UTILIZATION OF LOCAL AND GLOBAL IMAGE DESCRIPTORS FOR PHISHING WEB PAGE IDENTIFICATION

KİMLİK AVCISI WEB SAYFALARININ YEREL VE GENEL İMGE BETİMLEYİCİLERİ YARDIMI İLE TESPİTİ

ESRA EROĞLU

Asst. Prof. Dr. MURAT AYDOS
Supervisor

Submitted to
Graduate School of Science and Engineering of Hacettepe University
as a Partial Fulfillment to the Requirements
for the Award of the Degree of Master of Science
in Computer Engineering

2020

To my family who have supported me all my life

and *made me smile even in difficult times…*

ABSTRACT

UTILIZATION OF LOCAL AND GLOBAL IMAGE DESCRIPTORS FOR PHISHING WEB PAGE IDENTIFICATION

Esra EROĞLU

Master of Science, Computer Engineering Department

Supervisor: Asst. Prof. Dr. Murat AYDOS

January 2020, 85 pages

In recent years, the use of the Internet has increased in all areas of life, thus many cyber-attacks have emerged. These attacks aim to steal users' private information such as passwords, credit cards. During phishing attacks, attackers have an attitude of deceiving users by creating copies of a web page that is known and frequently used by users. In this thesis, a new approach which can be a solution for detecting phishing attacks on web pages has been introduced. In the proposed approach, experiments have been conducted with local and global descriptors that have not been used before in the literature. In addition, "holistic" and "multi-level patch" approach was used to increase detection of attacks.

The "holistic" approach referred to in these approaches is to process the image as a whole, while the "multi-level patch” approach is to separate the image into equal dimensions. The data set used in the evaluation phase of the proposed approach includes screenshots taken from websites of 14 different trademarks in total. This data set, with a total of 2852 samples, is "open set". The features obtained from the descriptors were then classified by support vector machine, random forest and XGBoost machine learning algorithms. According to the extensive test results, the best success rate is 90.38% with SIFT descriptor. This thesis suggests that the proposed approach may be effective in detecting possible counterfeiting attacks on web pages.

Keywords: Phishing, Computer Vision, Machine Learning, Local Descriptor, Global Descriptor.

ÖZET

KİMLİK AVCISI WEB SAYFALARININ YEREL VE GENEL İMGE BETİMLEYİCİLERİ YARDIMI İLE TESPİTİ

Esra EROĞLU

Yüksek Lisans, Bilgisayar Mühendisliği

Tez Danışmanı: Dr. Öğr. Üyesi Murat AYDOS

Ocak 2020, 85 sayfa

Son yıllarda, yaşamın her alanında internet kullanımı artmaktadır. Bu sebeple birçok siber saldırı ortaya çıkmaktadır. Bu saldırılardan kimlik avı saldırıları, kullanıcıların şifre, kredi kartı gibi özel bilgilerini çalmayı amaçlamaktadır. Kimlik avı saldırılarının genelinde saldırganların kullanıcılar tarafından bilinen ve sıklıkla kullanılan bir web sayfasının kopyasını oluşturarak kullanıcıları kandırma tutumu vardır. Bu tez çalışmasında web sayfalarında kimlik avı saldırılarının tespit edilmesine çözüm olabilecek bir yaklaşım getirilmiştir. Önerilen yaklaşımda literatürde kimlik avı saldırılarında daha önceden kullanılmamış yerel ve küresel tanımlayıcılarla deney gerçekleştirilmiştir.

Ayrıca, saldırıların tespit edilmesini artırmak için "bütünsel" ve "çok seviyeli parçalama" yaklaşımından yararlanılmıştır. Bu yaklaşımlarda bahsedilen "bütünsel" yaklaşım, görüntüyü bir bütün halinde işlemekteyken "çok seviyeli parçalama" yaklaşımı görüntüyü eşit boyutlara ayırma durumudur. Önerilen yaklaşımın değerlendirme aşamasında kullanılan veri seti toplamda 14 farklı ticari markanın web sitelerinden alınmış ekran görüntülerini içermektedir. Toplamda 2852 örneğin olduğu bu veri seti "açık küme" özelliğini taşımaktadır. Tanımlayıcılardan elde edilen özellikler daha sonrasında destek vektör makinesi, rastgele orman ve XGBoost makine öğrenme algoritmaları tarafından sınıflandırılmıştır. Kapsamlı olarak yapılan deney sonuçlarına göre en iyi başarı oranı SIFT tanımlayıcısı ile %90.38 olarak elde edilmiştir. Bu tezde önerilen yaklaşımın web sayfalarında olabilecek kimlik avı saldırılarını tespit etmede etkili olabileceğini göstermektedir.

Anahtar Kelimeler: Kimlik avı saldırıları, Bilgisayarlı Görü, Makine Öğrenmesi, Yerel Tanımlayıcılar, Genel Tanımlayıcılar

ACKNOWLEDGEMENTS

First and foremost, I would like to wholeheartedly thank to my supervisor Asst. Prof. Dr. Murat Aydos for his valuable advice, endless patience and guidance. At every stage of this thesis, he supported me with his knowledge, experience, motivation and encouragement.

In addition, I would like to express my gratitude to Dr. Ahmet Selman Bozkır, who shared his valuable knowledge with me throughout my dissertation and provided me with continuous assistance in terms of resources and methods.

Besides I would like to thank my thesis committee members for insightful comments for this thesis.

Finally, I would like to thank my mother Emel Eroğlu, my father Alpaslan Eroğlu and my brother Yunus Emre Eroğlu who never left me alone with their material and moral support during my studies.

Esra EROĞLU

January 2020, Ankara

CONTENTS

ABSTRACT i

ACKNOWLEDGEMENTS v

CONTENTS vi

FIGURES ix

TABLES x

ABBREVIATIONS xi

1. INTRODUCTION 1

1.1. Overview 1

1.2. Motivation 4

1.3. Aim of the Thesis 5

1.4. Thesis Structure 5

2. BACKGROUND 7

2.1. Phishing 7

2.2. Computer Vision Background 10

2.3. A Brief Overview of Machine Learning Background 13

2.3.1. Supervised learning 14

2.3.2. Unsupervised learning 14

2.3.3. Semi-supervised learning 15

2.3.4. Reinforcement learning 16

3. RELATED WORK 18

3.1. List-based approaches 18

3.2. Machine learning based approaches 20

3.3. Heuristic based approaches 23

3.4. Vision based approaches 23

4. METHODS AND TOOLS 28

4.1. Visual Descriptors 28
4.1.1. Global Descriptors 29
4.1.1.1. GIST 29
4.1.1.2. LBP 31
4.1.1.3. HOG 32
4.1.2. Local Descriptors 33
4.1.2.1. SIFT 33
4.1.2.2. DAISY 35
4.2. Image Representation 36
4.3. Machine Learning Methods 39
4.3.1. Support Vector Machines (SVM) 39
4.3.1.1. Linear SVM 40
4.3.1.2. Nonlinear SVM 42
4.3.2. Random Forest 42
4.3.3. XGBoost 44
4.4. Evaluation Criteria 46
4.5. Tools 48
4.5.1. OpenCV 48
4.5.2. Python 48
4.5.3. Pyleargist 49
4.5.4. Sklearn 49
5. APPROACH 51
5.1. Data Phase 51
5.2. Feature Extraction and Image Representation Phase 51
5.3. Machine Learning Phase 53
5.4. Validation Phase 54
6. EXPERIMENTS AND RESULTS 55
6.1. Dataset 55
6.2. Experiments 58
6.2.1. Global Descriptor Based Analysis 58
6.2.2.1 Spatial Multi-Level-Patch Based Analysis 60
6.2.2.2. Comparative Study- HOG Based Analysis 63

6.2.2. Local Descriptor Based Analysis 65

7. DISCUSSION 69

8. CONCLUSION 71

REFERENCES 73

APPENDIX 82

APPENDIX 1 – Application Programming Interface (API) 82

CURRICULUM VITAE 84

FIGURES

Figure 2.1. Phishing Activity Trends Report 3rd Quarter 2019 [16] 9

Figure 2.2. Most-Targeted Industry Sectors 3nd Quarter 2019 [16] 9

Figure 2.3. Phishing Activity Trends Report [16] 10

Figure 2.4. Machine Learning Methods. Adopted from [25] 13

Figure 2.5. Supervised learning 14

Figure 2.6. Unsupervised learning 15

Figure 2.6. Semi-supervised learning 16

Figure 2.7. Reinforcement learning 17

Figure 3.1. Taxonomy of anti phishing solutions [39] 18

Figure 4.1. Example of the LBP algorithm 32

Figure 4.2. Using HOG in an image 33

Figure 4.3. SIFT algorithm flowchart 34

Figure 4.4. SIFT keypoint detection [14] 34

Figure 4.5. SIFT Keypoint and orientation (Adopted from [84]) 35

Figure 4.6. DAISY Keypoint and orientation [85] 36

Figure 4.7. The Bag of Visual Words Model 37

Figure 4.8. Spatial Pyramid Matching Model 39

Figure 4.9. Margin types 40

Figure 4.10. Classification of non-linear samples [23] 42

Figure 4.11. Generation tree using Random Forest 43

Figure 4.12. Steps of Boosting Method 45

Figure 5.1. Model architecture 52

TABLES

Table 4.1. Confusion matrix..47

Table 6.1. Phish-IRIS Dataset..57

Table 6.2. Holistic Results of GIST and LBP..59

Table 6.3. Voting Classifier Results of GIST and LBP 59

Table 6.4. Multi-Level-Patch Pyramid GIST based analysis..61

Table 6.5. Multi-Level-Patch Pyramid LBP based analysis 62

Table 6.6. Combined Results of GIST and LBP based analysis.....................................62

Table 6.7. Results of voting classifier analysis 63

Table 6.8. Prediction results with HOG descriptors..64

Table 6.9. Results of SIFT based analysis 66

Table 6.10. Results of DAISY based analysis 67

Table 6.11. Results of voting classifier analysis 68

ABBREVIATIONS

API	Application Programming Interface
BOVW	Bag of Visual Words
BOW	Bag-of-Words
DDoS	Distributed Denial of Service
DoG	Difference of Gaussian
DOM	Document Object Model
DoS	Denial of Service
EMD	Earth Mover’s Distance
FPR	False Positive Rate
HOG	Histogram of Oriented Gradients
HTML	Hyper Text Markup Language
IP	Internet Protocol
ROC	Receiver Operating Characteristic
SIFT	Scale Invariant Feature Transform
SPM	Spatial Pyramid Match
SSL	Secure Socket Layer
SVM	Support Vector Machines
TPR	True Positive Rate

1. INTRODUCTION

1.1. Overview

Along with the development of technology, the spread of the Internet and information technology applications, many people have begun to be affected by them. It is an incontestable fact that developing technology has lots of harm as well as benefits in the internet area. Rapid changes in the development of technology affect life in many areas. Given that technology is widely used, these can not influence only individuals. They will have a greater impact on larger institutions, businesses and even governments. Therefore, it is crucial to develop protection against attacks [1]. Particularly, keeping personal information of users creates security problems in a virtual environment. Therefore, users may be exposed to different types of internet-based attacks. These attacks are called cyber-attacks.

By definition, cyber-attacks are a type of attack called by hackers to steal people's information or render organizations' systems inoperable. Attackers can create data such as seizing data, deleting data and making the system unusable through malicious software such as trojans and worms.

In the literature, cyber-attack types are described in a study by Bhuyan et al. [2]. The types of attacks carried out in the cyber-attack area are as follows:

- ***Sniffing:*** It is defined as listening to data traffic created by computers connected to a network. All packets coming to the network routers are handled with the method of storing confidential information such as passwords and email texts. Hackers try to check the packets on the network to obtain this kind of private information.
- ***Denial of Service (DoS):*** Basically, DoS aims to make the system unusable by placing a load above the capacity of the system. It disrupts the service of a system or destroy the function of the service. DDoS, which is called Distributed Denial of Service, means that the attack is initiated from a number of different sources rather than a single source. Additionally, this attack is done by the

attackers generated by the machines. Unlike the DoS, DDoS is more dangerous because many machines are used.

- ***IP Spoofing:*** Internet Protocol (IP) packets from a modified IP address are sent to the destination. Especially, the purpose of this attack is the process of showing the IP address differently to the destination system on a connection; on the other hand, this attack is to hide the identity of computers that are connected to the network through using a fake IP address. Since the source that carried out the attack is hidden on the computer, it is possible for a website to become unusable.
- ***Social engineering:*** Recently mentioned in many places, this attack is based on the lack of knowledge of people that evaluate the elements of human behavior as security gaps. Although individual awareness is important in social engineering, information of people with inadequate knowledge can be stolen in fake scenarios. Each person must inevitably use a number of information technologies that involve risks, and the key to protecting themselves is to be aware of the risk.
- ***Spyware:*** Rapidly increasing types of cyber-attacks, spyware collects user data by monitoring user activity. So they run in the background, they aim to extract some data from the system that is specified by the developer or user without their conscious. Also, they have a negative effect on computer performance. Normally, they can be installed by people with physical access to a system, or through malicious programs downloaded from the Internet. They are concealable in unlicensed and copy products. Therefore, it is a type of attack that has serious consequences for users.
- ***Virus:*** Viruses can be defined as small blocks of code attached to a program. They can copy themselves in the network or add themselves to other programs and thus easily propagate. Unlike other malware, viruses aim to make the computer or system unusable by infecting other files.
- ***Trojan horses:*** Trojan horses are generally malicious software that is reliable but has a destructive effect in the background. The computer programs developed for this attack try to access information by remotely managing the infected system.

- ***Worms:*** The worms multiply in excess and seize the system and its functions. So, it sends the information in the system to hackers. The structure of worms is similar to viruses, but they do not require human interaction as they do to spread. In terms of their spreading, excessive proliferation of worms is due to the realization that the system is running slowly, and this is the result of excessive use of system resources.
- ***Botnet:*** Botnet attacks are the cyber-attacks made by the computers of innocent users that were seized. These users often do not realize that the attack was from their computer. In particular, this attack can be made easier by using computers that do not have an anti-attack program such as a firewall.
- ***Keyloggers:*** This attack is carried out with the aim of communicating all the keyboard operations to the hackers. Commonly, users can record each key on the keyboard and easily get into the hands of others in banking transactions or in areas that are entered with a password.
- ***Phishing:*** Phishing is an attack that uses a lack of information of end-users in terms of web browser tips and security indicators, and uses similar-looking emails and websites of legitimate organizations to trick people into revealing such sensitive information. If a phishing site mimics a legitimate site, it can become a deceptive website compared to a legitimate site. So, the user can obtain important information.

In recent years, the increase in cyber-attacks and kinds of it has raised different concepts. One of them is cyberwar. Generally, wars between countries are being fought by using technology rather than traditional methods in these days. Many tools, used for war purposes in the past, are replaced by cyber-attacks. For this reason, countries ensure the security by taking measures against these attacks. In addition, they need cyber security centers. Many measures developed at these centers are becoming increasingly important. The reason for them is that the growth of the mass threatened by these attacks increases the possible cost loss at the highest rate. Furthermore, information and communication technologies are used by hackers for individual damage. In this way, attacking an individual, an institution or a state can be called cyberwar. Defense mechanisms against these attacks are being developed by IT experts [3].

In cyber-attacks, it is important to have a defense mechanism in advance. However, it is possible to carry out attacks on a predefined and specified date. Thus, continuous monitoring by IT experts should be performed to minimize the damage of attacks. However, existing cyber-attack technology may not always be enough. Therefore, different solution methods are needed [4]. One of these methods is the use of intelligent systems. The increase in the use of such method in almost every field is due to the developments in artificial intelligence in recent years. It is important for the defense systems that experts in the field follow these developments closely [5]. Also, machine learning methods which are very popular in artificial intelligence are used in this thesis. Nowadays, phishing attacks are a cyber-attack that can cause great damages both commercially and individually. In addition, with the development of the Internet, these attacks have become inevitable. Moreover, there is an ongoing race between the attackers and the defense mechanisms developed against it. Therefore, this problem is still unresolved. In this thesis, a solution will be developed to detect web pages that have been subjected to phishing attacks.

1.2. Motivation

The convenience provided by the cyber world increases day by day. Similarly, the scope and amount of transactions performed through information systems are increasing. In particular, the attackers who try to carry out cyber-attacks with different and more complex methods, are being harmed to institutions or individuals without notice. As a result of this situation, it becomes difficult to identify new methods of attack. Developed security methods are often insufficient to detect these attacks. Therefore, a lot of software has been developed to effectively protect and manage systems. Mostly, systems designed with artificial intelligence and machine learning techniques and having more than one function provide great benefits in information security. In this thesis, an approach to the recognition of phishing web pages using the global and local image identifier is proposed. In addition, experiments with spatial patch pyramid approach are supported. Different types of image descriptors are used for histogram conversion of visual properties.

In phishing attacks, especially financial losses are realized at the level of billion dollars annually. Furthermore, this attack has become an internet crime that grows much faster than other types of cyber-attacks [6]. However, the personal security of internet users against the different threats coming from the web is very weak. The importance of this study is to develop and explain a system that detects and prevents phishing attacks before reaching users.

1.3. Aim of the Thesis

Phishing attacks are generally used to access sensitive and confidential information such as user names, passwords, credit card information, and network credentials. These attacks are analyzed in four categories: list-based, machine learning based, heuristic based and vision based. So, computer vision based solution for detection of phishing attacks is presented in this thesis.

The solution for the problem of image classification among the phishing web pages is being developed. The objectives of this system are listed as follows:

First, in this thesis, phishing detection will be performed by using visual descriptors based on machine learning.

Second, the proposed method distinguishes visual descriptors in two ways: local and global descriptors, while extracting features from screenshots of web pages.

Third, the model will use a "pyramidal patches" approach for global descriptors to extract more and useful features.

Fourth, local descriptors will be experimented with different codebook numbers. In this way, the model will have the best performance.

Fifth, global and local descriptors will be evaluated with "Phish-IRIS" dataset. A comprehensive experimental study will be carried out for this purpose.

1.4. Thesis Structure

The following chapters are designed as follows:

In the second chapter, background information about the methods used in the thesis is given. This information is presented under separate headings. Some basic concepts in the thesis are mentioned. These concepts are examined under 3 main headings. The first section includes the phishing problem and the second section contains basic information about computer vision. In the third chapter, basic topics based on machine learning are mentioned.

In the third chapter, the studies in the literature are mentioned. The literature section examines the existing studies for the detection of phishing web pages.

The fourth chapter is the explanation of the methods that is the main contribution of the thesis. In this section, visual descriptors are described under global and local subheadings. GIST and LBP methods are global; The SIFT and DAISY methods are described in detail in the local descriptor. In the following chapter, image representation is expressed. Then, machine learning methods are explained. Finally, the tools used in the thesis and evaluation criteria are mentioned.

In the fifth chapter, information about the model developed for the solution of the problem is given. Moreover, the thesis is explained in detail with the methods and methods used by the aims and objectives of the thesis. The application architecture will be mentioned later. All the steps are explained in detail.

In the sixth chapter, experimental results related to the model developed are shown. For this, the dataset is explained first. Then, the experiments were examined under three subheadings.

In the seventh chapter, the results of the experiments are interpreted and compared.

In the eighth chapter, the general result is expressed and the results are discussed. This section is the last part of the thesis.

2. BACKGROUND

In this section, basic information about the methods required for the thesis will be given which consists of phishing attacks, computer vision and machine learning.

2.1. Phishing

Phishing is a type of cyber-attack based on deception. For this attack, innocent users are asked to share personal information such as passwords, user names and identification numbers [7]. This attack is characterized by the creation of web pages that visually mimic colleagues and are used to capture sensitive information. At the same time, it is an important factor for attackers to have exactly the same content and image as legitimate web pages. In addition, hackers using social engineering techniques identify targeted users. Then, these web pages are delivered to users and it is intended to steal personal information. The use of this information is the beginning of an illegal activity. Given the damages of phishing attacks, the e-commerce industry is affected enough to cost billions of dollars [8].

Social engineering and technical knowledge are often used in combination to conduct phishing attacks [9]. Therefore, the explanation of these two views helps to describe phishing attacks. From a technical point of view, the phishing attacker first accesses the HTML code of a legitimate website. Later, it transmits a fake website to the people it determines as a target with social engineering techniques. Phishing attacks using this approach are more effective because a web page containing phishing is prepared in a short time [10].

In terms of social engineering, users need to benefit from their weaknesses. So, users are sent messages that seem to come from real system administrators.

The phishing attack, which has been increasing rapidly from year to year, has been in existence since the invention of transactions on the Internet, such as internet banking. The Internet has recently become an important and necessary tool in many areas of life. The fact that the Internet is open to cyber-attacks, as well as the advantages of facilitating and accelerating daily life, reveals serious security gaps. The use of tangible elements in transactions such as online shopping, electronic banking, social networks, e-mail and electronic commerce attracts fraudsters. Due to the inattention of Internet users, technology literacy levels are low, or they are unaware of such attacks and are more exposed to these attacks [11, 12].

The general purpose of phishing attacks is financial fraud through imitation. However, there are many different types of this attack and they are usually classified according to who the target and the attacker are.

Phishing attacks are broadly classified into four categories such as Pharming, Spear Phishing, Clone Phishing and Whaling [13-15].

Pharming: An attacker would poison a DNS record and, in practice, redirect visitors to a legitimate website. It is a phishing technique in which an attacker impersonates a trusted entity to obtain sensitive information such as user names, passwords, and bank account numbers [13-15].

Spear Phishing: This phishing technique involves an email fraud attack against a person or an organization. In the ordinary phishing attack, the phished emails are sent to a random email id or account whereas in spear phishing, the emails would come from a known recipient. One of the instances where the spear phishing attack occurred and targeted the RSA security firm where the attackers sent phishing emails to 4 different workers at RSA’s parent company. Spear phishing usually targets a specific person or organization [13-15].

Clone Phishing: It simulates a legitimate email account by using an original email and changing links. In phishing cloning, an attacker uses a legitimate e-mail that has already been sent and copies its content to a similar e-mail with a link to a malicious site [13-15].

Whaling: In this technique, it targets famous people such as politicians, celebrities and executives. This is considered as the most serious form of phishing in which the content of the email includes customer compliant, executive issues, etc. Whaling is a kind of fishing that targets important and wealthy individuals such as CEOs or civil servants [3-15].

As the information and communication technologies became widespread in the world and the use of the internet increased, phishing attacks became a rapidly increasing attack. Moreover, the popularization of this is targeted attack. According to reports by the Anti-Phishing Working Group, the number was estimated at 182,465 in the second quarter of 2019. In the third quarter of 2019, the number of phishing attacks worldwide was determined as 266,387 as shown in Figure 2.1 [16]. In addition, the highest number of attacks in the last three years has occurred this year.

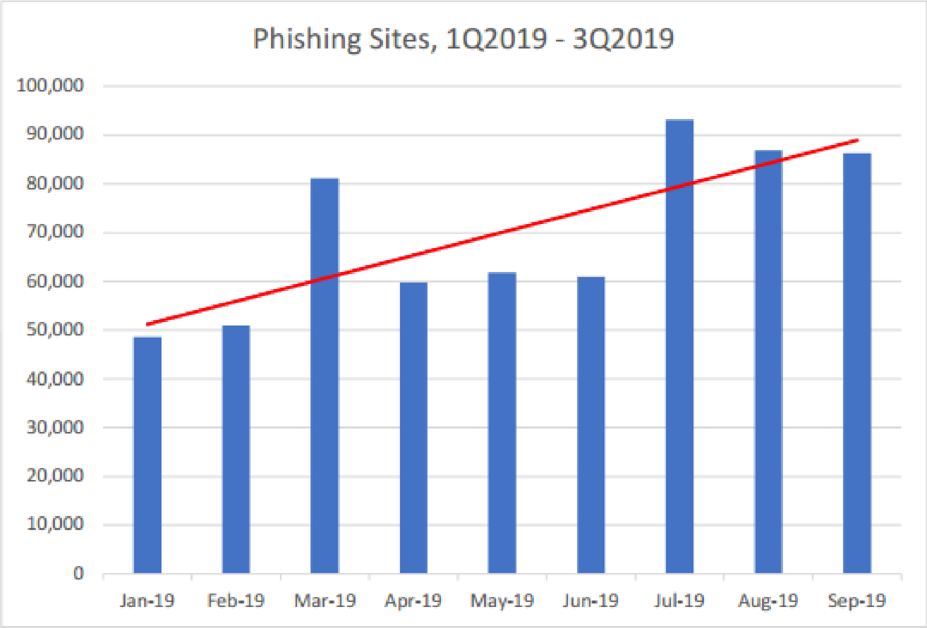


Figure 2.1. Phishing Activity Trends Report 3rd Quarter 2019 [16]

Regarding the phishing attacks by sectors, SaaS / Webmail has been most affected. According to Figure 2.2, there were fewer attacks on the Cloud Storage and eCommerce sectors.

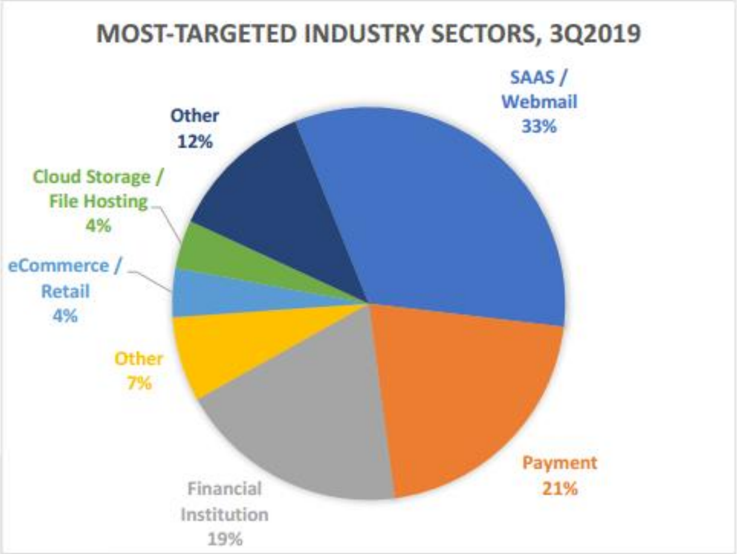


Figure 2.2. Most-Targeted Industry Sectors 3nd Quarter 2019 [16]

According to the report published in the first quarter of 2019, Brazil was the country with the highest share of attackers by 21.66%, followed by Australia (see Figure 2.3.). Also, the banking sector is ranked first in the number of attacks, the share of attacks on credit institutions increased by 5.23%. It increased to 25.78% compared to the fourth quarter of 2018 [16].

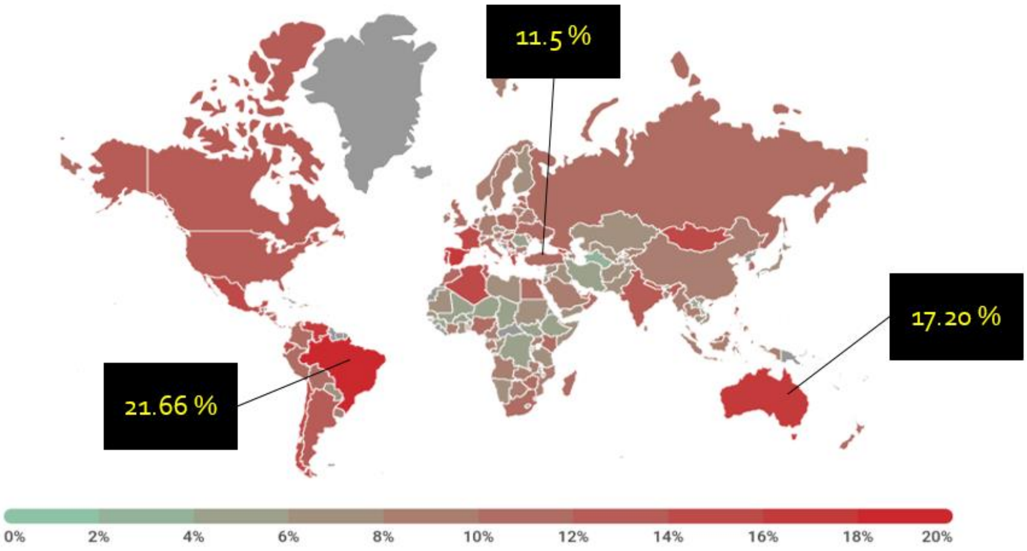


Figure 2.3. Phishing Activity Trends Report [16]

According to the widespread opinion around the world, phishing attacks are nowadays being described as an increasingly ongoing attack. It has continued to be popular, especially in the last two decades. Moreover, there is a constant competition between attackers and phishers, although various defense mechanisms are tried to prevent them. Therefore, phishing attacks are still unresolved.

2.2. Computer Vision Background

The methods developed for detecting and distinguishing objects in an image are called computer vision. Just like in the biological view, a visual element needs to be examined, modeled, and then interpreted. Therefore, some meaningful information should be extracted from the image and methods have been developed considering the characteristics of the images. All visual properties determined by these methods constitute the descriptive properties of isolated points, continuous curves, or connected regions of this image. First of all, concepts related to visual properties should be explained [20-24].

The first of these concepts is "feature". This is the first step of computer vision or image processing. It is defined as extracting the characteristic features of images. Features can be used in many areas such as point of interest detection, edge detection, corner detection and drop detection [17-21]. These fields will be described after "feature extraction".

The process of finding the characteristic features of an image in a vector is called "feature extraction" in computer vision or image processing. Feature extraction has algorithms developed to detect specific pixels in an image and obtain features from them, which are distinguished in terms of the computational complexity and the repeatability. It is to reduce the size of complex data into a simpler problem. These special algorithms find types of image features and the data structure that corresponds to multiple properties obtained as a result of this is called "feature vector" [17-21].

After explaining the feature extraction, other informative parts of the image must be defined. The first of these is edged. It is a set of points that separates one image from other images. Furthermore, it is one dimensional. Edge detection decides the boundaries of the image by finding points with a large gradient value. The second is interest point detection. This is selected taking into account the high gradient value when specifying edges. The other type of blobs is used to find points that the corner detector cannot detect. The final type of image is the ridge descriptor. With the gray-level image it finds the one-dimensional axis in the image. This is done for feature extraction in medical images [17-21].

Another notion used in this field is the histogram. A histogram is a kind of graph that a scatter plot gives the frequency of each number in an array. The histogram, a commonly used visualization method in statistics, is a slightly modified version of the column graph [22]. A column chart has a Y value for each column on the X-axis. In other words, while the X-axis in the column graph consists of discrete values, the X-axis in the histogram is continuous [23].

Image descriptors or visual descriptors are other notions that need to be explained. They produce visual features or descriptions of the contents in the image. Also, these create a visual feature vector that separates one image from another. In other words, image descriptors perform some operations on the image and extract properties of that image. Each feature is extracted by its unique methods in these operations [20].

Feature extraction is performed after feature detection in computer vision. The processing sequence proceeds in this way. In computer vision, image descriptors are classified in different ways in the literature. The first classification is global-local separation [34]. There are image descriptors grouped as local and global according to the features to be determined. First, global descriptors perform feature extraction by looking at an entire image [32]. Therefore, it makes inference according to the color, texture and shape properties of the image. Secondly, extracting the feature by taking into account a specific region instead of the entire image is called local descriptors. Local descriptors focus on image regions that can separate images, in other words, the interest points of the image [33].

In the second classification, image descriptors are examined under three groups as color, texture and shape [31]. Color is an important factor that can be used to extract the features of an image. Especially suitable for different image models can be obtained. The basis of these models is based on mathematical equations and formulas. There are different structures of color histogram, color moments, color coherence vectors and color correlogram [34]. Color histogram is a kind of histogram with GCL and LCH types [35]. It determines the color frequency according to the color distribution of an image. Color moments have 3 color moments: mean, standard deviation and skewness [36].

Color coherence vectors is a recommended method for classifying colors [37]. This focuses on the color value of each pixel. Classification is classified as coherent or incoherent. Color correlogram is recommended for image indexing [38].

Texture is a variant of global image descriptors [31]. This technique focuses on visual patterns that are homogeneous because the homogeneity of an image depends on its color and intensity. Therefore, the surface properties and their relationship with the surrounding areas play a role.

2.3. A Brief Overview of Machine Learning Background

Machine learning deals with the design and analysis of algorithms that extract information from given data. While common in almost every field, statistics are closely related to artificial intelligence and computer science. Machine learning methods are used in many areas of daily life such as object recognition, voice recognition and prediction [26].

The basis of commercial applications such as Facebook, Youtube and Amazon are based on machine learning algorithms. Also, these algorithms continue to be developed and used in areas such as image processing and suggestion systems [23].

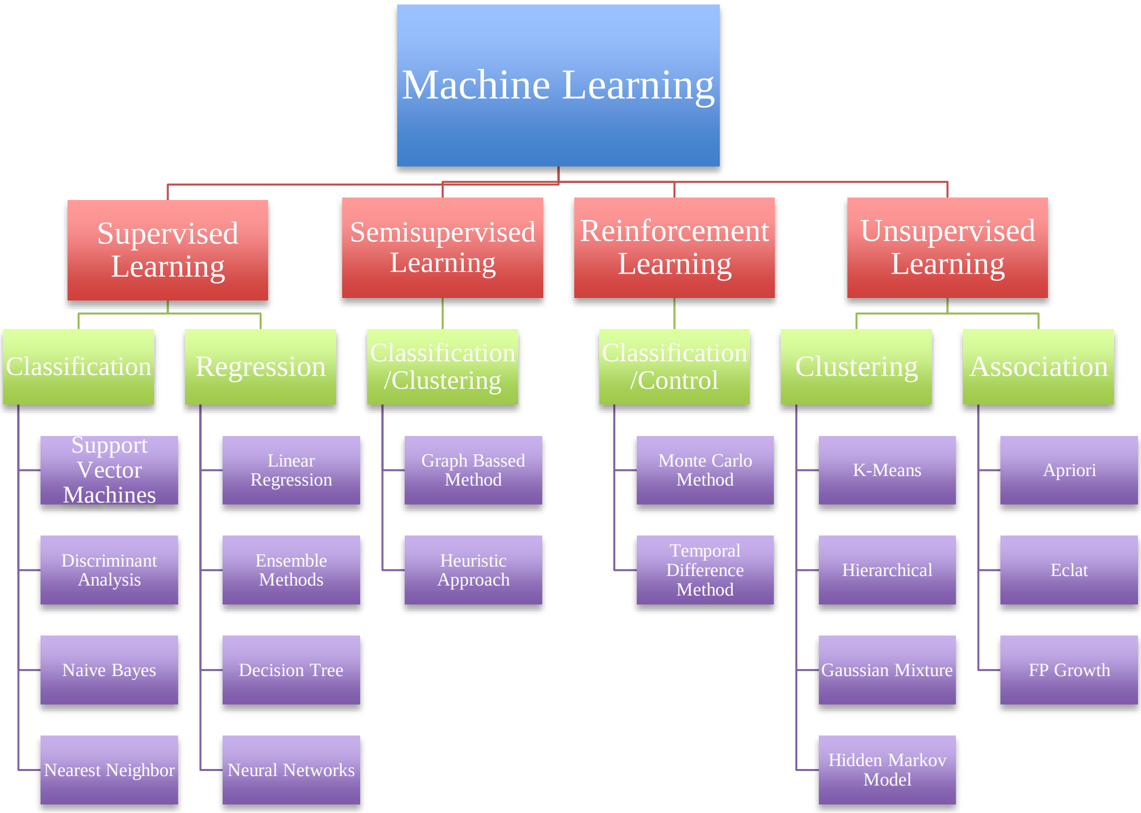


Figure 2.4. Machine Learning Methods. Adopted from [25]

Machine learning is a modeling using a specific data set to solve the problem. In this method, inference is made from the existing data. A number of estimates are then made for the data unknown from these inferences. Inference is based on learning in machine learning techniques [24]. Learning techniques are divided into supervised, unsupervised and semi-supervised learning, shown in Figure 2.4.

2.3.1. Supervised learning

The feature of this learning is to generate solutions by generalizing the system according to an input and output vector, shown in Figure 2.5. While training, the system changes the weight values by looking at these input data and produces the result. The system needs to generalize the examples shown to create a solution set.

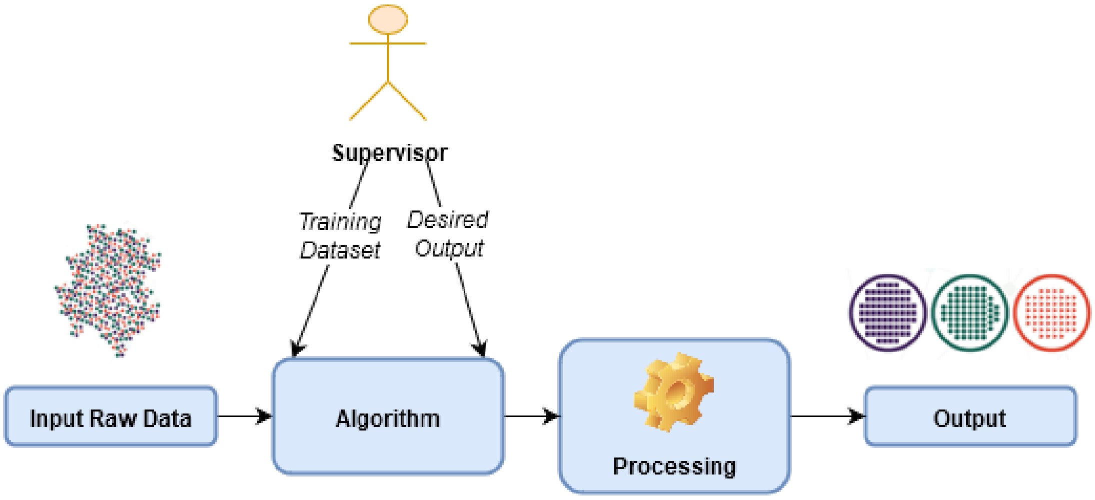


Figure 2.5. Supervised learning

Supervised machine learning models mainly try to solve two problems: regression and classification. To solve these problems, machine learning algorithms generate predictions appropriate to the data. If the estimation result consists of categorical data, this is called classification. The regression is called the numerical result [26].

2.3.2. Unsupervised learning

Unsupervised learning is a form of learning where there is no output data corresponding to input data. The algorithms used for this learning are aimed to reach more information about the data. Therefore, the data should be analyzed and modeling should be done.

Therefore, unknown outputs are estimated. According to Figure 2.6, it is necessary to use this data in order to analyze the relationships between the data. Especially, these learning algorithms are used in the interpretation of big data sets [24].

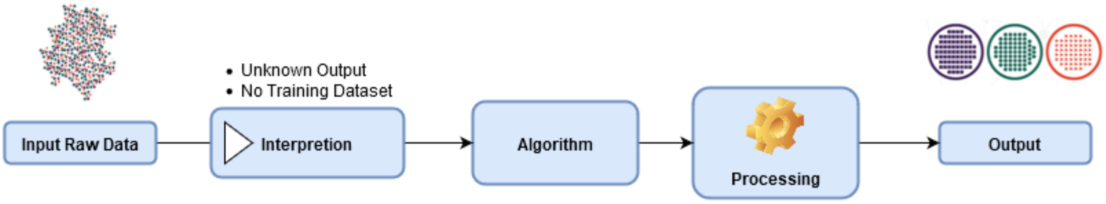


Figure 2.6. Unsupervised learning

Furthermore, in this learning model, the system is designed with a set of inputs without target outputs. The purpose of this model is to identify the data that may be in a similar group and to create a pattern for different ones.

Clustering and the association are two sub-branches of unsupervised learning. First, the clustering problem is to group similar data in a homogeneous distribution. Second, the association is the problem of finding certain rules between data.

2.3.3. Semi-supervised learning

Semi-supervised learning is a form of learning where a part of the input and output data in the data set is known and a part is not known. Therefore, it includes both aspects of supervised and unsupervised learning. During the training, the labeled data is used with supervised learning models, while unlabeled data is modeled with unsupervised learning (See Figure 2.6).

In the first step, labeled data is used to identify the groups to which the data can belong. Training with unlabeled data is the second phase and can label this data [28].

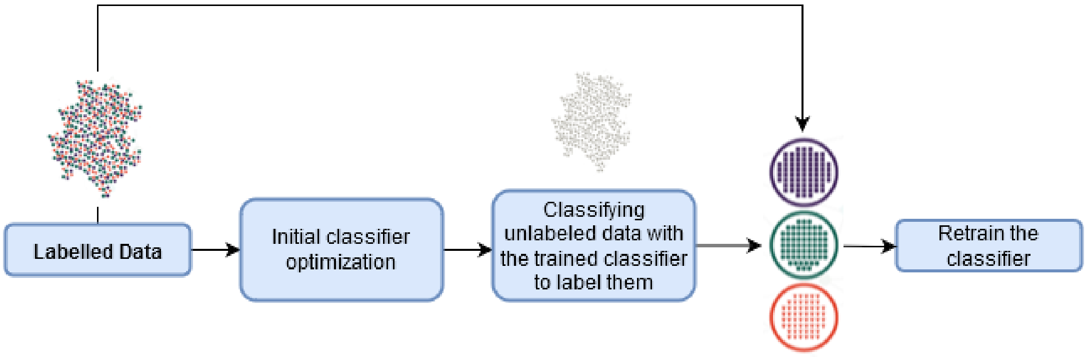

Labelled Data
Initial classifier optimization
Classifying unlabeled data with the trained classifier to label them
Retrain the classifier

Figure 2.6. Semi-supervised learning

The labels in the data set used in this thesis are determined. For this reason, classification algorithms mentioned in supervised learning algorithms were used.

2.3.4. Reinforcement learning

The Reinforcement Learning model is different from supervised or unsupervised learning models. Therefore, it has its own terminology. The first concept to be explained in this terminology is the agent. The agent is defined as a hypothetical entity that performs actions to reward in an environment. The second concept is action that all possible actions the agent can take. The third concept, the environment, is called a scenario in which the agent confronts. In the fourth concept state, the current state information returned by the environment is available. The fifth concept is the reward that is transformed back from the environment to evaluate the last action from the medium. In an agent and reward structure, the system is self-learning. In other words, it acts for a purpose. In Reinforcement Learning, a virtual model is created for each environment. Then the agent tries to learn in this particular environment. Since the model is different for each environment, there is no specific solution or algorithm for this type [26-29].

Figure 2.7 shows the structure of this learning model. In this structure, the action of the agent depends on the environment and a response is expected from the environment. These reactions are based on a predetermined reward system. The agent is trained according to the award. Therefore, the number of actions the agent tries is important.

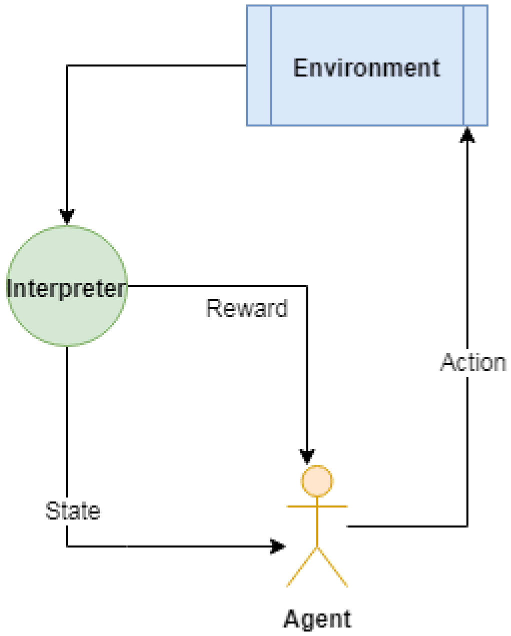


Figure 2.7. Reinforcement learning

3. RELATED WORK

In the literature, the methods developed for phishing attacks can be grouped in many different ways. In this thesis, the literature is classified according to the scheme made by Rao and Pais [40] (Figure 3.1). These include list-based methods, heuristic-based methods, vision-based methods and machine-based methods.

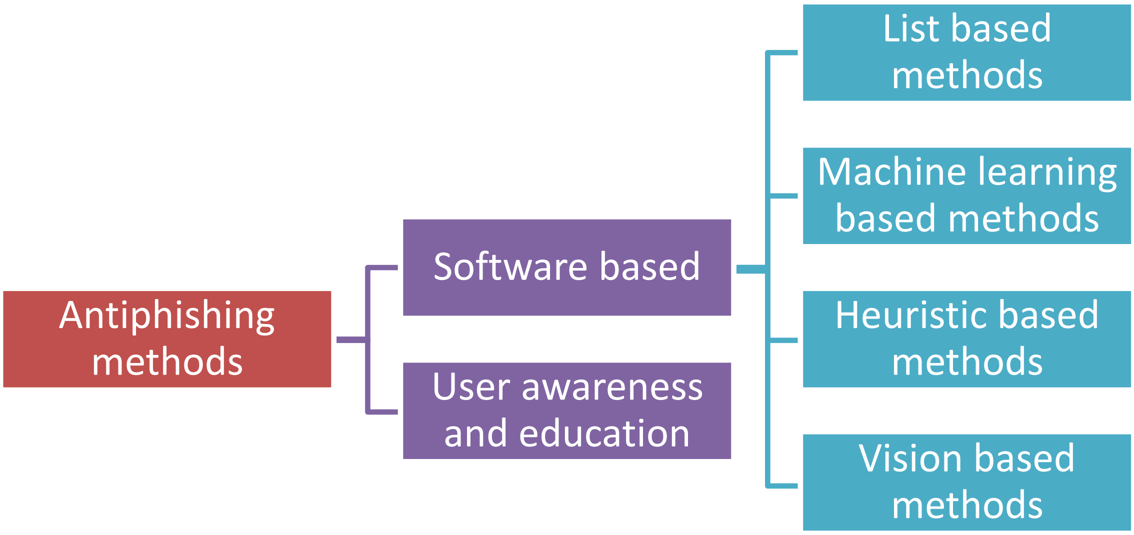


Figure 3.1. Taxonomy of anti phishing solutions [39]

First, heuristic-based methods use the text, image and URL sources of web pages to collect information. Then, a decision function based on various machine learning techniques must be established, and this is done through feature extraction [40]. Heuristic-based approaches include list-based methods and will be explained in the next section. In machine-based approach; It is based on machine learning algorithms such as Random Forest, logistic regression, multilayer sensor, Bayesian network, support vector machine (SVM) [41]. These methods are arranged according to their ability to work more efficiently in large data sets [39].

3.1. List-based approaches

It is based on lists created in response to re-attack in list-based security mechanisms. The lists created in these mechanisms are characterized by a previously attacked URL or IP address. In other words, lists need to be separated into harmful and harmless lists called black and white. Also, these lists must be constantly up to date. The disadvantage of these mechanisms is that they cannot detect an attack that has not occurred before or an attack that has not been detected before. The list-based methods used by the Google Safe Browsing API [42] are used. In the basic working principle, it divides web pages into black and white lists. It is usually based on URL information. This provides a website that provides protection against attacks. However, since a new phishing web page is published in a very short period of time, the blacklist needs to be updated regularly and quickly [46].

Black and white list-based methods are used to prevent phishing using a database of both trusted (whitelist) and phishing (blacklist) websites [43-45]. These databases can be saved and updated on the client's machine, or lists are stored centrally on the server's computer.

Whitelists are lists of trusted websites that an internet user regularly visits. This technique only allows the user to go to a website that is considered legitimate. This method is very effective in performing zero-day phishing attacks and also makes zero false-positive results. The biggest disadvantage of the whitelist is that it is difficult to manage all the websites which users will visit in the future. When a user chooses a legitimate website that is not listed in the white list, the system will consider it a phishing website that increases the false negative rate. This means that the whitelist is not very popular [44, 45].

In blacklist approaches, the requested URL is compared to a predefined phishing list. This approach is a well-known technique used to manage phishing attacks. It contains URL information of web pages that are likely to be used by attackers. Most famous web browsers, such as Google Chrome and Internet Explorer, use blacklists for phishing prevention [44, 45].

Cao et al. [47] studied a whitelist-based system. It used for recording the IP addresses of each website. However, it only consists of web pages with user interface with login screen and adds the first entered sites to the blacklist.

Jain and Gupta [48] have developed a method to solve the automatic update problem of list-based techniques. Domain-IP mapping and source code properties are used in this method. Also, the Google Safe Browsing API is an application made with this technique [49, 50].

PhishNet is a blacklist based application. It has two different components, the new URL discovery and matching algorithm. Different methods have been proposed to eliminate the disadvantage of the Blacklist approach. Also new phishing URLs have been added to the list [51].

3.2. Machine learning based approaches

In the literature, machine learning is used to detect phishing attacks. When these attacks is considered as a document classification or clustering problem, there is a classification problem that can be solved by using machine learning methods. Labeled data sets related to two classes, primarily harmful and clean, are used in this method [40]. Then, a model should be created with this data in accordance with the solution. During the creation of the model, the data must be passed through the training process. Finally, the success of the model is tested.

In the literature, a phishing protection system was developed by Pan and Ding [51] based on the properties of DOM objects. Malicious URLs detected in SVM. Moreover, in this study, the basic properties of the suspicious website such as title and URL were used to extract the properties of DOM objects. For these processes, a phishing detector has been developed according to the differences between the structural and URL information of a legitimate web page and the suspicious web page.

In the CANTINA study, harmful and harmless sites were removed according to the content of the websites. In this study, the TF-IDF algorithm was used. The text content of the website is considered. Also, intuitive features and suspicious and legitimate web sites were classified [52].

Miyamoto et al., based on CANTINA numerous machine learning algorithms are used in the classification, but the best result is AdaBoostM1. F1 measure was calculated as 85.81% [53].

In the study conducted by Xiang et al., CANTINA + used Bayesian network classifier, which has its HTML-based features. This study is based on the CANTINA study. According to the content information, 99% accuracy was achieved in the experiments where phishing sites were detected. However, it cannot work with the images on the website [54].

He et al. made a study using heuristic algorithms based on CANTINA study. Also, they used property extraction methods called Anomaly and PILFER for classification. However, the disadvantage of working is the high uptime because the search engine uses third-party services such as page rankings [55].

Gowtham et al. used the heuristic model and machine learning approach together. In this study, suspicious and legitimate pages were classified according to URL information, which is the support vector machine algorithm and reached a true positive rate of 99.65%. The disadvantage of the study is that the developed technique depends only on the text content [56].

Aggarwal et al. tried to detect phishing attacks on tweets. In this work PhishAri, URL text was handled. Then, classification was made with using Random Forest algorithm. Because this method is URL-based, it may not be able to detect phishing attacks on website content [57].

Tan, Chiew, Wong, and Sze [58] called their study PhishWHO. This system detects phishing in three steps. The keywords of a suspicious web page are identified in the first step. Then the keywords are detected in the search engine. The aim is to find out whether the target domain names will attack. At the last stage, the legitimacy status of the site is determined. They achieved 96.10% success using this method. Since the detection of image contents on the website does not exist in the developed method, it may not give effective results in attacks on them.

Le, Markopoulou and Faloutsos [59] used the URL properties of web pages to detect the phishing attack. They were later classified by Support Vector Machines. In this study, only Twitter data set was used and the data was evaluated only according to URL information.

In the study conducted by Jeeva and Rajsingh (2016) [60], attacks were detected according to the messages coming from suspicious web pages. For this purpose, incoming messages are filtered. With the multi-layered classifier they designed, URL-based features were extracted. Approximately 88% have achieved test accuracy success.

Babagoli, Aghababa and Solouk (2018) [61] detect phishing attacks with nonlinear regression strategy. In this method, harmony search and support vector machine algorithms are used together. The harmony search algorithm was used to find the relevant features. In the next step, support vector machine algorithm was used as a classifier. The success rate for detecting phishing attacks was 96.32%.

Buber, Diri and Sahingoz (2017) [62] proposed the NLP-based method for phishing attacks. In this method, words are expressed with NLP vectors and machine learning algorithms are used.

Mohammad et al. [63] used adaptive self-structuring neural networks to detect phishing attacks. They achieved a high success rate in noisy data. Additionally, more phishing sites have been removed with this method. However, since the dataset is old, the study should be repeated.

Jain and Gupta [48] labeled their website as legitimate and suspicious with machine learning techniques using hyperlinks. They classified online websites as true positive by 99.39% with machine learning techniques.

Feng et al. (2018) [64] distinguished phishing web pages from legitimate pages with artificial neural networks. They used URL-based features for classification. The artificial neural network model is based on Monte Carlo algorithm and risk reduction principle. They calculated 97.71% accuracy rate in this method. However, the disadvantage is that the data set used is out of date and work with the URL.

Smadi, Aslam and Zhang (2018) [65] conducted a study on the detection of phishing attacks with artificial neural networks. They calculated 98.6% accuracy rate using neural network. However, in this study, only phishing attacks with e-mails were discussed.

Rao and Pais [66] used both machine learning and vision-based methods in their study. The system improved the accuracy of phishing pages on the server side. Moreover, it has enabled the detection of new legitimate and phishing sites. It has been determined with the accuracy of 98.61% by calculating the similarity according to the content of the websites such as logo and image. In addition, this system is language independent.

In the study conducted by Rao, R.S. and Pais, A.R. [40], phishing attacks were detected using URL information. Therefore, CANTINA and CANTINA + studies are based on 99.55% accuracy rate.

Sahingoz et al. [39] have more than 20 handcrafted features that will be extracted from the only URL of web pages using Random Forest classifiers. Their detection accuracy has been reported by over 97%. However, these features are prone to be easily discovered by attackers.

3.3. Heuristic based approaches

The components of detection tools such as fraudulent messages, e-mails and web pages are used to detect the phishing attacks based on the heuristic method [66]. The content information of these tools is discussed in this chapter. In the literature, these attacks are detected by using heuristic methods and machine learning algorithms together. Therefore, Rao and Pais [66] can also be evaluated in this section. It is not necessary to ensure accuracy in heuristic methods. The aim of these methods is to make a complex problem simpler or the algorithm can find a successful result. So, the solution of the problem should be fast, but it should be achieved under all circumstances.

Heuristic algorithms can be used in applications for server or client machines, such as browser toolbars, firewalls, or antivirus software.

These algorithms have been studied by 3 different techniques in the literature. The first of these techniques is URL-based. This technique attempts to identify suspicious and legitimate websites by removing the URL properties. The URL properties mentioned in the method are determined by the number and structure of the characters and the frequency distribution of the characters. The second is based on source code that text content, DOM tree properties, tag and image properties can be accessed to examining the source code of web pages. Then, these are used to define the legitimacy of the website. The third is service-based techniques which are used in the search engine services such as indexing and page rankings.

3.4. Vision based approaches

Phishing is a type of attack based on the imitation of web pages. Phishing sites that are visually similar to target sites are used in this attack. The vision based approach is claimed to be a hidden feature of visual similarity between these two sites. Therefore, the detection of the images of the actual target sites is done with these techniques [40].

Recently, since phishing web pages are visually similar to their counterparts, vision-based approaches have emerged to create effective and efficient classifiers. In general, vision based approaches attempt to extract a visual signature (i.e. feature vector) from the source web pages by utilizing local or global image descriptors. These signatures are either compared or used to create multi-class classifiers.

The vision-based phishing protection literature covers numerous studies using different basic approaches. From these studies, [67] tried to detect phishing using machine learning methods, as well as performing corner analysis using contextual features in the form of an heuristic schema.

The work by E. Medvet et al. [68] has been developed to compare a phishing web page with a legitimate one that includes text pieces, images embedded in the page, and the final image of the page. A small data set of 41 samples was used for the evaluation of the model and the false negative rate (FNR) was calculated as 7.4%.

In [69], Zhang et al., have suggested an approach that considers spatial layout of web pages. They have constructed an r-tree based indexing technique for determining the visual similarity among the web pages under suspicion.

Rao and Ali [70], have proposed a scheme based on matching SURF features extracted from legitimate and phishing web pages. According to their idea, screenshots of phishing web pages can be identified through SURF based pairwise matching. Legitimate and suspicious web pages were determined and the similarities between them were calculated using these characteristics.

Hara, M. et al. [71] identify phishing web pages using images and URL information. In the vision-based detection section of the study, the authors used the “ImgSeek” tool to detect visual similarities between images hosted online and those under review. As a result of this study, 82.6% phishing detection rate was calculated. Although their work is correct, the proposed approach requires a third-party service and effectiveness depends largely on the quality of the interrogation and reception of the dependent service.

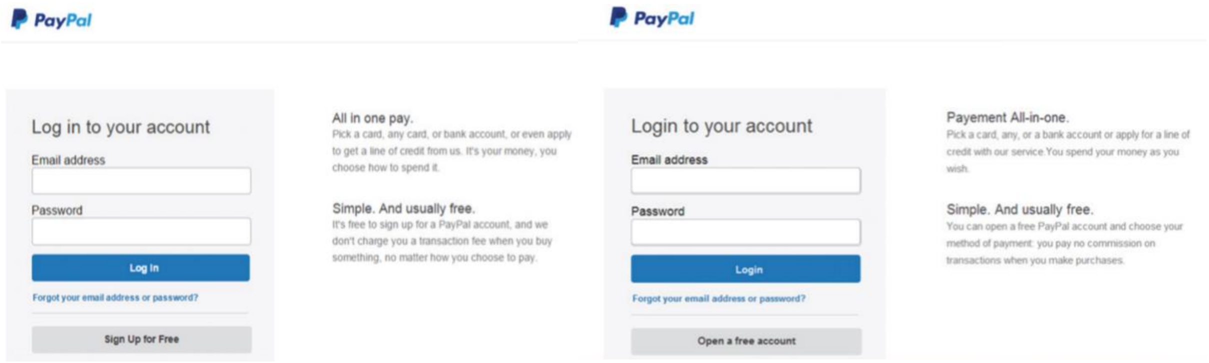
In [72], visual similarity between suspicious and legitimate web page pairs have been studied through earth mover’s distance metric (EMD), a measure of the distance between two objects. Although their results are satisfying, their proposal is not scalable due to underlying feature extraction and optimization strategy. The missing aspect of this study is that web pages are the same size and do not distinguish similar color representations on the pages.

In another vision based study [73], a scale and rotation invariant descriptor namely CCH (i.e. Color Context Histogram), have been used to find visual similarities between legitimate and suspicious web pages. In the first step of this approach, keypoints are extracted from the web page snapshot using this descriptor. Then, they are compared and a similarity ratio is obtained based on the matching. In this way, the web page is determined to be phishing.

Bozkir and Akcapinar Sezer [74] detected phishing attacks with HOG descriptor. In this method, the edge and corner properties of the image are discussed and feature extraction is performed according to these parameters. Then, feature extraction was performed using HOG. It was determined whether the histograms formed were phishing pages according to a certain similarity ratio. It is open to research that the number of samples in the data set used for testing is low. This thesis is based on the work of Bozkir and Akcapinar Sezer. In this thesis, feature extraction is performed with different descriptors. Also, suspicious and legitimate pages were separated based on the threshold value which is similarity value. However, in this thesis, classification is made with machine learning methods.

Dalgic et al. [75] distinguished their legitimate and phishing web pages using MPEG-7 and MPEG7-like compact visual descriptors. Experimental results were performed with SCD, CLD, CEDD, FCTH and JCD, in addition, feature vectors from these descriptors are expressed as “holistic” manner and “multi-level patches”. In the next step, phishing pages and legitimate pages were classified using SVM and Random Forest. A data set, called "Phish-IRIS", containing 1452 brands and 2852 samples was used in the evaluation stage. This data set was collected from PhishTank and Openphish platforms.

In the light of the information described above, the main purpose of phishing attacks is to trick innocent users into acquiring their information. This approach compares the visual characteristics of the suspicious web page and the legitimate web page. Firstly, the attackers create visually similar or even the same websites as their target websites using HTML text such as images, flashes, movies. Secondly, DOM-based solutions are inadequate in detecting such sites. Also, detection methods based on texts will not be effective in these cases. The methods developed by phishers, which based on DOM and HTML have become less effective.



All in one pay.
Pick a card, any card, or bank account, or even apply to get a line of credit from us. It's your money, you choose how to spend it.

Simple. And usually free.
It's free to sign up for a PayPal account, and we don't charge you a transaction fee when you buy something, no matter how you choose to pay.

Payement All-in-one.
Pick a card, any, or a bank account or apply for a line of credit with our service.You spend your money as you wish.

Simple. And usually free.
You can open a free PayPal account and choose your method of payment: you pay no commission on transactions when you make purchases.

Figure 3.2. Example of legitimate and phishing web sites (Adopted from [8])

Phishing attacks are a type of attack that enables hackers to steal credentials using social engineering techniques. In this attack, hackers are trying various ways to mislead users. Hackers make changes to legitimate web pages in various ways to trick users. Especially in terms of page layouts, images, text content, font size and font color features of legitimate and phished websites, the pages are compared to each other, but they cannot be exactly the same. In this case, the pages can be detected using vision based methods, although they are not noticed by the users. Initially, websites that were completely similar in appearance were using different URLs. However, nowadays, the address bar containing the URL information is hidden and the websites are delivered to the users. Many users are deceived into the appearance of the website and send it to hackers without even looking at the presence of the URL information. Some users consider URL information. For this reason, the security of URL information is proved by SSL (Secure Socket Layer) certificate. Hackers, who make the illegal form of web pages with this certificate, use methods of obtaining fake SSL certificates to trick users. In this way, web pages of exactly the same feature is formed. Hackers may not be able to create web pages with exactly the same visual characteristics, and this cannot actually be noticed by users. In other words, users cannot distinguish visually similar web pages. Therefore, visual based phishing detection methods should be used. In the literature [8] the similarity of phishing web pages to legitimate pages can be grouped under 4 different methods, which consist of visual appearance, address bar, embedded objects and favicon similarity. Figure 3.2 provides screenshots of legitimate and phishing web pages, respectively.

List-based and heuristic-based approaches used to identify phishing web pages address URL information. In the list-based approaches, the detection rate of phishing web pages in the literature is high, but the web pages used in the experiments remain in the out-of-date. Statistics show that the majority of these web pages are refreshed within twelve hours. Therefore, phishing blacklist solutions based on the URLs at that time need to be kept up to date. Similarly, heuristic based solutions detect phishing web pages based on predefined rules. In this case, it is not possible to determine the pages prepared according to the new features.

In line with these explanations, computer vision approaches have been proposed. In this thesis, it is suggested to realize the identification and recognition of phishing sites with a vision based approach, because today's web pages contain complex graphical elements. Phishers have also begun to provide the content they will attack through image elements. Computer vision methods can detect such attacks. In addition, it has become ineffective in new methods developed by attackers in DOM-based and HTML-based analyzes in list-based approaches. In addition, vision-based approaches give faster and more reliable results. This approach occupies low memory space, the detection speed is high. Moreover, vision-based approaches provide more reliable results by requiring a slightly more processing power. The current data set, namely “Phish-IRIS”, prepared in this thesis will be used. Details of the data set will be explained in the following sections. Unlike compact visual descriptors, global and local descriptors are used for feature extraction from the data set. After that, machine learning algorithms are used in the evaluation stage of the model.

4. METHODS AND TOOLS

In phishing attacks, there is a constant conflict between phishers and anti-phisher. The victims of the attack are innocent users. To capture the personal information of these users, deceptive visual content is created. Different methods are used for this, such as deceptive use of HTML (Hyper Text Markup Language) tags, but DOM (Document Object Model) based solutions are inadequate to detect these techniques. For this reason, computer vision approaches are used. In addition, this approach helps to identify machine learning. In this thesis, phishing web page identification is realized by vision based approach that many methods and tools have been utilized for this detection problem. Thus, computer vision techniques are focused in this chapter of the thesis to detect and classify phishing web page. These are called visual descriptors that have not been tried in the recognition of the brands targeted by that. At the same time, two feature extraction approaches such as “holistic” and “multi-level patches” have been tried in order to obtain fine-grained analysis.

In the first stage, the proposal and a holistic approach are discussed that the whole page snapshot is given as input. In the second stage "multi-level patches" approach are made equal-sized parts to whole snapshot. Then, these parts are brought together to form a multi-layer structure. So, visual features obtained have been the input of machine learning algorithms for the performance measurement.

4.1. Visual Descriptors

Developing technology and widespread use of the Internet increase the importance of multimedia usage. Especially through smartphones, tablets and cameras, multimedia data can be collected more easily and rapidly. Therefore, there is an increase in the capacity of the databases where the data will be stored. Various methods have been developed to access multimedia data in such large databases. Moreover, there is a need for systems that define and classify the content of visual elements.

In the field of computer vision, visual descriptors that determine the basic visual characteristics (shape, color, texture) of visual media such as pictures and videos [31, 76] are defined. In the light of the information obtained from these descriptors, information about the content of objects and events in visual elements can be accessed.

As described in section 2.2, visual descriptors can be classified in different ways in the literature, but within the scope of this thesis they are classified as global / local descriptors, which will be explained in the following chapters.

4.1.1. Global Descriptors

Global descriptors evaluate a visual object as a whole. A visual object can be expressed in a single vector by means of this descriptor. Therefore, they can easily work with machine learning classifiers. These descriptors make inference according to the color, texture and shape properties of the image. Therefore, shape and texture descriptors are included in this group. Moreover, this descriptor is used for problems such as object recognition because it evaluates the entire image.

4.1.1.1. GIST

The GIST identifier was developed for object identification using spatial enveloping of a scene or image in a 2001 study by Oliva and Torralba [77]. For this reason, they used various features of objects of certain shapes and sizes of the image. Accordingly, the developed model is called a spatial envelope, which shows both the properties of the objects in the image and the frame of the surface. It is also the low-dimensional state of the scene indicating the correlation between them. This descriptor was used in the stage classification problem, and the spatial envelope actually corresponded to that, in addition it could solve the combination of scenes according to similar characteristics. In order to determine the properties of the space, 5 basic properties which are expressed as naturalness, clarity, roughness, expansion and strength were determined.

Naturalness is related to the types of lines within the stage. Straight lines are said to be a human-made object, while wavy lines are natural. Openness is a feature that can be used to determine whether space is open or closed. Since it is more likely to be found in closed spaces, it is more likely that the space to be classified is more likely to be closed. Roughness is defined as the complexity of the place by looking at the detail information in the space. Expansion determines the orientation of the line according to the perspective in the scene. Approaching parallel lines mostly considered with a depth of inclination on the space with respect to observers’ view. In other words, it can be understood by looking at the lines outside the building to determine how high a building is [77].

In the scene description, it is shown that there are 3 levels including subordinate, basic and superordinate. Moreover, it is done by using the combination of the superordinate and basic level to perform stage classification in spatial envelopes. While the level of the Superordinate level is based on the color and corner characteristics of the scene, the basic level is expressed as a level that enables the objects in the scene to be grouped by similar shapes [77].

Image based representation is expressed as DFT and WFT conversions, as shown in (1) and (2). These transformations perform the necessary transformations by adjusting the pictures according to the energy spectra of the picture. WFT is used for modeling [77].

$$I(f_x, f_y) = \sum_{x,y=0}^{N-1} i(x,y)h(x,y)e^{-j2\pi(f_x x + f_y y)} \tag{1}$$

$$I(x, y, f_x, f_y) = \sum_{x\prime,y\prime=0}^{N-1} i(x', y')h_r(x' - x, y' - y)e^{-j2\pi(f_x x\prime + f_y y\prime)} \tag{2}$$

After this step, linear regression methods were needed to estimate the properties of the spatial envelope. To define these features, the WDST function was developed and compared with DST, as shown in (3) and (4). This function determines the appropriate results of the 5 basic properties and performs the dimension reduction process.

$$\begin{gathered} d_i\psi_i(f_x, f_y) \\ DST(f_x, f_y) = \sum_{i=1}^{N_G} \end{gathered} \tag{3}$$

$$\begin{gathered} d_i\psi_i(x, y, f_x, f_y) \\ WDST(x, y, f_x, f_y) = \sum_{i=1}^{N_L} \end{gathered} \tag{4}$$

Euclidean distance was calculated in order to measure the similarity of openness, ruggedness and roughness with the target stage. In-class k-NN algorithm was used. In this way, it is provided to classify the scenes by providing semantic information extraction about the scene without requiring the shape or identity of the objects [77].

The first step of the GIST descriptor is to extract properties [20, 21]. Therefore, the image is divided into nxn blocks. In this way, both information loss is prevented and useful features are obtained. Gabor filters are used in the second step. Those blocks are sent to them. After each block is processed in different directions and scales, the values are added. This consists of the vector representing the image. The mathematical equation is shown in (5) and (6).

$$u_0 x_{\theta_i} + v_0 y_{\theta_i}$$

$$2\pi j(\quad)$$

$$G_{\theta_i}^{s} = Cexp\left(\frac{-\left(x_{\theta_i}^2+y_{\theta_i}^2\right)}{2\sigma^{2(s-1)}}\right)exp \tag{5}$$

$$\begin{cases} x_{\theta_i} = xcos\theta_i + ysin\theta_i \\ y_{\theta_i} = -xcos\theta_i + ycos\theta_i \end{cases} \tag{6}$$

The vector consists of GIST descriptor and has a size of 960. In this calculation, 3x (4x4) x (8 + 8 + 4) is equations. The equation is constructed as follows: an image has 3 color channels (R, G, B), and each image is sent to 4x4 dimensional blocks in this descriptor. The GIST descriptor consists of 2 finer 8 orientations and 1 corser 4 orientations [78, 79].

4.1.1.2. LBP

The Local Binary Patterns (LBP) algorithm was developed by Ojala et al. And provided solutions to problems such as pattern classification, face recognition, pedestrian detection, and stage classification [80]. In the LBP algorithm, the central pixel and adjacent pixels are compared. The algorithm converts the binary code to each pixel and the corresponding equation is shown in (7).

$$LBP_{P,R}\,(x_c, y_c) = \sum_{i=0}^{p-1} s(g_i - g_c)x2^i \;;$$

$$s(x) = \{1 \;\; if\; x \geq 0;\; 0 \quad otherwise \tag{7}$$

In the LBP algorithm, the input image is used in grayscale. Therefore, the first step of the algorithm makes the image grayscale. The neighbors at the distance r of the center pixel are then determined and the LPB value is calculated. These values are saved as a two-dimensional array.

Figure 4.1 shows the LPB value calculated by the algorithm for the center pixel. At the central pixel highlighted in red, 3x3 neighbors are located on a fixed grid, and 8 neighboring pixels are called thresholds. The value of the central pixel is determined by looking at the neighbors. The center pixel, which is equal to or greater than its neighbor, is 1, otherwise 0. After this comparison for all neighbors, an 8-bit property vector is formed. After this calculation is made for the whole image, the vectorized version of the image is obtained.

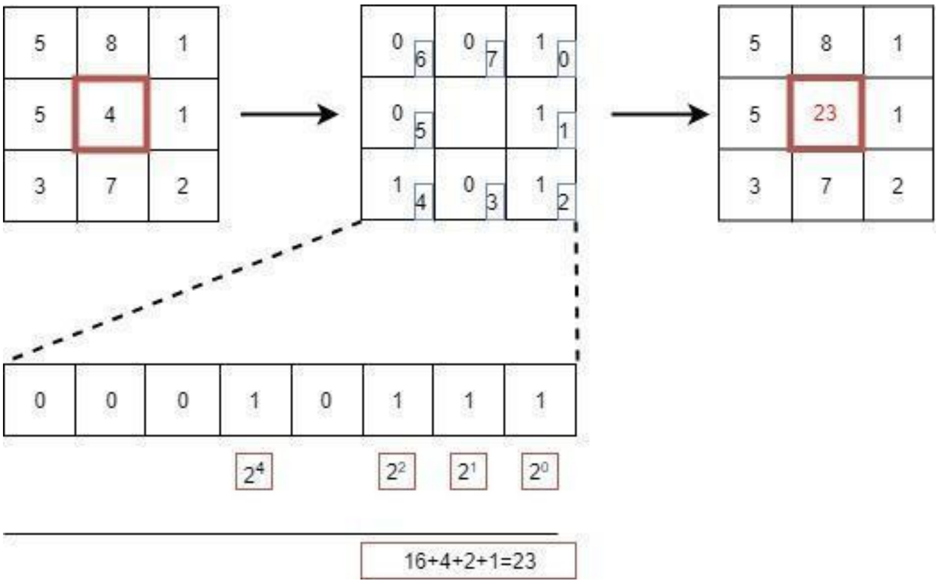


Figure 4.1. Example of the LBP algorithm

4.1.1.3. HOG

HOG is a descriptor that is frequently preferred in computer vision and image processing. This descriptor is used for object recognition in the field. The algorithm works as follows: first, the image is divided into cells and the gradient orientations are calculated for the pixels in these cells. The gradient sizes calculated from gradient orientations are for each pixel and histograms are generated from Dalal and Triggs [82]. In the next step, the histograms are normalized.

By definition, the HOG features produce a gradient based visual cues for revealing the corner-edge characteristic of the input image. In particular, HOG descriptor divides an image detection window into small connected regions called cells and calculates the histogram of the gradient directions or edge directions of the pixels within the cell for each cell followed by a normalization stage [86]. Figure 4.2 shows the histogram of oriented gradients of an image.

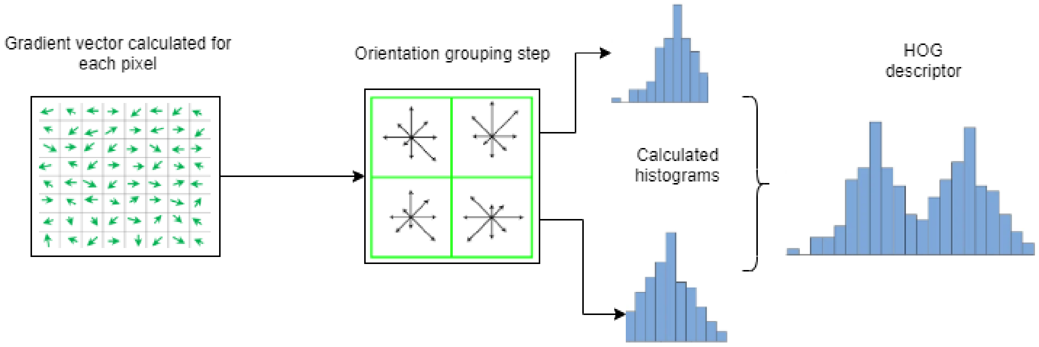

Gradient vector calculated for each pixel
Orientation grouping step
Calculated histograms
HOG descriptor

Figure 4.2. Using HOG in an image

HOG uses the Sobel filter, which detects horizontal (I_x) and vertical (I_y) edges. The formula for this process is given in (8).

$$\theta = arctan\frac{lx}{ly} \tag{8}$$

In this thesis, the cell size property of the HOG descriptor was utilized. Firstly, we have either resized or cropped. Secondly, the cropped image provides an information loss at the edges of the screenshots whereas resizing distorts the edge structures. Additionally, we have preferred two different cell sizes (32 and 64 pixels).

4.1.2. Local Descriptors

Local descriptors extract properties of a particular region in images. These descriptors focus on image regions that can separate images and keypoints, which are called areas of interest. These points are then compared [31].

4.1.2.1. SIFT

SIFT (Scale Invariant Feature Transform) descriptor was developed by Lowe [83]. The main purpose of this descriptor is that a vector state is generated from the points around the key points of the input image. Also, this vector is usually 128-dimensional and uses image gradients that are called rotational and scale invariant description. Image identification, logo detection, and various geometric transformations have been used more commonly with SIFT [84].

SIFT is an algorithm that identifies regional properties of an image against lighting, rotation, and scaling. SIFT performs feature extraction in four steps. The first step is that the SIFT introduces the detection keypoints. In other words, it determines the endpoints of the image at minimum-maximum points called "scale-space extrema detection". The second step is "keypoint localization". The third step is referred to as "orientation assignment". In the last step, keypoints descriptors are represented [83, 84].

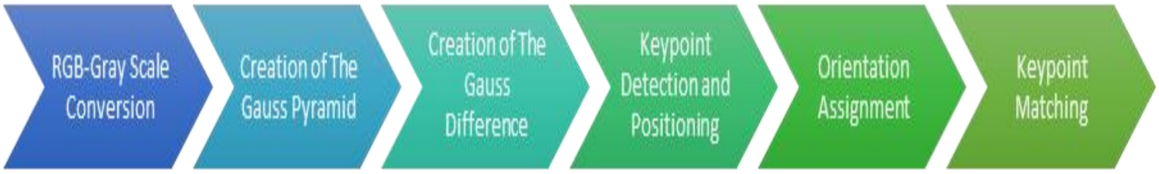

RGB-Gray Scale Conversion
Creation of The Gauss Pyramid
Creation of The Gauss Difference
Keypoint Detection and Positioning
Orientation Assignment
Keypoint Matching

Figure 4.3. SIFT algorithm flowchart

As can be seen from Figure 4.3, the first step of the SIFT algorithm is keypoint discovery. The algorithm starts with the creation of image sets of various sizes from the input image and thus the size space is formed. Then, the image is processed using Gaussian filters which are applied to measure the differences between the blurred image and the original. These differences are the DoG (Difference of Gaussian) points that give the keypoints of the related image that highlight the edges of each dimension, as shown in Figure 4.4 [21].

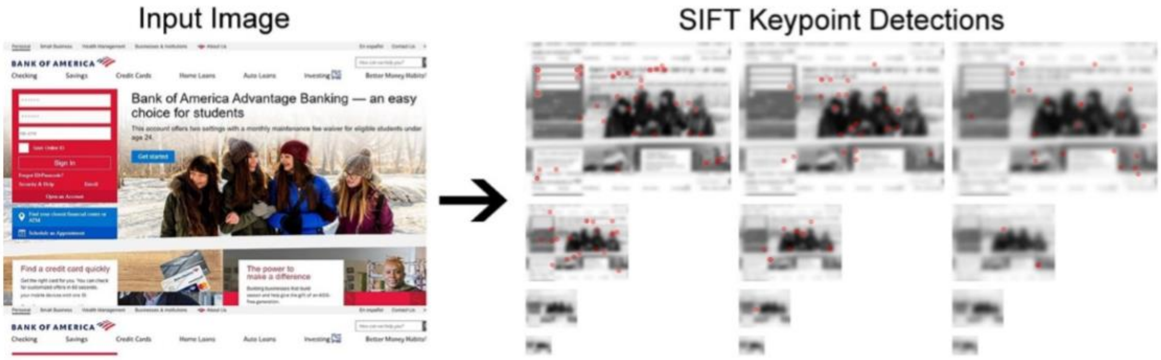

Input Image
SIFT Keypoint Detections
BANK OF AMERICA
Bank of America Advantage Banking — an easy choice for students
Find a credit card quickly
The power to make a difference

Figure 4.4. SIFT keypoint detection [14]

In the second step, the locations of the keypoints are determined. The purpose of this process is to identify points whose location is not exactly clear. Once these points have been identified, wrongly located points can be eliminated using two different techniques. First method is that the correct location is found by interpolating the data in the neighbors of the point. The second method utilizes the DoG function to detect keypoints, even with low-contrast. The formulas of Gaussian filter and DoG function used in SIFT are given in (9).

$$L\sigma(x,y) = I(x,y) * G\sigma(x,y) \tag{9}$$

$$DoGkn{+}1\sigma(x,y) = Lkn{+}1\sigma(x,y) - Lkn\sigma(x,y)$$

After the keypoints are determined, the orientation is assigned according to the slope size and direction of the neighbors of these points in the third step of the SIFT algorithm. It is depicted in Figure 4.5. In this step, the amplitude of the vectors defining the keypoints is calculated to determine the reference coordinate of the rotation angle of the image. Then, the values are placed on the histogram and the peak is expressed as the orientation value.

In the last step, the orientation values from the previous step are used for keypoint descriptor. Histograms which are adjacent to 4x4 pixels, each containing 8 boxes, are generated for these values. These histograms are calculated according to 16x16 neighborhoods [87]. As a result, these are 128 keypoint vectors of 16 * (4 * 4) each containing 8 boxes in total.

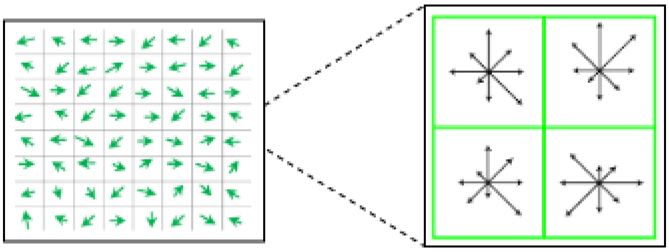


Figure 4.5. SIFT Keypoint and orientation (Adopted from [84])

4.1.2.2. DAISY

DAISY descriptor was developed by Tola [85] as a local descriptor and used in stereo applications. It is based on HOG and SIFT descriptors. It can perform better than the SIFT algorithm with less error rate and reduced calculation cost.

As shown in Figure 4.6, it establishes a daisy-shaped structure and generates interest points. 4 different parameters are needed to create this figure. The first of these parameters is "neighborhood areas radius (R)". The second is the "number of quantized orientations (o)". The third parameter is "number of convolved orientation rings (r)" and finally "number of circles on each ring" called "(c)". A 200-dimensional descriptor is obtained for each pixel in an image.

In the DAISY algorithm, circular configuration is used for the region to be determined as a keypoint. This is the difference from SIFT because it has rectangular grids. Similar to SIFT, Gaussian filters are used in the first step. Then, derivatives of these results are calculated for each pixel to determine the sub-samples. As a result, G_o orientation maps are created which are shown to be formulated in x.

$$G_o = \left(\frac{\partial I}{\partial o}\right)^+ \tag{10}$$

In this equation, *I* represents the input image. Each orientation in the image is indicated by *o*. G_o, the result of the equation, symbolizes the orientation map. The + sign is used to calculate positive values. After calculating, the Gaussian value for each pixel, a vector is formed for each circle. All vectors are then combined.

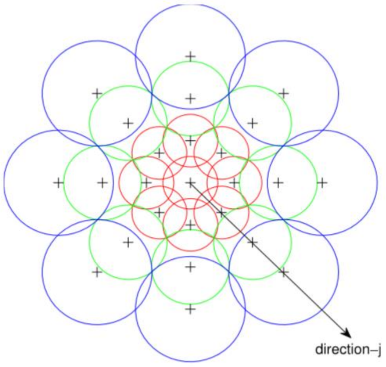


Figure 4.6. DAISY Keypoint and orientation [85]

4.2. Image Representation

Image representation is one of the approaches developed to solve the problems of computer vision such as image classification, object recognition or image segmentation. Many recommendations have been developed for this approach. One of them is the Bag-of-Words Model (BOW). Within this model to be explained is the spatial pyramid matching representation. These will be explained respectively in the thesis.

Many problems investigated in the field of computer vision (image categorization, object recognition, etc.) are realized with image representations. The first model proposed for this purpose is the BOW. This model focuses on a number of analysis results of features extracted using local descriptors. With the development of this model, BOW has emerged [31].

BOW has been used for problems such as texture recognition and document processing. When this model is used for documents, each word is called keywords and creates a dictionary of keywords. The expression of this document as words is the histogram.

The bag of visual words (BOVW) model was developed for images. As in the BOW model, visual words are extracted from images in this model. The difference between the two models is that in BOW, words are extracted from documents and called keywords, and in BOVW visual words are extracted from images. In dictionary, BOW model holds keywords and BOVW model holds visual words together.

The bag of visual words (BOVW) representation is a visual feature oriented version of the conventional well-known BOW concept which has been widely used in fields such as text classification and natural language processing. The main goal of BOVW is to represent an image as a set of pooled visual features regardless of how they were extracted.

There are three steps to define visual words in the BOVW model: feature detection, feature extraction, and codebook generation.

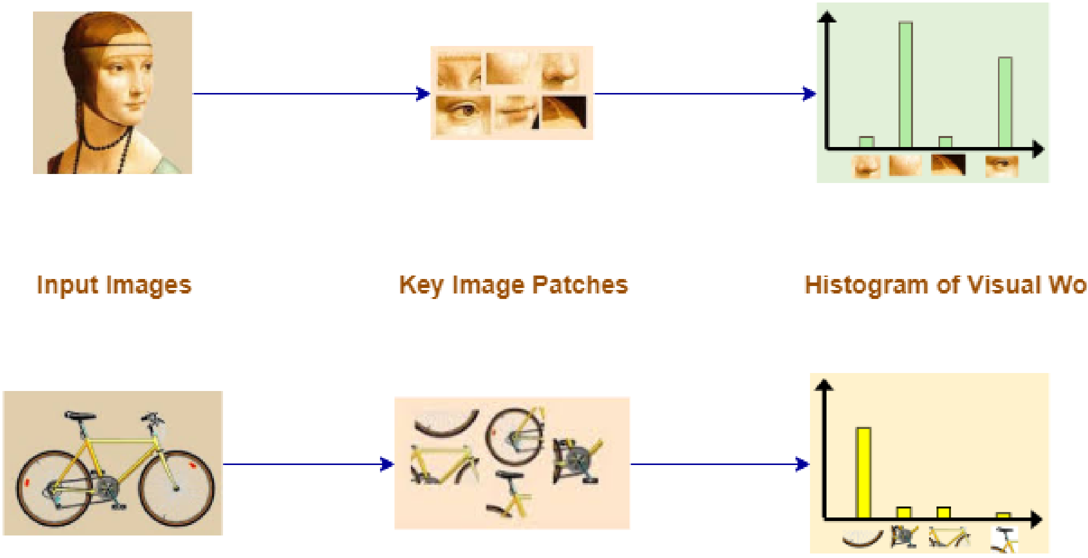


Figure 4.7. The Bag of Visual Words Model

In practice, generation of BOVW representation covers following stages: (a) extracting a number of feature vectors from the input documents (i.e. image), (b) clustering the accumulated features with a certain cluster count (i.e. visual words) and determining the cluster centroids , (c) for the image I, assigning each extracted feature vector to the nearest cluster centroid (i.e. quantization) and (d) representing an image with a histogram computed by counting the frequency of each visual word following to cluster assignment.

As a result of this pooling scheme, an image involving many feature vectors is being transformed into a histogram involving counts of visual words obtained before. As can be seen from Figure 4.7, the concept holds for BOW also exists in BOVW based representation that is, rather than of textual words, we use pooled visual features as the “words”. In this thesis, visual features are extracted using “holistic representation” in all screen shots and “pyramidal representation” that creates equal-sized patches.

During the installation of the visual dictionary using BOVW model, some attributes are extracted from the visual. The attributes are then clustered. After this, each different attribute in the dictionary is expressed in a different number of terms. The “nearest neighbors” algorithm is used to assign appropriate terms to qualifications. In the last stage, histogram vectors are obtained according to the number of terms.

Considering this approach, it is thought that the similarities can be determined by comparing the non-sequence properties of the two objects. From this point of view, the pyramid match kernel approach has been proposed, which identifies the interest points of the two images and forms the visual clusters. The similarity with the histogram intersection is then calculated. In this approach, there is an inability to hide relationships between term vectors and the Spatial Pyramid Match (SPM) approach has been proposed by Lazebnik et al [86].

Global descriptors produce properties based on the entire input image, but the spatial pyramid matching approach developed by Lazebnik et al. [86] produces more fine-grained features. This approach divides the image into rectangular regions of equal size, as shown in Figure 4.8. In addition, regions generate histograms at an increasing number of levels. The success of feature extraction according to the levels in the approach depends on the correct matching of visual words.

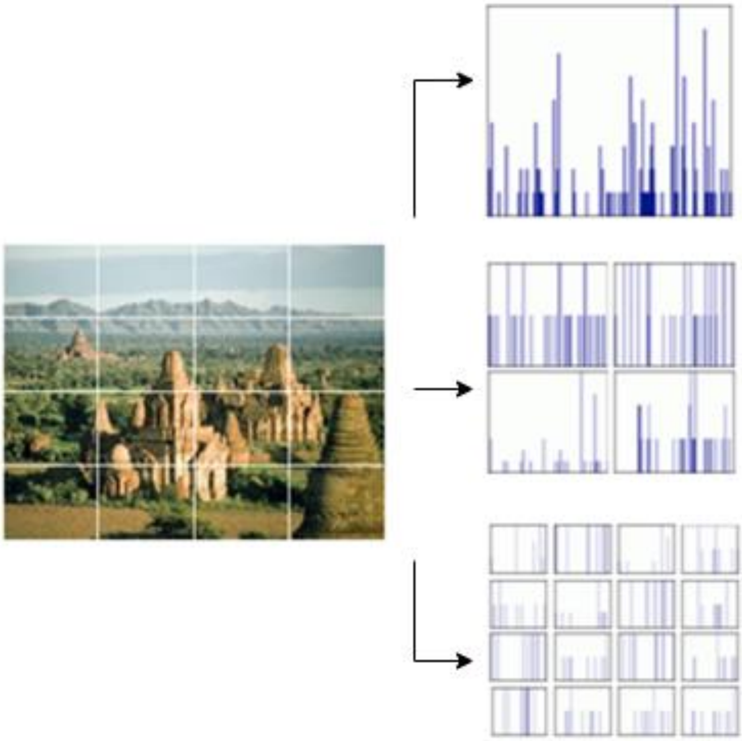


Figure 4.8. Spatial Pyramid Matching Model

4.3. Machine Learning Methods

Machine learning is a technique that allows computers to make decisions, similar to people's decision-making mechanisms [23]. With this technique, many problems such as classification and clustering can be solved besides decision making. Several algorithms specific to machine learning have been developed for this technique. Since the basis of algorithms is based on statistical science, algorithms are supported with many mathematical calculations. In this section, machine learning algorithms used for the thesis will be explained.

4.3.1. Support Vector Machines (SVM)

The Support Vector Machines (SVM) is a classification algorithm based on supervised learning that was first introduced by Vapnik [23]. Its basic logic is based on statistical learning theory. This algorithm is used to solve problems such as classification, regression and pattern recognition. The SVM algorithm can operate independently the distribution of data in space. Therefore, it can operate without the need for a combined distribution function for data.

Basically, Support Vector Machines are used to optimally separate data from two classes. This requires a boundary to separate the two classes. This boundary is called decision boundaries or hyperplanes.

The SVM algorithm works with kernel functions. The general representation of these functions is K (x, x_i). The results from this function are weighted with α Lagrange multipliers. This multiplier is used for weighting. After each core function is weighted with α Lagrange multipliers, the inner product is obtained. The internal product obtained as a result of the weighting is calculated for the whole network structure and then the sum is taken. This total value is the output value for an instance in the SVM network.

In the literature, SVM is expressed in three different types: linear SVM and nonlinear SVM.

4.3.1.1. Linear SVM

Linear SVM is one of the most basic types of SVM used in 2D classification problems. It is effective in determining the samples closest to the separation plane. Linear SVM aims to separate the data set with a decision line. This classification process is made with class labels (-1, +1). The determination of the most appropriate decision line depends on finding the most optimal one among the infinite linear decision lines. The change of the decision line depends on the data to be added to the solution space. However, it is appropriate that this line does not change. Therefore, the decision line must be as close as possible to the boundary line of the two classes.

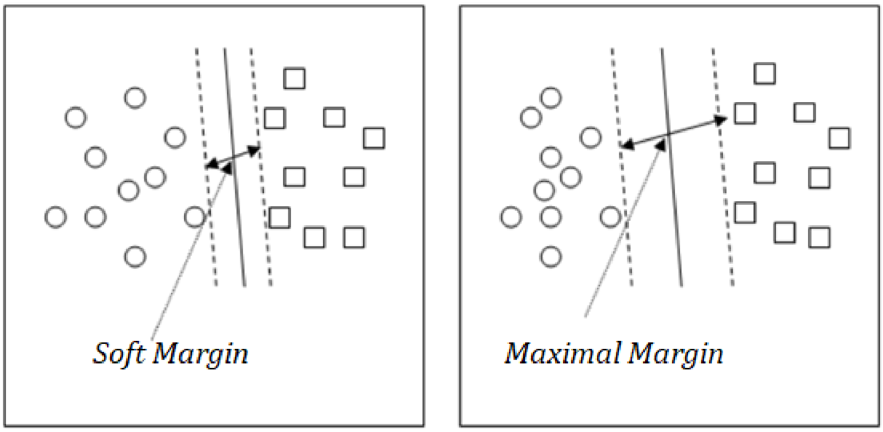


Figure 4.9. Margin types

Another concept in Linear SVM is margin. The margin is specified as the distance between parallel lines drawn equal to the decision line. The margin value is shown in Figure 4.9, in two types: maximal margin and soft margin. The maximum margin is to find the largest area in the plane that will separate the data. Therefore, it is more affected by noise for the data set. In the soft margin, a plane is determined to calculate the distance to be optimal.

$$f(x) = \sum_{i=0}^{n} w_i x_i + b \tag{2}$$

The correct equation used for linear SVM is as follows. In the equation, (2) represents the input data, b represents the threshold value and w represents the weight matrix. During the training of the model, the ones suitable for the vectors forming the weight matrix are calculated. According to this equation, SVM is similar to the Perceptron algorithm used in artificial neural networks. However, the kernel function in SVM makes the difference between them.

$$\exp\left(\frac{-\left||x-z|\right|^{6}}{\sigma^2}\right) : Exponential\ Kernel \tag{3}$$

$$\frac{1}{\sqrt{||x-x'||^2+c^2}} : Inverse\ Multiquadratic \tag{4}$$

$$-\sqrt{\left||x-x'|\right|^2 + c^2} \ : Multiquadratic \tag{5}$$

$$||x-x'||^{2n} \ln\left||x-x'|\right| : Thin\ Plate\ Spline \tag{6}$$

As shown in the kernel function (3), (4), (5), (6), there are different types, and the separation of data cannot be separated linearly using a plane.

$$y = sign(\left(w.\emptyset(x)\right) + b) \tag{7}$$

The equation shown in (7) forms the separation plane. This plane also depends on the weight matrix indicated by w. If the complexity of this matrix is low, a linear plane is formed. However, Quadratic optimization should be used for complex matrices.

4.3.1.2. Nonlinear SVM

Nonlinear SVM distinguishes between kernel functions and nonlinear classifiers (Figure 4.10). In this method, the model is developed with LibSVM library.

LibSVM is a ready-made library for SVM developed by Hsu and Chang, primarily used to solve 2-class problems [23]. Later, it was made to support multi-class problems. It provides a size increase for nonlinear classification with Kernel functions. In this way, multi-layer classification, cross-validation and dimensional property field transformation can be performed for the problem.

LibSVM also provides ease of use for parameters such as “linear, polynomial, radial basis function, sigmoid” used in SVM. Finding the optimal classification results for the data set to be used depends on the cross-validation process in LibSVM.

Additionally, The Kernel function creates a large Kernel matrix that cannot be stored in memory for large data input. The cross-validation in LibSVM can also increase the rate of classification to obtain an appropriately sized matrix.

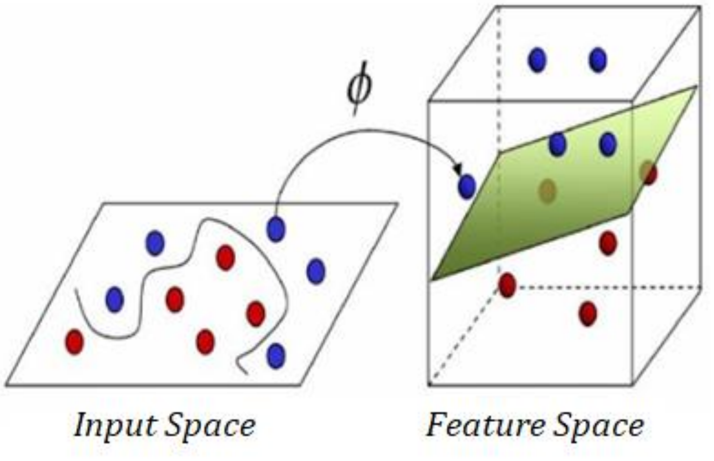


Figure 4.10. Classification of non-linear samples [23]

4.3.2. Random Forest

Random forest was developed in 2000 by Leo Brieman [87]. It is a multi-class classification algorithm based on supervised learning. Also, weather forecasting and object recognition can be solved using this algorithm. The use of more than one tree affects the classification result (See Figure 4.11). This is one of the differences with the decision tree algorithm. Another difference is the root node in the random forest algorithm. This algorithm works with nodes randomly. In this algorithm, trees of similar distribution are mapped to the vector of the randomly selected sample. The performance of the model increases when the number of trees corresponds to the data set.

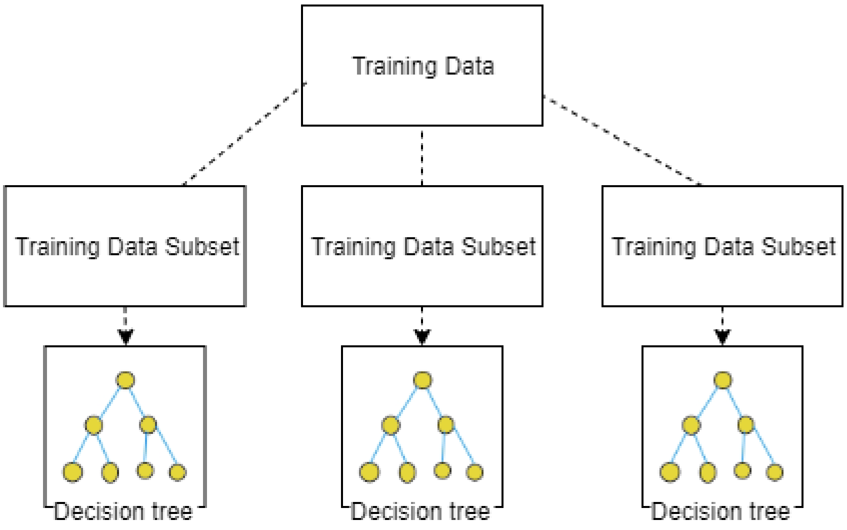


Figure 4.11. Generation tree using Random Forest

In the random forest algorithm, not only decision trees can be classified, but also regression problems can be solved. This algorithm can achieve success with both classification problems and regression problems because it can run different types of decision trees randomly. Also, for large data sets, it works more efficiently thanks to the independent variables and paralleling feature in missing data. Predictive performance is high not only for large data sets but also for low parameter problems.

Some of the operations performed in decision trees are not performed in this algorithm. One of these processes is a pruning. Pruning in decision trees is a method of eliminating overfitting. This method is to remove branches from the model that prevents the correct classification. However, in terms of classification performance, pruning did not show the expected effect. Therefore, pruning is not required for the random forest algorithm. Instead, predictors and each node determine randomly selected estimators to obtain the best prediction result.

Another difference is the bootstrap method. This method is used to randomly select trees in random forest. This method allows to make a random selection among the samples specified in the data set.

In this algorithm, there are basic steps to be followed for solving classification or regression problems. These were evaluated in three basic steps. In the first step, the data set to be used for modeling should be divided into two parts as training and test data. The most suitable compartment shape in terms of performance is 1/3 testing, 2/3 training. In the second step, the variables that can perform the best partitioning among the randomly selected samples should be determined. In the last step, the prediction results of the determined number of trees are collected. Thus, the best estimation results for the test data can be obtained.

The basic logic of the random forest algorithm is to divide the node into branches and determine which one of the randomly selected variables will work best. For this purpose, the CART algorithm, which operates according to the Gini index, is used to ensure that the data set is appropriately separated. The Gini index is a coefficient showing the homogeneity of the class. This value is calculated according to the equation shown in (8). The Gini coefficient specified in the equation is calculated according to the relative frequency of class D and p, j, j, and p relative to the relative probability of class j at node t, which contains samples from class n. Classes with small indexes are defined as homogeneous and large ones with heterogeneous class.

$$Gini\ (t) = 1 - \sum j[p(j\backslash t)]^2 \tag{8}$$

4.3.3. XGBoost

XGBoost algorithm was developed by Chen and Guestrin in 2016 [88]. This algorithm produces solutions for problems such as regression and classification. Therefore, the algorithm uses the CART algorithm. It separates the data by clustering the data. Moreover, XGBoost algorithm is designed by using algorithms that accelerate with decision trees. This algorithm have many parameters that regulate the calculation of dimensions and weights in decision trees. Accordingly, it aims to estimate the best test results. However, it is important to get the best parameter selections in case of overfitting and underfitting. Compared with other machine learning algorithms, it provides the convenient training model developed to the data set and produces test results giving appropriate estimation.

It is based on the Extreme Gradient Boosting Trees algorithm. The most primary version of this algorithm is the boosting algorithm. Therefore, this algorithm must be explained initially.

The Boosting algorithm was first introduced in 1990 by Schapire [89]. In this algorithm, the estimators follow a sequence in establishing the learning model. By creating strong estimators from weak estimators, new weights are determined and a new model is formed from these weights.

Boosting algorithm is to develop the previous model and then the training stage, as shown in Figure 4.12. Before the development of the new model, the wrong outputs of that model are examined. The development of the new model is made through these errors and the training model is created in turn. According to the data set, it is important that parameters such as number of stages in education and learning rates of sub-models are adjusted correctly.

Different algorithms such as AdaBoost and Gradient boosting are used for the boosting method. In fact, the basis of these algorithms is boosting, and the XGBoost algorithm is in this group. With the development of algorithms, new methods that will reduce the error rate have been added to the algorithms and become more efficient algorithms. XGBoost algorithm is based on Gradient boosting in terms of structure. Therefore, it would be useful to explain Gradient boosting. This algorithm was developed in 2001 by Friedman [90]. The working principle of this algorithm allows to predict errors in steps minimize the error that occurs after the development of the model.

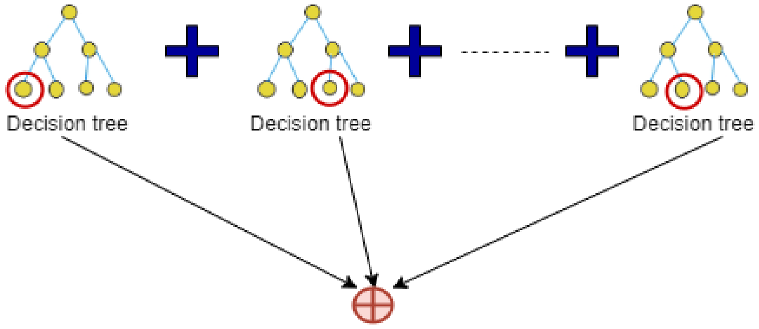


Figure 4.12. Steps of Boosting Method

For this reason, it tries to minimize the errors caused by the Gradient boosting model with optimization. For the optimization of errors, the calculation is made using the mean squared error function shown in (9) and used in decision trees. In this equation, y_i, i is the target value; y_{ip} the estimated value y_{ip} symbolizes.

$$Error = MSE = \sum(y_i - y_{ip})^2 \qquad (9)$$

In Gradient boosting algorithm, two methods called column sample and subsample are used to prevent overfitting. The first of these methods, column sample, creates trees that take values between zero and one according to the randomly selected variable ratio. In the second method, it is aimed to generate a sub-data set and trees are created according to randomly selected observations.

In the learning phase of the algorithm, the first weights are assigned equal value after the data set is separated into training and test data. Then the training set is used in the modeling stage and the results are calculated. In the establishment of the new model, the weights of the wrong classes obtained from the results are determined again. At this stage, it is generally recommended to increase the value for the wrong classes and to decrease the value for the lines when calculating the weights. After that, new results are obtained by remodeling.

4.4. Evaluation Criteria

Several evaluation criteria are needed to determine the success of models created using machine learning methods. Then, many evaluation criteria are used. The first of these methods is the confusion matrix. The purpose of this method is to compare the predictions obtained from the modeling results with the actual values. Classification accuracy is calculated from the comparison of the two values. The structure of this method is as given in Table 4.1.

The complexity matrix consists of four different parameters FP, FN, TP and TN in Table 4.1. To calculate these parameters, the actual values obtained from the data set are compared with the estimated values. Accordingly, TP is valid for calculations with positive and predictive values. For example, it is in this group to actually possible to estimate a phishing web page as a phishing web page. Similarly, the actual TN value is negative and the classification result is negative.

Table 4.1. Confusion matrix

	Predicted Negative	Predicted Positive
Actual Negative	True Negative (TN)	False Positive (FP)
Actual Positive	False Negative (FN)	True Positive (TP)

For example, it is in this group again actually possible to predict a legitimate web page as a legitimate web page. FP and FN are used to show that the prediction result is incorrect. FP represents a positive but negative estimate in real data; FN is the exact opposite. In other words, it is provided for data whose actual value is negative but after estimation it is positive. For example, it is actually the FP to predict a legitimate web page as a phishing web page. Also FN is actually phishing a web page as a legitimate web page to prediction [91].

Accuracy is the ratio of the total negative and positive observation results predicted by the classification result to the correct estimate results (10). Performance measurement is performed intuitively.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \tag{10}$$

Precision (π) is the calculation with real values (11). It is the ratio of positive values to the sum of positive and negative values.

$$\pi = \frac{TP}{TP+FP} \tag{11}$$

Recall (ρ) is correctly classified positive sample (TP) number, total positive sample count (TP + FN) ratio and it is also called TPR (12). (13) shows the FPR rate, which is used in the rate calculation of negative estimates.

$$\rho = TPR = \frac{TP}{TP+FN} \tag{12}$$

$$FPR = \frac{FP}{FP+TN} \tag{13}$$

F_1 score is the harmonic mean of precision and recall values (14). Since these two criteria are not sufficient for evaluation in some cases, F_1 score is calculated with these two criteria.

$$F_1 = \frac{2TP}{2TP+FP+FN} \tag{14}$$

ROC (Receiver Operating Characteristic) curve is defined as the precision ratio of recall according to the classification result of the model. The area below the ROC curve, which is referred to as the curve, is examined in order to determine the reliability of the system. In the interpretation stage of this curve, the positive ratio on the vertical axis is high; the false positive rate on the horizontal axis should be low for the ROC curve.

4.5. Tools

In this section, various tools and libraries used within the scope of the thesis are briefly introduced.

4.5.1. OpenCV

The OpenCV (Open Computer Vision) library is an open source class library. It is licensed under BSD license. It contains many general and specialized image processing and computer vision algorithms [92].

OpenCV is free except for some modules. Also, it can be used for academic and commercial purposes. It is basically prepared in C / C ++. Also, multi-core processors support hardware graphics accelerators. It includes many features from interactive art to mapping on web pages. Also, OpenCV is an image processing library hosting a community member.

4.5.2. Python

The Python programming language was written in the 1990s by Guido Van Rossum, a Dutch programmer [93]. One of the important features of this language is that it can be used in scientific studies and can process quickly.

Python is also compatible with programmable cards. Python is a programming language that is used in many areas such as web application or web site development, data collection and analysis, system management, machine learning. Some important features of Python include:

1. Python is a programming language that enables fast program writing and efficient integration into embedded systems. For this reason, it is used by many companies.
2. Python can run on Windows, Linux / Unix and Mac-OS
3. It is also integrated into Java and .NET virtual machines.
4. Python is open source software.
5. Applications written with more than one code in C ++ and C # languages can be written in Python languages on a single line.
6. It has large and functional libraries.
7. It has a modular structure.
8. Python is a programming language that supports multiple programming paradigms, such as object-oriented programming, functional or structured programming.
9. Python is preferred in many security-conscious applications. Most socket applications are developed in Python.

4.5.3. Pyleargist

The Pyleargist library is a library developed for GIST descriptor. The C source code of the descriptor developed by A. Torralba [77] is used with this library. An example of code written for this library is shown in Figure 4.13. The result is a 960-dimensional feature vector. Also, it is compatible for Python 2 and Python 3.

```
import leargist
from PIL import Image
im = Image.open(file)
descriptors = leargist.color_gist(im)
```

Figure 4.13. Steps of Boosting Method

4.5.4. Sklearn

The SkLearn [94] library, also called Scikit-Learn, was developed to use machine learning in python and is an open source. Generally, it is utilized in the sub-branches of artificial intelligence such as data mining, data analysis and machine learning. Moreover, classification, regression, clustering, dimensionality reduction, model selection and preprocessing operations can be performed by using fit / perform functions. Additional packages are also available for the classification, regression, clustering, dimensionality reduction, model selection and preprocessing required for these sub-branches. It can be recommended to use with the SciPy and NumPy libraries called scientific and numerical. Appendix 1 shows the Application Programming Interface (API) types and tasks in the Sklearn library.

5. APPROACH

The general scheme of the application architecture proposed in the thesis is shown in Figure 5.1. The system consists of 4 phases consisting of data, feature extraction and image representation, machine learning and validation phases. There is a certain flow pattern between the phases and the results obtained from one phase affect the other phase. In general, the business logic of the system is configured in this way. The model version of this architecture is shown in Figure 5.1.

5.1. Data Phase

This phase is used with the dataset required for the system. Since train and test data are automatically separated in dataset, no additional process is performed at this stage. Phish-IRIS data set was used as data set. The explanations of dataset are explained in section x.

5.2. Feature Extraction and Image Representation Phase

In this phase, feature is extracted for each image data in the data set. Therefore, visual descriptors are used. In the literature, these can be examined under different groups, but within the scope of this thesis, it is divided into local and global. This classification was taken into consideration during the modeling phase. SIFT, DAISY, GIST and LBP were used. Features were created separately from each. Furthermore, this phase uses the spatial multi-level patch pyramid approach to capture a greater number of visual features. Thus, the loss of information is minimized. This approach was introduced by Lazebnik et al. Also the bag of visual words approach is in this phase.

Different descriptors consisting of 50, 100, 200 and 400 visual words were selected in the section where local descriptors were used. The classification results in machine learning can be evaluated in this way. In addition, false positive rate (FPR), true positive rate (TPR) and accuracy can be measured according to the increase in the number of visual words. Experiments designed for the model were performed in Python 3. However, the “OpenCV” library and the “SkImage” library were used to encode these descriptors. A 3-step method is designed for local descriptors. In the first step of this method, K-means based clustering is performed.

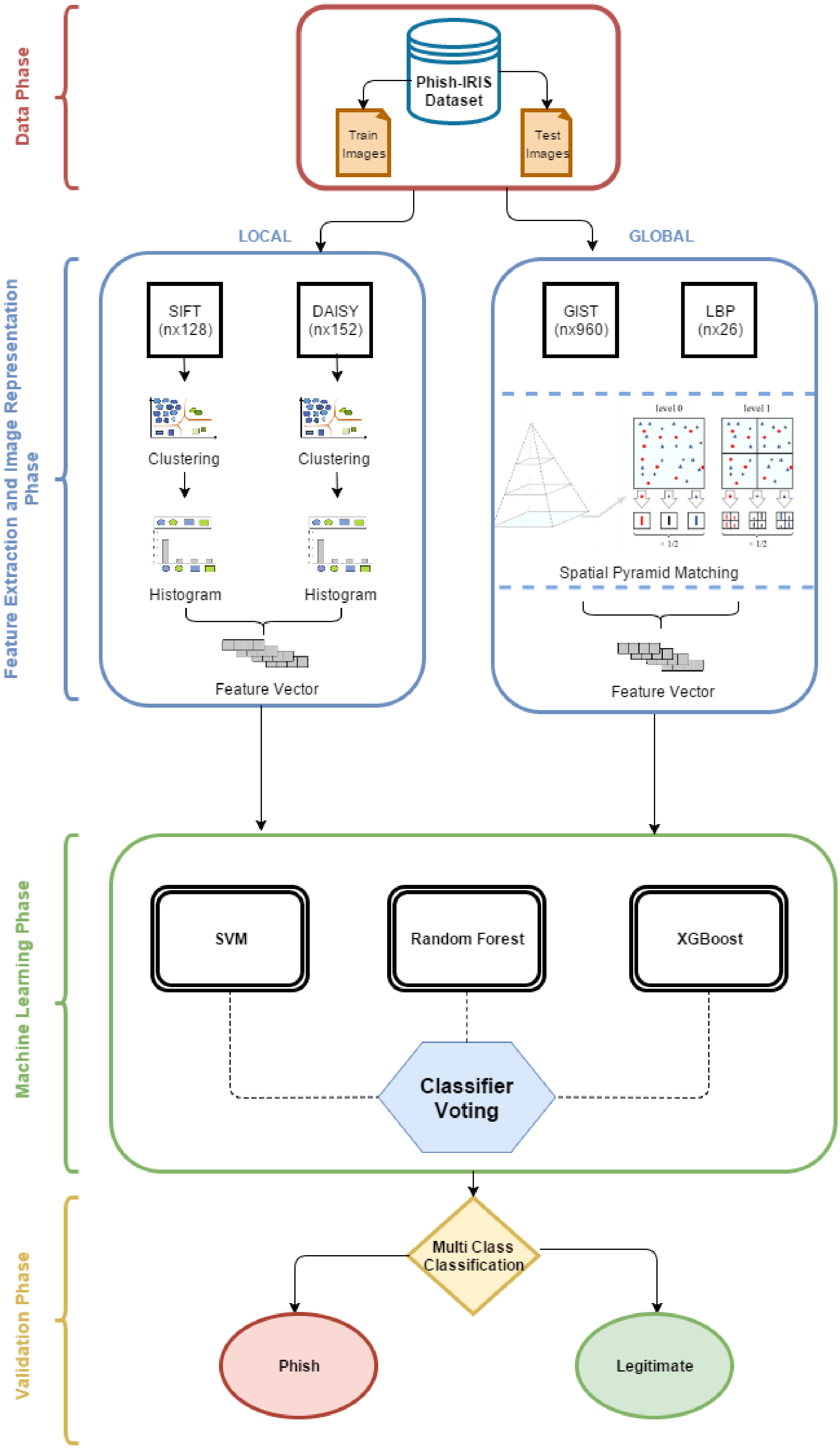


Figure 5.1. Model architecture

In the second step, the code book is generated. In the last step, feature vectors are created. Default parameters are used in feature extraction based on SIFT. However, for DAISY, the radius was determined to be 16, the number of rings 3, the number of histograms 6, and the number of directions 8, which resulted in significant changes in performance.

According to these results, the decrease in the number of rings has been determined as 3 because it affects performance negatively. In addition, since the data set is quite large, a number of memory problems have emerged in classical “K-Means” clustering. In order to solve this, "Mini-Batch KMeans" algorithm is used which contains small random data sets which can be stored in memory. The “k-means ++” parameter was used to increase the clustering speed and find the centroids.

In the second part of the model, global descriptors were used. Firstly, by use of GIST descriptor, visual feature extraction was performed. 960 dimensional feature vectors by extracting GIST descriptors in a holistic manner. Second, we have applied multi-level patching in order to produce finer detailed image descriptors that will eventually build single concatenated and larger “multi-level” feature representation. At this stage, each image has been recursively divided into 2 × 2 = 4 and 3 × 3 = 9 equal parts. Hence, for the multi-level representation, we have processed either 1 + 4 = 5 or 1 + 4 + 9 = 13 patches in total.

As a result, we have eventually generated descriptions of screenshots by employing either single or multi-level pyramidal like scheme. For GIST features, we have used “pyleargist” library developed for Python programming language. The same procedure has also been followed during the phase of LBP description generation. Each screen image data in the data set was firstly created as a single feature vector. In the next step, screenshots were divided into 4 parts and LBP descriptor was applied for each part. In the last step, 9 identical parts were distinguished and these 5 and 13 part vectors were created just as in GIST.

5.3. Machine Learning Phase

Sklearn, Numpy, XGB libraries were used in the classification methods based on machine learning. In addition, the machine has been carried out on several Ubuntu platforms by employing several Python libraries such as Scipy, Numpy, Matplotlib, Pandas and Sklearn. Following the feature vector generation, we have built three classification models including Random Forest, Support Vector Machine and Xgboost methods for predicting the class of screenshot samples in test group.

In the first experiments for SVM, the kernel tuning parameter was set to default, but was defined as radial basis function to improve performance. Other classification methods have better performance with default values. The XGBoost classifier works faster and more efficiently because it has CUDA-based GPU support.

In the last step of this phase, it is ensured that the best results are obtained from all classifiers. For this purpose, classifier voting library was used. With this approach, the best classification result is calculated.

5.4. Validation Phase

In this phase, the model is evaluated according to the data set. The Phish-IRIS dataset includes both phished brand classes and legitimate classes. This class is named "other" in the dataset. Therefore, the model is considered to be the brand name of a suspicious web page as phishing or legitimate. In addition, interpretation is performed according to evaluation criteria which are train accuracy, test accuracy, TPR, FPR and F1 score.

6. EXPERIMENTS AND RESULTS

The evaluation was carried out with a dataset containing the page snapshots of the original web pages with 14 different brands. This dataset includes a total of 1313 training and 1539 test samples. For these methods, python is used in UBUNTU operating system.

6.1. Dataset

There are many intrusion detection systems that can detect phishing attacks. However, these systems are usually run on existing dataset or with unreliable dataset. These datasets were mostly created with data collected from PhishTank, MillerSmiles and Google search operators. The datasets required for almost every study are obtained by this method. Particularly in studies based on vision-based techniques for phishing attack detection, there are difficulties in accessing an appropriate open dataset.

The Phish-Iris data set used in this thesis provides the vision based solution of phishing attacks. In addition, thanks to 15 different brand classes, multi-class phishing protection work is done. The 14 classes in the 15 brand classes include different brands of phishing. The latest class correspond to "unknown" or "legitimate" examples. In this data set, the data is divided into train and test folders. There are 1313 training and 1539 test samples. Legal and phishing samples are included in this data set. The samples labeled "other" in this dataset belong to legitimate sites. In the section marked with this label, there are mixed screen shots of legitimate web pages serving in different areas. Developers using this dataset can perform a wide range of testing steps after train with phishing pages of 14 different brands. It may not be within the legitimate pages label "other" of the brands used for the train. The dataset makers have collected screenshots of more and more different brands of web pages on this label because they have introduced the dataset as "open dataset". In this way, they aimed to develop and test models that distinguish real-life phishing pages and legitimate pages.

In addition, the data set includes screenshots of web pages collected from various platforms. Therefore, the image sizes in the form of screenshots are different from each other. If necessary, it is important to adjust these image sizes.

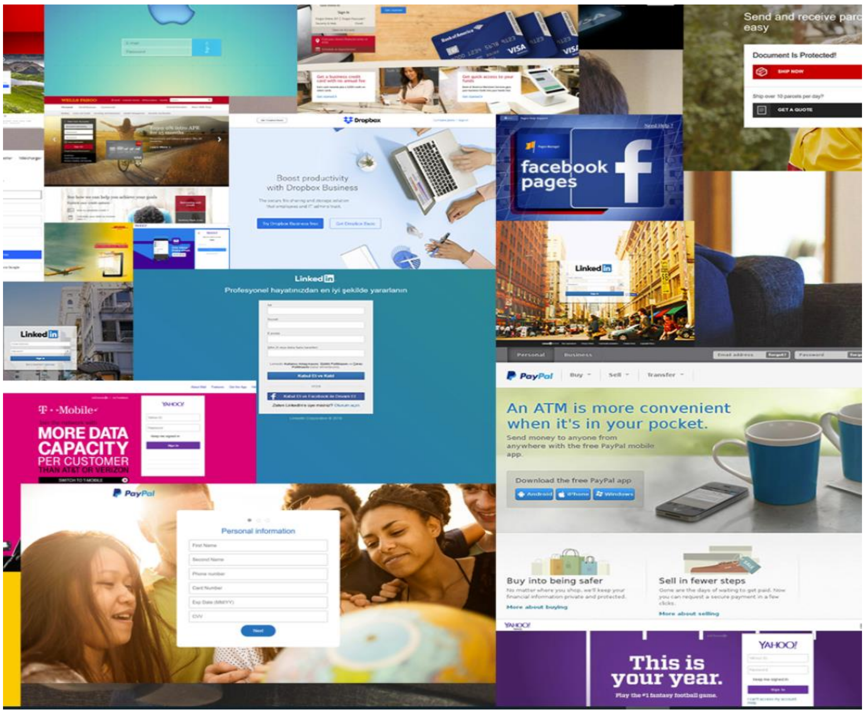

facebook pages
Linked in
T-Mobile
MORE DATA CAPACITY PER CUSTOMER
PayPal
Personal information
An ATM is more convenient when it's in your pocket.
Boost productivity with Dropbox Business
Buy into being safer
Sell in fewer steps
YAHOO!
This is your year.

Figure 6.1. Example of the Phish-IRIS dataset

In the thesis, phishing attacks are detected according to the screenshots, so the data set should be suitable for this purpose. The data sets in the literature were examined and the data set “Phish-IRIS” [75] was chosen because it is up-to-date and contains screenshots of the web pages of 14 different brands. It consists of a total of 2852 samples including training and test samples, and the distribution table by brands is shown in Table 6.1.

Note that, “Phish- IRIS” dataset is a publicly and free available dataset for academic purposes and it can be downloaded from the URL of https://web.cs.hacettepe.edu.tr/~selman/phish-iris-dataset/. According to the definitions of the dataset creators, “Phish- IRIS” dataset has been collected between the March-May 2018. Examples of the data set are shown in Figure 6.1.

Table 6.1. Phish-IRIS Dataset

Brand Name	Training Instances	Testing Instance
Adobe	43	27
Alibaba	50	26
Amazon	18	11
Apple	49	15
Bank of America	81	35
Chase Bank	74	37
Dhl	67	42
Dropbox	75	40
Facebook	87	57
Linkedin	24	14
Microsoft	65	53
Paypal	121	93
Wellsfarno	89	45
Yahoo	70	44
Other (i.e. Legitimate)	400	1000
Total	1313	1539

6.2. Experiments

In this part of the thesis, phishing web pages are classified according to their brand names. In this section, data preprocessing, model creation and experiments will be explained respectively.

After constructing the visual vectors with "holistic" and "multi-level-patch" approaches, the classes of the test data are estimated. For this reason, modeling was performed using Random Forest, Support Vector Machine and XGBoost, one of the machine learning algorithms. The computer is equipped with an Intel® Core ™ i7 4700HQ processor and 16 GB of memory. In the Ubuntu platform, several Python libraries such as Scunty, Numpy, Matplotlib, Pandas and Sklearn have been modeled. Experiments with local descriptors used different code book numbers and other parameters. Experimental results are given in the following sections.

In the first step in the experiments using GIST descriptor, visual features were extracted with holistic approach. The experiments in this descriptor were written in python and the "pyleargist" library was used. The result is a 960 dimensional feature vector for a screen display. After obtaining these vectors for each image, the second step in the experiment was initiated. In this step, the "multi-level-patch" approach is used, after dividing that image into equal parts, it processes each part in the descriptor. In the experiments, each image was divided into 4 and 9 parts. In section 6.2.1, the multi-level indication is expressed as 5 (1 + 4) and 13 (1 + 4 + 9). As a result, a holistic and multi-level structure has been established in order to fit the pyramid structure. Then, it is processed with GIST descriptor. The same approach was applied before processing images in the LBP descriptor.

6.2.1. Global Descriptor Based Analysis

In this section, screenshots in the data set are included in the experiment stage as a single piece. Before applying the machine learning model, the whole data set is feature extraction with local and global descriptors. Experiments were carried out with Random Forest, SVM and XGBoost, machine learning algorithms. TPR, FPR, F1 score and accuracy were calculated during the evaluation phase. The results for the holistic model are shown in Table 6.2. When GIST and LBP based results were compared, higher accuracy was obtained with GIST features and this result was 85.83%. This result was obtained with XGBoost learner. Although it is considered uncommon, more visual words were extracted with GIST compared to LBP. Therefore, increasing the number of visual words may be effective in increasing the accuracy of classification. However, when SVM, Random Forest and XGBoost were compared, it was observed that higher accuracy rate was obtained with XGBoost. Also, the best results were obtained in terms of TPR.

Table 6.2. Holistic Results of GIST and LBP

Descriptor	Algorithm	Train acc	Test acc	TPR	FPR	F1
GIST	SVM	0.533	0.746	0.7465	0.018	0.75
GIST	XGBOOST	0.732	0.8583	0.8583	0.010	0.86
GIST	RANDOM FOREST	0.740	0.860	0.860	0.009	0.86
LBP	SVM	0.272	0.629	0.629	0.0266	0.63
LBP	XGBOOST	0.602	0.751	0.751	0.0177	0.75
LBP	RANDOM FOREST	0.631	0.784	0.784	0.015	0.78

Secondly, the experiment was carried out with LBP descriptor. In the first step, holistic manner, 26 dimensional visual feature was created with LBP. Using the random forest, the test accuracy rate was 78.4%.

Table 6.3. Voting Classifier Results of GIST and LBP

Descriptor	Train acc	Test acc	TPR	FPR	F1
GIST	0.738	0.8596	0.8596	0.010	0.86
LBP	0.592	0.7647	0.7647	0.0168	0.76

As shown in Table 6.2, only the results of the analyzes performed with LBP yielded the best classification with Random Forest learner. Better results were obtained in GIST based analysis than LBP.

Voting classifier was also used for experiments using global descriptors. This analysis is a function in Python in the "Sklearn" library. This function allows the classifiers in machine learning to decide together to form a common classification result. Thus, the wrong decision of a classifier can be balanced with other classifier results. There is an increase in the results according to Table 6.3. The calculation was made according to the common decision of all classifiers.

When LBP and GIST were compared in terms of runtime, it took approximately 1.5 seconds to obtain classification results after feature extraction. Unlike Random Forest and SVM, the XGboost algorithm runs on the CPU in-place GPU. Therefore, XGBoost algorithm gives faster results.

6.2.2.1 Spatial Multi-Level-Patch Based Analysis

Spatial multi-level-patch pyramid configuration approach is the second stage of experiments using GIST and LBP descriptors, as shown in Table 6.4 and Table 6.5. For this reason, first, a 3-level pyramid design was made. This pyramid consists of the "1", "1 + 4" and "1 + 4 + 9" configuration. Of these, "1" refers to the whole screenshot. "1 + 4" symbolizes the whole screenshot and the sum of the images divided into 4 equal parts. That is, it is combined with a total of 5 different descriptors. This configuration is called a 2-level pyramid. When looking at the classification results calculated using GIST, first of all, visual features of 960 * 5 = 4800 were obtained. Then, SVM, XGBoost and Random Forest were classified and the best results for accuracy, TPR, FPR and F1 were calculated using XGBoost. The Accuracy ratio was calculated as 87.19%.

When the experimental results for LBP were examined, it was observed that a feature vector of 26 * 5 = 130 size was initially formed. The results of the classification algorithms produced the highest accuracy rate of 82.7% by Random Forest learner. The XGBoost is a bit behind in terms of accuracy.

In the third part of the experiment, the pyramid structure was established with the "1 + 4 + 9" configuration. In this configuration, "1" represents the whole screen image, "4" represents an image divided into four equal parts, and "9" represents the nine-part version of an image. In other words, a total of 13 different descriptor structures were formed and the pyramid layout was formed. This pyramid is called 3-level.

The first test in the 3-level pyramid was made using GIST, which consisted of 960 * 13 = 13440 feature vectors in total. The best test accuracy was calculated as 87.71% with XGBoost learner.

Table 6.4. Multi-Level-Patch Pyramid GIST based analysis

Descriptor	Algorithm	#Patches	#Features	Train acc	Test acc	TPR	FPR	F1
GIST	SVM	1+4	4800	0.549	0.757	0.757	0.017	0.76
GIST	XGBOOST	1+4	4800	0.757	0.8719	0.8719	0.0091	0.87
GIST	RANDOM FOREST	1+4	4800	0.755	0.858	0.858	0.01	0.86
GIST	SVM	1+4+9	13440	0.568	0.7868	0.786	0.01	0.79
GIST	XGBOOST	1+4+9	13440	0.779	**0.8771**	0.8771	0.0084	0.88
GIST	RANDOM FOREST	1+4+9	13440	0.768	0.8739	0.8739	0.009	0.87

In the second experiment, LBP was used and 26 * 13 = 364 dimensional feature vectors were obtained. The best result was calculated as 83.1% with XGB.

According to the multi-level configuration results of GIST and LBP, XBG-based machine learning method performed better. Moreover, using this machine learning method, GIST has yielded more successful results. The 3-level pyramid configuration, which is "1 + 4 + 9", has improved the accuracy performance of both descriptors, but in terms of performance, the LBP remains behind GIST.

In the 2-level pyramid configuration scheme, the best accuracy was obtained from GIST. The best accuracy for LBP was calculated from the Random Forest classifier. The lowest FPR was also obtained from these classifiers.

Table 6.5. Multi-Level-Patch Pyramid LBP based analysis

Descriptor	Algorithm	#Patches	#Features	Train acc	Test acc	TPR	FPR	F1
LBP	SVM	1+4	130	0.338	0.638	0.638	0.025	0.64
LBP	XGBOOST	1+4	130	0.687	0.798	0.798	0.0143	0.8
LBP	RANDOM FOREST	1+4	130	0.711	0.827	0.827	0.012	0.83
LBP	SVM	1+4+9	364	0.372	0.6621	0.662	0.024	0.66
LBP	XGBOOST	1+4+9	364	0.732	**0.831**	0.831	0.12	0.83
LBP	RANDOM FOREST	1+4+9	364	0.733	0.825	0.825	0.00124	0.83

It has been observed in Table 6.4 and Table 6.5 that GIST gives higher results than the results of multi-level patch approach in global descriptors. In terms of LBP, performance improvement was achieved in a positive way compared to holistic based analyzes. Since the feature vector is large in GIST-based analysis, training time was longer than LBP. For GIST, this time is about 1.2 seconds in a single image.

Table 6.6. Combined Results of GIST and LBP based analysis

Descriptor	Algorithm	#Patches	#Features	Train acc	Test acc	TPR	FPR	F1
GIST+LBP	SVM	1+4+9	13805	0.569	0.7836	0.7836	0.015	0.75
GIST+LBP	XGBOOST	1+4+9	13805	0.818	**0.89018**	0.8901	0.0078	0.88
GIST+LBP	RANDOM FOREST	1+4+9	13805	0.784	0.87329	0.8732	0.0090	0.86

In order to increase the success rate of global descriptor experiments, it is considered to use these descriptors together. For this purpose, descriptor setting is used. In other words, the best performance for GIST was obtained from the "1 + 4 + 9" structure.

Similarly, the best performance for LBP was obtained from the "1 + 4 + 9" structure. Therefore, this structure of GIST and LBP has been used together. A separate combined script written in Python combined features that produced two descriptors and new features were used in classifiers. As in Table 6.6, according to the results of this experiment, 89.018% success was achieved.

Table 6.7. Results of voting classifier analysis

Descriptor	#Patches	#Features	Train acc	Test acc	TPR	FPR	F1
GIST	1+4	4800	0.764	0.8641	0.86419	0.0097	0.86
GIST	1+4+9	13440	0.772	0.8739	0.87394	0.009	0.87
LBP	1+4	364	0.688	0.8206	0.82066	0.0128	0.82
LBP	1+4+9	364	0.719	0.8304	0.8304	0.0121	0.83
GIST+LBP	1+4+9	13805	0.800	**0.880**	0.8804	0.0085	0.88

In addition, voting classifier-based analysis was performed in the experiments as shown in Table 6.7. According to these results, the best results were obtained from combined GIST and LBP descriptors which is 88.0 %.

6.2.2.2. Comparative Study- HOG Based Analysis

In this section, the results of experiments with global descriptors are compared using HOG descriptor. Experiments with this descriptor have been performed in the literature before, but it is used for the first time with the data set in the thesis. The effectiveness and validity of the proposed method can be measured in this way.

By definition, the HOG descriptor extracts the characteristics of the input image according to the corner edge property. The division of this image into cells, which is called the detection window, is done in the first step. It then finds the normalization of each cell and gradient directions are calculated. In the last step, the histogram is created.

Table 6.8. Prediction results with HOG descriptors

Descriptor – Cell Size – Mode	Learner	Train Acc.	Test Acc.	TPR	FPR	F1
HOG – 32px cells – Cropped	XGB	0.714	0.8349	0.834	0.011	0.82
HOG – 32px cells – Cropped	RF	0.693	0.8258	0.825	0.012	0.81
HOG – 32px cells – Cropped	SVM	0.596	0.7543	0.754	0.017	0.73
HOG – 32px cells – Resized	XGB	0.719	**0.8408**	0.840	0.011	0.83
HOG – 32px cells – Resized	RF	0.71	0.8395	0.830	0.011	0.82
HOG – 32px cells – Resized	SVM	0.626	0.7673	0.767	0.016	0.75
HOG – 64px cells – Cropped	XGB	0.729	0.8245	0.824	0.012	0.81
HOG – 64px cells – Cropped	RF	0.705	0.8317	0.831	0.012	0.82
HOG – 64px cells – Cropped	SVM	0.579	0.74	0.74	0.018	0.72
HOG – 64px cells – Resized	XGB	0.747	0.8304	0.830	0.012	0.82
HOG – 64px cells – Resized	RF	0.722	0.8369	0.836	0.011	0.82
HOG – 64px cells – Resized	SVM	0.597	0.7563	0.756	0.017	0.74

The same data set and classifiers were used to compare the global descriptors HOG descriptor. The canonical resolution of this descriptor must be set in the property extraction. Therefore, resizing and cropping of the screen image is required. However, there are a number of drawbacks to these two methods: first, the breaking process leads to loss of information. Second, the edge structure may be distorted when an image is resized. Therefore, experiments were performed with different cell sizes in which 32 and 64 pixels were adjusted. Detailed results are given in Table 6.8.

According to the results, HOG features achieve 84.08% accuracy at best configuration. Experimental study reveals that Random Forest and XGBoost produce slightly similar results. Nevertheless, SVM (RBF kernel) has been clearly outperformed by Random Forest and XGBoost learners. Compared to the best model created with HOG features, GIST based analysis is superior to HOG and LBP.

6.2.2. Local Descriptor Based Analysis

This section includes experiments creating visual words with local descriptors. In the experiments of the thesis, visual words for SIFT and DAISY were generate with BoVW representation. Then, classification was made with machine learning algorithms. As in other experiments, SVM, XGB and Random Forest learners were used. These classification algorithms and descriptors are examined in the experimental results. Accuracy, TPR, FPR, F1 score measurements are calculated for training and test data. Results of the evaluation were given in Table 6.9 and Table 6.10 below.

In the first step, experiments were performed with 50 visual words. SIFT was observed to give a higher accuracy rate than DAISY with random forest learner. In the second step, SIFT worked better with 100 visual words. It is the third experiment to produce 200 visual words and has more performance than SIFT.

The final experiment was carried out with 400 visual words and the highest accuracy rate was obtained in all experiments. This ratio was calculated from SIFT with 89.34%. In addition, the XGB learner was effective in this success. Comparing SIFT and DAISY, the highest accuracy was obtained from SIFT and this ratio was achieved thanks to the 400-D codebook size. Although higher results were obtained with XGB, the accuracy rate was increased with Random Forest in DAISY.

Table 6.9. Results of SIFT based analysis

Descriptor	Algorithm	#Features	Train acc	Test acc	TPR	FPR	F1
SIFT	SVM	50	0.611	0.7732	0.7732	0.016	0.77
SIFT	XGBOOST	50	0.725	0.8187	0.8187	0.012	0.82
SIFT	RANDOM FOREST	50	0.729	0.842	0.842	0.112	0.84
SIFT	SVM	100	0.674	0.803	0.803	0.014	0.80
SIFT	XGBOOST	100	0.762	0.846	0.8466	0.010	0.85
SIFT	RANDOM FOREST	100	0.749	0.860	0.860	0.0099	0.86
SIFT	SVM	200	0.747	0.837	0.837	0.011	0.84
SIFT	XGBOOST	200	0.799	0.8589	0.8589	0.01	0.86
SIFT	RANDOM FOREST	200	0.774	0.8823	0.8823	0.0084	0.88
SIFT	SVM	400	0.821	0.8758	0.875	0.008	0.88
SIFT	XGBOOST	400	0.827	**0.8934**	0.893	0.0076	0.89
SIFT	RANDOM FOREST	400	0.8	0.8875	0.8875	0.0080	0.89

According to the experimental results tables using local descriptors, the common interpretation of both descriptors was the increase in the accuracy rate with the increase of visual words. Therefore, the use of a code book positively affected performance. In addition, SIFT-based modeling was more successful than DAISY. The calculation of the rational variance of the DAISY descriptor was not appropriate for the data set used in the thesis, so DAISY's performance was worse.

Table 6.10. Results of DAISY based analysis

Descriptor	Algorithm	#Features	Train acc	Test acc	TPR	FPR	F1
DAISY	SVM	50	0,648	0,7465	0,746	0,018	0,74
DAISY	XGBOOST	50	0,678	0,7849	0,784	0,015	0,78
DAISY	RANDOM FOREST	50	0,699	0,816	0,816	0,013	0,8
DAISY	SVM	100	0,709	0,7758	0,775	0,016	0,77
DAISY	XGBOOST	100	0,709	0,7953	0,795	0,014	0,79
DAISY	RANDOM FOREST	100	0,715	0,8226	0,822	0,012	0,81
DAISY	SVM	200	0,725	0,7901	0,79	0,014	0,79
DAISY	XGBOOST	200	0,722	0,8174	0,817	0,013	0,81
DAISY	RANDOM FOREST	200	0,719	0,831	0,831	0,012	0,82
DAISY	SVM	400	0,725	0,818	0,818	0,818	0,81
DAISY	XGBOOST	400	0,725	0,8122	0,812	0,013	0,8
DAISY	RANDOM FOREST	400	0,716	**0,8356**	0,835	0,011	0,82

In addition, when the runtime of both descriptors was compared, SIFT was observed to run faster than DAISY. SIFT worked on average 1.5 seconds, while DAISY worked 2.18 seconds. Therefore, SIFT may be considered more suitable for real-time applications.

Table 6.11. Results of voting classifier analysis

Descriptor	#Features	Train acc	Test acc	TPR	FPR	F1
SIFT	50	0.725	0.8343	0.8343	0.011	0.83
DAISY	50	0.703	0.8102	0.81026	0.0135	0.81
SIFT	100	0.763	0.8564	0.8564	0.0102	0.86
DAISY	100	0.725	0.8135	0.81351	0.0133	0.81
SIFT	200	0.803	0.8797	0.87979	0.0085	0.88
DAISY	200	0.737	0.8323	0.83235	0.0119	0.83
SIFT	400	0.845	**0.9038**	0.9038	0.0068	0.90
DAISY	400	0.735	0.8382	0.8382	0.0115	0.84
SIFT+ DAISY	400	0.813	0.87069	0.8706	0.0092	0.87

In Table 6.11, the voting classifier was applied to local descriptors. According to these results, the accuracy success obtained from SIFT was % 90.38. However, the results of the experiments using combined SIFT and DAISY feature vectors reached 87.06% accuracy.

7. DISCUSSION

In this thesis, phishing pages were determined by using global and local descriptors. According to the results of the experiment, it can be said that the study is a promising approach. Experimental results are based on comparison of global and local results. The best results were obtained by using SIFT in local descriptor. In addition, spatial pyramid matching approach in global descriptors gave better results than holistic based approach. That is, giving an image to descriptors in small pieces is more effective in finding out if a page is phishing or not. Likewise, best results can be explained by SIFT in local descriptors.

When GIST and LBP based holistic representation analyzes were compared, it was observed that the performance of the LBP based models was lower than that of GIST. This finding is thought to have arisen by defining the orientation, contour and smoothness information of the image with GIST filter set in different directions and dimensions. However, in LBP, texture information in images is determined as a relative gray level. This led GIST to produce effective and appropriate features. Furthermore, in GIST, image pro-processing is performed before processing images in Gabor filters. As a result, GIST was more effective in extracting important and relevant keypoints in the problem in the thesis than in the LBP for holistic representation.

As a result based on local descriptor, the accuracy rate increased when the number of visual words increased in both descriptors. So the larger codebook had a positive effect on performance. Furthermore, according to the experimental results, DAISY-based models performed less performance than SIFT, because DAISY exemplifies irrelevant keypoints. However, SIFT uses the DoG technique to find points that may be relevant. Therefore, performance differences occur between two descriptors. In addition, DAISY mostly uses rotational variance, which is not suitable for the dataset used in the thesis. So this observation is not important for phishing web pages.

In addition, the "Voting Classifier" function in the Sklearn library in Python was used for more comprehensive experiments. This function calculates the results according to the joint decision of the classifiers used, so there have been changes in the accuracy rate.

Another result that can be drawn from the thesis is the experiment with machine learning algorithms. XGBoost gave the best classification results if machine learning was evaluated in terms of classification algorithms. While Random Forest takes the second place, the classification success of SVM is very low. XGBoost is based on more than one tree, so its success is high. Furthermore, the reason for this algorithm is successful is the classification and repetition of more than one tree. There is no repetition for random forest. So it lags behind XGBoost. SVM, on the other hand, classifies with linear separators when data cannot be separated linearly. In this thesis, SVM is used with the kernel parameter, but it is not suitable for this data set.

8. CONCLUSION

In this thesis, a solution was provided for detection of phishing attacks. This attack was taken advantage of people being unconscious about the attack and attackers were aim to deception users. They did by creating a web page that looks clean but is harmful. Then, they redirect such a malicious webpage to users. Users did not realize that these web pages are fraudulent sites and enter their personal information. This information was then sent to the attackers. The attacker, who receives personal information, is now able to perform all kinds of attacks.

A feature of phishing attacks was the ability to attack quickly. In many studies in the literature, it was stated that the attack took place within 2 hours, but it took 12 hours to detect the attack. For this reason, it is important to pay attention to time in solution methods.

Phishing attacks were grouped as list-based techniques, heuristic-based techniques, vision-based techniques, and machine-based techniques. In this thesis, vision-based techniques and attacks were conducted.

Moreover, it was determined whether a suspicious web page is a harmful web page or not. Classifiers based on machine learning were classified according to brand names. Therefore, computer vision techniques have also been used. One of these techniques was to use visual descriptors. Within the scope of this thesis, these descriptors were divided into global and local. In this context, SIFT, DAISY, GIST and LBP descriptors were investigated and used. In this experiment, it was observed that phishing web pages can be classified according to brands.

According to the experimental results, the best results were obtained from local descriptors. It was ensured the representation of 400 dimensional SIFT as 90.38%. However, 89.018 % test accuracy was obtained from GIST in experiments with global descriptors. GIST and LBP descriptors were used together to increase the success rate of global descriptors and the success rate was 88.0 %.

Another finding is that, along with having higher accuracy rate, XGBoost has several advantages such as GPU based training. The short duration of visual descriptor based on inference makes it a suitable, lightweight and practical scheme for being used as the first stage classifier in phishing detection mechanisms.

In order to advance the work done, it can be planned to use color more effectively in future work. For this reason, auto encoder based modeling can be developed and it is considered to work with deep convolutional neural networks.

REFERENCES

[1] Drake, C.E., Oliver, J.J. & Koontz, E.J. Anatomy of a phishing email, In CEAS 2014, **2014**.

[2] Bhuyan, M. H., Bhattacharyya, D. K., & Kalita, J. K. Network anomaly detection: methods, systems and tools. *IEEE communications surveys & tutorials*, 16(1), 303-336, **2014**.

[3] Y. Vural ve Ş. Sağıroğlu, Kurumsal Bilgi Güvenliği ve Standartları Üzerine Bir İnceleme, *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, 23(2), pp.507-522, **2008**.

[4] K. Geers, The challenge of cyber-attack deterrence, *Computer Law & Security Review*, 26(3), pp.298-303, **2010**.

[5] Chiew, K. L., Yong, K. S. C., Tan, C. L. A survey of phishing attacks: their types, vectors and technical approaches. *Expert Systems with Applications*, 106: 1-20, **2018**.

[6] Kathrine, G. J. W., Praise, P. M., Rose, A. A., & Kalaivani, E. C. Variants of phishing attacks and their detection techniques. In *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, 255-259, **2019**.

[7] Milletary, J., Center, C. C. Technical trends in phishing attacks. Retrieved December, 1(2007), 3-3, **2005**.

[8] Jain, A. K., & Gupta, B. B. Phishing detection: analysis of visual similarity based approaches, *Security and Communication Networks*, **2017**.

[9] Li, Y., Xiao, R., Feng, J., Zhao, L. A semi-supervised learning approach for detection of phishing webpages. *Optik-International Journal for Light and Electron Optics*, 124(23), 6027-6033, **2013**.

[10] Hong, J. The state of phishing attacks. *Communications of the ACM*, 55(1), 7481, **2012**.

[11] F. Mouton, M. Malan, L. Leenen and H.S. Venter, Social Engineer Attack Framework, *IEEE Conference on Information Security for South Africa*, pp. 1 – 9, **2014**.

[12] J. Allen, L. Goman, M. Green, P. Ricciardi, C. Sanabria and Steve Kim, Social Network Security Issues: Social Engineering and Phishing Attack ,*CSIS*, Pace University, pp. B1.1 - B1.7., **2012**.

[13] Hong J, The state of phishing attacks. *Commun ACM*, 55(1):74–81, **2012.**

[14] Eroğlu, E., Bozkır, A. S., & Aydos, M. Brand Recognition of Phishing Web Pages via Global Image Descriptors. *Avrupa Bilim ve Teknoloji Dergisi*, 436-443, **2019**.

[15] Bozkır, A. S., & Aydos, M. (2019). Local Image Descriptor Based Phishing Web Page Recognition as an Open-Set Problem. *Avrupa Bilim ve Teknoloji Dergisi*, 444-451. [16] Phishing Activity Trends Report 2019, www.apwg.org • info@apwg.org

[17] Ricardo da Silva Torres and Alexandre X. Falc˜ao. Content-Based Image Retrieval: Theory and Applications. *Revista de Informa´tica Teo´rica e Aplicada*,13(2):161–185, 6, 17, **2006**.

[18] Ot´avio Augusto Bizetto Penatti, Eduardo Valle, and Ricardo da Silva Torres. Comparative study of global color and texture descriptors for web image retrieval. *Journal of Visual Communication and Image Representation*, 23(2):359–380, **2012**.

[19] Szeliski, R. Computer vision: algorithms and applications. *Springer Science & Business Media*, **2010**.

[20] Prince, S. J. Computer vision: models, learning, and inference. Cambridge University Press, **2012**.

[21] Forsyth, D. A., & Ponce, J. Computer vision: a modern approach. *Prentice Hall Professional Technical Reference*, **2002**. Sıft ve hog gist kmeans için

[22] Histograms, https://statistics.laerd.com/statistical-guides/understanding-histograms.php (**23 November 2019**).

[23] Géron, A. Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems. *O'Reilly Media, Inc.*, **2017**.

[24] Safavian, S.R. and D. Landgrebe, A survey of decision tree classifier methodology. *IEEE transactions on systems, man, and cybernetics*, 21(3): p. 660-674, **1991**.

[25] Popular Machine Learning Algorithms, https://medium.com/technology-nineleaps/popular-machine-learning-algorithms-a574e3835ebb

[26] SuperVize Me: What's the Difference Between Supervised, Unsupervised, Semi-Supervised and Reinforcement Learning?, https://blogs.nvidia.com/blog/2018/08/02/supervised-unsupervised-learning/ (**23 November 2019**).

[27] Supervised vs. Unsupervised Machine Learning, https://medium.com/@chisoftware/supervised-vs-unsupervised-machine-learning-7f26118d5ee6 (**23 November 2019**).

[28] Introduction to Unsupervised Learning, https://algorithmia.com/blog/introduction-to-unsupervised-learning (**23 November 2019**).

[29] Machine Learning Explained: Understanding Supervised, Unsupervised, and Reinforcement Learning, https://datafloq.com/read/machine-learning-explained-understanding-learning/4478

[30] Zhu, X., & Goldberg, A. B. Introduction to semi-supervised learning. Synthesis lectures on artificial intelligence and machine learning, 3(1), 1-130, **2009**.

[31] Ren, Y. Indexing and Searching for Similarities of Images with Structural Descriptors via Graph-cuttings Methods (Doctoral dissertation), **2014**.

[32] Douik, A., Abdellaoui, M., & Kabbai, L. Content based image retrieval using local and global features descriptor. *In 2016 2nd International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*, pp. 151-154, IEEE, **2016**.

[33] Winder, S. A., & Brown, M. Learning local image descriptors. *In 2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-8, IEEE, **2007**.

[34] Swain, M. J., & Ballard, D. H. Indexing via color histograms. In Active perception and robot vision (pp. 261-273). Springer, Berlin, Heidelberg, **1992**.

[35] Dean S. Messing, Peter van Beek, and James H. Errico. The MPEG-7 colour structure descriptor: image description using colour and local spatial information. In International Conference on Image Processing, 1, pages 670–673, **2001**.

[36] Michael J. Swain and Dana H. Ballard. Color indexing. International Journal of Computer Vision,7(1):11–32, 11, 77, 82, 87, **1991**.

[37] Greg Pass, Ramin Zabih, and Justin Miller. Comparing images using color coherence vectors. In ACM International Conference on Multimedia, pages 65–73, **1996**.

[38] Jing Huang, Ravi Kumar, Mandar Mitra, Wei-Jing Zhu, and Ramin Zabih. Image indexing using color correlograms. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pages 762–768, **1997**.

[39] Sahingoz, O. K., Buber, E., Demir, O., & Diri, B. Machine learning based phishing detection from URLs. *Expert Systems with Applications*, 117, 345-357, **2019**.

[40] Rao, R.S. & Pais, A.R., Detection of phishing web sites using an efficient feature-based machine learning framework, *Neural Computing and Applications*, 1-23, **2018**.

[41] Corinna Cortes, Vladimir Vapnik, Support-vector networks, *Machine learning*, vol. 20, no. 3, pp. 273-297, **1995**.

[42] Google Safe Browsing API, https://developers.google.com/safe-browsing/ (Online accessed: 13.7.2019)

[43] Cao Y, Han W, Le Y. Anti-phishing based on automated individual white-list. *In: Proceedings of the 4th ACM workshop on digital identity management, ACM*, pp 51–60, **2008**.

[44] Zhang J, Porras PA, Ullrich J. Highly predictive blacklisting. **In: USENIX security symposium**, pp 107–122, **2008**.

[45] Prakash P, Kumar M, Kompella RR, Gupta M. Phishnet: predictive blacklisting to detect phishing attacks. *In: INFOCOM, 2010 Proceedings IEEE, IEEE*, pp 1–5, **2010**.

[46] Almomani A, Wan TC, Altaher A, Manasrah A. Evolving fuzzy neural network for phishing emails detection. *J Comput Sci* 8(7):1099, **2012**.

[47] Cao, Y., Han, W., & Le, Y. Anti-phishing based on automated individual white-list. *In Proceedings of the 4th ACM workshop on Digital identity management*, pp. 51-60. ACM, **2008**.

[48] Jain, A. K., & Gupta, B. B. Two-level authentication approach to protect from phishing attacks in real time. *Journal of Ambient Intelligence and Humanized Computing*, 9(6), 1783-1796, **2018**.

[49] Sheng, S., Wardman, B., Warner, G., Cranor, L. F., Hong, J., & Zhang, C. An empirical analysis of phishing blacklists. In Sixth Conference on Email and Anti-Spam (CEAS), **2009**.

[50] Tyagi, I., Shad, J., Sharma, S., Gaur, S., & Kaur, G. A Novel Machine Learning Approach to Detect Phishing Websites. *In 2018 5th International Conference on Signal Processing and Integrated Networks (SPIN)*, pp. 425-430. IEEE, **2018**.

[51] Pan, Y., & Ding, X. Anomaly based web phishing page detection. *In 2006 22nd Annual Computer Security Applications Conference (ACSAC'06)*, pp. 381-392, IEEE, **2006**.

[52] Zhang Y, Hong JI, Cranor LF. Cantina: a content-based approach to detecting phishing web sites. *In: Proceedings of the 16th international conference on World Wide Web, ACM*, pp 639–648, **2007**.

[53] Miyamoto D, Hazeyama H, Kadobayashi Y. An evaluation of machine learning-based methods for detection of phishing sites. *In: International conference on neural information processing. Springer*, pp 539–546, **2008**.

[54] Xiang G, Hong J, Rose CP, Cranor L. Cantina+: a feature rich machine learning framework for detecting phishing web sites. *ACM Trans Inf Syst Secur (TISSEC)*, 14(2):21, **2011**.

[55] He M, Horng SJ, Fan P, Khan MK, Run RS, Lai JL, Chen RJ, Sutanto A. An efficient phishing webpage detector. *Expert systems with applications* 38(10):12,018–12,027, **2011**.

[56] Gowtham R, Krishnamurthi I. A comprehensive and efficacious architecture for detecting phishing webpages. *Comput Secur* 40:23–37, **2014**.

[57] Aggarwal A, Rajadesingan A, Kumaraguru P. Phishari: automatic realtime phishing detection on twitter. *In: eCrime Researchers Summit (eCrime), IEEE*, pp 1–12, **2012**.

[58] Tan C.L., Chiew K.L., Wong K., Sze S.N. Phishwho: phishing webpage detection via identity keywords extraction and target domain name finder Decis *Support Syst*, 88, pp. 18-27, **2016**.

[59] Lee S., Kim J. Warningbird: A near real-time detection system for suspicious urls in twitter stream, *IEEE Trans Depend Secure Comput*, 10 (3), pp. 183-195, **2013**.

[60] Jeeva, S. C., & Rajsingh, E. B. Intelligent phishing url detection using association rule mining. *Human-centric Computing and Information Sciences*, 6(1), 10, **2016**.

[61] Babagoli, M., Aghababa, M. P., & Solouk, V. Heuristic nonlinear regression strategy for detecting phishing websites. *Soft Computing*, 23(12), 4315-4327, **2019**.

[62] Buber, E., Diri, B., & Sahingoz, O. K. NLP based phishing attack detection from URLs. In International Conference on Intelligent Systems Design and Applications (pp. 608-618). Springer, Cham, **2017**.

[63] Mohammad, R. M., Thabtah, F., & McCluskey, L. Predicting phishing websites based on self-structuring neural network. *Neural Computing and Applications*, 25(2), 443-458, **2014**.

[64] Feng, F., Zhou, Q., Shen, Z., Yang, X., Han, L., & Wang, J. The application of a novel neural network in the detection of phishing websites. *Journal of Ambient Intelligence and Humanized Computing*, 1-15, **2018**.

[65] Smadi, S., Aslam, N., & Zhang, L. Detection of online phishing email using dynamic evolving neural network based on reinforcement learning. *Decision Support Systems*, 107, 88-102, **2018**.

[66] Rao, R. S., Pais, A. R. Jail-Phish: An improved search engine based phishing detection system. Computers & Security, 83: 246-267, **2019**.

[67] Basnet, R. B., & Sung, A. H. Learning to Detect Phishing Webpages. J. *Internet Serv. Inf. Secur*., 4(3), 21-39, **2014**.

[68] Medvet, E., Kirda, E., & Kruegel, C. Visual-similarity-based phishing detection. In *Proceedings of the 4th international conference on Security and privacy in communication networks*, 22, ACM, **2008**.

[69] Zhang, W., Lu, H., Xu, B., & Yang, H. Web phishing detection based on page spatial layout similarity. *Informatica*, 37(3), **2013**.

[70] Rao, R. S., & Ali, S. T. A computer vision technique to detect phishing attacks. *In 2015 Fifth International Conference on Communication Systems and Network Technologies*, pp. 596-601. IEEE, **2015**.

[71] Hara, M., Yamada, A., & Miyake, Y. Visual similarity-based phishing detection without victim site information. *In 2009 IEEE Symposium on Computational Intelligence in Cyber Security*, pp. 30-36. IEEE, **2009**.

[72] Fu, A. Y., Wenyin, L., & Deng, X. Detecting phishing web pages with visual similarity assessment based on earth mover's distance (EMD). *IEEE transactions on dependable and secure computing*, 3(4), 301-311, **2006**.

[73] Chen, K. T., Chen, J. Y., Huang, C. R., & Chen, C. S. Fighting phishing with discriminative keypoint features. *IEEE Internet Computing*, 13(3), 56-63, **2009**.

[74] Bozkir, A.S. & Akcapinar Sezer, E. Use of HOG Descriptors in Phishing Detection, *In 4th International Symposium on Digital Forensic and Security (ISDFS),* **2016**.

[75] Dalgic, F. C., Bozkir, A. S., Aydos, M. Phish-IRIS: A New Approach for Vision Based Brand Prediction of Phishing Web Pages via Compact Visual Descriptors. *In 2018 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies* (pp. 1-8). IEEE, **2018**.

[76] Ricardo da Silva Torres and Alexandre X. Falc˜ao. Content-Based Image Retrieval: Theory and Applications. *Revista de Informa´tica Teo´rica e Aplicada*,13(2):161–185, 6, 17, **2006**.

[77] Oliva, A. and Torralba, A. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision*, 42(3), 145-175, **2001**.

[78] Wang, Y., Li, Y., &amp; Ji, X. Recognizing human actions based on gist descriptor and word phrase. In Proceedings 2013 International Conference on Mechatronic Sciences, Electric Engineering and Computer (MEC) (pp. 1104-1107). IEEE, **2013**.

[79] Sikirić, I., Brkić, K., &amp; Šegvić, S. Classifying traffic scenes using the GIST image descriptor. arXiv preprint arXiv:1310.0316, **2013**.

[80] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions", Pattern Recognition, vol. 29, pp. 51-59, **1996**.

[81] Raj, N. S., & Niar, V. (2017). Comparison study of algorithms used for feature extraction in facial recognition. Int. J. Comput. Sci. Inf. Technol, 8(2), 163-166, **2017**.

[82] Dalal, N., Triggs, B. Histograms of oriented gradients for human detection. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Diego, USA, **2005**.

[83] Lowe, D. G. Distinctive image features from scale-invariant keypoints. International journal of computer vision, **2004**, 60.2: 91-110.

[84] David G. Lowe. Object recognition from local scale-invariant features. In International Conference on Computer Vision, pages 1150–1157, 8, 20, 71, **1999**.

[85] Tola, E. DAISY: A Fast Descriptor for Dense Wide Baseline Stereo and Multiview Reconstruction. *PhD thesis*, **2010**.

[86] Lazebnik, S., Schmid, C., & Ponce, J. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. *In 2006 IEEE*

Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), Vol. 2, pp. 2169-2178, IEEE, **2006**.

[87] Breiman L. Random forests, machine learning, 2001 Kluwer Academic Publishers, 45(1), 5-32, **2001**.

[88] Chen, T., & Guestrin, C. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 785-794, ACM, **2016**.

[89] Schapire, R. E. A brief introduction to boosting. In *Ijcai* (Vol. 99, pp. 1401-1406, **1999**.

[90] Friedman, J., Hastie, T., & Tibshirani, R. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics*, *28*(2), 337-407, **2000**.

[91] Joshi, M. V. On evaluating performance of classifiers for rare classes. In *2002 IEEE International Conference on Data Mining, 2002. Proceedings*. (pp. 641-644). IEEE, **2002**.

[92] OpenCV, http://opencv.org/ (**23 November 2019**).

[93] What is Python?, https://www.python.org/doc/essays/blurb/ (**23 November 2019**).

[94] Scikit-learn user guide, https://scikit-learn.org/stable/_downloads/scikit-learn-docs.pdf (**23 November 2019**).

APPENDIX

APPENDIX 1 – Application Programming Interface (API)

sklearn.base:	Base classes and utility functions
sklearn.calibration:	Probability Calibration
sklearn.cluster:	Clustering
sklearn.cluster.bicluster:	Biclustering
sklearn.compose:	Composite Estimators
sklearn.covariance:	Covariance Estimators
sklearn.cross_decomposition:	Cross decomposition
sklearn.datasets:	Datasets
sklearn.decomposition:	Matrix Decomposition
sklearn.discriminant_analysis:	Discriminant Analysis
sklearn.dummy:	Dummy estimators
sklearn.ensemble:	Ensemble Methods
sklearn.exceptions:	Exceptions and warnings
sklearn.experimental:	Experimental
sklearn.feature_extraction:	Feature Extraction
sklearn.feature_selection:	Feature Selection
sklearn.gaussian_process:	Gaussian Processes
sklearn.isotonic:	Isotonic regression
sklearn.impute:	Impute
sklearn.kernel_approximation	Kernel Approximation

sklearn.kernel_ridge	Kernel Ridge Regression
sklearn.linear_model:	Generalized Linear Models
sklearn.manifold:	Manifold Learning
sklearn.metrics:	Metrics
sklearn.mixture:	Gaussian Mixture Models
sklearn.model_selection:	Model Selection
sklearn.multiclass:	Multiclass and multilabel classification
sklearn.multioutput:	Multioutput regression and classification
sklearn.naive_bayes:	Naïve Bayes
sklearn.neighbors:	Nearest Neighbors
sklearn.neural_network:	Neural network models
sklearn.pipeline:	Pipeline
sklearn.inspection:	inspection
sklearn.preprocessing:	Preprocessing and Normalization
sklearn.random_projection:	Random projection
sklearn.semi_supervised	Semi-Supervised Learning
sklearn.svm:	Support Vector Machines
sklearn.tree:	Decision Trees
sklearn.utils:	Utilities