

Martin Schiefloe Bakken

Webpage Fingerprinting using Infrastructure-based Features

Master's thesis in Communication Technology and Digital Security

Supervisor: Jan William Johnsen

June 2023

Martin Schiefloe Bakken

Webpage Fingerprinting using Infrastructure-based Features

Master's thesis in Communication Technology and Digital Security
Supervisor: Jan William Johnsen
June 2023

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Dept. of Information Security and Communication Technology





Norwegian University of
Science and Technology

Webpage fingerprinting using infrastruc- ture based features

Bakken, Martin Schiefloe

Submission date: June 2023

Supervisor: Johnsen, Jan William, NTNU

Norwegian University of Science and Technology
Department of Information Security and Communication Technology

Title: Webpage fingerprinting using infrastructure based features

Student: Bakken, Martin Schiefloe

Problem description:

Criminal activity online is rapidly evolving and is now a crucial part of the business model of criminal actors. A new business model has emerged from the evolution, known as Crime as a Service. Here criminal actors are able to buy and sell expertise, goods and services, with the intention of facilitating further crime. In the big business of cybercrime, the leaders are overseeing the business from a distance, while frontmen put all their effort and skills in maintaining availability of their website.

Mitigating criminal activities online depends on takedown of illegal websites. However, taking down one site often results in the same site reappearing with minor changes. This way the criminal enterprise remains hidden from law enforcement, while maintaining the relationship to their customers. Alterations may involve hosting details, page layout or displayed contents, which makes continuous takedown difficult. Still, creating truly unique copies is difficult to achieve as the process does not scale well. This leaves persistent features that allows for detection of malicious websites.

This study will contribute to the current work of the digital forensic research group at the Norwegian University of Science and Technology (NTNU). It will do so by investigating identification of malicious websites based on infrastructure features. The features will mainly be derived from available WHOIS data, as well as SSL/TLS certificate details. Following information gathering and feature generation, the project will apply a machine learning algorithm to classify the domains in the dataset.

Interesting results could be whether the machine learning model is able to distinguish malicious websites from benign and if so, which features are most prominent for the classification. Thereby the project can provide knowledge to what distinguishes malicious websites from benign ones. Other interesting results can include the ability to detect different types of malicious websites, eg. separating phishing websites from websites containing malware.

Approved on: 2023-03-17

Supervisor: Johnsen, Jan William, NTNU

Abstract

Cybercrime is evolving at an ever-increasing rate. Criminal actors are using websites to host illegal content and sell illegal goods and services, closely collaborating with each other. This new way of collaboration, where exact needs can be filled to facilitate further crimes, has evolved into the business model known as Crime as a Service. Mitigating online criminality is dependent on website detection and takedown.

This Master's thesis presents an experiment identifying malicious websites based on features extracted from the WHOIS records and SSL certificates of the domains. Using this information can ease early detection of malicious websites as WHOIS records are generated when the domain is registered. Furthermore, SSL certificates reveal information about the server hosting the domain. This thesis performs classification with the five machine learning algorithms, Random Forest, AdaBoost, Naive Bayes, Quadratic Discriminant Analysis, and Multi-Layer Perceptron, whose performance is compared and assessed. The training dataset was resampled to improve the performance of the classifiers using undersampling and oversampling. Using a dataset containing personal information, the thesis also performs adequate risk assessments and addresses ethical considerations using personal information in research.

The top-performing classifier was the Random Forest model using random undersampling to generate the balanced training dataset, achieving a recall score of 0.78 and an accuracy score of 0.76. Through the experiment, the thesis aims at providing insight into promising machine learning models for website classification, and what features generated from WHOIS records and SSL certificates can be used to identify malicious websites.

Sammendrag

Kriminalitet på internett utvikler seg i dag med et stadig større tempo. Kriminelle aktører benytter seg av nettsider for å spre ulovlig innhold, og til kjøp og salg av ulovlige varer og tjenester. Gjennom disse sidene kan aktørene utvikle et tett samarbeid, hvor de kan kjøpe hverandres kunnskap for å fasilitere sin egen kriminalitet. Denne nye forretningsmodellen er kjent som kriminalitet som tjeneste (CaaS). For å bekjempe nettkriminalitet er det nødvendig å identifisere og avvikle disse nettsidene.

Denne masteroppgaven utfører et eksperiment for å identifisere ond-sinnede nettsider ved hjelp av informasjon hentet fra WHOIS og SSL sertifikatene tilknyttet domenene. Ved å bruke denne informasjonen kan klassifiseringen skje tidlig i nettsidenes levetid, ettersom WHOIS informasjon genereres ved nettsidens registrering. SSL sertifikatene gir informasjon om serverene som nettsidene er tilgjengelige fra. Klassifiseringen er gjort av de fem maskinlæringsalgoritmene, “Random Forest”, “AdaBoost”, “Naive Bayes”, “Quadratic Discriminant Analysis” og “Multi-Layer Perceptron”. Deres resultater og ytelse er videre vurdert. For å forbedre algoritmenes prestasjon, ble treningsdataen balansert ved å bruke to forskjellige resamplingsteknikker: undersampling og oversampling. Ettersom denne oppgaven har brukt et datasett som inneholder personlig informasjon, utføres også passende risikoanalyser. I tillegg er etiske dilemmaer ved bruk av personlig informasjon i forskning adressert.

Modellen som hadde best ytelse var Random Forest, som ved bruk av undersampling, oppnådde tilbakekall på 0.78 og en nøyaktighet på 0.76. Gjennom dette eksperimentet ønsker masteroppgaven å finne ut hvilke maskinlæringsalgoritmer som utpeker seg ved ondsinnet nettsideidentifikasjon og hvilke typer informasjon fra WHOIS og SSL sertifikater som kan bidra til å klassifisere ond-sinnede nettsider.

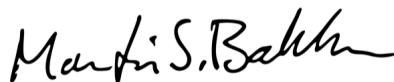
Preface

This thesis concludes my Master of Science in Communication Technology and Digital Security (MTKOM) at The Norwegian University of Science and Technology (NTNU). The topic selection was performed at the end of my 4th year of study, which was completed as an Erasmus exchange student at EURECOM in Sophia Antipolis, France. The topic was selected from a list of proposed thesis projects and was suggested by Jan William Johnsen.

The pre-project was conducted in the fall semester of 2022 and led to this thesis being written in the spring of 2023. Both the pre-project and this thesis were completed with Jan William as my supervisor. Jan William works as a researcher at NTNU Gjøvik at the Department of Information Security and Communication Technology (IIK). He has shown keen interest in the thesis topic and provided invaluable feedback throughout both the fall and the spring semesters. There are no strict prerequisites necessary to follow this thesis, as it will provide the required background information such that the reader may follow the later chapters. However, a general interest in technology, the Internet, and machine learning is beneficial.

I hope you will enjoy reading this thesis.

Trondheim, June 2023

A handwritten signature in black ink, reading "Martin S. Bakken". The signature is written in a cursive, slightly slanted style.

Martin Schiefloe Bakken

Acknowledgments

First and foremost, I would like to thank my supervisor Jan William Johnsen for his continuous support and enthusiasm for this work. Completing the thesis would be impossible without his contribution through feedback on my writing and weekly meetings discussing progress and future opportunities. Also, I want to direct my thanks to VirusTotal for providing me with an API key less restrictive than their publicly available key. This has been instrumental in facilitating this thesis experiment.

Furthermore, I want to thank everyone who has contributed to the good environment in our class. You have created an exceptional class by both organising social events throughout the five years we have been studying together, and creating an excellent study culture in our study room this final year. Also, I want to acknowledge the incredible people I met during my time at EURECOM as an exchange student. To all my fellow classmates, both at NTNU and EURECOM, I wish you all the best and thank you for being part of my time as a student.

Finally, I want to express my deepest gratitude to my family for providing me with continuous support throughout my time as a student and showing keen interest in my work, albeit not understanding the study field. You have pushed me onwards from elementary school until my final days as a student at NTNU, for which I am forever grateful. Also, a special mention to my closest friends with whom I have shared trips in Europe, beer tastings, sports events, quizzes, challenging courses, frustration, and great times through these years. I will always cherish the memories we made during my time as a student, and I am sure our friendships will endure and flourish in the coming years.

M. S. B.

Contents

List of Figures	xiii
List of Tables	xv
Source Code	xvii
List of Acronyms	xix
1 Introduction	1
1.1 Topics Covered	2
1.2 Keywords	3
1.3 Justification, Motivation and Benefits	3
1.4 Research Questions	3
1.5 Contributions	4
1.6 Thesis Outline	4
2 Background	7
2.1 Cybercrime	7
2.2 Infrastructure Information	9
2.2.1 WHOIS	9
2.2.2 SSL Certificate	10
2.3 Machine Learning	13
2.3.1 Supervised Machine Learning	14
2.3.2 Feature Preprocessing Techniques	15
2.3.3 Machine Learning Challenges	17
2.3.4 Evaluation Metrics	17
3 Related Work	19
3.1 Content Classification	19
3.1.1 Hashing	20
3.1.2 Website Appearance	21
3.1.3 Keywords	22
3.2 Infrastructure	23

3.2.1	WHOIS	23
3.2.2	SSL Certificate	24
4	Methodology	27
4.1	Choice of Methods	27
4.2	Dataset	28
4.3	Machine Learning Algorithms	28
4.3.1	Implemented Machine Learning Algorithms	29
4.3.2	Validation Techniques	32
4.3.3	Other Machine Learning Models	34
4.4	Research Questions	35
4.4.1	Machine Learning Evaluation	35
4.4.2	Feature Importance	35
5	Experiments	37
5.1	Environment Setup	37
5.2	Dataset	39
5.2.1	Feature Engineering	40
5.2.2	Preprocessing	49
5.3	Machine Learning Implementation	52
6	Results and Discussion	57
6.1	Machine Learning Performance	57
6.1.1	Precision and Recall	62
6.1.2	F1-score	64
6.1.3	Accuracy	66
6.2	Feature Importance	67
6.3	Theoretical Implications	68
6.4	Practical Recommendations	70
6.5	General Discussion	71
6.6	Conclusions	71
6.7	Future Work	74
7	Ethics	77
7.1	Pre-experiment	77
7.2	Research Ethical Considerations	82
7.2.1	General Research Ethics	82
7.2.2	Internet Research Ethics	84
	References	87
	Appendix	

A	Confusion Matrices (random undersampling)	97
B	Confusion Matrices (random oversampling)	99
C	Result Plots	101
D	Feature Importances	107
E	Notification Form for Personal Data	109
F	Risk Assessment	117
G	Data Management Plan	123

List of Figures

2.1	Sequence diagram illustrating the WHOIS protocol as defined by RFC3912 [Dai04]	9
2.2	Sequence diagram illustrating a client-server connection setup with SSL/TLS certificates [HREJ14]	11
2.3	Taxonomy of machine learning methods [KK07]	13
2.4	Binary classification	14
2.5	k-fold cross-validation with k= 5 [Sci:CV]	15
2.6	Two different discretisation strategies	16
3.1	A simple hash function mapping names to binary values [Wik23b]	20
4.1	A Quadratic Discriminant Analysis model with quadratic boundaries [Sci:QDA]	31
4.2	A Multi-Layer Perceptron model with one hidden layer [Sci:MLP]	32
4.3	A confusion matrix displaying the results from a malicious/benign classification	34
5.1	Overview of the method to acquire information about domains	39
5.2	A pie chart visualising the imbalanced dataset used in the experiment	52
6.1	Normalised confusion matrices with random undersampling as the resampling technique for training data	60
6.2	Normalised confusion matrices with random oversampling as the resampling technique for training data	61
6.3	Recall scores using random undersampling	63
6.4	F1-scores using random undersampling	65
6.5	Accuracy scores using random oversampling	67
6.6	Feature importances using random undersampling as resampling method for training data	68
A.1	Confusion matrices with random undersampling as balancing method for training data	98

B.1	Confusion matrices with random oversampling as balancing method for training data	100
C.1	Precision scores using random undersampling	102
C.2	Precision scores using random oversampling	102
C.3	Recall scores using random undersampling	103
C.4	Recall scores using random oversampling	103
C.5	F1-scores using random undersampling	104
C.6	F1-scores using random oversampling	104
C.7	Accuracy scores using random undersampling	105
C.8	Accuracy scores using random oversampling	105
D.1	Feature importances using random undersampling as resampling method for training data	107
D.2	Feature importances using random oversampling as resampling method for training data	108

List of Tables

4.1	Description of acronyms used to derive classification evaluation metrics	33
5.1	Technical specifications for the NTNU SkyHigh server	38
5.2	Attributes present in the used dataset [Sin20]	41
5.3	Relevant attributes available through the different VirusTotal APIs . . .	42
5.4	Features used in this thesis	44
5.5	Hyperparameter values for implementation of the Random Forest classifier	53
5.6	Hyperparameter values for implementation of the Multi-Layer Perceptron classifier	54
6.1	Classification performance metrics with random undersampling	58
6.2	Classification performance metrics with random oversampling	58
6.3	Classification precision and recall scores	62
6.4	Classification F1-scores	64
6.5	Classification accuracy scores	66
7.1	Consequence and probability levels according to NTNU risk assessment template [NTNUd]	78
7.2	Risk levels according to NTNU risk assessment template [NTNUd] . . .	79
7.3	Moderate risk events discovered through the risk assessment	80
7.4	Unwanted events and countermeasures to reduce risk level	81

Source Code

5.1	WHOIS creation time matching	46
5.2	SSL start and end time matching	46
5.3	SSL certificate issuer information	47
5.4	Code to generate bool_co by using the pycountry library	48

List of Acronyms

- API** Application Programming Interface.
- CA** Certificate Authority.
- CaaS** Crime as a Service.
- CE** Child Exploitation.
- CPU** Central Processing Unit.
- CSS** Cascading Style Sheets.
- csv** Comma-separated values.
- DNS** Domain Name Server.
- DSA** Digital Signature Algorithm.
- EC** Elliptic Curve.
- EU** European Union.
- FN** False Negative.
- FP** False Positive.
- FTP** File Transfer Protocol.
- GDPR** General Data Protection Regulation.
- GUI** Graphical User Interface.
- HTML** HyperText Markup Language.
- HTTP** HyperText Transfer Protocol.
- HTTPS** HyperText Transfer Protocol Secure.

ICANN Internet Corporation for Assigned Names and Numbers.

IETF Internet Engineering Task Force.

IOCTA Internet Organized Crime Threat Assessment.

IP Internet Protocol.

ISO International Organization for Standardization.

k-NN k-Nearest Neighbours.

LEAs Law Enforcement Agencies.

LLN Law of Large Numbers.

MDI Mean Decrease in Impurity.

ML Machine Learning.

MLP Multi-Layer Perceptron.

NFL No Free Lunch.

NTNU Norwegian University of Science and Technology.

PDNS Passive Domain Name Server.

PKI Public Key Infrastructure.

QDA Quadratic Discriminant Analysis.

RAM Random Access Memory.

RFC Request for comments.

RSA Rivest-Shamir-Adleman.

SCP Secure Copy Protocol.

SSH Secure Shell.

SSL Secure Sockets Layer.

SVM Support Vector Machine.

TCP Transmission Control Protocol.

TF-IDF Term Frequency-Inverse Document Frequency.

TLD Top Level Domain.

TLS Transport Layer Security.

TN True Negative.

TP True Positive.

URL Uniform Resource Locator.

VPN Virtual Private Network.

XSS Cross-Site Scripting.

Chapter 1

Introduction

The Internet has provided connectivity in the world, enabling communication, trade and commerce across borders and oceans. Opportunistic criminals have also taken advantage of the possibilities offered by the Internet. While this was only utilised by a few initially, this is now an approach every major criminal actor uses [Eur21]. The business model is now changed to a more organised way of conducting criminal activity online. Large groups can cooperate and take advantage of niche competence sold by other groups or individuals. Through acquiring knowledge, malware or vulnerabilities, criminal groups facilitate further crime. This is known as the business model Crime as a Service (CaaS) [Man13].

In an effort to reduce criminal activity online, several methods to detect illegal websites have been proposed, many of which focus on detecting the content on the sites. This approach has proved successful in detecting known Child Exploitation (CE) material, but has one major disadvantage by allowing content publication before detection. This thesis will investigate opportunities for detecting illegal (malicious) websites based on their infrastructure details, specifically WHOIS records and Secure Sockets Layer (SSL) certificates. These sources provide information about the domains and the servers that host the domains. Similar approaches have previously been used in different research projects. E.g., detecting abnormalities in WHOIS records was performed by Cheng and Chai et al. [CCZ+22]. Also, SSL certificates have been used for phishing detection by Sakurai and Watanabe et al. [SWO+21]. Many modern security applications using WHOIS records rely on the personal information contained in the records to determine the intentions of the websites. However, the implementation of the General Data Protection Regulation (GDPR) legislation has now limited the available personal information in WHOIS records [LLZ+21].

Unlike modern security applications, this thesis will only use features from WHOIS records that are available after the implementation of GDPR. It will thus investigate whether information from the WHOIS records may still be used to classify malicious

websites. In addition, the thesis will use other available features extracted from SSL certificates. The experiment will be performed on a large, imbalanced dataset with more than 350 000 samples, using five different Machine Learning (ML) classifiers.

1.1 Topics Covered

This thesis will cover relevant topics to provide the reader with a comprehensive understanding of the problem area and current leading trends in related work. The thesis will also give the reader sufficient insight into the relevant methodology used in the experiment.

In particular, the thesis describes leading trends in cybercrime, such as the CaaS business model enabling close collaboration between criminal actors worldwide [Man13; Eur21]. One of the major concerns with the evolution of this business model is how criminal actors now exchange information and expertise. E.g., an illegal marketplace no longer contains only illegal goods such as weapons and drugs, but zero-day vulnerabilities, ready-made exploit kits and malware can be bought as services and used for personal gain by the criminal actors. Combating online criminality requires malicious websites to be discovered.

Website classification is the process of determining the intentions of an investigated website. As the Internet contains approximately 1.13 billion websites [Haa23], this process is impossible to perform manually. ML thus presents opportunities for automated website classification and malicious website detection.

This thesis addresses ML topics to provide the reader with the necessary background information. Concepts such as supervised ML, preprocessing techniques, and evaluation methods are covered in detail. It is important to facilitate ML classification by preprocessing the data, i.e., the data must be manipulated such that the algorithms can handle it. The performed experiment will be described in detail, providing the reader with an insight into the implementation details of the experiment.

The thesis does not consider a distinction between malicious and illegal websites. Initially, the thesis set out to identify *illegal* websites. However, it proved difficult to obtain a dataset containing such websites. Therefore, the experiment uses a dataset with *malicious* labelled websites. The dataset contains personal information, and a thorough assessment of ethical considerations in research and Internet-based research using personal information is therefore also covered.

1.2 Keywords

Webpage fingerprinting, malicious website detection, supervised machine learning, WHOIS, infrastructure, SSL certificate and cybercrime.

1.3 Justification, Motivation and Benefits

Cybercrime is facilitated by illegal websites and websites holding illegal content. Europol has identified an increased use of the Internet for collaboration between criminal actors [Eur21]. Furthermore, a new business model known as Crime as a Service (CaaS) has emerged from this collaboration. Here criminals can buy and sell their expertise to facilitate crimes [Man13].

Many suggested website classification approaches are based on the content they hold. This enables the crime to be committed before the website is detected and discontinued [CCZ+22]. The problem with this approach is the need for the content to be published before the classification can take place. At the same time, detecting websites with malicious intentions as early as possible is vital to reduce the window in which online crimes can be committed.

Investigating classification based on registration information such as WHOIS records, which are created at domain registration, and SSL certificate information, it is possible to detect malicious websites in an early phase [SWO+21]. This serves as the main motivation for assessing infrastructure-based features in this thesis. Furthermore, the implementation of GDPR and the following restriction of available WHOIS information motivates the study to investigate whether WHOIS records can still be used for malicious website classification.

WHOIS records have been used for similar purposes in previous work, as described in Chapter 3, and thus justifies the implementation of WHOIS records as the basis for feature generation. Also, SSL certificates have been used in phishing detection and should therefore be applicable for malicious website detection.

1.4 Research Questions

The main ideas from the research questions coined during the pre-project are preserved [Bak22]. However, with the acquisition of a labelled dataset, a change from unsupervised to supervised ML was decided. This is reflected in the new research questions stated below.

1. Which supervised machine learning algorithm performs best in malicious website classification?

2. Which infrastructure-based features can distinguish a malicious website from a benign one?

1.5 Contributions

Given the limitations now imposed on available information from WHOIS records, this thesis seeks to provide insight into what information from WHOIS can still be used for malicious website classification. Also, it finds potential features from SSL certificates used in combination with the WHOIS records. As reflected in the research questions, the thesis will also investigate ML algorithms to identify promising candidates in malicious website classification.

The thesis will investigate the classification of malicious websites using only publicly available information from WHOIS records, combined with information from SSL certificates. The test dataset is imbalanced, reflecting the disproportion of malicious and benign websites currently online. This has, to our knowledge, not previously been done.

In the end, the thesis hopes to provide information which can ease the work with malicious website detection through infrastructure-based features, allowing early detection and takedown.

1.6 Thesis Outline

This section presents the thesis outline. It does so by listing the thesis chapters and briefly explaining what the reader can expect to find in each chapter.

- Chapter 2 provides the reader with insight into cybercrime which serves as the basis for this work. After the cybercrime section, background information about WHOIS and SSL certificates will describe important aspects relevant to the features used in the experiment. Finally, the chapter will provide substantial knowledge of supervised machine learning, as the proposed method in Chapter 4.
- Chapter 3 presents relevant related work, hereunder different approaches of website classification. In essence, there are two different approaches covered by this chapter, namely content- and infrastructure-based classification.
- Chapter 4 presents an overview of the proposed method to conduct the experiment. It presents dataset acquisition, relevant machine learning algorithms and their evaluation metrics. Finally, the chapter presents the stated research questions along with the suggested approach to answer them and the expected results.

- Chapter 5 presents the implementation details of the experiment which was performed. Hereunder, are the environment setup, dataset manipulation, and feature engineering. The chapter also presents the implementation details of the selected machine learning algorithms.
- Chapter 6 presents the findings from the experiments. It will present the results according to the stated research questions and provide a discussion covering the theoretical implications, practical considerations, general discussion, conclusions, and proposals for future work.
- Chapter 7 will cover ethical considerations this thesis had to make when processing a dataset containing personal information.

Chapter 2

Background

Following a brief introduction covering the problem area, justification, motivation, and research questions in the previous chapter, this chapter will present relevant background knowledge, providing further depth and insight into the problem area. The chapter is divided into sections covering cybercrime in Section 2.1, the infrastructure of domains in Section 3.2, and machine learning in Section 2.3. Note that the cybercrime section is partly based on findings from the pre-project [Bak22].

2.1 Cybercrime

The introduction of this thesis briefly described how connectivity across the globe facilitates an intercontinental criminal network where criminal actors cooperate and trade. Using the Internet to conduct criminal activity is no new phenomenon. Taking advantage of any opportunity provided to reach their goals and escape the law has long driven criminal actors. However, the latest tendencies in criminal activity online manifest the need to implement mitigating actions. Criminal actors cooperate at an entirely different level through the new business model, CaaS. Previously, one had to possess the expertise and unique skills to conduct criminal activity online. Now, it is possible to buy and sell services at illegal marketplaces making anyone a cybercrime expert [Man13]. In essence, the CaaS business model enables criminal actors to buy niche knowledge and expertise to facilitate their crimes through online marketplaces. For example, a criminal group may purchase a zero-day vulnerability, create an exploit, and then sell the exploit to a new actor wanting to attack a vulnerable service. This example shows the interconnectedness of criminal actors, where each serves as a link facilitating the crime conducted by the final party.

An ever more connected criminal network requires a cooperative effort to combat. Through Europol, the European police departments collaborate to face international organised crime. Europol publishes the Internet Organized Crime Threat Assessment (IOCTA) report, which assesses organised criminality online. The report from 2021

[Eur21] presents a world using the Internet far more because of the pandemic. This also offers opportunities for criminal actors to target users with phishing attacks and fraud. The report shows that the opportunities for fraud have multiplied through the increased use of online shopping. Furthermore, phishing attacks are now both more sophisticated and increased in volume. While Internet use is far from a revolution in criminal environments, the rate at which its use has increased, and its evolution has grown throughout the pandemic has never been seen before [Eur21].

The close cooperation and increased sophistication in their methods and organisation have enhanced the potential of criminal actors online. This requires intensified collaboration between Law Enforcement Agencies (LEAs) to mitigate. Europol lists several recommendations for easing the work to fight Internet criminality in the 2021 report, including removing obstacles for investigators [Eur21]. One of the most relevant recommendations for this thesis is to enforce stricter regulations on Internet Protocol (IP) address and domain registration. This thesis uses WHOIS information created during domain registration and SSL certificate information to classify malicious websites. There is little to no validation of the provided information when registering a domain, enabling criminals to provide invalid details and reuse information across several websites [CCZ+22; Kre18]. However, imposing clear registration restrictions could enhance data quality, improving the malicious website classification performance [Bak22].

Categories of illegal websites cover more than criminal marketplaces. Illegal streaming services annually inflict revenue losses in the billion-dollar range for both the film and television industries [Spa22]. Also, a more relevant concern for individuals is the flourishing of scam websites such as phishing. A recent report shows that 70% of newly registered websites have malicious intentions [Clu19; Fin23]. The report further indicates that newly registered domains are more suspicious than long-lived ones. Also, research suggests that some Top Level Domains (TLDs) are more likely to host malicious content [Clu19].

Still, the above examples have relatively trivial consequences compared to distribution of illegal content. The consequences inflicted by the distribution of CE material and human trafficking are hard to emphasise. Addressing the consequences of such severe topics are left for professional actors supported by adequate psychological assistance. This thesis will assess the methods used in related work to detect CE material, as they have relevance to the thesis methodology and serve as inspiration to detect other types of malicious or illegal websites. However, the topic and its consequences will not be covered in more detail.

2.2 Infrastructure Information

This section will provide background information about the infrastructure of domains. It will assess what information is available and provide insight into WHOIS and SSL certificates. Later, in Chapter 5, a more detailed description of how this thesis acquired information and what features were generated and used is given.

2.2.1 WHOIS

WHOIS is a Transmission Control Protocol (TCP)-based query-response protocol created to provide domain registration information, such as who has registered the domain, i.e., the *registrant* and who is responsible for distributing the domain, i.e., the *registrar*. The protocol is defined by the Internet Engineering Task Force (IETF) in Request for comments (RFC) 3912 [Dai04]. The protocol works by setting up a TCP connection to the WHOIS server and querying information about a requested domain. A sequence diagram illustrating the process of setting up a TCP connection and querying information about a domain according to the WHOIS protocol, is available in Figure 2.1. While the protocol offers standardisation for transporting WHOIS information, the format in which the information is sent varies among the different providers [LFS+15].

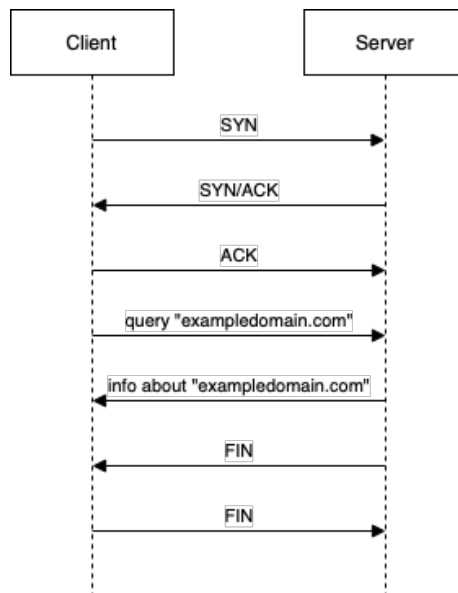


Figure 2.1: Sequence diagram illustrating the WHOIS protocol as defined by RFC3912 [Dai04]

The Internet Corporation for Assigned Names and Numbers (ICANN) requires each registrant to provide information upon domain registration. The information includes contact details about the registrant, e.g., name and telephone number, and is made publicly available after registration. This is done to allow network administrators to fix problems and maintain the stability of the Internet. Also, the information helps determine the availability of domain names, combat inappropriate use of the Internet, e.g., spam and facilitates identification of trademark infringement. Furthermore, it enhances the accountability of domain name registrants [ICANN].

WHOIS data includes information such as the name, phone number, address, email address, and country of the registrant [WHLM13]. As described in the previous paragraph, the benefits of publicly available WHOIS data are many. However, it also presents negative aspects, such as unwanted spam and fraudulent emails to the registrants' email addresses [ICANN]. Following the GDPR regulation, all personal information about the registrant in WHOIS records across Europe is redacted [Oli19; BDF18].

Cybersecurity professionals warned against the implementation of GDPR and its restrictions on the publicity of WHOIS information [BDF18]. Research showed that even before GDPR was implemented, criminals provided non-viable information when registering domains [CM14]. The information could thus not be used to contact or localise the criminal actors. However, as will be shown in Chapter 3, research has leveraged the presence of fraudulent information as a basis for feature generation and website classification.

2.2.2 SSL Certificate

The adoption of HyperText Transfer Protocol Secure (HTTPS) was initially low for malicious websites such as phishing. In 2015, less than 2% of the phishing websites used HTTPS. However, with the now widespread availability and low costs of Certificate Authority (CA) signatures, the percentage of phishing websites adopting HTTPS had risen to 74% by 2019 [SWO+21]. SSL certificates enable encrypted HyperText Transfer Protocol (HTTP) traffic and verification of the server to which the client is communicating. The certificates verify the server by assessing the issuer of its X.509 digital certificate [HREJ14]. The X.509 standard provides specifications in the semantics and format for the Internet Public Key Infrastructure (PKI) and is defined in RFC 5280 [CSF+08]. Since its creation, the SSL protocol has revolutionised the confidential use of the Internet through encryption. The protocol was first defined in RFC 6101 [FKK11] and has since been replaced by the Transport Layer Security (TLS) protocol as defined in RFC 5246 [DR08]. The names *SSL* and *TLS* are now, for certificates, used interchangeably. The remainder of this thesis will address the certificates according to their original name, i.e., SSL.

A sequence diagram illustrating the client-server connection with SSL certificates is illustrated in Figure 2.2. Like the WHOIS protocol, the initial steps set up a TCP connection through the SYN-SYN/ACK handshake. Then follows a ClientHello message from the client containing its supported cipher suites. This is replied to by the server with the certificate and the server-chosen cipher suite in the ServerHello message. The certificate contains information such as the server's public key and is digitally signed by a CA. Both the client and the server then derive session keys. In the last step, they inform each other that the following data will be encrypted according to the agreed-upon cipher in the ClientKeyExchange and ChangeCipherSpec messages [HREJ14].

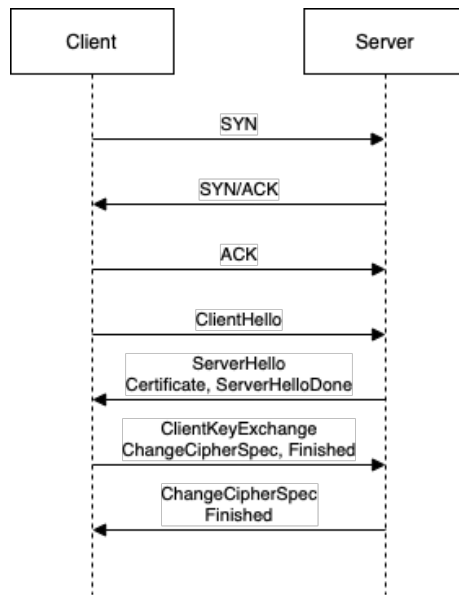


Figure 2.2: Sequence diagram illustrating a client-server connection setup with SSL/TLS certificates [HREJ14]

The client is also responsible for verifying the public key of the server it receives through the certificate. This is done by verifying that the signature is from a trusted CA. CAs serve as trusted third parties verifying the identity of the server. CAs are arranged in a tree structure where *root* CAs have implemented trust in the client's browsers. The tree structure represents the hierarchical arrangement describing interdependencies between different CAs, which culminates in the leaf CA verifying the server's identity. Avoiding signing server certificates with a root CA, is implemented as a security measure to mitigate the risk of compromise for a root CA [HREJ14].

It is worth noting that the server alone is responsible for selecting the cipher suite. The server selects one of the supported cipher suites provided by the client and informs the client of the selection in the ServerHello message. There is, however, no guarantee that the server selects the strongest supported cipher. Without any voting from the client, the server may thus opt for a weaker cipher. JARM hash-values can be used to fingerprint server setup as they investigate the cipher selection of the server in the SSL/TLS session establishment [Alt17].

JARM is a TLS fingerprinting tool based on the response created by the server in the ServerHello message in the TLS setup. The message differs based on several aspects, such as [Alt17];

- Operation system
- Operation system version
- Libraries used
- Custom configuration

With all these possible variations, it is unlikely that a server deployed by two different companies will have similar ServerHello responses. This is leveraged by JARM to facilitate the server fingerprinting. To create the fingerprints, the JARM client sends ten uniquely crafted ClientHello messages to the server and hashes the aggregated responses. The messages sent from the client are crafted to provide as much information from the server as possible. This includes trying different variations of ciphers, e.g., discovering if the server chooses the strongest available cipher and which TLS version it will choose if the client suggests TLS 1.3, which is the newest version available. All the answers to the questions the JARM client is asking are then hashed to form the 62-character JARM hash. The first 30 characters of the hash are directly related to the cipher and TLS version chosen by the server. While the last 32 characters are a SHA256 hash of the extensions provided by the server in the ServerHello messages [Alt17].

Aside from providing information about the server setup, the SSL certificates contain other information. This includes information such as validity dates, issuer information, hereunder name, country, and organisation, and encryption algorithm [SWO+21; VTssl]. Fingerprinting using this information will be investigated in the experiment in this thesis.

2.3 Machine Learning

This section presents background information about the machine learning part of the thesis. First, an introduction to supervised machine learning as the proposed method in Chapter 4, is given. This will be followed by an introduction to preprocessing techniques before general applications and challenges in machine learning classification are presented. Finally, a description of validation methods concludes this chapter.

As mentioned in Section 2.1 at the start of this chapter, the number of malicious websites is rising. Aside from the already discussed phishing problems, general tendencies in malicious website registration are rising. As many as 1 of 13 (7.7%) Uniform Resource Locators (URLs) were malicious in 2018 [SG19]. According to Forbes [Haa23], there are approximately 1.13 billion websites on the Internet. With 7.7% of them being malicious, this resolves to approximately 87 million malicious websites. Manually checking and labelling all of these would be impossible. A possible solution may be ML which can process huge amounts of data in a reasonable amount of time. Therefore, ML presents great opportunities for detecting malicious websites, and many research projects have been proposed with different strategies for detection [SG19].

ML algorithms are distinguished in the way that they use the obtained (induced) knowledge [KK07]. The methods are subsequently separated into classification, regression, clustering, learning of associations, relations, and differential equations, as illustrated in Figure 2.3. Note also the distinction between the classes of supervised and unsupervised learning. A closer description of supervised learning will follow in the next subsection. However, due to its irrelevance to the experiment in this thesis, the unsupervised learning approach will not be further assessed.

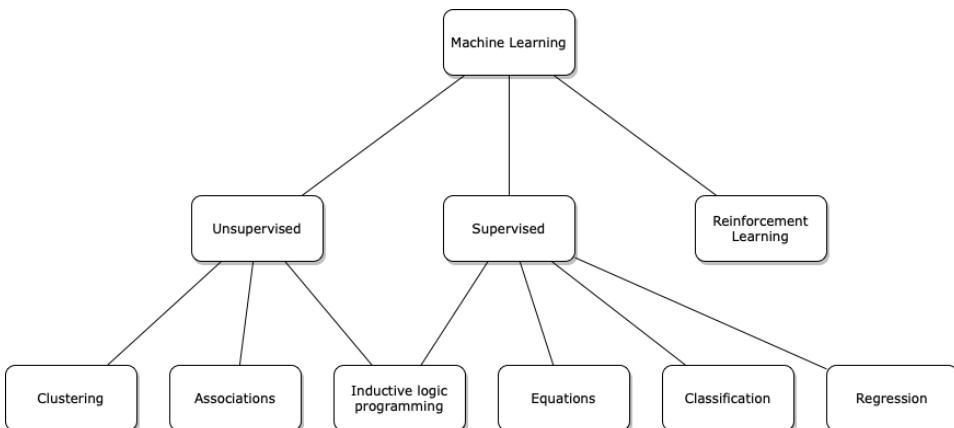


Figure 2.3: Taxonomy of machine learning methods [KK07]

2.3.1 Supervised Machine Learning

In supervised learning, the learning algorithm trains on a set of samples with associated labels [MRT18]. Through this learning process, the ML model learns the patterns of the training samples. I.e., it learns to map the features of the samples to a given label. The features comprise a set of attributes associated with the sample. The model’s performance is later tested with previously unseen data, known as test samples. As shown in Figure 2.3, supervised learning can be further split into *inductive logic programming*, *equations*, *classification*, and *regression*.

Classification, as used in this thesis, will be discussed in more detail here. ML models are used as classifiers to determine the *class* of a given test sample [KK07; MRT18]. The number of classes can be of arbitrary size to give a *multiclass* classification problem or, in the case of two distinct classes, a *binary* classification problem. To predict the class, the classifier has to create a mapping (function) between the input attributes and the target classes. This can be done by assigning a predefined function or through the learning process as described above [KK07]. The classification problem is illustrated in Figure 2.4, where the classification algorithm separates classes A and B in a binary classification problem.

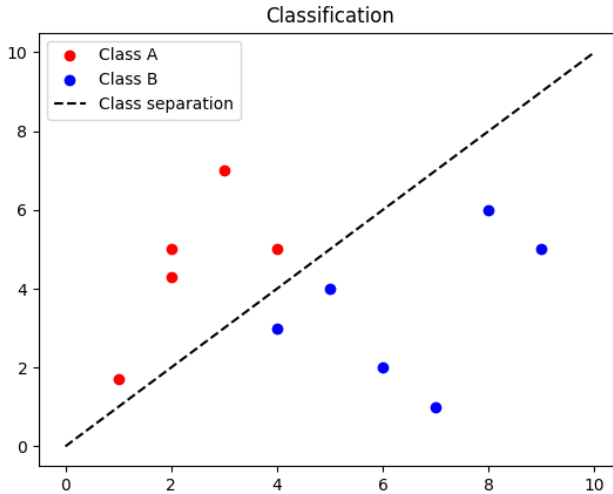


Figure 2.4: Binary classification

Supervised ML has the potential of *overfitting* the model when trained on training data [Dom12]. This phenomenon is characterised by good performance for the training data and poor performance for the testing data. I.e., the model becomes very

good at classifying the training data correctly but fails to see the general patterns of the samples. Thus, an overfitted model will have a poor performance on the test data [Dom12]. Cross-validation can be used to mitigate the problem of overfitting [Dom12; Ber19]. There are several ways to perform cross-validation, one of which is the k-fold cross-validation. This approach splits the training dataset into k disjoint subsets, where k-1 of them are used for training the model, and the last partition is used to verify the results. The process is repeated until all k subsets have served as validation set [Ber19]. Figure 2.5 illustrates the 5-fold cross-validation process used to find optimal parameters for a supervised machine learning model. Note also that the test data is used only for the final evaluation and is not previously seen by the algorithm.

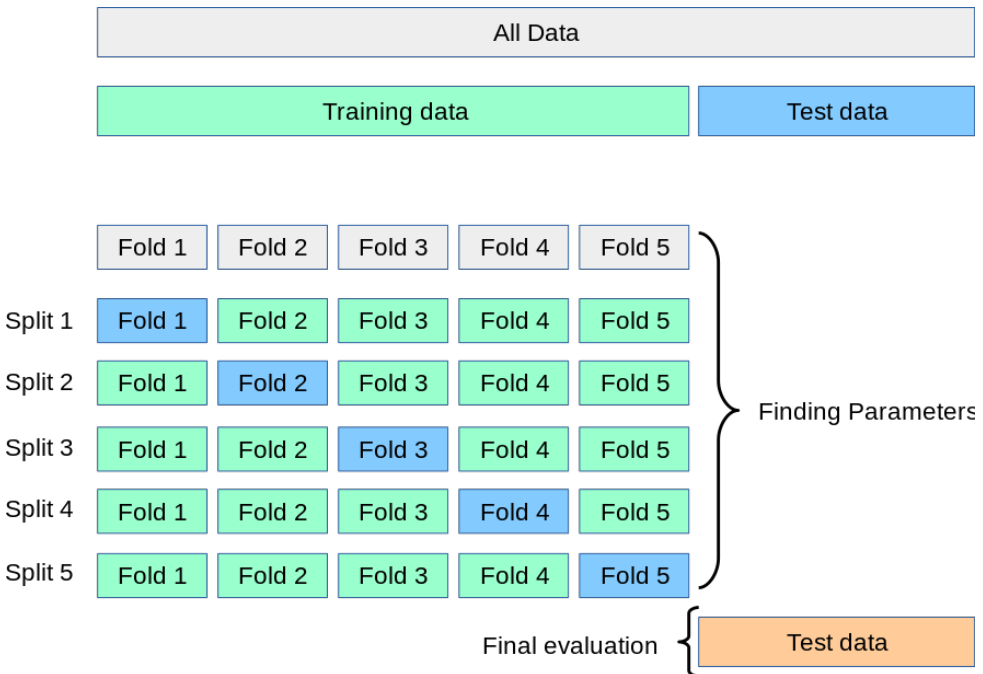


Figure 2.5: k-fold cross-validation with $k=5$ [Sci:CV]

2.3.2 Feature Preprocessing Techniques

The performance of an ML algorithm is directly related to the input features. Variations such as discrete or continuous variables can have a major impact on the algorithm's performance [KK06]. Challenges emerge when dealing with datasets with missing features or features in the wrong format. E.g., some algorithms may not support continuous features or may not support missing values. This requires the

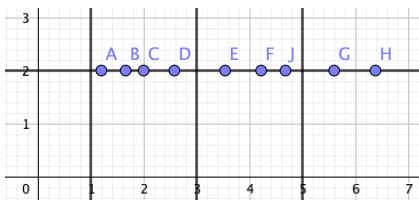
dataset to be preprocessed. In this subsection, the relevant preprocessing methods for the experiment are presented.

Imputation

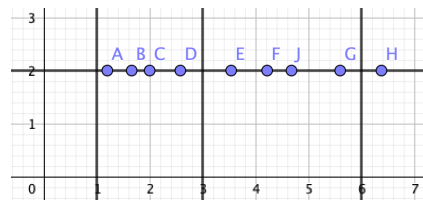
Handling datasets with missing values can be challenging, as not all ML algorithms support missing values [Sci:imp]. A straightforward solution to this problem is to drop the samples with missing values [Van18]. This does, however, result in a smaller dataset, and if the dataset is not large enough, this approach will disadvantageously affect the learning. A better approach is to fill the missing values with an estimation called imputation. Several methods to impute values have been suggested. Among the suggestions is filling the values with the mean value computed from the available (corresponding) features [TK06].

Discretisation

Many ML algorithms perform better when continuous attributes are discretised [KK06; ER04]. Discretisation of continuous variables is the process by which the variable is partitioned into a finite number of intervals. There are, however, an infinite number of possible ways to conduct the discretisation. In many cases, the user has to manually determine the number of intervals or samples in each interval. An illustration of discretisation is given in Figure 2.6, where Figure 2.6a and Figure 2.6b visualise discretisation by defining the range of the intervals and the number of samples per interval, respectively.



(a) Discretisation with constant intervals



(b) Discretisation with a constant number of samples in each interval

Figure 2.6: Two different discretisation strategies

Encoding

Most ML algorithms do not support categorical attribute values by default [Cho20]. These are attributes that are comprised of textual values, e.g., colours such as *red*, *blue*, *yellow*, and *green*. To prepare these values to be handled by the ML algorithms, they have to be *encoded*, that is, converted to numerical values.

Resampling

Training supervised ML algorithms with an imbalanced dataset will reward the algorithm for classifying samples in the majority class. This favouritism towards one of the classes is known as bias [MMS+21]. Several mitigation strategies are suggested to avoid training the models to favour one of the classes in the classification. This includes methods such as undersampling and oversampling, which are suggested by Hassan and Raja et al. [HRA+22]. Undersampling and oversampling work by ignoring samples in the majority class and duplicating samples in the minority class, respectively. This generates a balanced dataset with an equal number of samples of each class. The dataset can then be used for training without adding bias towards the dominant class [HRA+22].

2.3.3 Machine Learning Challenges

Although ML presents great opportunities, there are challenges that must be addressed when using ML algorithms. In relation to supervised ML, the problem of overfitting was explained in the previous section. Also, common preprocessing techniques mitigating dataset problematics were discussed. Here the No Free Lunch (NFL) theorem is presented.

Theorem 2.1. *The No Free Lunch (NFL) theorem states that for any machine learning algorithm, the elevated performance over one class of problems is offset by the performance over another class [WM97; Wik23d].*

The NFL theorem, as stated above, states that there is no one ML algorithm that, for any given problem, will outperform all other algorithms [Lis14]. This is explained by the way ML algorithms generate a simplified representation of reality while overlooking details. These simplifications are based on assumptions that hold for some situations but may fail for others. In turn, this implies that one should implement several models to obtain the best possible result in supervised learning [Lis14].

2.3.4 Evaluation Metrics

Evaluation metrics are typically used in two stages of classification problems. First, they are used to evaluate the performance of different parameter settings for an implementation of a classifier. This facilitates the optimal selection of hyperparameters to obtain the best classifier for the trained problem. Then the evaluation metrics are used to measure the effectiveness of the produced classifier when it is run with the previously unseen test data [HS15].

Evaluating the classifiers with relevant metrics is essential, as several common evaluation metrics have weaknesses. This can be related to the dataset if the dataset is imbalanced, particularly if the user is assessing the metric score for the majority class. There are also tradeoffs between different evaluation metrics, where increasing one score may lower another [BG16]. Different metrics are used to evaluate different characteristics of the classifier. The evaluation metrics can be categorised into three types, which are *threshold*, *probability*, and *ranking* metrics. The threshold and ranking metrics are the most commonly used [HS15]. Threshold metrics are derived from the classification’s confusion matrix and are used to assess the classifiers in this thesis. This thesis uses evaluation metrics both to fine-tune the classifiers’ hyperparameters and to assess the classification’s performance. A theoretical description of the used evaluation metrics is presented in Chapter 4. Also, the evaluation metrics are assessed along with the results of the experiment in Chapter 6.

Chapter 3

Related Work

Following the introduction, description of the problem area, and relevant background knowledge in the previous chapters, this chapter will assess related work. State of the art in website classification, as well as vital mechanisms which are used as building blocks to perform classification, will be discussed. The chapter is divided into subsections, each covering separate topics. In particular, the chapter covers related work using content-based classification in Section 3.1 and relevant literature focusing on infrastructure-based classification in Section 3.2. The infrastructure-based classification is further divided into work based on WHOIS and SSL certificate information. This chapter is based on findings from the pre-project [Bak22] and additional literature.

3.1 Content Classification

One of the major approaches to website classification is based on detecting and classifying the content they hold. Content classification can be based on several factors, and this section will present methods based on hashing, website appearance, and keyword searches. Albeit hosting illegal content, research has shown that criminals do not try to hide their intentions or content on their websites [WBG16]. Westlake and Bouchard et al. also showed that websites hosting illegal content, in their case CE material, are no more likely to be discontinued after a 14-month follow-up. This further facilitates content-based classification [Bak22; WBG16]. This section will provide insight into detection methods based on hashing, a mechanism also used by JARM, which is used as a feature later in this thesis. The hashing-based classification is followed by an introduction to appearance-based detection, which focuses on vital files of the websites. Finally, a section covering keyword-based detection presents a different approach to content-based detection.

3.1.1 Hashing

A hash function is a function which maps input data of arbitrary size to a fixed size output called the hash value [Wik23b]. There are many different ways to create a hash function, some of which are mathematically complicated. This is intentionally left out of this thesis as it only focuses on the overall hashing functionality. The hash function is usually one-way, meaning it is impossible to revert the function and thus obtain the original data from the hash value. Other properties of hash functions include collision resistance, meaning that different input data results in different hash values. Also, the hash function must be deterministic, meaning that the same input data will always result in the same hash value. Hashing functions should also be fast to compute and thus provide a fast and effective way to check whether data is similar [Wik23b]. An illustration of a hash function mapping the names “John” and “Wayne” to the binary values “00”, “01”, and “10” is provided in Figure 3.1. Note how the same name results in the same hash (deterministic property), and how different input data yields different hash values (collision resistance).

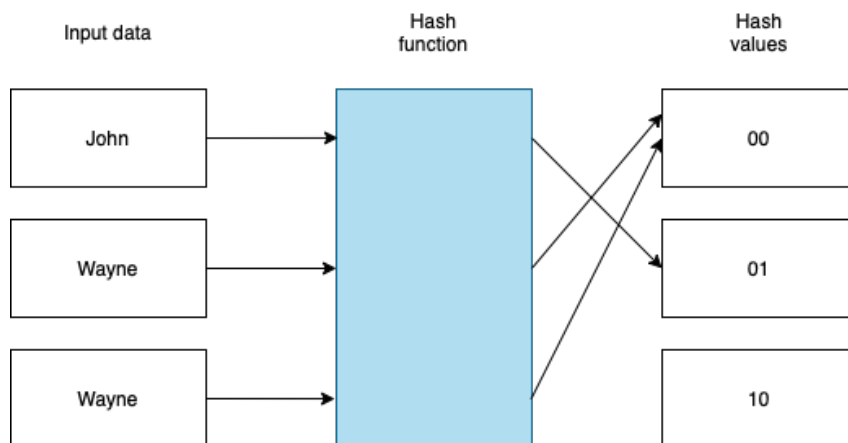


Figure 3.1: A simple hash function mapping names to binary values [Wik23b]

Hashing website content can be used to detect exact duplicates of content. This is the founding idea for the Interpol Base Line List, which contains the hash value of pictures known to be CE material [IntBL]. Comparing the content present on a website with this database can thus reveal duplicate images. This way of detecting CE material is further supported by Westlake and Bouchard et al. who state that if a site contains CE material, the inclusion of already-known content on the site is likely [WBG16].

However, detection based on exact duplicates faces some limitations. Due to the nature of hashing, a slight alteration to the input data results in a completely

different hash value. This is known as the avalanche effect [Hof10], which implies that adding a small noise filter or a slight rotation of an image would fool a detection mechanism based solely on hashing and exact duplicates. Such an attack could therefore avoid detection of already known CE material contained in the Base Line List. However, because of the topic’s severity, several actors have proposed methods to mitigate this limitation.

Microsoft has created a software called PhotoDNA, which uses a robust hashing method [PDNA09]. Robust hashing is a hashing method that is resistant to alterations in the content. Hashing robustness can be achieved through different approaches. Venkatesan and Koon et al. [VKJM00] proposed a method in which the image was split into different sections. The sections were then treated as image features, which are robust against malicious image alterations. To further improve robustness, the features were fed as input parameters in an error-correcting code to generate the hash value. This approach proved robust to a number of image alterations, such as rotation, cropping, scaling, and JPEG compression [VKJM00].

Wardman and Warner [WW08] performed a study proposing a method to identify phishing websites based on the MD5 hash of the content on the pages. This included content visually present for the end user, such as pictures, but also instrumental files such as Cascading Style Sheets (CSS) and JavaScript files. The study revealed an interesting observation about website creation; criminals often use webkits to generate websites in an effort to reduce the time spent creating websites [WW08]. This would yield similar websites when used. Furthermore, the interval for creation times of the websites would be limited. Together, these findings present opportunities for website classification.

3.1.2 Website Appearance

The appearance of a website is one of the most critical factors when deceiving users. A website with a professional appearance and grammatically correct language will look more legit to a targeted user. The appearance of websites is primarily controlled by the CSS file. These, therefore, serve as the basis for detecting phishing websites in the study “Phishing-alarm: Robust and efficient phishing detection via page component similarity” by Mao and Tian et al. [MTL+17]. The study examined the CSS files to quantify the visual similarity of different elements on the website. To determine the similarity, the study proposed an algorithm that assessed the visual characteristics of two websites and thereby computed their similarity score. The detection mechanism was further developed into a prototype extension for Google Chrome, and its effectiveness was evaluated using real-world phishing samples [MTL+17].

Another approach to website appearance detection is to assess the HyperText

Markup Language (HTML) source code. This approach was suggested by Roopak and Thomas [RT14]. They matched the HTML source code and computed the cosine similarity of the textual content on the sites. The source code matching was performed by attribute matching of the HTML tags and showed promising results using only limited computational power. However, the study identified obfuscation, i.e., tampering with the code to hide its behaviour, as a potential problem with the suggested method. As the method only assessed the HTML tags provided by the source code, an attacker may maliciously tamper with the HTML document, making it difficult to see what it does. Furthermore, by hiding the tags, the attacker may exploit the obfuscation weakness in Roopak and Thomas' proposed method.

As found by Wardman and Warner [WW08], phishing webpages are often generated using automated kits. This is supported by Feng and Qiao et al. [FQYZ22]. They investigated the possibility of grouping webpages belonging to the same family through a homology analysis of the webpage structure. Thereby, they could detect phishing websites. The study proposed a method that extracted structural features from the website. Then a similarity calculation would be done to facilitate clustering. The clusters were labelled with known webpage labels before the model was tested with unseen test samples. Their proposed method showed a good detection effect and high efficiency compared to other phishing detection methods based on structure clustering [FQYZ22].

3.1.3 Keywords

As stated in the introduction of this chapter, criminals make little effort to hide their intentions online [WBG16]. This facilitates the approach of keyword searches for detecting illegal content online and is an important method for detecting CE material [WBG16]. Furthermore, keyword search has been used to cluster similar websites to detect duplicates and near-duplicates by Broder and Glassman et al. [BGMZ97].

Broder and Glassman et al. [BGMZ97] developed a method to measure syntactical similarity. This was done by assessing the resemblance and containment in the comparison of two documents, A and B . The measures were computed by analysing each document as a canonical sequence of tokens, essentially shingling the words together. The study further compared different documents across the Internet by clustering. This was done by creating lists of potential shingles and documents and assessing whether a certain threshold was reached for resemblance. Broder and Glassman et al. found that their study enabled syntactical relations between documents to be discovered and envisioned a use case to track URLs over time [BGMZ97].

An approach based on keyword search is the Term Frequency-Inverse Document Frequency (TF-IDF) method, which identifies important keywords for the assessed

document [Wik23e]. This method evaluates a word’s importance in the document compared to other documents. This way, the method can differentiate keywords from commonly used words, which will feature in both the assessed document and other documents. The method may thus identify particularly important words in the examined document. The approach was used by Pang and Yao et al. [PYL+20], who combined this with HowNet [DD03] to identify lexical semantic similarity. Thus, their approach identified keywords and associated synonyms used in the document.

3.2 Infrastructure

Detection based on infrastructure features presents great opportunities for the early detection and takedown of illegal websites. Whereas content-based approaches require content to be available, thus also available for criminal actors, many infrastructure features can be generated even before the website is released [HHK+20]. Infrastructure features can be based on several elements, e.g., Ramachandran and Feamster [RF06] suggested inspecting the network-level behaviour of spammers. They argued that while the spammers may change the contents of the emails they send, network patterns in their network properties will persist. The study included features such as traceroutes, IP address space, and TCP fingerprints to find principles and guidelines for spam filters [RF06; Bak22]. It is also possible to extract other infrastructure-based features. The following subsections present work based on WHOIS information and SSL certificates in more detail.

3.2.1 WHOIS

WHOIS information can be used to generate features to detect malicious websites. One study based on this was performed by Cheng and Chai et al. [CCZ+22]. The study was motivated by early detection, which they encourage as malicious website detection often occurs after the crime has been committed. Their study explored identification based on various features from WHOIS by detecting irregularities at the time of domain creation. In particular, features were generated based on consistency, integrity, and validity. The features thus investigated whether the provided geographic information was consistent. This included investigating whether telephone numbers, postal codes, and registration addresses correlated. The integrity feature assessed whether all fields of the WHOIS record were covered or if information, intentionally or unintentionally, was left out in the registration process. Finally, validity checked whether the provided values were valid, e.g., whether the stated postal code existed or whether the provided telephone number was a valid telephone number. The study focused on Chinese websites and thus only assessed geographical information related to Chinese-registered websites [CCZ+22; Bak22]. Furthermore, the Chinese WHOIS records were not restricted by the European GDPR legislation, thus, generating features based on personal information was feasible.

WHOIS information can also be combined with other information to detect malicious domains. Kuyama, Kakizaki, and Sasaki [KKS16] combined Domain Name Server (DNS) information and WHOIS to detect malicious command and control servers and their associated domains. The information extracted from the WHOIS records included personal information, such as who was the technical contact person, and general information, such as the registrar. More details on the contact persons were also obtained. This included information such as address, phone number, and postal code. The study used a supervised ML algorithm, Support Vector Machine (SVM), to conduct a binary classification of the domains [KKS16].

Extracting personal information, as described above, has ethical aspects that must be addressed. This thesis includes a chapter devoted to ethical considerations in Chapter 7. Information about contact persons has also been greatly reduced following the implementation of GDPR. Lu and Zhang et al. [LLZ+21] performed a large-scale measurement study to investigate the impact of data regulation on WHOIS usage in security applications. The surveyed papers covered topics such as spam detection, cybercrime analysis, and domain security. The study revealed that 69% of the papers relied on now redacted information [LLZ+21]. Restricting information availability will hamper these detection methods. This thesis will, however, use only available information investigating whether that can be used for classification.

3.2.2 SSL Certificate

Hounsel and Holland et al. [HHK+20] performed a study using a number of infrastructure-based features, including features based on SSL certificates. They used a multi-label classifier to label the data as disinformation, authentic news, or “other”. Features used in the classification process included information about how many domains the certificate covered and general information about the certificate, such as details about the issuer and whether the certificate was self-signed. The project was deployed and tested for real-time detection of disinformation and showed promising results for the new features they suggested [HHK+20].

Adding encryption to HTTP websites was initially a sign of benign pages. However, with the rise of freely available CAs, many phishing websites have adopted HTTPS to appear legitimate and evade conventional detection mechanisms [SWO+21]. Sakurai and Watanabe et al. [SWO+21] used the presence of SSL certificates to their advantage, investigating patterns in the certificates to identify phishing websites. The identification was performed by clustering attributes from the certificates and generating templates from the clusters. The study found that the templates could be used to discover phishing websites with a low false positive rate. Furthermore, the templates revealed information about phishing website generation, and combined with clusters with a large number of similar certificates, the study found indications

of process automation in website generation [SWO+21].

SSL certificates were also used by Torroledo, Bahnsen, and Camacho [TCB18], who proposed a method to detect malicious use of certificates using deep neural networks. Their study was motivated by the increased use of SSL certificates on malicious websites hosting phishing attacks and malware. Furthermore, a recent survey revealed that 82% of Internet users said that they thought a website was safe when it displayed a “*Secure*” symbol in front of the URL in the browser. This symbol indicates the use of use SSL, which is no longer used only for benign websites. Torroledo, Bahnsen, and Camacho suggested several features based on information contained in the certificates to facilitate detection. While processing the certificates, they also found that several of the malicious certificates were missing information. The classification was done with two deep neural network models, one classifying malware and the other phishing. Each of which had good performance [TCB18].

Chapter 4

Methodology

This chapter will describe the general methodology used to perform the experiment in this thesis. In particular, this covers dataset acquisition, background on the implemented ML algorithms, and the proposed strategy to answer the stated research questions. The selected method is justified and compared to the explored alternatives. A comparison will also be made to the initial pre-project [Bak22], which did propose a different methodology than this thesis. It is worth noting that the chapter only provides a general overview of the methodology, while Chapter 5 will describe how the experiment was conducted in detail.

4.1 Choice of Methods

In the pre-project, the author explored several sources of information about domains. In addition to information from WHOIS, Passive Domain Name Server (PDNS) data was considered a potential source of features [Bak22]. However, investigating the different providers of PDNS data, such as Mnemonic [Mnemonic], revealed that acquiring large amounts of data could prove difficult. This is due to query limitations in their Application Programming Interfaces (APIs). As a replacement, SSL certificate details would be used as additional attributes to the information extracted from WHOIS records.

There are several ways to obtain WHOIS information about domains. As explored in the pre-project [Bak22], both VirusTotal [VTapi] and DomainTools [DomainTools] provide APIs which can be used. Also, the WHOIS protocol, as described in Section 2.2.1, is implemented as a command line tool [WHOIS]. VirusTotal and DomainTools provide similar services, but whereas VirusTotal is free of charge, the author did not successfully find any free API key to the DomainTools API. The benefit of using VirusTotal was further manifested as VirusTotal maintains a Python library called vt-py [vt-py]. This would facilitate seamless integration between feature engineering and the ML algorithms, also implemented using Python. The public API key from

VirusTotal had query limitations. However, reaching out to VirusTotal and explaining the Master’s thesis project, they kindly provided us with an academic API key with fewer restrictions. This allowed a larger dataset to be used in the experiment.

4.2 Dataset

Several ways to acquire a dataset were discussed in the pre-project [Bak22]. E.g., the author suggested creating a new dataset based on blacklists of known malicious websites. Such blacklists were examined in the early stages of the thesis, among them was URLhaus by abuse.ch [URLhaus]. URLhaus contains lists of malicious URLs and can be downloaded as Comma-separated values (csv) files which commonly are used for ML dataset formats. However, this approach would include little useful information about the domains, as the blacklist only provides the URL and the malicious label. This would, thus, require all additional information to be obtained from other sources. Another problem is the lack of benign websites in the list. A list of benignly labelled websites would have to be added to the dataset before it could be used in the experiment. Also, a manual approach of finding benign and malicious websites has several limitations, e.g., manually determining the label of websites is error-prone. Besides, this would require a labelling scheme to facilitate the labelling process, which would have to be created. Furthermore, manual labour is time-consuming and would imply severe limitations on the number of feasible websites in the dataset.

A dataset with both benign and malicious websites from Kaggle [Kaggle] was also assessed. It was beneficial because it contained both malicious and benign websites, but the URLs were all anonymised, making it impossible to acquire further information about the domains. The supervisor suggested a dataset from Data in Brief [Sin20]. One of the major advantages of the dataset was that it provided both the URL and IP address of the website. This allowed different queries to be sent to VirusTotal. Furthermore, the dataset contained the geographical location of the IP addresses, which was suggested as an additional source of information should WHOIS prove deficient in the pre-project [Bak22]. Given the labelled dataset, a change from an unsupervised clustering method, which was the suggested methodology in the pre-project [Bak22], to a supervised learning approach was made. A thorough assessment of the implemented supervised ML algorithms will follow in the next section, while the exact implementation details, such as hyperparameter selections, will be presented in Chapter 5.

4.3 Machine Learning Algorithms

This section will describe the ML algorithms used in the experiment, i.e., Random Forest, AdaBoost, Naive Bayes, Quadratic Discriminant Analysis (QDA), and Multi-

Layer Perceptron (MLP). Note that several algorithms are included to allow a comparison. This is directly related to the NFL theorem as described in Theorem 2.1 in Chapter 2. A brief introduction to some algorithms that were tested but not implemented, along with an explanation for the choice to scrap them, are also included in this section. Finally, an acknowledgement of important Python libraries used in the experiment concludes this chapter.

4.3.1 Implemented Machine Learning Algorithms

This subsection presents the implemented ML algorithms used in the experiment. It will justify the selected models by assessing their advantages and drawbacks. As this chapter only presents the general methodology, exact implementation details, such as hyperparameter selection for the relevant models, are presented in Chapter 5.

Random Forest

Random Forest is an ensemble method combining several base estimators' predictions to improve performance [Sci:ens]. The Random Forest model enhances the result of one base estimator by averaging the predictions from several randomised decision trees. The added randomness mitigates the overfitting tendencies of regular decision trees [Sci:ens]. As later discussed, this argues against implementing regular decision trees. From the Law of Large Numbers (LLN), it follows that Random Forest models do not overfit [Bre01]. The strong LLN states that the sample average almost surely converges to the expected value (μ) when the number of trials approaches infinity [Wik23c]. This is formalised in Equation 4.1. Furthermore, the Random Forest model can run in parallel, utilising all cores on the server and thereby reducing the overall time spent performing the classification. This is further beneficial as the dataset used in this experiment contains hundreds of thousands of samples.

$$\lim_{n \rightarrow \infty} \bar{X}_n \rightarrow \mu \quad (4.1)$$

Another benefit of the Random Forest model is that it is useful for extracting essential features. It thus provides direct use for the research questions in this thesis. Also, the model has good performance with imbalanced datasets [Gup20]. The model is, however, not flawless. The features must have some predictive power, i.e., they cannot be irrelevant for the classification [Gup20]. Also, the model is complex, and executing it in parallel makes it challenging to understand precisely what is happening. The Random Forest algorithm is used in related work with good scores [CCZ+22].

AdaBoost

The second ensemble method included in the experiment is the adaptive boosting algorithm, AdaBoost [FS96]. The boosting ensemble method works by combining classifiers generated by weak learning algorithms. The weak learners are classifiers which perform slightly better than random guessing [Bro21; Sci:ens]. To improve them, the boosting algorithm runs the weak classifiers on different sections of the training data, each iteration increasing the weights of the wrongly classified samples from the previous run. This way, the boosting algorithm reduces the bias of the weak learner and thereby improves the overall performance. Boosting methods can thus improve classification in problems where some samples are more challenging to classify than others. The AdaBoost algorithm was included in this experiment as it has proven useful in relevant related work, such as Cheng and Chai et al. [CCZ+22]. Furthermore, AdaBoost is, albeit not mathematically proven, less prone to overfitting [Kur20]. AdaBoost does not require hyperparameter tuning and thus presents an out-of-the-box solution which is fast to implement, which serves as another argument for inclusion.

Naive Bayes

Naive Bayes is a supervised ML algorithm based on Bayes' theorem. The theorem is stated in Theorem 4.1, and is mathematically expressed as Equation 4.2. The ML algorithm applies a simplified version of the theorem; whereas the theorem requires independent probabilities between events A and B , the Naive Bayes model assumes independence between the features in the dataset [Sci:NB]. This fundamental assumption is also one of the major drawbacks of the model, as feature independence does not hold for most real-world cases [Gup20]. Still, the model is used for classification in cases such as spam filtering of emails, with satisfactory results [Gup20]. The model is included in this experiment due to its performance in spam detection and beneficial qualities such as its speed. Also, the model can deal with large datasets [Sci:NB; Gup20], making it suitable for this experiment. The model is also simple to implement and does not require fine-tuning of hyperparameters [STS16], which makes it easy to implement and assess the initial performance and relevance of the model. A choice can thus be made early in the process to include the model in further experiments or abandon it for more suitable candidates.

Theorem 4.1. *The probability of an event is dependent on prior knowledge that might be related to the event [Wik23a]. This is mathematically expressed as follows:*

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (4.2)$$

Quadratic Discriminant Analysis

QDA is a model which seeks to maximise the distance between the target labels. It does so by combining the attributes in such a way that the differences stand out [STS16]. The model is closely related to Gaussian Naive Bayes. If the model assumes conditionally independent inputs in each class, the resulting classifier will be equivalent to Gaussian Naive Bayes [Sci:QDA]. The inclusion of the model in this experiment is due to promising results in the early implementation testing of the experiment. Furthermore, there are no hyperparameters to tune in the model, and the model has shown good results in practice, making it an attractive model to use [Sci:QDA]. A visualisation of QDA with quadratic boundaries is shown in Figure 4.1. Note how the model is able to separate the different classes using the quadratic boundaries. This makes the model more flexible than a linear boundary model such as Linear Discriminant Analysis [Sci:QDA].

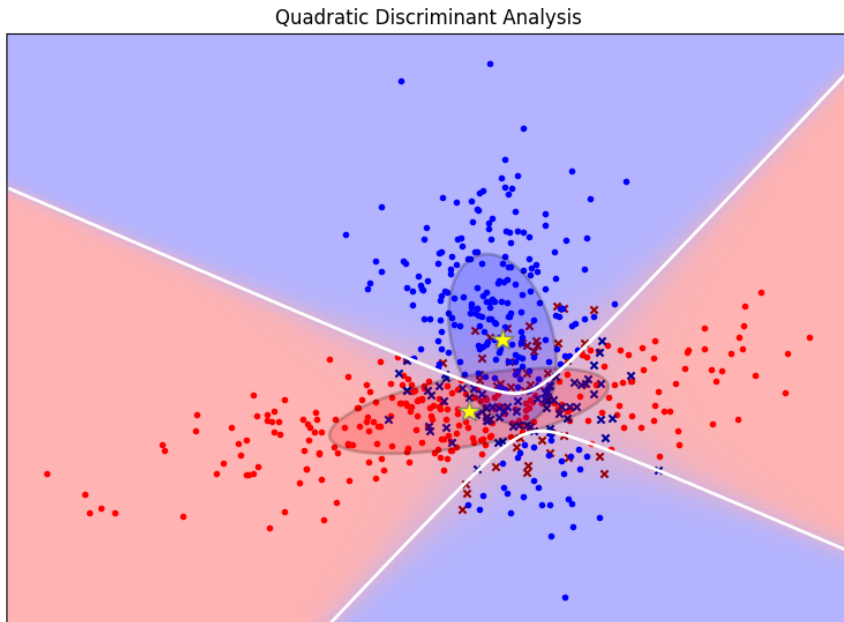


Figure 4.1: A Quadratic Discriminant Analysis model with quadratic boundaries [Sci:QDA]

Neural Network

A neural network model that was included in the experiment is the MLP classifier. The MLP model learns a function that maps the input features through a fixed

number of hidden layers to an output. The first hidden layer takes the features as input parameters before the next layer transforms the result of the previous layer by assigning them weights [Sci:MLP]. An MLP model may have several hidden layers. The hidden layer structure, with one hidden layer, is illustrated in Figure 4.2. There, the features X are fed to the first layer, a , which assigns weights and forwards the features to the final output function, $f(X)$. The MLP model is included in this experiment as it showed promising results in the initial testing, as well as its capabilities to ignore noise and robustness to irrelevant input [STS16; Kot07]. The model does, however, bring some challenges to the experiment. It requires hyperparameter tuning, and the performance of the classifier is highly dependent on the parameter selection and feature scaling [Sci:MLP]. Furthermore, determining the number of hidden layers is difficult [STS16]. Mitigating these challenges is done with a *randomised search* explained in Section 5.3 and scaling techniques to even the values of the features.

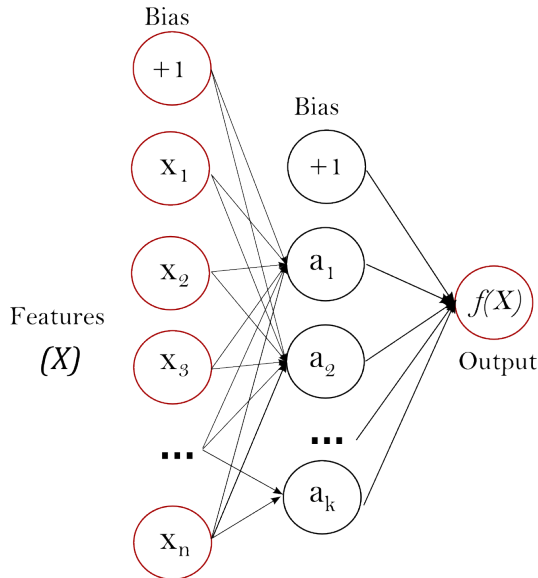


Figure 4.2: A Multi-Layer Perceptron model with one hidden layer [Sci:MLP]

4.3.2 Validation Techniques

This subsection describes the evaluation metrics used to assess the performance of the different classifiers implemented in the experiment. The section will introduce the metrics, explain what they represent, and justify them as validation techniques in this thesis. The four metrics *accuracy*, *precision*, *recall*, and *F1-score* are included in this experiment, all of which are different threshold metrics derived from the confusion matrix of the model. I.e., the metrics are based on various combinations of

True Positive (TP)-, False Positive (FP)-, True Negative (TN)-, and False Negative (FN). Table 4.1 presents a description of the used acronyms.

Table 4.1: Description of acronyms used to derive classification evaluation metrics

Acronym	Description
True Positive (TP)	Number of correctly positive labelled samples
False Positive (FP)	Number of incorrectly positive labelled samples
True Negative (TN)	Number of correctly negative labelled samples
False Negative (FN)	Number of incorrectly negative labelled samples

Accuracy

Accuracy is the most used evaluation metric for binary classification problems such as this experiment [HS15; HRA+22]. The accuracy score presents the quality of the classification by assessing the percentage of correct predictions over the total number of instances. Mathematically, this is formulated in Equation 4.3.

$$Accuracy(acc) = \frac{TP + TN}{TP + FP + TN + FN} \quad (4.3)$$

Precision

Precision measures the correctly predicted positive patterns in the positive class. It focuses on the positive rates, i.e., TP and FP, and is mathematically expressed in Equation 4.4 [HS15; HRA+22].

$$Precision(p) = \frac{TP}{TP + FP} \quad (4.4)$$

Recall

As precision, recall focuses on the correctly labelled samples and measures the fraction of correctly classified positive samples [HS15; HRA+22]. Equation 4.5 expresses the mathematical computation of recall.

$$Recall(r) = \frac{TP}{TP + FN} \quad (4.5)$$

F1-score

F1-score represents the harmonic mean between precision and recall [HS15; HRA+22]. A mathematical expression of the F1-score is presented in Equation 4.6.

$$F1 - score = 2 * \frac{p * r}{p + r} \quad (4.6)$$

Confusion Matrix

The confusion matrix visually presents the TP-, FP-, TN-, and FN rates by plotting the number of correctly and falsely labelled samples in the classification. A confusion matrix example is shown in Figure 4.3. Considering the *malicious* class as the positive class, the figure shows that the classifier has two TP samples and one FP sample. It also has two TN samples and one FN sample.

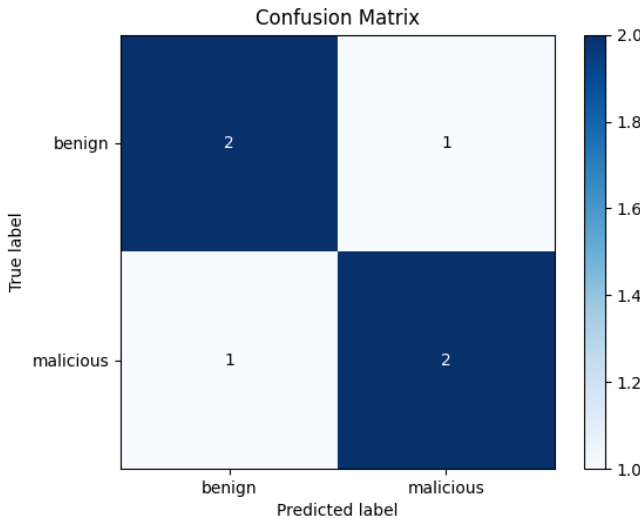


Figure 4.3: A confusion matrix displaying the results from a malicious/benign classification

4.3.3 Other Machine Learning Models

In addition to the aforementioned models used in the experiment and whose results are presented in Chapter 6, some models were unsuitable for this experiment. This subsection will give a brief explanation as to why some algorithms were excluded from the experiment.

k-Nearest Neighbours

k-Nearest Neighbours (k-NN) was initially implemented but performed poorly. A further assessment of the ML model revealed that it is unsuitable for imbalanced data [Gup20]. The poor results were thereby explained by the imbalanced dataset

used in this experiment. Furthermore, the model is slow for large datasets [Gup20]. Even with the implemented specifications of this experiment’s server, as shown in Table 5.1, the model required much time to process.

Support Vector Machine

As with k-NN, the SVM models are slow for large datasets. Furthermore, they have poor performance with overlapped classes [Gup20]. Initial implementations showed that the model required too much time to process. And with no guarantee that the classes in our dataset are clearly separated, we decided to abandon this model.

Decision Tree

Decision trees are prone to overfitting, creating overly complex trees that do not generalise well [Gup20; Sci:DT]. As a result, the training data will have excellent performance, while the tree will struggle to classify the test data correctly. Initial test implementation revealed that this model struggled to separate the two classes in the dataset, and the model was not further included in the experiment.

4.4 Research Questions

A description of how the author envisioned answering the stated research questions is presented in this section. The section will also state what types of results are expected from the experiment in relation to the research questions.

4.4.1 Machine Learning Evaluation

The evaluation metrics described above will be used to assess the implemented ML algorithms. Also, the two resampling methods random undersampling and random oversampling, will be used and compared in the performance evaluation. A conclusion of which algorithm performs better in the classification will answer research question 1. Including several ML algorithms in the study can reveal one, or several, that outperform the others. However, following the NFL theorem (Theorem 2.1) described in Section 2.3.3, ML algorithms do not perform equally well on different problems. It may thus be challenging to find a suitable ML model [Sar21]. Any hard conclusion as to what algorithm is best, in general, will therefore be impossible. However, the study may reveal algorithm(s) better suited for malicious website classification. They should thus be considered implemented in future related work.

4.4.2 Feature Importance

Feature importance can be derived from the different ML classifiers to assess which features are most prominent in the classification process. This will be used to evaluate

which features are most important in malicious website classification. The feature assessment answers research question 2. Feature importance can be measured in Mean Decrease in Impurity (MDI), which describes the mean of the trees' improvement when splitting is done on the assessed feature [Loe20].

Based on the results, we hope to discover important infrastructure-based features. Furthermore, we want to investigate whether the now restricted information contained in WHOIS records may still have a role in malicious website classification.

Chapter 5

Experiments

Following the general methodology presented in the previous chapter, this chapter will present the experiment in more detail. It will first describe the experiment environment setup in Section 5.1. In particular, this explains the specifications and practical aspects of handling the remote server. Then, an assessment of the dataset follows in Section 5.2. This section will describe the dataset used, the preprocessing steps that had to be implemented, and the final features that were generated and used. Finally, a specification of the ML implementation is provided in Section 5.3, concluding this chapter.

5.1 Environment Setup

Given the GDPR regulations, any research project conducted at Norwegian University of Science and Technology (NTNU) involving personal information must not be performed on a personal computer [NTNUa]. With this restriction, a server was created with NTNU OpenStack [NTNUc]. This allowed the thesis to be securely conducted, with data only stored on a remote NTNU server. The server was accessed by using a Virtual Private Network (VPN) connection to the NTNU network and a Secure Shell (SSH) connection to the server.

The OpenStack project provides different possibilities for specifications when requesting a server. For this thesis, a request for a server hosted at NTNU SkyHigh with the technical specifications shown in Table 5.1 was submitted before the experiment started. We requested 8 Central Processing Unit (CPU) cores to facilitate multiprocessing. The amount of Random Access Memory (RAM) was set to 64GB to ensure that the server would not run out of memory while processing the large dataset in the Python code. Finally, the amount of physical storage was set to 1TB. This allowed different copies of the dataset to be stored while facilitating temporary writes to file as the code ran. The temporary files were implemented as a security measure to reduce the impact of a code crash while running.

Table 5.1: Technical specifications for the NTNU SkyHigh server

Specification	Request
Amount of CPU cores	8
Amount of RAM (GB)	64
Amount of storage (GB)	1000

To ensure that no personal data was stored on the author’s private computer, the dataset was downloaded directly to the server using `wget` [NSR22]. `Wget` enables file downloads from the Internet through protocols such as HTTP and File Transfer Protocol (FTP). One significant advantage of `wget` is that it is non-interactive, meaning that the user can start a download and log off while `wget` finishes the job. This is particularly practical when handling large file downloads and limitations in network capacity [NSR22]. With the limited Graphical User Interface (GUI) of the server through the SSH connection, the author wrote all code locally and transferred the code to the server using the Secure Copy Protocol (SCP) [RY22]. The SCP protocol is used to securely transfer files between hosts on a network. The protocol ensures security by using the same authentication as a login through an SSH connection, and it is initiated with a similar command [RY22].

An overview of the general methodology to obtain information from VirusTotal is visualised in Figure 5.1. The figure shows how the client program interacts with the VirusTotal API, which is responsible for communicating with the VirusTotal server. The client program is the code used in this thesis, which was written locally and tested on test datasets. For testing the acquisition of information from VirusTotal, different URLs were sent to the VirusTotal API. However, as this is a repetitive process, no separate dataset was used for this code testing. Instead, only a few selected URLs were used to test the functionality.

A relevant test dataset hosted on Kaggle [Kaggle], was used to test the ML implementation and feature preprocessing code. This dataset contained relevant information such as WHOIS country and WHOIS registration date. However, unlike the dataset used in the full version of the experiment, the testing dataset did not contain any personal information. Furthermore, the URLs were all anonymised, resulting in a dataset that could be stored and processed on a private computer. As the dataset was only used for code quality assurance before it was implemented, it is not described in further detail. The following section covers the dataset used in the experiment in detail.



Figure 5.1: Overview of the method to acquire information about domains

5.2 Dataset

Previous research on malicious websites has used several different ways to acquire a suitable dataset, many of which created their own [AAA+22]. This thesis used a publicly available dataset created by A. K. Singh, published in Data in Brief [Sin20]. The dataset contains websites labelled as “good” or “bad”, as well as information about the domains such as URL, IP address, and raw web content. Table 5.2 shows all the attributes present in the dataset. A. K. Singh argued that at the time of creation, no similar dataset existed. To create the dataset, Singh used a webcrawler called MalCrawler [SG17].

MalCrawler, like a regular webcrawler, follows hyperlinks from the given start URLs but differs from a regular webcrawler by seeking more malicious websites. Singh and Goyal suggested several approaches to encounter more malicious websites. One approach was to start the crawl with a known malicious seed. This approach will encounter more malicious pages than a random web crawl, as malicious websites are more likely to host links to other malicious sites [SG17]. Singh and Goyal also suggested seeking out dynamic content, such as sites with JavaScript. Dynamic content will increase the attack surface of websites with malicious intentions by facilitating attacks such as Cross-Site Scripting (XSS).

Webcrawlers are not regular Internet users, and malicious websites may implement detection mechanisms to avoid crawlers. Such mechanisms enable the malicious site to detect if the requester is a real user, which will be redirected to the malicious content. However, if the user’s behaviour resembles that of a webcrawler, it will be redirected to a benign site. This client-platform conditional redirection is known as cloaking [SG17]. MalCrawler implements countermeasures to known crawler

detection mechanisms by emulating a regular user. It is able to emulate different browsers and combat cloaking by setting the HTTP user agent field to a browser name.

The dataset A. K. Singh created contains 1 561 934 URLs, split into test and training datasets. In total, there are 35 315 samples labelled bad (malicious). Singh derived the labels from queries to the Google Safe Browsing API [Goo23; Sin20]. A URL is deemed malicious if it contains malware, XSS, code injection, drive-by download, or if it resembles behaviour like phishing [Sin20]. Although MalCrawler was created to detect more malicious websites, the dataset has been modified to reflect the disproportion between malicious and benign websites online. This is reflected by the substantial imbalance of the dataset, containing significantly more samples with a benign label. The modification was done to better reflect the Internet and thus provide a more realistic representation of online websites.

The dataset is relatively new, being assembled between November 2019 and March 2020. However, as this is three years ago, the dataset will likely contain websites currently not in use. As indicated in Chapter 2, malicious websites are no more likely to be discontinued after a certain time period. As a result, implementation considerations had to be addressed in the code should the APIs fail to deliver any information about the requested domain, both for malicious and benign sites. The experiment handles exceptions by adding “N/A”-values to the attributes and later handles these missing fields. The process of handling missing values will later be described in detail. Furthermore, other modifications had to be made before the dataset was ready for the experiment. The following subsection describes the preprocessing steps taken before the dataset could be used.

5.2.1 Feature Engineering

Albeit already containing information about the different domains, the dataset lacked information that this thesis wanted to use for classification. Therefore, the WHOIS records and SSL certificates of the websites were requested from the VirusTotal domain API through their Python library vt-py [vt-py]. The obtained WHOIS records and SSL certificates were then used to generate new features to conduct the experiment. This subsection will introduce the features present in the original dataset, what features are preserved to feature in this experiment, and what features are generated from the attained information.

Existing Information

The original dataset includes relevant information about the different URLs. The attributes were included in the dataset based on their relevance to malicious webpage classification in previous research [Sin20; SG19]. Table 5.2 presents all the attributes

Singh included in the dataset. This thesis wants to perform classification based on infrastructure features. Hereunder, the experiment aims at leveraging information from WHOIS records and SSL certificates. Thus, not all attributes present in the dataset are relevant to this experiment. Therefore, the attributes *url_len*, *js_len*, *js_obf_len*, and *content*, were all dropped from the dataset.

Some of the information contained in the dataset can be considered personal information, i.e., information that can be used to identify individuals. This includes information such as IP address and geographical location [NTNUa]. An assessment of ethical dilemmas regarding the use of personal information in research follows in Chapter 7. For this thesis, the IP addresses are relevant as they can be used to query the VirusTotal APIs to obtain more information about the associated domain. As shown in the following subsection, the VirusTotal APIs can be queried to return information about both IP addresses, domain names, and URLs.

Table 5.2: Attributes present in the used dataset [Sin20]

Attribute name	Data type	Attribute description
url	String	URL of the webpage
ip_add	String	IP address of the webpage
geo_loc	Categorical String	Name of the country based on IP address location
url_len	Numerical	Length of URL - count of characters in a URL
js_len	Numerical	Length of JavaScript code (in kB) in the webpage
js_obf_len	Numerical	Length of obfuscated JavaScript (in kB) in the webpage
tld	Categorical String	Top Level Domain of the webpage
who_is	Categorical String	Gives out whether the WHOIS information of the registered domain is complete or incomplete
https	Categorical String	Gives out whether the website uses HTTP or HTTPS protocol
content	Text	Raw web content of the webpage. Includes filtered and processed text and JavaScript code
label	Categorical String	Classification label categorising the webpage as malicious (bad) or benign (good)

Added Information

This experiment adds further details about the domains to the dataset to facilitate potential identification based on WHOIS and SSL certificate details. The information is retrieved from VirusTotal, who has APIs that take domain, URL, or IP address as query parameters. The response object contains different information depending on which API is queried. Table 5.3 provides a detailed overview of the relevant information available through the different APIs.

Table 5.3: Relevant attributes available through the different VirusTotal APIs

Attribute name	Attribute description	Present in API		
		Domain	URL	IP
as-owner	Owner of the Autonomous System to which the IP belongs			✓
continent	Continent where the IP is placed			✓
country	Country where the IP is placed			✓
jarm	The domain's JARM hash	✓	✓	✓
last-analysis-results	Results from URL scanners	✓	✓	✓
last-analysis-stats	Number of different results from URL scanners	✓	✓	✓
last-dns-records	The domain's DNS records on its last scan	✓		
last-final-url	The end URL in case of redirections		✓	
last-https-certificate	SSL object retrieved the last time the domain was analysed	✓		✓
redirection-chain	History of redirections		✓	
whois	WHOIS record	✓		✓

Singh and Goyal [SG19] found that redirection is often linked to maliciousness. Redirection information is available through the URL API. Here, both the redirection chain and the final URL after redirection can be found. Investigating redirection is indeed interesting information and possibly presents a way of identifying malicious websites. However, the author has excluded the information as part of this experiment

as it is not related to WHOIS or SSL certificates. An interesting thought is to follow the redirections and query the VirusTotal API with the final URL. This will be suggested in future work in Section 6.7.

While IP addresses can easily change to serve the same domain, a maliciously labelled domain cannot change without being discontinued. As an example, the IP address serving a malicious website at the time the dataset was assembled may now serve a completely different website. This, therefore, serves as an argument for performing queries to the domain API. Furthermore, the domain API provides more relevant information compared to the URL API, mainly by providing WHOIS records. Given the redundant values provided by the IP and URL APIs compared to the domain API, the domain API stand out as the most important API to query in this experiment.

The API key obtained from VirusTotal limits the possible requests that can be sent daily. With this limitation, a decision was made only to send one query per website. This excludes the possibility of sending a request to both the URL, IP, and domain APIs. Although it would be beneficial to attain the unique values each API provides, it is not feasible in this experiment due to the limitations. It should be noted, however, that the limitation of daily queries has been drastically reduced after VirusTotal granted us an academic API key to use in this thesis. The author acknowledges their contribution and cooperation in this thesis by raising the quota from 500 per day with a public API key to 20 000 per day with the academic key. This facilitated a greatly expanded dataset, which in turn will provide better credibility for the results of this experiment.

VirusTotal also includes the results from 70+ antivirus organisations' evaluation of the queried domain. The API documentation states that the results will group the domain into the four categories listed below [VTdom]. Initially, the plan was to use the results to classify domains into different classes of malicious websites, effectively creating a multiclass classification problem. This was also reflected in the research questions from the pre-project [Bak22]. However, initial testing of the experiment revealed that this information was deficient, and a model change to a binary classification was made.

1. Malicious
2. Phishing
3. Suspicious
4. Clean

Feature Extraction**Table 5.4:** Features used in this thesis

Attribute name	Data type	Attribute description
whois_creation_time	Int	WHOIS creation time
whois_expiration_time	Int	WHOIS expiration time
whois_updated_time	Int	WHOIS updated time
whois_updated	Boolean	True (1)/False (0) value whether the domain has updated its WHOIS record since creation
ssl_start_time	Int	Validity start time of the SSL certificate
ssl_end_time	Int	Validity end time of the SSL certificate
ssl_alg	Categorical String	Algorithm used to generate the certificate. Any of “RSA”, “DSA”, or “EC”
ssl_issuer_co	Categorical String	Issuer country of the SSL certificate
ssl_issuer_org	Categorical String	Issuer organisation of the SSL certificate
whois_reg_co	Categorical String	Country of the WHOIS registrant
geo_loc	Categorical String	Name of the country based on IP address location
bool_co	Boolean	Boolean showcasing whether the WHOIS registrant country differs from geographic location. Values 0 for difference, 1 for the same country.
registrar	Categorical String	Registrar of the domain
jarm	String	JARM hash of the domain server setup
cipher	String	Cipher details extracted from the JARM hash
tld	String	Top Level Domain of the domain
label	Categorical String	Classification label categorising the webpage as malicious (“bad”) or benign (“good”)

This section describes how the features were generated from the information provided by the queried API. All features used in this experiment are presented in Table 5.4, and a description of each feature follows in subsections here. After the feature description, a detailed description of the ML implementation concludes this chapter.

A new Python script was written to extract the wanted information from the acquired WHOIS records and SSL certificates. The script is, in all essence, based on regular expressions extracting the wanted information from the WHOIS and SSL columns in our newly assembled dataset. The columns are treated as strings. The script then creates new features based on the extracted information.

whois_creation_time, whois_expiration_time and whois_update_time

WHOIS time features are extracted from WHOIS records provided by the VirusTotal domain API object [VTwho; VTdom]. The features were extracted as date objects after matching the WHOIS string using regular expressions. A code snippet to illustrate how the matching was done is shown in Source Code 5.1. As the code snippet shows, the code tries to match variations of ways to write creation time. This was necessary to mitigate the lack of standardisation of WHOIS data format [LFS+15], as described in Chapter 2. A similar matching procedure was also executed for both expiration and update times.

After the data object was extracted, the date was processed to be converted to a Unix timestamp [Wik23f]. This step required more matching through regular expressions, as the extracted dates were written with variations in the WHOIS strings. Finally, after converting the times to Unix timestamps, the preprocessing step discretised them by clustering them into groups. WHOIS time features are included in this dataset as previous research found that domains registered close in time to a known malicious website are much more likely also to be malicious [ICB+12]. This is closely related to automatic website generation, as malicious actors generate large numbers of websites to improve their chances of reaching their target. Furthermore, this serves as an argument for grouping the samples which are close to each other, as discussed in the preprocessing step, Section 5.2.2.

whois_updated

The boolean value `whois_updated` showcases whether the WHOIS record is updated. The code checks if the WHOIS update time is “N/A” and adds a boolean value accordingly. The feature is included to investigate whether the lack of updates of the WHOIS record may imply malicious websites.

Source code 5.1 WHOIS creation time matching

```

if re.match('.*[C|c]reation [D|d]ate: .*', whois, re.DOTALL):
    whois_creation_date.append(
        (re.split('.*[C|c]reation [D|d]ate: ', whois, re.DOTALL)[1])
        .splitlines()[0])

elif re.match('.*[C|c]reate [D|d]ate: .*', whois, re.DOTALL):
    whois_creation_date.append(
        (re.split('[C|c]reate [D|d]ate: ', whois)[1])
        .splitlines()[0])

[...]

else:
    whois_creation_date.append('nan')

```

ssl_start_time and ssl_end_time

The SSL certificate was used to extract time values similarly to WHOIS. As shown in Source Code 5.2, the SSL times were matched with “not before” and “not after” to generate *ssl_start_time* and *ssl_end_time*, respectively. The SSL time features are included on the same basis as the WHOIS time features, i.e., close relevance in time for malicious websites [ICB+12].

Source code 5.2 SSL start and end time matching

```

if re.match(".*'validity': ", cert, re.DOTALL):
    ssl_validity_end.append(
        re.split(".*'not_after': '|'", cert, re.DOTALL)[1])

    ssl_validity_start.append(
        re.split(".*'not_before': '|'", cert, re.DOTALL)[1])

else:
    ssl_validity_end.append('nan')
    ssl_validity_start.append('nan')

```

ssl_alg

The used encryption algorithm was extracted from the SSL certificate. This had potentially three different values, namely Rivest-Shamir-Adleman (RSA), Digital

Signature Algorithm (DSA), and Elliptic Curve (EC) [VTssl]. The *ssl_alg* feature is included in this study to investigate whether the used encryption algorithm can indicate malicious websites.

ssl_issuer_co and ssl_issuer_org

Information about the issuer of the SSL certificate was also present in the certificate retrieved from VirusTotal [VTssl]. The issuer-related features in this thesis are country and organisation. The regular expressions used to extract this information are shown in Source Code 5.3. As with the *ssl_alg* feature, these are included to investigate the potential pattern between the SSL certificate issuer and maliciously labelled websites.

Source code 5.3 SSL certificate issuer information

```
if re.match(".*'issuer': {'C':", cert, re.DOTALL):
    ssl_issuer_co.append(
        re.split(".*'issuer': \{'C': '|'", cert, re.DOTALL)[1])
else: ssl_issuer_co.append('nan')

if re.match(".*'issuer':.*'O': ", cert, re.DOTALL):
    ssl_issuer_org.append(
        re.split(".*'issuer':.*'O': '|' [}|,]", cert, re.DOTALL)[1])
else: ssl_issuer_org.append('nan')
```

whois_reg_co

The only information directly related to the registrant of the domain used as a feature in this thesis is the *whois_reg_co* attribute, which represents the registrant's country. Otherwise, all personal information about registrants in the WHOIS records from VirusTotal is anonymised to protect the privacy of the registrants [VTwho]. This information was also extracted from the WHOIS string by using regular expressions. The feature is a standalone feature and is also used to generate the *bool_co* feature discussed below.

geo_loc

The geographical location related to the IP address was mapped and included as a feature in the original dataset. To map IP addresses and geographical locations, Singh [Sin20] used the GeoIP database available from Maxmind [GeoIP]. Due to limitations in time, this thesis has not prioritised verifying the mapping from Singh's study.

bool_co

The feature *bool_co* showcases whether the registrant is in the same country as the geographical location of the IP address. I.e., the boolean value tells if *geo_loc* is the same as *whois_reg_co*. To facilitate the check, ensuring that both country names were written in the same format was necessary. This was done by translating country names to International Organization for Standardization (ISO) 3166 alpha-2 country codes, effectively changing from full country names to a two-letter country code [ISO3166]. A Python library called *pycountry* [The22] was used to ensure the alpha-2 format of country names before they were compared. The use of the library is exemplified in Source Code 5.4.

Source code 5.4 Code to generate *bool_co* by using the *pycountry* library

```
#CONVERT COUNTRY NAMES TO ISO-3166-1 alpha_2 values:
def co_name_to_alpha(co_name):
    try:
        alpha_name = pycountry.countries.get(name=co_name).alpha_2
    except Exception as e:
        alpha_name = co_name #set alpha name to input if fail
    return alpha_name

#COMPARE COUNTRY IN WHOIS REGISTRANT WITH GEO_LOC FROM IP:
for i in range(0,len(geo_loc)):
    #convert country names from geo_loc and whois:
    geo_loc_name = co_name_to_alpha(geo_loc[i])
    whois_co_name = co_name_to_alpha(whois_co[i])

    #compare alpha-2 values:
    if geo_loc_name.upper() != whois_co_name.upper():
        boolean_co.append(0) #add 0 if different
    else:
        boolean_co.append(1)
```

registrar

The WHOIS registrar attribute examines who is responsible for distributing the domain, i.e., the domain registrar. The attribute was extracted from the WHOIS string using regular expressions. As *ssl_issuer_co*, this feature is included to investigate potential correlations between registrars and malicious websites.

jarm

The domain object retrieved from VirusTotal contains the JARM hash of the server hosting the domain [VTdom]. JARM is an active TLS server fingerprinting tool as described in Chapter 2, assessing the TLS details of the server. The JARM hash was included as a feature in this experiment as it may allow identification of malicious servers by hashing the server’s TLS responses. Furthermore, in a fleet of malicious servers, malicious actors tend to have similar configurations of their servers [Alt17].

cipher

It is possible to extract more precise information about the negotiated ciphers from the JARM hash. In particular, the JARM hash is comprised of two parts. The first 30 characters comprise the cipher and TLS version chosen by the server. Studying the use of JARM has shown that servers with malicious intentions produce similar JARM hash values [Alt17]. Furthermore, if the first 30 characters are the same, the servers have very similar configurations [Alt17]. As the first part of the JARM hash will be the same for similarly configured servers, it is added as a standalone attribute in this experiment.

tld

TLD of the domain. This feature is preserved from the original dataset [Sin20].

label

The label attribute is preserved from the original dataset. The dataset relied on the Google Safe Browsing API to determine the website’s intentions, thereby labelling it good or bad [Sin20; Goo23]. Verification of the labelling was not performed in this thesis.

5.2.2 Preprocessing

To better facilitate the experiment, the dataset had to be pre-processed. This included several steps to make the dataset fit the ML models. Removing unwanted attributes and performing general cleaning of the dataset were necessary before the classification could be initiated. Also, as some information collected from VirusTotal was partially missing, the generation of new information as well as the removal of duplicates, was therefore necessary.

Originally, the used dataset was divided into two sets; one training dataset and one test dataset. To obtain as many malicious websites as possible, we decided to merge the two datasets before the query process to VirusTotal was initiated. The final dataset was then split into a training and test set before the ML algorithms performed

classification. The split of test and train sets was done using the implemented method *train_test_split()* from sklearn [PVG+11]. Finding a suitable split involved familiarising with leading practices through related work and blogs, such as “Machine Learning Mastery” [Bro20], and practical testing. In the end, the author set the test size to 0.2. This reflects the original split of the dataset, in which 22% (8062) of the malicious websites were in the test dataset. Furthermore, this is used as a common split between training and test sets in supervised learning [Bro20].

The code which queried information from VirusTotal handled exceptions by adding “N/A”-values to the information fields. Handling the exceptions further was left to the preprocessing step. After extracting the information which would be used as features, a check was made to remove any sample which was missing both *whois_creation_time*, *whois_expires_time*, *ssl_start_time*, and *ssl_end_time*. The experiment would later fill in missing values, and the author decided to remove the samples that were missing too much information before this step.

After removing samples, the next step in the preprocessing phase was to fill the “N/A”-fields. This had to be done as not all of the ML algorithms handle “N/A”-values by default. To ensure a suitable dataset size, a decision not to remove all domains with any “N/A”-value was made. Instead, the fields were to be filled based on k-Nearest Neighbours. This was done with the *KNNImputer* implemented in Scikit-learn [Sci:Imp]. The method imputes the missing value of each sample based on the mean value of its five nearest neighbours in the dataset. The imputer deems two samples to be close if the features, which neither of the samples are missing, are close in value.

The time attributes extracted from both WHOIS and SSL certificate provide values which can be used as features in the experiment. However, extracting comparable dates directly from the retrieved information was not feasible as the WHOIS records and the SSL certificates differed in the time format they provided. To get an equal format that was comparable for all time values, they were converted to Unix timestamps [Wik23f]. Unix time counts the seconds elapsed since 01.01.1970 and thus provides an easy way to compare all time features. To convert the date format, the built-in Python module *datetime* was used.

However, having just the Unix timestamp does not provide any insightful information. To better discover correlations in the dataset, the time features were all discretised by clustering to have the same value if they were close in time. The clustering was done with the *KBinsDiscretizer* method implemented in Scikit-learn [Sci:KB]. The method creates a user-defined number of bins and groups the input into the given bins. For this experiment, the number of bins was set to reflect the number of days in the interval for the given time feature. E.g. for *whois_creation_time*, 13

473 bins were set as it was 13 473 days between the earliest and latest timestamp. This is an example of discretising, as described in Chapter 2. Having clustered the time features, they were all given new values. The new value did not represent time but which bin the sample was in, i.e., which day the sample’s feature belonged to.

For categorical features such as *ssl_alg* and *registrar*, an encoding procedure had to take place before the ML algorithms could perform the classification. This step was required as the algorithms, by default, do not handle categorical string values. To encode the categorical values, the method *OrdinalEncoder* from Scikit-learn was used [Sci:enc]. The transformer takes categorical features as input and converts the values to ordinal integers. The ML algorithms can then handle the features as integer values.

The dataset was, as previously stated, imbalanced. This posed challenges to the supervised ML algorithms, which develop a bias towards the dominant class during training [HRA+22]. Therefore, balancing the data before training the models was vital to achieving good test results. Different resampling strategies were described in Chapter 2. This thesis used both random undersampling and random oversampling as opposing strategies, with their results presented in the next chapter. The undersampling and oversampling methods were implemented using the imbalanced-learn library in Python [LNA17].

Finally, the dataset was cleared of any duplicates. This was vital to ensure that no domain could feature in both the training and test sets and thus cause data leakage, i.e., the model is trained on a sample on which it is also tested. As the queries sent to VirusTotal stripped the full URL and only used the domain name, several duplicates were present in the dataset. The pandas library [pdtea20] was used to remove unwanted duplicates through the implemented method *remove_duplicates()*. After duplicate removal and the other preprocessing steps, the dataset consisted of 10 485 maliciously labelled samples and 342 800 benignly labelled samples. A visualisation of the percentage of each class in the dataset is provided in Figure 5.2. It is worth noting that the limitations of 20 000 daily queries to VirusTotal made it impossible to query all samples in the original dataset. However, the academic API key they provided enabled a much larger dataset than it would have with a public key. The code was designed first to query all maliciously labelled samples to ensure both classes are represented in the final dataset used in the classification.

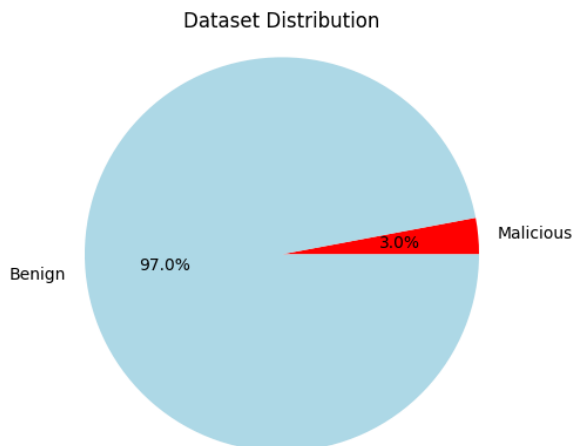


Figure 5.2: A pie chart visualising the imbalanced dataset used in the experiment

5.3 Machine Learning Implementation

This section presents the specifications of the implemented ML algorithms presented in Chapter 4, Section 4.3.1. In particular, the section will describe the hyperparameter tuning for the algorithms which require tuning, before an acknowledgement to relevant Python libraries concludes this chapter.

The Scikit-learn implemented method *RandomizedSearchCV* [Sci:RSCV] was used to tune the hyperparameters of the Random Forest and MLP classifiers. The method works by randomly selecting parameters from a user-defined list and combining them to find the optimal parameter settings. The method will run as many combinations as the user sets. In this experiment, a randomised search with 50 iterations and 3-fold cross-validation was performed to find optimal parameter settings. The randomised search approach differs from the widely used grid search method, which, unlike the randomised search, will try every permutation of the parameters. Although the grid search is widely used, Bergstra and Bengio [BB12] showed that the randomised search is more efficient than both grid search and manual search to finetune hyperparameters.

By default, the randomised search will evaluate the classifiers based on the accuracy score. However, as discussed in Section 2.3, it is important to select appropriate evaluation metrics to obtain good results when assessing ML performance [HS15]. As the dataset is imbalanced, the accuracy evaluation metric will favour the majority class. It may thus be high as long as enough correct predictions are

made for this class [BG16]. However, this experiment wants to detect the minority class of maliciously labelled websites. Therefore, the evaluation metric for the randomised search was changed to the recall score of the malicious class to improve the performance for the minority class [BG16]. The final hyperparameters for the Random Forest and MLP classifier implementations are presented in Table 5.5 and Table 5.6, respectively.

The MLP classifier is, as stated in Chapter 4, highly sensitive to feature scaling [Sci:MLP]. The model performance can thus be enhanced if the features in both the training and test sets are scaled before the classifier is put into action. This was done with the method *StandardScaler*, which was implemented through Scikit-learn [Sci:MLP]. The method standardises the input attributes to have a mean value of 0 and a variance of 1.

Random Forest

The Random Forest classifier was implemented with the hyperparameters shown in Table 5.5. *n_estimators* tells how many trees are included in the forest. The parameter *min_samples_split* gives the minimum number of samples in a node before the node is split. Likewise, the *min_samples_leaf* states the minimum number of samples required in a leaf node. If a split is to happen, the number of samples in the leaf node must be satisfied for both the left and right nodes after the split. The *max_features* parameter sets the number of features to consider when finding the best split. *max_depth* sets the depth of the trees in the forest. Finally, the *bootstrap* parameter denotes whether the whole dataset is used for building the trees or if bootstrap samples are used [Sci:RFC]. Note that the different resampling strategies yielded different optimal parameter selections.

Table 5.5: Hyperparameter values for implementation of the Random Forest classifier with random undersampling (RUS) and random oversampling (ROS)

Parameter name	Parameter value (RUS)	Parameter value (ROS)
<i>n_estimators</i>	837	450
<i>min_samples_split</i>	5	2
<i>min_samples_leaf</i>	2	2
<i>max_features</i>	sqrt	sqrt
<i>max_depth</i>	20	30
<i>bootstrap</i>	True	False

Multi-Layer Perceptron

Finetuning the hyperparameters for the MLP classifier revealed the optimal hyperparameters for both resampling strategies as shown in Table 5.6. The *max_iter* parameter determines the maximum number of iterations. *hidden_layer_size* determines the number of neurons in the hidden layers. As described in Chapter 2, this is one of the main challenges to determine using this classifier. The *activation* parameter states what function will be used to activate the hidden layer. Logistic represents the logistic sigmoid function. The *solver* is used for weight optimisations and will continue to iterate until it converges or until the *max_iter* is reached. The *learning_rate* parameter is also related to the solver, stating the learning rate schedule for weight updates. This parameter was left at the default value “constant” [Sci:MLPC].

Table 5.6: Hyperparameter values for implementation of the Multi-Layer Perceptron classifier with random undersampling (RUS) and random oversampling (ROS)

Parameter name	Parameter value (RUS)	Parameter value (ROS)
max_iter	500	1500
hidden_layer_sizes	50, 100, 50	50, 100, 50
activation	logistic	tanh
solver	adam	adam
learning_rate	constant	constant

Python Libraries

A short summary and acknowledgement of the significant Python libraries that were used in this thesis will be given here.

pandas

pandas is a Python library facilitating data processing. It was used in this experiment to read the dataset as csv format to a dataframe object. The dataframe object was then used to generate features and used throughout the preprocessing steps, as explained in subsection 5.2.1 and 5.2.2, respectively. The pandas software is available here: [pdtea20], and the project is further explained in “Data Structures for Statistical Computing in Python” [McK10].

Scikit-learn

Scikit-learn is an ML library for Python. The library features implemented ML models able to perform classification and regression [PVG+11]. The ML classifiers in this thesis were all implemented using the Scikit-learn library.

imbalanced-learn

The dataset used in this thesis was imbalanced. To improve the learning process of the supervised ML algorithms, the training data was resampled. The resampling methods were implemented using the Python library `imbalanced-learn` [LNA17].

matplotlib

Visualisation of the results was made possible by the `Matplotlib` library [Hun07].

Chapter 6

Results and Discussion

Following the methodology and experiment descriptions in the previous chapters, this chapter will present the experiment results and discuss what was found. The results are presented in two separate sections, assessing the ML performance and the feature importance, respectively. The results and discussion are followed by a discussion of the theoretical implications of the work in section 6.3, before a section assesses actionable suggestions based on the findings in Section 6.4. Section 6.5 will cover a general discussion of the work, before the conclusions section summarises the thesis. Finally, the proposed future work will conclude this chapter.

6.1 Machine Learning Performance

The ML performance will, as described in Chapter 4, be assessed based on common ML evaluation metrics. These are presented in plots to visually clarify the differences between the different classifiers and as tables to showcase the exact numerical values. The most important metric plots for answering the stated research questions are included in this chapter, while all metric plots are included in Appendix C. In particular, for random undersampling and random oversampling, respectively, *precision* score plots are available in Figure C.1 and Figure C.2, *recall* scores in Figure C.3 and Figure C.4, *F1-score* in Figure C.5 and Figure C.6, and *accuracy* scores in Figure C.7 and Figure C.8.

The first part of the experiment was performed with random undersampling to create a balanced training dataset. Table 6.1 presents the numerical values of the evaluation metrics using this resampling method. The table is accompanied by the normalised confusion matrices in Figure 6.1, visualising the performance of the classifiers. The results from the experiment with random oversampling are presented in Table 6.2, and the coherent confusion matrices are presented in Figure 6.2.

Table 6.1: Classification performance metrics with random undersampling

Classifier	Precision	Recall	F1-score	Accuracy
Random Forest	0.09	0.78	0.16	0.76
AdaBoost	0.08	0.78	0.14	0.72
Naive Bayes	0.05	0.68	0.10	0.64
QDA	0.05	0.71	0.10	0.64
MLPC	0.06	0.79	0.11	0.62

Table 6.2: Classification performance metrics with random oversampling

Classifier	Precision	Recall	F1-score	Accuracy
Random Forest	0.36	0.34	0.35	0.96
AdaBoost	0.08	0.77	0.14	0.73
Naive Bayes	0.05	0.68	0.10	0.64
QDA	0.05	0.71	0.10	0.63
MLPC	0.12	0.38	0.18	0.90

Note that the confusion matrices are normalised to better visualise the classifiers' relative performance on the two classes in the imbalanced dataset. The reader is encouraged to compare the matrices with the standard confusion matrices included in Appendix A and Appendix B for random undersampling and random oversampling, respectively. The matrices in the appendix highlight only the majority class and are thus difficult to use to assess the performance of the minority class classification. Therefore, the matrices are excluded from this chapter.

The confusion matrices in Figure 6.1 visualise the classification performance using random undersampling as the resampling technique to create a balanced training dataset. The matrices show that the TP and TN scores are higher than the FP and FN scores for all classifiers. While this is the bare minimum to expect from a classification, it does imply that the classifiers, in most cases, are able to correctly label the samples. Note also that these results are normalised, so "most cases" is relative to the total number of samples for the given class. Although the MLP classifier was fine-tuned to increase the detection of malicious samples, this seems to have caused a significant number of FP. This is reflected by the FP score of 0.39 in Figure 6.1e. On the other hand, the Random Forest classifier does not show signs of an increased number of FP after the fine-tuning. Contrary, it shows the lowest FP score of all classifiers for random undersampling, reflected by the FP score of 0.24 in Figure 6.1a.

Furthermore, the Random Forest classifier achieved a TP score of 0.78, which is joint second-best with the AdaBoost classifier, with only the MLP classifier achieving a better score of 0.79. This may, however, be partly caused by the classifier tending to label more samples as malicious, which is also reflected in the number of FPs. The AdaBoost classifier did achieve the same TP score as the Random Forest classifier. However, AdaBoost did have a higher number of FPs, showing the same tendencies of labelling more samples as malicious like the MLP classifier.

Figure 6.2 shows the confusion matrices using random oversampling as the resampling strategy to create a balanced training dataset. It is interesting to see that Naive Bayes and QDA achieved the exact same results for both resampling strategies as shown in Figure 6.1c and Figure 6.2c for Naive Bayes and Figure 6.1d and Figure 6.2d for QDA. This indicates that these classifiers perform equally well with both resampling strategies. The AdaBoost classifier performs slightly better for the benign class with oversampling as the resampling strategy as shown in Figure 6.1b and Figure 6.2b for undersampling and oversampling, respectively. The better performance is reflected by the TN score with random oversampling, which is 0.73, while the score is 0.72 with random undersampling. However, AdaBoost performs worse for oversampling when considering the malicious class, decreasing 0.01 from 0.78 to 0.77 TP score.

Evaluating the performance of the Random Forest and the MLP classifier with the random oversampling strategy indicates that the hyperparameter tuning has not worked. This is reflected by the low TP score of 0.34 for Random Forest (Figure 6.2a) and 0.38 for MLP (Figure 6.2e). The models do, however, have a high TN score, correctly labelling most of the benign samples. This is reflected by the TN score of 0.98 and 0.91 for Random Forest and MLP classifiers, respectively. Assessing the scores individually provides insight into the classification performance for each class. However, combining them allows a closer examination of the overall classification performance. This is easily done with the numerical values presented in the tables above and the evaluation metrics as described in Chapter 4.

The following subsections will assess the results in the tables in more detail. Each subsection assesses a single evaluation metric and will provide a table repeating only the discussed evaluation metric score. Note that these tables are repeated values from Table 6.1 and Table 6.2. First, *precision* and *recall* are discussed together in Section 6.1.1. These metrics are discussed in the same section because of their interdependency. A subsection discussing the *F1-score* follows in Section 6.1.2, before Section 6.1.3 discusses the *accuracy* score. The discussion will also include a comparison of the results from random undersampling and random oversampling. To support the comparison, relevant metric plots will be provided, as described in the introduction of this chapter.

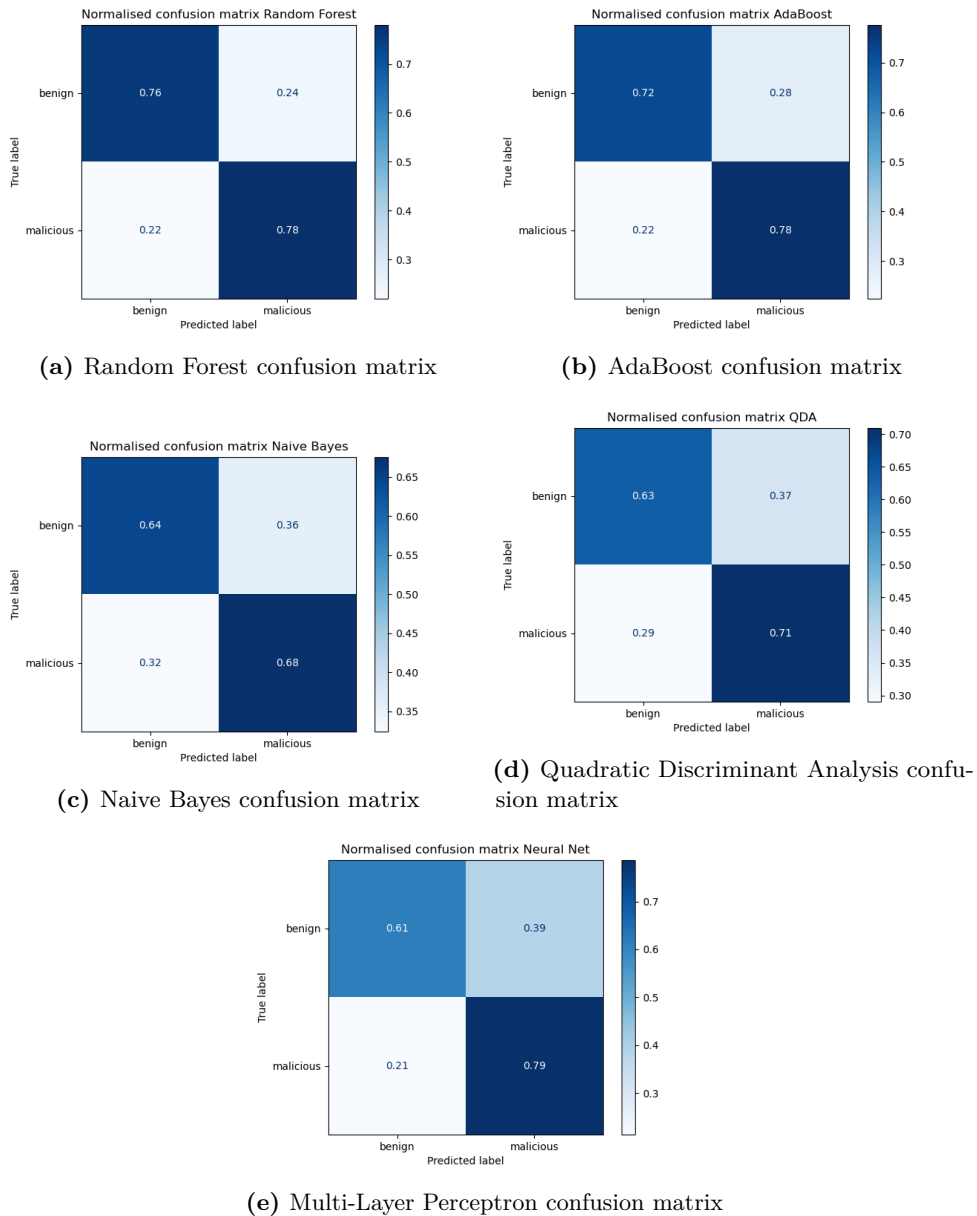
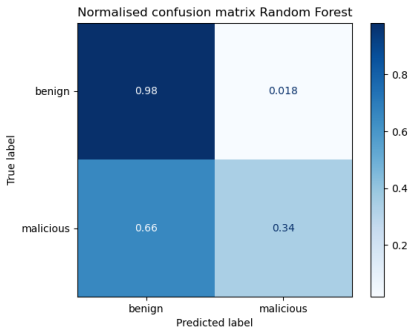
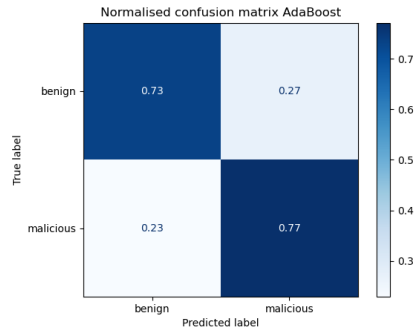


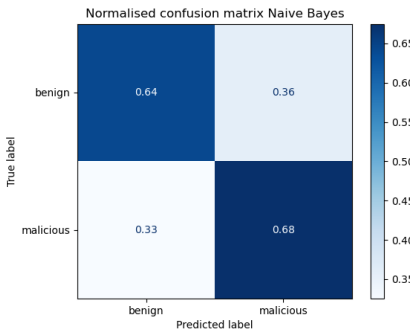
Figure 6.1: Normalised confusion matrices with random undersampling as the resampling technique for training data



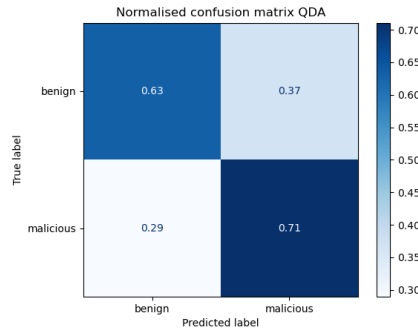
(a) Random Forest confusion matrix



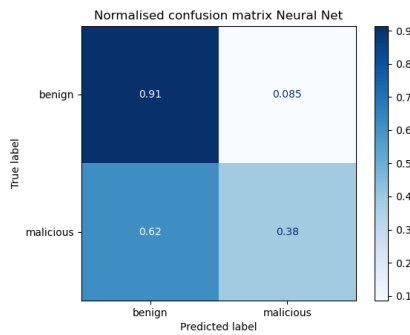
(b) AdaBoost confusion matrix



(c) Naive Bayes confusion matrix



(d) Quadratic Discriminant Analysis confusion matrix



(e) Multi-Layer Perceptron confusion matrix

Figure 6.2: Normalised confusion matrices with random oversampling as the resampling technique for training data

6.1.1 Precision and Recall

As presented in Table 6.3, the precision score is low for all classifiers using both random undersampling and random oversampling to resample the training data. However, as presented in Equation 4.4, the precision score is computed using the number of FPs in the denominator. As the ML classifiers in this thesis wanted to discover the minority malicious class, it is inevitable that FP samples occur [BG16]. This is explained by the high difference in number of samples in each class. Trying to label the malicious minority class, will most likely incorrectly label some samples from the majority benign class as malicious, causing an increased FP score [BG16]. The imbalanced dataset will significantly impact the precision score, as more FPs will be classified than TPs, resulting in a low precision score. This is directly verifiable in Equation 4.4, where $FP \gg TP$ will trivially yield a low precision score.

Table 6.3: Classification precision and recall scores with random undersampling and random oversampling

Classifier	Random Undersampling		Random Oversampling	
	Precision	Recall	Precision	Recall
Random Forest	0.09	0.78	0.36	0.34
AdaBoost	0.08	0.78	0.08	0.77
Naive Bayes	0.05	0.68	0.05	0.68
QDA	0.05	0.71	0.05	0.71
MLPC	0.06	0.79	0.12	0.38

As shown in Table 6.3 and Figure 6.3, the recall scores for random undersampling are relatively high for all classifiers. However, it is interesting to find Random Forest and the MLP classifiers to have the highest recall score after fine-tuning their hyperparameters. This manifests the importance of hyperparameter selection and shows that when done correctly, it can enhance the performance of the ML classifiers. Figure 6.3 further shows that Naive Bayes and QDA have the lowest recall score using random undersampling, with the QDA classifier performing slightly better of the two.

Assessing the recall scores with the random oversampling strategy in Table 6.3 and coherent plots in Figure C.4, clearly shows that Random Forest and MLP classifiers have the lowest score. As with random undersampling, the hyperparameters of these models were fine-tuned to increase the recall score. However, with the significantly lower recall scores compared to the other classifiers, it is clear that the hyperparameters have not successfully been tuned to increase the recall score with random oversampling. Although the author supplied a list with great variance among the different parameters to use in the randomised search, it is possible that

neither of the combinations tried in the 50 iterations provided a better recall score. Furthermore, the supplied parameters could also be out of the range for the optimal parameter selection for oversampling, even with the significant variation provided.

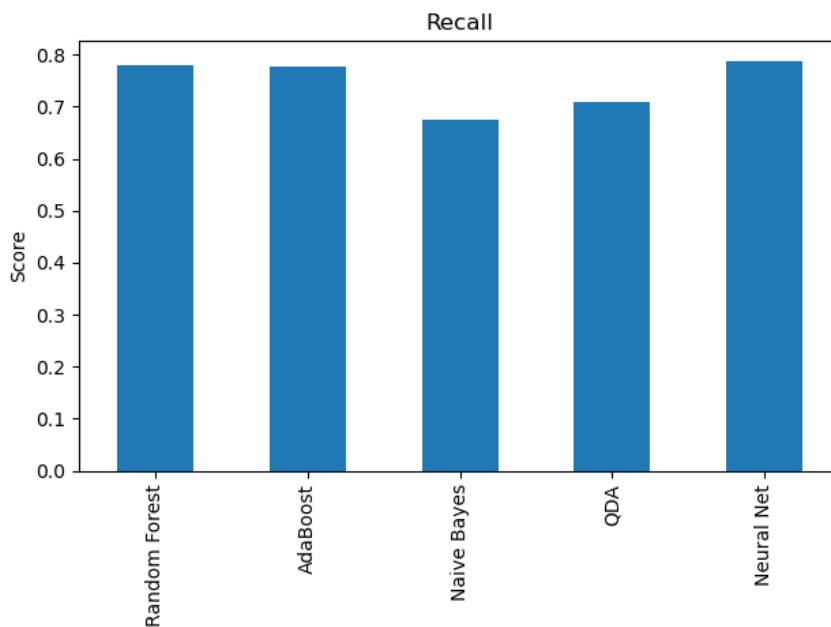


Figure 6.3: Recall scores using random undersampling

There is a trade-off between the precision score and the recall score [BG16]. The Random Forest and MLP classifiers in this experiment were optimised for a high recall score of the malicious class. However, tuning a classifier to have a higher recall score will inevitably result in lower precision. This can be described by the following example; classifying a class A with a high recall score means that the classifier correctly predicts most of class A. The classifier will likely also predict other classes as class A, resulting in a lower precision score [BG16]. As the used dataset is imbalanced, with only 3% labelled malicious, it is clear that the models optimised for a high recall score will classify many benign samples as malicious, effectively reducing the precision score. Classifying benign samples as malicious in this experiment is reflected by the FP score.

The high number of FPs is a concern when assessing the ML classifiers. While it is not optimal to have many FPs, it is arguably better to have more FPs and detect the malicious samples than to reduce the number and miss out on the malicious samples. I.e., it is better to detect the malicious sites and potentially flag several benign sites

in the same process, than not to detect the malicious sites at all. Assessing the low precision score, combined with the recall score and the normalised confusion matrices, clearly shows that while precision is low, the classifiers can successfully label the malicious classes.

The precision score for Random Forest for the oversampling method is substantially higher than the other classifiers, as visually presented in Figure C.2. However, assessing its recall score, it is evident that the model has not been successfully tuned to detect the malicious class and increase the recall score. On the contrary, the model has increased detection of the benign majority class, reflected by the TN score of 0.98 and FN score of 0.66, as shown in Figure 6.2a. The high precision is therefore neglected, and the model is not considered to have a better overall performance than the other classifiers. Similarly, the MLP classifier with random oversampling has a higher precision score than the other classifiers, albeit not as substantially as the Random Forest classifier. Like the low recall score, the substantial increase in TN and FN indicates that the randomised search conducted to determine the hyperparameters failed for random oversampling.

6.1.2 F1-score

Assessing the precision score and recall score individually, facilitates assessment of the classification performance for a particular class, such as the malicious class in this experiment assessed through the recall score. Combining the two scores according to Equation 4.6, yields the F1-score of the classifiers. While the F1-score for random oversampling clearly favours the Random Forest as of its high precision score addressed above, the Neural Network model MLP classifier scores second best. However, as shown in the confusion matrix for the two classifiers (Figure 6.2a and Figure 6.2e), these models have a low number of TPs for the malicious class. This is also reflected by the low recall score presented in Table 6.2. Ignoring these two models, AdaBoost scores the best F1-score using random oversampling as the resampling strategy. A plot of the F1-scores for random oversampling is included in Appendix C, Figure C.6.

Table 6.4: Classification F1-scores with random undersampling (RUS) and random oversampling (ROS)

Classifier	F1-score (RUS)	F1-score (ROS)
Random Forest	0.16	0.35
AdaBoost	0.14	0.14
Naive Bayes	0.10	0.10
QDA	0.10	0.10
MLPC	0.11	0.18

The F1-scores using random undersampling present interesting results, and are visualised in Figure 6.4 for a comparison of the different classifiers. The exact numerical values are provided in Table 6.4. The Random Forest model has the best F1-score with 0.16, followed by AdaBoost with 0.14. Interestingly, while the Random Forest model with random undersampling produces the best F1-score (0.16), the other fine-tuned model, the Neural Net MLP classifier, only achieved an F1-score of 0.11. That is only marginally better than the two worst classifiers Naive Bayes and QDA. Investigating the numbers in greater detail revealed that the MLP classifier achieved a marginally higher recall score than the Random Forest, but achieved a precision score which was 0.03 lower. This explains the significant difference in the F1-score between the two classifiers.

Combined with the high recall score, as shown in Figure 6.3, it is clear to see that hyperparameter tuning has worked significantly better for the undersampling strategy. Assessing the F1-score combined with the confusion matrices in Figure 6.2 it is evident that the ensemble classifiers Random Forest and AdaBoost have outperformed the others.

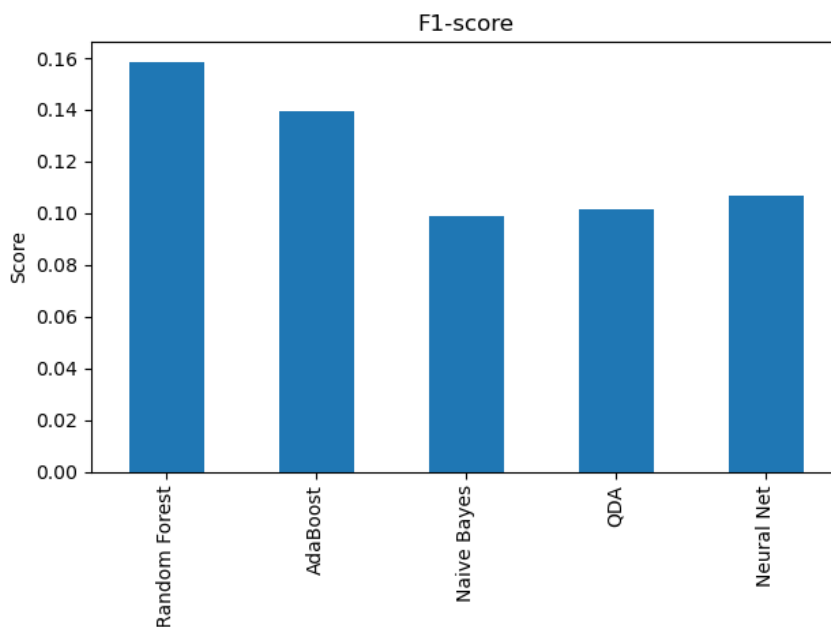


Figure 6.4: F1-scores using random undersampling

6.1.3 Accuracy

As described in Chapter 4, evaluating the ML classifiers with relevant evaluation metrics is important to get a correct impression of the results. E.g., using only the accuracy score can give deceiving results when aiming at classifying a minority class. Therefore, assessing the accuracy score with such an imbalanced dataset as this experiment has used is not a preferred strategy. The score is included here to help differentiate the classifiers when compared to each other but is not used as a metric to assess the performance of the classifiers individually.

A significant argument for scrapping the accuracy score as an evaluation metric is its dependency on the number of correctly classified samples, i.e., TP and TN, as shown in Equation 4.3. If the classifier correctly labels most of the majority class, it will achieve a high accuracy score. This is exemplified by the Random Forest classifier with random oversampling achieving an accuracy score of 0.96 as shown in Table 6.5. Assessing its performance compared to the other implemented classifiers in Figure 6.5, it is clear that the Random Forest model stands out as the model with the evidently best accuracy score. However, as shown in the confusion matrix for Random Forest in Figure A.1a, the model hardly labels any sample malicious. This is also reflected in the low recall score addressed in Section 6.1.1. The example shows that a classifier completely failing to label any sample as malicious (minority class) will achieve a high accuracy score if it correctly labels the majority class (benign).

Table 6.5: Classification accuracy scores with random undersampling (RUS) and random oversampling (ROS)

Classifier	Accuracy (RUS)	Accuracy (ROS)
Random Forest	0.76	0.96
AdaBoost	0.72	0.73
Naive Bayes	0.64	0.64
QDA	0.64	0.63
MLPC	0.62	0.90

Assessing the accuracy scores when the random undersampling method is used to resample the training data, shows again that Random Forest and AdaBoost stand out as the most promising classifiers. The accuracy scores of the different classifiers are visualised in Figure C.7. It is also interesting to see that the Neural Net model MLP classifier achieved the lowest accuracy score using random undersampling. This could be explained by the low precision score indicating a high number of FPs compared to the other classifiers. As seen in Equation 4.3, the number of TNs is used in the numerator of the equation. As the MLP classifier has a higher number of FP, it will also have a lower number of TN, resulting in a lower accuracy score. The lower TN

and higher FP labellings of the MLP classifier are visually presented in Figure 6.1e, where the classifier achieved 0.39 FP and only 0.61 TN. That is also the lowest TN score of all classifiers.

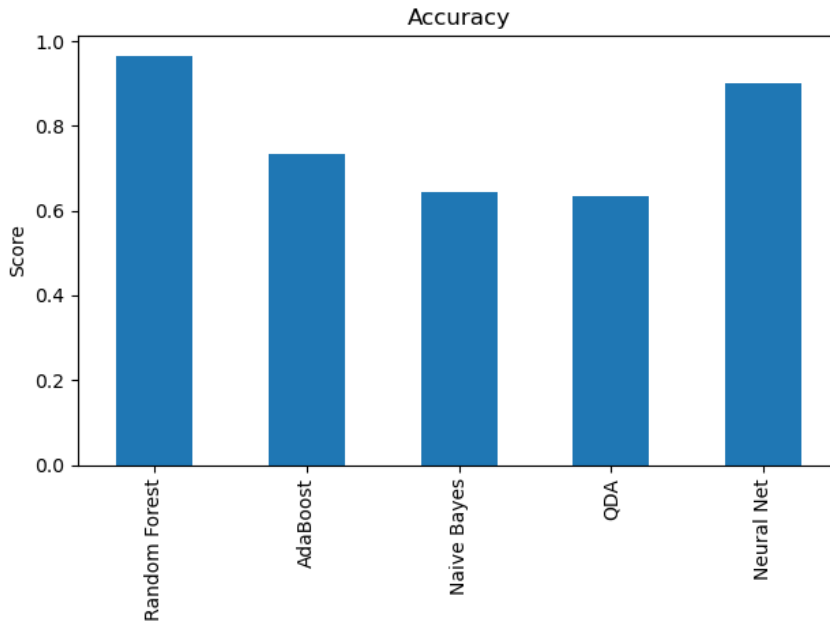


Figure 6.5: Accuracy scores using random oversampling

6.2 Feature Importance

The feature importances are extracted from the Random Forest and AdaBoost classifiers. The feature importances from each classifier are presented in a bar chart to ease the comparison of feature importance for the different classifiers. Figure 6.6 and Figure D.2 present the feature importances using random undersampling and random oversampling, respectively.

The feature importances are measured in MDI as described in Chapter 4. Interestingly, the classifiers appear to evaluate the features as equally important, independent of the resampling method. This is exemplified by the *tld* feature being the most important feature for AdaBoost using both random undersampling and random oversampling. It is also worth noting the importance of the *cipher* feature as the third most important feature in AdaBoost, which can indicate some particular server settings are used for TLS setup with malicious servers. Both AdaBoost and Random Forest value the *registrar* to have some importance. However, the most

significant features of the Random Forest classifier are related to the WHOIS times, i.e., *whois_creation_time* and *whois_expires_time*.

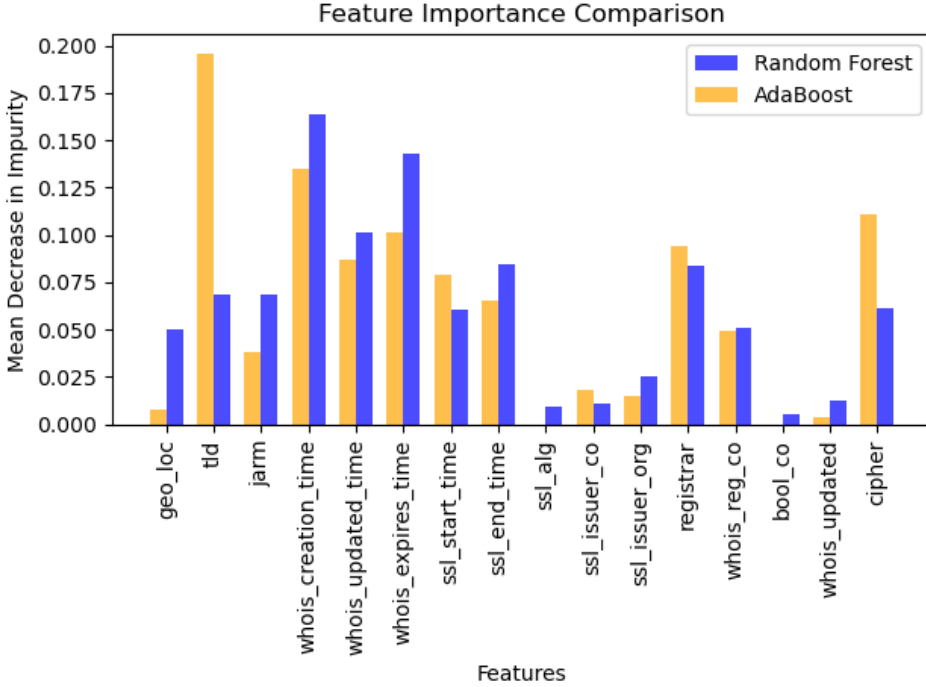


Figure 6.6: Feature importances using random undersampling as resampling method for training data

The MDI measure for feature importances is, however, not flawless. The model is based on statistics derived from the training data and may thus not represent feature importances for the unseen test data. Furthermore, the model favour features with high cardinality, i.e., features with many unique values [Sci:ens]. E.g., with more than 13 000 unique values for the *whois_creation_time* feature, this feature may be assigned more importance than the *ssl_alg* feature taking only three unique values. Also, this may lead neglecton of the two boolean features, *bool_co* and *whois_updated*.

6.3 Theoretical Implications

This thesis wanted to investigate the possibility of using WHOIS information in a world restricted by GDPR, combined with SSL certificates, to identify malicious

websites. The thesis has identified the following theoretical implications of the work that can contribute to the field of cybersecurity and malicious website classification:

Validation of ML effectiveness: This thesis has implemented five different ML classifiers to mitigate the known ML challenges related to the NFL theorem. All classifiers have successfully classified the maliciously labelled domains, albeit with different performances, as assessed in the previous subsections. The successful implementation strengthens the theoretical foundation for using ML classifiers for malicious website detection.

Feature importance: Assessing the feature importances in the classification process of the two ML models Random Forest and AdaBoost, has provided insight into what attributes are important for distinguishing malicious websites from benign ones. The thesis has assessed strictly non-personal information from WHOIS records, which aids future feature selection from WHOIS records.

Methodological implications: To facilitate the proposed methodology used in this thesis, we started by performing a literature review. It was conducted to discover leading practices in malicious website detection, with a particular interest in the used ML models and features. Assessing commonly used ML models revealed five promising candidates which were implemented in this thesis. Also, relevant ML problems such as the “No Free Lunch”-theorem (Theorem 2.1), discussed in Chapter 2, was important to bear in mind when proposing the methodology. Initial work also revealed that the performance of ML models could be significantly affected by the data used for training [HRA+22]. Thus, the need to resample training data was included in the methodology. The preprocessing steps described in this thesis have provided insights into challenges for feature engineering.

Initial preparations of the methodology also revealed common evaluation metrics used to assess the performance of the classifiers. The performance of the ML classifiers was evaluated accordingly. Discussing the different evaluation metrics in light of the results obtained in the thesis experiment, has contributed to assisting future work evaluating malicious website classification with ML classifiers more accurately.

Leading practices: The initial literature study was also used to identify current trends in malicious website classification. Currently, as much as 69% of security applications based on WHOIS records use information that is now redacted following the implementation of GDPR [LLZ+21]. In relation to relevant work assessed in Chapter 3, this thesis’ proposed method is distinctive by using solely available information. This can assist future work in developing innovative solutions to malicious website classification.

Limitations: A notable limitation in this work is the lack of distinction between

illegal and *malicious* websites, as described in the introduction. However, in the real world, such a distinction is necessary. The distinction is necessary to separate illegal websites from malicious ones, such that LEAs can prioritise certain websites for detection and takedown.

6.4 Practical Recommendations

This thesis has found the following practical recommendations for policy-makers, industry professionals, or other stakeholders in the field of cybersecurity and malicious website classification:

Implement ensemble learners: As shown in the previous subsections, both the Random Forest and AdaBoost classifiers achieved good performance for the classification using random undersampling. Furthermore, AdaBoost also achieved a high performance using random oversampling as the resampling method. Although the NFL theorem must be considered for future implementations, these models present promising results in this thesis and should thus be tested.

Continuously update and adapt: This thesis used a dataset assembled three years prior to the study. Therefore, there might be discontinued websites in the dataset. Also, there might be samples that no longer justify their associated label. This thesis assumed that the labelling was correct and did not investigate any discontinued websites in detail. Thereby, there may be irregularities in the two distinct classes. It is recommended that new samples are included in the training data to ensure the ability to identify emerging malicious website patterns with the ML classifiers. Also, resampling methods were used in this thesis with mixed success. Resampling the training data can also improve the learning of the classifiers.

Evaluate and mitigate ethical concerns: This thesis has shown that classification without personal data from WHOIS is possible. To ensure user privacy, it is necessary to comply with relevant protective legislations, such as GDPR. It is recommended that any relevant related work updates the feature selection from WHOIS records to only use non-personal information.

Reproducibility and transparency: Collective collaboration in the research environment is encouraged as it can provide a more comprehensive and more accurate classification of malicious websites. This includes information sharing, such as collective access to a continuously updated labelled dataset for training, feature engineering techniques, and providing insight into how research is conducted. This is further described in relation to good research ethics in Chapter 7. However, this thesis uses a dataset containing personal information that will not be published or shared after the thesis closure. A more detailed justification for this choice is provided

in Chapter 7. Given the detailed methodology explained in Chapter 4, the specific implementation details in Chapter 5, and the publicly available dataset used, this thesis work is transparently conducted. Furthermore, with the specified information collection from the VirusTotal APIs, the work should be easy to reproduce.

6.5 General Discussion

ML presents great opportunities to detect malicious websites by being able to process large amounts of data in a reasonable amount of time. Most ML algorithms implemented in this experiment successfully classified most of the malicious samples. However, a FP score of 26% as the best result using random undersampling and the Random Forest classifier, still results in 16 488 falsely labelled benign websites as shown in Figure A.1a. Although this experiment fine-tuned hyperparameters to detect the malicious (minority) class, the high amount of falsely labelled benign samples cannot be ignored. A maliciously labelled sample would have to be manually checked before determining the final label. Thus, manually checking 26% of all benign websites is a substantial amount of work, making the proposed solution not yet ready for large-scale implementation.

One of the most important features in the classification was the time-based features derived from the WHOIS records. As described in Chapter 5, time-based features were grouped to represent the same day. Memory constraints limited the number of possible clusters for time-based features. Fine-graining the time clusters, e.g., to every hour, could be beneficial to reduce the number of FPs as the models now learn that one particular day of domain registration is related to malicious websites.

Finding the cipher specifications of the server as one of the most important features, while the SSL algorithm is not important at all, is also interesting. This discovery indicates that more complex features, such as the full server cipher information, provide insight into servers with malicious intentions. In contrast, simple settings such as only the encryption algorithm have no significance. This may be related to the fact that only three different algorithms are possible values for the *ssl_alg* feature, but still, these represent the most known encryption algorithms. This discovery must also be seen in relation to the limitation of MDI and low cardinality features, as discussed in Section 6.2.

6.6 Conclusions

Illegal websites and websites hosting illegal content facilitate a globalised criminal network, cooperating in an increasingly closer manner. Through the CaaS business model, criminals buy and sell services on an international stage, requiring an increased

collaboration between LEAs. This thesis set out to investigate the identification of malicious websites based on publicly available WHOIS information after the implementation of the GDPR legislation, in combination with SSL information.

The research used a publicly available dataset from Singh [Sin20] and features obtained through VirusTotal’s APIs. The classification was done with five different ML algorithms, some of which were previously used for phishing website detection. To our knowledge, we are the first to perform classification with restricted WHOIS records, in combination with SSL certificate information, on this dataset. The features were fed to different ML classifiers trained on a resampled training dataset to label the testing samples.

This thesis coined two research questions to help guide the work. Following the completion of the thesis, the following conclusions can be made:

Research question 1: *Which supervised machine learning algorithm performs best in malicious website classification?*

The five Machine Learning (ML) algorithms Random Forest, AdaBoost, Naive Bayes, Quadratic Discriminant Analysis (QDA), and Multi-Layer Perceptron (MLP) were implemented and evaluated in this thesis. The ensemble methods Random Forest and AdaBoost showed the best results, as discussed in Section 6.1. The performance of the classifiers was assessed using relevant evaluation metrics, and their performance was compared and discussed through the use of tables, bar charts and confusion matrices. Normalising the results in the confusion matrices helped better visualise the relative performance of the classifier on classification for the two classes. The performance was further assessed using both random undersampling and random oversampling to resample the training dataset. To determine the optimal hyperparameters, a randomised search was conducted for the Random Forest and Multi-Layer Perceptron (MLP) classifiers. This was done for both resampling strategies. The models were among the top-performing classifiers for random undersampling. However, these ML models achieved the lowest performance using the random oversampling strategy, indicating that the hyperparameter selection was wrong.

Because of the No Free Lunch (NFL) theorem, it is difficult to provide a definite conclusion to this research question. However, it is evident that the two models, Random Forest and AdaBoost, achieved the best performance considering both resampling strategies in this thesis. Therefore, they should be considered implemented for any future related work.

Research question 2: *Which infrastructure-based features can distinguish a malicious website from a benign one?*

This thesis performed a binary classification problem using ML classifiers. The classification was based on a number of features as defined in Chapter 5. Feature importances were assessed using built-in functions from the Random Forest and AdaBoost classifiers implemented in the study. The Mean Decrease in Impurity (MDI) was used to quantify the importances. Albeit discussed in relation to the limitations imposed by MDI, time-related features were some of the most prominent features in the classification. Also, the registrar from the WHOIS records was an important feature. Assessing the features derived from the SSL certificates did not find any particularly important feature. However, the cipher settings selected by the server in the TLS session setup was a significant feature. This was derived from the JARM hash created from the TLS setup with the server. For AdaBoost, the most prominent feature was the Top Level Domain (TLD) feature.

The thesis identified the theoretical implications of the performed study to provide a higher level understanding of the research questions, methodology and analysis. The findings highlighted the significance of the results of the study. In particular, the thesis provided validation of ML effectiveness for malicious website classification and insight into what attributes are important for distinguishing malicious websites from benign ones. Also, methodological implications included strategies to mitigate known ML challenges, such as the NFL theorem and resampling strategies of training data. The theoretical implications covered leading practices comparing our study to the leading trends in security applications based on WHOIS records, showing that while our research solely relies on public information, a majority of modern security applications based on WHOIS records, rely on personal information to perform the identification. Finally, the section stated a limitation in the lack of distinction between *malicious* and *illegal* websites in this thesis.

Following the theoretical implications, practical recommendations were also suggested based on the findings in this thesis. The section covered actionable suggestions for industry professionals and cybersecurity experts. Hereunder, practical guidance, such as implementing the top-performing ensemble learners used in this study, was given. Furthermore, it is important to address the evolving cybercrime environment and train the classifiers on continuously updated data to discover emerging patterns in malicious website WHOIS records and SSL certificates. Also, as this thesis has a chapter devoted to ethical considerations, it is important to assess the ethical aspects and privacy regulations when implementing cybersecurity solutions. Finally, the practical recommendations section covered reproducibility and transparency for conducting ethically sound research.

Our work shows that WHOIS records still provide information which can be used in malicious website detection. Although a recent survey showed that 69% of related work based on WHOIS records relied on now redacted information, the work can be

modified to comply with the GDPR legislation. Combined with available information from SSL certificates, infrastructure-based information still presents opportunities for early malicious website detection.

6.7 Future Work

This thesis has investigated malicious website detection using infrastructure-based features. We hope this research has inspired the reader and motivated further work. We here propose several suggested topics for future work in malicious website detection. Some of the topics are directly related to limitations that restricted the work in this thesis.

Multiclass classification: Originally, the pre-project leading to this thesis wanted to investigate whether the ML classifiers were able also to label different types of malicious websites. E.g., if they were able to separate phishing sites from illegal marketplaces. However, limitations in the available information from VirusTotal and the used dataset resulted in a binary classification only. Other sources of information were not investigated in this thesis. Suggested approaches for finding multiclass labels can be other antivirus companies or datasets already prepared with multiclass labels.

Hyperparameter fine-tuning: This thesis fine-tuned the hyperparameters of the Random Forest and MLP ML algorithms, using a randomised search optimised to detect the malicious minority class. Fine-tuning the algorithms' hyperparameters even more, could enhance the performance. E.g., a combination of evaluation metrics could be used to not only optimise for the malicious recall score but also improve precision, essentially lowering the number of FPs. Also, other fine-tuning methods, such as *grid search*, could be explored and practically tested and compared to the randomised search used in this thesis.

Random oversampling parameter tuning: Fine-tuning for the experiment using random oversampling yielded poor results compared to random undersampling. This was not discussed or investigated in depth. A closer assessment of why this was the case could be useful for other work wanting to use resampling methods to create a balanced training dataset. We propose investigating this with a broader parameter space, enabling more variety in the randomised search parameter selection. Also, increasing the number of iterations could improve the selection by testing several combinations. Finally, a *grid search* could be performed to test all possible combinations of the parameter space. Note, however, that this approach is time-consuming.

Reduce FPs: As discussed in Section 6.5, the number of FPs in this thesis

experiment is substantial. Reducing the number would significantly improve the performance of the ML classifiers and ease the manual work should they be used to flag malicious websites in a practical implementation. As suggested above, this could be solved by parameter tuning.

Balanced test data: This thesis used an imbalanced dataset for testing as it reflected the original dataset, as well as the actual number of malicious websites online. It could be interesting for future work to assess the differences in classification using a balanced and an imbalanced test set. This could easily be performed by resampling the test set similarly to the training set resampling as described in this thesis.

Fine-graining time features: The time features used in this thesis are clustered into groups reflecting one particular day in the time interval. It would be interesting to see whether time-based features could serve more use when they are fine-grained to, e.g., every hour. Potentially, this could help reduce the number of false positives by teaching the classifiers that an hour is associated with malicious websites rather than a full day. Reducing the total interval in the dataset is suggested to mitigate memory overflow, should this future work be investigated.

Cluster classification: As proposed in the pre-project conducted in the fall of 2022, the original methodology was to use clustering algorithms to identify malicious websites [Bak22]. Although this thesis chose to use supervised ML algorithms with the labelled dataset, a suggestion for future work is to assess the performance of both supervised learning and clustering, an unsupervised ML method. This could easily be implemented using the Python libraries referred to in this thesis.

Redirecting domains: As described in Chapter 5, redirection information, i.e., redirection chain and final URL is available through the URL API of VirusTotal. This thesis did not investigate the redirection behaviour of the domains, but as previously described, this is linked to maliciousness. Therefore, an interesting idea for future work is to investigate and classify the domains both in the redirection chain and the final URL. This information can be attained from the VirusTotal API as described in Chapter 5.

Permutation-based feature importances: This thesis assessed feature importances using the MDI measure. It is also possible to assess the feature importances using other methods, such as permutation. It would be interesting to compare different results from the different measures and see if other feature importances measures will yield different important features. This can also help determine the feature importances for the features with low cardinality neglected by the MDI measure. Permutation importances can also be attained using the Python library Scikit-learn.

Chapter 7

Ethics

This chapter will assess the ethical aspects related to this thesis. A thorough assessment had to be done before, during, and after the experiment was conducted because of personal information in the dataset. As previously discussed, the ethical concerns resulted in a setup including an NTNU-based server. The setup itself will not be further described in this chapter, which will focus on the ethical concerns and put the ethical aspects of the thesis in relation to relevant ethical guidelines and practices. The chapter is divided into sections covering topics relevant to research involving personal information. First, Section 7.1 will describe what types of personal information the thesis handled and what measures had to be in place before the experiment could occur. Then, Section 7.2 will cover ethical considerations when conducting research and issues related to Internet-based research.

7.1 Pre-experiment

The experiment used a dataset which included personal information. In particular, IP address and geographical location [Sin20] were present in the dataset. Before processing this information, precautions must be taken to ensure that the participants' privacy is handled responsibly. NTNU provides several resources to help guide a research project involving personal information. The resources give an introduction to what is considered personal information, links to performing risk assessment, and a description stating which projects have to be reported to Sikt [Sikta], the Norwegian Agency for Shared Services in Education and Research. The NTNU website "Collection of personal data for research projects" [NTNUa] contains links to useful resources as well as a definition of what is considered personal information.

According to NTNU, personal data is information that can directly or indirectly identify a person [NTNUa]. Information that may indirectly identify an individual is information that can be combined to trace back to one individual. This includes information such as geographical location and nationality. Directly identifiable

information is information that, without further context or information, may identify an individual. IP addresses are classified as directly identifiable information and, thus, require this thesis to take appropriate actions in data management [NTNUa].

Actions that had to be in place before the experiment started included a risk assessment which evaluated potential risks associated with handling personal information [NTNUa; NTNUd]. The risk assessment helped ensure that relevant regulations and guidelines in relation to personal information handling were upheld throughout the thesis. We considered practical aspects such as the storage and processing of personal data in the risk assessment and got an overview of potential events that could happen during the thesis. Furthermore, the risk assessment considered the possible consequences of an event and compared them with the probability that the event would occur. An elaboration of the probability, consequence and coherent risk levels is available in Table 7.1 and Table 7.2, respectively. The risk level is computed as a result of the multiplicative of the probability and consequence levels. Following the risk assessment, relevant countermeasures were put in place where required. The risk assessment as a whole is available in Appendix F.

Table 7.1: Consequence and probability levels according to NTNU risk assessment template [NTNUd]

Level	Consequence	Probability
1	Mild. Has negligible harmful effects for individuals or the institution	Very unlikely - will most likely not happen during the project period
2	Less serious. Has certain harmful effects for individuals or the institution. Example: Unauthorised exposure of a small amount of general personal data	Unlikely - may occur during the project period
3	Serious. Has noticeable harmful effects for individuals or the institution. Example: Unauthorised exposure of confidential/sensitive personal data	Likely - likely to happen during the project period
4	Very serious. Has major damaging effects for individuals or the institution. Example: Unauthorised exposure of large amounts of confidential/sensitive personal data	Very likely - could potentially happen several times during the project period

Table 7.2: Risk levels according to NTNU risk assessment template [NTNUd]

Risk value	Risk level	Measures
1-3	Low risk	No measures required
4-7	Moderate risk	New measures should be considered
8-16	High risk	New measures must be introduced

The risk assessment revealed three events with moderate risk, as shown in table 7.3. First, the risk assessment revealed the unwanted event of personal data being accessible through open computers and thus being available for unauthorised personnel. The event was evaluated to have a consequence level of 2, meaning it is less severe and may expose some common personal information. The probability level of the event was set to 2, meaning that it is unlikely, but *may* occur during the project. This resulted in a total risk evaluation level of 4 (moderate risk). The second event is related to data analysis and covers downloading data to a private computer from secure storage. Finally, the risk assessment revealed the unwanted event of personal data remaining on the used storage medium after the project had ended.

Moderate risk is considered more severe than wanted, and countermeasures should be considered, as shown in Table 7.2. To avoid the events of storage on private computers, the dataset containing personal information was only downloaded to the NTNU SkyHigh server, as described in Chapter 5. Furthermore, a strict regime of locking the computer used to access the data through the SSH connection with the server was upheld throughout the project's duration. Following the described countermeasures, also represented in Table 7.4, the probability level of unwanted data disclosure was reduced. To mitigate the potential data disclosure following incomplete deletion at the project's end, it was suggested that both the author and the supervisor do a follow-up in the final stage of the thesis.

Table 7.3: Moderate risk events discovered through the risk assessment

Phase of data processing	Unwanted event	Risk Level		
		Consequence	Probability	Risk
Storage	Personal data is available through a computer which is left open and accessible for unauthorised personnel	2	2	4
Processing/analysis	Personal information is downloaded from secure storage to an insecure private computer	2	2	4
End	Personal information is left in the secure storage after project closure	2	2	4

Table 7.4: Unwanted events and countermeasures to reduce risk level

Unwanted event	Countermeasure	New risk Level		
		Consequence	Probability	Risk
Personal data is available through a computer which is left open and accessible for unauthorised personnel	Computer used to access data containing personal information must always be locked when not used. It should be actively prevented insight when the dataset is processed	2	1	2
Personal information is downloaded from secure storage to an insecure private computer	According to NTNU guidelines, personal computers shall not be used for storage of personal data [NTNUa]. The dataset will be downloaded directly to the server	2	1	2
Personal information is left in the storage after project closure	Both the student and the responsible supervisor verify that the data has been deleted and the server shut down at the project end	2	1	2

To conduct a valuable and trustworthy risk assessment, it was important to get an overview of best practices for handling personal data in research projects. As a starting point, the author completed NTNU's course "Introduction to personal information in research" [NTNUb]. The course provided an overview of relevant regulations when handling personal information, principles of ethical and responsible

research, knowledge of relevant tools and resources provided by NTNU, and support services. Furthermore, the course stated formalities, such as reporting to Sikt, which must be in place before the project can start [NTNUb; Sikt^c].

A notification form was sent to Sikt before performing the experiment. The notification form included details about the thesis and, in particular, how it handled the personal information in the dataset. Explaining the thesis in the notification form gave us a clear insight into how we were using the data, how it was stored, and to what extent the data would be reflected in the result section of the thesis. Following a discussion with the responsible processor at Sikt, we specified in greater detail what types of information would be used. Furthermore, the notification form facilitated reflection on ethical issues such as who is included in the dataset in use, how consent is handled and how the data is managed throughout the project. A further assessment of ethical concerns related to consent and Internet research will follow in the next subsection. The notification form as a whole is included in Appendix E.

Sikt also provides a template for data management [Sikt^b]. The template helps guide projects in all aspects of data management, from start to end, such that guidelines and regulations from the Research Council of Norway (“Forskningsrådet”) [RCN] and the European Union (EU) are met. We used the template to serve as a guide for data management throughout this thesis. The data management plan included information about the dataset, how it was acquired, and what personal information it contained. Our data management plan is included in Appendix G. The plan also allowed reflection on the societal benefits the thesis brings, as well as its intention. Furthermore, we have specified which ethical guidelines are relevant to the thesis and data processing. The template from Sikt [Sikt^b] provided links to resources where the ethical guidelines could be explored in more detail. This is assessed in the following subsection.

7.2 Research Ethical Considerations

7.2.1 General Research Ethics

Conducting research projects requires the participants to act according to standards and values set by not only the affiliated institution, but also by the government and international regulations and laws. Unlike legal considerations, ethical aspects involve aspects that are not always black or white in terms of what is right and wrong. In many cases, the research may touch grey areas where the researchers themselves must determine if the research is justifiable.

In general, ethically sound research involves four primary principles, as shown in the list below [NREC19b]. The research must be based on respect, meaning

everyone involved in the research project, whether as informants or otherwise, is treated with respect. Good consequences are based on the outcome of the research project and include limiting the possible unfortunate consequences of the project to benefit society. Any adverse consequence shall be limited to an acceptable level. The research project must also be fair, which means that both the preparation and execution of the project are done in a sound way. The last principle is based on the researcher's integrity and assures the researcher is adhering to relevant norms, acting responsibly, and transparently performing research for any interested colleagues, public actors, or government [NREC19b].

1. Respect
2. Pursue good consequences
3. Fairness
4. Integrity

This thesis was conducted with high academic standards, adhering to good research ethics. It has done so by following the aforementioned principles. In particular, the respect and fairness principles are handled by protecting personal information present in the dataset. This was facilitated by the substantial preparational work, such as the notification form, data management plan, and risk assessment. The thesis intends to serve society by investigating opportunities to detect malicious websites. In combination with limiting privacy disclosure under the respect principle, this yields good consequences. We have performed an experiment which is clearly presented in general in Chapter 4, and in detail in Chapter 5. Combined with proper citations and acknowledgement of the dataset, this assures the reproducibility and transparency of the research project.

In addition to the aforementioned principles, the National Research Ethics Committee [NREC] specifies several goals for research activities. A discussion of the most relevant goals in light of this thesis will follow. First, the research shall be a *quest for truth*, where openness, peer review, and documentation are fundamental preconditions for facilitating new knowledge. *Academic freedom* is a goal that enables the researcher to conduct an experiment with freedom in method selection and choice of topic. The affiliated research institution shall help facilitate this freedom and support the researchers in their choice of method and topic. The *quality* goal states that the research shall be conducted at an appropriate academic level. Hereunder, the researchers must possess the required competence, coin relevant research questions, and ensure a sound and appropriate project implementation regarding data collection and processing. Data processing must be in compliance with applicable *laws and*

regulations. In addition, the researchers must generally assure a *voluntary informed consent* in advance of data processing where the data include information that may be linked to any individual in the experiment [NREC19b].

The author emphasises the importance of the quality goal, which includes appropriate data collection, processing, and storage. In collaboration with the supervisor and NTNU, this issue was solved as previously described in Section 5.1 by handling all personal information on a remote server only accessible by the author through an SSH connection. Here it is clear to see the involvement of the affiliated institution, as described in the academic freedom goal. In addition to the given regulations from NTNU when processing personal information, this ensures compliance with relevant laws and regulations. However, ensuring voluntary, informed consent is not always possible when conducting Internet-based research. The ethical dilemmas of consent in this thesis are further assessed in the following subsection.

7.2.2 Internet Research Ethics

The National Research Ethics Committee has issued a guide to Internet research ethics. The guide aims to assist ethical reflection in Internet-based research and promote responsible and ethical practices among researchers and research institutions [NREC19a]. The guidelines serve as additional guidelines to the general ones described in the previous subsection. Internet-based research does not raise completely new ethical issues, and it is therefore necessary to ensure adherence to general norms and guidelines. However, Internet-based research raises unique circumstances which require additional ethical thought.

As a fundamental guideline, all research shall be based on human dignity and human rights [NREC19a]. In addition to general norms and values such as dignity, freedom, solidarity, and trust, the following factors are especially relevant in Internet-based research ethics: accessibility in the public sphere, the sensitivity of the information, the vulnerability of the participants, and interaction with the participants. Furthermore, communication through the Internet raises additional ethical considerations in assessing how information is stored and processed. The guideline presents the Internet-related ethical considerations in five distinct areas, as presented in the following list [NREC19a]:

1. Distinction between public and private
2. Concern for children and vulnerable groups
3. Responsibility to inform and obtain consent
4. Responsibility for confidentiality and anonymisation

5. Sharing of data and big data

In an online world, the distinction between what is considered public and private is reduced. It is evident that information shared between two people in the real world is private information, and although the conversation is happening at a public location, the exchanged information is not necessarily public. Compared to online forums, knowing precisely who has access to the information one share may be difficult. However, the greater the access restrictions, the less public the information is considered to be. This thesis has conducted research using a dataset including IP addresses, which are considered personal information. It is, however, not interesting for this thesis to investigate the IP addresses of individuals. The IP addresses included in the project are strictly related to the registered domains. Hosting the domain may be done locally. However, it is most likely done on a server, linking the IP address to the server, not an individual person. Furthermore, the only service hiding the IP address from any end-user is the DNS-server, which translates the queried domain to an IP address. Therefore, we consider IP addresses to have high expected publicity and thus deem it ethically responsible to conduct this thesis using the stated dataset.

A significant distinction between public and private in online environments is the level of expected publicity for the individual providing the information. This thesis has used domain names to query VirusTotal for information about the domain, including registrant information. Unlike IP addresses, this information does not necessarily have a high level of expected publicity. However, information which can be obtained and thus traced back to one individual from WHOIS records has been drastically reduced following the implementation of GDPR. The author emphasises the importance of protecting personal life and information and embraces the limitations of information availability. It is, however, interesting to see how the information still available can be used in this experiment.

Use of personal information, in general, requires consent from the participating individuals. There are, however, exceptions. In many cases, obtaining consent from all participating actors may be difficult, or even impossible, to achieve. And thus, exceptions from the requirement to obtain permission can be granted. It is also challenging to achieve consent in a satisfactory manner when conducting Internet-based research. In particular, it is difficult to ensure that the individual giving consent has a complete understanding of the research project and thus provides an informed consent [NREC19a]. Exceptions can be made for research which is in the public interest and where the practical aspects of obtaining consent complicate the process. These cases include research on criminal activity, which this thesis seeks to discover through malicious website detection. Following the notification form to Sikt, the responsible processor agreed that the responsibility to inform and obtain

consent is exempted in this project. The evaluation is part of the notification form in Appendix E (“Vurdering av behandling av peronopplysninger”). Difficulties in providing information to the participants are one of the main arguments for the exemption. Furthermore, when there are many registered participants, it is infeasible to inform every participant. For this thesis, the personal information itself serves no interest and is only used to obtain more information from VirusTotal. Also, as confirmed by Sikt in the evaluation, the personal information used has a high level of expected publicity.

Good research ethics requires projects to be transparent and verifiable, i.e., it should be possible for other researchers to replicate the project and thus verify the results. Sharing data is important to facilitate this process. However, when performing research on Internet-related topics, several ethical considerations must be assessed before sharing data. If the dataset contains personal information, restrictions on publishing it will apply. The researchers must consider what is ethically appropriate for the project. Given the personal information in the dataset used in this thesis, the author has decided not to publish the assembled dataset. However, the author emphasises that the original dataset is available at Data in Brief [Sin20], and through the method explained in Chapter 5, the work should be verifiable by a third party.

Given the importance of shielding the participants in a research project, research ethical considerations present an excellent opportunity for the researcher to reflect on the work before, during, and after conducting it. The author has dedicated a substantial amount of time to the ethical aspects of this thesis. Ethical reflection has provided a foundation for the mitigation of privacy disclosure in this thesis. It has done so through a risk assessment, a notification form, and a data management plan. Ethical considerations are assessed throughout the project, resulting in a sound thesis that does not expose personal information. Furthermore, it acts according to leading norms and regulations set by both NTNU [NTNUa], The Research Council of Norway [RCN], and the National Research Ethics Committees [NREC]. This has, to a great extent, been facilitated by the resources provided by Sikt, which have been discussed in this chapter in relation to relevant research ethical considerations.

References

- [Alt17] J. Althouse, Salesforce Engineering Easily Identify Malicious Servers on the Internet with JARM, Nov. 2017. [Online]. Available: <https://engineering.salesforce.com/easily-identify-malicious-servers-on-the-internet-with-jarm-e095edac525a/> (last visited: May 1, 2023).
- [Bak22] M. S. Bakken, «Webpage clustering using infrastructure-based features», Department of Information Security, Communication Technology, NTNU – Norwegian University of Science, and Technology, Project report in TTM4502, Dec. 2022.
- [BB12] J. Bergstra and Y. Bengio, «Random search for hyper-parameter optimization», *Journal of machine learning research*, vol. 13, no. 2, pp. 281–305, 2012.
- [BDF18] F. Badii, R. Dammak, and A. Férdeline, Internet Governance Project WHOIS afraid of the dark? Truth or illusion, let’s know the difference when it comes to WHOIS, Apr. 2018. [Online]. Available: <https://www.internetgovernance.org/2018/04/25/whois-afraid-dark-truth-illusion-lets-know-difference-comes-whois/> (last visited: May 12, 2023).
- [Ber19] D. Berrar, «Cross-validation», in *Encyclopedia of Bioinformatics and Computational Biology*. Academic Press, Sep. 2019, vol. 1, pp. 542–545.
- [BG16] S. Bleik and S. Gauher, Revolutions Computing Classification Evaluation Metrics in R, Mar. 2016. [Online]. Available: https://blog.revolutionanalytics.com/2016/03/com_class_eval_metrics_r.html (last visited: May 20, 2022).
- [BGMZ97] A. Z. Broder, S. Glassman, *et al.*, «Syntactic clustering of the web», *Computer Networks and ISDN Systems*, vol. 29, no. 8, pp. 1157–1166, Sep. 1997.
- [Bre01] L. Breiman, «Random forests», *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001.
- [Bro20] J. Brownlee, *Machine Learning Mastery train-test split for evaluating machine learning algorithms*, Jun. 2020. [Online]. Available: <https://machinelearningmastery.com/train-test-split-for-evaluating-machine-learning-algorithms/> (last visited: Apr. 20, 2023).

- [Bro21] J. Brownlee, *Machine Learning Mastery strong learners vs. weak learners in ensemble learning*, May 2021. [Online]. Available: <https://machinelearningmastery.com/strong-learners-vs-weak-learners-for-ensemble-learning/> (last visited: May 3, 2023).
- [CCZ+22] Y. Cheng, T. Chai, *et al.*, «Detecting malicious domain names with abnormal whois records using feature-based rules», *Computer Journal*, vol. 65, no. 9, pp. 2262–2275, Sep. 2022.
- [Cho20] J. Chong, *Towards Data Science Guide to Encoding Categorical Features Using Scikit-Learn For Machine Learning*, Dec. 2020. [Online]. Available: <https://towardsdatascience.com/guide-to-encoding-categorical-features-using-scikit-learn-for-machine-learning-5048997a5c79> (last visited: May 18, 2023).
- [Clu19] G. Cluley, *Tripwire Block newly-registered domains to reduce security threats in your organisation*, Aug. 2019. [Online]. Available: <https://www.tripwire.com/state-of-security/block-newly-registered-domains-to-reduce-security-threats-in-your-organisation> (last visited: May 19, 2022).
- [CM14] R. Clayton and T. Mansfield, «A study of whois privacy and proxy service abuse», in *13th Workshop on the Economics of Information Security*, 2014.
- [CSF+08] D. Cooper, S. Santesson, *et al.*, RFC 5280 Internet X. 509 public key infrastructure certificate and certificate revocation list (CRL) profile, May 2008. [Online]. Available: <https://www.rfc-editor.org/rfc/rfc5280.html> (last visited: May 14, 2022).
- [Dai04] L. Daigle, RFC 3912 WHOIS Protocol Specification, Sep. 2004. [Online]. Available: <https://www.rfc-editor.org/rfc/rfc3912.html> (last visited: Nov. 4, 2022).
- [DD03] Z. Dong and Q. Dong, «HowNet - a hybrid language and knowledge resource», *International Conference on Natural Language Processing and Knowledge Engineering*, pp. 820–824, Oct. 2003.
- [Dom12] P. Domingos, «A few useful things to know about machine learning», *Communications of the ACM*, vol. 55, no. 10, pp. 78–87, 2012.
- [DomainTools] DomainTools API Documentation. [Online]. Available: <https://www.domaintools.com/resources/api-documentation/> (last visited: Oct. 20, 2022).
- [DR08] T. Dierks and E. Rescorla, RFC 5246 The Transport Layer Security (TLS) Protocol Version 1.2, Aug. 2008. [Online]. Available: <https://www.rfc-editor.org/rfc/rfc5246> (last visited: May 14, 2022).
- [ER04] T. Elomaa and J. Rousu, «Efficient multisplitting revisited: Optima-preserving elimination of partition candidates», *Data Mining and Knowledge Discovery*, vol. 8, pp. 97–126, 2004.
- [Eur21] Europol, *Iocta, Internet Organised Crime Threat Assessment : 2021*. Europol, 2021.

- [Fin23] L. Finsrud, Telenor 18 000 nye falske nettsider – hver eneste dag, Mar. 2023. [Online]. Available: https://www.online.no/sikkerhet/nye-falske-nettsider/?cid=11897_mob_nyb_upa_all_03042023_nul_falsk (last visited: May 19, 2022).
- [FKK11] A. Freier, P. Karlton, and P. Kocher, RFC 6101 The Secure Sockets Layer (SSL) Protocol Version 3.0, Aug. 2011. [Online]. Available: <https://www.rfc-editor.org/rfc/rfc6101.html> (last visited: May 14, 2022).
- [FQYZ22] J. Feng, Y. Qiao, *et al.*, «Detecting phishing webpages via homology analysis of webpage structure», *PeerJ. Computer science*, vol. 8, pp. 1–23, Feb. 2022.
- [FS96] Y. Freund and R. E. Schapire, «Experiments with a new boosting algorithm», in *Proceedings of the Thirteenth International Conference on International Conference on Machine Learning*, Jul. 1996, pp. 148–156.
- [GeoIP] Maxmind GeoIP2 Databases. [Online]. Available: <https://www.maxmind.com/en/geoip2-databases> (last visited: May 1, 2023).
- [Goo23] Google, Google Safe Browsing, 2023. [Online]. Available: <https://developers.google.com/safe-browsing> (last visited: Apr. 24, 2023).
- [Gup20] S. Gupta, Towards Data Science Pros and cons of various Machine Learning algorithms, Feb. 2020. [Online]. Available: <https://towardsdatascience.com/pros-and-cons-of-various-classification-ml-algorithms-3b5bfb3c87d6> (last visited: May 2, 2023).
- [HHK+20] A. Hounsel, J. Holland, *et al.*, «Identifying disinformation websites using infrastructure features», *arXiv*, Sep. 2020.
- [Hof10] S. Hoffman, «An illustration of hashing and its effect on illegal file content in the digital age», *Intellectual Property & Technology Law Journal*, vol. 22, no. 4, pp. 6–14, Apr. 2010.
- [HRA+22] I. U. Hassan, H. A. Raja, *et al.*, «Significance of machine learning for detection of malicious websites on an unbalanced dataset», *Digital*, vol. 2, no. 4, pp. 501–519, 2022.
- [HREJ14] L. S. Huang, A. Rice, *et al.*, «Analyzing forged ssl certificates in the wild», in *2014 IEEE Symposium on Security and Privacy*, 2014, pp. 83–97.
- [HS15] M. Hossin and M. N. Sulaiman, «A review on evaluation metrics for data classification evaluations», *International Journal of Data Mining & Knowledge Management Process*, vol. 5, no. 2, pp. 1–11, 2015.
- [Hun07] J. D. Hunter, «Matplotlib: A 2d graphics environment», *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.
- [Haa23] K. Haan, Forbes Top Website Statistics For 2023, Feb. 2023. [Online]. Available: <https://www.forbes.com/advisor/business/software/website-statistics/> (last visited: Jun. 1, 2022).
- [ICANN] ICANN Policy Issue Brief - gTLD WHOIS. [Online]. Available: <https://www.icann.org/resources/pages/whois-2012-06-14-en> (last visited: May 12, 2023).

- [ICB+12] L. Invernizzi, P. M. Comparetti, *et al.*, «Evilseed: A guided approach to finding malicious web pages», *2012 IEEE Symposium on Security and Privacy*, pp. 428–442, 2012.
- [IntBL] Interpol Baseline List. [Online]. Available: <https://www.interpol.int/Crimes/Crimes-against-children/Blocking-and-categorizing-content> (last visited: Oct. 11, 2022).
- [ISO3166] ISO 3166 Country Codes. [Online]. Available: <https://www.iso.org/iso-3166-country-codes.html> (last visited: May 1, 2023).
- [Kaggle] C. Urcuqui, Kaggle Malicious and Benign Websites. [Online]. Available: <https://www.kaggle.com/datasets/xwolf12/malicious-and-benign-websites?resource=download> (last visited: Mar. 20, 2023).
- [KK06] S. Kotsiantis and D. Kanellopoulos, «Discretization techniques: A recent survey», *GESTS International Transactions on Computer Science and Engineering*, vol. 32, no. 1, pp. 47–58, 2006.
- [KK07] I. Kononenko and M. Kukar, *Machine Learning and Data Mining*. Elsevier Science, 2007.
- [KKS16] M. Kuyama, Y. Kakizaki, and R. Sasaki, «Method for detecting a malicious domain by using whois and dns features», in *The Third International Conference on Digital Security and Forensics (DigitalSec2016)*, Sep. 2016, pp. 74–80.
- [Kot07] S. B. Kotsiantis, «Supervised machine learning: A review of classification techniques», *Emerging artificial intelligence applications in computer engineering*, vol. 160, no. 1, pp. 3–24, 2007.
- [Kre18] B. Krebs, KrebsOnSecurity Security Trade-Offs in the New EU Privacy Law, Apr. 2018. [Online]. Available: <https://krebsonsecurity.com/2018/04/security-trade-offs-in-the-new-eu-privacy-law/> (last visited: Nov. 1, 2022).
- [Kur20] V. Kurama, Paperspace A Guide to AdaBoost: Boosting To Save The Day, 2020. [Online]. Available: <https://blog.paperspace.com/adaboost-optimize-r/> (last visited: May 3, 2023).
- [LFS+15] S. Liu, I. Foster, *et al.*, «Who is .com? learning to parse whois records», in *Proceedings of the 2015 Internet Measurement Conference*, ser. IMC '15, Tokyo, Japan: Association for Computing Machinery, 2015, pp. 369–380.
- [Lis14] P. Lischinsky, Machine learning lesson of the day – the “no free lunch” theorem, 2014. [Online]. Available: <https://chemicalstatistician.wordpress.com/2014/01/24/machine-learning-lesson-of-the-day-the-no-free-lunch-theorem/> (last visited: May 15, 2023).
- [LLZ+21] C. Lu, B. Liu, *et al.*, «From whois to whowas: A large-scale measurement study of domain registration privacy under the gdpr», in *Network and Distributed Systems Security (NDSS) Symposium*, Feb. 2021.

- [LNA17] G. Lemaître, F. Nogueira, and C. K. Aridas, «Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning», *Journal of Machine Learning Research*, vol. 18, no. 17, pp. 1–5, 2017.
- [Loe20] M. Loecher, «Unbiased variable importance for random forests», *Communications in Statistics - Theory and Methods*, vol. 51, pp. 1–13, May 2020.
- [Man13] D. Mankly, «Cybercrime as a service: A very modern business», *Computer Fraud & Security*, vol. 2013, no. 6, pp. 9–13, Jun. 2013.
- [McK10] W. McKinney, «Data Structures for Statistical Computing in Python», in *Proceedings of the 9th Python in Science Conference*, S. van der Walt and J. Millman, Eds., 2010, pp. 56–61.
- [MMS+21] N. Mehrabi, F. Morstatter, *et al.*, «A survey on bias and fairness in machine learning», *ACM Computing Surveys (CSUR)*, vol. 54, no. 6, pp. 1–35, 2021.
- [Mnemonic] Mnemonic mnemonic offers passive DNS data to the public. [Online]. Available: <https://www.mnemonic.io/company/whats-new/2015/mnemonic-offers-passive-dns-data-to-the-public/> (last visited: Oct. 24, 2022).
- [MRT18] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning*. MIT Press, 2018.
- [MTL+17] J. Mao, W. Tian, *et al.*, «Phishing-alarm: Robust and efficient phishing detection via page component similarity», *IEEE Access*, vol. 5, pp. 17 020–17 030, 2017.
- [NREC] National Research Ethics Committees About us. [Online]. Available: <https://www.forskningsetikk.no/en/about-us/> (last visited: Apr. 21, 2023).
- [NREC19a] National Research Ethics Committees A Guide to Internet Research Ethics, Jun. 2019. [Online]. Available: <https://www.forskningsetikk.no/en/guidelines/social-sciences-humanities-law-and-theology/a-guide-to-internet-research-ethics/> (last visited: Apr. 9, 2023).
- [NREC19b] National Research Ethics Committees General guidelines, Jul. 2019. [Online]. Available: <https://www.forskningsetikk.no/en/guidelines/general-guidelines/> (last visited: Apr. 4, 2023).
- [NSR22] H. Nikšić, D. Shah, and T. Rühnen, *Linux Manual page wget*, Dec. 2022. [Online]. Available: <https://man7.org/linux/man-pages/man1/wget.1.html> (last visited: Mar. 2, 2023).
- [NTNUa] NTNU Collection of personal data for research projects. [Online]. Available: <https://i.ntnu.no/wiki/-/wiki/English/Collection+of+personal+data+for+research+projects> (last visited: Apr. 21, 2023).
- [NTNUb] NTNU Introduksjon til personopplysninger i forskning. [Online]. Available: <https://digit.ntnu.no/courses/course-v1:NTNU+RD001+2022/course/> (last visited: Mar. 25, 2023).
- [NTNUc] NTNU OpenStack. [Online]. Available: <https://www.ntnu.no/wiki/display/skyhigh/Openstack+at+NTNU> (last visited: Mar. 2, 2023).

- [NTNUd] NTNU Risk assessment of research projects with personal data. [Online]. Available: <https://i.ntnu.no/wiki/-/wiki/English/Risk+assessment+of+research+projects+with+personal+data> (last visited: Apr. 21, 2023).
- [Oli19] R. V. A. Oliemans, «Whois versus gdpr», 2019.
- [PDNA09] Microsoft New Technology Fights Child Porn by Tracking Its "PhotoDNA", 2009. [Online]. Available: <https://news.microsoft.com/2009/12/15/new-technology-fights-child-porn-by-tracking-its-photodna/> (last visited: Oct. 17, 2022).
- [pdtea20] T. pandas development team, *Pandas-dev/pandas: Pandas*, version latest, Feb. 2020. [Online]. Available: <https://doi.org/10.5281/zenodo.3509134>.
- [PVG+11] F. Pedregosa, G. Varoquaux, *et al.*, «Scikit-learn: Machine learning in Python», *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [PYL+20] S. Pang, J. Yao, *et al.*, «A text similarity measurement based on semantic fingerprint of characteristic phrases», *Chinese Journal of Electronics*, vol. 29, no. 2, pp. 233–241, Mar. 2020.
- [RCN] The Research Council of Norway. [Online]. Available: <https://www.forskningsradet.no/en/> (last visited: Mar. 25, 2023).
- [RF06] A. Ramachandran and N. Feamster, «Understanding the network-level behavior of spammers», *Applications, Technologies, Architectures, and Protocols for Computer Communication: Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*, pp. 291–302, Sep. 2006.
- [RT14] S. Roopak and T. Thomas, «A novel phishing page detection mechanism using html source code comparison and cosine similarity», in *2014 Fourth International Conference on Advances in Computing and Communications*, 2014, pp. 167–170.
- [RY22] T. Rinne and T. Ylonen, *Linux Manual page scp*, Sep. 2022. [Online]. Available: <https://man7.org/linux/man-pages/man1/scp.1.html> (last visited: Mar. 2, 2023).
- [Sar21] I. H. Sarker, «Machine learning: Algorithms, real-world applications and research directions», *SN Computer Science*, vol. 2, no. 3, May 2021.
- [Sci:CV] Scikit Learn Cross-validation: evaluating estimator performance. [Online]. Available: https://scikit-learn.org/stable/modules/cross_validation.html (last visited: May 21, 2023).
- [Sci:DT] Scikit Learn Decision Trees. [Online]. Available: <https://scikit-learn.org/stable/modules/tree.html#> (last visited: May 2, 2023).
- [Sci:enc] Scikit Learn OrdinalEncoder. [Online]. Available: <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OrdinalEncoder.html> (last visited: Apr. 25, 2023).

- [Sci:ens] Scikit Learn Ensemble methods. [Online]. Available: <https://scikit-learn.org/stable/modules/ensemble.html#> (last visited: May 2, 2023).
- [Sci:Imp] Scikit Learn KNNImputer. [Online]. Available: <https://scikit-learn.org/stable/modules/generated/sklearn.impute.KNNImputer.html> (last visited: Apr. 25, 2023).
- [Sci:imp] Scikit Learn Imputation of missing values. [Online]. Available: <https://scikit-learn.org/stable/modules/impute.html#impute> (last visited: May 19, 2023).
- [Sci:KB] Scikit Learn KBinsDiscretizer. [Online]. Available: <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.KBinsDiscretizer.html> (last visited: Apr. 25, 2023).
- [Sci:MLP] Scikit Learn Neural network models (supervised). [Online]. Available: https://scikit-learn.org/stable/modules/neural_networks_supervised.html#neural-networks-supervised (last visited: May 4, 2023).
- [Sci:MLPC] Scikit Learn MLPClassifier. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html#sklearn.neural_network.MLPClassifier (last visited: May 18, 2023).
- [Sci:NB] Scikit Learn Naive Bayes. [Online]. Available: https://scikit-learn.org/stable/modules/naive_bayes.html (last visited: May 3, 2023).
- [Sci:QDA] Scikit Learn Linear and Quadratic Discriminant Analysis. [Online]. Available: https://scikit-learn.org/stable/modules/lda_qda.html#lda-qda (last visited: May 3, 2023).
- [Sci:RFC] Scikit Learn RandomForestClassifier. [Online]. Available: <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html#sklearn.ensemble.RandomForestClassifier> (last visited: May 18, 2023).
- [Sci:RSCV] Scikit Learn Tuning the hyper-parameters of an estimator. [Online]. Available: https://scikit-learn.org/stable/modules/grid_search.html#randomized-parameter-search (last visited: May 18, 2023).
- [SG17] A. K. Singh and N. Goyal, «Malcrawler: A crawler for seeking and crawling malicious websites», in *Distributed Computing and Internet Technology*, Springer International Publishing, Nov. 2017, pp. 210–223.
- [SG19] A. K. Singh and N. Goyal, «A comparison of machine learning attributes for detecting malicious websites», in *2019 11th International Conference on Communication Systems & Networks (COMSNETS)*, Jan. 2019, pp. 352–358.
- [Sikta] Sikt – Norwegian Agency for Shared Services in Education and Research. [Online]. Available: <https://sikt.no/en/about-sikt> (last visited: Apr. 21, 2023).
- [Siktb] Sikt Create a data management plan (DMP). [Online]. Available: <https://sikt.no/en/data-management-plan> (last visited: Apr. 21, 2023).

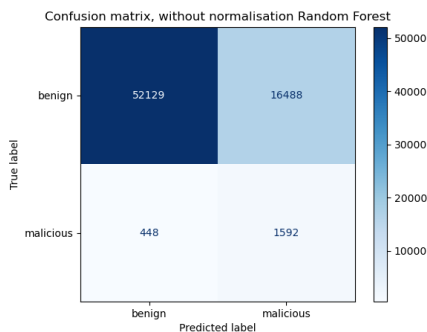
- [Sikt.c] Sikt Notification form for personal data. [Online]. Available: <https://sikt.no/en/notification-form-personal-data> (last visited: Mar. 25, 2023).
- [Sin20] A. K. Singh, «Malicious and benign webpages dataset», *Data in Brief*, vol. 32, Oct. 2020.
- [Spa22] D. J. Spajic, DataProt Piracy Is Back: Piracy Statistics for 2022, Oct. 2022. [Online]. Available: <https://dataprot.net/statistics/piracy-statistics/> (last visited: Oct. 31, 2022).
- [STS16] A. Singh, N. Thakur, and A. Sharma, «A review of supervised machine learning algorithms», in *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, 2016, pp. 1310–1315.
- [SWO+21] Y. Sakurai, T. Watanabe, *et al.*, «Identifying the phishing websites using the patterns of tls certificates», *Journal of Cyber Security and Mobility*, vol. 10, no. 2, pp. 451–486, Apr. 2021.
- [TCB18] I. Torroledo, L. D. Camacho, and A. C. Bahnsen, «Hunting malicious tls certificates with deep neural networks», in *Proceedings of the 11th ACM workshop on Artificial Intelligence and Security*, Oct. 2018, pp. 64–73.
- [The22] C. Theune, Python Package Index (PyPI) pycountry 22.3.5, Mar. 2022. [Online]. Available: <https://pypi.org/project/pycountry/> (last visited: May 1, 2023).
- [TK06] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, 3rd ed. Elsevier, 2006.
- [URLhaus] abuse.ch, URLhaus About. [Online]. Available: <https://urlhaus.abuse.ch/about/> (last visited: May 2, 2023).
- [Van18] S. Van Buuren, *Flexible imputation of missing data*. CRC press, 2018.
- [VKJM00] R. Venkatesan, S.-M. Koon, *et al.*, «Robust image hashing», *Proceedings 2000 International Conference on Image Processing*, vol. 3, pp. 664–666, Sep. 2000.
- [vt-py] VirusTotal, VirusTotal Welcome to vt-py’s documentation! [Online]. Available: <https://virustotal.github.io/vt-py/> (last visited: May 2, 2023).
- [VTapi] VirusTotal, VirusTotal API v3 Overview. [Online]. Available: <https://developers.virustotal.com/reference/overview> (last visited: May 1, 2023).
- [VTdom] VirusTotal, VirusTotal Domains. [Online]. Available: <https://developers.virustotal.com/reference/domains-1> (last visited: May 1, 2023).
- [VTssl] VirusTotal, VirusTotal SSL Certificate. [Online]. Available: <https://developers.virustotal.com/reference/ssl-certificate> (last visited: May 1, 2023).
- [VTwho] VirusTotal, VirusTotal Whois. [Online]. Available: <https://developers.virustotal.com/reference/whois> (last visited: May 1, 2023).
- [WBG16] B. G. Westlake, M. Bouchard, and A. Girodat, «How obvious is it? the content of child exploitation websites», *Deviant Behaviour*, vol. 38, no. 3, pp. 282–293, Jul. 2016.

- [WHLM13] P. A. Watters, A. Herps, *et al.*, «Icann or icant: Is whois an enabler of cybercrime?», in *2013 Fourth Cybercrime and Trustworthy Computing Workshop*, 2013, pp. 44–49.
- [WHOIS] Debian Package whois (5.5.14) intelligent WHOIS client. [Online]. Available: <https://packages.debian.org/sid/whois> (last visited: Nov. 4, 2022).
- [Wik23a] Wikipedia contributors, *Bayes' theorem* — *Wikipedia, the free encyclopedia*, 2023. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Bayes%27_theorem&oldid=1152352939 (last visited: May 3, 2023).
- [Wik23b] Wikipedia contributors, *Hash function* — *Wikipedia, the free encyclopedia*, 2023. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Hash_function&oldid=1158312728 (last visited: Jun. 11, 2023).
- [Wik23c] Wikipedia contributors, *Law of large numbers* — *Wikipedia, the free encyclopedia*, 2023. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Law_of_large_numbers&oldid=1151744784 (last visited: May 2, 2023).
- [Wik23d] Wikipedia contributors, *No free lunch theorem* — *Wikipedia, the free encyclopedia*, 2023. [Online]. Available: https://en.wikipedia.org/w/index.php?title=No_free_lunch_theorem&oldid=1154448734 (last visited: May 15, 2023).
- [Wik23e] Wikipedia contributors, *Tf-idf* — *Wikipedia, the free encyclopedia*, 2023. [Online]. Available: <https://en.wikipedia.org/w/index.php?title=Tf%E2%80%93idf&oldid=1143165699> (last visited: May 21, 2023).
- [Wik23f] Wikipedia contributors, *Unix time* — *Wikipedia, the free encyclopedia*, 2023. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Unix_time&oldid=1149173188 (last visited: Apr. 25, 2023).
- [WM97] D. Wolpert and W. Macready, «No free lunch theorems for optimization», *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997.
- [WW08] B. Wardman and G. Warner, «Automating phishing website identification through deep md5 matching», *2008 eCrime Researchers Summit*, pp. 1–7, Oct. 2008.
- [AAA+22] M. Aljabri, H. S. Altamimi, *et al.*, «Detecting malicious urls using machine learning techniques: Review and research directions», *IEEE Access*, vol. 10, pp. 121 395–121 417, 2022.

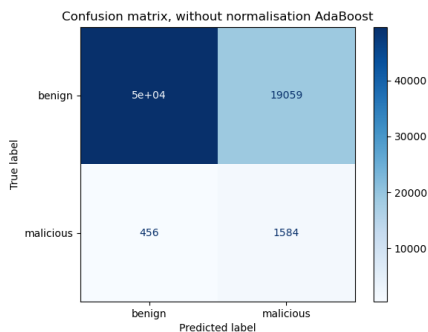
Appendix

Confusion Matrices (random undersampling)

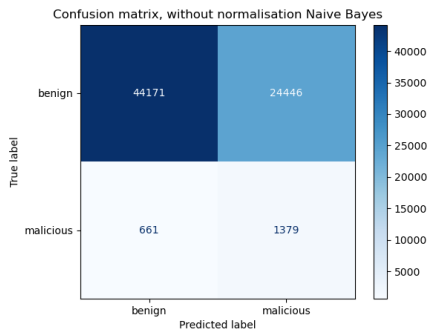
This appendix presents the confusion matrices from the classifiers using random undersampling as the resampling method to create a balanced training dataset.



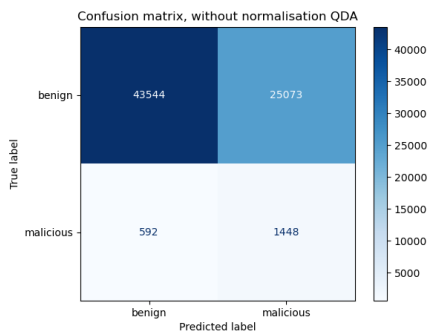
(a) Random Forest confusion matrix



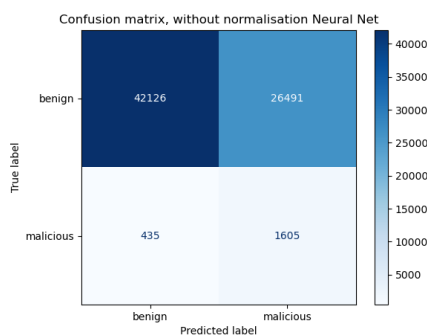
(b) AdaBoost confusion matrix



(c) Naive Bayes confusion matrix



(d) Quadratic Discriminant Analysis confusion matrix



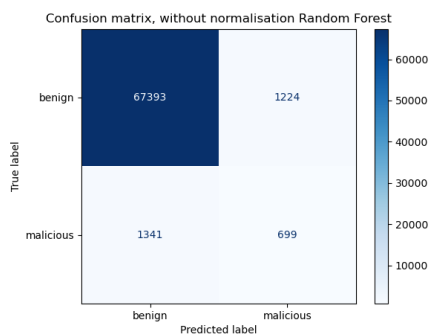
(e) Multi-Layer Perceptron confusion matrix

Figure A.1: Confusion matrices with random undersampling as balancing method for training data

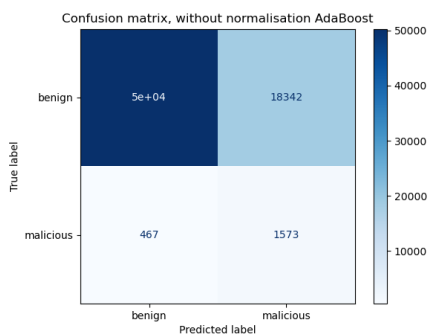
Appendix **B**

Confusion Matrices (random oversampling)

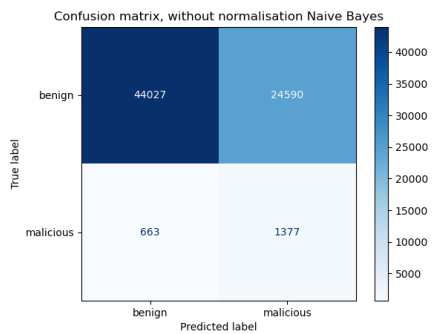
This appendix presents the confusion matrices from the classifiers using random oversampling as the method to create a balanced training dataset.



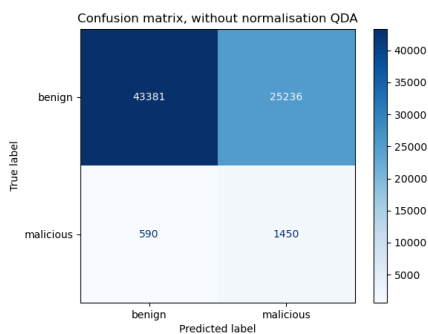
(a) Random Forest confusion matrix



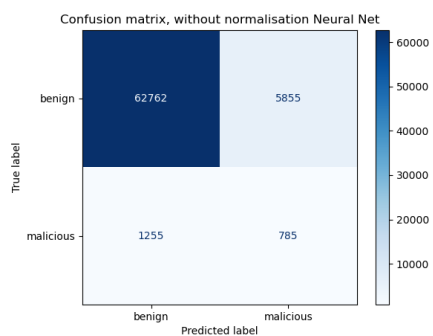
(b) AdaBoost confusion matrix



(c) Naive Bayes confusion matrix



(d) Quadratic Discriminant Analysis confusion matrix



(e) Multi-Layer Perceptron confusion matrix

Figure B.1: Confusion matrices with random oversampling as balancing method for training data

Appendix **C**

Result Plots

This appendix includes all evaluation metric plots used to assess the classifiers. The metrics are plotted both for random undersampling and random oversampling, as the resampling methods used to generate a balanced training dataset. Note that both plots are included in this appendix such that the reader may easily compare the different results from the different resampling strategies and that some of the plots are already presented in Chapter6.

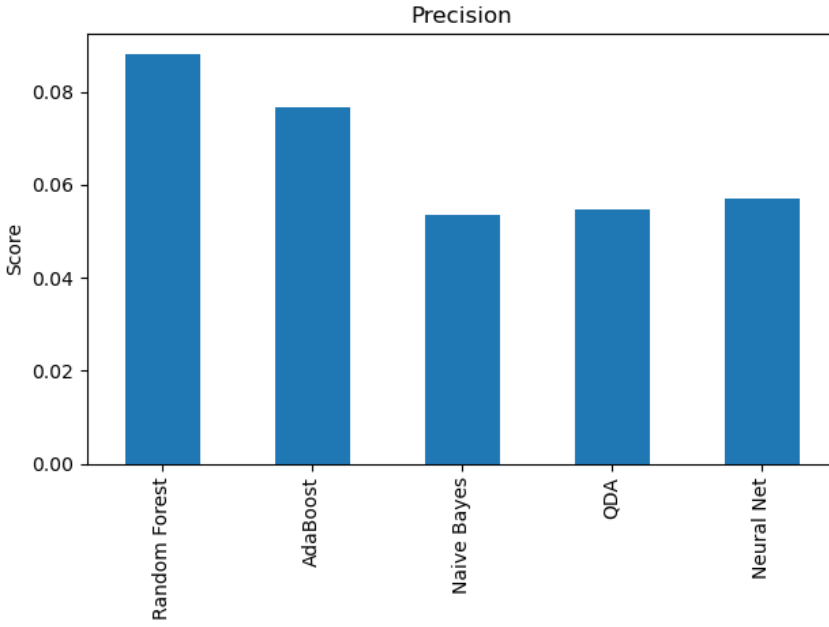


Figure C.1: Precision scores using random undersampling

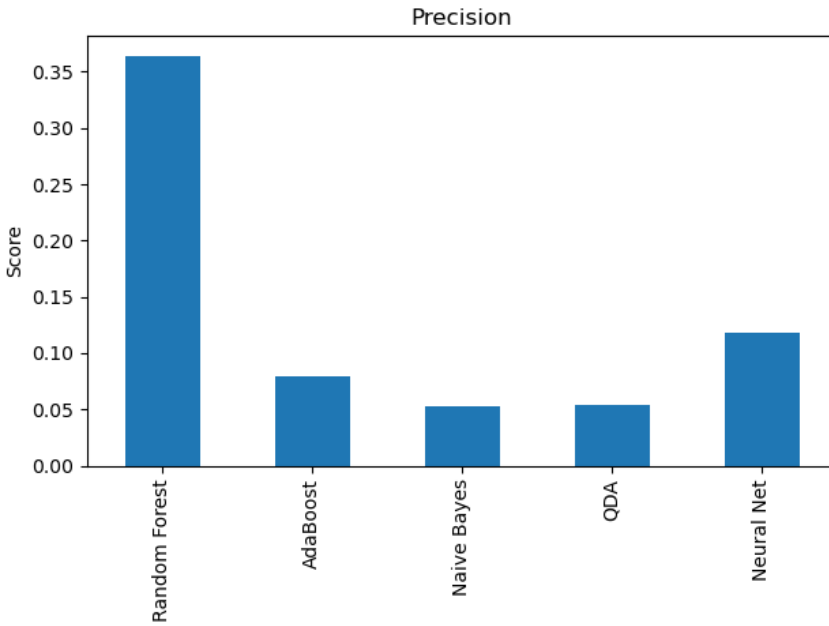


Figure C.2: Precision scores using random oversampling

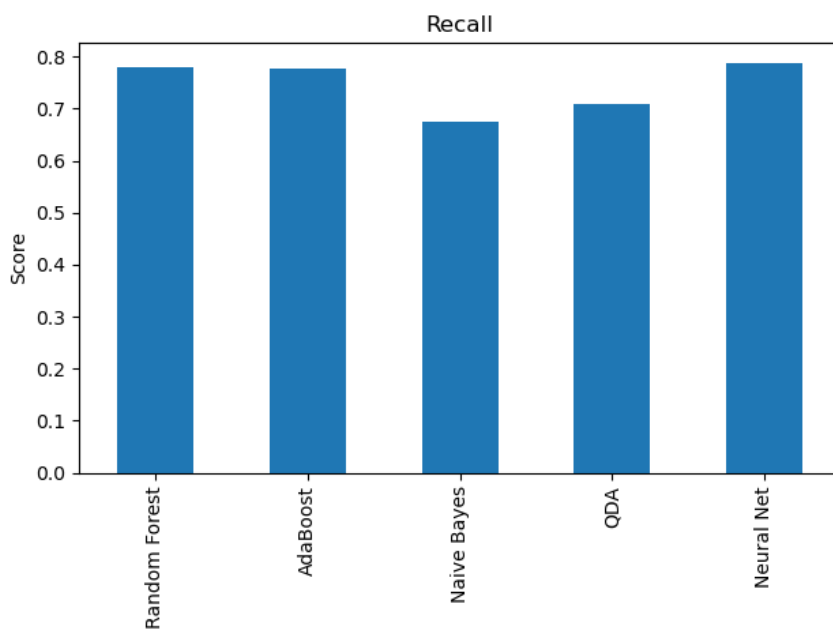


Figure C.3: Recall scores using random undersampling

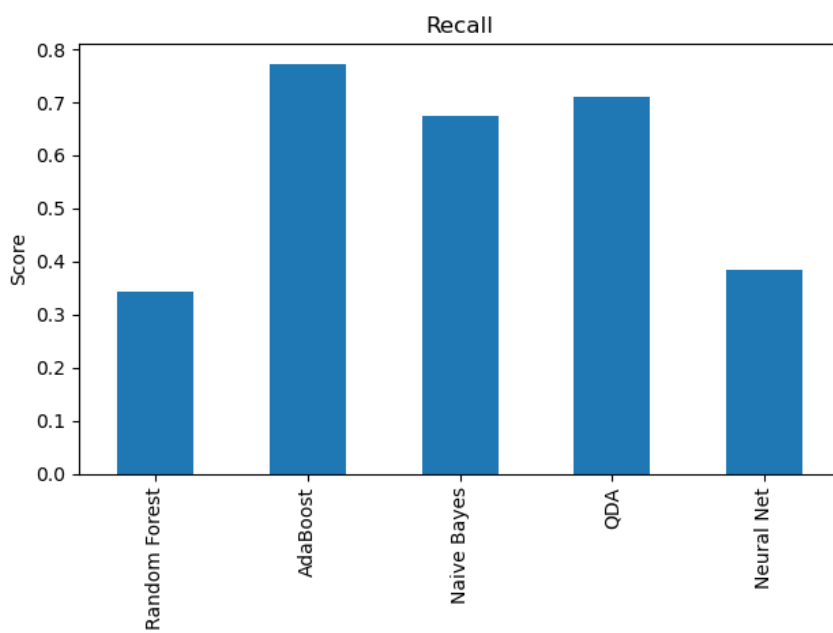


Figure C.4: Recall scores using random oversampling

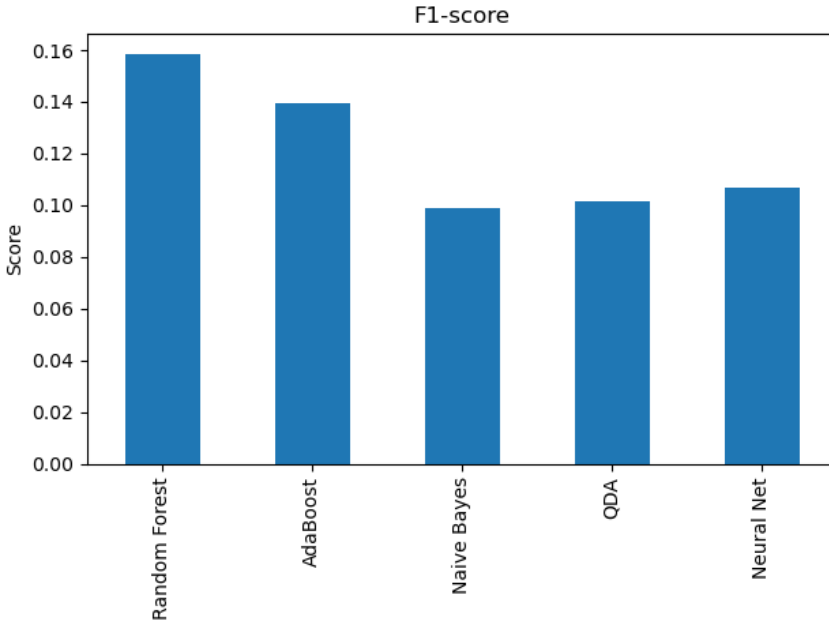


Figure C.5: F1-scores using random undersampling

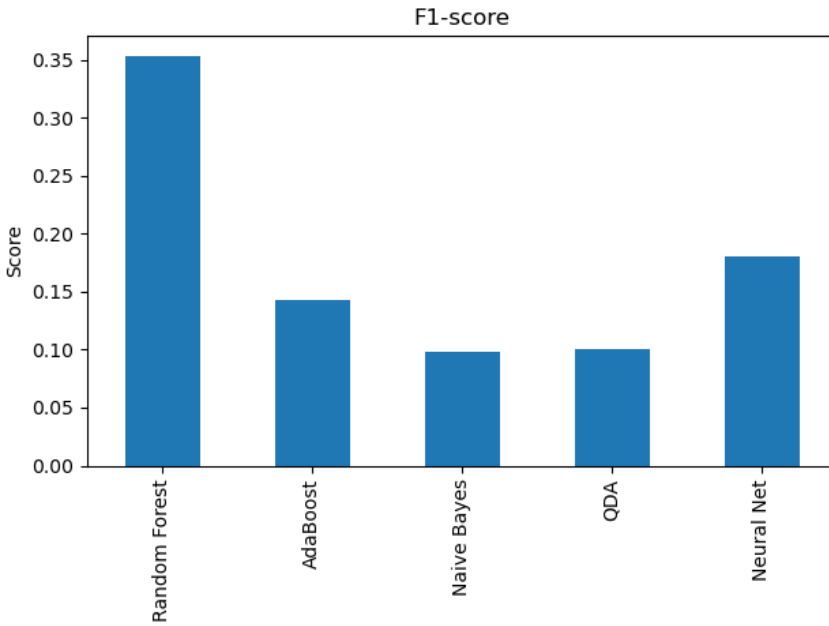


Figure C.6: F1-scores using random oversampling

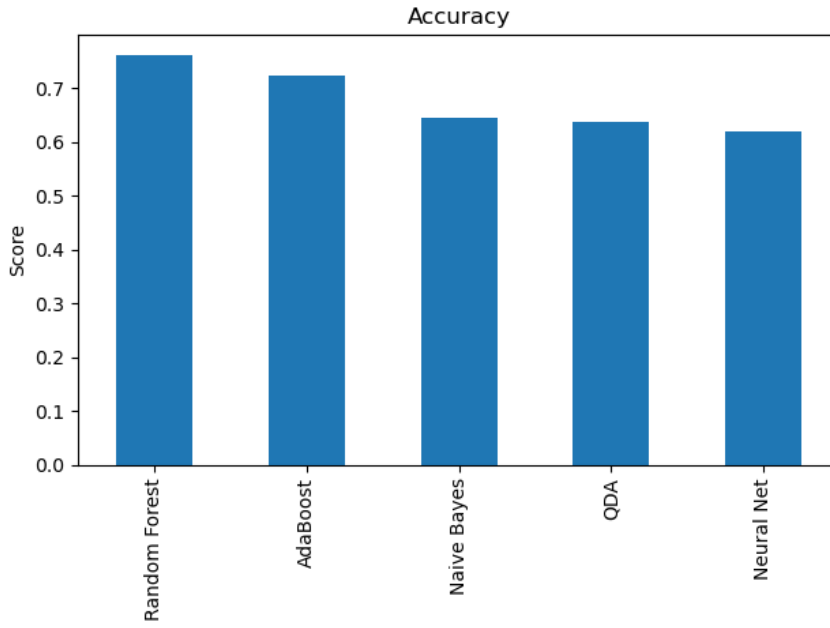


Figure C.7: Accuracy scores using random undersampling

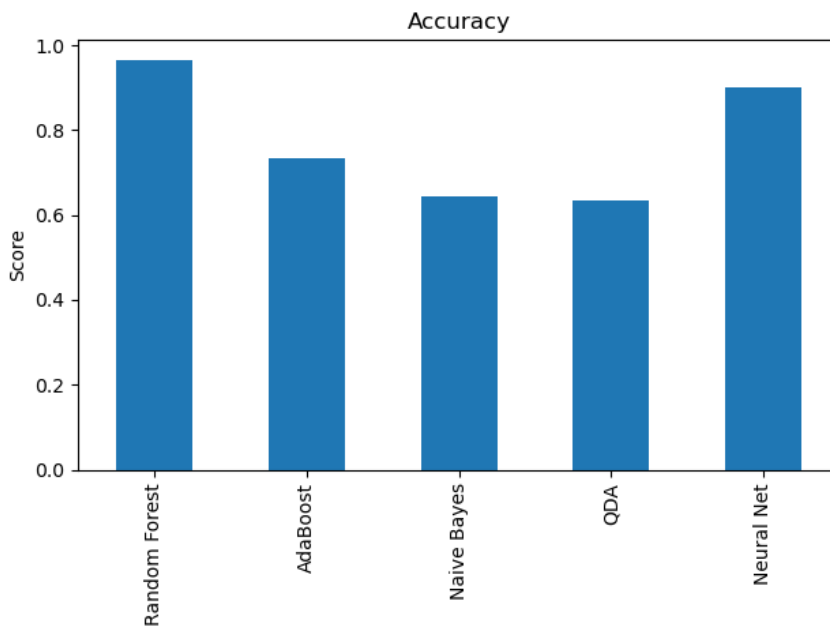


Figure C.8: Accuracy scores using random oversampling

Appendix D

Feature Importances

This appendix includes the feature importances plots derived from the Random Forest and AdaBoost classifiers. Figure D.1 shows the MDI values for the classification using random undersampling as the resampling method. Likewise, Figure D.2 shows the feature importances using random oversampling. Note that Figure D.1 is also presented, along with a discussion in Chapter 6.

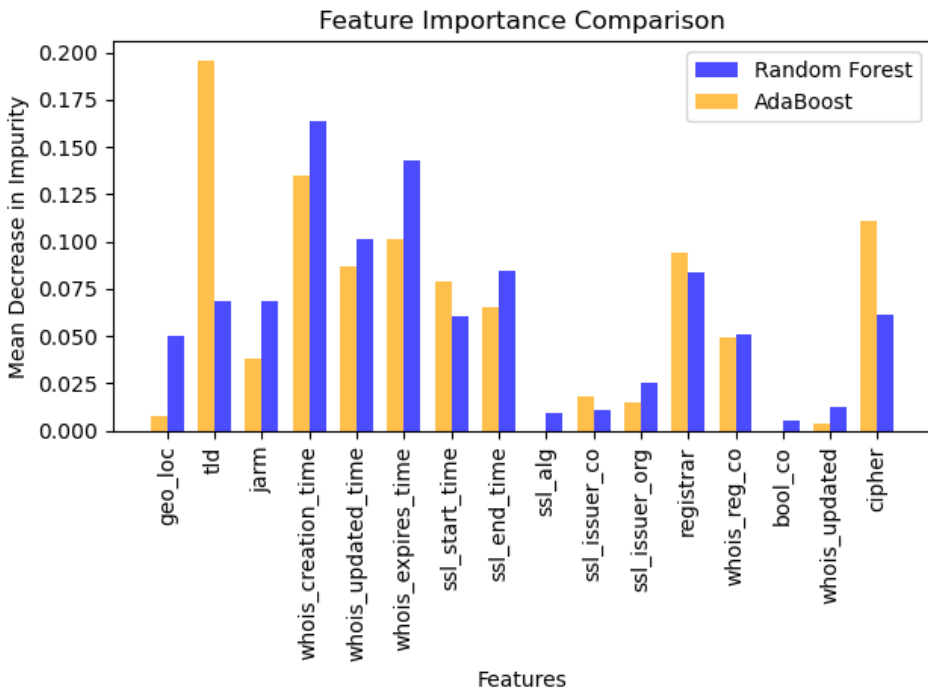


Figure D.1: Feature importances using random undersampling as resampling method for training data

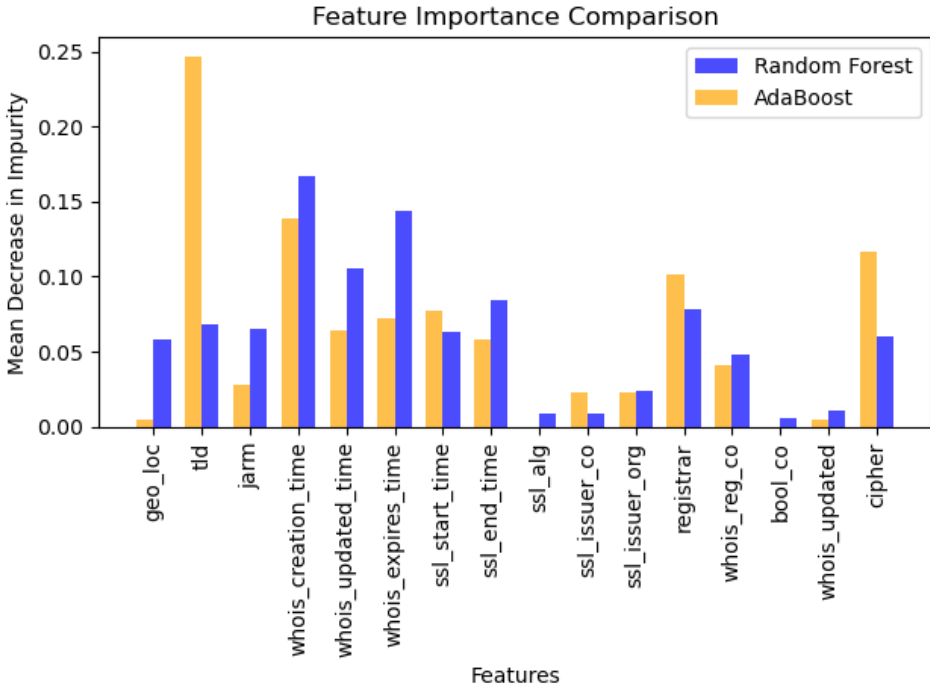


Figure D.2: Feature importances using random oversampling as resampling method for training data

Appendix **E**

Notification Form for Personal Data

This appendix presents the notification form sent to Sikt 30 days prior to data processing in accordance with the leading practices discussed in Chapter 7. The form is included as a whole to illustrate the considerations we had to make before the project. In particular, the form includes justifications for using personal information, responsibilities, and a description of who is included in the dataset. The form also states the general intentions of the project in a summary to ease the understanding of the processor at Sikt. The processor requested additional information before the project was approved. The additional information follows directly after the notification form and describes in detail what information is available in the existing dataset. Finally, the final evaluation of the project is included. It states that the project is in accordance with the practices and can go ahead as planned.

[Meldeskjema](#) / [Webpage fingerprinting](#) / Eksport

Meldeskjema

Referansenummer

999641

Hvilke personopplysninger skal du behandle?

- E-postadresse, IP-adresse eller annen nettidentifikator
- Gps eller andre lokaliseringsdata (elektroniske spor)

Prosjektinformasjon

Prosjekttittel

Webpage fingerprinting

Prosjektbeskrivelse

Master prosjekt ved NTNU, Institutt for informasjonssikkerhet og kommunikasjonsteknologi. Intensjonen med prosjektet er å utforske muligheter for å identifisere ondsinnede nettsider basert på infrastruktur opplysninger.

Begrunn hvorfor det er nødvendig å behandle personopplysningene

Prosjektet tar sikte på å identifisere like ondsinnede nettsider gjennom å se på opplysninger om domenet. Opplysningene skaffes gjennom et allerede eksisterende datasett (<https://doi.org/10.1016/j.dib.2020.106304>) og API til VirusTotal. For å sende forespørsler til VisusTotal API trenger prosjektet å benytte seg av personopplysninger (IP adresse, URL) som innhentes fra datasettet. Informasjon prosjektet tar sikte på å innhente er ikke personopplysninger, men inkluderer informasjon som WHOIS data, herunder når domenet ble registrert, når informasjon om de som registrerte domenet ble oppdatert, land og region for registrering av domenet. I tillegg vil andre opplysninger om domenet utforskes som mulige identifikatorer for å identifisere ulovlige nettsider, til eksempel informasjon om ciphers støttet av serveren i forbindelse med sikker kommunikasjon gjennom SSL/TLS. En komplett liste med informasjon som er tilgjengelig i datasettet og informasjon som er tenkt å innhentes vil beskrives i Tilleggsopplysninger. Personopplysningene (IP adresser og lokasjon) vil være nødvendig for å etterspørre informasjon om domenet. Det er ikke personopplysningene i seg selv som i dette tilfellet er av interesse, men de vil være nødvendige som et ledd for å skaffe nødvendig informasjon. En maskinlæringsmodell skal så brukes på informasjon knyttet til domener/nettsider for å identifisere tilsvarende ondsinnede nettsider. Dette kan bidra til at ulovlige nettsider enklere og raskere kan oppdages, for så å tas ned.

Ekstern finansiering

Ikke utfyllt

Type prosjekt

Studentprosjekt, masterstudium


Kontaktinformasjon, student

Martin Schiefloe Bakken, martisba@stud.ntnu.no, tlf:

Behandlingsansvar

Behandlingsansvarlig institusjon

Norges teknisk-naturvitenskapelige universitet / Fakultet for informasjonsteknologi og elektroteknikk (IE) / Institutt for informasjonssikkerhet og kommunikasjonsteknologi

Prosjektansvarlig (vitenskapelig ansatt/veileder eller stipendiat)Jan William Johnsen, jan.w.johnsen@ntnu.no, tlf: **Skal behandlingsansvaret deles med andre institusjoner (felles behandlingsansvarlige)?**

Nei

Utvalg 1

Beskriv utvalget

Personer som har registrert domener kan omfattes av prosjektet.

Beskriv hvordan rekruttering eller trekking av utvalget skjer

Prosjektet vil bruke åpne datasett som bestemmer utvalget.

Alder

18 - 120

Inngår noen av disse gruppene i utvalget?

- Personer bosatt i land utenfor EU/EØS-området

Personopplysninger for utvalg 1

- E-postadresse, IP-adresse eller annen nettidifikator
- Gps eller andre lokaliseringsdata (elektroniske spor)

Hvordan samler du inn data fra utvalg 1?

Stordata (Bigdata)

Beskriv

Et datasett med informasjon om domener skal benyttes. Det kan bli aktuelt å innhente ytterligere informasjon dersom datasettet ikke allerede inkluderer nok til å benytte maskinlæringsmodellen. Innhenting av mer informasjon vil i så tilfelle gjøres gjennom åpne protokoller og vil ikke strekke seg forbi personopplysningene som er listet.

Grunnlag for å behandle alminnelige kategorier av personopplysninger

Allmenn interesse eller offentlig myndighet (Personvernforordningen art. 6 nr. 1 bokstav e)

Redegjør for valget av behandlingsgrunnlag

Alle som har registrert et domene og tilknyttet en IP adresse kan omfattes av prosjektet, alt etter hvilke domener som er inkludert i datasettet. Det vil være umulig i dette prosjektet å innhente samtykke fra alle potensielt berørte personer. Understreker at all informasjon som skal benyttes er/har vært offentlig tilgjengelig informasjon.

Informasjon for utvalg 1

Informerer du utvalget om behandlingen av personopplysningene?

Nei

Begrunn hvorfor du ikke informerer utvalget om behandlingen.

Et datasett bestemmer utvalget av domener som skal omfattes av prosjektet. Dette kan inkludere domener fra forskjellige land og uten videre kontaktinformasjon til den enkelte administrator av domenet. Å innhente samtykke fra alle som er inkludert i datasettet (flere hundretusen) vil i praksis være umulig. Videre er dette informasjon som er tilgjengelig, og som domeneiere har samtykket til å gi fra seg på et tidligere tidspunkt (ved registreringen av domenet).

Tredjepersoner

Skal du behandle personopplysninger om tredjepersoner?

Nei

Dokumentasjon

Hvordan kan de registrerte få innsyn, rettet eller slettet personopplysninger om seg selv?

Prosjektet ønsker å benytte informasjon som allerede er offentlig tilgjengelig. Personene som omfattes av prosjektet vil ikke nødvendigvis vite at deres domene og informasjon er inkludert i prosjektet. Samtidig er det viktig at prosjektet ivaretar personvernet på best mulig måte, og begrenser persondata til det minimale.

Totalt antall registrerte i prosjektet

100.000+

Tillatelser

Skal du innhente følgende godkjenninger eller tillatelser for prosjektet?

Ikke utfyllt

Behandling

Hvor behandles personopplysningene?

- Maskinvare tilhørende behandlingsansvarlig institusjon

Hvem behandler/har tilgang til personopplysningene?

- Student (studentprosjekt)
- Prosjektansvarlig

Tilgjengeliggjøres personopplysningene utenfor EU/EØS til en tredjestat eller internasjonal organisasjon?

Nei

Sikkerhet

Oppbevares personopplysningene atskilt fra øvrige data (koblingsnøkkel)?

Nei

Begrunn hvorfor personopplysningene oppbevares sammen med de øvrige opplysningene

Alle personopplysninger som skal benyttes er allerede tilstede i det aktuelle datasettet. Dette prosjektet vil derfor ikke bidra med nye personopplysninger gjennom innhenting fra individer. Videre er det fordelsmessig at data lagres sammen i et datasett da dette letter arbeid med prosessering i maskinlæringsmodellen. Risiko for å benytte personopplysningene vurderes likevel til å være liten da andre sikkerhetstiltak som adgangskontroll og kryptering ved sending og lagring vil benyttes.

Hvilke tekniske og fysiske tiltak sikrer personopplysningene?

- Opplysningene krypteres under lagring
- Opplysningene krypteres under forsendelse
- Adgangsbegrensning

Varighet

Prosjektperiode

25.01.2023 - 21.06.2023

Hva skjer med dataene ved prosjektslutt?

Data slettes (sletter rådataene)

Vil de registrerte kunne identifiseres (direkte eller indirekte) i oppgave/avhandling/øvrige publikasjoner fra prosjektet?

Nei

Tilleggsopplysninger

Prosjektet ønsker å benytte informasjon som er tilgjengelig om enkelte domener. Prosjektet ønsker videre å identifisere tilsvarende ondsinnede nettsider basert på tilgjengelig informasjon. Dataen som sammenfattes vil lagres kryptert i henhold til retningslinjer fra NTNU. Prosjektet er ikke interessert i enkeltpersoners opplysninger, men heller hvilke typer informasjon kan identifisere tilsvarende ulovlige nettsider. Likevel trenger prosjektet å bruke personopplysninger (IP adresser) for å innhente tilstrekkelig informasjon. For å redusere personvernulempen, vil prosjektet fjerne all personlig informasjon som oppdages, men som ikke er relevant for eksperimentet, samt fjerne informasjon som er tilgjengelig i datasettet, men som ikke skal benyttes i eksperimentet. Avvik skal håndteres i henhold til NTNUs retningslinjer for avvikshåndtering i behandling av personopplysninger. Sending og lagring av data skal også håndteres i henhold til overnevnte retningslinjer. All data skal slettes ved prosjektet slutt og ingen personlig data skal inkluderes i rapporten.

Se vedlagt liste med type informasjon som er tilgjengelig i datasettet. Dersom det er ønskelig med mer informasjon, finnes ytterligere beskrivelse av hvordan informasjonen er innhentet i beskrivelsen av datasettet (<https://doi.org/10.1016/j.dib.2020.106304>). For informasjon som skal innhentes i prosjektet henvises det til VirusTotals API dokumentasjon, hvor prosjektet ønsker å innhente Domain objekter (<https://developers.virustotal.com/reference/domains-1>), IP objekter (<https://developers.virustotal.com/reference/ip-object>) og URL objekter (<https://developers.virustotal.com/reference/url-object>). De tilgjengelige attributtene er i dokumentasjonen beskrevet. Det vektlegges at personlig informasjon gjennom WHOIS attributtene er anonymisert, slik at ingen ny personlig informasjon vil innhentes (<https://developers.virustotal.com/reference/whois>). En komplett beskrivelse av eksakt hvilke attributter som vil brukes i prosjektet er på nåværende tidspunkt vanskelig å gi. Likevel vil ikke prosjektet strekke seg utover attributtene som er tilgjengelig fra nevnte objekter. Videre vil heller ikke alle attributtene bli brukt.

Andre vedlegg[NSD_Information_in_existing_dataset.pdf](#)

Information present in existing dataset:

Attribute name	Attribute description	Interesting for this project
url	URL of the Webpage	yes
ip_addr	IP Address of the webpage	yes
geo_loc	Name of the country based on IP Address location	yes
url_len	Length of URL - count of characters in a URL	no
js_len	Length of JavaScript code in KB in the webpage	no
js_obf_len	Length of Obfuscated JavaScript (in KB) in the webpage	no
tld	Top Level Domain of the webpage	yes
who_is	Gives out whether the WHOIS information of the registered domain is complete or incomplete	yes
https	Gives out whether the website uses https or http protocol	yes
content	Raw web Content of the webpage. Includes filtered and processed text and JavaScript code	no
label	Classification label categorizing the webpage as malicious or benign	yes

[Meldeskjema](#) / [Webpage fingerprinting](#) / Vurdering

Vurdering av behandling av personopplysninger

Referansenummer

999641

Vurderingstype

Standard

Dato

22.02.2023

Prosjekttittel

Webpage fingerprinting

Behandlingsansvarlig institusjon

Norges teknisk-naturvitenskapelige universitet / Fakultet for informasjonsteknologi og elektroteknikk (IE) / Institutt for informasjonssikkerhet og kommunikasjonsteknologi

Prosjektansvarlig

Jan William Johnsen

Student

Martin Schiefloe Bakken

Prosjektperiode

25.01.2023 - 21.06.2023

Kategorier personopplysninger

Alminnelige

Lovlig grunnlag

Allmenn interesse eller offentlig myndighet (Personvernforordningen art. 6 nr. 1 bokstav e)

Behandlingen av personopplysningene er lovlig så fremt den gjennomføres som oppgitt i meldeskjemaet. Det lovlige grunnlaget gjelder til 21.06.2023.

[Meldeskjema](#) **Kommentar**

OM VURDERINGEN

Sikt har en avtale med institusjonen du forsker eller studerer ved. Denne avtalen innebærer at vi skal gi deg råd slik at behandlingen av personopplysninger i prosjektet ditt er lovlig etter personvernregelverket.

IKKE BEHOV FOR DPIA

Prosjektet behandler personopplysninger i stor skala med hensyn til utvalgsstørrelse på en slik måte at de registrerte hindres i å utøve sine rettigheter. Vanligvis krever dette en mer omfattende vurdering (DPIA). Vi mener det likevel ikke er høy risiko for personvernet og at prosjektet derfor ikke trenger en DPIA. Dette fordi det behandles få opplysninger, ingen særlige kategorier, varigheten på behandlingen er kort og data hentes utelukkende fra offentlige kilder.

ALLMENNHETENS INTERESSE

Behandlingen av personopplysninger er nødvendig for allmennhetens interesse (forskning), jf. personvernforordningen art. 6 nr. 1 e), jf. personopplysningsloven § 8. Prosjektet gjør nødvendige tiltak for å ivareta de registrertes rettigheter og friheter, jf. art. 89 nr. 1. I vår vurdering har vi lagt vekt på at formålet er å bruke opplysninger om infrastruktur til å identifisere ondsinnede nettsider, og at det å gjøre prosessen for å identifisere slike nettsider enklere å raskere har en høy samfunnsnytte. Prosjektet vil heller ikke behandle flere opplysninger om de registrerte i datamaterialet enn det som er nødvendig for å oppfylle dette formålet.

UNNTAK FRA RETTEN TIL INFORMASJON

De registrerte får ikke informasjon fordi det er uforholdsmessig vanskelig å skulle gi informasjon sett opp mot verdien deltagere vil ha av å motta denne, jf. personvernforordningen art. 14 nr. 5 b. Personopplysningene behandles til forskningsformål, og behandlingsansvarlig gjør egnede tiltak for å verne den registrertes rettigheter og friheter. I vår vurdering har vi lagt vekt på at det er svært mange registrerte, forsker har ikke kontaktinformasjon, opplysningene som behandles har en høy grad av forventet offentlighet og varigheten for behandlingen av personopplysningene er relativt kort.

FØLG DIN INSTITUSJONS RETNINGSLINJER

Vi har vurdert at du har lovlig grunnlag til å behandle personopplysningene, men husk at det er institusjonen du er ansatt/student ved som avgjør hvilke databehandlere du kan bruke og hvordan du må lagre og sikre data i ditt prosjekt. Husk å bruke leverandører som din institusjon har avtale med (f.eks. ved skylagring, nettspørreskjema, videosamtale el.

Personverntjenester legger til grunn at behandlingen oppfyller kravene i personvernforordningen om riktighet (art. 5.1 d), integritet og konfidensialitet (art. 5.1. f) og sikkerhet (art. 32).

MELD VESENTLIGE ENDRINGER

Dersom det skjer vesentlige endringer i behandlingen av personopplysninger, kan det være nødvendig å melde dette til oss ved å oppdatere meldeskjemaet. Se våre nettsider om hvilke endringer du må melde: <https://sikt.no/melde-endringer-i-meldeskjema>

OPPFØLGING AV PROSJEKTET

Vi vil følge opp ved planlagt avslutning for å avklare om behandlingen av personopplysningene er avsluttet. Lykke til med prosjektet!

Appendix **F**

Risk Assessment

This appendix presents the risk assessment conducted before data handling started. The risk assessment assesses relevant events that could occur before, during, and after data processing. As discussed in Chapter 7, a risk assessment was required before handling personal information in research projects at NTNU. The risk assessment first presents general information about the project, and then the risk assessment of unwanted events follows. Finally, the guide to performing the risk assessment is included.

Prosjektinformasjon

Prosjekt	Webpage fingerprinting
Institutt	Institutt for Informasjonssikkerhet og Kommunikasjonsteknologi, IIK
Fakultet	Fakultet for Informasjonsteknologi og Elektroteknikk, IE
Prosjektleder	Jan William Johnsen
NSD/REK-referanse	999641
Dato for risikovurdering	31.01.2023
Godkjent (dato og signatur prosjektleder)	
Virksomhetsområde	Forskning
Formål	Identifisere like ulovlige nettsider
Type personopplysninger	Alminnelige
Konfidensialitetsklasse	Intern
Varighet	25.01.2023-21.06.2023
Antall prosjektdeltakere	1

Gi en kortfattet beskrivelse av planlagt dataflyt i prosjektet:

Personlig informasjon (gule/interne) er tilgjengelig i datasettet som skal benyttes. Datasettet vil lastes ned direkte til NTNU SkyHigh server og aksesseres kun gjennom SSH tilkobling. Adgangen vil være begrenset til studenten. All databehandling vil foregå på nevnte server. Datasettet slettes etter prosjektet er ferdig.

Nr.	Fase i dataflyt	Utsætt hendelse/situasjon/sårbarhet (risikoenhet)	Konsekvens	Eksisterende beskyttelses tiltak	Risiko nå		Tiltak som reduserer risiko (sannsynlighet og/eller konsekvens)	Risiko nå etter tiltak		Ansvarlig for tiltak	Prist for gjennomføring	Kommentar	Tiltak gjennomført
					S	K		S	K				
1	Planlegging	Hvordan medfølge en sikker for personopplytningene i dette prosjektet? Beskriv risikofaktorer og situasjoner som kan oppstå i løpet av prosjektperioden.	Har det de mulige konsekvensene? Hvis det ikke skal være på negative konsekvenser for de involverte. Husk at konsekvenser ligger med i risikoen (som regel er større for særlige kategorier (ansatte) enn for administrative personopplytninger (se veiledningsplan)).	Erne det allerede tatt med NTHU/I prosjektet som kan bidra til å hindre at det skjer?	Sannsynlighet og konsekvens på en skala fra 1 til 4 (se veiledningsplan)	Erne det allerede tatt med NTHU/I prosjektet som kan bidra til å hindre at det skjer?	Sannsynlighet og konsekvens på en skala fra 1 til 4 (se veiledningsplan)	Erne det allerede tatt med NTHU/I prosjektet som kan bidra til å hindre at det skjer?	Sannsynlighet og konsekvens på en skala fra 1 til 4 (se veiledningsplan)	Erne det allerede tatt med NTHU/I prosjektet som kan bidra til å hindre at det skjer?	Erne det allerede tatt med NTHU/I prosjektet som kan bidra til å hindre at det skjer?	Erne det allerede tatt med NTHU/I prosjektet som kan bidra til å hindre at det skjer?	Erne det allerede tatt med NTHU/I prosjektet som kan bidra til å hindre at det skjer?
2	Lagring	Datasett lagres på minneopplag for å hindre at datasett oppbevares i andre PC-er.	Datasett med sammenhengende personopplytninger (IP-adresser og URL) kan finnes av andre. Datasettet inneholder personlige personopplytninger som ikke vil være direkte belastende for domstolene.	NTHUs retningstiltak for behandling av personopplytninger og informasjonssikkerhet. NTHUs godkjente tjenester og verktøy for behandling av personopplytninger.	1	2	2	NTHUs retningstiltak for behandling av personopplytninger skal gjennomføres i prosjektet. Tiltakene for behandling av personopplytninger starter.	1	2	2	Prosjektleder	Er behandling av personopplytninger startet?
4	Lagring	Datasett lagres på sikret PC.	Datasett med tilhørende personopplytninger kan ikke behandles til andre utvalgte tjenester som igjen kan eksponere data.	NTHUs retningstiltak for behandling av personopplytninger og informasjonssikkerhet. NTHUs godkjente tjenester og verktøy for behandling av personopplytninger.	1	2	2	NTHUs retningstiltak for behandling av personopplytninger skal gjennomføres i prosjektet. Tiltakene for behandling og lagring av personopplytninger starter.	1	2	2	Prosjektleder og prosjektleder	Lagring må foregå på sikret PC, dette skal ikke på lagring av datasett skal ikke.
6	Behandling/analyse	Erne med persondata ligger åpne på PC og tilgjengelige for uvedkommende.	Uvedkommende får rynn i de aktuelle persondataene.		2	2	4	PC skal alltid låses når den ikke er i bruk. Når persondata behandles skal det sikret jobbes for at uvedkommende ikke har tilgang til.	1	2	2	Prosjektleder	Tiltak iverksettes umiddelbart og opprettholdes daglig.
7	Behandling/analyse	It-skuffer av personopplytninger kommer på avveie.	Uvedkommende får tilgang til utskrift med personopplytninger.		1	2	2	Datasett med tilhørende personopplytninger skal ikke skrives ut.	1	2	2	Prosjektleder og prosjektleder	Umsiddelbart
8	Behandling/analyse	Personopplytninger lagres med fra sikret lagringssystem til usikret privat PC.	Datasett med tilhørende personopplytninger kan ikke behandles til andre utvalgte tjenester som igjen kan eksponere data.	NTHUs retningstiltak for behandling av personopplytninger og informasjonssikkerhet. NTHUs godkjente tjenester og verktøy for behandling av personopplytninger.	2	2	4	NTHUs retningstiltak for behandling av personopplytninger skal gjennomføres i prosjektet. Tiltakene for behandling av personopplytninger starter.	1	2	2	Prosjektleder og prosjektleder	Er behandling av personopplytninger startet?
9	Deling	Vel 3 g uvedkommende tilgjengelig til informasjon kan personopplytninger komme på avveie.	Vel 3 g uvedkommende tilgjengelig til informasjon kan personopplytninger komme på avveie.	NTHUs retningstiltak for behandling av personopplytninger og informasjonssikkerhet. NTHUs godkjente tjenester og verktøy for behandling av personopplytninger.	1	2	2	Et deling av informasjon mellom prosjektleder (veileder) og prosjektleder (studenter) skal dobbelt sjekkes for sikkerhet slik at ingen tilfeller går til andre.	1	2	2	Prosjektleder og prosjektleder	Tiltak som opprettholdes gjennom hele prosjektet.
10	Avslutning	Persondata blir liggende på lagringssystem da etter at prosjektet er avsluttet.	Uvedkommende kan på et senere tidspunkt få tilgang til persondataen.	NTHUs retningstiltak for behandling av personopplytninger og informasjonssikkerhet. NTHUs godkjente tjenester og verktøy for behandling av personopplytninger.	2	2	4	Både veileder og studenten ettersøke sletting av data ved prosjektets slutt.	1	2	2	Prosjektleder og prosjektleder	01.07.23

Denne malen er utviklet med utgangspunkt i Uninetts veileder:
Risikovurdering av informasjonssikkerhet med tilhørende mal. For nærmere beskrivelse og veiledning fra Uninett, se https://www.uninett.no/sites/default/files/imce/veileder_risikovurdering.pdf

Framgangsmåte

Beskriv den planlagte dataflyten i prosjektet. Gå gjennom og beskriv hendelser og situasjoner som kan føre til at personopplysninger kommer på avveie, går tapt eller blir utilgjengelige for de som skal ha tilgang i alle faser av datahåndteringen (innsamling, overføring, lagring, behandling/analyse, eventuell deling, avslutning). Aktuelle tema kan være hendelser knyttet til overføring av data, utskrift, tilgangskontroll og kontroll på fysisk utstyr. Det er allerede fylt inn noen eksempler på hva som kan gå galt, tilpass disse til prosjektet og legg til nye.

Hovedfokus skal være på mulige konsekvenser for de registrerte (tap av anseelse/integritet dersom opplysninger som oppleves som følsomme eller som kan misbrukes, kommer på avveie), men konsekvensene for institusjonen (økonomisk tap, økonomiske sanksjoner, tap av omdømme) skal også tas med i betraktning. Husk at konsekvensen (og dermed risikoen) som regel er større for særlige kategorier ("sensitive") enn for alminnelige personopplysninger.

Gå deretter gjennom hvilke eventuelle tiltak som allerede eksisterer, og hvilke tiltak som kan settes inn for å redusere risikoen (sannsynlighet og/eller konsekvens) ytterligere.

I mange tilfeller vil konsekvensen av brudd på personopplysningssikkerheten ikke la seg redusere. Det vil likevel som regel gå an å redusere sannsynligheten.

I noen tilfeller vil det ikke la seg gjøre å få sannsynligheten ned til grønt. Noe restrisiko må som regel aksepteres, og det er opp til hvert enkelt prosjekt (eventuelt institutt, dersom restrisikoen er høy) å avgjøre hvor mye risiko prosjektet kan håndtere.

[Se også wikien Behandle personopplysninger i student- og forskningsprosjekt.](#)

Begreper

Personopplysninger

Personopplysninger: Opplysninger og vurderinger som direkte eller indirekte kan knyttes til en enkeltperson. Eksempler på personopplysninger er navn, fødselsnummer, bilde (dersom personer kan gjenkjennes), video og lydopptak, logg fra bruk av adgangskort, informasjon fra en kilde, IP-adresser, blodprøver og googlesøk på person.

Særlige kategorier

Særlige kategorier av personopplysninger ("sensitive") krever strengere beskyttelse og omfatter blant annet informasjon om rase, etnisitet, seksualitet, politisk overbevisning, helseopplysninger m.m.

Registrerte

De personene opplysningene gjelder, for eksempel informanter i et forskningsprosjekt.

Risiko

Risiko forstås i denne sammenhengen som et produkt av sannsynlighet ganget med konsekvens.

Risikonivå

I disse kolonnene noteres tallverdier for sannsynlighet og konsekvens/skade. Dette gjøres for hver enkelt uønsket hendelse som er notert i regnearket. Sannsynligheten varierer fra svært lite sannsynlig (tallverdien 1) til svært sannsynlig (tallverdien 4). Konsekvensen/skaden varierer fra lite alvorlig (tallverdien 1) til svært alvorlig (tallverdien 4).

Konsekvens

- | | |
|----------|---|
| 1 | Lite alvorlig. Har ubetydelige skadevirkninger for enkeltpersoner eller institusjonen. |
| 2 | Mindre alvorlig. Har visse skadevirkninger for enkeltpersoner eller institusjonen. Eksempel: Uautorisert eksponering av noen få alminnelige personopplysninger |
| 3 | Alvorlig. Har merkbare skadevirkninger for enkeltpersoner eller institusjonen. Eksempel: Uautorisert eksponering av fortrolige/sensitive eller større mengder alminnelige personopplysninger. |
| 4 | Svært alvorlig. Har store skadevirkninger for enkeltpersoner eller institusjonen. Eksempel: Uautorisert eksponering av større mengder sensitive personopplysninger. |

Sannsynlighet*

- | | |
|----------|---|
| 1 | Svært lite sannsynlig - vil mest sannsynlig ikke skje i løpet av prosjektperioden |
| 2 | Lite sannsynlig - kan forekomme i løpet av prosjektperioden. |
| 3 | Sannsynlig - vil sannsynligvis skje i løpet av prosjektperioden. |
| 4 | Svært sannsynlig - kan potensielt skje flere ganger i løpet av prosjektperioden. |

Risikoverdi (rød/gul/grønn)

- | | |
|--------------------------------|--|
| Rød (risikoverdi 8-16) | Hendelser med høy risiko. Nye tiltak skal innføres. |
| Gul (risikoverdi 4-7) | Hendelser med moderat risiko. Nye tiltak bør vurderes. |
| Grønn (risikoverdi 1-3) | Hendelser med lav risiko. |

Tiltak

List opp aktuelle organisatoriske, menneskelige og teknologiske tiltak som kan redusere sannsynlighet og/eller konsekvens.
Eksempler på organisatoriske tiltak: Utforming av retningslinjer, avviksrutiner, organisering av tilganger, internkontroll etc.
Eksempler på menneskelige tiltak: Opplæring, bevisstgjøring av brukere, endring av praksis etc.
Eksempler på teknologiske tiltak: Tofaktorautentisering, kryptering, tilgangskontroll, avidentifisering etc.
(innebærer ofte valg av NTNU-godkjente løsninger for innsamling, overføring og lagring)

- * Med tanke på sannsynlighet, må også prosjektets varighet tas med i betraktning. Er det snakk om et langvarig prosjekt, kan det for eksempel heller være aktuelt å bruke år som målestokk framfor hele prosjektperioden.

Appendix **G**

Data Management Plan

As discussed in Chapter 7, Sikt provides a template for data management to help researchers obtain a clear plan for data handling. The template enables researchers to consider how data is collected, stored, and processed throughout the project. The data management plan also provides a general description of the project and the data that will be used.

Webpage fingerprinting

Master prosjekt ved NTNU, Institutt for informasjonssikkerhet og kommunikasjonsteknologi. Intensjonen med prosjektet er å utforske muligheter for å identifisere ulovlige nettsider basert på infrastruktur opplysninger.

Fagområder

Teknologi

Forskningsansvarlig institusjon

Norges teknisk-naturvitenskapelige universitet / Fakultet for informasjonsteknologi og elektroteknikk (IE) / Institutt for informasjonssikkerhet og kommunikasjonsteknologi

Prosjektvarighet

25.01.2023 — 21.06.2023

Formål

Prosjektet tar sikte på å identifisere like, ulovlige nettsider basert på informasjon knyttet til deres infrastruktur. Dette gjøres med følgende foreløpige forskningsspørsmål: Hvilke infrastruktur attributter kan skille en ondsinnet nettside fra en harmløs? Hvilken maskinlæringsmodell er best egnet til å identifisere ulovlige nettsider uten testdata? I hvilken grad kan clustering av nettsider identifisere ulovlige nettsider, og kan det skille på ulike typer ulovlige nettsider?

Nytteverdi

Prosjektet er et forskningsprosjekt som kan lette arbeid med å motarbeide ulovlige nettsider.

Etiske retningslinjer

- Generelle forskningsetiske retningslinjer
- Forskning på Internett
- Naturvitenskap og teknologi

Relaterte ressurser

- Potensielt datasett [https://www.data-in-brief.com/article/S2352-3409\(20\)31198-7/fulltext#secsec002a](https://www.data-in-brief.com/article/S2352-3409(20)31198-7/fulltext#secsec002a)

Datasett

Beskrivelse

Datasett med informasjon om domener. Vil brukes til å identifisere like, ulovlige nettsider basert på infrastruktur opplysninger

Datatype

Datasett

Språk

Engelsk

Nøkkelord

informasjonssikkerhet

Data om personer

Ja

Er det noen andre grunner til at dataene dine trenger ekstra beskyttelse?

Nei

Kategorier av personopplysninger

Alminnelige

Utvalgets størrelse

150000

Konfidensialitetsklassifisering

Intern

Kommentar

Personopplysningene som er i datasettet er IP adresser og lokasjonsdata (land og område, ikke nøyaktig plassering)

Innsamlingsperiode

01.03.2023 — 01.04.2023

Innsamlingsenheter

- 6. Annen innsamlingsmåte

Metode

Annet

Beskrivelse

Program leser inn IP address/URL fra datasettet og foretar spørringer mot VirusTotal API for å hente ytterligere informasjon om domenet. Dette skrives så til nytt datasett. Datasettene leses og skrives fra

NTNU server

Størrelse

1000 MB

Format

csv

Programvare

Python

Lagring

- 05. NTNU Office 365 (SharePoint, Teams, Onedrive)
- 08. Tjenester for sensitive data (TSD)

Overføring

- 2. Office 365 (SharePoint, Teams, Onedrive)

Kommentar

Utforsker per nå muligheter for å bruke SkyHigh NTNU server for lagring av datasett

Arkivering

Nei

Kommentar

Plan per nå er å slette all data ved prosjektets slutt. En helhetlig forskningsetisk vurdering skal gjøres gjennom prosjektet hvorvidt datasett skal anonymiseres og gjøres tilgjengelig for videre forskning i henhold til god forskningsetikk

