REPUBLIC OF TURKEY
ADANA ALPARSLAN TÜRKEŞ SCIENCE AND TECHNOLOGY UNIVERSITY

GRADUATE SCHOOL
CYBER SECURITY DEPARTMENT

A NOVEL TWO PHASED APPROACH COMBINING DEEP LEARNING AND MACHINE LEARNING CLASSIFIERS FOR EFFECTIVE DETECTION OF TURKISH PHISHING WEB SITES

İhsan DENİZ

M.Sc.

ADANA 2024

REPUBLIC OF TURKEY
ADANA ALPARSLAN TÜRKEŞ SCIENCE AND TECHNOLOGY UNIVERSITY

GRADUATE SCHOOL
CYBER SECURITY DEPARTMENT

A NOVEL TWO PHASED APPROACH COMBINING DEEP LEARNING AND MACHINE LEARNING CLASSIFIERS FOR EFFECTIVE DETECTION OF TURKISH PHISHING WEB SITES

İhsan DENİZ

M.Sc.

THESIS ADVISOR
Asst. Prof. Dr.
Çağatay Neftali TÜLÜ

ADANA, 2024

ABSTRACT

A NOVEL TWO PHASED APPROACH COMBINING DEEP LEARNING AND MACHINE LEARNING CLASSIFIERS FOR EFFECTIVE DETECTION OF TURKISH PHISHING WEB SITES

Ihsan DENİZ

M.Sc., Department of Cyber Security

Supervisor: Asst. Prof. Dr.

Çağatay Neftali TÜLÜ

January 2024, 49 pages

With the increase in internet speed and the parallel rise in the number of internet-connected devices, online fraud has exhibited a significant surge in recent years. Attackers exploit platforms such as WhatsApp, email, SMS, mobile notifications, and social media messages to disseminate content that is attention-grabbing, intriguing, or fear-inducing. By inducing users to interact with these contents and click on embedded links, these malevolent actors redirect users to counterfeit websites that closely mimic authentic ones, thereby obtaining users' confidential information or engaging in various forms of deception. Commonly referred to as "phishing" sites, these malicious web pages are often used for such deceptive operations. Consequently, it is of paramount importance that mobile applications or browsers possess the capability to identify such harmful websites even before users access them. This study employs a two-stage approach to achieve a 98.4% success rate in identifying malicious sites. The dataset used consists of a list of malicious sites from the National Cyber Incident Response Center (USOM) alongside legitimate domain names. The dataset is divided into two subsets, namely Dataset1 and Dataset2. Dataset1 is employed to train a deep learning-based artificial intelligence model, which yields an accuracy rate of 92% upon completion of training. The websites within Dataset2 are subjected to the deep learning model in the initial stage to acquire phishing scores. Subsequently, by incorporating additional features pertaining to each website and employing a machine learning model for binary classification, the second stage of training facilitates the culmination of the ultimate outcome. Test results demonstrate the capacity to predict phishing incidents with a 98.4% accuracy score for a given website.

Keywords: Online Fraud, Cyber Attack, Machine learning, Deep learning, Malicious URL

ÖZET

TÜRKÇE KIMLIK AVI WEB SITELERININ ETKIN TESPITI IÇIN DERIN ÖĞRENME VE MAKINE ÖĞRENIMISI SINIFLANDIRICILARINI BIRLEŞTIREN YENI, İKI AŞAMALI BIR YAKLAŞIM

İhsan DENİZ
Yüksek Lisans, Siber GüvenlikAnabilim Dalı
Danışman: Dr. Öğr. Üyesi
Çağatay Neftali TÜLÜ
Ocak 2024, 49 sayfa

Son yıllarda internet hızının artması ve internete bağlı cihaz sayısının da paralel olarak artması ile birlikte online dolandırıcılık da önemli ölçüde artış göstermiştir. Saldırganlar, WhatsApp, e-posta, SMS, mobil bildirimler ve sosyal medya mesajları gibi platformları kullanarak dikkat çekici, ilgi çekici veya korkutucu içerikler yaymaktadırlar. Kullanıcıları bu içeriklere etkileşimde bulunmaya ve gömülü bağlantılara tıklamaya teşvik ederek, kötü niyetli aktörler kullanıcıları otantik sitelere çok benzeyen sahte web sitelerine yönlendirir ve bu şekilde kullanıcıların güvenli bilgilerini ele geçirir veya farklı yollarla menfaat temin eder.

Bu tür aldatmaca işlemleri için hazırlanan ve "phishing" siteleri olarak adlandırılan bu kötü niyetli web sayfalarının, kullanıcıların erişiminden önce mobil uygulamalar veya tarayıcılar tarafından tespit edilmesi son derece önemlidir. Bu çalışma, oltalama sitelerini tanıma konusunda %98,4'lük bir başarı oranına ulaşmak için iki aşamalı bir yaklaşım önermektedir. Kullanılan veri seti, Ulusal Siber Olaylara Müdahale Merkezi'nin (USOM) oltalama siteleri listesi ve meşru alan adlarından oluşmaktadır. Veri seti, Dataset1 ve Dataset2 olmak üzere iki alt küme halinde ayrılmıştır. Dataset1, derin öğrenme tabanlı bir yapay zeka modelini eğitmek için kullanılarak, eğitim sonucunda %92'lik bir doğruluk değeri elde etmiştir.

Dataset2 ise derin öğrenme tabanlı modelin bir site için verdiği oltalama puanının yanında yine o web sitesine ilişkin ek özellikleri içeren ve ikili sınıflandırma için bir makine öğrenimi modelini kullanan modelin hazırlanmasında kullanılmıştır. Yapılan testler, önerilen bu yaklaşımın bir web sitesi için %98,4'lük bir doğruluk puanı ile oltalama sitesi olup olmadığına dair tahmin yapabildiğini göstermektedir.

Anahtar Kelimeler: Phishing, Online Dolandırıcılık, Siber Saldırı, Makine öğrenmesi, Derin öğrneme, Zararlı URL

ACKNOWLEDGEMENTS

I would like to express my gratitude to my supervisor Asst. Prof. Dr. Çağatay Neftali TÜLÜ for the valuable help and guidance he provided and his patience.

I would like to express my gratitude to my precious family and my precious wife, who supported me financially and morally throughout my education journey and brought me to this day.

I would like to express my gratitude to Dear Sir Mehmet Fatih KAYA and his family, who have supported me in every way since the day I met him.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

USOM	: Turkish National Cyber Incident Response Center
SVM	: Support Vector Machine
URL	: Uniform Resource Locator
HTML	: HyperText Markup Language
DOM	: Document Object Model
CNN	: Convolutional neural network
GSB	: Google Safe Browsing
LSTM	: Long Short-Term Memory
BiLSTM	: Bidirectional Long Short-Term Memory
NLP	: Neuro Linguistic Programming
BTK	: Bilgi Teknolojileri ve İletişim Kurumu
HTTPS	: Secure Hypertext Transfer Protocol
SSL	: Secure Sockets Layer
CVS	: Comma Separated Values
RNN	: Recurrent Neural Network
GRU	: Gated Recurrent Unit
TANH	: Hyperbolic Tangent
API	: Application Programming Interface
XML	: Extensible Markup Language
TXT	: TEXT
TP	: True Positive
TN	: True Negative
FP	: False positive
FN	: False Negative

1. INTRODUCTION

Phishing-based cybercrimes have become an increasingly formidable threat with the widespread adoption of the internet and the growing number of devices connected to it. In the past, attackers would entice users by sending emails persuading them to open an attached file. Upon the user downloading and opening this malicious file, the attacker would seize control of the user's computer, utilizing it for cybercrimes. Additionally, attackers would exploit the situation by obtaining the user's confidential information and files, either demanding ransom or gaining benefits through various methods and threats[1]–[3].

With improved digital literacy and increased awareness about cybercrimes, along with enhanced security measures implemented by email servers and operating systems, attackers have been compelled to refine their methods further. In the next stage, attackers send warning emails such as "your mailbox is full, log in to increase storage," inducing panic in individuals and leading to the compromise of email access credentials. Subsequently, attackers gain control over email accounts, thereby gaining control over various internet services linked to the compromised email[2]–[4].

The rise in the use of social media, where many services are accessed through email credentials, has exacerbated the impact of phishing threats. Despite continuous efforts to prevent evolving and changing email-based phishing attacks, complete prevention remains elusive. While measures like marking emails as spam or phishing have helped mitigate the risks to some extent, attackers persistently experiment with different methods, sending enticing content to users to increase the effectiveness of phishing attacks.

Examining phishing attacks specific to Turkey, it can be observed that the number of attacks increases by an average of 50% each year.[1]

Currently, many phishing attacks initiate through short messages, electronic communications, or social media, where content that captures the user's attention and instills concern is sent. The

[1] www.usom.gov.tr

success of these attacks relies on the victim clicking on a link in the message, providing requested confidential information, or opening an attached file. While attacks were initially more prevalent through email and obtaining access credentials, advancements in preventive measures by email providers have led to a decline in such attacks.

The stringent measures taken by massive online social media and email platforms based on user features and behaviors have made it challenging for attackers. However, this has also pushed attackers to employ more sophisticated methods in their phishing endeavors.

Irrespective of the differentiation in communication channels, the goal of attackers remains consistent: to obtain the user's confidential information and illicitly gain benefits. Attackers often acquire a substantial amount of user-specific confidential information by enticing users to click on links embedded in the messages they convey. At this juncture, it is crucial to determine whether the shared link is a phishing link. Phishing links often share common characteristics, typically utilizing the name of a popular shopping site, bank, or social media platform. However, the domain name of the site is imitated, requiring a user with a high level of awareness to discern this. Users with a lower level of awareness may, out of curiosity, click the link and provide the requested information, allowing the attacker to achieve their goal. If the user is on a public network without additional security measures against potential external or internal attacks, the link is often opened[5]–[8].

According to the Global Digital Headlines January 2023 DataReportal[2], as depicted in Figure 1.1, as of July 2023, the global population has reached 8.05 billion, with an increase of 70 million people in the past year. GSMA Intelligence's analysis reveals 5.56 billion unique mobile subscribers worldwide, equivalent to 69.1% of the global population. Mobile phone usage increased by 2.7% last year, gaining nearly 150 million new users. The number of internet users has grown by 2.1% in the past year, reaching 5.19 billion as of July 2023. However, due to reporting delays, actual internet penetration is likely higher than these figures suggest. Furthermore, the number of active social media users worldwide reached 4.88 billion,

[2] https://datareportal.com

representing 60.6% of the global population, as of July 2023. Social media usage increased by 3.7% compared to the previous year, adding 173 million new active users.

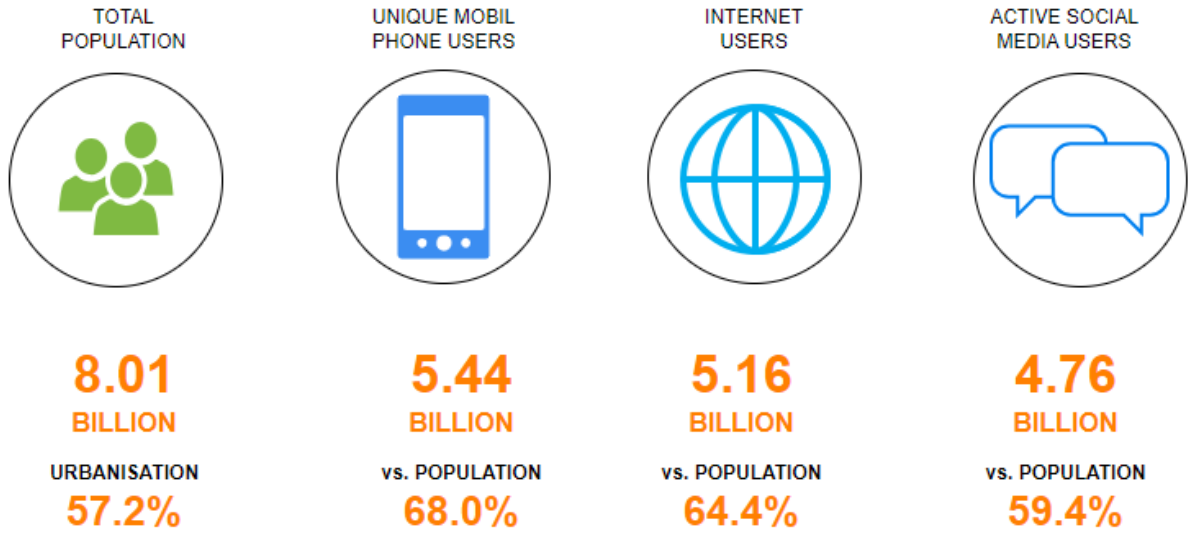


Figure 1. 1 Global Digital Headlines January 2023 DataReportal

As per the DataReportal[3], the internet penetration rate in Turkey was at 83.4%, with a total of 71.38 million internet users. Additionally, there were 62.55 million social media users in Turkey, constituting 73.1% of the total population. At the beginning of 2023, there were 81.68 million active cellular mobile connections in Turkey, accounting for 95.4% of the total population. The number of internet users in Turkey increased by 416 thousand people, registering a growth of 0.6% between 2022 and 2023. Globally, there are 5.16 billion internet users, indicating that 64.4% of the world's population is online. However, delays in data reporting suggest that actual growth may exceed this figure.

Identity theft attacks have exhibited a significant increase in recent years. According to Slashnext's "The State of Phishing 2022" report[4], as depicted in Figure-1.1, phishing attacks on malicious URLs have consistently increased by 61% since 2021. The detection of 255 million phishing attacks in 2022 further supports this alarming trend.

[3] https://datareportal.com/reports/digital-2023-turkey
[4] https://www.slashnext.com/wp-content/uploads/2022/10/SlashNext-The-State-of-Phishing-2022.pdf

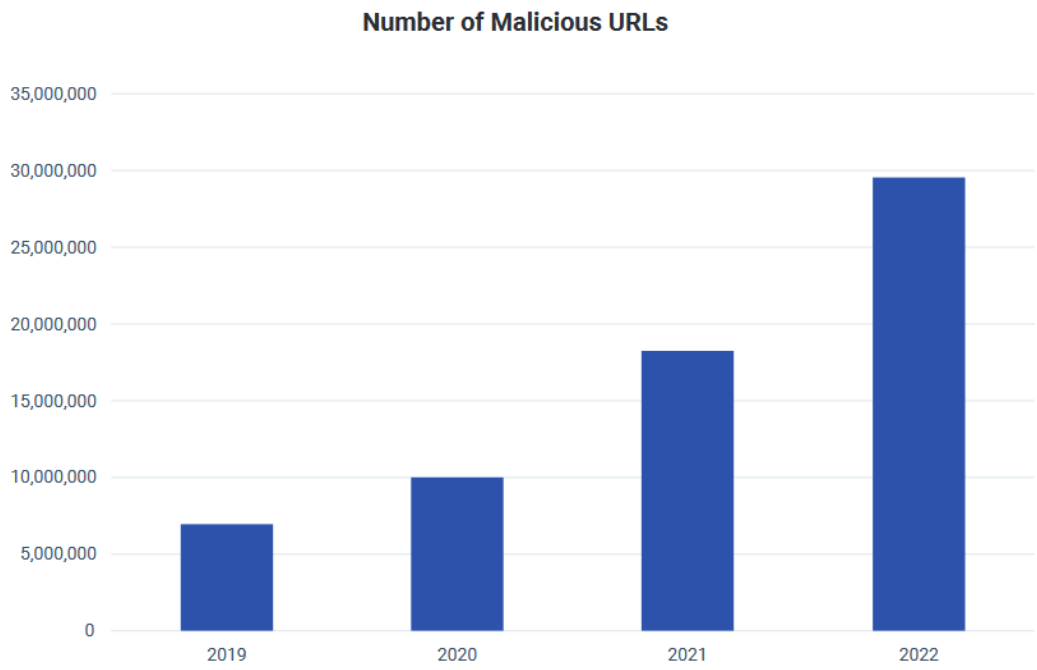


Figure 1. 2 Malicious URLs 255M phishing attacks detected in 2022.

According to the Phishing Activity Trends Report Q4 2022 by the Anti-Phishing Working Group (APWG)[5], a total of 1,270,833 phishing attacks occurred in the year 2022. This data signifies a record increase in phishing attacks. Additionally, SOCRadar's 2021 report detected a total of 42,136 phishing attacks targeting millions of consumers in the past 12 months[6]. According to Kaspersky's 2022 report, in the second quarter of 2022 alone, 3,990,546 phishing attacks were identified in Turkey[7].

In this study, we propose an innovative approach consisting of two steps for detecting Turkish phishing websites. In the first step, we created a deep learning-based model using an LSTM network[9]. We trained this model with our own phishing dataset, achieving a model that accurately predicted phishing sites with a rate of 91.6%. Upon reviewing existing literature, we observed that this success rate is relatively low for phishing prediction. Therefore, we decided to take a second step.

[5] https://docs.apwg.org/reports/apwg_trends_report_q4_2022.pdf
[6] https://socradar.io/wp-content/uploads/2022/02/2021-Turkey-Threat-Landscape-Report-TR-2.pdf
[7] https://www.kaspersky.com.tr/about/press-releases/2022_turkiyede-kimlik-avi-ve-dolandiricilik-girisimleri-2022nin-ikinci-ceyreginde-79-artisla-tavan-yapti

In the second step, we contemplated the common characteristics of phishing websites. Our observations revealed that phishing sites often have newly acquired domain names, domain expiration dates usually set for one year, SSL support is lacking in over half of them, and for those with SSL, certificates typically begin recently and have a validity period of around 90 days, often provided by 'Let's Encrypt.' Combining these common features with the phishing score obtained from the pre-trained deep learning model, we developed a deep learning model capable of predicting whether a URL is phishing or not. For classification in the deep learning model, we employed Logistic Regression, Decision Trees, Random Forest, and SVM. The most successful model was observed to be the one generated using the Random Forest method. The created model accurately predicted websites in our test dataset with a rate of 98.4%. Furthermore, in this study, we utilized publicly available data from the Turkish National Cyber Incident Response Center (USOM), containing over 250,000 phishing sites, to create training and test datasets.

2. THEORETICAL FOUNDATIONS AND LITERATURE REVIEW

Efforts to prevent phishing attacks can be categorized into rule-based approaches and intelligent automatic phishing detection. In the cyber world, one of the most effective and crucial measures is to detect an attack or vulnerability promptly and generate a temporary solution instantly. At this point, the detection of attacks known as Zero-Day attacks is of utmost importance[10], [11].

In systems operating on rule-based principles, static measures are taken against known attack techniques. The simplest approach involves blacklisting the domain names or URL addresses of websites used for phishing, thereby blocking access to the relevant site or Uniform Resource Locator (URL). The operation of rule-based systems involves controlling users' internet access within a certain flow. When a site or URL is marked as unsafe in the system's database, access is restricted. The list of unsafe sites is regularly updated by pulling information from specific sources, ensuring that security firewalls are continually kept up to date [12]–[14].

The next stage of rule-based operation can be considered as scenario-based operation. In scenario-based operation, an algorithmic structure is created. Security firewalls in this scenario can perform phishing detection more intelligently using advanced algorithms. By evaluating the URL structure, a certain ratio is determined for the phishing potential or probability of the respective URL. One of the most critical indicators in this determination is the domain name of the URL. If the relevant domain name matches a popular domain name with a high potential for phishing, the probability of the URL being a phishing URL is also high[1]–[3].

Additionally, there are dynamic approaches that make decisions based on content-based analyses. In this analysis, the URL address, HyperText Markup Language (HTML) codes of the content at the address, text content, and embedded links are used. The analysis results provide strong evidence regarding the presence of grammar errors, suspicious patterns, or observed contradictions in the content, helping to determine whether the site is legitimate or a phishing site. Methods that perform visual similarity comparisons alongside content-based analysis are also available. Here, the suspicious website is likened to a legal website, raising

the potential for deceiving the user. By visually analyzing the suspicious website alongside legal ones, it is possible to detect whether a copy of the legal site has been created[15]–[17].

In addition to visual and content analyses, other popular methods involve URL analysis to decide whether a website is legal or phishing. Syntax analysis of the website's URL, analyses based on the number of repeating characters, and similarity analyses are conducted. Systems that make decisions based on user behavior assist in determining whether a website is a phishing site by considering user actions such as mouse movements and keyboard usage. In [18], conducted a study using cryptographic hashing of each page's rendered Document Object Model (DOM), providing a zero false-positive rate and identifying more than half of detectable phishing attempts in a controlled study. The study [19] developed a phishing detection method that compares texts, text styles, embedded images, and overall site views of two sites to distinguish between legal and phishing sites. In study[20], used text format, text content, embedded images, HTML tags, and CSS in their analysis. In the study [10] VisualPhishNet is created, a triple-layered Convolutional Neural Network (CNN) based method that produces visual similarity scores between phishing websites and legal ones.

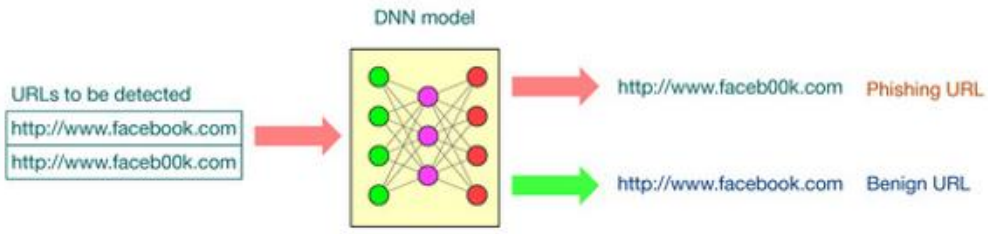


Figure 2. 1 High level overview of the deep learning-based phishing classifiers

In URL-based approaches, training is conducted by extracting features from URL information or by directly inputting the URL itself into a machine learning model. Deep learning models, particularly those based on Long Short-Term Memory (LSTM) and/or Bidirectional Long Short-Term Memory (BiLSTM), are commonly used in these supervised learning scenarios. [15] developed a real-time anti-phishing protection system where they tested seven different classification algorithms on their custom dataset. Using word vectors, Natural Language

Processing (NLP) features, and hybrid features, they achieved a remarkable 97.98% success rate with the random forest classification method. In [21] a combination of machine learning (SVM) and Fuzzy Logic in URL-based phishing detection produced successful results. The study [22] has achieved high success in their work using a character-based CNN approach. It is crucial to note that phishing attacks are continuously evolving, and no method is flawless. Therefore, combining multiple detection techniques will enhance the overall effectiveness of a phishing detection system.

3. MATERYAL AND METHOD

3.1. Dataset Preparation

Phishing detection commonly relies on supervised learning models in both machine learning and deep learning approaches. Expertly labeled datasets are crucial for these methods, utilizing either content, URL, or a combination of both. Phishing-labeled datasets often leverage platforms like PhishTank, while legitimate sites are labeled using datasets such as Alexa [23], [24].

Phishing detection is a dynamic process, with phishing sites evolving daily to closely mimic legitimate ones and employing various techniques to evade detection. Given the advancements in web technologies and the diverse range of techniques employed, it is evident that phishing datasets need constant updates. Therefore, the necessity to refresh phishing datasets is evident due to the continuous influx of user complaints, automatic phishing detection systems running consistently, and notifications from voluntary cybersecurity organizations.

In this context, the idea of creating a phishing dataset from the National Cyber Incident Response Center (USOM), which continuously updates itself based on user complaints and voluntary reports from cybersecurity organizations, has gained prominence. This dataset would consist of always-up-to-date sites, addressing both current and relevant domain names.

3.1.1. Reliable Source for Up-to-Date Phishing Sites (USOM)

The National Cyber Incident Response Center (USOM) was established within the Information Technologies and Communication Authority (BTK) to identify threats to Turkey's cybersecurity in the digital environment, develop measures to reduce or eliminate the impact of potential cyberattacks and incidents, and share these measures with relevant actors. Specifically, web pages and URLs created for phishing purposes, targeting users in Turkey, are continuously detected by USOM through its automated methods or reported by reliable sources.

Websites and URLs confirmed to be created for phishing purposes are promptly blacklisted by USOM and added to the list of malicious URLs published by USOM. As of November 2023, this list is increased with 400-500 phishing sites daily, accumulating to more than 260,000 malicious URLs so far. Government institutions and large-scale private companies use this list shared by USOM to keep their own firewall systems updated with information about malicious URLs.

This list serves as a reliable data source for researchers aiming to develop dynamic methods for detecting Turkish-language web pages created for phishing purposes. The URLs provided in this list are verified by experts as phishing sites, making it a trustworthy data set for researchers using supervised learning methods to automatically determine whether a site is a phishing site or a legitimate one. An example list is given in Figure 3.1.

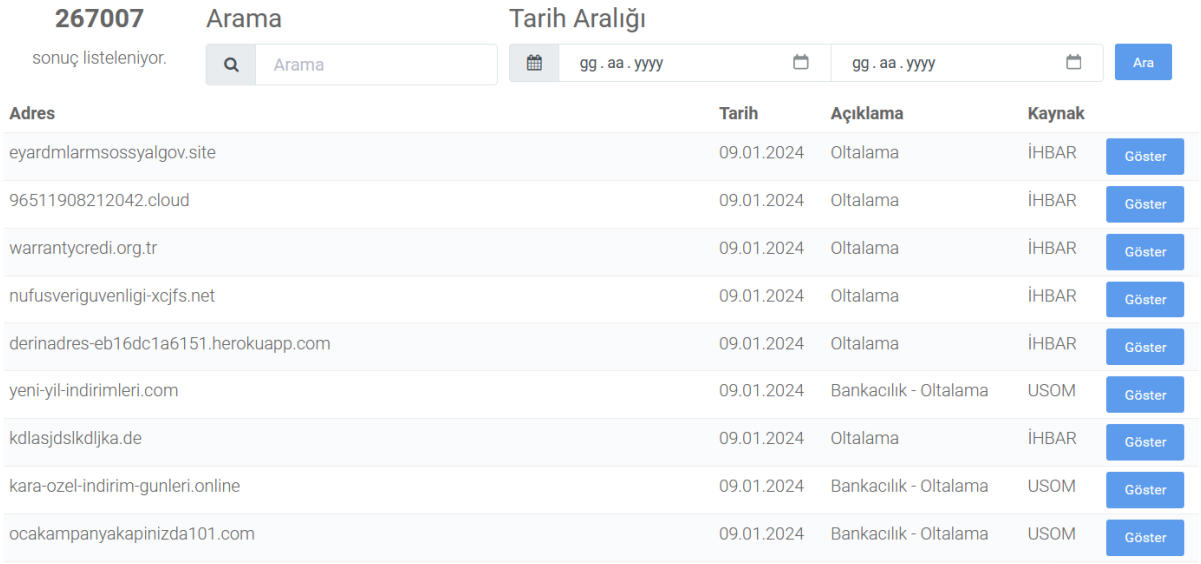


Figure 3. 1 USOM Website Phishing Urls List

Phishing websites exhibit noticeable characteristics, which can be outlined as follows;

a. These sites are typically created around topics that can arouse curiosity or concern in individuals, often related to current national issues. If there is a legal counterpart, the phishing site's name is designed to resemble it. They are promoted through visuals on social media, SMS, WhatsApp, mobile applications, etc., with links supported by images to entice users to click impulsively.

b. Phishing sites are frequently opened regarding issues such as notices, payments, information, cash rewards, campaign participation, for example, vehicle inspection, fast-pass system, tax payment, traffic fines, complaints about you, metropolitan transportation card balance loading, Bim credit loading, copyright infringement, secure e-commerce by owner, individual credit application, mobile banking, electronic payment, social aid payment, housing development administration lottery, New Year's drawing, New Year campaign, internet banking, and similar topics. New phishing sites are continuously created in these and related areas.

c. Site names often mimic those of legal sites that are popular in the given field, with characters and certain characters hidden among the characters representing the site name in a way that is not easily noticeable.

d. Users are warned by browsers that phishing sites without Secure Hypertext Transfer Protocol (HTTPS) are not reliable, and this warning is ignored when proceeding. In contrast, HTTPS sites open directly without any security warnings.

e. Phishing sites with valid and reliable certificates (HTTPS) have some notable features.
f. First, most certificates are generated with "Let’s Encrypt."
g. The creation dates of certificates are almost always one or two days ago.
h. The expiration periods of certificates are generally 90 days.
i. Most domain names are obtained through the same domain name provider.
j. Start times are usually the day the site is registered or one or two days before.
k. Domain name durations are typically limited to one year.
l. When examining the contents of phishing sites, it is observed that many are produced by the same hands and with the same template.

Two types of datasets have been created using web pages on USOM. The content, creation methods, and usage purposes of both datasets are provided in the relevant subsections.

3.1.2. Creating Dataset1

As of March 15, 2023, the domain names of all phishing sites in the first dataset have been collected and saved in a CSV file. Approximately 200K phishing websites were obtained from this process. As seen in the "Method" section, these domain names from Dataset1 will be used as training and testing data for the deep learning model. However, for this dataset to be used as training data in a deep learning classifier, we also need legal website names of the same size. A dataset of approximately 200K Turkish domain names, legally in use, was obtained from a public web page[8].

In total, a dataset has been created for training the deep learning model, consisting of approximately 168K legal and 165K phishing data, totaling around 333K data points. This dataset will be referred to as Dataset1 in the subsequent stages of the study.

3.1.3. Creating Dataset2

Following the establishment of the first dataset, a new dataset was prepared for use in the machine learning-based classifier in the second stage. The main objective in creating this dataset is to develop a machine learning classifier that classifies based on the phishing score provided by the deep learning model, created with the first dataset, along with other distinctive

[8] https://github.com/agmmnn/tr-domains

features of the web site. To construct this dataset, USOM's phishing sites were retrieved daily, and certain features of the sites were queried. As mentioned earlier, these features are common characteristics of phishing sites on USOM. Every night, using an agent script, phishing sites for that day were pulled, and the following features were queried using the same script: Domain registration start date, domain registration end date, domain registrar, SSL certificate start date, SSL end date, SSL certification issuer. This information was queried and stored in a comma-separated values (CSV) file. The ultimate goal is to train the machine learning classifier using a dataset created from the information queried every day. A total of 21K phishing and 24K legal websites were used to create the second dataset. It will be referred to as Dataset2 in the subsequent stages of the study.

3.1.3.1. Data Collection

USOM's website publishes lists of phishing sites daily, available either as text files or in XML format. These lists are crucial for corporate firewall devices, allowing them to update their phishing site databases daily. A Python script has been developed to connect to the USOM website every night and retrieve the most up-to-date list of phishing sites. After obtaining the list, it is compared with the list from the previous day to identify newly added sites. The local database is then updated with the identified sites. Besides updating, certain features of phishing sites are queried for use in Dataset2.

The queried features include:

- Domain registration start date.
- Domian registration end date
- SSL certificates start date.
- SSL end date.
- SSL certification issuer

These details are individually queried and saved. The saved information is also marked separately as phishing sites.

To perform machine learning and deep learning-based classification explained in the methodology, it is necessary to collect data not only from phishing sites but also from legal sites. For this purpose, legal sites related to the same topics as phishing sites are typically collected, as described in the first dataset.

3.2. Preliminary Preparation

In the first stage of the study, deep learning methods were employed to analyze the character sequences of URL addresses. Specifically, a type of Recurrent Neural Network (RNN) known as Long Short-Term Memory (LSTM) deep learning networks, capable of processing this in a sequential manner, were utilized. Subsequently, in the second phase, features related to the domain name and security certificate of the URL, along with the phishing website score generated by this deep learning model, were used to train machine learning models. The final classification result was obtained through this process. In this section of the study, basic explanations will be provided regarding how the employed deep learning and machine learning techniques generally operate.

3.3. Deep Learning Techniques

Information was given about the deep learning techniques used in the study.

3.3.1. Recurrent Neural Network (RNN)

Recurrent Neural Network (RNN) is a type of deep learning architecture known as Recurrent Neural Networks. Unlike other neural networks, RNNs are designed for situations where the order of inputs is important. Inputs are not independent of each other; thus, the outputs generated after processing an input can be reused in the next step. This is particularly suitable for handling sequential states and time series. RNNs store previous states in their memory through a loop added to the system, and these states can be used in the next step. This is useful for modeling situations where inputs are correlated and for creating memory.

RNNs are especially useful when dealing with sequential data or time series because they can attempt to predict future states by understanding past states. RNN is given in Figure 3.2.

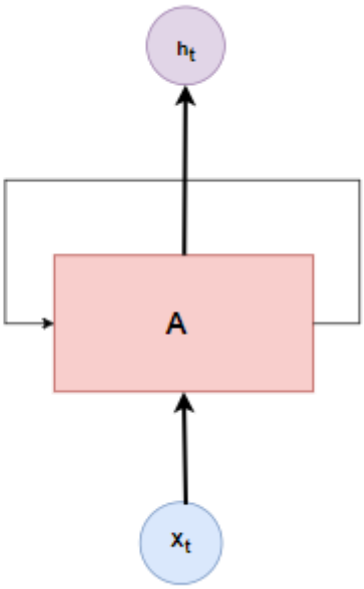


Figure 3. 2 The structure of a Recurrent Neural Network

In the diagram above, a portion of the neural network, denoted as A, looks at some input x_t and produces an output value h_t. If we break down the loop shown in the figure:

RNN takes the input x_0 and combines it with the current hidden state. This process produces an output containing information about the current state and updates the hidden state. Then, the X1 input and the updated hidden state are taken, provided as input to the next layer, and after the operations are completed, the hidden state is updated again. This process means that whenever the model receives a new input, it takes the information about the previous inputs along with that input. This way, information about past inputs is retained in memory. There is an RNN cell whose unrolling is shown in Figure 3.3.

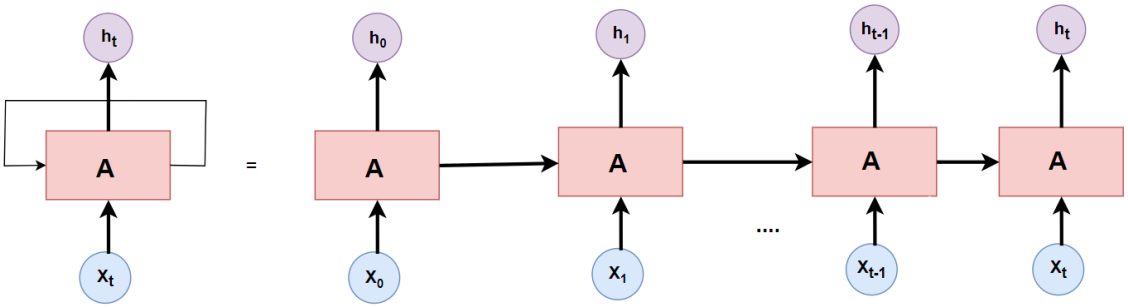


Figure 3. 3 Unrolling State RNN cells.

Formula for calculating the current state:

$$h_t = f(h_{t-1}, x_t) \tag{3.1}$$

- h_t -> current state
- h_{t-1} -> previous state
- x_t-> input state

Formula for applying the tanh activation function:

$$h_t = \tanh(w_{hh}h_{t-1} + w_{xh}x_t) \tag{3.2}$$

- w_{hh} - - > Weight on the recurrent neuron:
- w_{xh} - - > Weight on the input neuron:

Output calculation formula:

$$y_t = w_{hy}h_t \tag{3.3}$$

- y_t ---> Output
- w_{hy} ----> Weight in the output layer:

However, RNNs encounter two problems during recurrent training:

-Vanishing Gradient Problem: This occurs during the training of deep recurrent structures. When long time series or deep recurrent networks are used, the gradients become very small over time. During backpropagation, the updating of information from previous time steps becomes very weak, hindering the effective learning of long-term dependencies by the model.

-Exploding Gradient Problem: This occurs when the gradients reach extremely large values during training. It is typically observed in large recurrent networks or high-scale models. The excessive growth of gradients can lead to very large parameter updates, causing the model to become unstable.

To address both problems, several solution strategies can be employed, such as adjusting the activation functions or parameters used in the architecture, implementing gradient clipping to ensure gradients do not exceed a certain threshold, or using specialized RNN cells like LSTM (Long Short-Term Memory) and GRU (Gated Recurrent Unit), which are designed to alleviate these issues.

3.3.2. Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) networks are specially designed structures to address the issue of long-term dependencies. In other words, they excel at remembering information for extended periods. This ability makes them particularly useful for understanding long-term dependency relationships that might be challenging to learn otherwise. Figure 3.4 available for example LSTM.

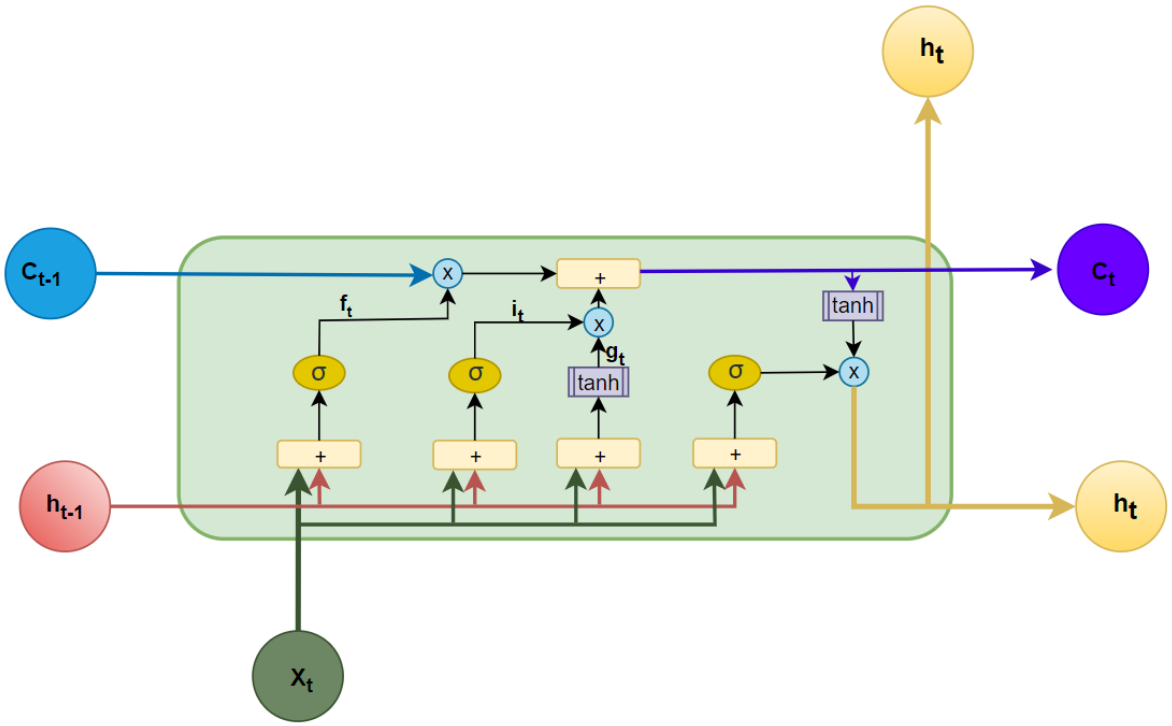


Figure 3. 4 The LSTM structure model

At the core of the successful operation of an LSTM are three gates: the "input gate," the "forget gate," and the "output gate." These gates regulate the cell state and cell output of the LSTM, enabling it to effectively learn long-term dependencies.

I. Forget Gate:

- Forget Gate: Determines which information to extract from the cell state.
- This gate assists in cleaning up unnecessary or irrelevant information by deciding how much of the information in the cell state should be forgotten.
- It involves the sigmoid activation function.
- Produces a value between 0 (completely forget) and 1 (completely remember).

Forget Gate working principle is as Figure 3.5.

Equation:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (3.4)$$

In this formula:

* f_t is the output of the forget gate.

* σ represents the Sigmoid function.

$*$ W_f denotes the weight matrix.

* $[h_{t-1}, x_t]$ is the combination of the previous time step's hidden state and the current input vector.

* b_f is the bias term.

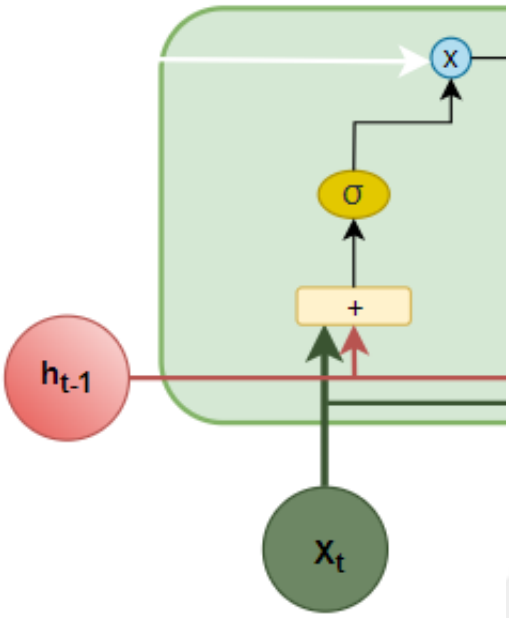


Figure 3. 5 The Forget Gate Structure

The output of the forget gate, denoted as f_t, takes values between 0 and 1 due to the sigmoid function. This value determines which information will be discarded from the cell state. If f_t, is close to 1, then the cell state from the current time step is preserved entirely. Conversely, if it is close to 0, the information in the cell state from the current time step is largely forgotten. This flexibility in adjusting how much past information the forget gate will retain enables the LSTM to effectively handle long-term dependencies in the data.

II. Input Gate:

- The input gate determines which new information will be added to the memory (cell state).
- This gate decides which portion of the incoming new information is relevant and should be added to the cell state, thereby determining its importance.
- It includes a sigmoid activation function.
- It also incorporates the hyperbolic tangent (tanh) activation function, which produces values between -1 and 1.

Input Gate working principle is as Figure 3.6.

Equation:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (3.5)$$

$$g_t = \tanh(W_g \cdot [h_{t-1}, x_t] + b_g) \quad (3.6)$$

In this formula:

***** i_t represents the output of the input gate.

* σ signifies the sigmoid function.

* W_i denotes the weight matrix.

***** $[h_{t-1}, x_t]$ is the combination of the previous time step's hidden state and the current input vector.

* bi is the bias term.

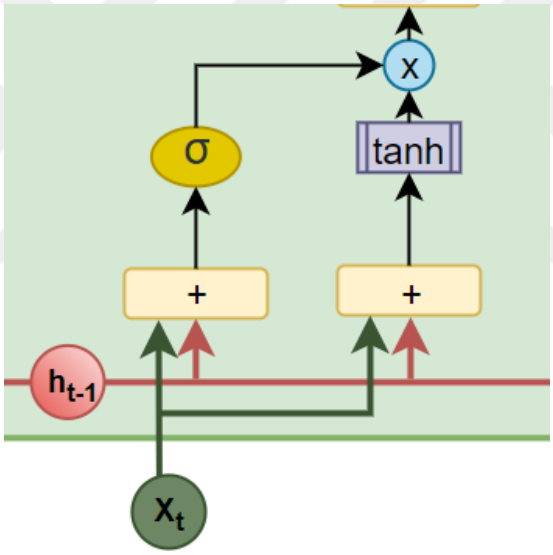


Figure 3. 6 Input Gate Structure

The output of the input gate, it, takes values between 0 and 1 (due to the sigmoid function). This value controls the amount of new information to be added to the cell state. If it is close to 1, then the input information at that time step will be fully added to the cell state; if it is close to 0, the information will be completely ignored. This allows the input gate to determine which information is crucial during the learning process of the LSTM.

III. Cell State Update

It enables forgetting the past memory and adding new information. The cell state update is shown in Figure 3.7.

Equation:

$$C_t = f_t . C_{t-1} + i_t . g_t \quad (3.7)$$

In this formula:

- C_t represents the new cell state.
- f_t is the output of the forget gate.
- C_{t-1} represents the previous cell state.
- i_t is the output of the input gate.
- g_t is the updated cell candidate.

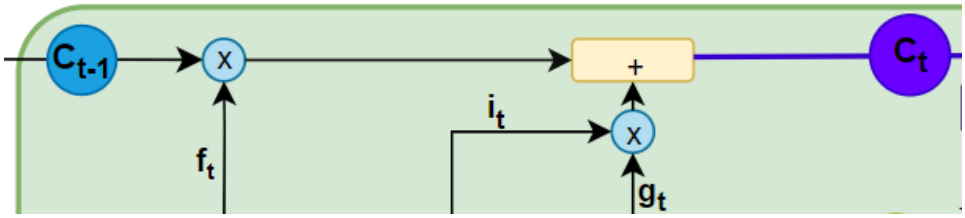


Figure 3. 7 The Update of The Cell State

These formulas represent the memory update mechanism of LSTM. In the first formula, a weighted sum is performed between the forget gate f_t and the previous cell state C_{t-1}, and another weighted sum is performed between the input gate i_t and the updated cell candidate g_t. The result obtained is used as the new cell state C_t. This mechanism illustrates how the memory cell is updated to forget past information and learn new information.

IV. Output Gate:

The output gate determines the value to be output from the updated memory. The output gate is shown in Figure 3.8.

Equation:

$$O_t = \sigma(W_o.[h_{t-1}, x_t] + b_o) \quad (3.8)$$

$$h_t = O_t \cdot \tanh(C_t) \quad (3.9)$$

In this formula:

- O_t is the output of the output gate.
- σ represents the Sigmoid function.
- W_o denotes the weight matrix.
- $[h_{t-1}, x_t]$ is the combination of the previous time step's hidden state and the current input vector.
- b_o is the bias term.

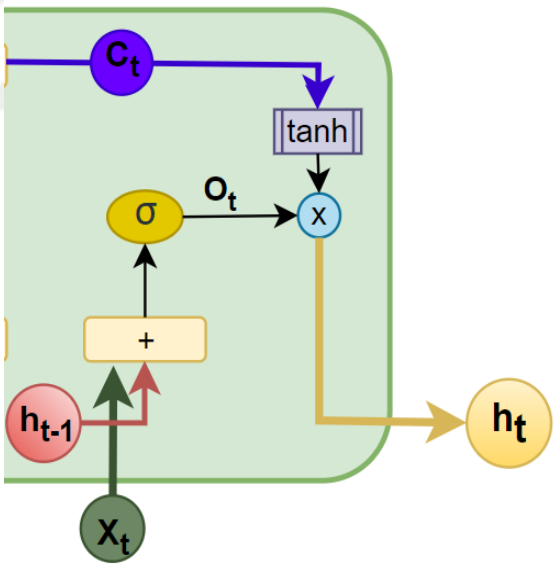


Figure 3. 8 Output Gate Structure

The output of the output gate O_t takes a value between 0 and 1 (due to the sigmoid function). This value determines how much information the current hidden state at the time step contributes to the overall output of the model based on the cell state. The output gate controls how the information from the cell state contributes to the model's general output.

Advantages:

Learns Long-Term Dependencies Effectively:

The fundamental design purpose of LSTM is to learn and retain long-term dependencies more effectively. This feature enables it to successfully handle longer-term connections in applications such as time series or language modeling.

Cellular Structure:

The cellular structure of LSTM, with gates that control the cell state and output separately, provides a more flexible learning mechanism.

Mitigates the Vanishing Gradient Problem:

LSTM is designed to mitigate the vanishing gradient problem encountered in traditional neural network models. This allows it to more effectively learn dependencies that occur over longer time intervals.

Successful in Various Applications:

LSTM has proven successful in many applications, especially in areas such as language modeling, speech recognition, text translation, and time series analysis.

Disadvantages:

High Computational Power Requirement:

Due to the complexity of LSTM models, training and using them on large datasets require more computational power.

Difficulty in Hyperparameter Tuning:

LSTM has several hyperparameters, and finding their optimal values can be challenging. This can affect the model's performance, and hyperparameter tuning can be time-consuming.

Need for Sufficient Data:

LSTM models sometimes perform better with more data. When data is limited, issues such as overfitting may arise.

Black Box Model:

The complex internal structure of LSTM can make it difficult to understand how the model makes decisions. This can be a disadvantage for users who want to understand and interpret the internal workings of the model.

3.4. Machine Learning Based Classification Methods

In the first stage, a model will be developed using deep learning techniques to predict the phishing score based on the character sequence in the website's domain name using Dataset1. In the second stage, Dataset2 will be used to develop a machine learning model capable of binary classification. Binary classification will be modeled using commonly preferred and effective techniques such as Logistic Regression, Support Vector Machines, Decision Trees, and Random Forest. Therefore, below is some basic information about these models.

Classification is a significant task in machine learning and statistics. Essentially, it is the process of assigning a data point to a specific category or class. These categories are commonly referred to as labels or classes. Classification algorithms are designed to assign new and unknown data points to specific classes using a model learned on training data.

3.4.1. Decision Trees

A decision tree model Figure 3.9 is a supervised learning algorithm that is effectively used for classification and regression tasks. This algorithm utilizes a tree structure to understand patterns in the dataset and predict a specific target variable. The tree structure performs a feature test at each node, starting from the root node, and based on the results of these tests, it divides the dataset into subgroups.

During training, the decision tree algorithm uses criteria such as entropy or Gini impurity to determine which attribute to select at each node. These criteria evaluate the homogeneity of the obtained subgroups after splitting the dataset. Stopping criteria, such as the depth of the tree and the minimum number of samples to split a node, help prevent overfitting.

Decision trees enhance the model's performance by maximizing information gain and splitting the dataset into more homogeneous subgroups. Due to its interpretability and high learning capacity, this method is preferred in various application domains.

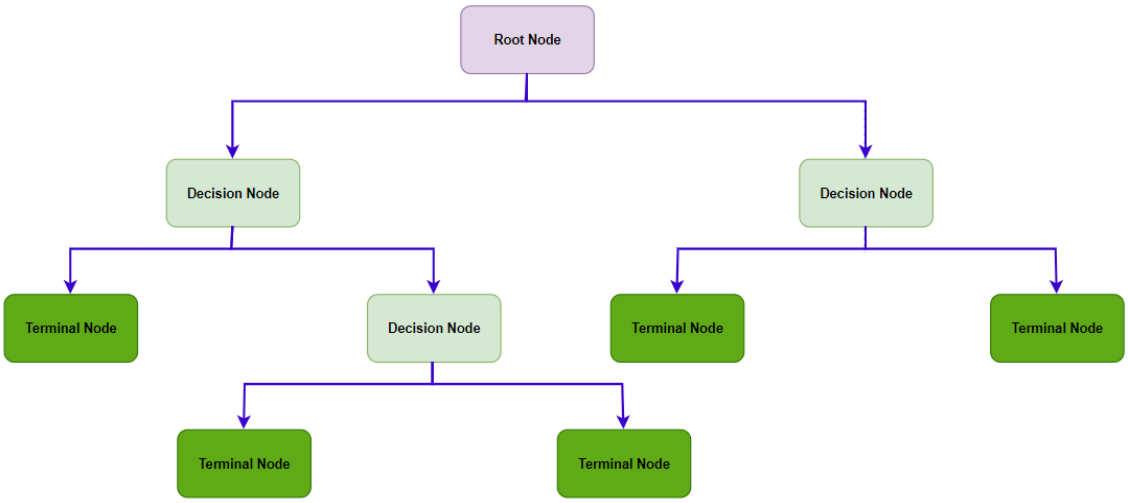


Figure 3. 9 Decision Tree Structure Model

Decision Tree Structure:

The proper execution of splitting is one of the factors that influence the accuracy of a decision tree. Through correct splitting, the tree achieves a more homogeneous and balanced structure.

In decision trees, Entropy and Gini index measures are used to accomplish this homogeneous separation.

Entropy Calculation:

Entropy is an information measure that quantifies the degree of homogeneity in a dataset. The more homogeneous the dataset, the lower the entropy.

The formula for entropy is as follows:

$$E(S) = 1 - \sum_{i=1}^{c} p_i \, log_2(p_i) \tag{3.10}$$

In this formula:

- $E(S)$, represents entropy.
- c, denotes the number of classes.
- p_i , represents the proportion of examples belonging to the i-th class to the total number of examples.

Gini Index Calculation:

The Gini index is another metric used to measure the homogeneity of a dataset. The Gini index quantifies the probability of misclassification of an example.

The formula for the Gini index is as follows:

$$Gini(S) = 1 - \sum_{i=1}^{c} p_i^2 \tag{3.11}$$

In this formula:

- $Gini(S)$, represents the Gini index.
- c, denotes the number of classes.
- p_i , expresses the ratio of examples belonging to the ii-th class to the total number of examples.

Information Gain sums:

Information Gain is a metric used in the process of creating a decision tree. When we divide a dataset based on a specific feature, Information Gain measures how homogeneous each partition is. Information Gain quantifies how much the feature reduces the uncertainty in a specific partition.

The formula is as follows:

$$information\ Gain = Entropy\ before\ split - \sum_{i=1}^{k} \frac{|S_i|}{|S|} x\ Entropy\ after\ split_i \quad (3.12)$$

In this formula:

- Entropy before split: Entropy of the dataset before the split.
- k: The number of subsets obtained after the split.
- $|S_i$ The number of examples in the i-th subset.
- |S|: The total number of examples.
- Entropy after split: Entropy of the i-th subset after the split.

Decision Tree Advantages:

- Decision trees are easy to understand and interpret due to their tree structure. This feature provides an advantage to users in explaining and comprehending the model's results.
- They often do not require preprocessing steps such as normalization or feature engineering. Trees can easily handle both numerical and categorical data types.
- They have the ability to deal with situations like class imbalances. Trees can address imbalanced classes using homogeneity criteria.
- Decision trees usually do not require complex parameter tuning compared to many other models. This allows users to quickly and easily build a model.
- Decision trees can be used for both classification and regression tasks, supporting both categorical and numerical outputs.

Disadvantages of Decision Trees:

- They are prone to overfitting on training data, meaning they may fit the training data very well but lose generalization ability.
- Decision trees are sensitive to noisy data or small changes in data. Performance may decline in the presence of noise or errors in training data.

- They have limitations in modeling nonlinear relationships. Some other models can capture more complex relationships compared to decision trees.
- Decision trees tend to favor learning more from classes with larger sample sizes in imbalanced datasets. This may result in poor classification of classes with fewer examples.
- The structure of the tree can change based on variations in the dataset, impacting the model's stability.

3.4.2. Logistic Regression

Logistic Regression is a supervised machine learning algorithm primarily used for classification tasks. Its main objective is to predict the probability of an instance belonging to a particular class. This algorithm is a statistical method that analyzes the relationship between a series of independent variables and a dependent binary variable. It serves as a powerful tool in decision-making processes; for example, it can be used to predict whether an email is spam or not.

Logistic regression is employed to predict a categorical dependent variable using a given set of independent variables. It transforms the continuous output of a linear regression function into a categorical output using the sigmoid function figure 3.10. This sigmoid function, also known as the logistic function, converts the set of real numbers into a value between 0 and 1.

In summary, logistic regression is a model used in classification tasks. It predicts the categorical probability of the dependent variable using the set of independent variables and transforms this prediction into a value between 0 and 1 through the sigmoid function.

Let the independent input features be:

$$\begin{bmatrix} x_{11} & \ldots\ldots & x_{1m} \\ x_{21} & \ldots\ldots\ldots & x_{2m} \\ . & . & . \\ . & . & . \\ . & . & . \\ x_{n1} & \ldots\ldots & x_{nm} \end{bmatrix} \quad (3.13)$$

And let the dependent variable Y have only binary values, consisting of 0 or 1:

$$Y = \begin{cases} 0 & if\ Class\ 1 \\ 1 & if\ Class\ 2 \end{cases} \tag{3.14}$$

Next, apply the multiple linear function to the input variables X:

$$z = \left(\sum_{i=0}^{n} w_i\, x_i\right) + b \tag{3.15}$$

Here, w_i represents the weights or coefficients, $w_i = w_1, w_2, w_3, \ldots . w_m$ for the i-th observation of X, and b is the bias term, also known as the intercept. This can be simply represented as the dot product of weights and bias.

$$z = w.X + b \tag{3.16}$$

Now, using the sigmoid function with input z, we find the probability y_hat, which is the estimated y in the range of 0 to 1.

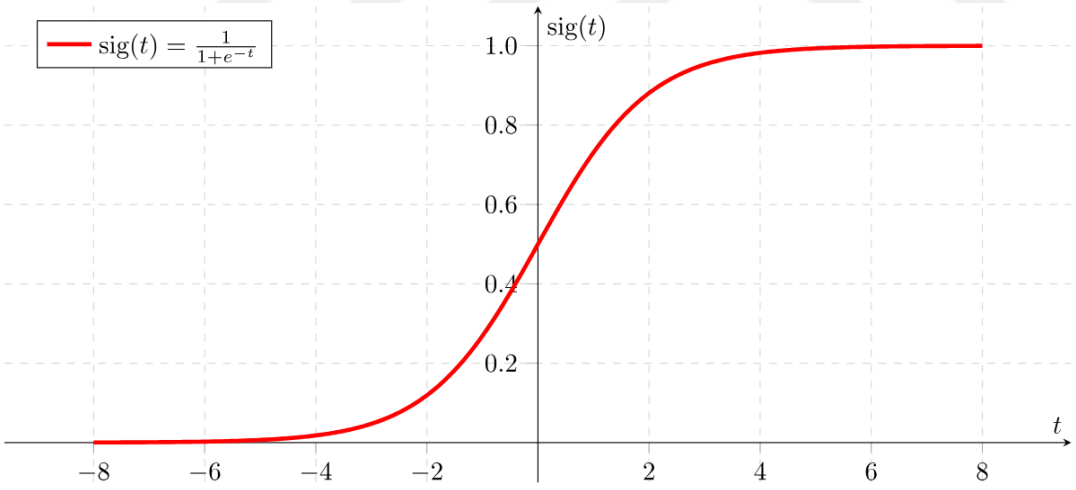


Figure 3. 10 Sigmoid-Function

As shown above, the sigmoid function transforms continuous variable data into a probability range between 0 and 1.

Logistic Regression Equation:

The odd is the ratio of something occurring to something not occurring. it is different from probability as the probability is the ratio of something occurring to everything that could possibly occur. so odd will be:

$$\frac{p(x)}{1-p(x)} = e^{z} \tag{3.17}$$

Applying natural log on odd. then log odd will be:

$$\log\left[\frac{p(x)}{1-p(x)}\right] = z \tag{3.18}$$

$$\log\left[\frac{p(x)}{1-p(x)}\right] = w.X + b \tag{3.19}$$

then the final logistic regression equation will be:

$$p(X; b, w) = \frac{e^{w.X+b}}{1+e^{w.X+b}} = \frac{1}{1+e^{-w.X+b}} \tag{3.20}$$

3.4.3. Support Vector Machines

Support Vector Machine (SVM) is a supervised machine learning algorithm used for both classification and regression. Although it can be used for regression problems, it is generally more suitable for classification problems. The primary goal of the SVM algorithm is to find an optimal hyperplane in the feature space that can separate data points into different classes. The hyperplane aims to maximize the margin between the closest points of different classes. The dimension of the hyperplane depends on the number of features. If the number of input features is two, the hyperplane is just a line. If the number of input features is three, the hyperplane becomes a 2-D plane. Visualizing this concept becomes challenging when the number of features exceeds three. Support Vector Machines are shown in Figure 3.11.

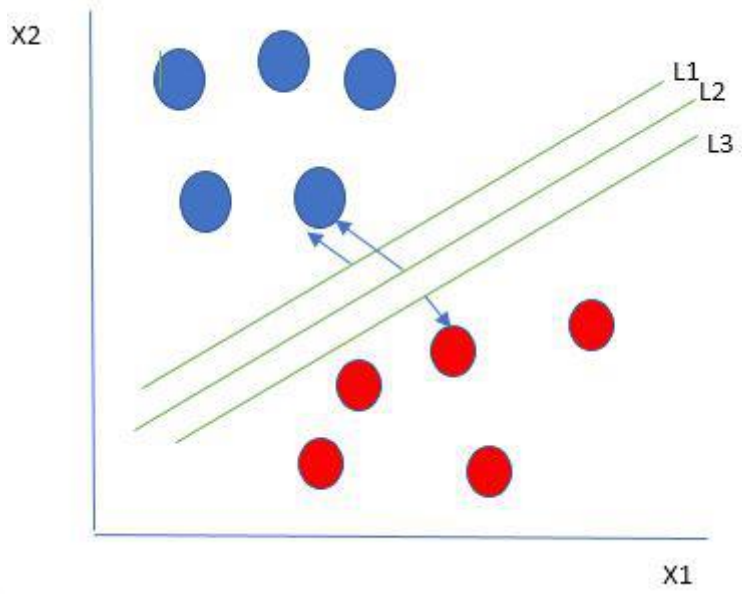


Figure 3. 11 Multiple Hyperplanes Separate The Data Into Two Classes

Support Vector Machine Mathematical Understanding

Let's consider a binary classification problem with two classes labeled as +1 and -1. We have a training dataset consisting of input feature vectors X and their corresponding class labels Y. The equation for a linear hyperplane can be written as:

$$w^T x + b = 0 \tag{3.21}$$

Vector *W* represents the normal vector pointing in the direction perpendicular to the hyperplane, i.e., the normal vector to the hyperplane. The parameter bb in the equation represents the distance or offset of the hyperplane from the origin along the normal vector *w*.

The distance between a data point x_i and the decision boundary can be calculated as follows:

$$d_i = \frac{w^T x + b}{||w||} \tag{3.22}$$

Here ||w||, represents the Euclidean norm of the weight vector ww. The Euclidean norm for the normal vector *W* is given by:

$$||W|| = \sqrt{\sum_{j=1}^{m} w_j{}^2} \tag{3.23}$$

For a linear SVM classifier:

$$\hat{y} = \begin{cases} 1 \ : \ w^T x + b \ \geq 0 \\ 0 \ : \ w^T x + b \ < 0 \end{cases} \tag{3.24}$$

Types of Support Vector Machine

The Support Vector Machine (SVM) can be divided into two main categories based on the nature of the decision boundary:

Linear SVM: Linear SVMs use a linear decision boundary to separate data points from different classes. Linear SVMs are highly suitable when data can be perfectly separated linearly. This means that a single straight line (in 2D) or a hyperplane (in higher dimensions) can completely divide data points into relevant classes. The hyperplane that maximizes the margin between classes is the decision boundary.

Nonlinear SVM: Nonlinear SVM comes into play when data cannot be separated by a straight line (in the case of 2D data). It is used to classify data that is not linearly separable. Nonlinear SVMs handle such cases by using kernel functions. The original input data is transformed into a higher-dimensional feature space where data points become linearly separable. In this modified space, a linear SVM is then employed to determine the position of a nonlinear decision boundary.

3.4.4. Random Forest Regression

Random Forest Regression is a versatile machine learning technique designed for predicting numerical values. It combines predictions from multiple decision trees to reduce overfitting and enhance accuracy. The working principle of Random Forest Regression is shown in Figure 3.12.

Ensemble Learning:

Ensemble learning is a machine learning technique that combines predictions from multiple models to generate a more accurate and stable prediction. It is an approach that leverages the collective intelligence of multiple models to improve the overall performance of the learning system.

Random Forest:

Random Forest is a type of ensemble learning method that combines predictions from multiple decision trees to produce a more accurate and stable prediction. It is a supervised learning algorithm that can be used for both classification and regression tasks.

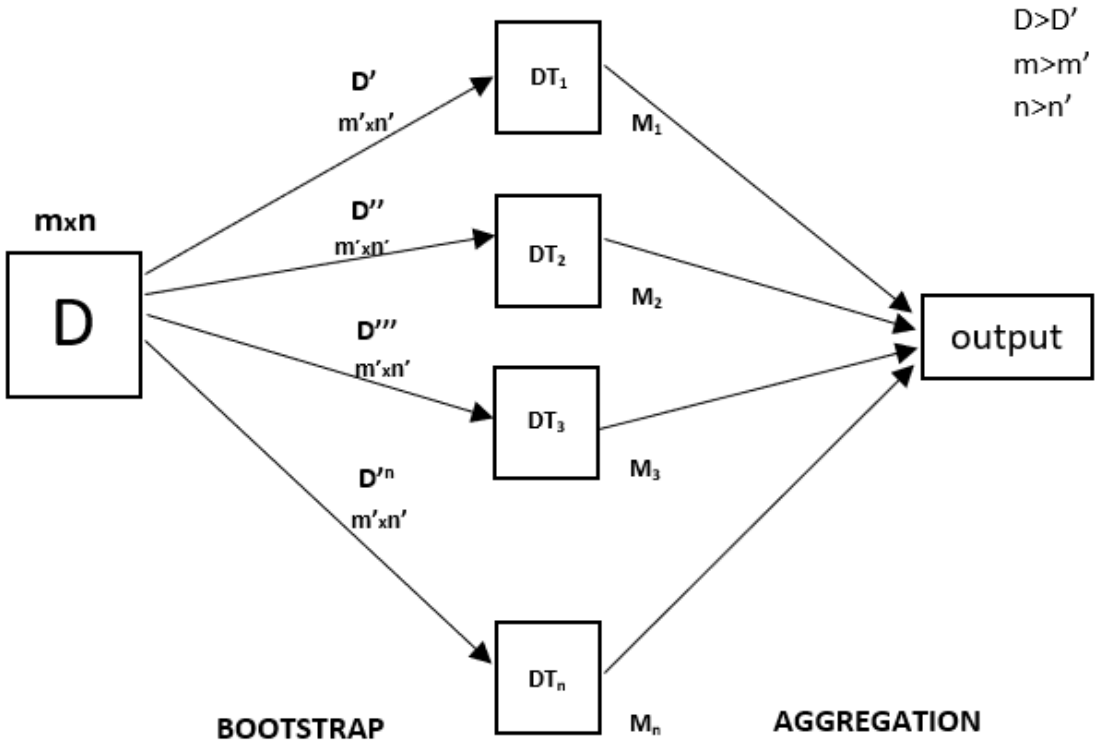


Figure 3. 12 Random Forest Regression Model Working

In machine learning, Random Forest Regression is a community technique capable of performing both regression and classification tasks using multiple decision trees and a technique commonly known as bagging, which involves Bootstrap and Aggregation. The

fundamental idea behind it is to combine multiple decision trees when determining the final output, instead of relying on individual decision trees.

Random Forest has multiple decision trees as its basic learning model. For each model, random row sampling and feature sampling are performed from the dataset, creating example datasets. This process is called Bootstrap.

The Bootstrap Random Forest algorithm combines community learning methods with the decision tree framework to create multiple decision trees randomly drawn from the data. By averaging the results, it typically produces robust predictions/classifications.

3.4.5. Machine Learning Classification Metrics

During the evaluation of machine learning classifications, Accuracy, Precision, Recall, and F1-Score are utilized. These metrics can provide insights about our model created during our research and whether there are errors in our model. True Positive (TP) denotes the correctly identified phishing attacks; True Negative (TN) represents non-phishing instances correctly classified as such; False Positive (FP) indicates non-phishing instances misclassified as phishing attacks; False Negative (FN) signifies phishing attacks misclassified as non-phishing instances.

Accuracy measures the percentage of correctly identified data and is calculated as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.25}$$

Precision is the percentage of phishing attacks correctly detected and is calculated as follows:

$$Precision = \frac{TP}{TP + FP} \tag{3.26}$$

Recall (Detection Rate) is the percentage of actual phishing attacks correctly identified and is calculated as follows:

$$Recall = \frac{TP}{TP + FN} \tag{3.27}$$

F1-Score is the weighted average of precision and recall.:

$$F1 - Score = 2 \times \frac{precision \times recall}{precision + recall} \quad (3.28)$$

3.5. Method

When examining the phishing sites in the dataset created for the first stage, a prominent observation is the apparent resemblance of their domain names to legitimate ones, with the introduction of additional letter characters to a level of detail that might go unnoticed by the user (e.g., garantibanksi, ypkredi, etc.). Conversely, in an effort to address contemporary issues and generate curiosity, phishing site names have been crafted by inserting different characters into normal Turkish words. Example names of phishing sites Table 3. 1. include:

Table 3. 1. Example phishing URLs.

Phishing Url Adress
pribuylayasa.online
firsathaftasisimdi.com
trbncsdestekgiris.com
evevlelgelletlelte.com
eturkiyegovyardm.website
garentayilsonufirsatlari.com.tr
sokmarket.store

At first glance, it is deemed that each of these URL addresses, when encoded at the character level and the encoded characters are sequentially fed into the LSTM network, can be effectively trained for binary classification. Structurally, it is observed in many studies in the literature that LSTM networks can learn such a sequence and make accurate predictions (Phishing methods using LSTM).

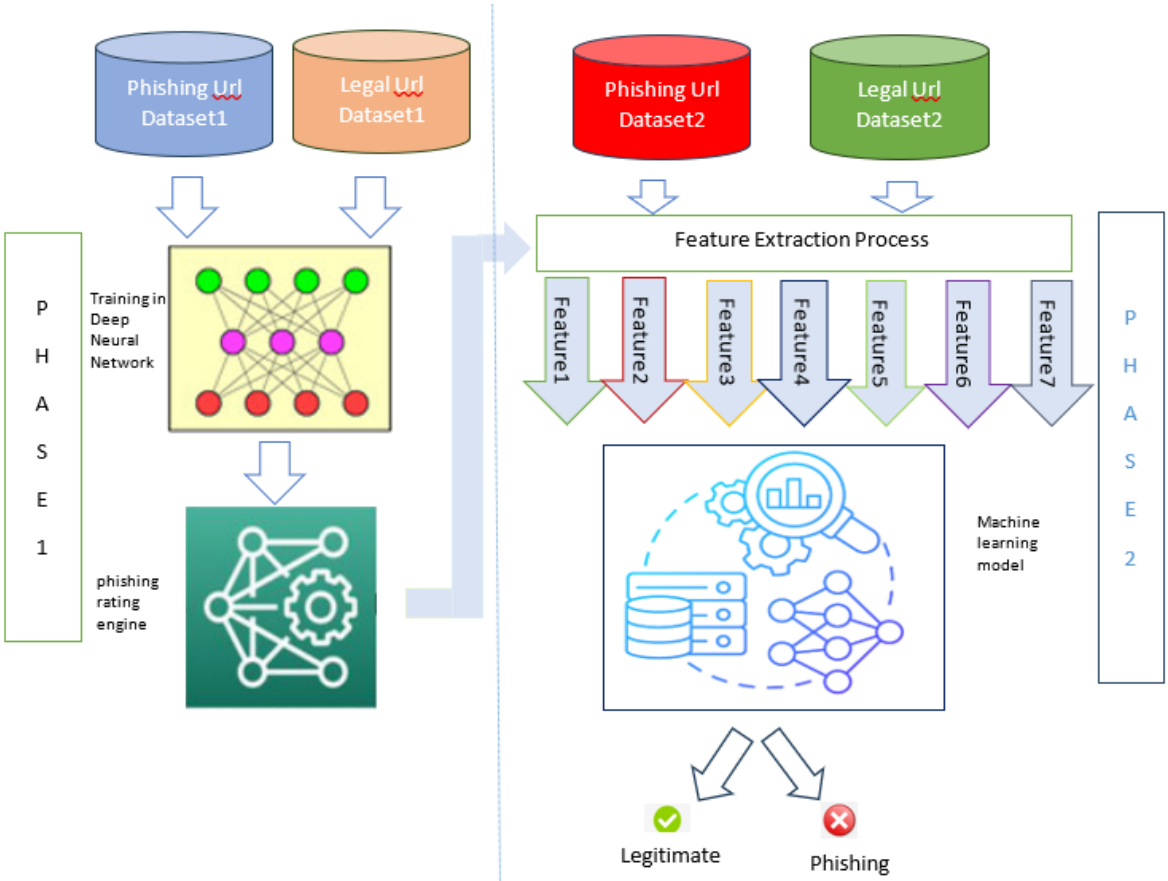


Figure 3. 13 End to end overview of the proposed method.

Feature1: Domain name
Feature2: Phishing rating score given by deep learning model
Feature3: Domain name registration start date (How many days passed after domain registration)
Feature4: Domain name registration end date (how many days, it will be active)
Feature5: ssl certificate start date (How many days passed after domain registration)
Feature6: ssl certificate end date (how many days, it will be active)
Feature7: ssl certificate issuer

Figure 3. 14 Features Of The Proposed Method

3.5.1. Dataset Preprocessing and Data Deduplication

USOM's official website provides a chronological list of all malicious websites detected by USOM when the link containing harmful connections is clicked. Additionally, malicious websites are presented to users as an XML or TXT file format or via API, alongside the web page. The XML file, provided as part of a Python script, is parsed, and information used during

model training is obtained from this source. The script parses the XML file, extracting the domain name of each malicious website. The parts other than the obtained domain name are not considered, as distinguishing features are found within the domain name, and the remaining parts resemble each other in both legal and phishing websites. Following the deduplication process, 168K phishing domain names are obtained since multiple URLs with the same domain name are listed.

Up until March 2023, USOM has shared a total of 168K phishing sites, constituting the phishing site addresses in Dataset1. However, for the deep learning model to perform classification, approximately an equal number of legal domain names are required. To achieve this, a list of .tr extension domain names is obtained from a webpage where .tr domain names are published. As the number of obtained .tr extension domain names exceeds 400 thousand, more than 200 thousand of them are removed to leave around 168K legal site names. Thus, a dataset containing both legal and phishing website names necessary for binary classification is obtained. This dataset is named Dataset1.

3.5.1.1 First Stage: Preparation of the Pre-trained Deep Learning Model

Long Short-Term Memory (LSTM) is a type of Recurrent Neural Network (RNN) structure, specifically used for processing time series, natural language processing, and other sequential data. LSTM distinguishes itself from other types of RNNs due to its ability to capture long-term dependencies. In fact, in this context, we can think of domain names as sequential data consisting of characters. Each sequentially arriving character vector enters the LSTM network, and a representation vector is obtained from it.

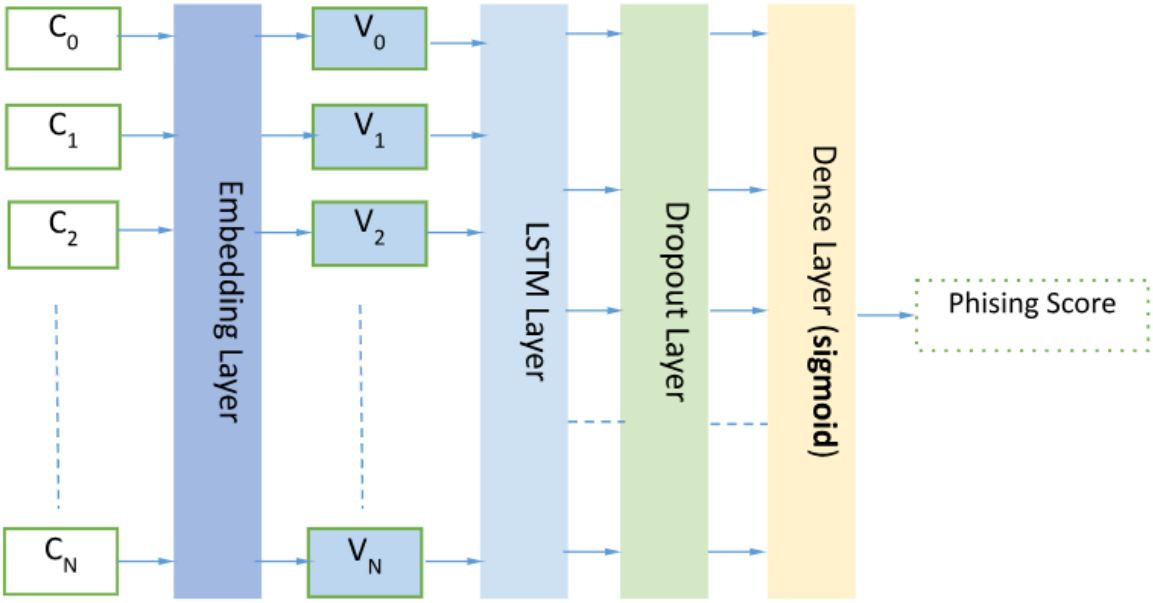


Figure 3. 15 The Schema Of The Created Deep Learning Model

As seen in Figure 3.15, in the input layer of the created deep learning model, characters related to the domain are first converted into character vectors through the embedding layer. The ASCII representation of each character is given as input to the embedding layer. The ASCII character code given as input to the layer is transformed into an 8-dimensional vector at the output. Although higher dimensions were also tried for the embedding size, it was observed that the best and fastest result was achieved at this dimension, so this dimension was decided upon. To prevent overfitting and memorization, a Dropout layer is added to the model. In the output layer, the sigmoid function is used to reduce the output to a two-dimensional vector. The first element of the vector produces a probability value between zero and one for whether the input is legal or not, while the second element produces a value between zero and one for whether the input is phishing.

The parameters used during the training of the model are as follows: The 80% of the available dataset, which consists of 336000 instances, is used for training, and the remaining 20% is used for testing purposes. When examining the results obtained, accuracy and consistency of up to 0.93 are achieved on the training dataset, while around 91.62% success is achieved on the test dataset. The deep learning training, which used a total of 125 epochs, resulted in the following loss and accuracy values for both training and validation in the last 5 epochs.

Epoch 120/125
2350/2350 [==============================] - 18s 7ms/step - loss: 0.1666 - accuracy: 0.9348 - val_loss: 0.2188 - val_accuracy: 0.9149
Epoch 121/125
2350/2350 [==============================] - 17s 7ms/step - loss: 0.1679 - accuracy: 0.9345 - val_loss: 0.2213 - val_accuracy: 0.9159
Epoch 122/125
2350/2350 [==============================] - 18s 8ms/step - loss: 0.1666 - accuracy: 0.9348 - val_loss: 0.2224 - val_accuracy: 0.9146
Epoch 123/125
2350/2350 [==============================] - 18s 7ms/step - loss: 0.1670 - accuracy: 0.9346 - val_loss: 0.2188 - val_accuracy: 0.9160
Epoch 124/125
2350/2350 [==============================] - 17s 7ms/step - loss: 0.1664 - accuracy: 0.9348 - val_loss: 0.2230 - val_accuracy: 0.9143
Epoch 125/125
2350/2350 [==============================] - 17s 7ms/step - loss: 0.1665 - accuracy: 0.9346 - val_loss: 0.2227 - val_accuracy: 0.9162 accuracy: 91.62%

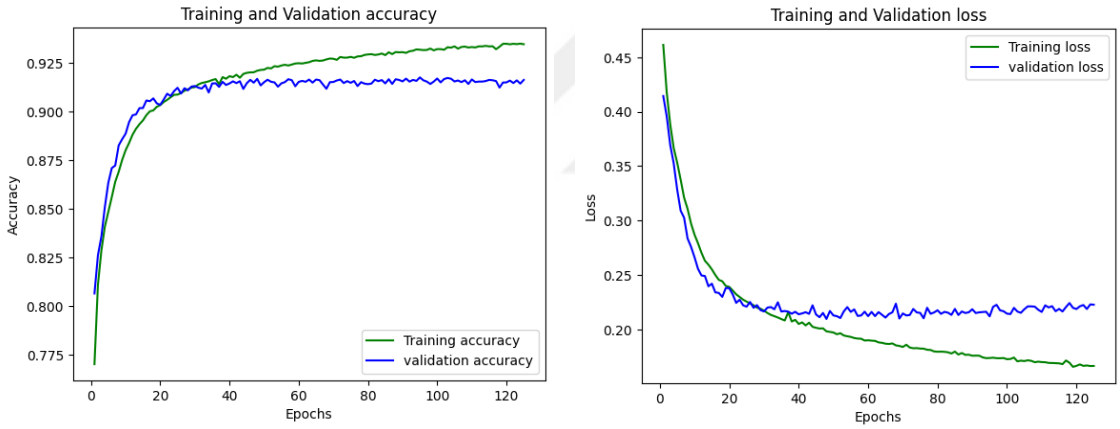


Figure 3. 16 Changes In Accuracy And Loss During Training.

This success is below the success achieved in recent studies on phishing detection. Assuming that a success of over 95% is achieved using Phistank, it can be concluded that deep learning has not demonstrated the necessary success. Despite the differences in training and test data, the lower success compared to other studies has prompted us to search for alternatives. It has been anticipated that higher success can be achieved by adding other features of the website along with the model obtained, considering not only the domain name but also different features of the website. From this perspective, it has been suggested that features such as HTTPS

support, start and end dates of the domain, start and end dates of the secure connection certificate, secure certificate provider, and domain provider, in addition to the name of the web page, could be effective in deciding whether it is a phishing site or not. Therefore, it has been decided to combine and use both the domain name and the other mentioned features in a machine learning model. The deep learning model trained in Phase 1 has been saved for use in Phase 2 to generate the phishing score feature as in example Table 3.2.

Table 3. 2 The scores generated by the model obtained in the first stage for some test URLs.

Url	DlPhishingScore
clouddefsystems	0.195639
everestmadencilik	0.003258
novapsikoloji	0.005055
garantieytliler	0.720906
hillspet	0.082515
islembilgiacikdnzgrs	0.999967
karaka	0.049356
acikdeniz	0.837007
odentihizmetleriimiz	0.999931
mertdizel	0.005977

3.5.1.2 The Combination of Other Features of the Web Page with the Pre-trained Model in the Second Stage

In today's world, many web browsers mark websites without secure connection support as phishing sites. Therefore, the necessity for web pages to support a secure connection has become a standard. When we randomly selected and examined some of the phishing sites published on USOM, we noticed that certain characteristics of SSL certificates providing secure connections were common. Certificates are usually created to expire within 90 days, and we observed that their validity had just started a few days ago. Additionally, most certificates are created by the same certificate provider. Furthermore, when querying the domain, we observed that the domains were created within the last few days, and their validity period is generally one year. It was also noted that the organizations providing the domain are mostly the same.

In the second phase, considering the idea that the phishing score provided by the pre-trained model for the web page and all the mentioned features can be combined to create a new learning model, a feature table like the one below could be created. This table could then be used as input to train a machine learning algorithm, and the results could be compared. The newly created dataset is named Dataset2, and care has been taken to ensure that the records in Dataset1 and Dataset2 are not repetitive and that the two datasets are different from each other. When creating Dataset2, phishing sites detected from March 2023 to September 2023 were selected, along with legal sites with .tr extension domains, excluding legal sites from Dataset1.

Table 3. 3 Example dataset to be used for the second stage of training.

URL	dlPhishingScore	domianStart	domainEnd	sslStart	sslEnd	sslIssuer	sslIssuerNum
clouddefsystems	0.195639	150	215	-1	-1	NaN	7
everestmadencilik	0.003258	4701	45	21	69	Let's Encrypt	1
novapsikoloji	0.005055	612	482	13	77	Let's Encrypt	1
garantieytliler	0.720906	12	353	-1	-1	NaN	7
hillspet	0.082515	3642	374	42	48	Let's Encrypt	1
islembilgiacikdnzgrs	0.999967	0	365	-1	-1	NaN	7
karaka	0.049356	815	973	20	70	Let's Encrypt	1
acikdeniz	0.837007	0	365	-1	-1	NaN	7
odentihizmetleriimiz	0.999931	520	209	-1	-1	NaN	7
mertdizel	0.005977	2504	50	61	29	Let's Encrypt	1

3.5.1.3 Feature Normalization

The features used during training need to be numerical. Here, the only non-numerical feature is '**sslIssuer**,' which represents the authority that issues the ssl certificate as a string value. In the training dataset, all ssl providers are made unique and assigned a number for each provider. If a website does not have ssl support, this feature is named NaN. The numerical value for ssl providers of the sites labeled as NaN is seven. Since there are a total of 25 different ssl providers, they are numbered from 0 to 24. If a different ssl provider appears during testing, it

will be numbered as 25. All other features are numerical, so after this process, normalization will be applied to bring all values between 0 and 1. The **MinMaxScaler** module from the sklearn library is used for this purpose.

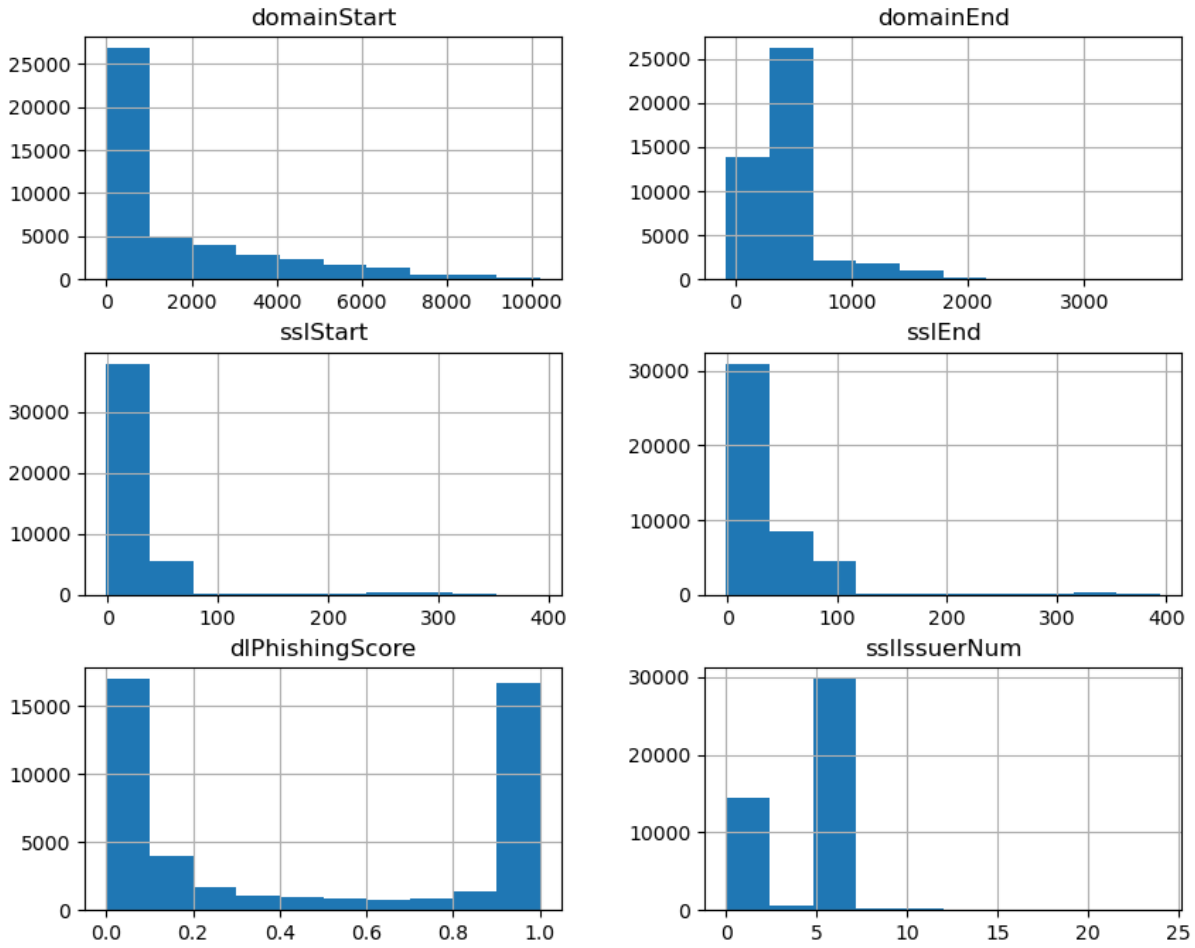


Figure 3. 17 Histogram Graphs For Each Property Of The Training Dataset

Table 3. 4 Number of Top 10 unique values in each feature

Domian Start Feature Top 10 Unique Values:	Domian End Feature Top 10 Unique Values:	Ssl Start Feature Top 10 Unique Values:	Ssl End Feature Top 10 Unique Values:	Ssl Issuer Feature Top 10 Unique Values:
0 7537	365 7614	-1 28041	-1 28041	7 28041
1 3781	364 3423	1 1017	89 998	1 11740
2 1973	363 1934	2 418	88 420	0 2079
3 1118	362 1377	59 358	31 355	6 1607
4 744	361 871	3 335	87 341	2 618
5 501	360 583	58 305	32 304	4 475
6 431	359 456	32 300	58 300	11 165
7 345	358 403	11 294	79 300	9 146
8 280	357 344	57 285	77 284	3 135
9 236	356 275	13 274	33 382	8 65

Table 3. 5 Machine Learning Techniques Accuracy Predictions

	Accuracy	Precision	Recall	F1
Logistic Regression	0.964558	0.978935	0.957030	0.967858
Support Vector Machine	0.921387	0.958882	0.902918	0.930059
Decision Trees	0.977807	0.981163	0.978191	0.979674
Random Forest	0.98.4432	0.984606	0.986805	0.985704

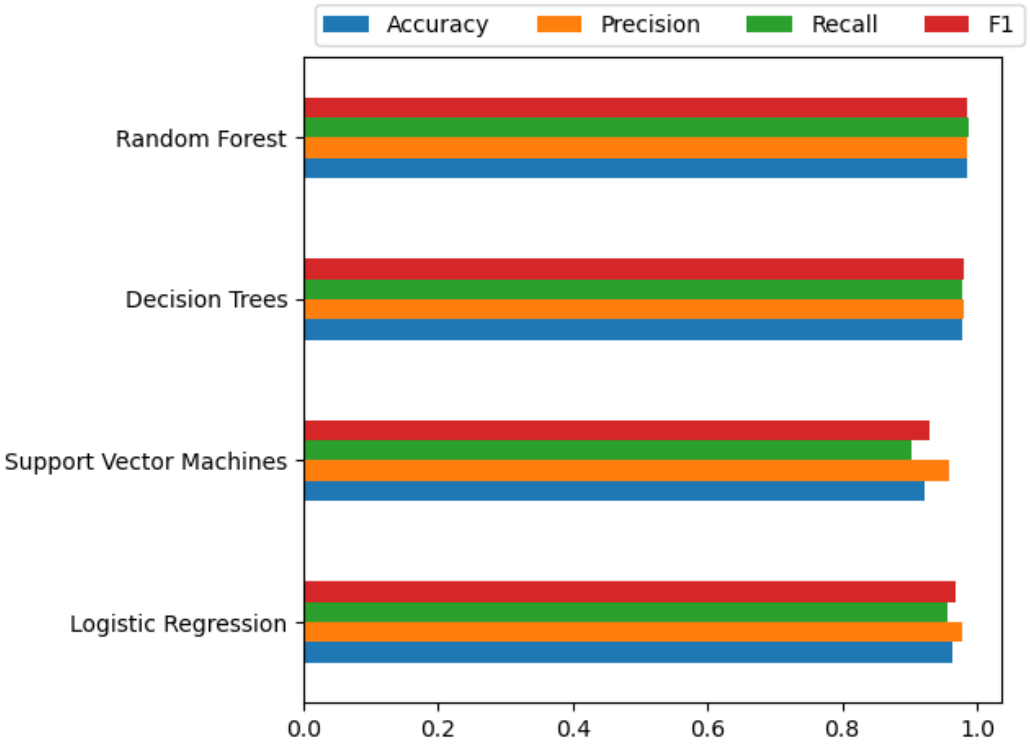


Figure 3. 18 Machine learning techniques for accuracy predictions

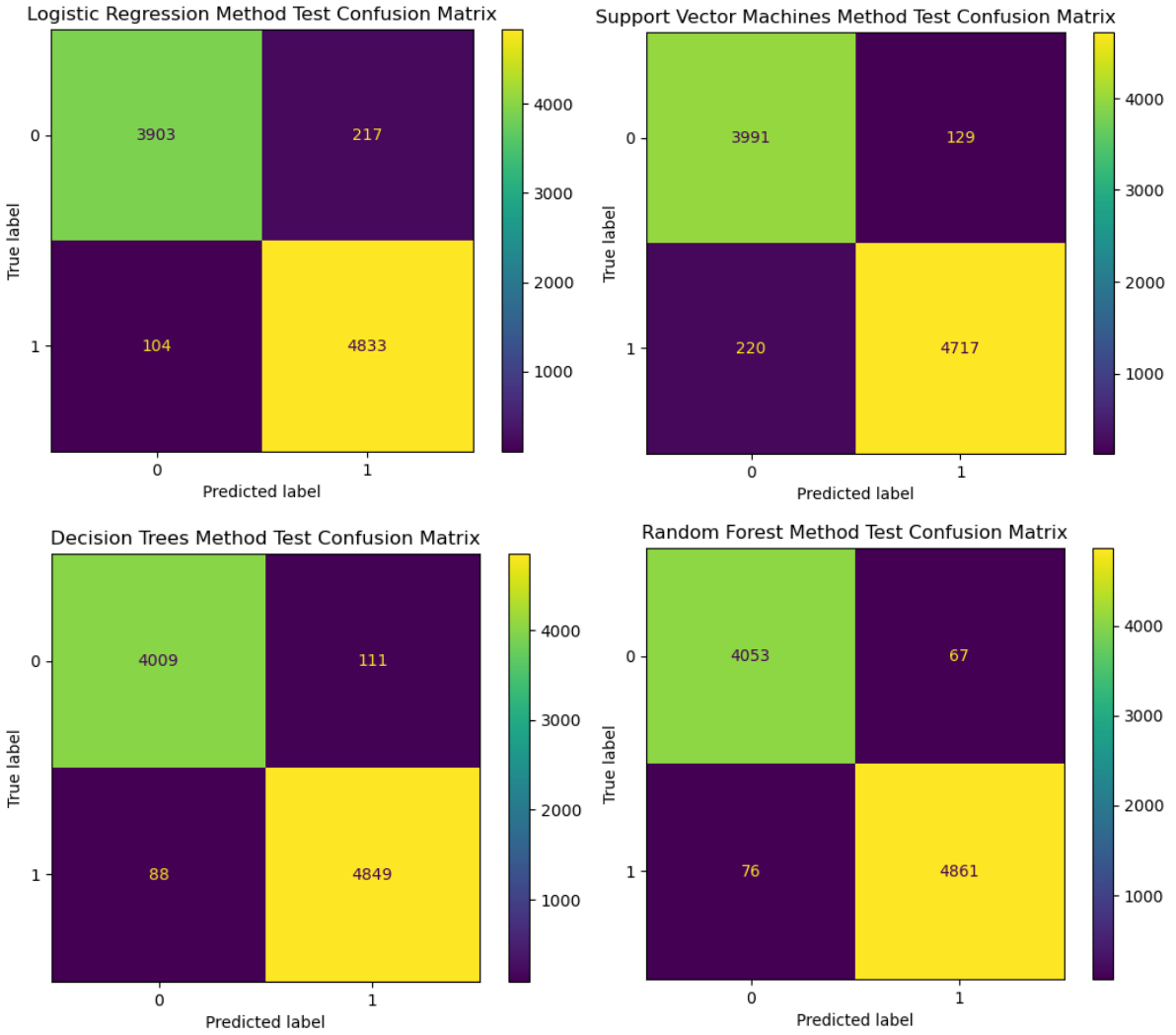


Figure 3. 19 Graphic representation of the confusion matrix

4. RESULTS AND DISCUSSION

When the results are initially examined, the success achieved, especially through the random forest method, is actually due to the input data being somewhat categorical. Particularly, the success close to 92% achieved in the first stage has revealed the necessity for us to discover additional determinant features. Since phishing sites inherently start phishing as soon as the domain registration is made, it was thought that the time elapsed since this domain registration could be a determinant, and for this reason, it was selected as one of the features. Indeed, when the dataset is examined, it is observed that the beginning of phishing sites' domains is generally 0, 1, 2, ..10, indicating that not much time has passed since the domain was registered. On the

other hand, in the training data, this situation is completely random, with values like 50, 100, 25, 250 being random. For this reason, the time elapsed between the start of the domain registration and the day the query is made is one of the determinant features. This is also one of the determinant parameters in the formation of decision tree decision trees. On the other hand, the end date of the domain is another determinant feature that stands out. Generally, the end date of phishing sites is seen to be at most one year. In addition to this, whether or not there is SSL support actually provides serious clues about whether the site is a phishing site. Also, SSL start and end dates again allow us to have an idea about phishing. If there is no SSL support, the SSL provider parameter is considered as NaN, while SSL start and end values are given as -1. SSL providers are generally 'Let's Encrypt' on phishing sites, while on legal sites, there are completely different providers.

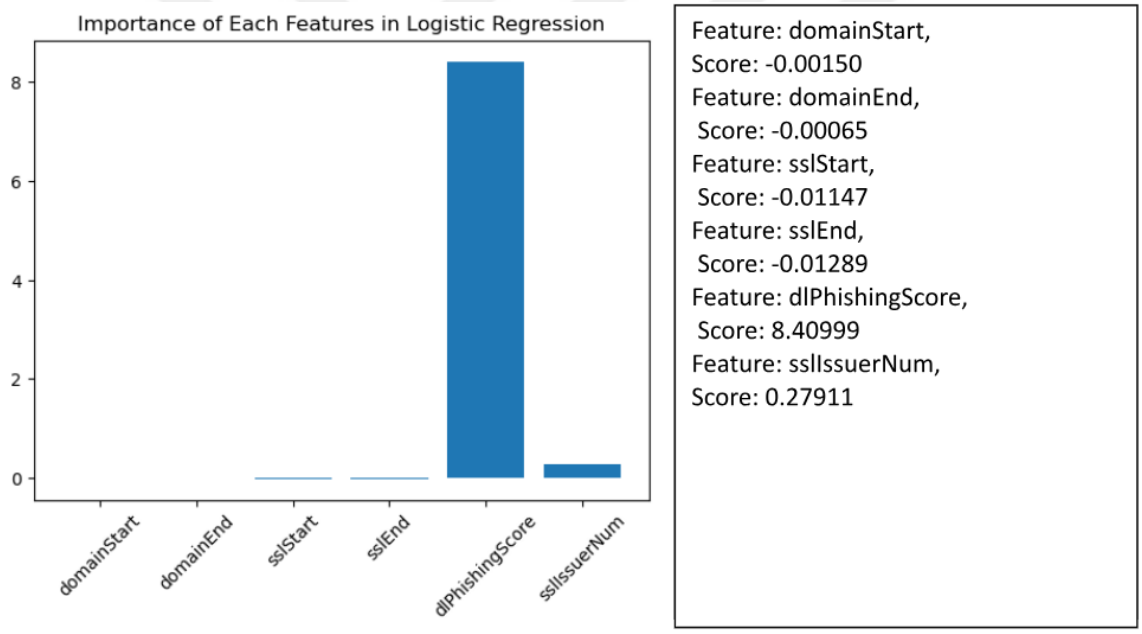
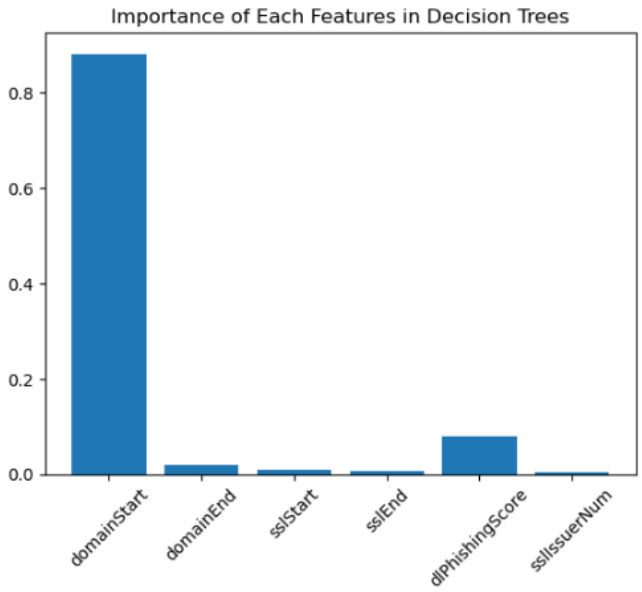
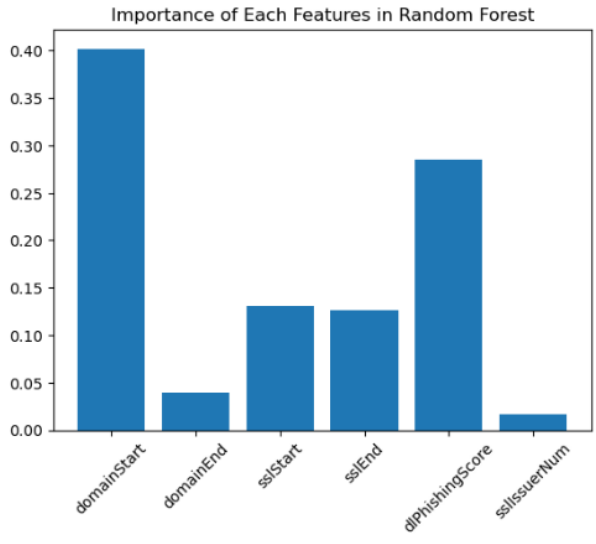


Figure 4. 1 List Of Graphs And Values Showing The Importance Of Each Feature In The Logistic Regression Model



Feature: domainStart,
Score: 0.88193
Feature: domainEnd,
Score: 0.01915
Feature: sslStart,
Score: 0.00924
Feature: sslEnd,
Score: 0.00505
Feature: dlPhishingScore,
Score: 0.08019
Feature: sslIssuerNum,
Score: 0.00444

Figure 4. 2 Graph And List Of Values Showing The Importance Of Each Feature In The Decision Trees Model



Feature: domainStart,
Score: 0.40199
Feature: domainEnd,
Score: 0.03997
Feature: sslStart,
Score: 0.13086
Feature: sslEnd,
Score: 0.12659
Feature: dlPhishingScore,
Score: 0.28441
Feature: sslIssuerNum,
Score: 0.01619

Figure 4. 3 Graph and List Of Values Showing The Importance Of Each Feature In The Random Model

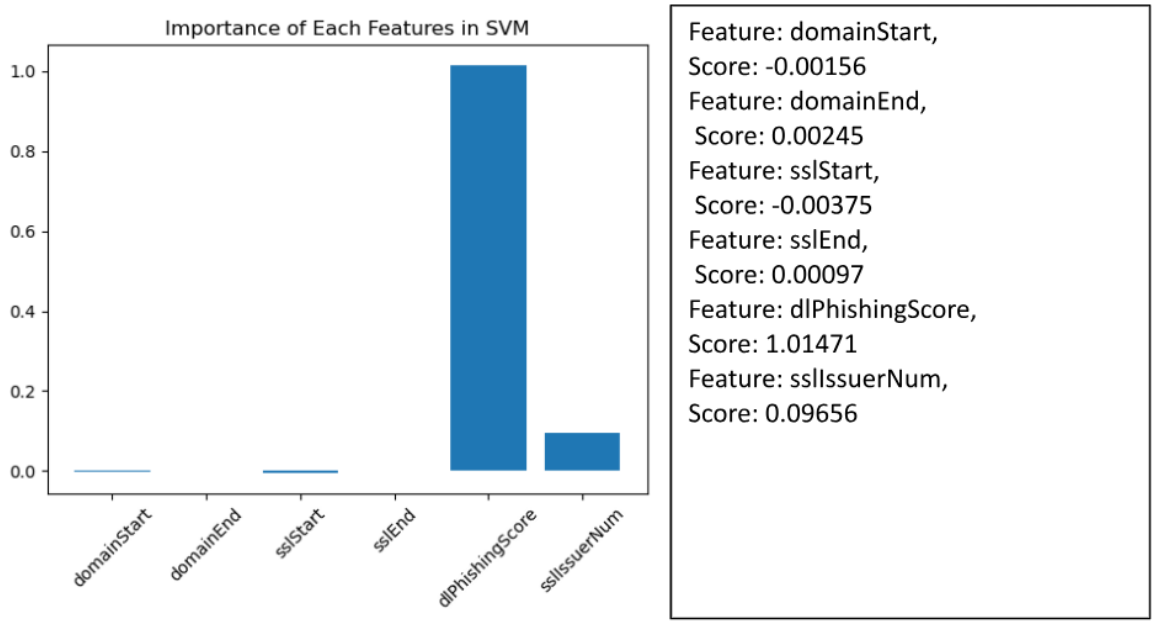


Figure 4. 4 Graph And List Of Values Showing The Importance Of Each Feature In The SVM Model

5. CONCLUSIONS

The first stage of our study involves encoding the characters of website names one by one and using deep learning to score whether a website is a phishing site based on the order of these encoded characters. In this deep learning phase, the domain name and SSL-related features of the relevant site are collected. The scores obtained in the first stage are combined with these features, and the data is input to a machine learning classifier for training. At the end of this training, our model achieves a success rate of 98.4% in predicting whether a website is a phishing site.

We believe that our approach and method in this study will contribute to the literature. Phishing detection solutions that rely on deep learning-based character encoding and learn the order of characters for prediction have often used shared phishing and secure sites from datasets such as PhisTank and Alexa as training datasets. When looking at these datasets, especially considering that PhisTank was created in 2015, it is anticipated that a model learning website name mentioned here will not be very successful against changing nomenclatures and new attacks developed based on them. Therefore, models that continuously update themselves by feeding

from a reliable source are thought to be more successful. For this reason, USOM is a suitable source for creating a training dataset, as it is both reliable and continuously updated. Furthermore, data on USOM is verified and reliable, making this phishing list, created through different stages from different sources and verified at each stage, a reliable source for cybersecurity researchers.

After the first stage, the problem at hand is transformed into a multivariate binary classification problem. While the score produced by the deep learning model in phase 1 can predict the site with about 92% success for the domain name variable, it is assumed that this rate is insufficient. Consequently, additional factors influencing the decision are revealed by adding queryable features specific to the website, and the success rate reaches a high level of 98.4%. It is assumed that a model suitable for phishing classification has been produced. The fact that the study is completely independent of the website content, dynamically produces a score for the website name, and supports it with queryable features is crucial, especially in terms of speed. On the other hand, it is essential to periodically update the model using the USOM dataset for effective use of the model. Additionally, it is important to investigate whether the two-phase approach is feasible for different prediction problems and to experiment with identifying relevant problems, accordingly, shedding light on future studies.

REFERENCES

[1] B. Gyawali, T. Solorio, M. Montes-y-Gómez, B. Wardman, and G. Warner, "Evaluating a semisupervised approach to phishing url identification in a realistic scenario," in *Proceedings of the 8th Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference*, Perth Australia: ACM, Sep. 2011, pp. 176–183. doi: 10.1145/2030376.2030397.

[2] K. Parsons, A. McCormac, M. Pattinson, M. Butavicius, and C. Jerram, "Phishing for the Truth: A Scenario-Based Experiment of Users' Behavioural Response to Emails," in *Security and Privacy Protection in Information Processing Systems*, vol. 405, L. J. Janczewski, H. B. Wolfe, and S. Shenoi, Eds., in IFIP Advances in Information and Communication Technology, vol. 405. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 366–378. doi: 10.1007/978-3-642-39218-4_27.

[3] M. R. Pattinson, C. Jerram, K. Parsons, A. McCormac, and M. A. Butavicius, "Managing Phishing Emails: A Scenario-Based Experiment.," in *HAISA*, 2011, pp. 74–85.

[4] F. Tchakounte, V. S. Nyassi, D. E. H. Danga, K. P. Udagepola, and M. Atemkeng, "A game theoretical model for anticipating email spear-phishing strategies," *EAI Endorsed Trans. Scalable Inf. Syst.*, vol. 8, no. 30, pp. e5–e5, 2021.

[5] M. Bossetta, "The weaponization of social media: Spear phishing and cyberattacks on democracy," *J. Int. Aff.*, vol. 71, no. 1.5, pp. 97–106, 2018.

[6] W. Lei, S. Hu, and C. Hsu, "Uncovering the role of optimism bias in social media phishing: an empirical study on TikTok," *Behav. Inf. Technol.*, pp. 1–15, Jul. 2023, doi: 10.1080/0144929X.2023.2230305.

[7] S. Alam and K. El-Khatib, "Phishing Susceptibility Detection through Social Media Analytics," in *Proceedings of the 9th International Conference on Security of Information and Networks*, Newark NJ USA: ACM, Jul. 2016, pp. 61–64. doi: 10.1145/2947626.2947637.

[8] A. Alharbi, A. Alotaibi, L. Alghofaili, M. Alsalamah, N. Alwasil, and S. Elkhediri, "Security in social-media: Awareness of Phishing attacks techniques and countermeasures," in *2022 2nd International Conference on Computing and Information Technology (ICCIT)*, IEEE, 2022, pp. 10–16.

[9] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[10] S. Abdelnabi, K. Krombholz, and M. Fritz, "VisualPhishNet: Zero-Day Phishing Website Detection by Visual Similarity," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, Virtual Event USA: ACM, Oct. 2020, pp. 1681–1698. doi: 10.1145/3372297.3417233.

[11] A. Niakanlahiji, M. M. Pritom, B.-T. Chu, and E. Al-Shaer, “Predicting Zero-day Malicious IP Addresses,” in *Proceedings of the 2017 Workshop on Automated Decision Making for Active Cyber Defense*, Dallas Texas USA: ACM, Nov. 2017, pp. 1–6. doi: 10.1145/3140368.3140369.

[12] K. S. Adewole, A. G. Akintola, S. A. Salihu, N. Faruk, and R. G. Jimoh, “Hybrid Rule-Based Model for Phishing URLs Detection,” in *Emerging Technologies in Computing*, vol. 285, M. H. Miraz, P. S. Excell, A. Ware, S. Soomro, and M. Ali, Eds., in Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 285. , Cham: Springer International Publishing, 2019, pp. 119–135. doi: 10.1007/978-3-030-23943-5_9.

[13] R. M. Mohammad, F. Thabtah, and L. McCluskey, “Intelligent rule-based phishing websites classification,” *IET Inf. Secur.*, vol. 8, no. 3, pp. 153–160, May 2014, doi: 10.1049/iet-ifs.2013.0202.

[14] F. Thabtah and F. Kamalov, “Phishing Detection: A Case Analysis on Classifiers with Rules Using Machine Learning,” *J. Inf. Knowl. Manag.*, vol. 16, no. 04, p. 1750034, Dec. 2017, doi: 10.1142/S0219649217500344.

[15] U. Ozker and O. K. Sahingoz, “Content based phishing detection with machine learning,” in *2020 International Conference on Electrical Engineering (ICEE)*, IEEE, 2020, pp. 1–6.

[16] B. Wardman, T. Stallings, G. Warner, and A. Skjellum, “High-performance content-based phishing attack detection,” in *2011 eCrime Researchers Summit*, IEEE, 2011, pp. 1–9.

[17] Y. Zhang, J. I. Hong, and L. F. Cranor, “Cantina: a content-based approach to detecting phishing web sites,” in *Proceedings of the 16th international conference on World Wide Web*, Banff Alberta Canada: ACM, May 2007, pp. 639–648. doi: 10.1145/1242572.1242659.

[18] C. Ardi and J. Heidemann, “Auntietuna: Personalized content-based phishing detection,” in *NDSS Usable Security Workshop (USEC)*, 2016. Accessed: Jan. 10, 2024. [Online]. Available: https://www.ndss-symposium.org/wp-content/uploads/2017/09/auntietuna-personalized-content-based-phishing-detection.pdf

[19] E. Medvet, E. Kirda, and C. Kruegel, “Visual-similarity-based phishing detection,” in *Proceedings of the 4th international conference on Security and privacy in communication netowrks*, Istanbul Turkey: ACM, Sep. 2008, pp. 1–6. doi: 10.1145/1460877.1460905.

[20] A. K. Jain and B. B. Gupta, “Phishing detection: analysis of visual similarity based approaches,” *Secur. Commun. Netw.*, vol. 2017, 2017, Accessed: Jan. 10, 2024. [Online]. Available: https://www.hindawi.com/journals/scn/2017/5421046/abs/

[21] C.-Y. Wu, C.-C. Kuo, and C.-S. Yang, “A phishing detection system based on machine learning,” in *2019 International Conference on Intelligent Computing and its Emerging Applications (ICEA)*, IEEE, 2019, pp. 28–32.

[22] A. Aljofey, Q. Jiang, Q. Qu, M. Huang, and J.-P. Niyigena, “An effective phishing detection model based on character level convolutional neural network from URL,” *Electronics*, vol. 9, no. 9, p. 1514, 2020.

[23] A. A. Orunsolu, M. A. Alaran, A. A. Adebayo, S. O. Kareem, and A. Oke, “A Lightweight Anti-Phishing Technique for Mobile Phone.,” *Acta Inform. Pragensia*, vol. 6, no. 2, pp. 114–123, 2017.

[24] K. S. Jishnu and B. Arthi, “Enhanced Phishing URL Detection Using Leveraging BERT with Additional URL Feature Extraction,” in *2023 5th International Conference on Inventive Research in Computing Applications (ICIRCA)*, IEEE, 2023, pp. 1745–1750.