Phishing Neural Algorithm (APNA)

Step 1 Initialization

X←Training the dataset of size s*z
y←labels for feature in X(Result)
w←weights of the respective hidden neural layers
ln← number of layers in neural network, 1 to Ln
$Er_{ij}^{(ln)} \leftarrow$ error for all ln,*i*,*j*
$Er_{ij}^{(ln)} \leftarrow 0$ for all ln,*i*,*j*

Step 2 Computation

For i = 1, 2, ... s do
$p^{ln} \leftarrow feedforward(x(i), w)$
$di^{ln} \leftarrow p(Ln) - y(i)$
$t_{ij}^{(ln)} \leftarrow t_{ij}^{(ln)} + p_j^{(ln)}.t_{ij}^{(ln+1)}$
If j $\neq 0$ then
$Er_{ij}^{(ln)} \leftarrow \frac{1}{s} t_{ij}^{(ln)} + \lambda w_{ij}^{(ln)}$
Else
$Er_{ij}^{(ln)} \leftarrow \frac{1}{s} t_{ij}^{(ln)}$

Step 3 Generation

Where $\frac{\delta}{\delta w_{ij}^{(ln)}} J(w) = Er_{ij}^{(ln)}$
End for

4.6. Email-Notification and Website Blocker

The Email-Notification and Website Blocker acts as an attack prevention layer. If the supervised machine learning model predicts a positive output then an email notification is sent to the user and the URL is blocked. If it is not a phishing URL then nothing will happen and the user will be able to continue browsing with full trust in the URL.

5. Experimental Setup and Results

This section will expound on the setup and processes that are used in DPSS with in-depth details and will establish how these processes have been used to obtain the results

and reduce phishing and data breaching attacks. DPSS provides the following solution to help people secure their data and feel more secure in this technical world.

The operating system used for this research is Microsoft Windows 10 which provides the platform to perform all the necessary tasks required. It has a user-friendly GUI or graphical user interface and provides security such as firewalls to help in identifying harmful or malicious attacks that may be required. VS code is used as a code editor due to its advantages, e.g., auto-indentation, syntax highlighting, box-selection, etc. Python and Javascript are used as programming languages to build DPSS. Python is an open-source language that means many people contribute to the packages that are available in python and that is one of the profitable parts of using python. JavaScript is used to integrate all the requirements by getting the IP address using the Application Programming Interface (API) in python which gives the IP address and finds the location of that IP address using IP stack and showing that location on a map using leaflet. Ipstack is an API that finds the geolocation of the given IP and gives latitude, longitude, city, state, and country. The leaflet is a JavaScript library that visually presents the location given to it on the map. The location is given geographically in the form of latitude and longitude. DPSS can also decide the marker user interface and write information about that location. It uses this for detection and getting all the geographical information of that website and showing that to the user visually and also providing more information such as the number and name of trackers present on that website. DPSS uses flask to make an API. Flask is a framework of python which is used to make a web API. Furthermore, Google Collaboratory is used for training the dataset for APNA and ABPA. It is provided with a Jupyter notebook that is used to design and implement models for machine learning in python.

Attack generation is conducted using Go-Phish and Gmail. It is an open toolkit that is used to generate a phishing email and then generate a website to link it to the phishing email. The email is then sent to respective places, these emails when opened or the link in the given mail is used to track if the user entered the details, it generates the credentials they added regarding data. Gmail is a platform where people can get electronic mail and one of the most famous platforms for a phishing attack. It gives us many advantages such as sending emails faster, checking spam emails which can be fraudulent or "scam" emails.

5.1. Dataset

The dataset for this research was attained from the UCI Machine Learning Repository [31–34], which is open to the public. The Phishtank archive, MillerSmiles archive, and Google's searching operators were used to compile the phishing websites dataset. This Dataset consists of 11,055 websites. A value of -1 represents the website of the phishing category and 1 represents the non-phishing category. Various website features were extracted from this dataset in which 6157 entries were non-phishing websites and 4898 were phishing websites.

5.2. Attack Detection

The process of IP finder in the proposed architecture is divided into two parts: the first part focuses on finding the IP address of the specific URLs and the second part focuses on finding the IP address through an email that has been sent to the user on Gmail. These URLs are known as web addresses which refer to the location of the web resources that are stored at a particular network in the computer and similarly, a method to retrieve those resources. A URL is comprised of three parts; namely protocol, hostname, and file name and it is displayed as "https://www.domainname.com//fp.html (accessed on 4 March 2021)".

It uses this domain name to get the IP address of the particular URL. The code uses a socket feature in Python that is used to set up a connection between two nodes and using the particular function in that component. DPSS uses it to get the desired output which is the IP address of the domain. As the domain name is entered in the input command it is sent to the next line where the "socket.gethostbyname (URL)" function reads the input

and resolves the URL. Then, the resolved URL gives back the IP address that the socket is connected to. The second part is finding an IP address through Gmail. DPSS opens the given email and goes to the more option and selects the option of more where it sees the original form of the email. Where it sees the received section and that section contains an IP address in the square bracket and that is the IP address through which it has been sent to us and DPSS can use this IP address to locate the sender.

Tracker auditor is an API made from flask it runs on http://127.0.0.1:3001/api/res (accessed on 4 March 2021) and DPSS has to provide a URL in id for getting trackers on that particular page example (http://127.0.0.1:3001/api/res?id=https://in.pinterest.com (accessed on 4 March 2021)). Furthermore, in the tracker auditor, DPSS has scraped the src link of all script tags present on that web page, and then it uses regex to check the particular tracker present by matching with the code and network signature of the particular tracker. DPSS is counting the number of trackers and displays it to the user with the name and link of that tracker for more information. If the signature found on that page gets validated by the signature present in the dataset by regex, then the user gets to know the complete information of all the trackers present on that page. So, they will know who is tracking them by having their complete interaction data with that site. If the user wants to know what type of data the site is interested in then they might click the link which redirects them to that site and gives them the complete information about what type of personal data might be stored by that site. A URL is inputted then from the IP address, and from geo-location, latitude and longitude is found; then, using that information, it is visually shown on the map with the marked city as shown in Table 1.

Table 1. IP address, latitude, longitude, location of the main server of provided URL.

URL	IP Address	Latitude/Longitude	Location
dvwa.co.uk	185.199.110.153	37.76784896850586/-122.39286041259766	San Francisco
www.itsecgames.com	184.168.131.241	33.50938034057617/-112.08255004882812	Alhambra
testphp.vulnweb.com	18.192.172.30	50.11090087890625/-8.682100296020508	Frankfurt am Main
www.searchnu.com	82.160.48.60	40.7589111328125/-7397901916503906	Manhattan

The first model that gave better accuracy was APBA. APBA is superior model of Extreme Gradient Boosting (XG-Boost). XG-Boost is a gradient boosting technique based on ensemble Machine Learning. In the case of forecasting problems concerning unstructured information such as text, photos, and so on, neural networks incline to outclass any algorithm. Nevertheless, models based on decision tree are seen as the superlative model in the case of lesser to medium structure/tabular data. Since a medium-scale, the tabular dataset is used for classification, few options based on ensemble learning, e.g., bagging, boosting, etc., can only be opted. The evaluation in this method is based on feedback from prior predictions. As a result, XG-Boost was chosen for as a base model for APBA because it is appropriate for the use case.

In Figure 2 the variables are named e.g., f0 and f29. It corresponds to the 30 features that a URL can have in this figure there is a split decision within each node which corresponds to yes or no (yes is denoted by blue color and no is denoted by red color. The value of the leaf node indicated the raw score of class 1 which is a phishing site. The raw score has been converted to a probability score using logistic functions. It does not explore all possible tree structures but builds a tree greedily. An alternative model that was used by DPSS to detect phishing sites was APNA. It is based on a collection of various units or nodes which are called artificial neurons that vaguely resemble the neurons in the biological brain which are interconnected. Neurons have dendrites and axons. In dendrites, they receive stimulation

from another axon of another neuron and thus makes a connection. In the same way, ANN has an input layer as dendrites and hidden layer as artificial neurons and an output layer as axons training and validation accuracy has been visualized in Figure 3.

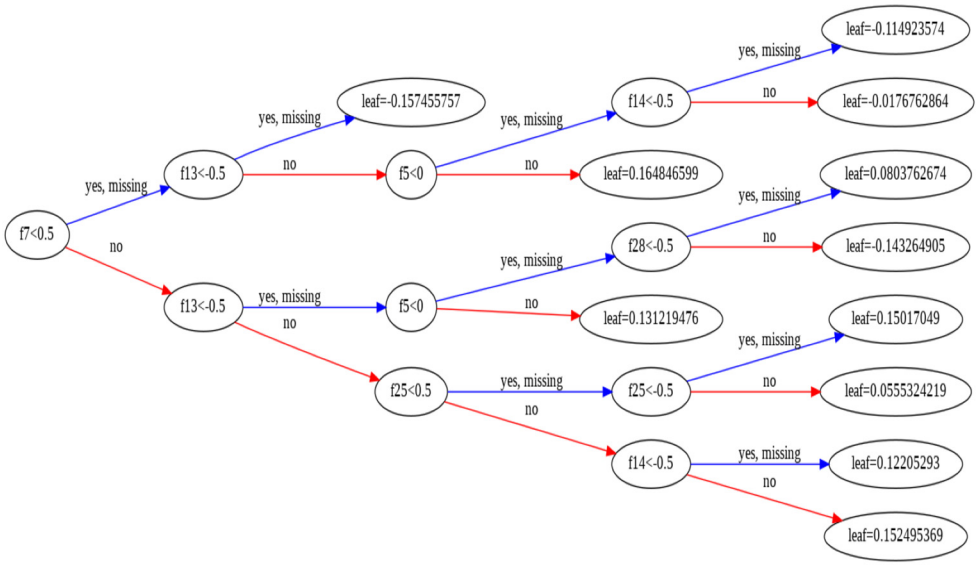

Figure 2. Visualizing gradient boosting decision tree with APBA.

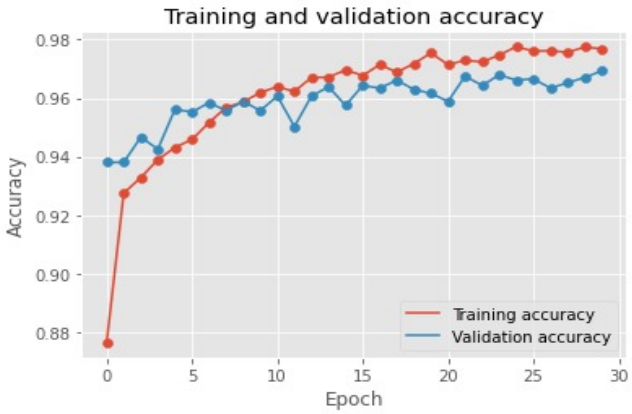

Figure 3. Visualization training and validation accuracy of APNA.

5.3. Measurement Metrics of Machine Learning Models

As part of testing, the erudite classifier is assessed against the testing dataset to determine its classification accuracy. If the training dataset can be classified accurately, then the classifier which is trained can be utilized in the applications of the real world. Otherwise, various additional operations, such as data processing or hyperparameter tuning, might be applied to enhance the classification accuracy. In case the accuracy cannot be improved, a different algorithm of machine learning can be used to determine which machine learning method is the most efficient. DPSS used these formulas as evaluation metrics:

$$\text{Precision} = \frac{TNPS}{(TNPS + FNPS)} \tag{6}$$

$$\text{Recall} = \frac{TNPS}{(TNPS + FPS)} \tag{7}$$

$$\text{CCR} = \frac{(TNPS + TPS)}{(TNPS + TPS + FNPS + FPS)}\ (\%) \tag{8}$$

$$\text{F1} - \text{Score} = \frac{TNPS}{\left(TNPS + \frac{1}{2}(FNPS + FPS)\right)} \tag{9}$$

Our model capabilities were assessed by using different evaluation metrics shown in Equations (6)–(9). TNPS is the true non-phishing site, FNPS is the false non-phishing site, TPS is a true phishing site, FPS is a false phishing site and correct classification rate is used to determine the accuracy of the models. In addition to CCR (Correct Classification Rate), F1-score is also used for evaluation. APNA, APBA and Logistic Regression (LR) are trained with hyperparameter tuning such as Maxdepth, n_estimator, learning rate, number of epochs and batch size to predict the output and the best performance has been given by APBA as shown in Table 2 and the graph is expounded in Figure 4. Given the intricacy, and bulkiness of the data with 30 distinct features, logistic regression was applied. It allows one to categorize data into distinct classes by analyzing the relationship between a collection of labeled data. It is the most basic binary classification model. Moreover, it is simpler to implement, interpret, and train in terms of CPU consumption. However, when compared to the APNA and APBA models, accuracy was lower.

Table 2. Performance evaluation of machine learning models.

Model	Recall	Precision	F1-Score	Accuracy
APNA	98.78	96.75	97.69	97.11
APBA	98.62	96.81	97.70	97.82
LR	92.00	94.00	93.00	93.71

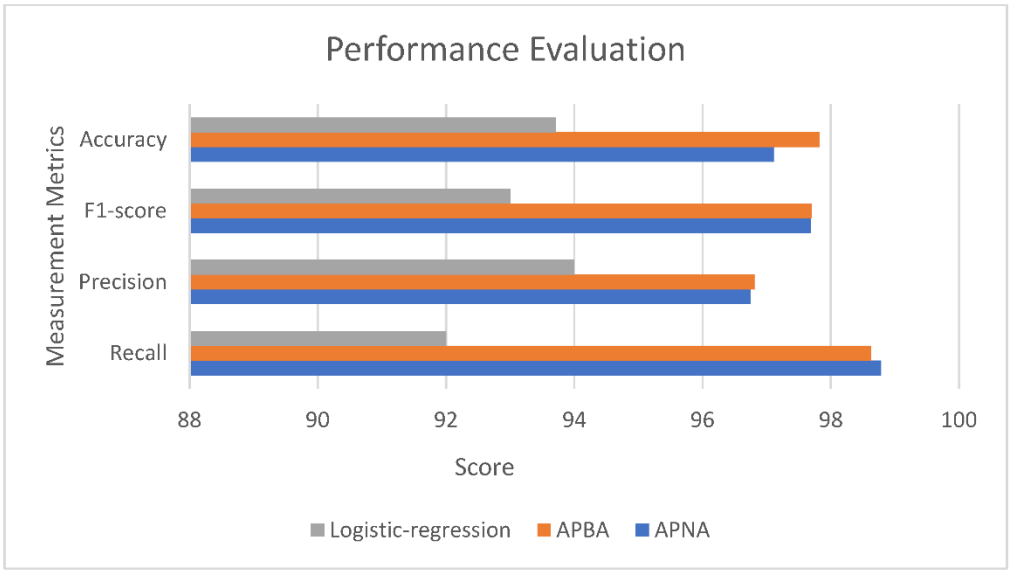

Figure 4. Performance evaluation of DPSS.

UCI machine learning repository is used for training APNA. This URL is divided into 30 different features based on various parameters such as having an IP address, having a symbol, URL length, having subdomain, favicon, etc.; based on these features, Supervised-Machine learning-based forecasting algorithm assigns -1 to the column if it does not have the permissible value and 1 if it is in the permissible range and 0 if it is a bit suspicious, lying on the borderline cases. In the end, the model predicts whether a site is phishing or not by assigning -1 to phishing sites and 1 to non-phishing sites.

Figure 5 demonstrates different features of the URL with their importance in regards to which feature constitutes more for class 1 (which is a phishing site). f0–f9 are address bar-based features, f10–f16 are abnormal based features, f17–f21 are HTML and Javascript-based features, f22–f27 are domain-based features, f28 and f29 are output. The feature

importance was extracted by getting explain ability of why our algorithm (APBA) was giving us which sites are phishing websites or not. Each attribute split point's contribution to the performance measure is weighted by the number of observations it is accountable for when determining the importance of a single decision tree. The Gini index used for choosing the split points may be a performance metric or another error function. The features importance levels are then calculated on an average across all of the model's decision trees. This figure explains that feature 14 which is Server Form Handler (SFH) that forms abnormal based features of an URL [35–37] constitutes most for class 1. One of the possible explanations could be that when a user submits information on a web page, it is sent to the authentication server (SFH)from which the web page is loaded. However, phishers redirect the URL to a different one, which is not the right one. If a site asks users to submit personal information via a popup window, the site may be a phishing site. Phishers deceive users by using bogus HTTP protocols. The site is likely phishing if objects are loading from a different URL than the one requested. A phishing web page is one with links that take you to a domain other than the one you typed.

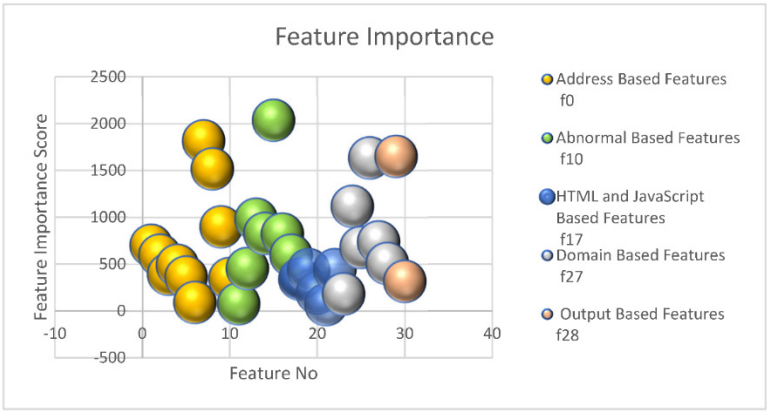

Figure 5. Feature importance distribution of URL extracted by APBA.

5.4. Attack Prevention

This layer of dynamic phishing safeguard system prevents any phishing attack by blocking the URL as shown in the Table 3. Threatening websites that can jeopardize user privacy can be blocked and save the user from accidentally accessing those websites.

Table 3. URL blocked or not on private browser.

URL	Website-Blocker
www.testphp.vulnweb.com	Blocked = Yes
www.dvwa.co.uk	Blocked = Yes
www.itsecgames.com	Blocked = No

All computers have a host document which keeps the basic details of the computer e.g., localhost address; however, if the address is added with a domain name in that host file, that URL can be blocked. The code is entered with the host file path address in the system then the redirected IP address is stored in the code. After this, it enters the list of websites that are perilous and their domain name is stored in that list. Then a loop is started to read the host file in the system and check if a website is already added to it and if not, the website is added to it by concatenating the redirected IP address and domain name. DPSS can also schedule it to the operating system by adding the python code to the task manager.

5.5. Performance Comparisons

In Figure 6, a comparison of models used in dynamic phishing safeguard systems is compared with 7 models previously used in paper [38]. The comparison is conducted based on CCR using Equation (2). Actual values can be seen in the Table 4 that APBA outperforms all the models used in previous papers. APNA gives a tough competition to APBA in terms of F1-score and accuracy as shown in Table 2 however APBA surpasses it too.

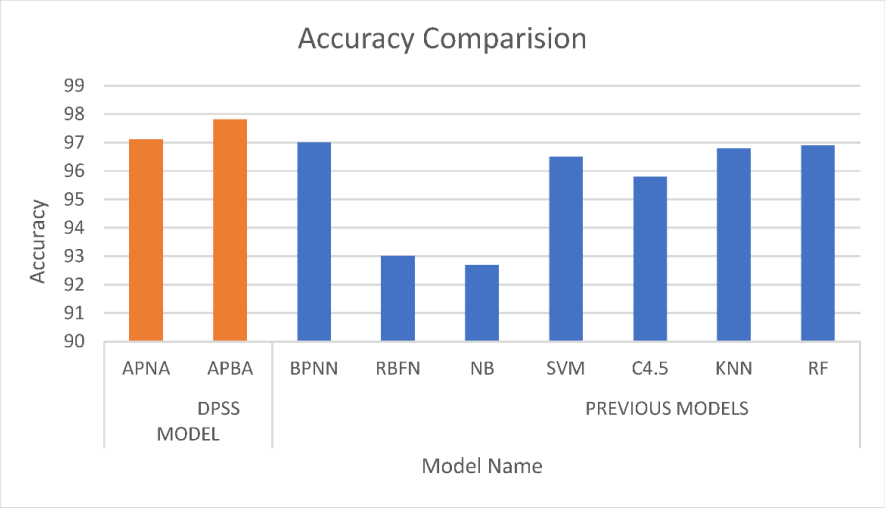

Figure 6. Comparison of our model with other models based on accuracy (CCR).

Table 4. Table for comparison of DPSS model with other models based on accuracy (CCR).

DPSS Model		Previous Models						
APNA	**APBA**	**BPNN**	**RBFN**	**NB**	**SVM**	**C4.5**	**KNN**	**RF**
97.11	97.82	97	93	92.7	96.5	95.8	96.8	96.9

6. Conclusions and Future Work

In this paper we present a state-of-the-art approach that eliminates online subterfuge through the implementation of a dynamic phishing safeguard system that utilizes the neural boost phishing protection algorithm targeting phishing, fraud, and optimizing the problem of data breaches. The best accuracy was achieved by APBA among three different machine learning approaches. APBA outperforms the previous models used on this dataset with a CCR of 97.82%. Furthermore, this paper explains the features that play the most significant role in forecasting by comparing 30 distinct features. At this point, DPSS knows that the Google analytics or Google tag manager link is present in a script tag, since DPSS has the code and network signature for these services. In the near future, the power of machine learning and Natural Language Processing (NLP) can be exploited to identify all the trackers and tracers present on the URL. Instead of leaving the work in the hand of the user, NLP can be exploited for classifying the information that came from the email, SMS, etc., and crosscheck it with location produced from the Geo-IP location approach of supervised machine learning-based forecasting. Integration of all the services in one extension can be conducted to prevent and notify the user of any potential attack using the rules generated by multiple algorithms.

Author Contributions: A.Q.M., D.J., J.D., S.H., C.I. and S.K.J. have equally contributed to the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Vratonjic, N.; Huguenin, K.; Bindschaedler, V.; Hubaux, J.-P. A Location-Privacy Threat Stemming from the Use of Shared Public IP Addresses. *IEEE Trans. Mob. Comput.* **2014**, *13*, 2445–2457. [CrossRef]
2. Rajashree, S.; Soman, K.S.; Shah, P.G. Security with IP Address Assignment and Spoofing for Smart IOT Devices. In Proceedings of the 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Bangalore, India, 19–22 September 2018.
3. Gruschka, N.; Mavroeidis, V.; Vishi, K.; Jensen, M. Privacy Issues and Data Protection in Big Data: A Case Study Analysis under GDPR. In Proceedings of the 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 10–13 December 2018.
4. Gupta, B.; Madan, G.; Md, A.Q. A Smart Agriculture Framework for IoT Based Plant Decay Detection Using Smart Croft Algorithm. *Mater. Today Proc.* **2022**, *62*, 4758–4763. [CrossRef]
5. Bernhard, R. Breaching System Security. *IEEE Spectr.* **1982**, *19*, 24–31. [CrossRef]
6. Hammouchi, H.; Cherqi, O.; Mezzour, G.; Ghogho, M.; Koutbi, M.E. Digging Deeper into Data Breaches: An Exploratory Data Analysis of Hacking Breaches Over Time. *Procedia Comput. Sci.* **2019**, *151*, 1004–1009. [CrossRef]
7. Sabireen, H.; Neelanarayanan, V. A Review on Fog Computing: Architecture, Fog with IoT, Algorithms and Research Challenges. *ICT Express* **2021**, *7*, 162–176.
8. Abdul Quadir, M.; Prassanna, J.; Christy Jackson, J.; Sabireen, H.; Gupta, G. Efficient Algorithm for CSP Selection Based on Three-Level Architecture. In Proceedings of the Artificial Intelligence and Technologies, Chennai, India, 6–7 July 2020; pp. 515–531.
9. Mathew, S.A.; Md, A.Q. Evaluation of Blockchain in Capital Market Use-Cases. *IJWP* **2018**, *10*, 54–76. [CrossRef]
10. Quadir, M.A.; Christy Jackson, J.; Prassanna, J.; Sathyarajasekaran, K.; Kumar, K.; Sabireen, H.; Vijaya Kumar, V. An Efficient Algorithm to Detect DDoS Amplification Attacks. *J. Intell. Fuzzy Syst.* **2020**, *39*, 8565–8572. [CrossRef]
11. Floyd, T.; Grieco, M.; Reid, E.F. Mining Hospital Data Breach Records: Cyber Threats to U.S. Hospitals. In Proceedings of the 2016 IEEE Conference on Intelligence and Security Informatics (ISI), Tucson, AZ, USA, 26–30 September 2016; pp. 43–48.
12. Rajakumaran, G.; Venkataraman, N.; Quadir, A. Early Detection of LDoS Attack Using SNMP MIBs. *ITM Web Conf.* **2021**, *37*, 01025. [CrossRef]
13. Dhandapani, K.; Balasundaram, A.; Dhanalakshmi, R.; Sivaraman, A.K.; Ashokkumar, S.; Vincent, R.; Rajesh, M. Energy and Bandwidth Based Link Stability Routing Algorithm for IoT. *Comput. Mater. Contin.* **2021**, *70*, 3875–3890.
14. Joseph, R.C. Data Breaches: Public Sector Perspectives. *IT Prof.* **2018**, *20*, 57–64. [CrossRef]
15. Md, A.Q.; Varadarajan, V.; Mandal, K. Correction to: Efficient Algorithm for Identification and Cache Based Discovery of Cloud Services. *Mob. Networks Appl.* **2019**, *24*, 1198. [CrossRef]
16. Balasundaram, A.; Dilip, G.; Manickam, M.; Sivaraman, A.K.; Gurunathan, K.; Dhanalakshmi, R.; Ashokkumar, S. Abnormality Identification in Video Surveillance System Using DCT. *Intell. Autom. Soft Comput.* **2021**, *32*, 693–704. [CrossRef]
17. Prassanna, J.; Quadir, A. Secrecy protector: A novel data analytics based credit score management system. *Int. J. Sci. Technol. Res.* **2020**, *9*, 29–38.
18. Md, A.Q.; Agrawal, D.; Mehta, M.; Sivaraman, A.K.; Tee, K.F. Time Optimization of Unmanned Aerial Vehicles Using an Augmented Path. *Future Internet* **2021**, *13*, 308. [CrossRef]
19. Rani, S.; Kataria, A.; Chauhan, M.; Rattan, P.; Kumar, R.; Kumar Sivaraman, A. Security and Privacy Challenges in the Deployment of Cyber-Physical Systems in Smart City Applications: State-of-Art Work. *Mater. Today: Proc.* **2022**, *62*, 4671–4676. [CrossRef]
20. Srinivasan, A.; Md, A.; Varadarajan, V. Hybrid Cloud for Educational Sector. *Procedia Comput. Sci.* **2015**, *50*, 37–41. [CrossRef]
21. UCI Machine Learning Repository, California, USA. Available online: http://archive.ics.uci.edu/ml (accessed on 15 April 2022).
22. Decanio, S.; Soltys, M.; Hildreth, K. Voyager: Tracking with a Click. *Procedia Comput. Sci.* **2020**, *176*, 98–107. [CrossRef]
23. Christy Jackson, J.; Prassanna, J.; Abdul Quadir, M.; Sivakumar, V. Stock Market Analysis and Prediction Using Time Series Analysis. *Mater. Today Proc.* **2021**. [CrossRef]
24. Yogesh, P.R.; Satish R, D. Backtracking Tool Root-Tracker to Identify True Source of Cyber Crime. *Procedia Comput. Sci.* **2020**, *171*, 1120–1128. [CrossRef]
25. Wang, C.; Li, W.; Liu, F.; Lei, Z. Research of Domain Name Mapped IP-Address Distribution on the Internet. In Proceedings of the 2012 3rd IEEE International Conference on Network Infrastructure and Digital Content, Beijing, China, 21–23 September 2012; pp. 284–288.
26. Matthew, K.M.; Quadir Md, A. An Effective Way of Evaluating Trust in Inter-Cloud Computing. *IJCNIS* **2017**, *9*, 36–42. [CrossRef]
27. Qadir Md, A.; Vijayakumar, V. Combined Preference Ranking Algorithm for Comparing and Initial Ranking of Cloud Services. *Recent Adv. Electr. Electron. Eng. (Former. Recent Pat. Electr. Electron. Eng.)* **2020**, *13*, 260–275. [CrossRef]
28. Kirthica, S.; Sabireen, H.; Sridhar, R. Unified Framework for Data Management in Multi-Cloud Environment. *Int. J. Big Data Intell.* **2019**, *6*, 129–139. [CrossRef]
29. Ali, W. Phishing Website Detection Based on Supervised Machine Learning with Wrapper Features Selection. *Int. J. Adv. Comput. Sci. Appl.* **2017**, *8*, 9–15. [CrossRef]
30. McCluskey, L.; Thabtah, F.; Mohammad, R.M. Intelligent Rule-based Phishing Websites Classification. *IET Inf. Secur.* **2014**, *8*, 153–160.
31. Wang, W.; Huang, X.; Li, J.; Zhang, P.; Wang, X. Detecting COVID-19 Patients in X-Ray Images Based on MAI-Nets. *Int. J. Comput. Intell. Syst.* **2021**, *14*, 1607–1616. [CrossRef]

32. Gui, Y.; Zeng, G. Joint Learning of Visual and Spatial Features for Edit Propagation from a Single Image. *Vis. Comput.* **2022**, *36*, 469–482. [CrossRef]
33. Wang, W.; Li, Y.; Zou, T.; Wang, X.; You, J.; Luo, Y. A Novel Image Classification Approach via Dense-MobileNet Models. *Mob. Inf. Syst.* **2020**, *2020*, 7602384. [CrossRef]
34. Zhou, S.-R.; Yin, J.-P.; Zhang, J.-M. Local Binary Pattern (LBP) and Local Phase Quantization (LBQ) Based on Gabor Filter for Face Representation. *Neurocomputing* **2013**, *116*, 260–264. [CrossRef]
35. Song, Y.; Zhang, D.; Tang, Q.; Tang, S.; Yang, K. Local and Nonlocal Constraints for Compressed Sensing Video and Multi-View Image Recovery. *Neurocomputing* **2020**, *406*, 34–48. [CrossRef]
36. Zhang, D.; Wang, S.; Li, F.; Tian, S.; Wang, J.; Ding, X.; Gong, R. An Efficient ECG Denoising Method Based on Empirical Mode Decomposition, Sample Entropy, and Improved Threshold Function. *Wirel. Commun. Mob. Comput.* **2020**, *2020*, 8811962. [CrossRef]
37. Li, F.; Ou, C.; Gui, Y.; Xiang, L. Instant Edit Propagation on Images Based on Bilateral Grid. *CMC-Comput. Mater. Contin.* **2019**, *61*, 643–656. [CrossRef]
38. Song, Y.; Zeng, Y.; Li, X.; Cai, B.; Yang, G. Fast CU Size Decision and Mode Decision Algorithm for Intra Prediction in HEVC. *Multimed Tools Appl.* **2017**, *76*, 2001–2017. [CrossRef]