



ENhancing seCurity and privAcy in the Social wEb: a user-centered approach for the protection of minors



WP4 – User Profiling for Detection and Prediction of Malicious Online Behavior

Deliverable D4.3 “Implementation of browser add-on that detects distressed or aggressively behaving users”

Editor(s):	Athena Vakali (AUTH), Vaia Moustaka (AUTH)
Author(s):	Michael Sirivianos (CUT), Peter Papagiannis (CUT), Antonis Papasavva (CUT), Savvas Zannettou (CUT), Charalampos Partaourides (CUT), Athena Vakali (AUTH), Antigoni Maria Founta (AUTH), Despoina Chatzakou (AUTH), Vaia Moustaka (AUTH), Dimitris Antoniadis (CYRIC), Rafael Constantinou (CYRIC), Ioannis Agrotis (CyRIC), Marios Vodas (LSTech), Dimitrianos Savva, Evangelos Kotsifakos (LSTech)
Dissemination Level:	Public
Nature:	Demonstration
Version:	0.8









PROPRIETARY RIGHTS STATEMENT

This document contains information, which is proprietary to the ENCASE Consortium. Neither this document nor the information contained herein shall be used, duplicated or communicated by any means to any third party, in whole or in parts, except with prior written consent of the ENCASE consortium.

ENCASE Project Profile

Contract Number	691025
Acronym	ENCASE
Title	ENhancing seCurity and privacy in the Social wEb: a user-centered approach for the protection of minors
Start Date	Jan 1 st , 2016
Duration	48 Months

Partners

	Cyprus University of Technology	Cyprus
	University College London	United Kingdom
	Aristotle University	Greece
	Universita Degli Studi, Roma Tre	Italy
	Telefonica Investigacion Y Desarrollo SA	Spain
	Cyprus Research and Innovation Center, Ltd	Cyprus
	SignalGenerix Ltd	Cyprus
	LSTech	United Kingdom

Document History

AUTHORS

- (CUT) Michael Sirivianos, Antonis Papisavva, Peter Papagiannis, Savvas Zannettou, Charalampos Partaourides
- (AUTH) Despoina Chatzakou, Athena Vakali, Antigoni Maria Founta, Vaia Moustaka
- (CYRIC) Dimitris Antoniadis, Rafael Constantinou
- (LSTech) Marios Vodas, Dimitrianos Savva, Evangelos Kotsifakos

VERSIONS

Version	Date	Author	Remarks
0.1	03.07.2018	AUTH	Initial Table of Contents (TOC)
0.2	08.10.2018	AUTH	Complete TOC
0.3	10.11.2018	All Authors	Contributions from Partners Received
0.4	07.12.2018	AUTH & CUT	Revision and Restructure
0.5	10.12.2018	CUT	Editing of Executive Summary, Introduction and Conclusion
0.6	13.12.2018	CUT	Revision and addition of video links
07	28.12.2018	AUTH & CUT	Proposed Final Version
08	29.12.2018	CUT	Final Version

Executive Summary

Online Social Networks (OSNs) are a very big part of our lives today in a daily basis. Unfortunately, OSNs have also become the main mechanism for cyber bullies to hide behind a screen while distressing their victims. In particular, numerous studies reveal that hate speech, offensive language, sexism, racism and other types of abusive behavior have become a common phenomenon in many online social media platforms.

The aforementioned types of abusive behavior in OSNs can pose a major problem, especially when youngsters are the receiving end of this malicious act. One of the main objectives of the ENCASE project is to develop a browser add-on, along with its corresponding Intelligent Web-Proxy (IWP) which is responsible to host the trained classifiers extracted from its OSN Data Analytics Software Stack, for identifying the previously stated types of problematic behaviors.

To this end, in this document we demonstrate the ENCASE tool which is consisted by: a) the OSN Data Analytics Software Stack, a strong machine where we host the datasets collected throughout our research to train our machine learning algorithms for extracting the trained malicious behavior detection classifiers; b) the Intelligent Web-Proxy, a small machine that can be configured in the network of the protected group, designed to host the trained classifiers and monitor the online traffic activity of minors in numerous OSNs. The Intelligent Web-Proxy is the heart of our architecture, which in a sophisticated, seamless and secure way protects our focus group – youngsters; c) the browser add-on, designed to notify the minors about any malicious behavior detected by the IWP. In addition, the browser add-on seamlessly protects the minors by censoring hateful content, raid, abuse, and cyberbullying; and last, we demonstrate d) the Parental Console, the bridge that connects the browser add-on of the minor to their parents. This fine-grained console enables the custodians of the minors, in a privacy preserving way, to be notified about the detected malicious behavior detected by the IWP.

This demonstrator document provides a brief glance to the first functioning, alpha tested prototype of the ENCASE tool.

Table of Contents

Executive Summary	4
List of Figures	7
List of Tables	7
1. Introduction	8
2. ENCASE Cloud Infrastructure	9
2.1. Virtual Machines	9
2.2. Network	10
2.2. Security	10
3. ENCASE Architecture.....	11
3.1. The Intelligent Web-Proxy	12
3.2. Data Access Layer.....	14
3.2.1. Data Access Layer API Calls	16
3.2.2. Data Access Layer API calls outputs.....	16
3.2.3. Data Access Layer Scenario.....	18
3.2.4. SWAGGER API Documentation	20
3.3. Parental Console and Design Principles.....	20
3.3.1. Panic button (demo)	24
3.4. Browser add-on.....	25
4. Detection of Abusive Users in Twitter (demo).....	26
4.1. Project Description.....	26
4.1.1. Methodology.....	27
4.1.2. Conclusion.....	28
4.2. Demonstration Description.....	28
4.3. Section References.....	28
5. Detection of Hateful and Racist memes (demo).....	29
5.1. Project Description.....	29
5.2. Demonstration Description.....	29
6. Chat Monitoring Complemented with Sentiment and Affective Analysis (demo)	29
6.1. Overview	29
6.1.1. Methodology.....	31

6.1.2.	Conclusion.....	32
6.2.	Demonstration Description.....	34
7.	Cyberbullying Detection through Emotion Recognition of Minor’s Chat Conversation (demo) ..	35
7.1.	Project Description.....	35
7.2.	Methodology.....	36
7.3.	Demonstration Description.....	36
8.	Conclusions and Future Work.....	37
9.	Copyright and Intellectual Property.....	38

List of Figures

Figure 1. Finalized ENCASE Framework Architecture	12
Figure 2. Installing MITMproxy certificate	13
Figure 3. Facebook’s certificate issued by our IWP (MITMproxy)	13
Figure 4. IWP hides the Facebook feeds wall on the browser of the minor.....	14
Figure 5. IWP hides the Facebook chat on the browser of the minor	14
Figure 6. Data Access Layer.....	15
Figure 7. The Intelligent Web-Proxy Architecture	19
Figure 8. SWAGGER Framework API documentation for all modules	20
Figure 9. Visibility and Cybersafety options on the Parental Console	23
Figure 10. Minor’s Facebook news feed monitored through the Parental Console	23
Figure 11. Minor’s Facebook chat monitored through the Parental Console	24
Figure 12. Block Unwanted sites through the Parental Console	24
Figure 13. Message displayed to the child when accessing blocked web-pages.....	25
Figure 14. Browser add-on interface	25
Figure 15. Visibility Options as seen by the minor through the browser add-on.....	26
Figure 16. Pipeline of Abusive Detection Process.....	27
Figure 17. Data Retrieval and Storage	32
Figure 18. Top 1 affect - Joy	33
Figure 19. Top 2 affect - Anticipation	33
Figure 20. Top 3 affect – Trust	34
Figure 21. SWAGGER framework API documentation.....	35

List of Tables

Table 1. Virtual machines along with their specifications	9
Table 2. Virtual Private Network.....	10
Table 3. Firewall rules	10
Table 4. Data Access Layer API Calls	16
Table 5. Data Access Layer API Calls Inputs and Outputs	16
Table 6. Data Access Layer API calls types and permissions.....	17
Table 7. ExecID and DataID type and permissions.....	17
Table 8. Literature Review	30
Table 9. Structure of the obtained dataset.....	32

1. Introduction

OSNs today are the faster and easier way to connect with people. These networks allow individuals to make new friends, build business connections or simply extend their personal base by connecting and interacting with friends of friends, etc. On the other hand of this bright side, a distressing side also appears. Recent studies show that huge numbers of young people fall victims of cyberbullying each year.

A joke among friends may not be so harmful, but a joke that all your connections can see, has other extends on the physiological state of a minor. When potentially offensive content is posted online, the amount of feedback can be excessive and is often brutal. Use of social networks may expose individuals to other forms of harassment or even inappropriate contact. This can be especially true for teens and younger children. Unless parents diligently filter the Web content their family views, children could be exposed to inappropriate content, or worse, be victims of any type of aggressive behavior without their knowledge.

This document attempts to address the above mentioned problems and demonstrates the prototype solution offered by the ENCASE project. We developed a browser add-on that can notify a minor when any type of problematic behavior is detected. In addition, the custodians of the minor may be notified of any malicious behavior through the Parental Console. This console is hosted on the Intelligent Web-proxy which also hosts automated techniques and trained classifiers with the aim to detect early indications of malicious behavior and cyberbullying, hateful memes, malicious users and posts in Twitter. Specifically:

- a) In Section 2, we go over ENCASE Infrastructure that hosts the ENCASE tools;
- b) In Section 3, we provide a brief explanation of the updated ENCASE Architecture;
- c) In Section 4, a demonstration of abusive users in Twitter is provided;
- d) In Section 5, we demonstrate the detection of hateful and racist memes in OSNs;
- e) In Section 6, we demonstrate the sentiment and affective analysis of the chat the minors have in OSNs;
- f) In Section 7, we demonstrate early cyberbullying detection based on sentiment and affective analysis;
- g) Last, we conclude this document in Section 8.

2. ENCASE Cloud Infrastructure

The ENCASE cloud infrastructure is hosted in Google Cloud Platform (GCP) in a project named EncaseEU with Project ID encaseeu-196116. It comprises one private network, six virtual machines (VM), six disks and a lot of disk snapshots, which are described in the following subsections.

2.1. Virtual Machines

There are two main VMs, which are: a) *cutbackend* and b) *cutraspberry*. Cutbackend runs the machine learning algorithms and hosts the data collected by the Intelligent Web Proxy which lies on the cutraspberry VM. On the other hand, cutraspberry runs the Intelligent Web Proxy (IWP) which normally will be installed in users’ homes. It collects the activity of users on OSNs and uses classifiers to detect malicious activity.

The cutbackend VM comes with many development tools pre-installed, such as GCC compilers, Java JDKs, Apache Maven, node.js, a pre-configured docker daemon, a docker registry etc. Researchers and practitioners can build, test and deploy new services and algorithms needed by the IWP in the cutbackend VM. For the data management developers may choose between MariaDB and MongoDB.

For our data analysts and Spark developers, a 4-node Spark cluster which runs the Apache Spark v2.2 is provided. Each node has a 15 GB RAM, 12 of which are dedicated to Spark works. Accessing the cluster is allowed only from within the cutbackend or cutraspberry instances.

The full list of virtual machines along with their specifications is presented below (Table 1)

Table 1. Virtual machines along with their specifications

Name	Machine type	Memory	Disk	Internal IP	External IP	Running OS
cutbackend	1 vCPU	3.75 GB	20 GB	10.132..0.3	32.205.106.185	Centos 7
cutraspberry	1 vCPU	3.75 GB	10 GB	10.132..0.2	32.205.100.70	
Spark-master	4 vCPU	15 GB	10 GB	10.132..0.4	none	
Spark-slave1	4 vCPU	15 GB	10 GB	10.132..0.5	none	
Spark-slave2	4 vCPU	15 GB	10 GB	10.132..0.7	none	
Spark-slave3	4 vCPU	15 GB	10 GB	10.132..0.6	none	

All VMs can be resized upon request to satisfy the growing needs of the project. The Snapshots built-in feature provided by Google Cloud is used for data protection and recovery. A new snapshot for each virtual disk in our infrastructure is created daily.

2.2. Network

An auto mode Virtual Private Network (VPC) is used for our inter-VM communication. The auto mode VPC automatically creates one subnet in each GCP region. Our default VPC consists of 17 subnets (same name), one for each region, as shown in the following Table (Table 2).

Table 2. Virtual Private Network

Network Name	Region	Subnet Name	IP Address Ranges
default	us-central1	default	10.128.0.0/20
	europa-west1		10.132.0.0/20
	us-west1		10.138.0.0/20
	asia-west1		10.140.0.0/20
	us-east1		10.142.0.0/20
	asia-northeast1		10.146.0.0/20
	asia-southeast1		10.148.0.0/20
	us-east4		10.150.0.0/20
	australia-southeast1		10.152.0.0/20
	europa-west2		10.154.0.0/20
	europa-west3		10.156.0.0/20
	southamerica-east1		10.158.0.0/20
	asia-south1		10.160.0.0/20
	northamerica-northeast-1		10.162.0.0/20
	europa-west4		10.164.0.0/20
	europa-north1		10.166.0.0/20
	us-west2		10.168.0.0/20

2.2. Security

All VMs lie behind the default network firewall provided by GCP. By default, Internet access to the VMs is prohibited except for the port 22 (for password-less SSH remote access connections) only for VMs cutbackend and cutraspberry. Table 3 depicts the configured firewall rules for the default network and their purpose in terms of application functionality.

Table 3. Firewall rules

Name	Type	Targets	Filters	Action	Priority
allow-11080	Ingress	cutbackend, cutraspberry	IP ranges: 0.0.0.0/0	Allow	1000
allow-11082	Ingress	cutbackend, cutraspberry	IP ranges: 0.0.0.0/0	Allow	1000

allow-80	Ingress	cutbackend, cutraspberry	IP ranges: 0.0.0.0/0	Allow	1000
allow-8090	Ingress	cutbackend, cutraspberry	IP ranges: 0.0.0.0/0	Allow	1000
allow-18082	Ingress	cutbackend, cutraspberry	IP ranges: 0.0.0.0/0	Allow	1000
allow-8585	Ingress	cutbackend, cutraspberry	IP ranges: 0.0.0.0/0	Allow	1000
allow-8686	Ingress	cutbackend, cutraspberry	IP ranges: 0.0.0.0/0	Allow	1000
default-deny-ssh	Ingress	Spark-master, Spark-slave1, Spark-slave2, Spark-slave3	IP ranges: 0.0.0.0/0	Deny	1000
default-allow-ssh	Ingress	Apply to all	IP ranges: 0.0.0.0/0	Allow	65534

3. ENCAGE Architecture

The overall architecture of the ENCAGE Framework has been finalized. Figure 1 below, depicts the updated Architecture of the ENCAGE Framework. As shown in the diagram, the OSN Data Analytics Software Stack Environment (Back-End) is one and unique in our Architecture. This strong server is responsible to keep any datasets and ML Algorithms developed throughout the whole project. In addition, after training these ML algorithms, the Back-End is responsible to send all the trained classifiers and rules to all the Intelligent Web-Proxies (IWP) registered in our system.

The Intelligent Web-Proxy is one per customer. It hosts the trained classifiers and rules and it monitors and protects the traffic of the minor in OSNs. In addition, the Parental Console is hosted in this environment. The Parental Console is a web-based platform, Designed for the custodians of the minors to review the malicious behavior detected by the trained classifiers located in the IWP.

Last, our architecture comprises the browser add-on. The browser add-on is supposed to be installed on the browser of the minor. This add-on is the only tool the minor has to communicate with the ENCAGE system. It notifies the youngster about any malicious activity detected and whether it is about to do something that might expose the privacy of the minor. In addition, it allows the child to see what their custodians can see and also provide them with the ability to change their visibility options.

The following subsections 3.1, 3.2, 3.3, and 3.4, explain the IWP, the Parental Console and the browser add-on in more detail.

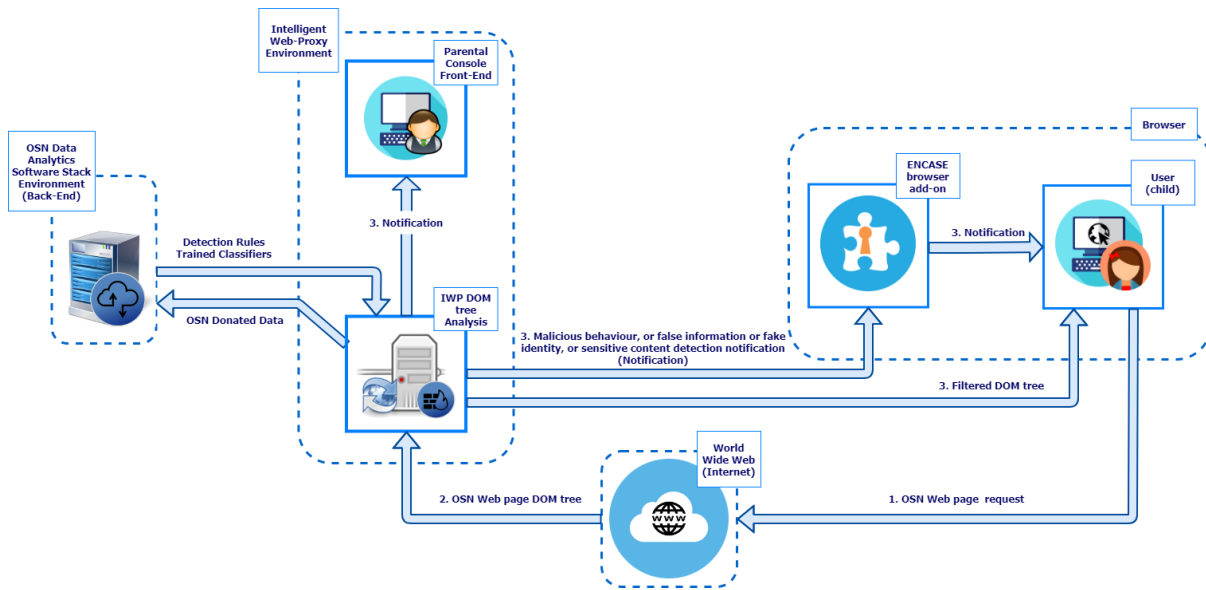


Figure 1. Finalized ENCASE Framework Architecture

3.1. The Intelligent Web-Proxy

The Intelligent Web-Proxy (IWP), the beating heart of the ENCASE architecture, is responsible to host all the trained detection classifiers developed under WP4, WP5 and WP6.

To protect the minor, it captures the online activity of the child before it reaches its browser. Then, it executes TLS Termination in order to be able to read this activity. After successful TLS Termination, the traffic of the child will be sent to the hosted trained classifiers in order to detect any malicious acts. Based on the outputs of these classifiers, the IWP is responsible to show or hide parts of the traffic activity of the child according to the malicious activity detected by the classifiers.

The technologies used to build this IWP are Python, MongoDB, PHP and Javascript. In addition, for capturing and executing TLS Termination on the online activity of the child, the MITMproxy¹ was used. MITMproxy is based on python and can intercept, read and write to a web page before it reaches the end user by installing an SSL certificate on the machine of the end-user-the child in our case.

Figure 2 depicts the process of installing the certificate of our IWP on the child’s browser. After successful installation, the user will be able to see that the certificate is issued by the ENCASE MITMproxy as shown in Figure 3.

¹ MITMproxy is an open-source, interactive man-in-the-middle proxy for HTTP and HTTPS. www.mitmproxy.org

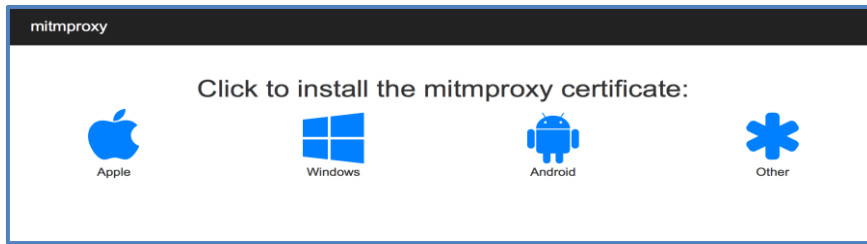


Figure 2. Installing MITMproxy certificate



Figure 3. Facebook's certificate issued by our IWP (MITMproxy)

Custom scripts running in the MITMproxy were created to retrieve the whole web-page content, send it to the appropriate classifiers and whilst waiting for their answer, hide the body of the web-page so the minor cannot see any content that is not analysed for malicious activity. Figure 4 and Figure 5 respectively, depict the case where the web-page and the chat of Facebook are not presented to the minor's browser until the IWP analyses it for malicious acts.

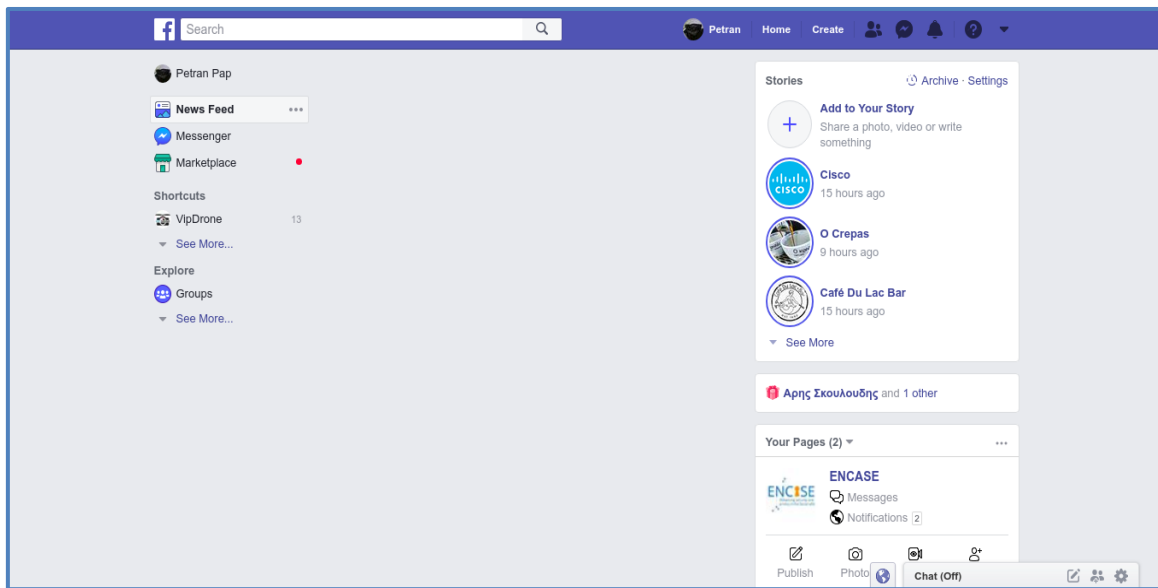


Figure 4. IWP hides the Facebook feeds wall on the browser of the minor

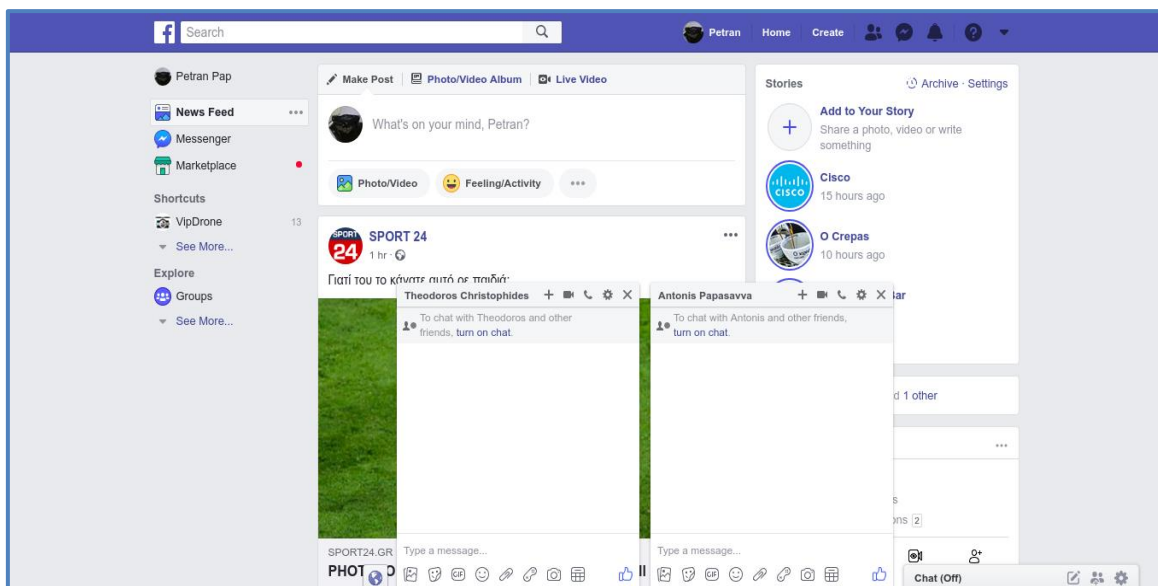


Figure 5. IWP hides the Facebook chat on the browser of the minor

3.2. Data Access Layer

The Data Access Layer (DAL) is the main storage area hosted in the IWP and the Back-End of the ENCASE infrastructure. Its logic and functions were presented in Figure 6.

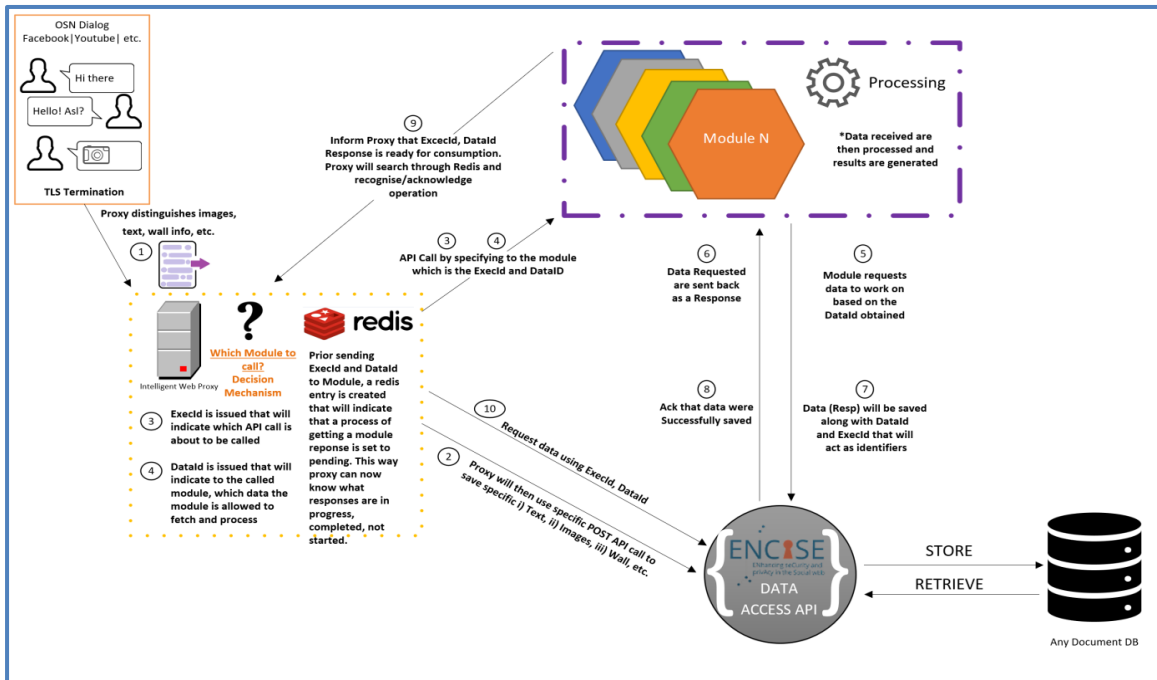


Figure 6. Data Access Layer

1. IWP distinguishes images, text, wall info, etc.
2. IWP will then use specific POST API Call of Data API to store content appropriately
 - a. Text
 - b. Images
 - c. Wall
 - d. Etc.
- IWP then must decide which module(s) will call and more specifically which API calls to execute. (Decision process)
 3. ExecID is issued that will indicate which API call is about to be called.
 4. DataID is issued that will indicate to the called module, which data the module is allowed to fetch and process.
 5. Module requests data to work on based on the DataID obtained.
 6. Data Requested are sent back as a response.
 7. Data (Resp) will be saved along with DataID and ExecID that will act as identifiers when there is a need to identify past execution with responses.
 8. Data Access Layer acknowledges that data were successfully saved to the Module that asked in the first place to save data.
 9. Module then informs IWP (desirable via Redis cache) that the process is finished (Process is identified via ExecID and DataID) and ready for consumption.
 10. IWP will then contact Data Access Layer to retrieve response.

Execution ID

Execution ID is generated as a unique nonce and it can be created with many ways.

Data ID

Data ID is constructed as follows:

- **Example 1:** URI_Data_Access_Layer?CaselD=Conv_Petros_Antonis_FB&From=2018-09-13T20:00&To=2018-09-13T21:15&Type=Chat
- **Example 2:** URI_Data_Access_Layer?CaselD=Conv_Petros_Antonis_FB & Type=Chat&UniquelD=cvsnvusdvndswn3r43245325b
- **Example 3:** URI_Data_Access_Layer?CaselD=Conv_Petros_Antonis_FB & Type=Image&UniquelD=cvsnvusdvndsw

3.2.1. Data Access Layer API Calls

All API calls of the Data Access Layer are presented in Table 4. It is important to note that all the Modules hosted in the classifier, i.e., malicious behavior detection classifiers, are data agnostic.

No module knows the Data Access Layer, other than the hosting machine, i.e., Intelligent Web-Proxy, or the Back-End.

Table 4. Data Access Layer API Calls

#	HTTP Method	API Call	*Input	*Output	Affected Collection	WHO? P-IWP M-Module A-All
1	POST	SaveChat	**ChatDataSchema	Ack	Chat	P
2	POST	SaveImages	**ImageBlobAndMetadata	Ack	Images	P
3	POST	SaveWall	**WallDataSchema	Ack	Wall	P
4	GET	ObtainData	GetDataFilter	Chat, Images, Wall [1,1]	-	All
5	PUT	SaveExecResult	ExecResult	Ack	Produced Results	M
6	GET	GetExecResult	GetExecResult	ExecResult	-	All

** Chat/Image/Wall schema must be well maintained since we need to cover up all cases. i.e. each time IWP intercepts a message we need to be able to reconstruct the correct CaselD. CaselD needs to be a unique identifier between a child and a potential predator in an OSN. We do not have a way to know that a potential predator uses a different name in another OSN. So each time new data are streamed in, the Save_ API calls need to know the correct CaselD so as to save data which are then retrieved by modules.

3.2.2. Data Access Layer API calls outputs

The lists, the inputs and outputs of the DAL API calls were presented in Table 5, while Table 6 outlines the type of each parameter and who can use it.

Table 5. Data Access Layer API Calls Inputs and Outputs

GetDataFilter			
Name	Type	Use When	Description

		Obtaining P-IWP M-Module A-All	
CaseId	String	All	Unique identifier between a child and a potential predator
From	DateTime	All	-
To	DateTime	All	-
Type	String [Chat, Image, Wall, etc]	All	Type identification distinguishes from where the data are going to be retrieved. Data can be retrieved by either the chat collection, image collection, wall collection, etc.
Uniqueld	String	All	In case of an image and or metadata of other than text, Uniqueld will act as an identifier to what a module requests. e.g. when Christos and Loizos will ask for particular images, these requests will be fulfilled with the use of Uniqueld

Table 6. Data Access Layer API calls types and permissions

ExecResult			
Name	Type	Use When Storing P-IWP M-Module A- All	Description
ModuleName	String	All	Module Name is a unique string identifier that distinguishes each module
ExecID	String	All	ExecID as this is being send from IWP
DataID	String	All	DataID as this is being sent from IWP, this way each execution is related to the data that were used
Date	DateTime	All	-
Resp	Object	All	Whichever response is returned from a Module

The types and permissions of the ExecID and DataID were presented in Table 7.

Table 7. ExecID and DataID type and permissions

GetExecResult			
Name	Type	Use When Storing	Description

		P-IWP M-Module A- All	
ExecID	String	All	ExecID as this is being send from IWP
DataID	String	All	DataID as this is being sent from IWP, this way each execution is related to the data that were used

3.2.3. Data Access Layer Scenario

The following scenario demonstrates the actual purposed flow using both the steps outlined above as well as the various API calls purposed.

1. A new chatline is being intercepted by the IWP
 - a. The IWP know who is sending to whom that chatline
 - b. Thus, the followings are being stored
 - Actual Text (Chatline)
 - Date of exchanged message
 - Sender’s/Receiver’s Username
 - Medium of Communication
 - Origin (e.g., Facebook, YouTube, Twitter, etc.)
 - c. An object is prepared to be saved via the DataAccessAPI
 2. DataAPI Call – SaveChat is then used along with the object that previously created which complies with the ChatDataSchema. The IWP then received an acknowledgement that verifies that the storing was successful.
- *IWP then must decide which module(s) will call and more specifically which API calls to execute.*
3. ExecID is then created which is a unique Nonce that acts as an identifier of a unique process which is initiated as follows: *ExecID = 5af42fa1d4d53d034eda12c5*
 4. DataID is issued that will indicate to the called module, which data the module can fetch and process.

DataID = URI_Data_Access_API?CaselD=Conv_Petros_Antonis_FB&From=2018-09-13T20:00&To=2018-09-13T21:15&Type=Chat
- *Each module needs to obtain as an input the ExecID and DataID. This way the DataID will be used to indicate which data are consumed by the module for processing. The DataID is actually the complete URL that when performing the GetDataFilter API call, appropriate data are returned.*
5. Module requests data to work on via the DataID endpoint obtained from IWP.
 6. Data Requested are being send back as a response from Data Access Layer (DAL).
- *Excessing Processing produces results*
7. Module then prepares a SaveExecResult object that is then being sent to SaveExecResult API call to DAL.

8. DAL then acknowledges that data were successfully saved to the ProducedResults collection.
9. Module then informs IWP (via Redis) that the process is finished (Process 7 is identified via ExecID and DataID) and that the results are ready for consumption.
10. IWP will then contact DAL via GetExecResult to obtain the produced response via constructing a GetExecResult.

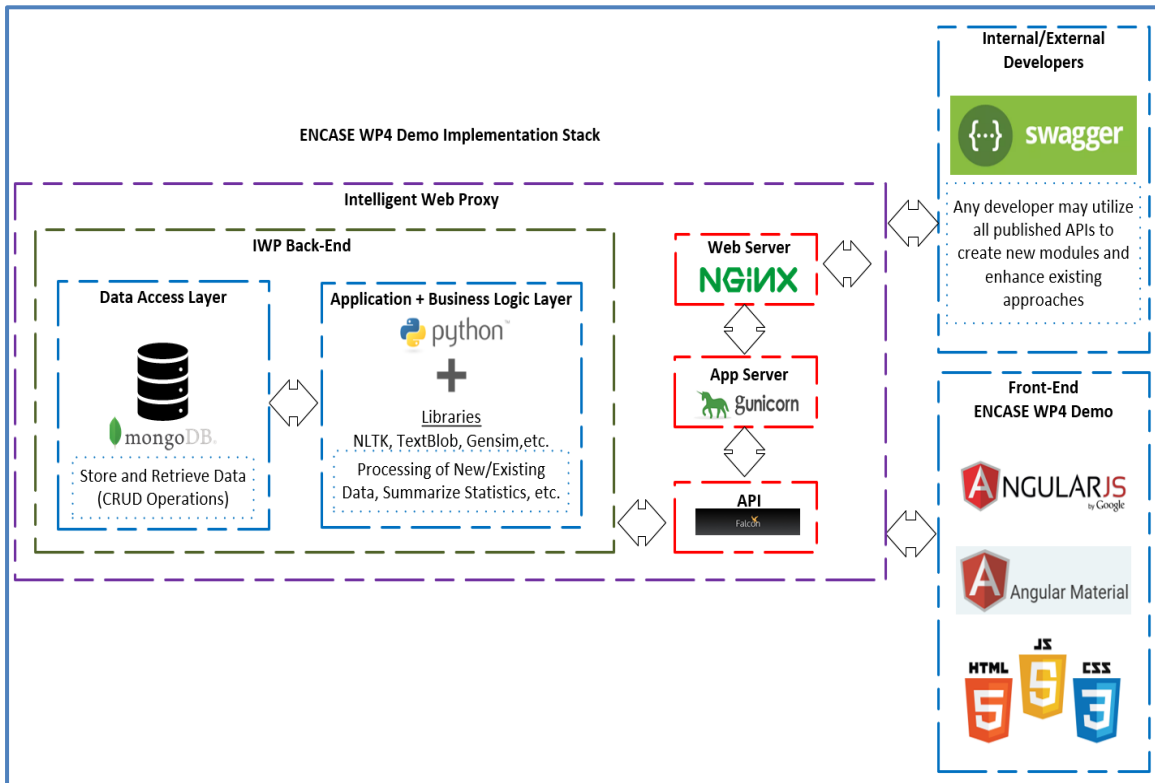


Figure 7. The Intelligent Web-Proxy Architecture

Figure 7 presents the Architecture and Implementation Stack that was used for deploying this implementation to the ENCASE IWP. It is important to note that a similar approach was used for all the detection mechanisms, hosted in the IWP.

In detail, the Data Access Layer contains the database where all the incoming traffic of the minor is stored. MongoDB was selected to keep this data. The Application and Business Logic Layer contains the actual application of the classifier, along with its API calls. For the APIs the FalconX environment was used. This communicates with the gunicorn App Server which then communicates with the Nginx web server. The front-end is based on AngularJS. Finally, all APIs are available through swagger. This is a very powerful tool for sharing APIs with other users.

For the ease of deployment, the SWAGGER framework is used where the documentation of all the APIs of each module are stored, as shown in Figure X.

3.2.4. SWAGGER API Documentation

In order for all the modules to have an easy documentation with regards to their API calls, all the API calls of each module hosted in the IWP, or the Back-End, are now required to upload their *yaml file* with their API documentation to a SWAGGER framework hosted platform as depicted in Figure 8.

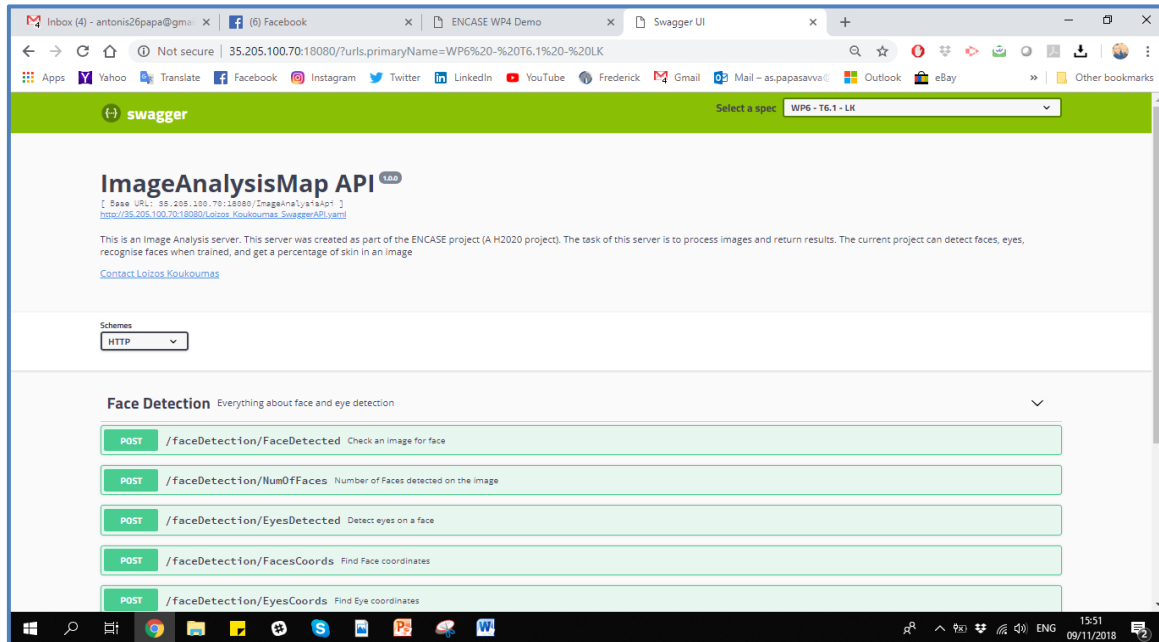


Figure 8. SWAGGER Framework API documentation for all modules

3.3. Parental Console and Design Principles

The Parental Console is the admin panel created for the custodians of the minors. Through this fine-grained tool, the parents can register their self and their children. Through this console, the parents can monitor the malicious activity detected by the IWP and receive notifications that will clearly show the date and time of the incident, as well as the chat, post and the user that caused it.

Advanced machine learning algorithms were chosen instead of simple rule-based filters. The downside of having such specific rules is that they are blunt. There are situations where we might have a specific piece of content that technically does not violate our policies, but when this content is analysed with advanced machine learning techniques, it might turn out to be hate speech, sarcasm, raid, sexual grooming, etc. Such techniques allow us to detect bullies or predators that are close to the line. Thus, our aim is to have these granular standards so that we can control for bias.

The Intelligent Web-Proxy was designed and implemented based on the following design principles:

1. Placing functionality (filters, text replacement, notifications, data submission to the Back-End, Account Management, etc.) in the Intelligent Web-Proxy instead of the browser add-ons when it can be correctly and efficiently implemented. The reasoning behind this principle is:
 - a. We wish to prevent a minor from modifying or disabling the functionality of the ENCASE system through the browser add-ons. For instance, in case a minor accidentally or willingly disables the ENCASE browser add-ons, the Intelligent Web-Proxy does not get affected and all the processes and functionalities can continue their operation normally (given that the

- device is still configured to route social network services through the Intelligent Web-Proxy; we assume that the minor user does not have the permissions or access to do so). Also, the Intelligent Web-Proxy will be able to notify the parent through the Parent Platform that the browser add-on of the minor user is not responding anymore.
- b. ENCASE aims to provide the ability to seamlessly support multiple types of clients (desktop browsers, mobile apps, etc.) with minimal client or client platform configurations or modifications.
 - c. Browser add-ons do not support complex functionalities other than javascript and HTML scripts. Text replacement, picture encryption, filtering etc., are too complex functionalities to run on a browser add-on.
 - d. Without The Intelligent Web-Proxy, the browser add-ons should call REST API request from the Back-End every time they need to identify suspicious content, e.g., cyberbullying.
 - e. Having some functionalities on the Intelligent Web-Proxy prevents it from calling REST API requests from the Back-End every time it needs to identify suspicious content, e.g., cyberbullying.
2. Placing some functionalities on the Intelligent Web-Proxy, solves the potential problem of the whole ENCASE system being down in case of a Back-End unavailability, thus solving the problem of single point failure. The functionality (filters, text replacement, push notification to the browser add-on, etc.,) placed on the Intelligent Web-Proxy is able to operate normally. For instance:
- a. The Intelligent Web-Proxy can push notification to the browser add-on without the need of the Back-End.
 - b. The Intelligent Web Proxy can replace cyberbullying content before it reaches the device of the minor, without calling REST API requests from the Back-End.
 - c. Rules and trained classifiers are generated in the Back-End. Trained classifiers will be placed on the Intelligent Web-Proxy only if they can run efficiently.
 - d. The Back-End collects data from all the Intelligent Web-Proxies to generate detection rules or trained classifiers.
 - e. Data collected from the Intelligent Web-Proxies will be used in order to generate cyberbullying detection rules, sexual cyber grooming detection rules, distressed behavior detection rules, aggressive behavior detection rules, fake identity detection rules and false information detection rules.
3. Warning, flagging and feedback functionality on the browser add-ons
- a. Warnings will be displayed to the user through the add-ons, after a notification pushed by the Intelligent Web-Proxy based on the suspicious behavior detected.
 - b. The users should be able to flag content as cyberbullying activity, sexual cyber grooming activity, aggressive behavior activity, fake identity, false information and sensitive pictures through the browser add-ons in case the Intelligent Web-Proxy failed to identify them.
 - c. The users should be able to give their feedback based on the activity detected by the Intelligent Web-Proxy. For example, in case the Intelligent Web-Proxy detects cyberbullying it pushes a notification to the browser add-on. The browser add-on will

present the notification/warning to the user explaining that cyberbullying was detected. Then, the user will be able to give feedback whether the detection of the Intelligent Web-Proxy was right or not.

4. Visibility of content that the parent and/or the Back-End can see
 - a. The parent should be able to set up the Visibility settings of content in a fine-grained way and always with the consent of the minor
 - b. This way we enable various levels of monitoring for parents and the Back-End with the consent of the child, while keeping the child fully aware of what his parents and the Back-End can see (wall, chat messages, friends list, etc.).

Overall, we aim to build a system that eases the tension of ensuring the safety of minors while respecting their privacy with respect to what their custodians and 3rd parties can see. By automating the detection of malicious communication we enable custodians to be as appropriate aware with respect to the safety of their child. This will be achieved without the parent having to manually go through the minor's online communication; thus, without having to invade his/her privacy. Our intention is to warn the parent about the suspicious online activity that was detected. For instance, in case the minor has a conversation with sexual content with somebody, then, once this activity is detected, the custodian of the minor will receive a warning that such conversation is taking place. Still, the parent will not be able to see the actual content because that would violate the privacy of the teenage. Instead, the parent will only be able to see the actual conversation through his parental platform once the explicit consent of the child has been granted.

To sum up, we intent to force custodians to have a conversation with the minor; thus, bringing families closer and spreading awareness about the numerous threats that exist in contemporary OSN.

Figure 9 shows the options that the parent can add about what the parent can see (parental visibility options), what the ENCASE platform can see (Back-End visibility options) and last the cyber security options. It is important to mention that these options are enforced only when the child approves them. When the parent edits these options, the child receives a notification on his browser add-on in order to visit these options and approves and/or rejects them.

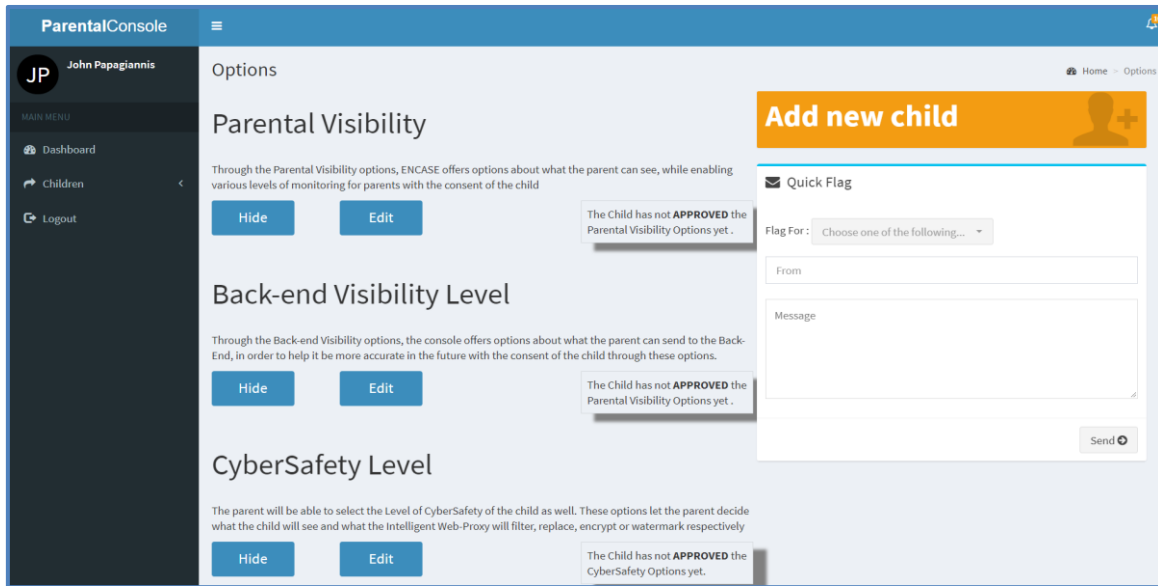


Figure 9. Visibility and Cybersafety options on the Parental Console

Based on the Visibility and Cybersafety options approved by the child, the parents will be able to monitor the child's online activity in OSN, as shown in Figure 10 and Figure 11.

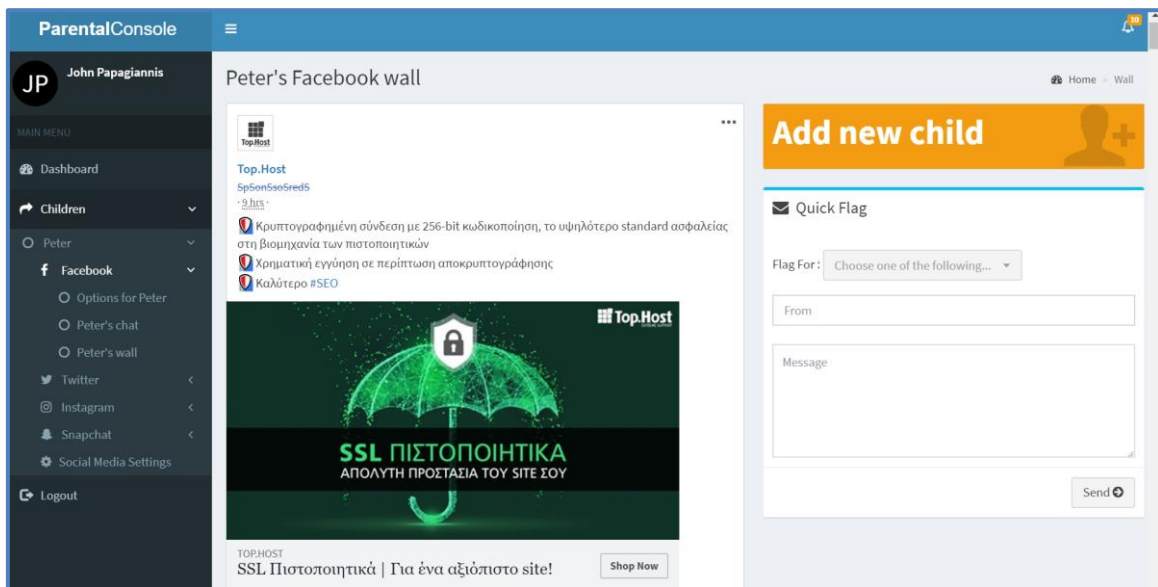


Figure 10. A Minor’s Facebook news feed monitored through the Parental Console

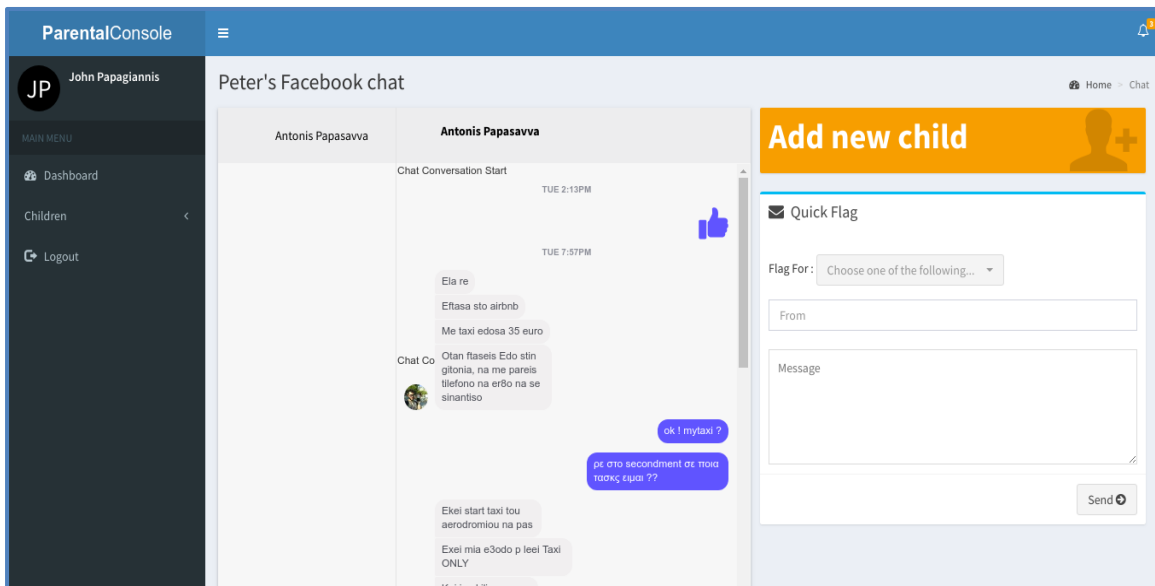


Figure 11. A Minor’s Facebook chat monitored through the Parental Console

The technologies used for the creation of the Parental Console are HTML with Bootstrap Framework, CSS, JavaScript, jQuery, Vuejs, PHP with the Slim Micro Framework² and a MySQL database.

The URL for accessing the ENCAGE Parental Console is provided here: <https://encase-proxy.socialcomputing.eu:8090>

3.3.1. Panic button (demo)

The panic button was designed to enable the custodians, through the Parental Console, to block minors accessing certain websites. As shown in Figure 12 below, there are prefixed sites for the custodian to choose, or he can add new ones by providing the full URL of the website. Then these websites will be sent and stored to the appropriate database for the IWP to know not to allow incoming and incoming traffic to these sites.

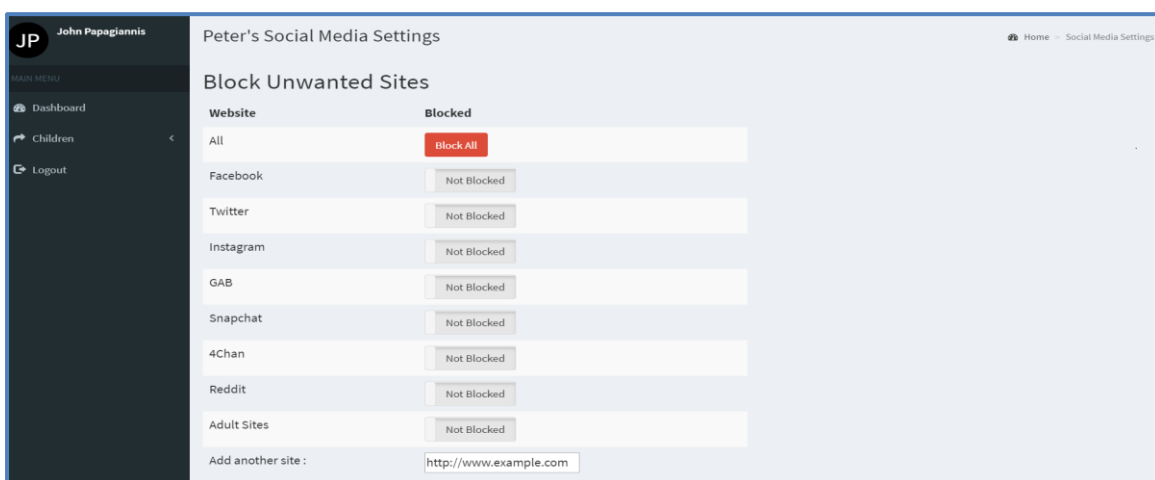


Figure 12. Block Unwanted sites through the Parental Console

² <https://www.slimframework.com/>

In case the child tries to access a website that is blocked, then the child will see the message shown in Figure 13 – “Website Blocked by the Parent with ENCASE”.

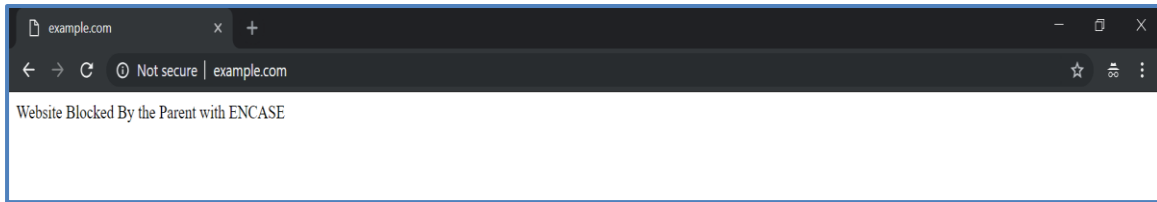


Figure 13. Message displayed to the child when accessing blocked web-pages

3.4. Browser add-on

The ENCASE chrome browser add-on is created with all the guidelines of Google. Its main purpose is to send notifications to the child and let it know about the malicious activity detected, and to also approve or reject the Visibility Options the parent set through the Parental Console.

The technologies used for the creation of the browser add-on are HTML CSS for the design, and JavaScript for retrieving and sending the options if they are approved or not.

Figure 14 shows what the child can see when pressing the add-on symbol on their Chrome browser.

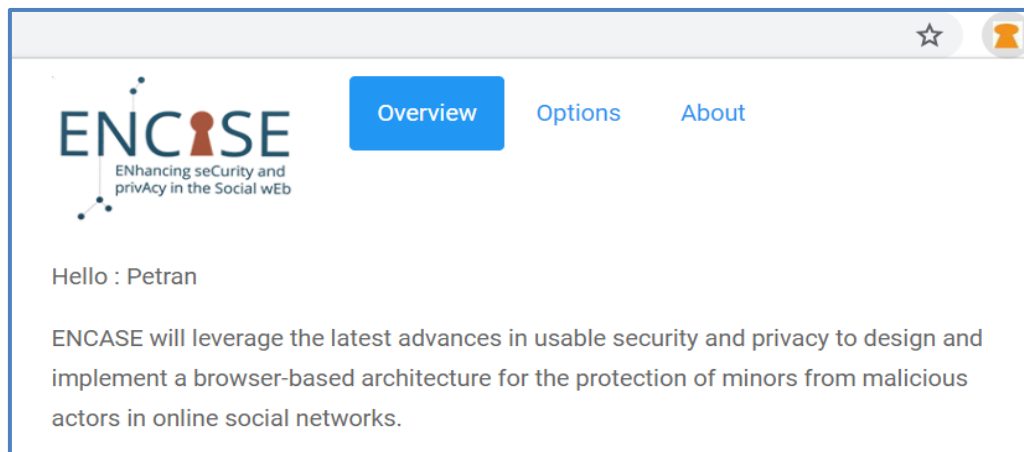


Figure 14. Browser add-on interface

Figure 15 shows what the child can see when they wish to see the Visibility Options set by their custodians. This menu is accessible from the “Options” button shown in Figure 14, above.

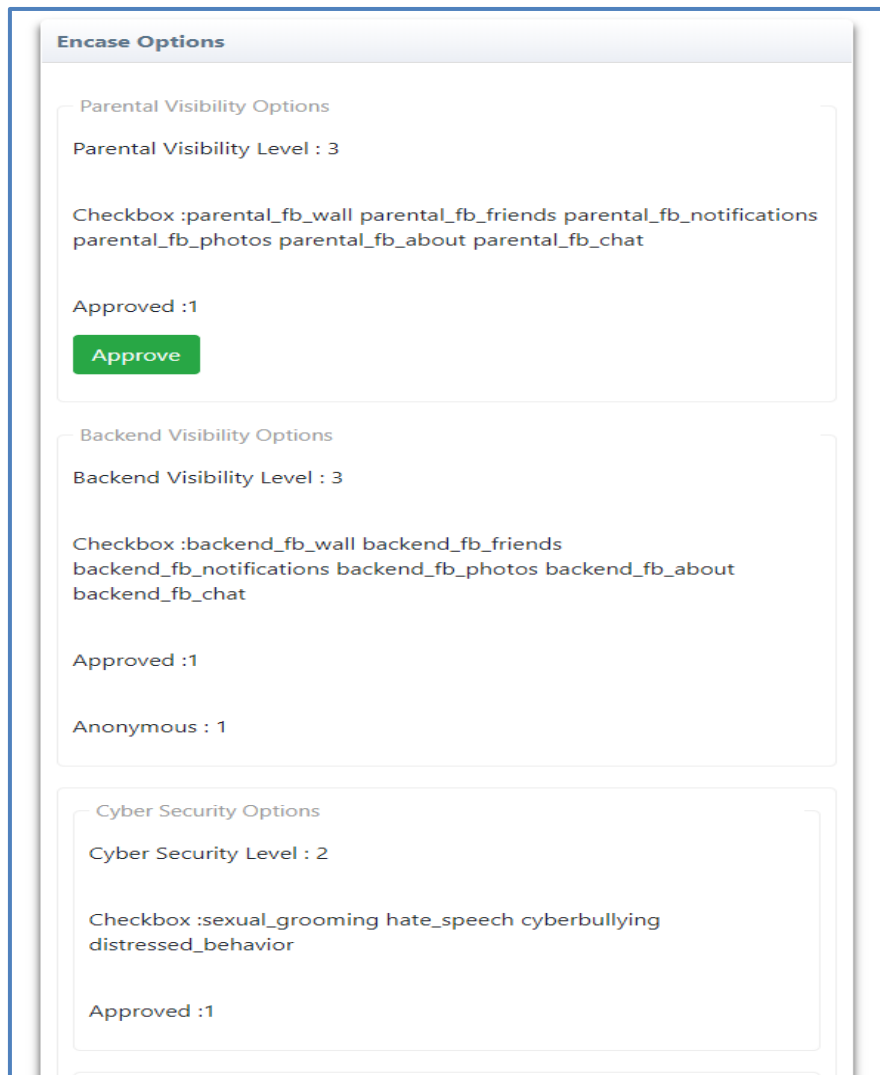


Figure 15. Visibility Options as seen by the minor through the browser add-on

4. Detection of Abusive Users in Twitter (demo)

4.1. Project Description

Cyberbullying and cyberaggression are serious and widespread issues affecting increasingly more Internet users. Arguably, in today's hyper-connected society, bullying, once limited to particular places or times of the day (e.g., school hours), can instead occur anytime, anywhere, with just a few taps on a keyboard. Cyberbullying and cyberaggression can take many forms and definitions [1, 2], however, the former typically denotes repeated and hostile behavior performed by a group or an individual and the latter intentional harm delivered via electronic means to a person or a group of people who perceive such acts as offensive, derogatory, harmful, or unwanted [1]. So, in relation to T4.1, we focused on understanding and detecting abusive phenomena that take place on Twitter, since it is one of the most popular OSN sources. More specifically, we targeted in detecting bully and aggressive users via analyzing content produced in Twitter. Figure 16 outlines the followed process.

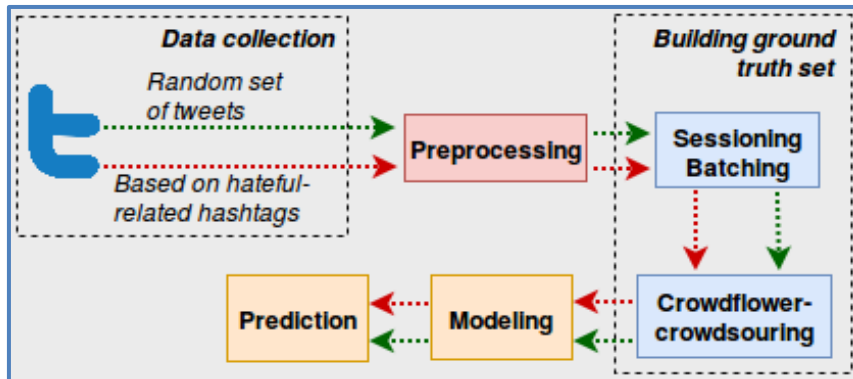


Figure 16. Pipeline of Abusive Detection Process

4.1.1. Methodology

The aforementioned content analysis was conducted utilizing the *abusive detection method*. The main modules of this method are described below.

Data Collection. Our first step was to collect tweets and, naturally, there are a few possible ways to do so. We relied on Twitter's Streaming API, which provides free access to 1% of all tweets. The API returns each tweet in a JSON format, with the content of the tweet, some metadata (e.g., creation time, whether it is a reply or a retweet, etc.), as well as information about the poster (e.g., username, followers, friends, number of total posted tweets).

Preprocessing. We removed stop words, URLs, and punctuation marks from the tweet text and performed normalization -- i.e., we eliminated repeated letters and repetitive characters; e.g., the word 'yessss' is converted to 'yes'. This step also involved the removal of spam content based on tweeting behavior (i.e., number of hashtags per tweets and number of duplicate posts).

Sessioning/Batching. Since analyzing single tweets does not provide enough context to discern if a user is behaving in an aggressive or bullying way, we grouped tweets from the same user, based on time clusters, into sessions. This allowed us to analyze contents of sessions rather than single tweets. Next, we divided sessions in batches, as otherwise they would contain too much information to be carefully examined by a crowdworker within a reasonable period of time (see crowdsourcing).

Crowdsourcing. We built ground truth (needed for machine learning classification) using human annotators. For this we used a crowdsourced approach (based on crowdflower), by recruiting workers who were provided with a set of tweets (batches) from a user, and were asked to classify them according to predefined labels. More specifically, the goal was to label each Twitter user -- not single tweets -- as normal, aggressive, bullying, or spammer by analyzing their batch(es) of tweets.

Modeling. This process entails the selection of such features that can better describe the behaviors under examination, i.e., bullying, aggressive, spam, and normal. Different features selections have been considered, based on the posts themselves (e.g., number of hashtags, emoticons, urls), on a user's profile (e.g., number of followers/friends, and a user's account age), or on a user's social network (e.g., who follows who, who posted to whom).

Classification. The final step was the classification process using the (extracted) features and the ground truth. Different machine learning techniques were considered in this task, including probabilistic classifiers (e.g., Naive Bayes), decision trees (e.g., J48), and ensembles (e.g., Random

Forests). The best performance with respect to training time and performance, obtained with the Random Forest classifier.

4.1.2. Conclusion

Although the digital revolution and the rise of OSN enabled great advances in communication platforms and social interactions, wider proliferation of harmful behavior has also emerged. Unfortunately, effective tools for detecting harmful actions are scarce, as this type of behavior is often ambiguous in nature and/or exhibited via seemingly superficial comments and criticisms. Aiming to address this gap, we developed a novel system geared towards automatically classifying two specific types of harmful online behavior, i.e., cyberbullying and cyberaggression. We relied on crowd-workers to label 1.5k users as normal, spammers, aggressive, or bullies, from a corpus of almost 10k tweets (distilled from a larger set of 1.6M tweets), using an efficient, streamlined labeling process. We investigated 30 features from 3 types of attributes (user, text, network based) characterizing such behavior. While prior work almost exclusively focused on user- and text-based features (e.g., linguistics, sentiment, membership duration), we performed a thorough analysis of network-based features, and found them to be very useful, as they actually are the most effective for classifying aggressive user behavior (half of the top-12 features in discriminatory power are network-based).

4.2. Demonstration Description

The developed tool is able to detect abusive behaviors as these are manifested on Twitter. Next, we describe the followed process in order to conclude to the observed behavior. When a child visits the page of another Twitter user an alert is created in order to check whether such a user is abusive or not. So, initially the top 20 posts of the user's profile are collected through the Twitter API. With the successful completion of the data collection process a .json file is created which holds the aforementioned data.

The next step is the data processing. First, any posts that are not originated by the analyzed user are removed. For instance, if we are analyzing the Twitter page of user EVE which has 20 posts, but the 5 out of those 20 are posted on EVE's wall from user BOB, then we will only take into consideration the 15 posts originated by EVE for the analysis. Then, a set of attributes is extracted (i.e., user and text based) and an .arff file is created (with the extracted attributes). Finally, the .arff file feeds the classifier and the detection process starts. Once the classifier finishes the aforementioned process, a notification is produced which characterizes the user as bully, aggressor, or normal.

We present a demonstration of the above described detection. In order to see the demonstration video, please visit this link:

<https://www.dropbox.com/s/eouv98cpg7zar7k/AbusiveUsersTwitter.mp4?dl=0>

4.3. Section References

- [1] Grigg, D. W. (2010). Cyber-aggression: Definition and concept of cyberbullying. *Journal of Psychologists and Counsellors in Schools*, 20(2), 143-156.
- [2] Tokunaga, R. S. (2010). Following you home from school: A critical review and synthesis of research on cyberbullying victimization. *Computers in human behavior*, 26(3), 277-287.

5. Detection of Hateful and Racist memes (demo)

5.1. Project Description

Internet memes are increasingly used to sway and manipulate public opinion, thus prompting the need to study their propagation, evolution, and influence across the Web. This task is focused on the detection of hateful and racist memes in Facebook, using a processing pipeline based on perceptual hashing and clustering techniques, and a dataset of 160M images from 2.6B posts gathered from Twitter, Reddit, 4chan’s Politically Incorrect board (/pol/), and Gab over the course of 13 months. The complete report of this work and how the Machine Learning algorithms were generated and trained is included in D4.2 - Software libraries built on Graphos.ml using data mining for the detection of aggressive or distressed behaviors in OSN.

5.2. Demonstration Description

The trained classifier that resulted from this research is deployed in the production server of the Intelligent Web-Proxy (IWP), port 8686. When a minor accesses Facebook, the DOM tree of Facebook is captured by the IWP.

The IPW moves to detect any images contained in the DOM tree. All the images detected from the IWP are sent to the trained classifier in order to detect if the image is a hateful/racist meme.

Demonstration Scenario:

- a) A bad user, Eve, posts a hateful meme in Facebook.
- b) The minor, protected by ENCASE accesses Facebook and scrolls through their Newsfeed. In the background, the ENCASE IWP captures this hateful meme posted by Eve and it detects that it is a racist meme.
- c) The child cannot see the meme posted by Eve. The IWP seamlessly replaces that hateful meme with the message: “This picture was censored by ENCASE as hateful or racist meme.”
- d) At the same time, the custodian of the child receives a notification that the IWP detected a hateful meme. When the parent presses on the link provided in their notification, they are able to see the hateful meme posted by Eve.

We present a demonstration of the above described detection. In order to see the demonstration video, please visit this link:

https://www.dropbox.com/s/qpugtqnjaohrg74/hateful_meme_detection.mp4?dl=0

6. Chat Monitoring Complemented with Sentiment and Affective Analysis (demo)

6.1. Overview

The IWP allows the custodians of the minor to monitor the chat dialogs of the child through a well-designed web-based platform, the Parental Console, if the child also consents. Based on previous work, reported in D4.1 - Development of Automated Techniques to Detect Early Indications of Malicious Behavior of Social Network Users, Section 8, the custodians are now also able to monitor

the chat of their children, with a complementary analysis to this chat, sentiment and affective analysis.

To implement the algorithms that will be able to efficiently perform sentiment and affective analysis, the following literature review listed in Table 8 below was performed.

Table 8. Literature Review

#	Dataset Description	URL
1	Formspring.me Labeled for Cyberbullying	http://www.chatcoder.com/DataDownload
2	Semantically Analysed Metadata of Tumblr Posts and Bloggers	https://data.mendeley.com/datasets/hd3b6v659v/2
3	Challenge that urges people to speak up against online harassment. Online contest to achieve this via software	https://github.com/MLH/mlh-policies
4	Partial Dataset, 56 out of 622 convictions of dialogs that contain sexual grooming GeneralData (Predation Labelled)	http://www.chatcoder.com/DataDownload
5	Repository contains datasets with online conversations threads collected and analysed by different researches	https://www.upf.edu/web/mdm-dtic/-/the-online-conversation-threads-repository?inheritRedirect=true#.WlTveBt96Un
6	A collection of 12,696 Tweet Ids representing 4,232 three-step conversational snippets extracted from Twitter logs (Last published June 1st, 2015)	https://www.microsoft.com/en-us/download/details.aspx?id=52375
7	A collection of adult sexual conversation from chatrooms	http://web.archive.org/web/20091026233407/http://geocities.com/urgrl21f/
8	NPS Internet Chatroom Conversations, Release 1.0 consists of 10,567 English posts (45,068 tokens) gathered from age-specific chat rooms of various online chat services in October and November 2006.	https://catalog ldc.upenn.edu/LDC2010T05
9	A large unlabeled Formspring dataset, from a Summer 2010 crawl	https://www.kaggle.com/swetaagrwal/formspring-data-for-cyberbullying-detection
10	Page contains our data sets and code release for the scientific research of bullying.	http://research.cs.wisc.edu/bullying/data.html

Due to the limitations, it was decided to analyse the Perverted Justice dataset, which is an American NGO (Non-Governmental Organization). Our purpose is to awaken parents and adolescents of the importance of identifying that not always the person who speaks on the Internet is the one he/she claims to be. Our aim is to prevent the above statements by turning the website into a conviction machine to raise awareness and help with this way the parents. In this context, a full list of current convictions (by the time this presentation is formalized, it reaches **622 convicted cases**) is available here: <http://www.perverted-justice.com/?con=full>

The Perverted Justice dataset is considered a good match because it provides actual ground truth of predator behavior in his conversations with the victim. Specifically, it involves:

- Real convicted predators
- Participating in real conversations
- Multi-channel sources
- Directly approaching their victims with the target of solicitation

The limitation of the dataset is that all victims' exchanged dialogs cannot be considered valid since are victims are impersonated by volunteers and are not valid teenagers. This does not stop the purposed methodology to consider these dialogs since it is an extra indication even though these dialogs are not entirely valid.

Additionally, this dataset does not have non-harassment conversations that can be used to differentiate the identified patters. Addressing these limitations is already planned as future work, where we plan to validate our approach with additional online conversations.

6.1.1. Methodology

Since the entire *Perverted Justice dataset is not currently available* for download, a significant effort was taken to retrieve the data directly from the site. Via the use of *curl* command, we managed to extract all 622 dialogs directly from site. Out of the 622 reported cases only 581 where containing data. The rest of the cases led to dead links and not well-maintained data. Thus, our final dataset consists of these 581 convicted sexual predator cases. Instead of analyzing individual chat lines, there is an attempt to organize chat lines from the same person (predator or victim) into common behavior blocks (sessions).

These blocks allow the identification of specific patterns of behavior and how these affect the behavior or reaction of the chat responder.

After crawling incoming data via *curl*, data are then being pre-processed and then are stored in a MongoDB instance for efficient retrieval and analysing, as shown in Figure 17.

Data collection and cleansing process is the following:

NLTK and TextBlob Sentiment Analysis: Sentiment polarity. (scale [-1,1]) → showing how negative or positive is a phrase. Sentiment subjectivity (scale [-1,1]) → showing how objective or subjective the sentence is.

NRC word-emotion association lexicon. Identify the affect (emotion) of the phrases in the dataset. [anger, anticipation, disgust, fear, joy, sadness, surprise]

Genism library. Identify the topics of phrases in each exchanged post via the use of Latent Dirichlet Allocation for Topic Modeling.

Translated internet used acronyms to actual text

Session Identification. Messages exchanged within X seconds from the previous one belong to the same session Values selected for X: Change of Communication Medium, 30sec, 60sec, 5min and 30min. Session field is central for a number of values and statistics calculated.

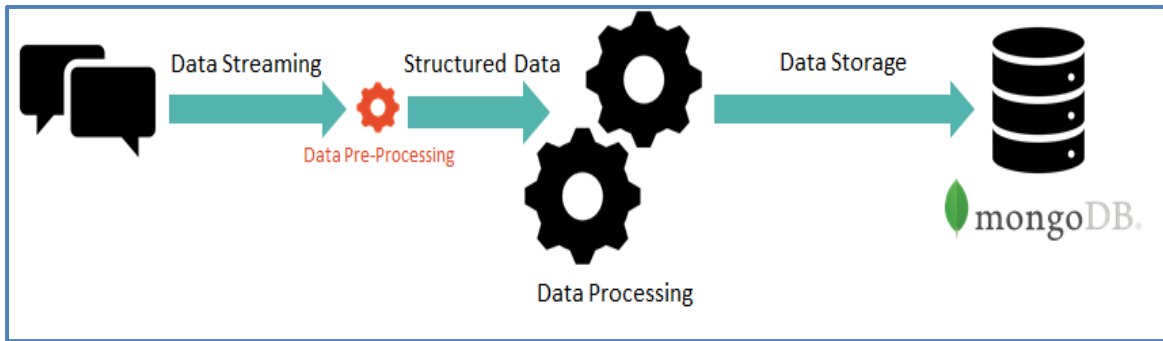


Figure 17. Data Retrieval and Storage

Table 9 shows the final structure of the obtained dataset.

Table 9. Structure of the obtained dataset

Field Name	Type	Description
Caseld	PK ()	Unique key that denotes each individual convicted case
Text	String	Exchanged Dialog Text
TextCleaned	String	Cleaned Exchanged Dialog Text as explained in the previous schema
SessionIds	Array[int]	Indexes that separates a session from another in terms of changed medium of communication and/or specific time interval
Username	String	Denoting the Predator/Victim Name
Date	DateTime	Full Date Time Information
Affect	Object	Affect Object that consists of an affect vector and a counter per affect that indicates the existence of the particular affect in the exchanged dialog
Sentiment	Object	Sentiment object that provides sentiment polarity and subjectivity of the exchanged dialog
PV	String	Indication for predator or victim [P, V]
Index	Number	Index Indication (also used for indexing purposes and efficient retrieval)
Origin	String	This field is set to ‘Perverted Justice’, but for any future additions to the database the correct origin must be used, e.g. Facebook YouTube etc.

6.1.2. Conclusion

Our analysis showed that three are the most prevalent affects showed by the predator: *i) Joy*, *ii) Anticipation*, and *iii) Trust*. Goal of the predator in the initial steps of the conversation is to come close to the victim through a joyful conversation and gradually build trust to be able to “ask for more”.

Anticipation is experience since the predator constantly request for the response of the victim to progress the conversation as fast as possible. Other affects like anger and disgust also appear later in the discussion (in subsequent sessions). Manually observing some of the conversations these affects depict the effort of the predator to find some common ground with its victim by exploiting some common “enemy” like parents, city authorities etc. Figure 18, Figure 19, and Figure 20 present the findings of this analysis.

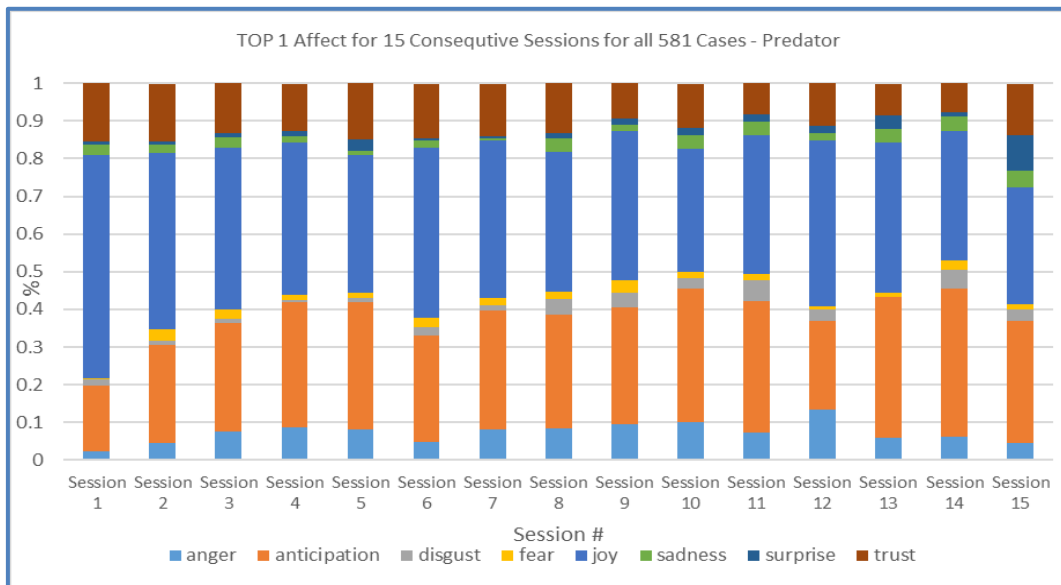


Figure 18. Top 1 affect - Joy

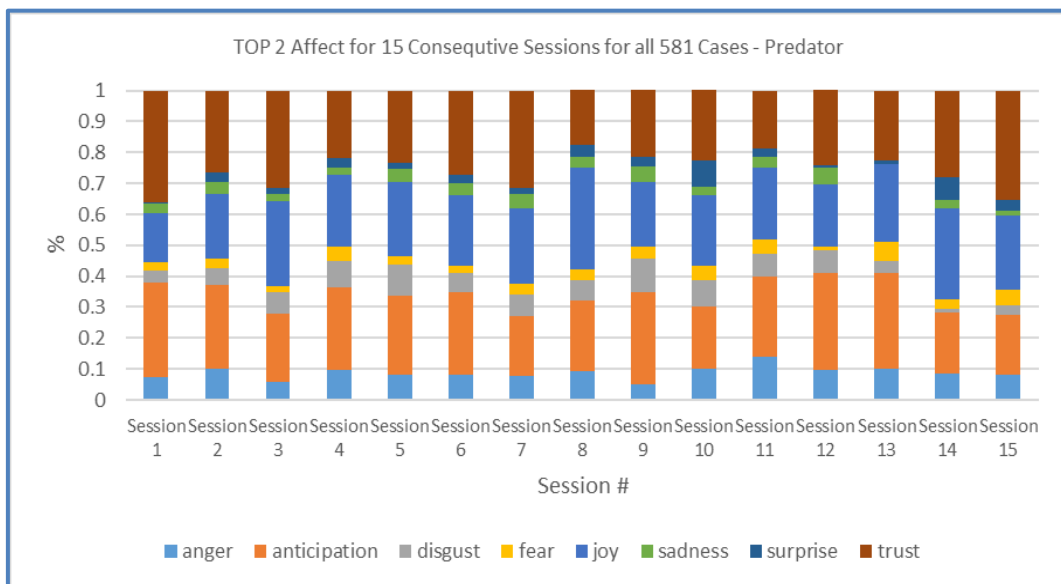


Figure 19. Top 2 affect - Anticipation

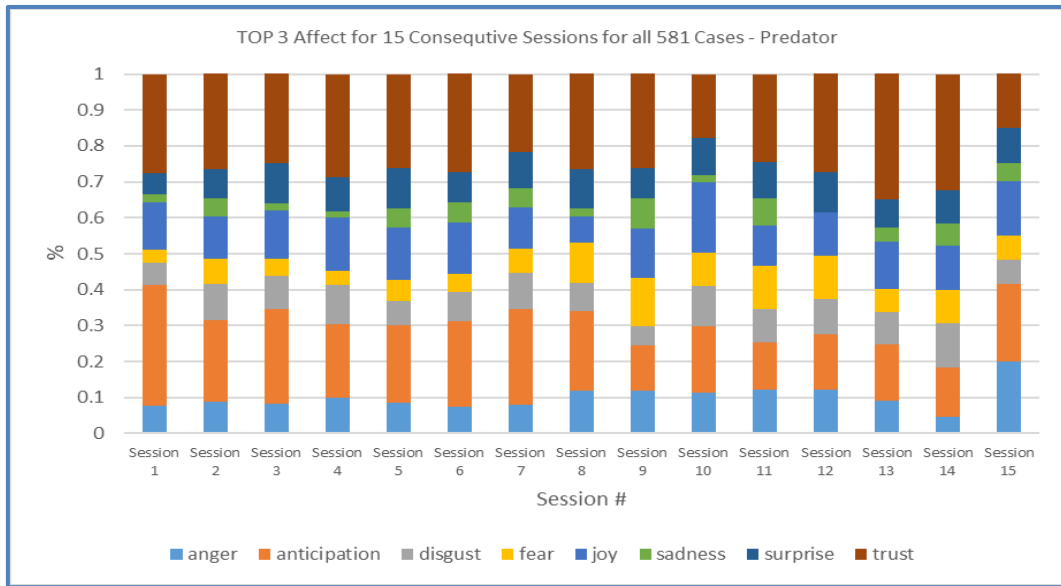


Figure 20. Top 3 affect – Trust

6.2. Demonstration Description

The IWP allows the custodians of the minor to monitor the chat dialogs of the child through a well-designed web-based platform, the Parental Console, if the child also consents.

Important outcome of the effort is the deployment of the above analysis in ENCASE infrastructure so custodians can monitor the sentiment and affective analysis of the chatting of their children through the parental console. This effort complies with the architecture described in D2.2. Thus, it was decided that a WP4 Dashboard will assure the following demonstrating purposes:

1. Visualize a convicted case’s dialog;
2. Sentiment and Affective Analysis outcome;
3. Time-Related (session base) Analysis;
4. Expose a complete set of API Calls that any ENCASE developer could use and consume raw data and/or analysis outcome

Figure 21 below, depicts the SWAGGER framework API representation of this work. Using this tool, the developers can easily test each API, see the documentation of this work and how it works.

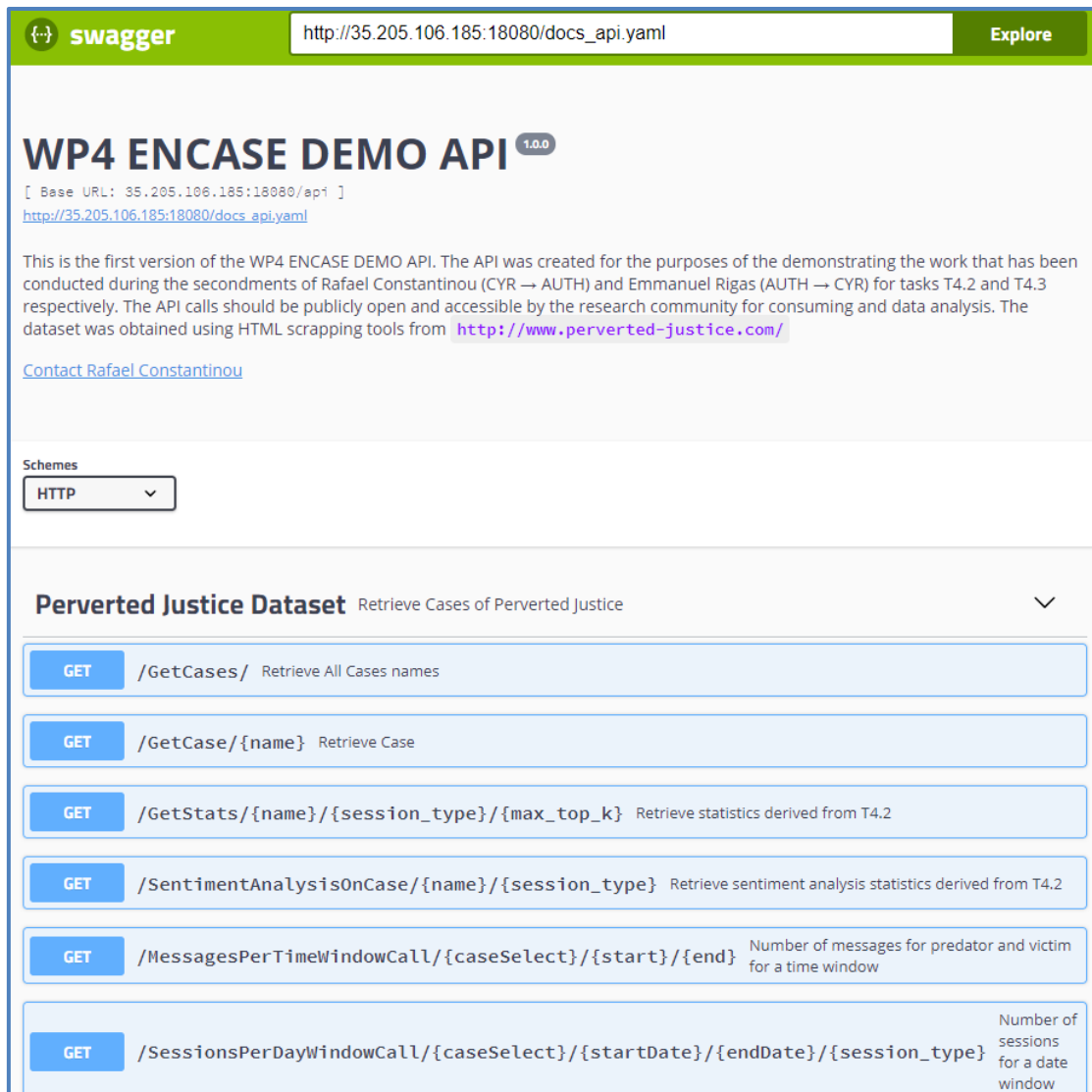


Figure 21. SWAGGER framework API documentation

In addition, we created a video, demonstrating this analysis. The parent is able to select the conversation his child had with another user in Facebook and see the feelings recorded throughout that conversation.

Please see the video by visiting this link here:

https://www.dropbox.com/s/5co9i6yxjev177a/SentimentAffectiveAnalysis_DEMO.mp4?dl=0

7. Cyberbullying Detection through Emotion Recognition of Minor’s Chat Conversation (demo)

7.1. Project Description

In this task, we aim to detect cyberbullying behavior against minors through their emotional state. If a minor is angry/sad/frustrated this can be an early indication of being cyberbullied. In this task, we formulate, train and deploy a machine learning algorithm to predict three emotional states based on the minor’s OSN conversation. The emotions in question are anger, sadness and

frustration and their predictions will give an early indication of cyberbullying towards the minor. To achieve it, we developed an innovative machine learning model that efficiently captures the correlation of the conversation advancement with the emotional state of the minor.

7.2. Methodology

Due to the nature of the task, we have given special attention to make the model fast, accurate and real-time applicable. To this end, the model can predict the minor's emotions live and on the conversation progress so far. There is no limitation on what the length of the conversation can be which is mainly due to the usage of RNNs and their inherent nature to support variable sequence length. More details about the tool used for training our Machine Learning algorithm, are provided in D4.2 - Software libraries built on Graphos.ml using data mining for the detection of aggressive or distressed behaviors in OSN.

Following the training process, we deployed the trained Early Cyberbullying Detection Deep Learning model on the ENCASE Proxy Server where the REST APIs are co-located. It was necessary to do that in order to be accessible by our REST API calls. In order to detect whether a conversation has any indications of cyberbullying our Front-end applications can call the `<encase_proxy_server_url>/early_cyberbullying_detection` REST API call that we have implemented.

To call this REST API call one has to provide as a POST parameter, a conversation URL. This URL allows our API call to request from the Data Access Layer (DAL) REST API all the information of the conversation in question. In our API call after retrieving the details of the conversation, we preprocess them in order to convert them to the format required by our machine learning model. Then we call our model providing as input the preprocessed conversation to infer the emotions of the minor. At the end, we receive from our model the inferred emotions along with their confidence scores, which are also the output of our API call.

Following is an example of the output return of our Early Cyberbullying API Response:

```
{
  "case_id": "conv_https://www.facebook.com-somebaduser.papas.5_https://www.facebook.com-petran88",
  "predictions": {
    "angry": "99.98%",
    "frustrated": "0.02%",
    "sad": "0.00%"
  },
  "desc": "This is the predicted scores for the current emotional state of the child at the end of the dialog based on the messages exchanged during the provided conversation. A score is accepted only if it is > 50%."
}
```

7.3. Demonstration Description

The chat of the minor, protected by ENCASE, is captured real-time by the ENCASE IWP. This chat is securely stored in the IWP and every 1 hour, the IWP runs this trained classification to detect early cyberbullying traits. The scenario of this demonstration is as follows:

- a) The child talks with a friend over Facebook messenger. His friend is aggressive over the chat and makes inappropriate jokes.

- b) After this conversation is ended, the kid continue browse the internet.
- c) In the background, the IWP sends this conversation to the trained classifier for analysis. This analysis shows that the child was frustrated and sad during the conversation over Facebook with his friend.
- d) Then, the IWP pushes a notification to the browser add-on of the minor indicating that the user Eve Eve shows signs of cyberbullying.
- e) At the same time, the parent receives the same notification in their Parental Console. Also, the parent is able to see the conversation his child Peter and Eve had over Facebook. In addition, the parent can see the sentiment analysis of this chat.

We created a video to demonstrate the above described detection. In order to see the demonstration video, please visit this link:

<https://www.dropbox.com/s/9d9iv7cunnp3uc5/EarlyCyberbullyingDetection.mp4?dl=0>

8. Conclusions and Future Work

In this document we demonstrated the first automated, functioning alpha-tested ENCASE tool, which is consisted by the OSN Data Analytics Stack, the Intelligent Web-Proxy, and the Browser add-on. We demonstrated how our prototype can detect:

- a) early-cyberbullying in Facebook;
- b) abusive users in Twitter; and
- c) hateful and racist memes in Facebook.

Also, this document demonstrated how the parent can monitor the problematic behavior detected by the Intelligent Web-Proxy using their Parental Console. The Parental Console enables the custodians of the minors to:

- a) see the sentiment and affective analysis of a conversation between a user and the minor in Facebook;
- b) see the hateful and/or racist memes filtered by the Intelligent Web-Proxy;
- c) monitor the online activity of the minor in Facebook and Twitter;
- d) block sites that the custodian does not wish the minor to visit; and
- e) receive notifications of any other malicious behavior detected by the IWP.

Although the ENCASE tool is still in progress and this is just a prototype, the research and development shows very promising signs of a feasible and successful tool. The efforts listed in this document helped the project reach a big milestone with regards to identifying online abuse and how to protect minors from it. In addition, the ENCASE Framework, equipped with the aforementioned detection techniques will be tested and further improved during WP7.

This deliverable is an important milestone for the ENCASE project towards better protecting minors from abusive and malicious behavior over OSNs.

9. Copyright and Intellectual Property

The intellectual property will be jointly owned between the Institutions that each of the ENCASE partners. If a project partner decides to move institutions for the duration of the project the Institution to which they move would not become a join owner, and the ownership will remain with the institution at which partners are originally based.