CZECH TECHNICAL UNIVERSITY IN PRAGUE
FACULTY OF INFORMATION TECHNOLOGY

ASSIGNMENT OF MASTER'S THESIS

Title:	ManaTI: Web Assistance for the threat Analysis supported by Domain Similarity
Student:	Raúl Carmelo Benítez Netto
Supervisor:	Ing. Sebastián García, Ph.D.
Study Programme:	Informatics
Study Branch:	Computer Security
Department:	Department of Computer Systems
Validity:	Until the end of winter semester 2018/19

Instructions

Contact CISCO to agree on the research goals and on their specific needs of a web system to assist the threat analysis.
Design and develop a web application to assist CISCO researchers, named ManaTI, that will provide web logs detection, visualization, storage. Develop modules for ManaTI to analyze the web logs in the background. The modular system should be designed to allow other modules to be added by CISCO. Study the WHOIS protocol. Research about the possible differences between malicious and normal domains in the registration of WHOIS data.
Develop an algorithm to find similarities in WHOIS registration data as an external module for ManaTI. Apply machine learning to find related domains, best distance measures and similarities in WHOIS registrations.
Evaluate if the web application helps the analysts with their job of labelling weblogs faster and easier.
Evaluate if the algorithm correctly detects related domains using the WHOIS information of a domain given by the analyst.

References

[1] Jane Webster, Jaspreet S. Ahuja Enhancing the Design of Web Navigation Systems: The Influence of User Disorientation on Engagement and Performance. MIS Quarterly 2006.
[2] Masahiro Kuyama, Yoshio Kakizaki and Ryoichi Sasaki. Method for Detecting a Malicious Domain by using WHOIS and DNS features. DigitalSec 2016.
[3] Yi, Mun Y., Hwang, Yujong. Predicting the use of web-based information systems: Self-efficacy, enjoyment, learning goal orientation, and the technology acceptance model. International Journal of Human Computer Studies 2003.
[4] Vojtech Letal. Discovering of malicious domains using WHOIS database. CTU FEL Master's Thesis 2015.
[5] Kalyan Veeramachaneni, Ignacio Arnaldo, Vamsi Korrapati Constantinos Bassias, Ke Li. AI2: Training a big data machine to defend. IEEE 2016.

prof. Ing. Róbert Lórencz, CSc.
Head of Department

prof. Ing. Pavel Tvrdík, CSc.
Dean

Prague February 20, 2017

Czech Technical University in Prague
Faculty of Information Technology
Department of Theoretical Computer Science

Master's thesis

ManaTI: Web Assistance for the Threat Analyst, supported by Domain Similarity

Ing. Raúl Carmelo Benítez Netto

Supervisor: Ing. Sebastían García, Ph.D.

9th May 2017

Acknowledgements

I would like to thank my advisor Sebastían García, Ph.D for his inexhaustible patience and willingness towards to my work, to all the staff of FEE's Department of Computer Science who in one way or another collaborated with my work, and also to the company Cisco Systems Inc. for giving me the opportunity to work with their great team of researchers.

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis.

I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended. In accordance with Article 46(6) of the Act, I hereby grant a nonexclusive authorization (license) to utilize this thesis, including any and all computer programs incorporated therein or attached thereto and all corresponding documentation (hereinafter collectively referred to as the "Work"), to any and all persons that wish to utilize the Work. Such persons are entitled to use the Work in any way (including for-profit purposes) that does not detract from its value. This authorization is not limited in terms of time, location and quantity. However, all persons that makes use of the above license shall be obliged to grant a license at least in the same scope as defined above with respect to each and every work that is created (wholly or in part) based on the Work, by modifying the Work, by combining the Work with another work, by including the Work in a collection of works or by adapting the Work (including translation), and at the same time make available the source code of such work at least in a way and scope that are comparable to the way and scope in which the source code of the Work is made available.

In Prague on 9th May 2017 .....

Czech Technical University in Prague
Faculty of Information Technology

Citation of this thesis

Benítez Netto, Raúl Carmelo. *ManaTI: Web Assistance for the Threat Analyst, supported by Domain Similarity.* Master's thesis. Czech Technical University in Prague, Faculty of Information Technology, 2017.

Abstrakt

Nárůst výskytu malware a jeho různorodosti vede k vývoji nových metod jeho detekce. Jedním z problémů, kterým musí bezpečností analytici čelit je velké množství dat z provozu na síti, které je nutné manuálně kontrolovat. Zvětšující se množství dat vede k tomu, že tato ruční analýza je zdlouhavá a též k nárůstu nepřesností. Tento problém nastává i když firmy povolují uživatelům pouze použití protokolů HTTP a HTTPS, které často stačí k jejich práci. Tvůrci malware se ale adaptují na tento trend a malware analytici zaznamenali posun komunikace malware právě ke zmíněným protokolům. Základní jednotka při analýze provozu na síti se nazývá weblog. Přestože Analýza weblogů může být použita k detekování útoků, tento proces je velice komplikovaný a vyžaduje mnoho znalostí. Mezi ně patří například analýza vzorců chování nebo využití informací ze služby WHOIS. V obou zmíněných technikách je velmi důležitý lidský faktor. Protože zajištění přesnosti při ručním zpracování milionů zaznamů je zvášť obtížný úkol, bezpečnostní analytici potřebují nástroj, který tento proces usnadní a urychlí. Projekt ManaTI vznikl jako asistenční nástroj pro bezpečností experty. ManaTI využívá algoritmů strojového učení k urychlení a zpřesnění procesu detekce hrozeb. Projekt má dva hlavní cíle: Za prvé: vytvořit webové rozhraní napomáhající analytikům s analýzou weblogů a hledáním doplňujících informací o jednotlivých domén. Za druhé: využití strojového učení k vyhledávání a určování podobností v informacích ze služby WHOIS pro jednotlivé domény. Tato metoda pracuje na základě vzdáleností mezi vektory získanými z WHOIS informací jednotlivých domén. Mezi dosažené výsledky patří: Za prvé: použití nástroje ManaTI zvyšuje rychlost analitika při analýze domény v průměru 3,4 krát. Za druhé: potvrzení hypotézy, že příbuzné domény mají měřitelné podobnosti v informacích služby WHOIS a je tedy možné tyto informace použít k přesné klasifikaci. Za třetí: experimenty ukazují, že některé z informací ve WHOIS mají na přesnost klasifikace zásadní vliv, zatímco jiné ji téměř neovlivňují. Přesnost metody ve vyhledávání příbuzných domén za pomocí WHOIS je přibližně 98%.

Klíčová slova Malware, WHOIS, Domény, Strojové učení, Software, Kybernetická bezpečnost

Abstract

The increasing diversity and amount of malware traffic is pushing researchers to find better detection methods. When security practitioners analyze such large amount of traffic, they are usually overwhelmed and, therefore they analyze each time less traffic with less accuracy. This overwhelming problem data happens even when companies filter out part of their outgoing traffic. Given that users inside a company mostly need web services to work, it is usual only allow web traffic is going out of the enterprise. However, malware is aware of this filtering, and in the last years, we have witnessed a shift in malware towards using web services for their connections. For analyzing HTTP/S traffic, the default unit of analysis is called a weblog, from a log for the web traffic. These weblogs are used to find threats in the network, but a significant amount of expertise is needed for doing so. The required knowledge ranges from looking for domains which have been reported as malicious, to analyzing the patterns in the URLs and using the WHOIS information of the domains. These techniques highly depend on humans. All in all, analyzing millions of weblogs with speed and accuracy, balancing the amount of information and finding threats is at least a daunting task. Security analysts need a tool to help them organize their work and a machine learning algorithm that can improve the detection and speed up the analysis. It is in this context that we researched and created a new tool to assist the network security analysts to find threats: the ManaTI project. This project has two primary goals: First, to help analysts by means of a web interface, in evaluating the weblogs to better find and process the information. Second, to create a machine learning method that can identify domains which share some similarity in their WHOIS Information. Our algorithm can work as a WHOIS classification of similar domains also called WHOIS similarity distance. The conclusions of our research are: First, ManaTI can increase the speed of the security analysts by a factor of 3.4. Second, the WHOIS information of related domains has quantifiable similarities that make possible an accurate comparison. Third,

there are WHOIS fields which are more important for relating domains than others. Finally, the accuracy of finding related domains using a linear model classifier based on the WHOIS Similarity Distance algorithm is around 98%.

Keywords Malware, WHOIS, Domains, Machine Learning, Software, Cybersecurity

Contents

List of Figures

List of Tables

Introduction

Cyber-attacks are much more severe threat nowadays than several years ago [1]. Not only the number of devices increased exponentially, but also the amount of attackers increased, and their motivations changed substantially. Any device connected to the Internet may be vulnerable, and they are likely to be used to perform attacks to others computers around the world without authorization.

Malware, short for *Malicious Software*, refers to software designed to perform attacks or do unauthorized actions on a computer [2]. The purposes of malware vary, but commonly they include to steal sensitive information and to attack other computers. The information stolen can range from credit cards, bank information, to personal data. Fortunately, there is plenty of people working in cyber-security that deals with this issue through the analyses of network traffic [3].

The efforts to prevent these situations are extensive. However, the amount of new malware created every day increases continuously. The Figure 1.1 illustrates a statistic graph of the numbers of malware in the range between March 2015 and February 2017 according to an estimation of "The AV-TEST Institute". It registers over 390,000 new malicious programs every day [4].

Despite the massive efforts of AntiVirus companies, their work to stop malware may not be enough. Detecting, understanding and providing solutions for new malware are hard tasks. Security researchers are trying to understand how malware's families work and how is the communication process between malware and their controllers. Typically, modern malware needs to communicate with their controllers to receive orders and receive information. The channels used for communication are called Command and Control (C&C) channels [5].

Among the most used methods to find new threats in a network is the analysis of HTTP traffic. Since most organizations only allow this protocol to reach the Internet, most internal attacks and malware use it for communication. The default unit of analysis is usually called weblog, from a log for the

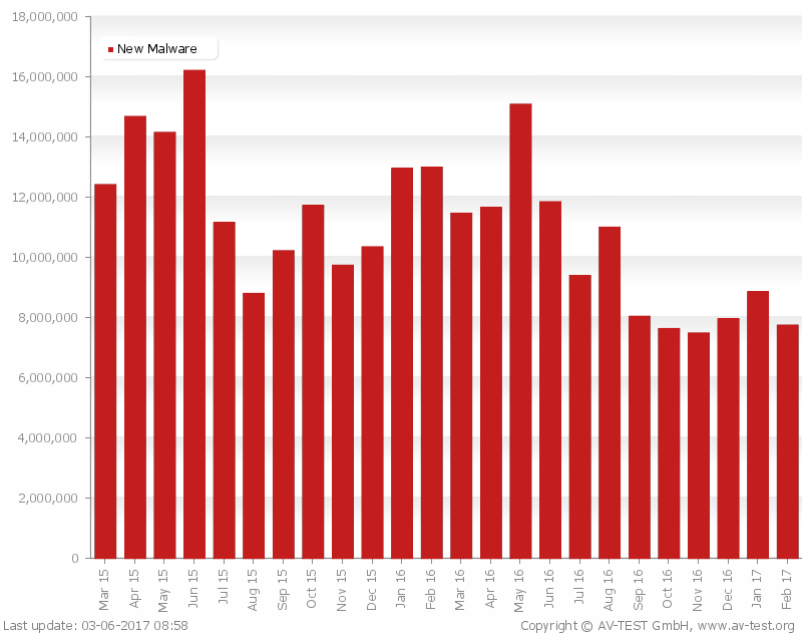


Figure 1.1: It shows the new malware created in the range between March 2015 and February 2017. X-axis shows the numbers of malware created and y-axis, the month when they were created.

web traffic. Security analysts use weblogs to detect threats[6].

Analyzing millions of weblogs with speed and accuracy, balancing the amount of information and finding threats is a daunting task. Security analysts need a tool to help them organize their work and a machine learning algorithm that can improve the detection and speed up the analysis. For this reason was developed ManaTI, to assist the analysts, organize their work and provide tools to enhance their analysis.

One of the characteristics of the Internet is that to obtain a new domain; it has to be bought from certain registered entities. These entities request some information about the buyer in order to create the domain. This information is called WHOIS information (or WHOIS data). Some researchers have already tried to detect malicious domains and differentiate them of legitimate domains using the WHOIS information. One of the goals of this thesis is to propose a Machine Learning method to find a relationship between domains used for the same purpose by using their WHOIS data. Our method is designed to work regardless of whether the domains are malicious or not. The result of our method is both, a distance measure between domains and a classification algorithm of similar domains.

Machine Learning techniques have been used to detect malicious domains and predict malware behavior successfully before[7]. However, in most cases,

the errors produced by Machine Learning algorithms are still too large to be acceptable

Frequently, the task of the security researchers that work finding threats involves complex expert knowledge. This experience ranges from searching domains which have being reported as malicious to analyzing the patterns in the URLs and using the WHOIS information of the domains. Although these techniques may work for the average analyst, they highly depend on humans generating the reputation rules and on the malware being analyzed. This thesis focuses on the analysis of weblogs.

It is in this context, Cisco and Faculty of Electrical Engineering of the Czech Technical University in Prague collaborated to create the ManaTI project in the Stratosphere Lab [8]. This thesis is one part of this collaboration where I researched, created and published a new tool to help the network security analysts to find threats in the network called ManaTI. This thesis has two main goals:

- To create a web application called ManaTI to assist the analysts in evaluating the web traffic to find better and process the information.

- To research a machine learning method that can confidently identify domains which WHOIS information is related. The idea of our algorithm is to work as a WHOIS classification of similar domains or as a WHOIS Similarity Distance.

As part of the evaluation of ManaTI and the distance measure, through experiments were conducted. The WHOIS Similarity Distance was evaluated to know its performance and ManaTI was assessed to know how much it helps the analysts.

This thesis is organized as follows: Section introduces the Background information about the topics discussed. The section describes state of the art. Then there are two chapters to discuss the two most important parts of this thesis: Chapter 2 describes the WHOIS similarity measure and the classification algorithm, and Chapter 3 describes the web application ManaTI. Finally, Section 3.4.1 presents the conclusion of the thesis.

Background

This section introduces relevant topics to the correct understanding of the methods proposed and the necessary to discern the problem correctly.

The main subjects of this thesis are network security, malware traffic analysis, assisting the threat analyst and the use of WHOIS information for finding similar domains. Also, it is necessary to understand the concepts of weblogs and why ManaTI uses them.

Cyber security

Computer security, also known as cyber security or IT security, has increased in the last years. The problem of data security has taken the attention of the media and governments, nowadays the work in the area is made of more importance. Computer security is not only focused on protecting the machines physically, but also it must take care of the most critical aspect of computers, the data [9]. Cyber security includes controlling physical access to the hardware, as well as protecting against the harm that may come via network access, data, and code injection [9].

Malware Short for malicious software, *malware* refers to software programs designed to damage or do other unauthorized actions on a computer system [2]. They are used to steal potentially sensible data, like numbers of credit cards, password of web-banking, until getting full access to your device. They could be a laptop, computer, tablet, mobile or others.

Malware is an umbrella term used to refer to a variety of forms of hostile or intrusive software, including computer viruses, worms, Trojan horses, ransomware, spyware, adware, scareware, and other malicious programs [10].

Ransomware One the most widely extended and dangerous malware are the *ransomware*, it affects an infected computer encrypting data, databases, or removing some files from the computer.Then it demands payment to reverse the damage.

Weblogs files The weblog is a composition of "web" and "log", basically means that is a register of an HTTP traffic. Weblogs is a log of HTTP requests and responses occurred during a period in the network traffic of a device. In simple words is a log file of everything sent or received using the HTTP protocol (in the Web)[11]. The following table illustrates an example of an HTTP

	POST /cgi-bin/process.cgi HTTP/1.1
	User-Agent: Mozilla/4.0 (compatible; MSIE5.01; Windows NT)
	Host: `www.tutorialspoint.com`
	Content-Type: application/x-www-form-urlencoded
GET request:	Content-Length: length
	Accept-Language: en-us
	Accept-Encoding: gzip, deflate
	Connection: Keep-Alive
	licenseID=string&content=string&/paramsXML=string

Then the HTTP traffic (request or response) is converted by special tools to a weblog, something similar as the Figure **??**[1]:

[1]The weblog example only has the most common and important fields of the most commons structures

212.555548 192.168.1.115 49161 23.21.92.252 80 POST i-50.b-000.xyz.bench.utorrent.com /e?i=50 text/json

Figure 1.2: Example of HTTP traffic.

ManaTI supports weblogs files produced by Bro project[2]. Bro is a tool for network security monitoring. Bro has many functions, but one of them can extract information of PCAP (Packet CAPture) files. And it can provide a http.log file where has all the HTTP connection in the PCAP selected; http.log is an example of weblogs file. Bro files were chosen to be compatible with ManaTI for several reasons:

- It has good documentation,
- Its use is widely extended in the network security monitoring area,
- Its weblogs format is easy to understand,
- The malware traffic captures provided by Stratosphere Lab use the Bro format for their weblogs [12].

Also, ManaTI provides support to the Cisco weblogs file format. It is a type of format only used by the researchers working in Cisco.

Cisco's research with weblogs The Cognitive Threat Analytics (CTA) group inside Cisco System, is in charge of analyzing weblogs in order to detect attacks and infections. They have software and methods that can analyze weblogs and identify possible malicious behaviors. The weblogs used are stored in a particular format used by Cisco. Their software and algorithms can process the weblogs to find anomalies, infections and attacks. However, the verification of the errors from the algorithms must be made by hand by the threat security analysts.

The most important tasks of the threat analysts is to study the traffic of one client device in a short period in order to find if it is infected or not. The traffic analyzed is represented in their custom weblog format. Typically the weblog files has between 2,000 and 10,000 lines. Such a large amount of weblogs per device makes the task of finding new threats and attacks very time consuming and prone to errors.

The analysis of weblogs requires advanced skills to understand the flow and communication state of the weblogs. After this analysis, a verdict should be assigned to each weblog. The verdicts can be:

- Legitimate: it is used to label weblogs that the analyst knows normal and therefore they are not related to any attack or infection,

[2]www.bro.org

- Malicious: it is used to label weblogs that are related to malware actions or somehow used by malware,
- Suspicious: it is used to label weblogs that the analyst is not sure about. The weblog may show some connections with malware or attacks but it can be also be highly connected to normal domains,
- False Positive: it is used on weblogs that were labeled as malicious before, but now the analyst is sure this was an error. However, it is important to remember the error to further improve the algorithms.

ManaTI was specially created for the task of assisting the analysts to study weblogs faster and more efficiently.

DNS

The Domain Name System (DNS) is a hierarchical decentralized naming system for computers, devices, services and other resources connected to private networks or the Internet and it has been in use since 1985. In essential the work of the DNS is translates more readily memorized domain names to the numerical IP addresses. For locating and identifying services and devices inside a network protocol. By providing a worldwide, distributed directory service, the Domain Name System is an essential component of the functionality of the Internet [13] [14]. In Figure 1.2 illustrates the workflow of a request for a website.

Domain name It is "the part of a network address which identifies it as belonging to a particular domain" [15]. Also, the domain name is an identification string that determines a field of administrative autonomy, authority or control within the Internet.

Domains are organized in levels; the first level is for top-level domains (TLDs), including the generic top-level domains (gTLDs), such as the prominent domains .com, .info, .net, .edu, and .org, and the country code top-level domains (ccTLDs).

The levels belong of these in the DNS hierarchy are typically open for reservation by end-users who can connect local area networks to the Internet. In Figure 1.3 the reader could see examples of how the DNS hierarchy works [16].

In the rest of the book when the author refer *domain name* is specially referring to the domains names in the TLDs and second level domain.

Malicious and legitimate domains Legitimate or normal domains are domains which belong to real entities or for a true purpose. Entities could be private or governmental companies, regular people, organizations, projects, marketing or educational use, etc. However, malicious domains are domains

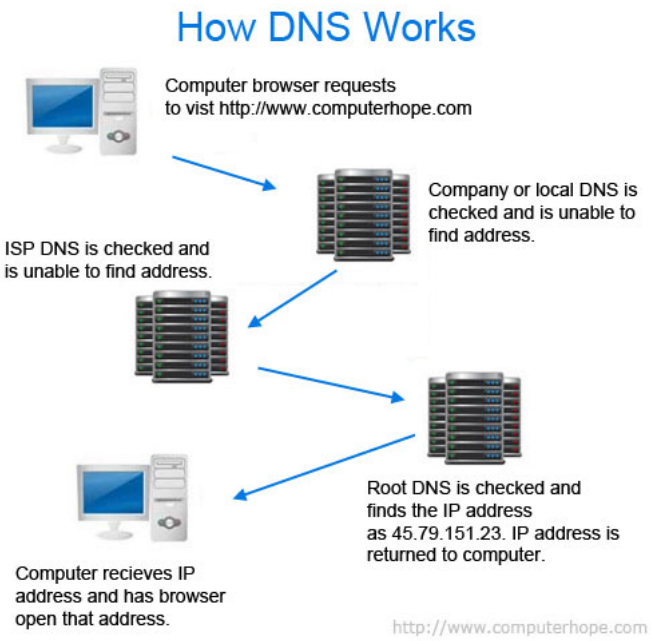


Figure 1.3: DNS basic workflow.

which are used to propagate malware, belong to bot or ransomware networks or created with a malicious purpose like phishing or spam campaign.

WHOIS Protocol

WHOIS is a TCP-Based transaction oriented query and response protocol is typically used to provide information about Internet services to the users [17].

WHOIS information When an entity wants to register a web domain, encompasses normal users, organization, governments, companies and others. They need to provide some contact information, like for examples: name of the entity, phone number, contacts' emails, name, address, and more details to identify the owner of the domain. This information is called *WHOIS information* or *WHOIS data* [18].

Although the WHOIS service is not a single, centrally-operated database, there are many companies in charge of registering the domains and hence, to ask the owner information, these companies are called *registrars* and *registries*. Any entity that wants to become a registrar must have the approved and accreditation of Internet Corporation for Assigned Names and Numbers (ICANN) [18].

There is not a standard structure of the WHOIS information. And it is not mandatory for the registrars fill in all fields. Often they do not use the same name of fields for the same information, e.g. Some registrars to specify the

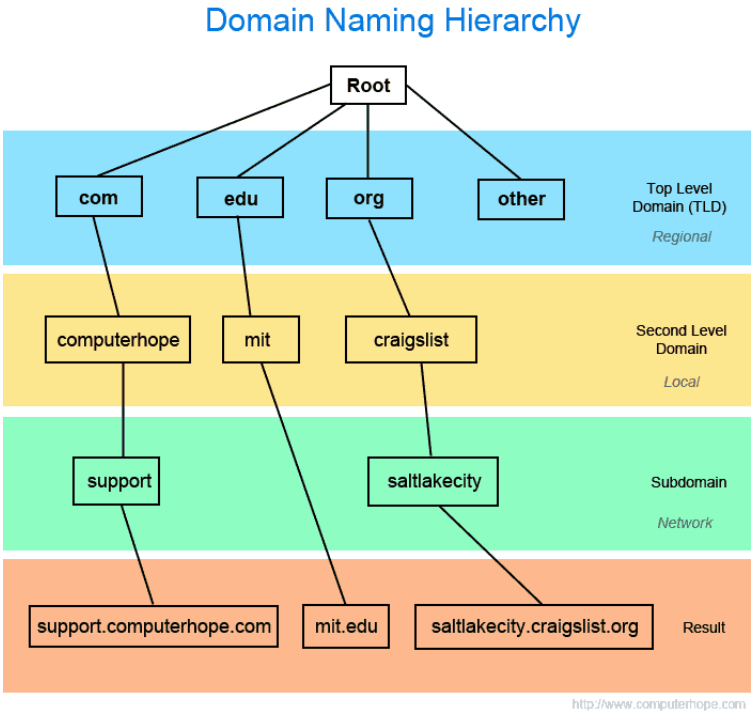


Figure 1.4: The examples of domain naming hierarchy.

date of creation of a domain, use the key field *created_at* and others registrars use *creation_date*.

Domain Name:	ASM.COM
Registrar:	NETWORK SOLUTIONS, LLC.
Sponsoring Registrar IANA ID:	2
Whois Server:	whois.networksolutions.com
Referral URL:	http://networksolutions.com
Name Server:	CBRU.BR.NS.ELS-GMS.ATT.NET
Name Server:	CMTU.MT.NS.ELS-GMS.ATT.NET
Status:	clientTransferProhibited
Updated Date:	12-sep-2014
Creation Date:	07-oct-1997
Expiration Date:	06-oct-2019

Figure 1.5: WHOIS information of *asm.com*.

WHOIS features In this work when the author refers to *WHOIS features* is talking about the extracted fields of the WHOIS information of domains,

being utilized in the WHOIS distance algorithm explained later in the chapter 2. The reader can find more information about WHOIS features in the section 2.1.

State of the Art

One of the objectives of this thesis is to propose a distance algorithm to relate domains using their WHOIS information. ManaTI does not pretend to detect malicious domains. However, it can connect malicious domains from previous domains analyzed.

WHOIS Similarity Distance

To achieve the WHOIS Similarity Distance algorithm is necessary to understand the characteristics of legitimate and malicious domains, and the techniques used to detect malicious domains. In others words, how different or similar are malicious domains on normal domains. WHOIS Similarity Distance must be able to relate domains, no matter if they are malicious or legitimate domains.

Bilge et al. have proposed a system called EXPOSURE, used to classify malicious/legitimate domains in real time [19]. The system uses 15 features grouped in 4 categories and the classifier is the J48 algorithm. The experimentation took around 17 months; they were using real-time dataset and offline logs. The system works fine in practice, and it has a high rate of detection, only 1% of false positive rate. Also, the system EXPOSURE has some limitation, for example, it depends on the feature selection, and the attacker could try to avoid it.

In the case of Shuang Hao et al. they have not proposed a system, but they provided a good acknowledgment of the behavior of malicious domains in their work "Monitoring the Initial DNS Behavior of Malicious Domains" [6]. They found that the malicious domains have some characteristics:

- Resource records of malicious domains tend to resolve to particular IP address range and Autonomous Systems (AS). The legitimate domains rarely have resource files within the tainted AS set that host scam domains,
- They discovered that malicious domains display distinct clusters, regarding the networks that are searching these domains,
- They found that malicious domains become widely popular more quickly after their initial registration time.

Further, Shuang Hao et al. explain to us that the legitimate and malicious domains have characteristics and it may be possible to fingerprint domains

based on their resource records and lookup traffic of TLD name servers before and attack.

Masahiro Kuyama et al. have proposed a "Method for Detecting a Malicious Domain by using WHOIS and DNS features" [20]. They design a method for detecting the Command and Control server (C&C) by using supervised machine learning (SML) using obtained feature from WHOIS information and DNS of domains of C&C servers and legitimate domains.

The feature points collected of email addresses used for C&C domains as contact information. With this information was proposed a method to determine C&C servers by using machine learning with WHOIS and DNS information. Furthermore, they classified the features of WHOIS information of C&C domains, by showing a relation of words of extracted email addresses in the co-occurrence networks. Lastly, they evaluated domain names and email addresses from the WHOIS information as input values for machine learning. They obtained as result of the experiment 98.5 % of rate detection.

Letal, V. in his master's thesis have proposed "Discovering of malicious domains using WHOIS database". It is about a reputation based system which is using WHOIS information to be able to give an estimation of maliciousness. This system also can work even for domains which have not appeared in blacklists or were not previously observed.

Based on information extracted from WHOIS records, the system learns the behavior of observed domains and it can generalize to other unobserved domains. Its model is probabilistic and uses Variational Bayes to determine its parameters.

To do the experiments were used proxy logs from a primary Intrusion Prevention System (IPS). The system has a reasonable false positive rate, around 0.03%, with 52% true positive rates and a precision of 92%. Also, the system only works for off-line learning.

Relation with WHOIS Similarity Distance algorithm

As far we know, it does not exist an algorithm or implementation which try to create distance between the WHOIS information of domains and relate them.

Weblogs

As far we know, does not exist a software to assist analysts like ManaTI does it. There is much software to visualize weblogs, but any of them provide tools to analyze them. They are mostly focused on system administrator task or as HTTP logs viewer. The following software are weblogs viewer:

- Apache-scalp
- Apache Log Viewer

The AI^2 system assists the analysts, but it is more focused in detection malicious behavior than providing tools for the analyst. AI^2 has become a commercial application [3]. Kalyan Veeramachaneni et al. proposed a system called AI^2, "Training a big data machine to defend" [21]. AI^2 is a system where Analyst Intuition (AI) is put together with current and well-proved machine learning algorithm to build a complete end-to-end Artificially Intelligent solution (AI). AI^2 tries to learn from the analyst behavior and from machine learning algorithm for detecting malicious behavior. The Figure 1.5 illustrates a small summary of how AI^2 works.

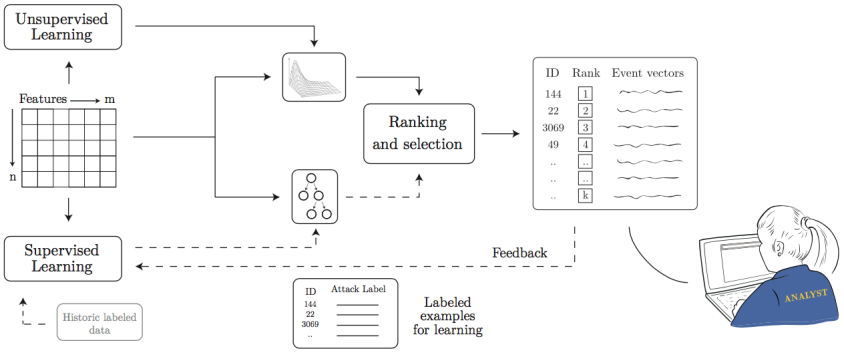


Figure 1.6: Workflow of how AI^2.

The system has four keys features:

- a platform to analyze big data
- a modeler for outlier events
- Logic to learn of security analysts through their feedback
- and a supervised learning module for detecting and determining malicious behavior

The system was validated with a real-world data with 3.6 billion log lines. It can learn to defend against unobserved attacks. And on supervised outlier analysis, the software can improve the detection rate in 2.92 times and reduce false positives by more than five times. AI^2 is part of our motivation for ManaTI, although ideally, ManaTI is thought only to assist analysts. In the future we want to allow the system to learn from the analysts' behavior to improve their efficiency.

[3] www.patternex.com

Assisting tools and testing

As far as we know, do not exist published techniques to measure or diagnostic the usability of an assisting tool especially in the area of cybersecurity.

Chapter 2

WHOIS Similarity Distance Method

Frequently the domains of one company or organization share common information in the WHOIS data; like contacts' emails, organization names or others. The situation for malicious domains is similar; domains registered by the same person or for the same purpose very often repeat some characteristics in their WHOIS data and also typical patterns of malicious domains [22].

The use of WHOIS information for malware detection has been studied and has gotten good results [20] [23]. Detecting malware domains using WHOIS information is possible to some.... Therefore this research goes further to try a method to measure a distance between domains called WHOIS Similarity Distance and find a way to relate them. Several domains are connected if they share the minimum number of relevant information in the WHOIS data.

To accomplish this goal, several fields from WHOIS information were extracted and analyzed of many records. Taking account the more relevant and usual fields of the WHOIS data. Also were studied several libraries of Python to get the WHOIS records.

2.1 Feature selection

The registrars provide WHOIS information, and they can share the information using their structure. Therefore the fields names (key) can differ among registrars. And also, some information can be missed. For that reason were selected several key features (or fields) for the algorithm. These essential features must be discriminative. The list of the selected features is the next:

1. Basic contact information
    a) Registrar's name k_n,
    b) Contact's name or registrant's name k_{cn}.

c) Organization's name k_o ,

d) Contacts emails k_e ,

e) Post address or zip code k_p,

f) Domain's name k_d.

2. Duration of a domain

a) The numbers of days counted between from creation date and expiration date of the domain k_{pd},

3. Domains' name Server

a) Servers' name k_{ns},

Each domain d is represented as a set of features $K_d = (k_n, k_o, k_e, k_p, k_{pd}, k_{ns}, k_d, k_{cn})$.

2.1.0.1 Basic contact information

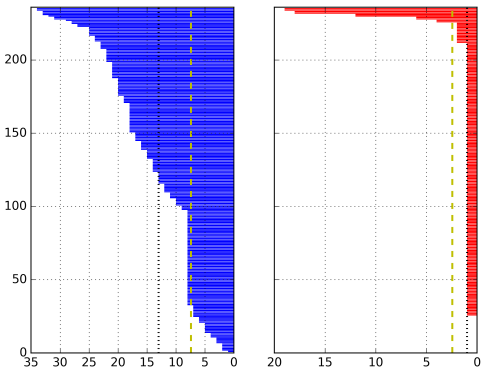
The basic information typically provided by all registrars are registrar's name k_n, contact's name k_{cn}, organization's name k_o, contacts emails k_e and post address or zip code k_p. They are tested to be useful for detecting malicious domains [20] or rather, for getting a prior estimate about maliciousness of domains d [23].

2.1.0.2 Duration of a domain

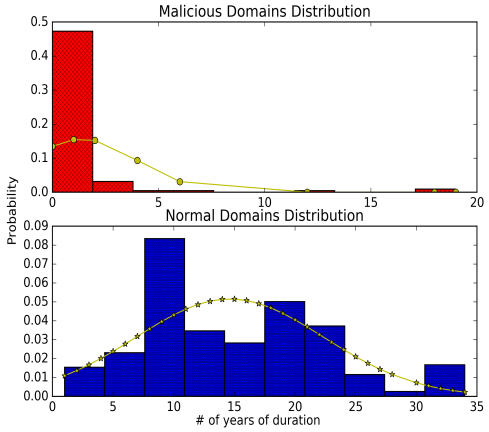
This feature, the numbers of days between creation date and the expiration date k_{pd} is selected because it helps to propose a differentiating feature for legitimate or malicious domains. Typically the legitimate domains are created for terms greater or equal to 2 years ($\geq$ 720 days); however, the malicious domains often have a shorter duration [24] [22]. Although, it is also common that some malware source hijacks old normal domains and uses them for its purpose.

The Figure 2.1 shows a comparison between 236 random normal domains and 236 malicious domains used by the Locky ransomware network. The Figure 2.1a shows the sorted years duration of the domains, from the longest to the shortest. The average of duration:

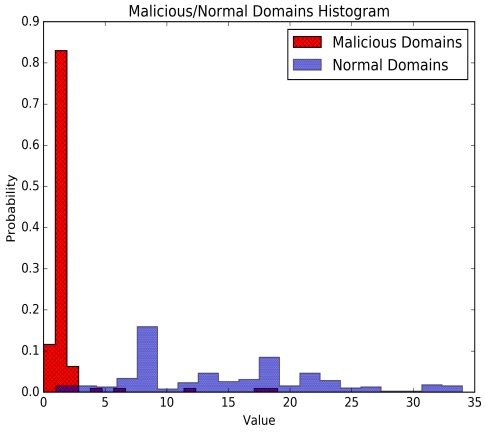
- Normal domains are around ten years;
- Malicious domains is around two years. Also, some malicious domains showed in the Figure 2.1a have a period more than a decade; we suppose that those domains were hijacking.



(a) Sorted years of duration per sample, from the longest to the shortest. The plot of the left side represents the duration of normal domains and the plot of the right side is about the malicious (Locky) domains.



(b) Distribution of the years of duration of Locky/Normal domains separated.



(c) Distribution of the years of duration of Locky/Normal domains together.

Figure 2.1 Comparison of duration between 236 normal domains and 236 malicious domains used by Locky ransomware. The distribution of the years of duration of Locky/Normals domains. Locky's domains have high probability to have a duration of between 1 and 2 years, and normal domains have high probability to have a duration between 8 - 10 years. The normal domains have more uniform distribution than malicious domains.

Figures 2.1b 2.1c represent the distribution of the duration of the selected normal and Locky domains. The first plot separates the distributions and the second plot put together. The x-axis represents the range of years length of the selected domains and y-axis represents the probability to get that duration. The Locky domains have more probability to have two years of duration than normal domains. Normal domains have a high likelihood of having a duration around 10 years. And also normal domains have a distribution more uniform than the Locky domains.

2.1.0.3 Domains' name Server

Domains of several entities could be hosted by same servers, this is a relationship between them. The relationship could be high or not depend on the case [20].

2.2 Datasets - WHOIS records

The used domains were found in several entities' websites and other websites for researcher purposes. To choose the legitimate domains, it must be confident that they belong either to a regulated entity or well-known company or use. For determining malicious domains, they must be reported as being used by a malware network. Legitimate domains were downloaded from the website *OpenDNS*[4] or using the *YouGetSignal.com* [5] tool to do *Reverse IP Domain Check*. And also it was used lists of well-known companies domains as Facebook, Google, Apple, Cisco, Microsoft, Oracle, Twitter, Toyota, HP, IBM, Amazon, Intel, Qualcomm, Xerox, eBay, Danaher, Thermofisher, Micron, Jabil, WDC, CSC, Ti and others.

Malicious domains were obtained from the following projects:

- DNS-BH - Malware Domain Blocklist[6],
- Malware Domains List[7],
- Ransomware Tracker[8].

More especially the following malware families were used:

- CryptoWall Ransomware [25],
- Locky Ransomware [26] ,

[4] www.opendns.com
[5] www.yougetsignal.com/tools/web-sites-on-web-server/
[6] www.malwaredomains.com
[7] www.malwaredomainlist.com
[8] ransomwaretracker.abuse.ch

- TeslaCrypt Ransomware [27],
- TorrentLocker Ransomware [28].

In total, we have approximately 1300 domains, which were labeled if they are malicious or legitimate domains and by the relationship with some entity or purpose. In this thesis, we refer to a malicious purpose for examples: malware's network, phishing campaign, and others.

2.2.0.1 Tools for getting the WHOIS records

Registrants can provide the WHOIS information in different formats, therefore was necessary to adapt to a format for the experiments. There are several tools which provide different formats also is common that no all of them contain the same information fields, often some fields are empty or directly do not exist. For this experiment was used four WHOIS libraries of Python:

- pythonwhois *(pw)* [9],
- virustotal *(vt)* [10],
- passivetotal *(pt)*[11],
- whois *(ww)* [12].

For testing the listed libraries were realized the next tasks:

1. to get the WHOIS information of all our available domains,
2. to extract the necessary features for the algorithm,
3. and count how many of them have all the features (fields) filled,
4. in the end, it was taken an average of all the WHOIS feature domain per library

The table 2.1 shows in average the percentage of selected features filled by the libraries.

This experiment concluded that the most efficient library to get WHOIS information of domains is *passivetotal*. To use the PassiveTotal's API (used by library selected) is necessary to have an account. All the experiments made for the Whois Similarity Distance algorithm was used information provide by PassiveTotal's tool, but for the WHOIS Similarity Distance module implemented in ManaTI is used the library *pythonwhois*.

[9] https://pypi.python.org/pypi/pythonwhois
[10] https://github.com/nu11p0inter/virustotal/
[11] https://pypi.python.org/pypi/passivetotal
[12] https://pypi.python.org/pypi/python-whois

Library	Average percentage of obtained features
VirusTotal	43%
python-whois (pywhois)	60%
pythonwhois	76%
Passive Total	89%

Table 2.1: Comparison of the studied libraries to get WHOIS information in Python. Exist 8 features for the WHOIS Similarity Distance obtained from the WHOIS information. Not all the tools obtained all the features. This table show that the library of Passive Total obtained in average the most amount of features per domain.

2.2.0.2 Grouping Domains

During this research was necessary to see the WHOIS information of each domain, which could be malicious or not. Because:

- To be sure of data provided by the libraries and compare them with different WHOIS providers,
- To search for patterns to relate domains,
- To compare different WHOIS data structure and find things in common,
- For processing the WHOIS information also is necessary to handle the encoding of the data, especially for non-Latin characters. It was a need to see which characters encodes would be required.

And after this, we learned that: some domains even if they belong to some group or entity, may differ their WHOIS information, sometimes the domains do not share information in common; the registrars allocated outside of Europe or USA often use different structures and sometimes omit some fields; dealing with non-Latin characters is hard task. In table 2.2 compares the WHOIS information of two well know domains: *instagram.com* and *facebook.com*; both domains belong to the company Facebook Inc. The information that they share are: one contact email and the zip code. The table information was obtained from DomainTool[13].

The WHOIS features of domains of the same entity were compared measuring the distances between themselves. The distance, in this case is a numeric sum of the distances features of those domains. In the section 2.3.1, the reader can see how this WHOIS Similarity Distance algorithm works. The result obtained can be appreciated in Figure 2.2.

With this experiment was learned that often domains of the same group do not have the same WHOIS information but could share some data. So it was

[13]http://whois.domaintools.com

WHOIS fields	Instagram.com	Facebook.com
domain's name	instagram.com	facebook.com
Registrar	REGISTRARSEC LLC	MARKMONITOR INC.
Registrant name	Domain Admin	Domain Administrator
Registrant Org	Instagram, LLC	Facebook, Inc.
Email	[abusecomplaints@ registrarsec.com, domain@fb.com]	[abusecomplaints@ markmonitor.com, domain@fb.com]
Creation Date	2004-06-04	1997-03-29
Expiration Date	2022-06-04	2025-03-30
Zip code	94025	94025
Name Server(s)	[NS-1349.AWSDNS-40.ORG, NS-384.AWSDNS-48.COM, NS-2016.AWSDNS-60.CO.UK, NS-868.AWSDNS-44.NET]	[a.ns.facebook.com, b.ns.facebook.com]

Table 2.2: Although *instagram.com* and *facebook.com* belong to the same company (Facebook Inc.), they do not have same WHOIS information. However, in this example we can see that they share one contact email and the zipcode.

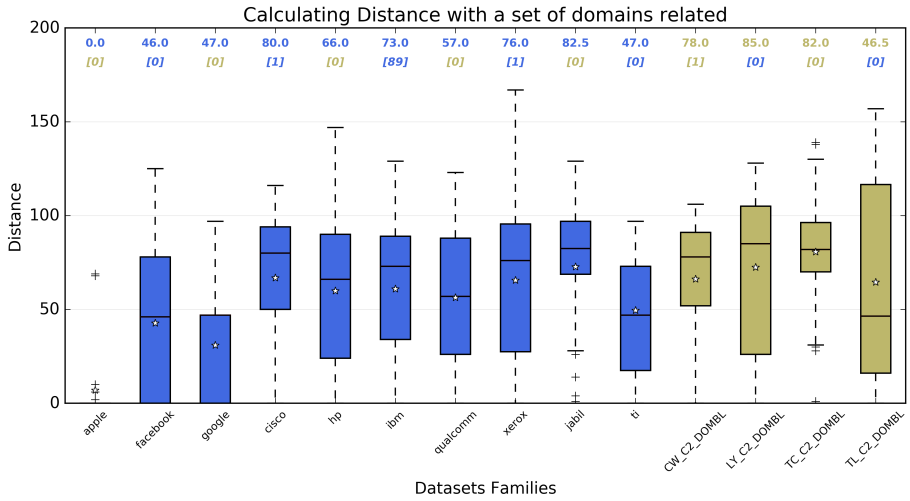


Figure 2.2: We expected that the distance between domains of the same group should be close to zero, with this plot we can see that this assumption is false. Still, the average distance to sort out the groups is not infinite and is close to 100.

decided to create subgroups of the groups (or entities) manually and only have been chosen for the WHOIS Similarity Distance's experiments the subgroups

with more than ten elements. This because if we have only a few items that belong to the same subgroups could go unnoticed in the tests. Figure 2.3 represents the same dataset of the Figure 2.2 but separated in sub-group were the WHOIS Similarity Distance tends to zero.

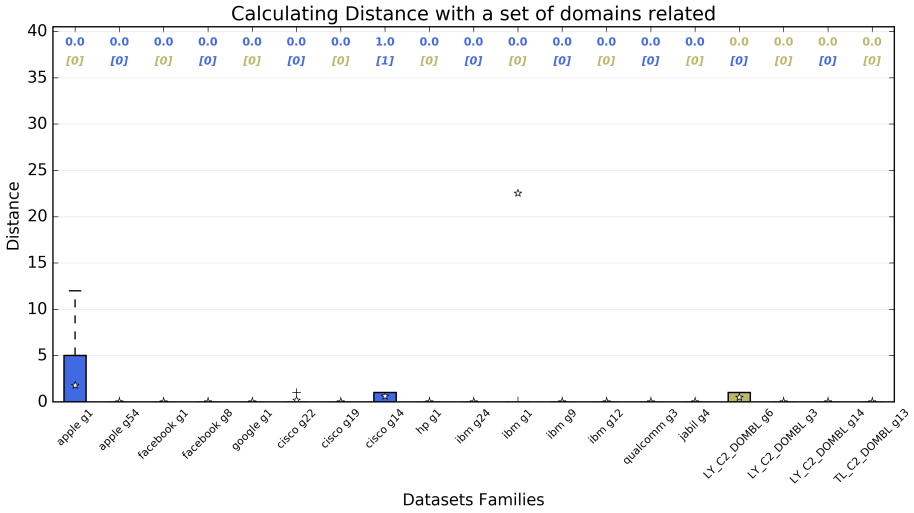


Figure 2.3: After the sub-grouping were obtained groups like was needed, well separated and related between them. And with the WHOIS distance inside the subgroups tending to zero.

2.2.0.3 Construction of training and testing dataset

The used dataset is a comparison between all the domains available for this thesis. The distances of each feature are stored with references to the compared domains. Also, a boolean field is saved to identify if both compared domains are related according to the sources where were acquired the domains. In total, we have around 172.000 comparisons. Of these comparisons, randomly, 80% were used to train and choose the proper linear model classifier. And the rest 20% were used to get the finals results of the most suitable algorithm. This topic will be addressed in more detail in the section 2.3.1.2.

2.3 Domains Distance Proposed

One of the objectives of this research is to create a method which would be able to calculate a numeric distance between two domains (d_A and d_B) and relate them using their WHOIS information called WHOIS Similarity Distance.

The next section is talking about the WHOIS relation algorithm between two domains also with the experiments performed and their respective results. At the end, the reader could see experiments made with the idea to compare

and analyze: the numeric distance algorithm between two domains and the algorithm to determine if they are related or not.

For plotting all the results of the experiments made, was used the Python library called Matplotlib [29].

2.3.1 WHOIS Similarity Distance

The numeric value of the WHOIS Similarity Distance method is obtained comparing two domains. The value is a sum of the string distance obtained per selected feature of the compared domains. The numeric value of WHOIS Similarity Distance also is called Global Distance (GD).

$$Dist_T(d_A, d_B) = \sum_{i=1}^{n} dist_{k_i}(k_{i_{d_A}}, k_{i_{d_B}}) \tag{2.1}$$

where:

- d_A, d_B WHOIS information of the domains A and B
- $k_{i_{d_A}}$ and $k_{i_{d_A}}$ the i-features of the domains A and B
- $dist_{k_i}(k_{i_{d_A}}, k_{i_{d_B}})$: the distance between the i-feature of the domain A and B.
- $Dist_T(d_A, d_B)$: The Global Distance of WHOIS information of two domains.

2.3.1.1 Distance per features

When the algorithm measures distance, takes the WHOIS feature of the two domains A and B. And it compares the string distance $dist_{k_i}(k_{i_{d_A}}, k_{i_{d_B}}, ...)$ of each duple of features. The table 2.3 is comparing the WHOIS information of *google.com* and *facebook.com* demonstrates how WHOIS Similarity Distance works:

For measuring the individuals distances $dist_{k_i}$ are used the following rules:

- If the features $k_{i_{d_A}}$ and $k_{i_{d_B}}$, are strings, the distance is obtained using the Levenshtein algorithm for strings distances [31]. Levenshtein algorithm is a well know method to measure a distance between texts. The distance provided is the number of changes necessary to convert one string to the another. The " number of changes" refers to numbers of to insert a character, remove a character and change the order. The Figure 2.4 code in Python, shows examples of how the Levenshtein algorithm works:
- If i-feature has an array of strings (for example contact's emails), all the elements of both arrays are compared using their string distance, and the final result is the shorter distance (or shorter comparison).

Features list k_{i_d}	WHOIS info A	WHOIS info B	Numeric Distance $dist_{k_i}(k_{i_{d_A}}, k_{i_{d_B}})$
k_n registrar's name	MARKMONITOR INC.	MARKMONITOR INC.	0.0
k_{cn} contact's name.	DNS Admin	Domain Administrator	13.0
k_o org.'s name	Google Inc.	Facebook, Inc.	8.0
k_e contacts emails	[dns-admin@google.com]	[domain@fb.com]	11.0
k_p zip code	94043	94025	2.0
k_d domain's name	google.com	facebook.com	8.0
k_{pd} duration in days	8401	10229	0.82
k_{ns} servers' name	[ns1.google.com,...]	[a.ns.facebook.com ...]	11.0
Global Distance $Dist_T(d_A, d_B) =$			53.82

Table 2.3: Comparison of the WHOIS information of google.com and facebook.com

```
import Levenshtein
print(Levenshtein.distance('googlee.com','google.com'))
# output >> 1
print(Levenshtein.distance('toogle.com','google.com'))
# output >> 1
print(Levenshtein.distance('tooglee.com','google.com'))
# output >> 2
print(Levenshtein.distance('google..com','google.com'))
# output >> 1
print(Levenshtein.distance('gglooe.com','google.com'))
# output >> 4

```

Figure 2.4: Example in Python how to do Leveshtein Distance

- In the case of the numbers of days between two dates k_{pd}, it is taking a ratio between both values k_{pd}. The ratio gets a decimal number between 0 and 1. So if two domains have a similar number of days of duration, the result is closer to zero. Otherwise is the number of days is different, the result is closer to 1.

2.3.1.2 Experiments with WHOIS Similarity Distance

The WHOIS Similarity Distance algorithm is used to have a numeric distance of the WHOIS information of two domains . Theoretically, it could identify related domains without regard if the domains are malicious or legitimate. And supposing that the weights of the features are equals, that is, all the features have the same importance. To achieve that is necessary to define a threshold.

Using a Receiver Operating Characteristic (ROC) curve is possible to find the most optimums threshold to relate domains. Setting a range of possible thresholds between 0 and 200 in Figure 2.5, we can appreciate that the apogee value is 75 of Global Distance. However, it would have a True Positive Rate (TPR) of 50% and around 30% for the False Positive Rate (FPR). The number 75 was found searching the shorter distance between the point TPR equal 100% and FPR equal 0%.

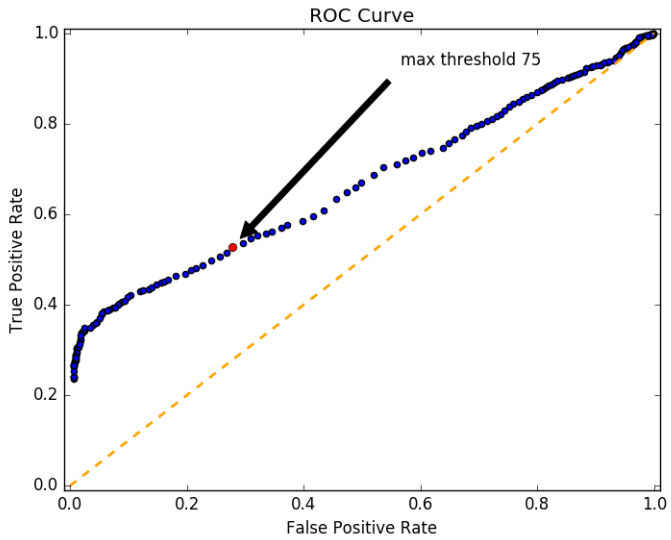


Figure 2.5: ROC curve of algorithm using all samples available. It is the result obtained using a thresholds' range between 0 and 200. Looking the plot is posible to see that the optimum threshold with weights equal to 1 is 75, with 50% of TPR and 30% of FPR.

The error is wide; it has only around 50% of chances to relate correctly two domains. In this experiment, it was learned that some features are more important that others. To get better results is necessary to apply a classifier algorithm to find the correct weights for the features.

2.3.2 Classification Based on the WHOIS Similarity Distance

For the WHOIS Similarity Distance is needed to define a threshold for the distance to relate two domains. In the section before was learned that some features are more important than others. To facilitate the task of looking for the correct weights and threshold, we decided to use a classifier algorithm. For this work were chosen linear model algorithms. The implementations of the linear models classifiers were obtained from the Python library ScikitLearn [32].

The classifiers analyzed were Linear Regression and Polynomial Regression algorithms with degree 2 and 3 [33]. These classifiers were selected for several reasons:

- The selected classifiers are well known, and there are much documentation about their results,
- Their implementations are not hard to do,
- They are fast to train if we compare with Support Vector Machine (SVM) classifiers or Polynomial Regression with degree 4. And also their predict function is not time-consuming,
- Once the coefficients are trained, they can be changed manually. Using some techniques the user can modify the factors, doing the classifier more or less sensitive, without need to train the classifier again. This was not done in this thesis, and we let it as a future line.

2.3.2.1 Linear Models

Linear models were mostly developed before the computer age of statistics. Nowadays they are being studied with good results. One main characteristic of linear models is their simplicity, and they provide a good description of how the inputs affect the output. When the number of training dataset is small, they can produce better results than nonlinear models [34].

For the experiments were done two linear models algorithms: Linear Regression and Polynomial Regression.

2.3.2.2 Linear regression model

A linear regression model assumes that the regression function $E(Y|X)$ is linear in the inputs $X_1, ..., X_p$. The input vector $X^T = (X_1, X_2, ..., X_p)$ [14] is provided, and want to predict a real-valued output Y. The linear regression function has the form:

$$f(X) = \beta_0 + \sum_{j=1}^{p} X_j \beta_j \tag{2.2}$$

[14] T indicates Transposed

In Linear regression, the linear predictor functions are estimated from the data. The most commonly used is the conditional mean of y given the value of X, is assumed to be an affine function of X. Linear regression concentrates on the conditional probability distribution of y given X.

Linear Regression was the first type of regression analysis being studied and to be applied in practical situations. Because models have a linear dependency of unknown parameters are easier to find than model with non-linear parameters, and also the statistical properties of the resulting estimators are easier to understand and determine.

Linear Regression has many practical approaches. The following are some of them:

- It can be used for prediction, anticipate, or error reduction,
- It analysis can be applied to quantify the clout of the relationship between y and the X_j.

Given a data set $\{x_{i1}, ..., x_{ip}, y_i\}_{i=1}^{n}$ of n units. A Linear Regression model assumes that the variable y_i and the regressors x_i has a linear relationship. It is necessary to add a disturbance term or error variable ε_i. This variable ε_i is an unobserved random variable that adds noise to the linear model between y_i and the regressor x_i. The prediction formula y_i

$$y_i = \beta_0 1 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i = \mathbf{x}_i^T \beta + \varepsilon_i, \qquad i = 1, \ldots, n, \tag{2.3}$$

where ^{T} denotes the transpose, so that $x_i^T \beta$ is the inner product between vectors x_i and β [34].

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \quad \mathbf{X} = \begin{pmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_n^T \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1p} \\ 1 & x_{21} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{np} \end{pmatrix}$$

$$\beta = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{pmatrix} \qquad \varepsilon = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix}$$

also, it could be expressed using a matrix as:

$$\mathbf{y} = X\beta + \varepsilon, \tag{2.4}$$

2.3.2.3 Polynomial regression models

Polynomial Regression is a form of Linear Regression. Where the relationship between x and the dependent variable y is represented as a polynomial in

x of n-th degree. It uses nonlinear relationship between the value of x and the conditional mean of y, stand for $E(y|x)$. Polynomial Regression could be utilized for the growth rate of tissues, the distribution of carbon isotopes in lake sediments and progression of disease epidemics.

Polynomial Regression is considered to be a special case of multiple linear regression. Because Polynomial Regression fits in a nonlinear model to the data, as a statistical estimation problem it is linear.

In simple linear regression, the model is $y = a_0 + a_1 x + \varepsilon$. However in this model, for each unit increased in the value of x, the conditional expectation of y increases by a_1 units.

In many cases, such linear relationship may not hold, so is necessary to propose a quadratic model $y = a_0 + a_1 x + a_2 x^2 + \varepsilon$, in general, the expected value of y could be polynomial of an n-th degree like $y = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \cdots + a_n x^n + \varepsilon$.

The Polynomial Regression model

$$y_i = a_0 + a_1 x_i + a_2 x_i^2 + \cdots + a_m x_i^m + \varepsilon_i \ (i = 1, 2, \ldots, n) \tag{2.5}$$

can be expressed as a matrix form [34]. Then the model can be written as a system of linear equations:

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} 1 & x_1 & x_1^2 & \ldots & x_1^m \\ 1 & x_2 & x_2^2 & \ldots & x_2^m \\ 1 & x_3 & x_3^2 & \ldots & x_3^m \\ \vdots & \vdots & \vdots & \vdots & \\ 1 & x_n & x_n^2 & \ldots & x_n^m \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_m \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \vdots \\ \varepsilon_n \end{pmatrix}$$

The rest of the equations are the same to Linear Regression. The main difference between Linear and Polynomial Regression is working with non-linear unknown variables. Also, the linear features of Linear Regression could be taken and rebuilt, creating new non-linear features. Something similar was made for the experiments performed for the WHOIS Similarity Distance algorithm.

2.3.2.4 Learning curve of the classifiers

For measuring the quality of the WHOIS Similarity Distance method was used learning curve . Learning curve refers to a plot of the prediction accuracy or error, versus the training set size; that is how better the model can predict the target as increasing number of instances used to train it.

The Figure 2.6 shows an example of how a Learning Curve should look. The x-axis is the number of samples and the y-axis is the score of the prediction. The above line represents the curve of the same model using samples set for training and a distinct set for testing (training score). The bottom line

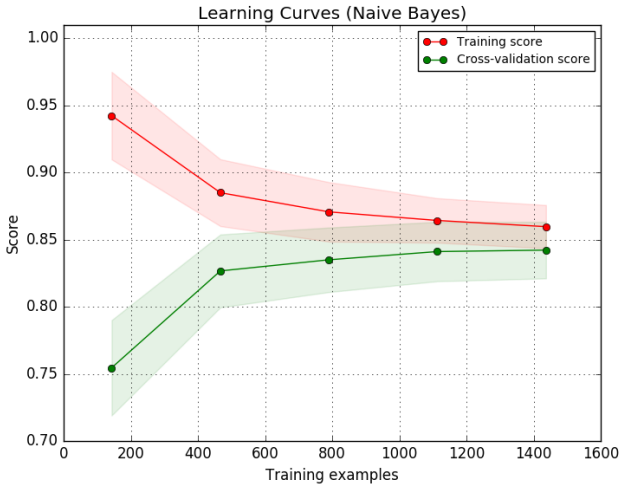
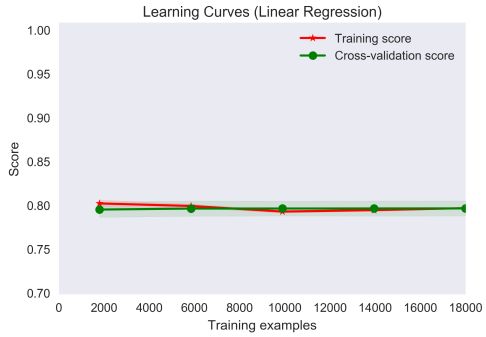


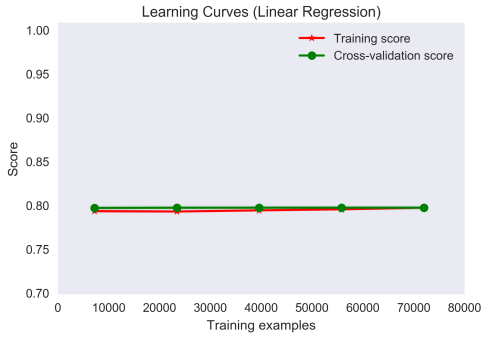
Figure 2.6: An example of how the Learning Curve looks.

represents the curve of the model using the same samples set for training and testing (cross-validation score). However, the score can be between 0 and 1. When the score is closer to 1, means that the quality of the model is good, otherwise, when the score is closer to 0, the quality of the design is wrong.

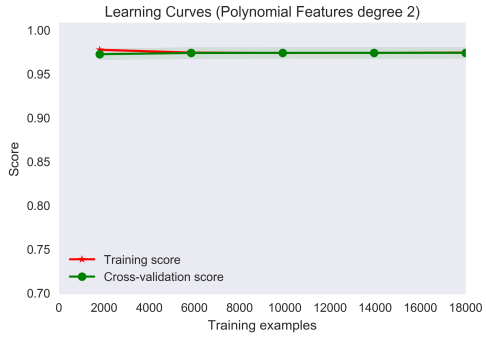
The linear models algorithm to be compared are linear regression, polynomial regression with degree 2 and with degree 3 [33]. The set of figures 2.7 show the learning curves of the studied classifiers. For the experiment was used the training dataset elements and the cross-validation was performed using K-Fold technique. K-Fold splits data in into k-consecutive folds to get train/test sets. Each fold is then used once as a validation while the k - 1 remaining folds form the training set.



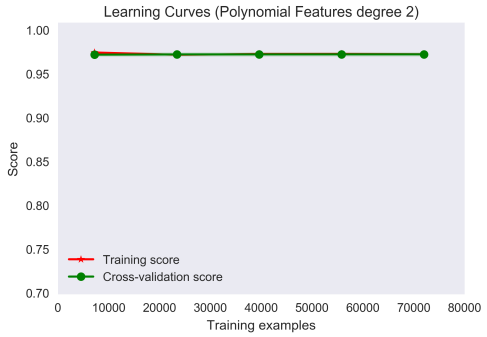
(a) Linear regression with 20,000 samples. The score is around 0.8.



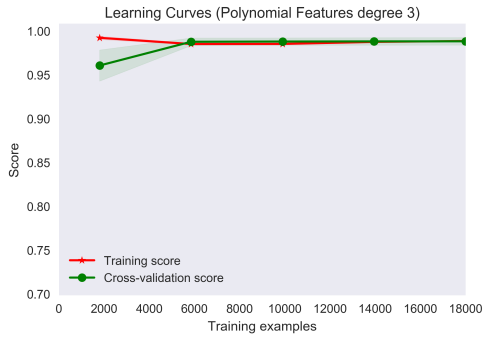
(b) Linear regression with 80,000 samples. The score is slightly higher than 0.8.



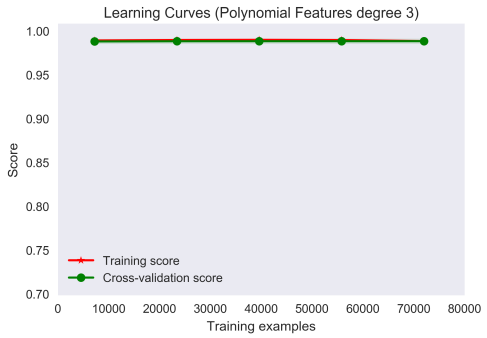
(c) Polynomial regression with degree 2 and 20,000 samples. The score is around 0.97.



(d) Polynomial Regression with degree 2 and 80,000 samples. The score is around 0.97.



(e) Polynomial regression with degree 3 and 20,000 samples.



(f) Polynomial regression with degree 3 and 80,000 samples.

Figure 2.7: Learning curve of the linear models proposed in this thesis with 20% of sample as testing samples. Polynomial regression with degree 2 and 3 has better score than the linear regression.

In the set of figures 2.7, the lines explained before, are not well appreciated. That means the scoring of the cross-validation and the training score are

slightly similar.

The figures 2.7a and 2.7b show the Learning Curve of the Linear Regression algorithm with different size of samples. The score is slightly improving according to as samples sizes increase. Again, the score is in the range of 0.76 and 0.80.

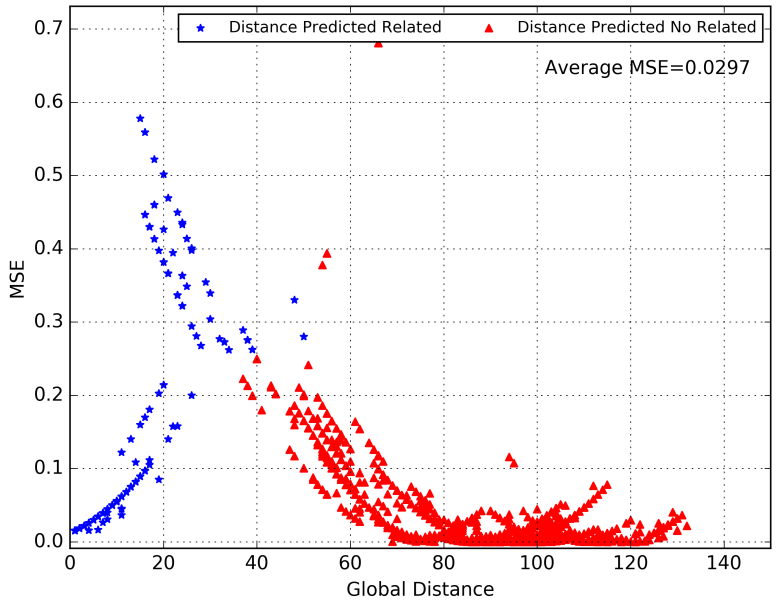
The figures 2.7c and 2.7d show Polynomial Regression with degree 2 and with different size of samples. The score is slightly constant according to as samples sizes increase. The score is moving around the score 0.97. The results of polynomial regression with degree 2 are better than the particular results of Linear Regression.

The figures 2.7e and 2.7f show Polynomial Regression with degree 3 and with different size of samples. With small sample sizes, the score is not well appreciated. As the sample size increases, the score is getting constant and it is around 0.99. According to this, with large samples sizes the Polynomial Regression with degree 3 is better than polynomial regression with degree 2.

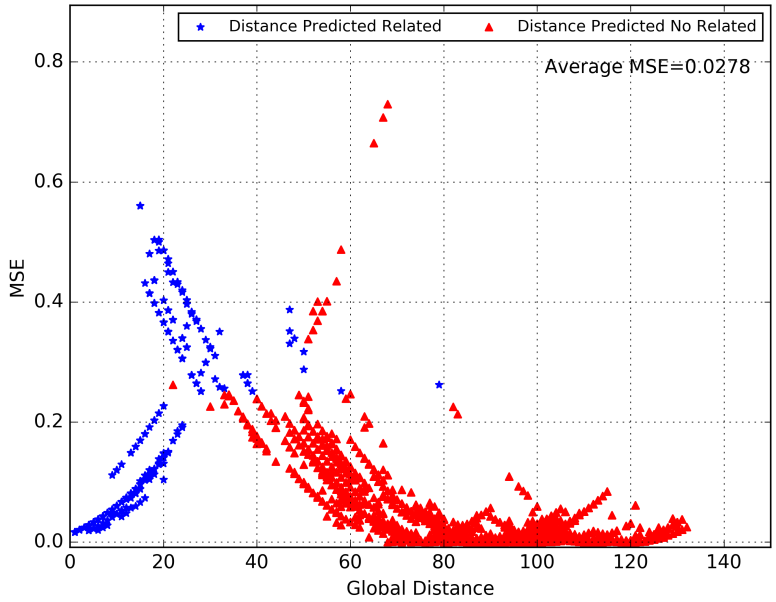
2.3.2.5 Mean Squared Error of WHOIS Similarity Distance using training dataset

In this section, the reader can see how we have chosen the most suitable classifier to connect domains using their similarities in the WHOIS information. The analyzed classifiers of linear models were: Linear Regression, Polynomial Regression with degree 2 and degree 3.

The experiments were made getting the Mean Squared Error (MSE) of the predicted relationship between domains. The plots 2.8, 2.9 and 2.10 show a relation between the distance values with the MSE obtained comparing real target and the target predicted by the WHOIS Similarity Distance algorithm. The x-axis represents the numeric distance value and y-axis is the MSE. If the MSE of a relationship is closer to one means that it has high odds have been wrong predicted, otherwise if the MSE is close to 0, says that the target was predicted correctly with high chances. The size of the testing samples is 20% with respect the total number of samples. For the next experiments were only used the dataset for training.

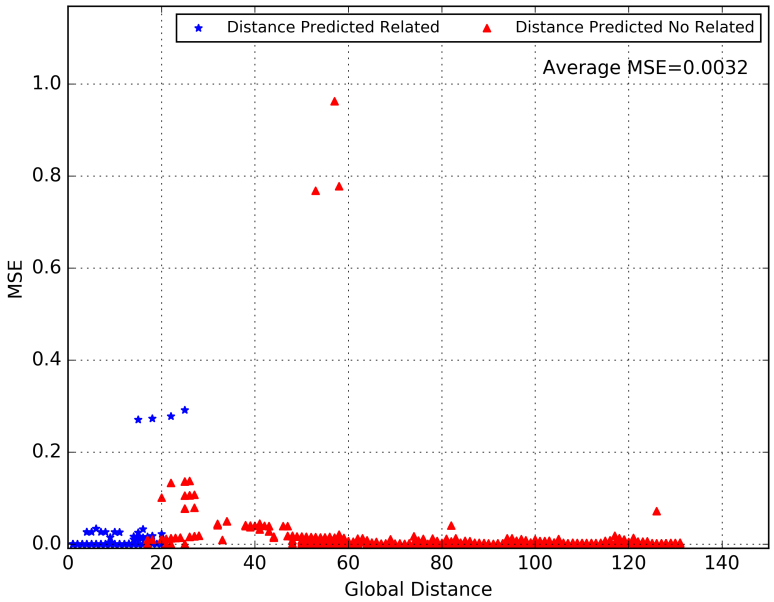


(a) Linear regression with 20,000 samples.

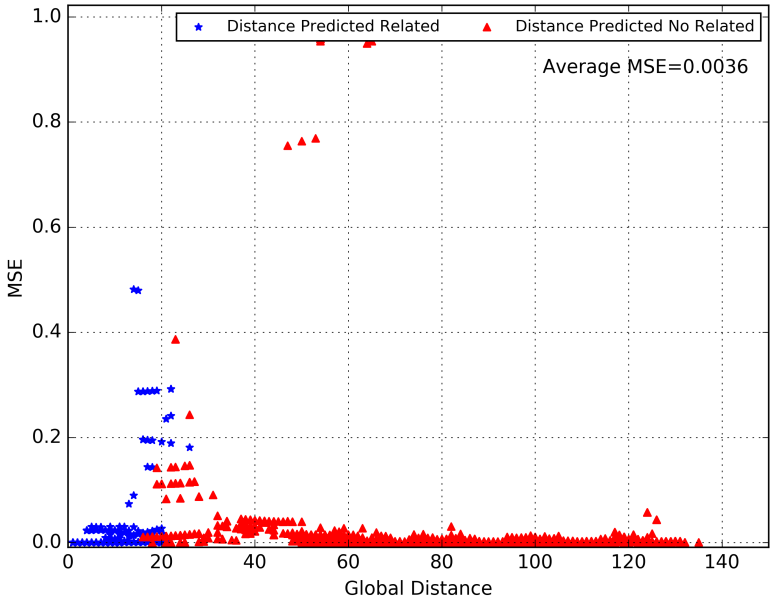


(b) Linear regression with 80,000 samples.

Figure 2.8: MSE of the Linear Regression classifier with GD value obtained. Linear regression produces in average a MSE high with respect the rest of the classifier. It means that the odds of bad predictions are high as the size of samples increases.

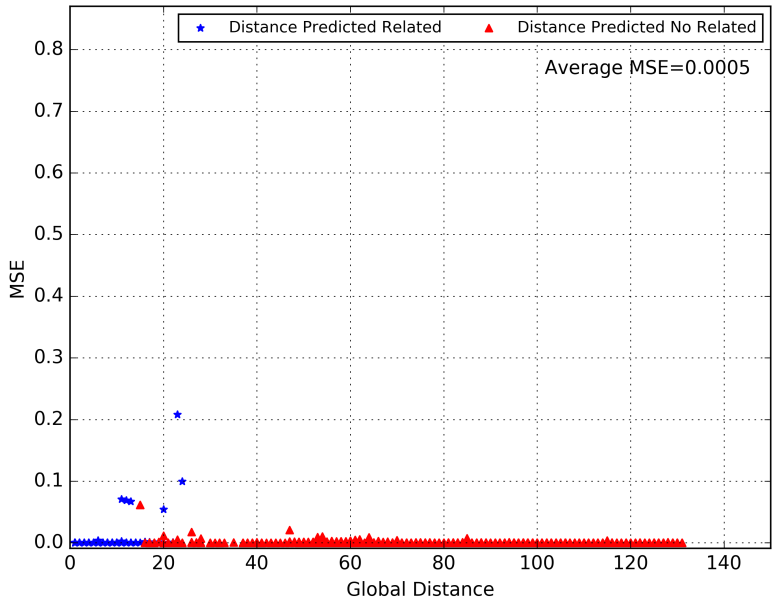


(a) Polynomial regression with degree 2 and 20,000 samples.

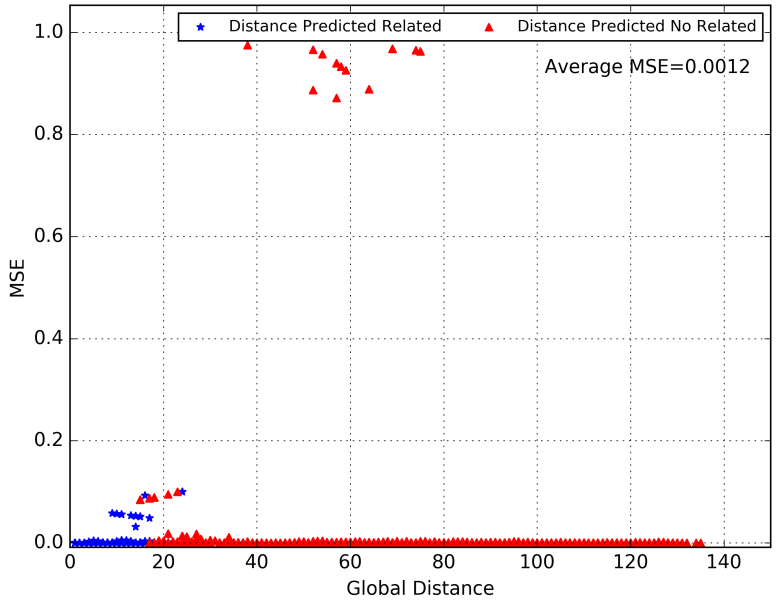


(b) Polynomial regression with degree 2 and 80,000 samples.

Figure 2.9: MSE of the Polynomial Regression with degree 2 with GD value obtained. Polynomial regression with degree 2 produces in average a MSE low with respect to the rest of the classifier, but it is not the best MSE produced.

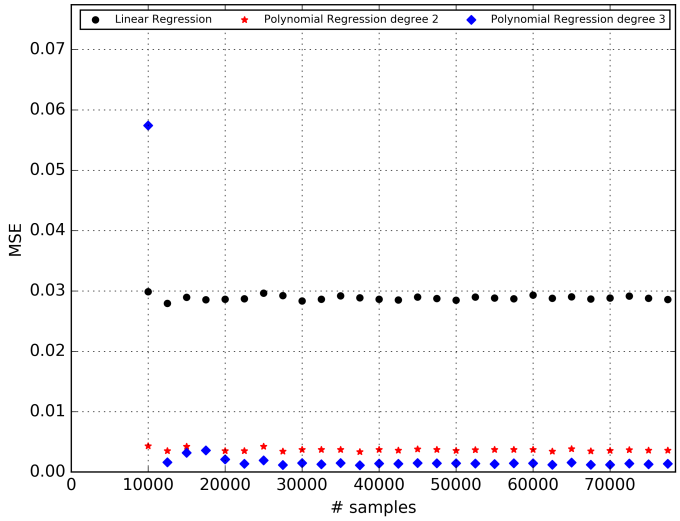


(a) Polynomial regression with degree 3 and 20,000 samples.

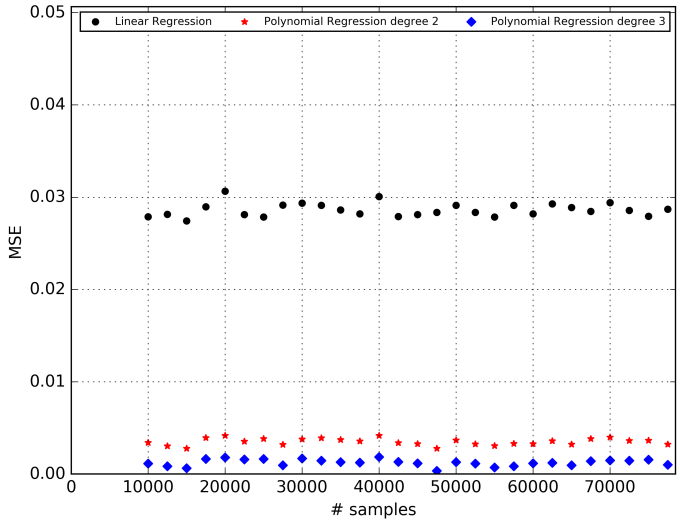


(b) Polynomial regression with degree 3 and 80,000 samples.

Figure 2.10: MSE of the polynomial regression with degree 3 with GD value obtained. Polynomial regression with degree 3 produces in average the lowest MSE with respect to the rest of the studied classifier. Low MSE means that the odds of bad predictions are low as the size of samples increases.



(a) The test sample size is the 80% with respect to the total numbers of samples provided. Polynomial regression with degree 3 has the lowest MSE, with small samples size can get MSE higher than the average. However when the sample sizes increases, the MSE is more constant.



(b) The test sample size is the 20% with respect to the total numbers of samples provided. Polynomial regression with degree 3 has the lowest MSE, and it looks constant as the samples sizes increases.

Figure 2.11: Comparison of the studied classifiers. X-axis is the number of samples provided and the y-axis is the MSE obtained. It is provided two plots, one with the test sample size of 20% and the another one with 80%. The polynomial regression with degree 3 has the lowest MSE, so is highly likelihood, that it can predict correctly. Thus, polynomial regression with degree 3 is more suitable than the others classifiers for our method WHOIS Relation Algorithm

In the Figure 2.11 shows that Polynomial Regression with degree 3 is more suitable than the other classifiers for our method WHOIS Similarity Distance. The error of the prediction is little, and it maintains constant on the samples size.

2.3.2.6 Experiments with testing dataset

In the section before, were compared the different studied classifiers looking for the most suitable classifier for our case using dataset for training. The most appropriate linear model classifier according to the MSE experiment is the Polynomial Regression with degree 3.

In this section initially, The WHOIS Similarity Distance is compared with the predicted relationship of the Polynomial Regression with degree 3. Then the MSE is plotted and also, the confusion matrix is showed. At the end of this section, the reader can see a statistic table where can be analyzed with more details the results obtained.

For the next comparison and experiments were used all the elements of the training dataset to teach the selected classifier, around 134000 elements. And for testing the algorithm were used all the testing dataset around 34000 items. The testing dataset was not used before.

The Figure 2.12 shows us that the comparative with the testing dataset looks similar to the comparatives performed in the section 2.3.3. The mean and the R^2 are almost equals. Figure 2.14 explains the MSE of the Polynomial Regression with degree 3 using the testing dataset. The error still is small, and it could be able to predict correctly almost 99% of the cases. The Figure 2.13 shows the confusion matrix of the predicted values.

The table 2.4 shows several metrics used to analyze the results of linear models classifiers. The statistics measures that we are interested are:

- The Kappa coefficient is 0.99,
- The Matthews correlation coefficient is 0.99,
- The F1 score is 0.99,
- The accuracy is 0.99,
- The percentage of error of the total number of predictions is 0.112%.

Thus, we conclude that the method of classification based on WHOIS Similarity Distance using the Polynomial Regression model with degree 3, has an accuracy between 98% and 99%. This accuracy with respect to the data provided by our built dataset.

Classes:	(related, unrelated)
Testing population:	34554
Condition positive:	6167
Condition negative:	28387
Test outcome positive:	6128
Test outcome negative:	28426
True Positive (TP):	6128
True Negative (TN):	28387
False Positive (FP):	0
False Negative (FN):	39
Sensivity (TPR):	0.993676017513
TNR = Specificity (SPC):	1.0
Pos Pred Value (PPV) = Precision :	1.0
Negative Pred Value (NPV):	0.998628016605
False-out (FPR):	0.0
False Discovery Rate (FDR):	0.0
Miss Rate (FNR):	0.00632398248743
Accuracy (ACC):	0.998871331828
F1 score:	0.996827978853
Matthews Correlation Coefficient (MCC):	0.996148939926
Informedness:	0.993676017513
Markedness:	0.998628016605
Cohen's kappa coefficient:	0.996141525
Prevalence:	0.178474272154
Positive Likelihood Ratio (LR+):	inf
Negative Likelihood Ratio (LR-):	0.00632398248743
Diagnostic Odds Ratio (DOR):	inf
False Omission Rate (FOR):	0.00137198339548
Error :	0.112%

Table 2.4: The results obtanied after to apply several metrics to evaluate classification models specifically on linear models. The statistics measures that we are more interested are: Kappa coefficient with 0.99, MCC with 0.99, F1 score with 0.99, ACC with 0.99 and the percentage of error of the total number of predictions with 0.112 %.

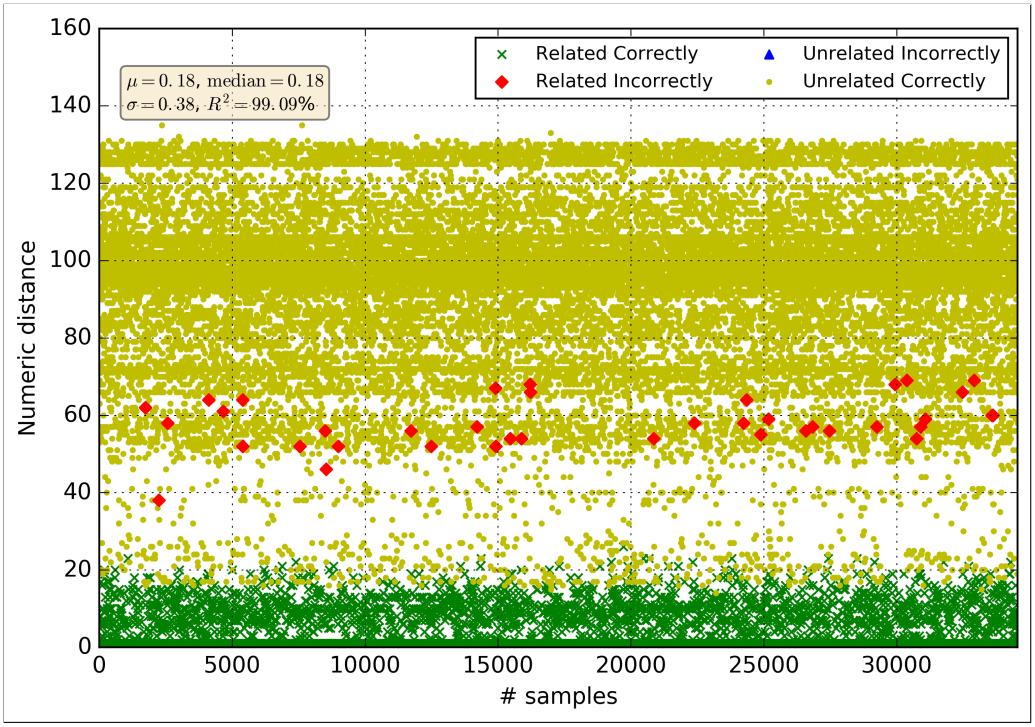


Figure 2.12: The Figure is showing the comparative between the WHOIS Similarity Distance algorithm and the predicted values of the classifier. The training of the classifier Polynomial Regression with degree 3 was made with all the data available in dataset of training. The predicted values were obtained and validated using the testing dataset.

2.3.3 Comparative study of the relationship between the WHOIS Similarity Distance and the classifiers

The following comparatives show a relation between WHOIS Similarity Distance value and the values predicted by the classifiers. For these comparatives was used the training dataset. The y-axis represents the Global Distance or WHOIS Similarity Distance value and x-axis are allocated the size of the samples. In the figures were used some references to make easier to understand the comparisons:

- **The Coefficient of Determination** R^2, also denoted as r^2, is a number that indicates the proportion of the variance in the dependent variable that is predictable from the independent variable(s). It goes in the range of 0 to 1. When the value is close to 1, the linear model method predicts better results.

- **Mean** μ.

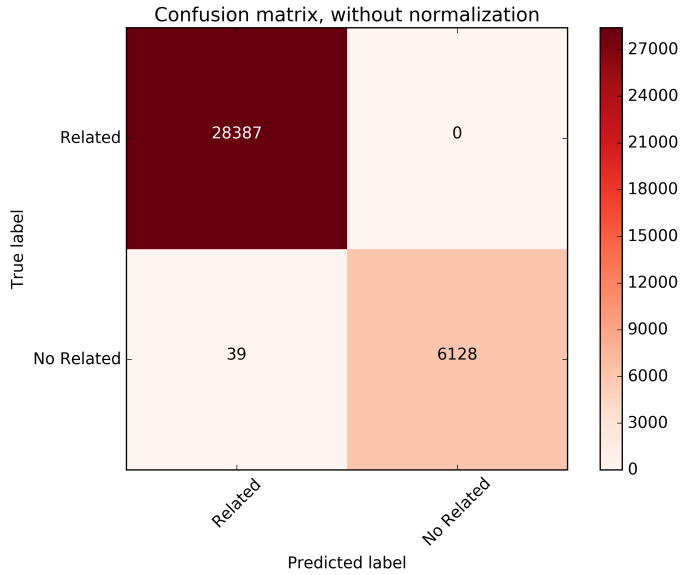


Figure 2.13: The confusion matrix shows us that around the 99% of the decisions made with respect to the relationship of the domains were correct. It was used the testing dataset and the linear regression with degree 3.

- **Median**.
- **Standard Deviation** σ
- $\times$, domains which were correctly related.
- $\cdot$, domains which were correctly unrelated.
- $\diamondsuit$, domains which were incorrectly related.
- $\blacktriangle$, domains which were incorrectly unrelated.

In the Figure 2.15 shows the results obtained using Linear Regression with respect the values of the GD. The value R^2 of this model is around 80.2%, does not improve with high significant when the numbers of samples increase. It could mean that the linear regression was performed with its maximal potential for this problem. Also, Linear Regression predicted in wrong way several comparisons. And as the size of samples increases the number of wrong predictions also increase.

In the Figure 2.16 shows a relation between the values of the GD obtained with a classifier polynomial regression with degree 2, the R^2 is around 97%, better than the R^2 obtained with the linear regression model. Also, the R^2 does not increase with respect the number of samples. The R^2 is an excellent value, and it says that the polynomial regression with degree 2 could suit our

problem. However, we can appreciate that the Polynomial Regression with degree 2 has some wrong predictions. Polynomial Regression with degree 2 has more wrong prediction than polynomial regression with degree 3; this can be appreciated in the Figure 2.16. Further the R^2 of the Polynomial Regression with degree 3 is around 99% and also it looks almost constant with respect the number of samples provided. Again, polynomial regression with degree 3 has better results than polynomial regression with degree 2.

The plots also show that the majority of the related domains have a numeric distance close to 0 and the majority of unrelated domains have a numeric distance greater than 40. That is, when the Global Distance of two domains, is closer to zero the probability to be related is higher.

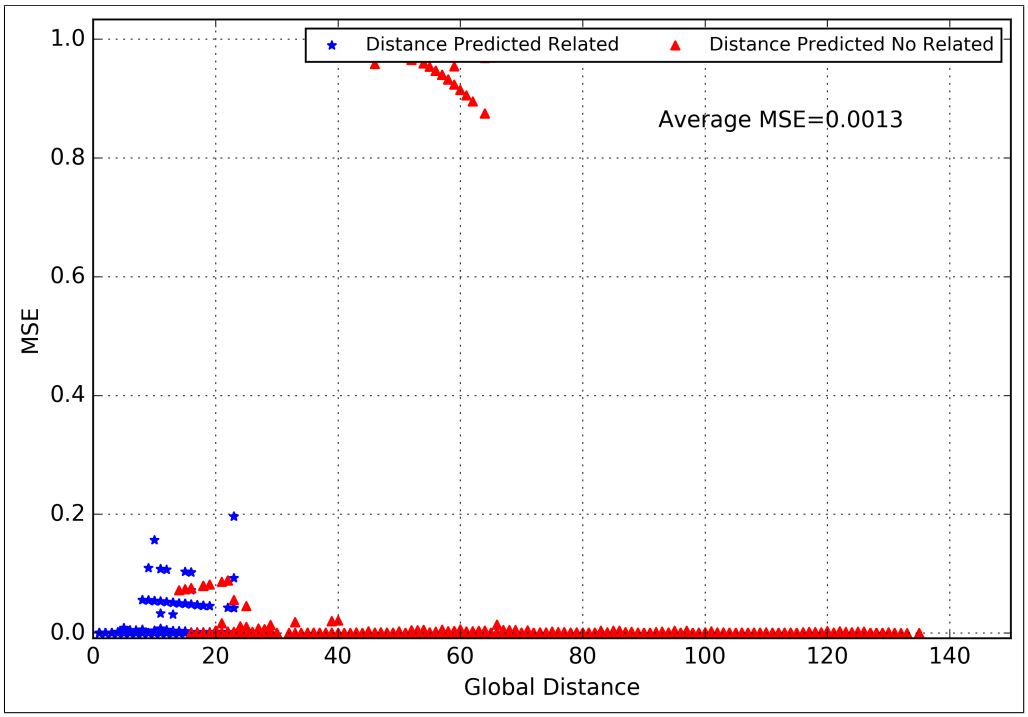
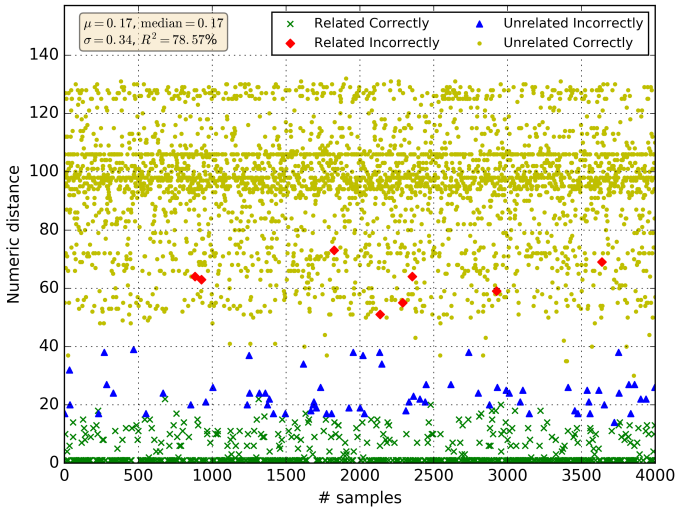
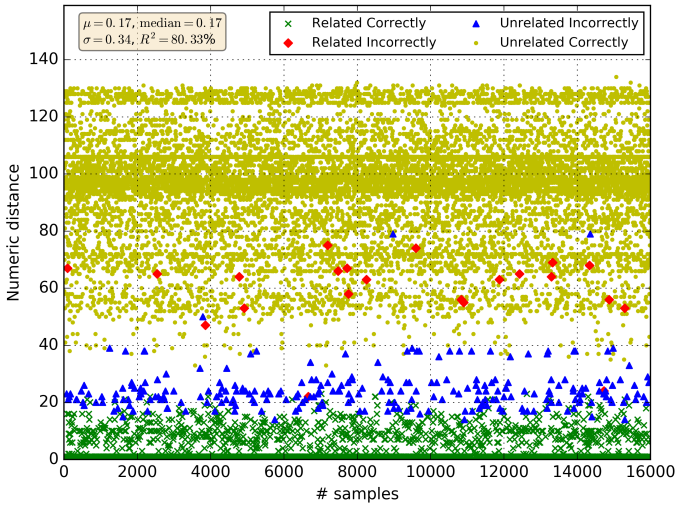


Figure 2.14: The MSE of the WHOIS Similarity Distance algorithm obtained using the testing dataset and the classifier of linear regression with degree 3.

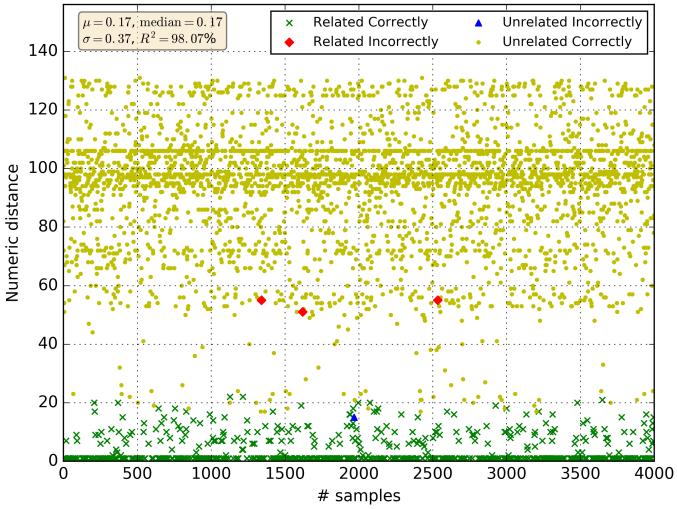


(a) Linear regression algorithm: with 20,000 samples. It has R^2 of 80% and an SE of 0.0063.

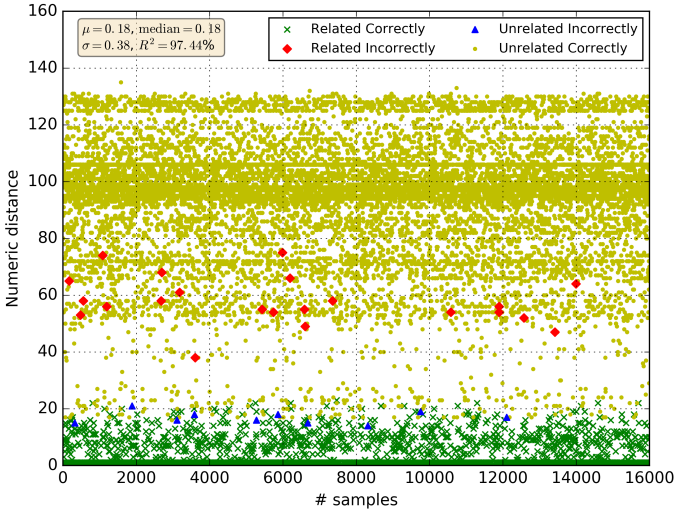


(b) Linear regression with 80,000 samples. It has R^2 of 80% and an SE of 0.0032.

Figure 2.15: Results of Linear Regression with a sample size of 20%. The results show the Linear Regression could not relate or unrelated correctly several domains. And as the size of samples increases the number of wrong predictions also increase. Also the plots show that the majority of the related domains have a numeric distance close to 0 and the unrelated domains have a numeric distance greater than 40.

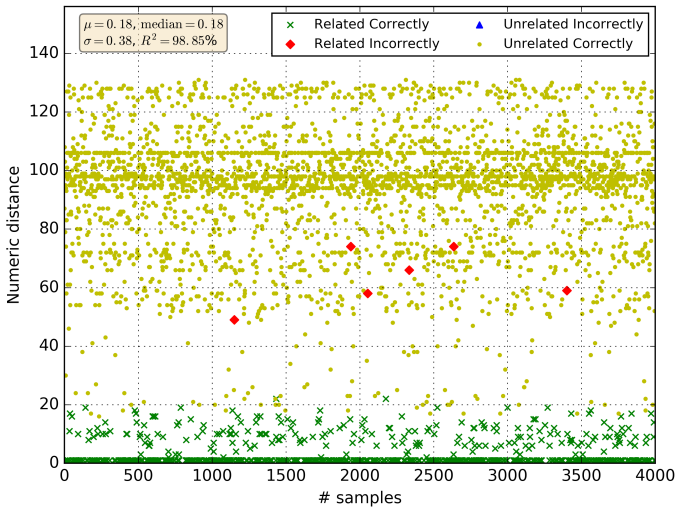


(a) Polynomial regression with degree 2 and 20,000 samples. It has R^2 of 97% and an SE of 0.0026.

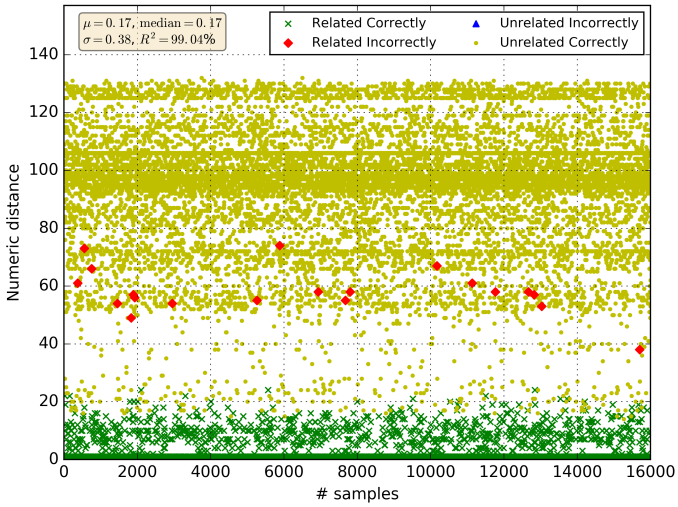


(b) Polynomial regression with degree 2 and 80,000 samples. It has R^2 of 97% and an SE of 0.0013.

Figure 2.16: Results of Polynomial Regression, degree 2 with a sample size of 20%. The numbers of wrong prediction is lower than Polynomial Regression with respect to Linear Regression.



(a) Polynomial regression with degree 3 and 20,000 samples. It has R^2 of 99% and an SE of 0.0013.



(b) Polynomial regression with degree 3 and 80,000 samples. It has R^2 of 99% and an SE of 0.0007.

Figure 2.17: Results of Polynomial Regression degree 3 with a sample size of 20%. The numbers of wrong prediction is lower than Polynomial Regression with degree 2.

Chapter 3

ManaTI Software

This Chapter describes the development of ManaTI.

ManaTI provides a GUI for visualizing weblogs and has several tools to increase the efficiency of the analyst analyzing weblogs files. ManaTI is highly scalable, giving the possibilities to develop easily more features. And also, it provides an API, so the users can create their modules and connect them to ManaTI without needing to understand many technical details of the system.

3.1 Description of functionalities

In this section, the most important features of ManaTI are described. How they work and how they were developed. All the features of ManaTI before to be developed, they were analyzed and decided meticulously in monthly meetings through all the development process, with the Cisco's analysts and the members of the ManaTI project.

Further, after each stable version of ManaTI deployed during the development process, was received feedback from the Cisco's analysts. This situation contributed to adapt the features and GUI of ManaTI according to the real needs of the analysts.

3.1.1 Table to visualize weblogs files and simple labeling of weblogs

This function allows to the user to upload the files and visualize them in the table. Where the user can filter or search in the rows by any text, regular expression or by labels.

Further, the table allows paginating the rows and choice the number of rows per page that you want to see. The Figure 3.2 shows an illustration of the dynamic table UI see. To achieve the dynamic table is used the jQuery plug-in Datatable.

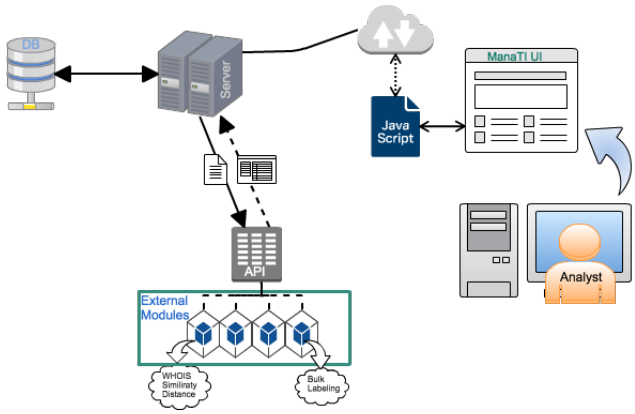


Figure 3.1: Workflow of the functions of ManaTI.

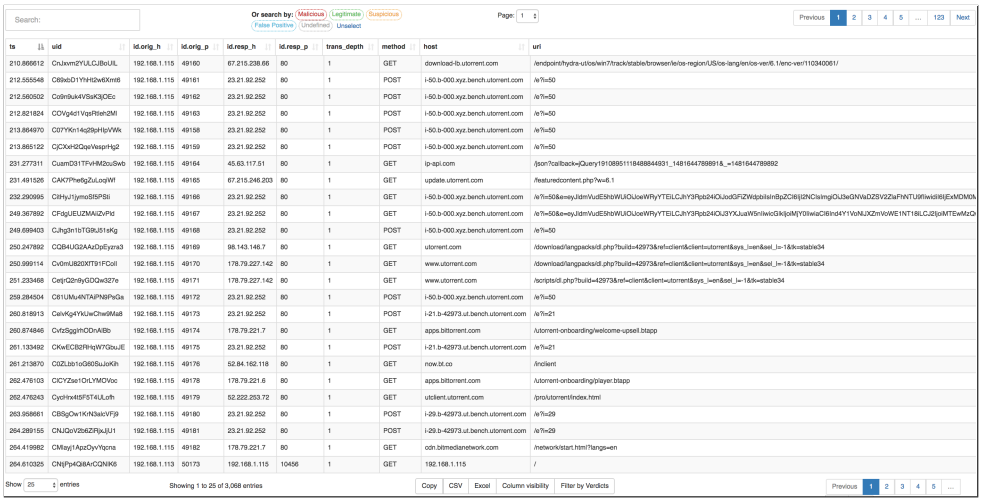


Figure 3.2: ManaTI table UI.

ManaTI supports Bro[15] weblog format and a custom format file used by Cisco, which looks like a CSV file. To parse the format files was is implemented a logic in the source file *reader_files.js*. For the implementation was necessary to uses several external libraries and source optimization to make the process to upload fast and efficient.

ManaTI was thought initially to work only with the Cisco custom format, but after a time the functionality to see and analyze the Bro HTTP files in ManaTI was added.

For labeling manually weblogs or simple labeling, the user has to select one or many weblogs, one after another, perform right click over it and then choose the label or verdict that the user believes suitable for the selected weblog. The

[15] www.bro.org

Figure 3.3 illustrates the process to label one or many weblogs manually in the table.

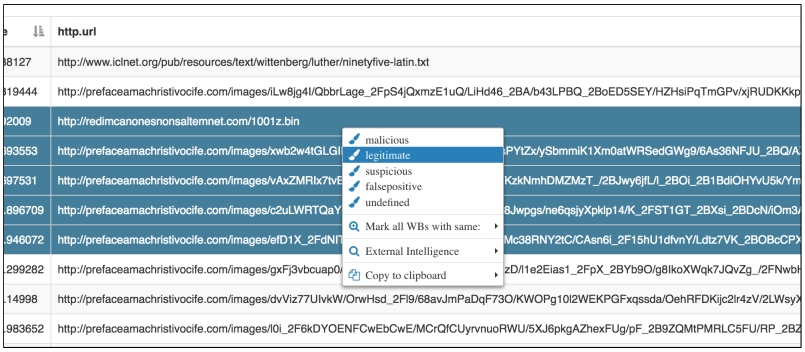


Figure 3.3: The images shows the process to label several weblogs and then apply a verdict. The rows with a darker background are weblogs selected. The rows with white background are weblogs have not been selected and with undefined verdict.

3.1.2 Exporting weblogs

ManaTI allows to the user to export the content of the dynamic table with the verdicts assigned in several formats like CSV, Excel files or copy in the clipboard. It is useful for the analyst to export the analysis in different formats and it can use them for many purpose or many other tools that the analyst uses.

3.1.3 Bulk labeling

The idea of Bulk Labeling is to facilitate the weblogs' labeling process made by the user.

Using the menu context shown when you perform right click, then "Mark all WBs with same:" and you can choose if "by IP" or "by Domain", also it indicates the column name of the file where it takes the information and the numbers of weblogs will be affected after applying the operation. In Figure 3.4 shows a screenshot of the necessary steps to perform a mass-tagged.

The implementation of this function was hard because was used techniques on JavaScript to do background processing or "multitasking", called Workers. In Figure 3.5 shows the implementation of the Bulk Labeling with the Workers.

3.1.4 Intelligence Tools

Often analysts check several websites or services with information about ranking of domains, WHOIS information, IPs, and others. Several sites provide reputation information of domains, IPs or file hash. The most common are

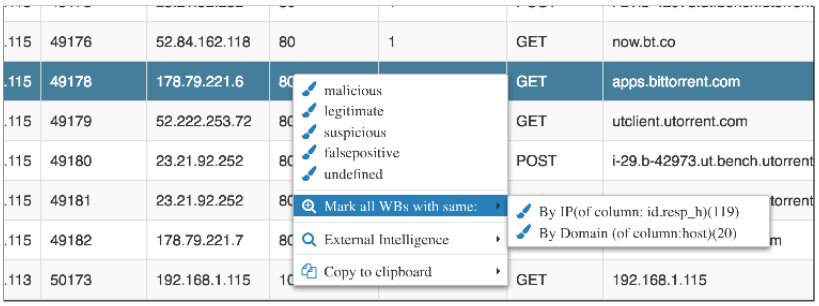


Figure 3.4: Option of BL in the menu context on ManaTI.

```
var setBulkVerdict_WORKER = function (verdict, flows_labelled){
    _dt.rows('.selected').nodes().to$().removeClass('selected');
    showLoading();
     var blob = new Blob([ "onmessage = function(e) { " +
        "var verdict = e.data[1];"+
        "var rows_data = e.data[2];"+
        "var col_dt_id = e.data[3];"+
        "var col_verdict = e.data[4];"+
        "var origin = e.data[5];"+
        "var col_reg_status = e.data[6];"+
        "var reg_status = e.data[7];"+
        "self.importScripts(origin+'/static/manati_ui/js/libs/underscore-min.js');"+
        "var flows_labelled = _.map(e.data[0],function(v,i){ return v.dt_id});"+
        "for(var i = 0; i< rows_data.length; i++) {"+
            "var row_dt_id = rows_data[i][col_dt_id]; "+
            "var index = flows_labelled.indexOf(row_dt_id); "+
            "if(index >=0){"+
               "rows_data[i][col_verdict] = verdict ;"+
               "rows_data[i][col_reg_status] = reg_status.modified ;"+
            "}"+
         "};" +
         "self.postMessage(rows_data)"+
    "}"]);
    var blobURL = window.URL.createObjectURL(blob);
    var worker = new Worker(blobURL);
    worker.addEventListener('message', function(e) {
        var rows_data = e.data;
        var current_page = _dt.page.info().page;
        _dt.clear().rows.add(rows_data).draw();
        _dt.page(current_page).draw('page');
        hideLoading();
    });
    var rows_data = _dt.rows().data().toArray();
    worker.postMessage([flows_labelled,verdict,rows_data,
        COLUMN_DT_ID, COLUMN_VERDICT,document.location.origin, COLUMN_REG_STATUS, REG_STATUS]);
};
```

Figure 3.5: Implementation of Bulk Labeling using Workers in JavasScript.

VirusTotal[16], PassiveTotal[17], Metadefender[18] and others. And also like was explained before the analysts contently consult the WHOIS information of the domains.

Because that, in the menu context was added an option to get information from VirusTotal (using its API) and also the user has the possibility to consult the WHOIS information. In both cases, the user can choose how to do the consultation using domain name or IP.

The Figure 3.6 shows an example of how looks the information obtained from VirusTotal using the provided IP and the Figure 3.8 shows another examples but providing a domain name. Once the user selects the way how to

[16] www.virustotal.com/
[17] www.passivetotal.org
[18] www.metadefender.com

make the request, by IP or by Domain, an internal window or modal (also called pop-in) is displayed where the user can see the requested information. The response can take time if the query was not performed before and stored in the ManaTI database.

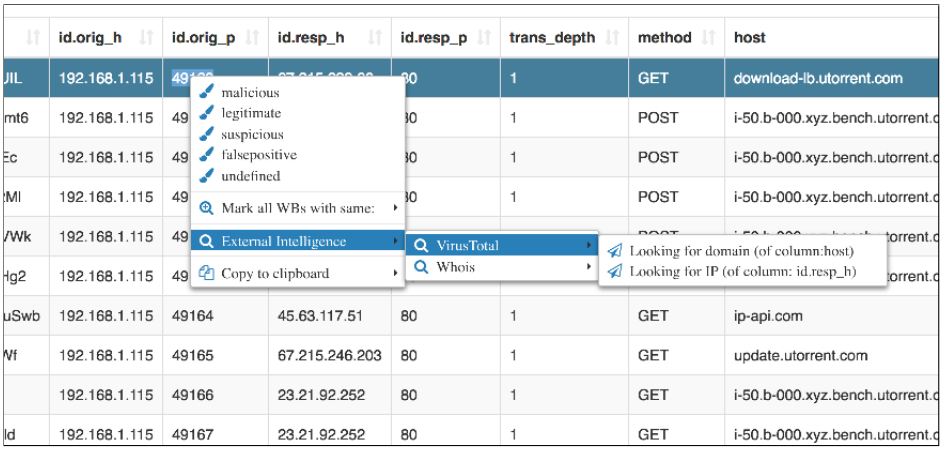


Figure 3.6: Menucontext option to get external information.

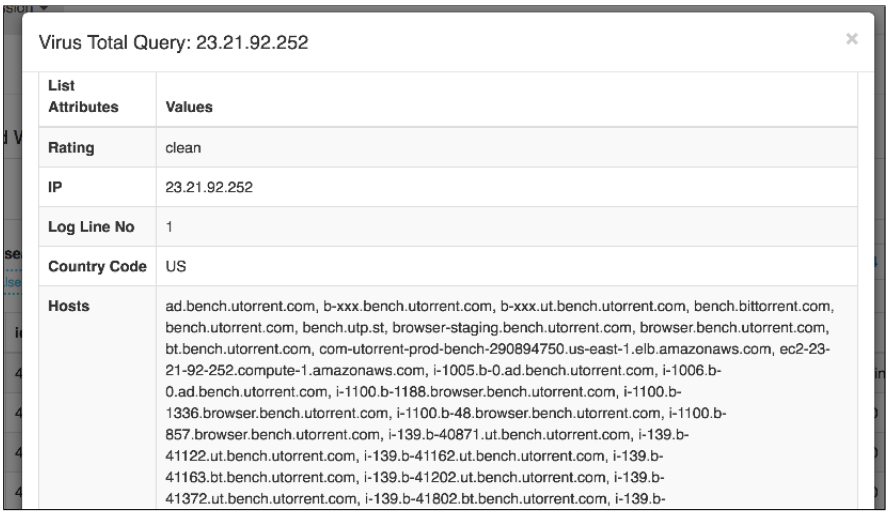


Figure 3.7: Modal showing the information returned by VirusTotal given an IP address.

All the queries performed to get information from VirusTotal or WHOIS data by domains or IPs, are storage in the database. The goal is to have a history changes for the future and also, for avoiding duplicated requests, improving in that way the time to show information in the modal.

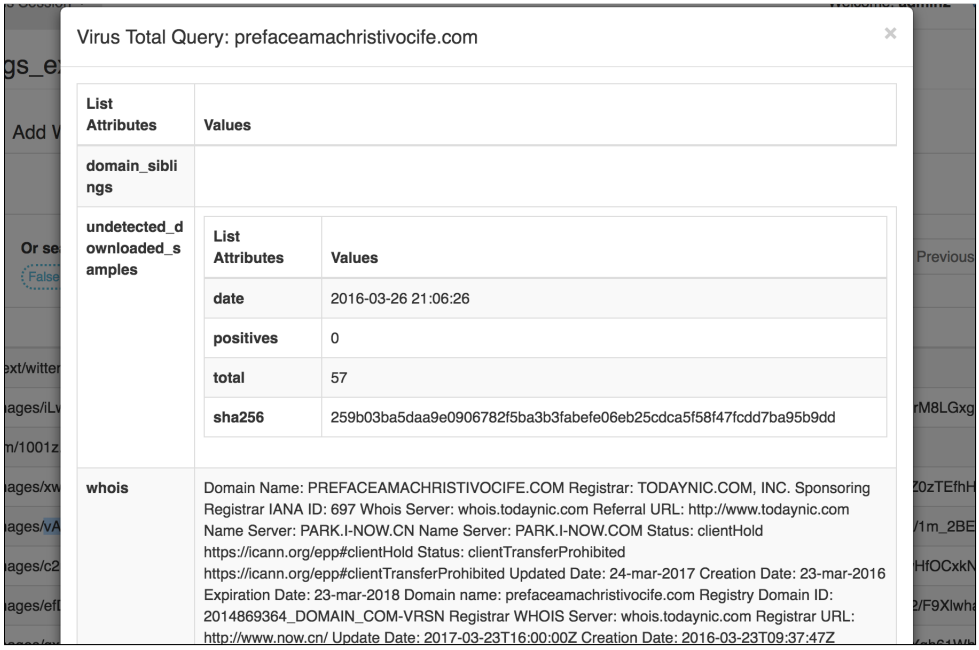


Figure 3.8: Modal showing the information returned by VirusTotal given a Domain Name .

3.1.5 Statistics Section

When the analysts are reviewing some weblog file looking for malicious or suspicious weblog, is common that they need statistics information, e.g. how many times some domain or IP is used in the whole file, therefore is developed a section on the web page to put that kind of information. The information provided is the number of domains/IP and how many time they appears in the weblogs file analyzed. The Figure 3.9 illustrates the statistics area.

Often the weblogs files have more than 1000 lines (or rows) so can be time-consuming to the JavasScript analyses the whole data looking for statistics information. Thus the user experience could be affected. To do this totally transparent for the user was implemented a *FlowsProcessed*() class in the *struct_helper.js* file and the class was instanced using Workers. The Figure 3.10 shows the implementation of the logic for processing the weblogs and displaying statistics area. For seeing the statistics section is not necessary to have the weblogs file stored in the database.

3.1.6 Comments

Exist the possibility to do comments per analysis sessions, so the users can add notes about the weblogs analyzed. This features is important because for example the analysts can notify to another analysts about possible malware

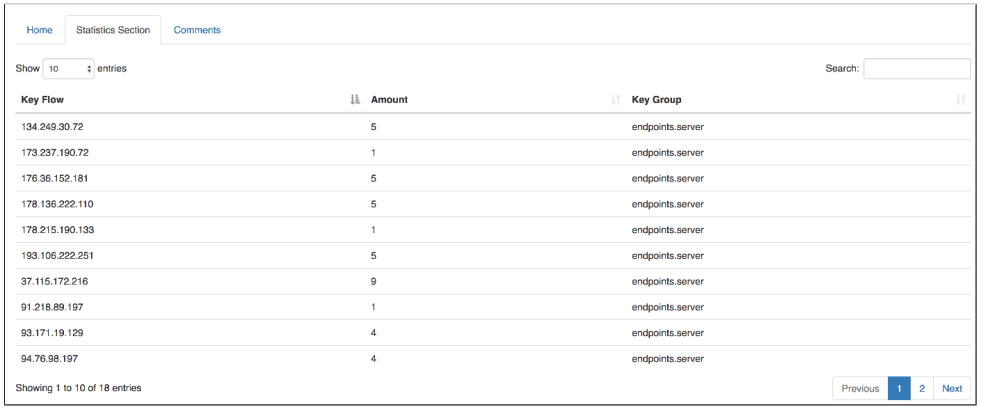


Figure 3.9: Statistics section view. It is showing how many times an IP or domain appears in the weblogs file and in which column.

```
var processingFlows_WORKER = function (flows) {
     $("#statical-section").html('');
    _flows_grouped = {};
    var blob = new Blob([ "onmessage = function(e) { " +
        "var flows = e.data[1];"+
        "var flows_grouped = e.data[0];"+
        "var origin = e.data[2];"+
        "self.importScripts(origin+'/static/manati_ui/js/libs/underscore-min.js');"+
        "self.importScripts(origin+'/static/manati_ui/js/struct_helper.js');"+
        "var helper = new FlowsProcessed(flows_grouped);"+
        "for(var i = 0; i< flows.length; i++) helper.addFlows(flows[i]);"+
        "self.postMessage(helper.getFlowsGrouped());" +
    "}"]);

    // Obtain a blob URL reference to our worker 'file'.
    var blobURL = window.URL.createObjectURL(blob);

    var worker = new Worker(blobURL);
    worker.addEventListener('message', function(e) {
        worker.terminate();
        _flows_grouped = e.data;
        _helper = new FlowsProcessed(_flows_grouped);
        _helper.makeStaticalSection();
        console.log("Worker Done");
    });
    worker.postMessage([_flows_grouped,flows,document.location.origin]);

};
```

Figure 3.10: Implementation when the class *FlowsProcessed*() is instanced using Workers.

activity detected in the file or suspicious domains found. The Figure 3.11 shows the UI to do comments in the analysis sessions.

3.1.7 Anonymous Users and Sessions shared

ManaTI is a web application and has a module of user authentication, roles and permissions to control the user activity inside the system. So the users to get access to the system need corresponding credentials, username and password. However, it is also possible that some users wants to show some analysis to someone without access to the system to get some feedback.

Thanks to the UI of ManaTI to read and interpret weblogs files is easier than only watching them in a plain text. Hence is implemented the possibility

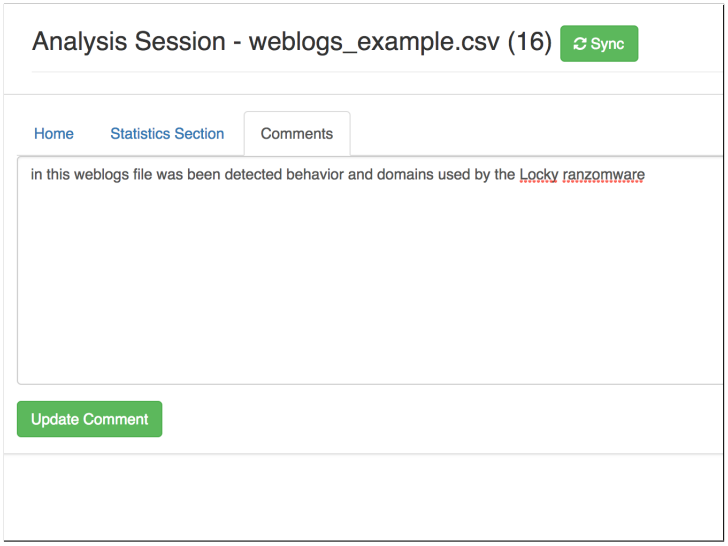


Figure 3.11: Text area to provide comments in the analysis session UI.

to post analysis sessions and make them public. Thus, users without access to the system could see the analysis session posted and work with them without affecting the database of the web application. They are called Anonymous Users (AU) .

Further, the AU can export the analysis session with the labels. Also, the Anonymous Users can upload new files without the possibility to save them in the database. The External Modules can still help the AU to find malicious weblogs or improve its labeling process.

3.1.8 History of changes

Often the analysts cannot remember all the decision taken in all the weblogs analyzed. In ManaTI is developed a function to see all the changes made to a particular weblog. The Figure 3.12 demonstrates how to look the history of changes modal of one selected weblog. The function to show the history of changes is located in the menu context.

To store all the changes of one weblog, made by users or External Modules, is implemented a model *WeblogHistory*. If the reader wants to know more details about the database model of ManaTI, Could go to the Appendix C.

Every time that the user changes the verdict (or label) of one weblog, one *WeblogHistory* is created. By default all the weblogs have the verdict undefined assigned.

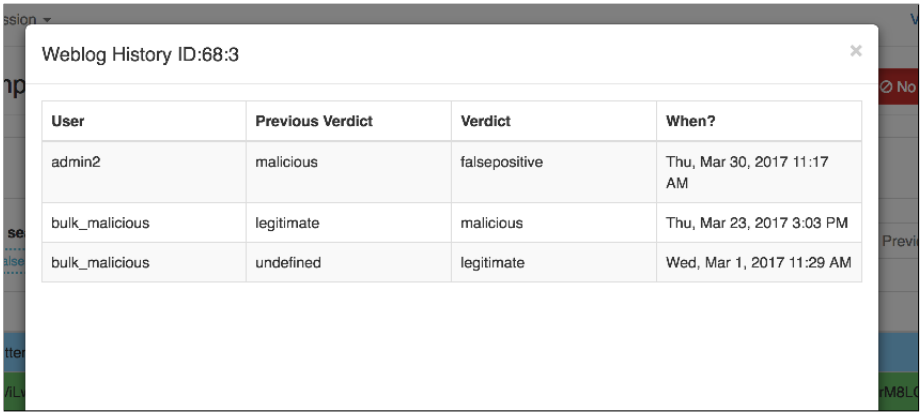


Figure 3.12: History of changes of a weblog. It is possible to see in the modal, the users or modules that have labeled the weblog during the time.

3.1.9 External Modules

The goals of the external modules are to help the analyst to detect and label weblogs faster. So was developed an API where the analyst can use it for creating its Python scripts.

With the external modules, the user could be able to do heavy processing using a significant amount of data. Therefore, all the modules are running in background tasks (multi-threading). The modules' processing is not affecting the user experience and the web server performance. For the implementation was necessary to create a Django application, and the API implementation, abstract class and a database model for external modules.

One external module to bulk labeling by domain was developed and called *Bulk Labeling*. The module works with the next logic: when the user labels one weblog (or many), the module will use the domain name of the labeled weblog, and it will search for all the weblogs in the database with the same domain name assigned, and tag them with the same verdict of initial weblog.

3.1.10 WHOIS Similarity Distance Module

The WDS algorithm explained before in the Chapter 2 was implemented as an external module in ManaTI.

It allows to the user to select one specified domain and search for all the related domains inside the file. Also, there is a function to label all the connected domains that have being found. Both functions described are in the background and totally transparent to the user. This feature is allocated in the menu context.

3.1.11 Metrics

For measuring the performance of web application, is implemented a logic to get data from the front-end (GUI) and save them in the database. The data is only used to measure if ManaTI is useful for the analysts or not.

The logic in the front-end to record the events produced by the user in the GUI is located in the class *Metrics*, which is implemented in the file *metrics_logic.js*.

The *Metrics* class has methods that are called when some events occur in the GUI, e.g. the user labels a weblog, uploads some file or others events. The register of an event is temporally stored in a local database inside the browser called *localStorage*. Each minute the synchronization process starts and the data collected is sent to the web server and saved in the database of ManaTI.

3.2 Software Development Methodology

For the development of ManaTI, was used the Kanban methodology [35], specifically the Kanban cards, through the web tool *MeisterTask*[19]. Kanban is an agile work methodology that allows adapting quickly to changes that arise when developing a project.

During the 40s, Toyota created a better engineering process for the supermarket. The grocers' "just-in-time" delivery process did that the Toyota engineers think about their methods to control the inventory. In simplest words, *"by better communication through visual management"* [36]. Kanban is Japanese word for "visual signal" or "card." Toyota workers used a *kanban* to mark steps in their development process [36] [35].

Kanban Principles principles

- **Guaranteed Quality.** Everything done must go well at first, there is no margin of error. Hence in Kanban does not reward the speed, but the final quality of the tasks performed. It is based on the fact that it often costs more to fix it than to do it right the first time.
- **Reduction of waste.** Kanban is based on doing only what is right and necessary, but doing it well. This supposes the reduction of everything that is superficial or secondary (principle YAGNI)
- **Continuous improvement.** Kanban is not merely a management method, but also a system of improvement in the development of projects, according to the objectives to be achieved.
- **Flexibility.** The next thing to do is to decide the backlog (or accumulated pending tasks), being able to prioritize those incoming tasks

[19] www.meistertask.com

according to the needs of the moment (ability to respond to unforeseen tasks [37].

3.3 Software Resources used

Detail of all the resources software required to develop ManaTI.

- **Programming Language Python**. It is a high-level programming language created by Guido van Rossum. Python is an interpreted language and has a philosophy which emphasizes code readability. It has very simple syntax, easily scalable, is very fast and flexible. Python is free and open source and is widely used in educational environments around the world [38] [39]. There are a lot of documentation and a great support of the community. ManaTI was developed with Python version 2.7.

- **Web Application Framework Django**. It is an open-source web framework made in (and for) Python, Django follows the mode-view-template (or MVT) architectural pattern and is maintained by an independent organization called Django Software Foundation (DSF). Django also follows the Python's philosophy, trying to be simple, scalable, rapid development and respects the principle of "don't repeat yourself" or just DRY[20]. The figure 3.13 shows the summary of the workflow of Django [40].

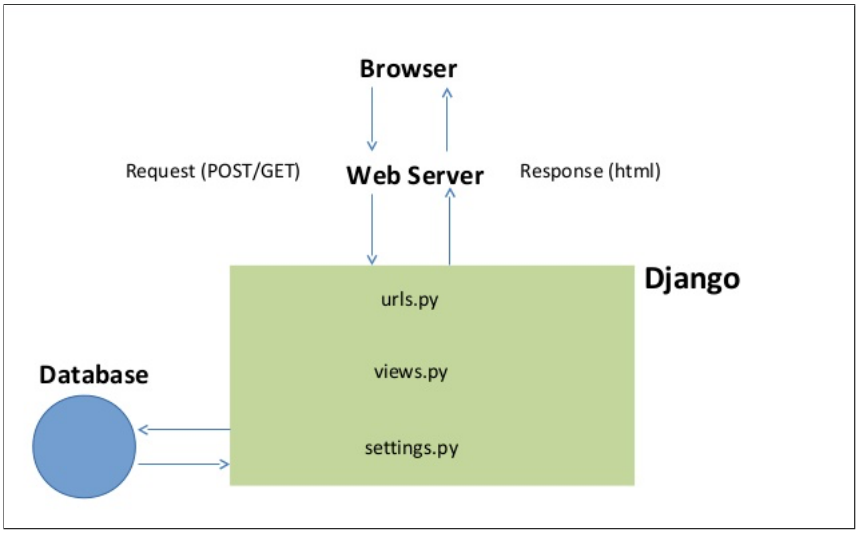


Figure 3.13: Illustration of the Workflow of Django.

[20] www.djangoproject.com

- **Database Engine PostgresSQL**. PostgreSQL[21] is an object-relational database management system, distributed under the BSD license and freely available source code. It is one of the most powerful open source database management system on the market and in its latest versions has nothing to envy to other commercial databases.

  PostgreSQL uses a client-server model and uses multiprocessing instead of multithreading to ensure system stability. A failure in one of the processes will not affect the rest and the system will continue to function.

  The system was developed using this database engine because it is integrated in an easy way to the Django framework allowing to make operations to the database from the business layer without having to implement business logic in the database.

- **JavaScript and external libraries**. It is a high-level, dynamic, untyped, and interpreted programming language and used is one of the three core technologies of in World Wide Web (WWW). It has been standardized in the ECMAScript language specification [41].

  One of the most important aspects of ManaTI is its User-Interface (UI), to make it as friendly as possible was used several JavaScript (JS) libraries and others implementations. Two of the most used JS libraries in this project were **jQuery**[22] and **DataTable**[23] .

  jQuery is a cross-platform JavaScript library designed to simplify the client-side scripting of HTML.

  DataTable "is a plug-in for the jQuery JavaScript library. It is a highly flexible tool, based upon the foundations of progressive enhancement, and will add advanced interaction controls to any HTML table".

- **Compatible browsers**. ManaTI is only compatible with Chrome version 56+.

- **Operative System**. ManaTI was developed in macOS, but also it is compatible with any Linux distro based on Debian with Python version 2.7 installed.

- **VCS Git and repository**. Git is a reliable, versatile and multipurpose Version Control System (VCS). It was designed by Linus Torvalds, thinking about the efficiency and reliability of the maintenance of versions of applications when they have a large number of source files. It is free and open-source[24].

[21] www.postgresql.org
[22] https://jquery.com
[23] https://datatables.net
[24] https://git-scm.com

The ManaTI source code was storage in Bitbucket[25]. It is a web-based hosting service for projects using the Mercurial and Git revision control system. Bitbucket offers free and commercial plans.

- **IDE PyCharm**. It is an Integrated Development Environment (IDE) used in programming, specifically for the Python language. It provides code analysis, a graphical debugger, an integrated unit testing, integration with version control systems (VCS), and supports web development with Django. It is developed by JetBrains [42].

- **Task Manager MeisterTask**. It is an intuitive online task manager that uses smart integrations and task automation to make your team more productive. MeisterTask[26] uses cards based on Kanban methodology. The Figure 3.14 illustrates how the dashboard on ManaTI in MeiterTask looks.

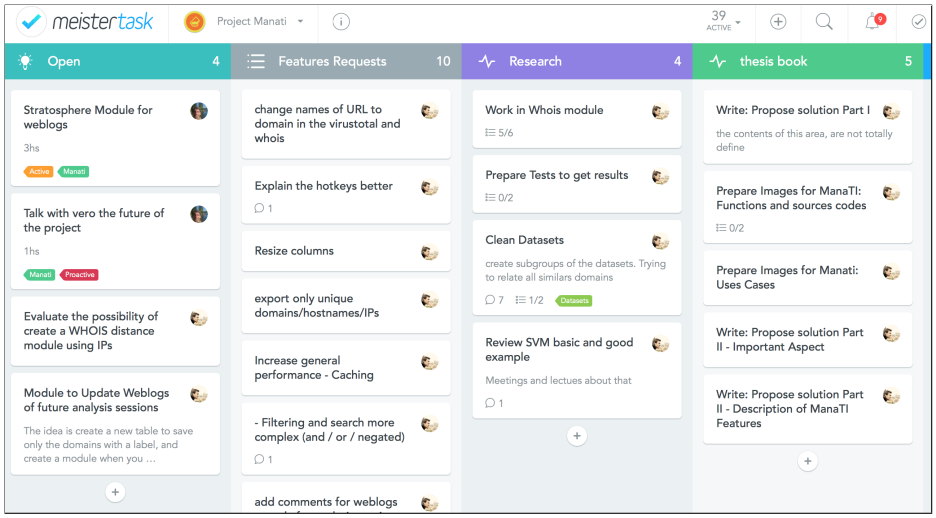


Figure 3.14: Dashboard of ManaTI project in MeisterTask.

3.4 Experiments and discussion

The idea of the test is proved if ManaTI assists appropriately to the analysts or not. The experiment was planned as following:

- First, were selected, six people. Three of them are professional threat analyst and 3 are IT professional working in the area of web development.

[25] www.bitbucket.com
[26] www.meistertask.com

- Second, were chosen eight weblogs file, provided by Stratosphere project [27]. All the files are traffic captured of well-known malware. Each file had between 2,000 and 3,000 rows (weblogs) to analyze.

- Then, were created two instances of ManaTI called instances A and B. They were hosted on a server that belongs to Stratosphere Lab.

  - Instance A: It had the last developed version of ManaTI, with all the function listed before. The WHOIS Distance module was unable for the experiments because was detected a performance problem. This issue could affect the resources of the server and consequently, others services deployed on the server.
  - Instance B: All the analyst tools and features developed in ManaTI were disconnected, with exception to the dynamic table to visualize weblogs files and the possibility to assign verdict.

- The users were provided with proper credentials to both instances.

- Each user had to analyze four weblogs files in one instance and others four files in another one. With anticipation, the weblogs files were determined to be used in one or another instance.

- The three users non-professional threat analyst were introduced in the topic with an explanation video and manuals about how to use ManaTI and how to analyze weblogs. Also, they were practicing with ManaTI one week before to start the experiments. To understand the functionalities of ManaTI and how they work. The others three users, professionals threat analysts, already had used ManaTI and also, they have been participating in the whole process of ManaTI development.

- Thence, the users had two weeks to finish the experiment. A button to close analyzing session was implemented, so the users could determine when the analysis of one file was done.

- When all the users finished their reviews, the performance metrics collected were extracted from the database to be analyzed.

3.4.1 Results

The performance metrics were processed and the results are in the table 3.1.

References to the table 3.1:

Av. # of labels/ # weblogs It is the mean number of labeled in total over the average of the number of weblogs to be labeled per analysis.

[27] https://stratosphereips.org/category/dataset.html

Instance	Av. # of labels / # weblogs	Av. labels/sec	Av. labels/min
A	1304.34/3339.83	17.832	31.334
B	75.25/2518.75	6.779	9.354

Table 3.1: Results obtained of the experiments performed in ManaTI. The user in average are more productive using the instance A then using the instance B. The instance A had all the tools implemented in ManaTI working, although the instance B only had the dynamic table working.

Av. labels/sec It is the average number of labeled per seconds. Only moments with reported activities were used. Moments, where the user was doing nothing, were discarded.

Av. labels/min It is the average number of labeled per minutes. Again only were taken moments with reported activities.

The first column, the number of labeled in the instance A are bigger than the same column in the instance B. This situation is due to two condition. First, the users after a time analyzing in the instance B did not want to continues or to spend more time in that. So they just labeled representatives weblogs and then decide to close the analysis. And second, the instance A had several tools for mass-tagged (bulk labeling). They made a big different in the statistics.

The second column, in the instance A the average number of labeled per seconds by the users is higher than the results obtained in the instance B. This situation is also repeated in the last column, where the labels are evaluated per minutes. Similar to the first column, the tools for bulking labeling made a difference at the moment to estimate the number of verdicts applied per unit time. Using the data obtained the labeling process in the instance A were faster than the instance B by a factor 3.4 times.

Conclusion

In this thesis we researched, created and published a new tool, called the ManaTI, to assist the network security analysts to find threats in the network. It has two primary goals: First, to assist the analysts in evaluating the network traffic to find better and process the information. Second, to research a machine learning method that can confidently identify domains which WHOIS information is related. Our algorithm is both a WHOIS classification tool of similar domains or as a WHOIS similarity distance.

For the experiments with the WHOIS similarity distance, we worked with a dataset of WHOIS records made by us with approximately 1,300 different domains. They were hand labeled by experts in the area based on a manual inspection. We conclude that this manual labeling was paramount for training the algorithms with precise data. We also find that our use of the PassiveTotal library proved to be most effective to get the WHOIS records.

For the experiments to evaluate ManaTI, we asked real network analysts to evaluate real malware weblogs obtained from the Stratosphere project. We concluded that this combination of analysts and malware contributed to reproduce the most realistic environments to work on. The feedback received by the professional threat analysts after the experiment was valuable. With it, we found some weaknesses of ManaTI that are currently under improvement. Our general conclusions about the techniques are that:

- The WHOIS information for domains registered for the same purpose are not entirely similar, but in most cases, they share enough information to be measurable by our distance metric. Also, we found it very common for entities to hire third-party companies to register their domains to maintain certain privacy in the WHOIS records. These cases were the most difficult ones to classify.

- There are WHOIS information fields which are more important to relate domains than others fields.

- The 3rd-degree Polynomial Regression algorithm was the most suitable for our WHOIS Similarity Distance method.
- The accuracy of the WHOIS similarity distance algorithm is around 98%.
- ManaTI can increase the speed of the security analysts in studying unknown traffic by a factor of 3.4.

In summary, ManaTI is a useful tool for the threats analysts that can help them speed up the finding of more and better information in web network traffic. It is still a prototype, but we got good results in real environments. Moreover, it is used actively by the Cisco company. Further, the ManaTI project is supported by Cisco in the Stratosphere Lab[28] was renovated one year more. It reflects the confidence of the Cisco company with the work made in ManaTI. Despite our experiments, designs, and datasets, ManaTI is an evolving tool, and it needs to be improved. This is because Cisco Systems is currently using it for real analysis, and therefore there are more features requests to implement, especially in the area of machine learning. Our future work can be divided in:

1. To implement active learning techniques on ManaTI by learning from the analysts. Since ManaTI remembers how the users used the system, it is possible to model this behavior and teach ManaTI how to work better.
2. To show the users, the statistics graphs of their performance in real time, to help them measure their progress in real time.
3. To develop a new WHOIS Similarity Distance algorithm using others techniques, a new classifier like SVM or applying Neural Networks (NN).
4. To implement a module for detecting malicious domains using some of the techniques already studied.
5. To use more and different domains in the training of the classifier and distance measure.
6. To better help the analysts by incorporating the new scenarios of analysis used by Cisco's researchers.

[28]Project Reference: 13141/830-8301351C009, AIC Group, Department of Computer Science, CTU University

Bibliography

[1] Kolbitsch, C. Effective and Efficient Malware Detection at the End Host.

[2] TechTerms. Malware. Available from: `http://techterms.com`

[3] *Botnets: The Killer Web App*. Syngress, 2007.

[4] Institute, A.-T. Malware Statistics.

[5] Takumi Yamamoto, S. S., Kiyoto Kawauchi. Proposal of a method detecting malicious processes. *IEEE*, 2014.

[6] Shuang Hao, R. P., Nick Feamster. Monitoring the Initial DNS Behavior of Malicious Domains. *ACM*, 2011.

[7] Firdausi, I. Analysis of Machine learning Techniques Used in Behavior-Based Malware Detection. *IEEE*, 2010: pp. 201–203.

[8] Project, S. Reference Number of Project ManaTI in FELK, 13141/830-8301351C009.

[9] Gasser, M. *Building a Secure Computer System*. Van Nostrand Reinhold, 1988.

[10] Moir, R. Defining Malware. October 2003. Available from: `https://technet.microsoft.com`

[11] Fielding, R. *Hypertext Transfer Protocol – HTTP/1.1 - RFC2616*. The Internet Society, 1999.

[12] Garcia, S. Capture Malware Traffic. 2017. Available from: `https://stratosphereips.org/`

[13] Mockapetris, P. *Domain Names - Concepts and Facilities [RFC 1034]*. Network Working Group.

[14] Mockapetris, P. *Domain Names - Implementation and Specification [RFC1035]*. The Internet Society.

[15] Oxford. Domain Name. Available from: `https://en.oxforddictionaries.com`

[16] ComputerHope. DNS. April 2017. Available from: `http://www.computerhope.com`

[17] Daigle, L. *WHOIS Protocol Specification [RFC 3912]*. The Internet Society, September 2009.

[18] ICANN. About WHOIS. Available from: `https://whois.icann.org/en/about-whois`

[19] Bilge, L. EXPOSURE: A Passive DNS Analysis Service to Detect and Report Malicious Domains. *ACM Trans. Inf. Syst. Secur.*, April 2014: p. 28.

[20] Masahiro Kuyama, Y. K.; Sasaki, R. Method for Detecting a Malicious Domain by using WHOIS and DNS features. *SDIWC*, 2016.

[21] Kalyan Veeramachaneni, V. K., Ignacio Arnaldo. AIˆ2: Training a big data machine to defend. *IEEE*, 2016.

[22] DomainTools. The DomainTools Report Distribution Malicious Domain. Technical report, DomainTools, 2016.

[23] Letal, V. *Discovering of malicious domains using WHOIS database*. Master's thesis, Czech Technical University in Prague - Faculty of Electrical Engineering, 2015.

[24] Hao, S. PREDATOR: Proactive Recognition and Elimination of Domain Abuse at Time-Of-Registration. *ACM*, 2016.

[25] Symantec. Ransom.Cryptowall. June 2014. Available from: `https://www.symantec.com`

[26] Symantec. Ransom.Locky. February 2016. Available from: `https://www.symantec.com`

[27] Symantec. Major TeslaCrypt ransomware offensive underway. December 2015. Available from: `https://www.symantec.com`

[28] Kaspersky. What is TorrentLocker? Available from: `https://www.kaspersky.com`

[29] Hunter, J. D. Matplotlib: A 2D graphics environment. *Computing In Science & Engineering*, volume 9, no. 3, 2007: pp. 90–95, doi:10.1109/MCSE.2007.55.

[30] McClenathan, M. Angles and Non-Right Triangles.

[31] Levenshtein, V. I. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 1965.

[32] Pedregosa, F.; Varoquaux, G.; Gramfort, A.; et al. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, volume 12, 2011: pp. 2825–2830.

[33] Sebastián Basterrech, G. R. Real-Time Estimation of Speech Quality through the Internet Using Echo State Networks. *Journal of Advances in Computer Network*, volume 1, no. 3, September 2013.

[34] Trevor Hastie, J. F., Robert Tibshirani. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, second edition edition, 2009.

[35] Y. Sugimori, F. C. . S. U., K. Kusunoki. Toyota production system and Kanban system Materialization of just-in-time and respect-for-human system. *International Journal of Production Research*, 1977.

[36] Leankit. What is Kanban? March 2017. Available from: `https://leankit.com`

[37] Wester, J. What is Kanban? 2017. Available from: `http://www.everydaykanban.com`

[38] Programming, P. Python (programming language). Available from: `https://en.wikibooks.org/wiki/Python_Programming`

[39] Chun, W. J. *Core Python Programming*. Prentice Hall PTR, 2001.

[40] rajkumar2011. Getting started with django 1.8. Available from: `https://www.slideshare.net/rajkumar2011/`

[41] Flanagan, D. *JavaScript, the Definitive Guide*. O'Really, 2011.

[42] eWeek, D. K. T. JetBrains Strikes Python Developers with PyCharm 1.0 IDE. October 2010. Available from: `http://www.eweek.com`

Appendix A

Abbreviations

AI Artificially Intelligent.

API Application Program Interface.

AS Autonomous Systems.

AU Anonymous Users.

C&C Command and Control.

ccTLDs country code Top-Level Domains.

DNS Domain Name Service.

DoS Denial-of-Service.

DSF Django Software Foundation.

FPR False Positive Rate.

gTLDs generic Top-Level Domains.

GUI Graphic Users-Interface.

HTTP Hyper Text Transfer Protocol.

HTTPS Hyper Text Transfer Protocol Secure.

ICANN Internet Corporation for Assigned Names and Numbers.

IDE Integrated Development Environment.

IP Internet Protocol.

IPS Intrusion Prevention System.

IT Information Technology.

JS JavaScript.

ML Machine Learning.

MSE Mean Squared Error.

MVT Mode-View-Template.

NN Neural Networks.

PCAP Packet Capture.

ROC Receiver Operating Characteristic.

SML Supervised Machine Learning.

SVM Support Vector Machine.

TLDs Top-Level Domains.

TPR True Positive Rate.

UI User-Interface.

URL Uniform Resource Locator.

VCS Version Control System.

WWW World Wide Web.

Appendix B

Contents of CD

- `readme.txt` ..... the file with CD contents description
- `manati_project` ..... ManaTI Web Application source code
  - `api_manager` Django app. to provide the API for the external modules
  - `examples_weblogs` ..... Examples of files compatible with ManaTI
  - `login` .... Django app. for user authentication, roles and permissions
  - `logs` ..... Directory of logs files generated in the ManaTI
  - `manati` ..... Main directory of Django project, it has files settings
  - `manati_ui` ..... Django app. with the source for the ManaTI UI
  - `share_modules` ..... Implementations to be shared in all the Django applications inside ManaTI
  - `static` ..... Javascript and CSS files shared between the Django applications
  - `templates` ..... Views or HTML files shared between the Django applications
  - `requirements.txt` ..... Python libraries used in ManaTI
  - `manage.py` ... Django file to create applications, changes the database, run the Django server, and many other task
  - `README.txt` . contains a description about ManaTI and instructions to install it
- `manati_experiments` ..... The instances and weblogs files used for the experiments on ManaTI
- `WSD` . Experiments, dataset, source code and others assets used to develop the method WHOIS Similarity Distances
  - `dataset` The dataset used for WHOIS Similarity Distance experiments
  - `experiments_notes` ..... Source code used for the WSD experiments
  - `source` ..... The source of the WHOIS Similarity Distance algorithm
- `thesis_book` ..... The directory of LaTeX source codes of the thesis

Appendix C

Database Model

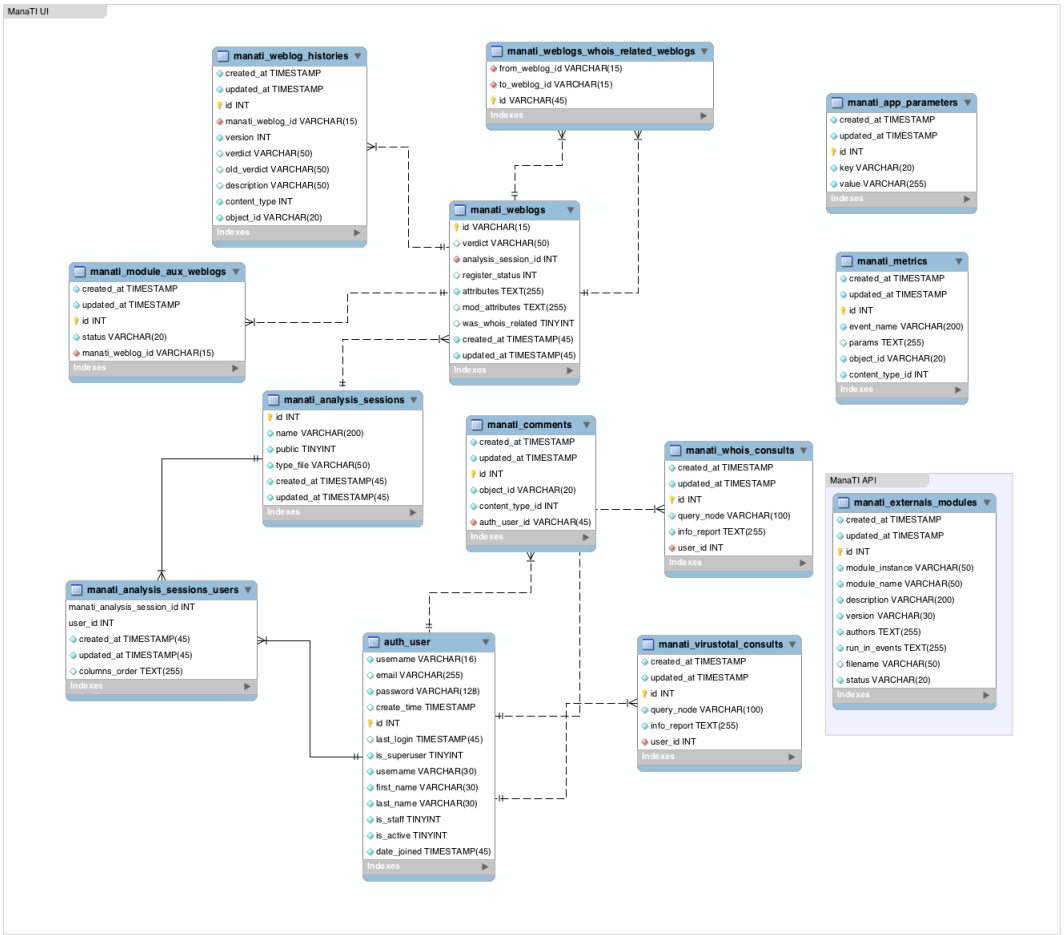


Figure C.1: Database Model of ManaTI.