Web Bot Detection Using Mouse Movement

Master Thesis
submitted to the Faculty of the
Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona
Universitat Politècnica de Catalunya
by

Santiago Escuder Folch

In partial fulfillment
of the requirements for the master in
Master's degree in Advanced Telecommunication **ENGINEERING**

Advisor: Albert Calvo Ibáñez
Barcelona, Date 22/10/2022

Contents

List of Figures

Listings

List of Tables

Revision history and approval record

Revision	Date	Purpose
0	20/03/2022	Document creation
1	20/10/2022	Document revision

DOCUMENT DISTRIBUTION LIST

Name	e-mail
[Santiago Escuder Folch]	
[Albert Calvo Ibáñez]	
[José Adrián Rodríguez Fonollosa]	

Written by: Santiago Escuder Folch		Reviewed and approved by:	
Date	20/10/2022	Date	21/10/2022
Name	Santiago Escuder Folch	Name	Albert Calvo Ibáñez
Position	Project Author	Position	Project Supervisor

Abstract

Non-Legitime traffic in terms of automated internet bot traffic is a long-standing problem causing a huge economic impact and lack of trust in companies and administrations worldwide. For years, Artificial Intelligence and especially Machine Learning have been a key players fighting and helping the stakeholder to analyse and detect fraud instances automatically. However, it does not exist a reliable ground truth public dataset to evaluate and compare the proposed methodologies in the literature. Throughout this thesis, it is developed a public dataset consisting of legitimate and fraudulent web mouse movements extracted from real bot engines. In addition, it is evaluated using two Machine Learning models based on Decisions Tree classifier called LightGBM whilst the second one is based on Recurrent Neural Networks outperforming the accuracy .

Acknowledgment

I would like to give huge thanks to the people at i2CAT. Specially Albert Calvo for taking this project and helped me reformulate it when the project experienced some back-draws.

1 Introduction

This Master's thesis has been carried at i2CAT Foundation. i2CAT Foundation is a non-profit research and innovation centre that promotes mission-oriented Research and Development (R&D) activities on advanced Internet architectures, applications and services. This project has been carried in the Distributed Artificial Intelligence (DAI) group under the research line of artificial intelligence applied to cybersecurity.

Research in Artificial Intelligence (AI) applied in online web fraud detection is a hot topic of research. Fraud detection and anomaly detection in the cyberdigital world is one of the principal source of economic loses. This loses are quantified in $5.8 billion only in the US targeting a wide umbrella of victims: private companies, public organizations and even critical systems such as hospitals or utilities companies [1]. In this thesis, it is reviewed the specific case of web fraud target to identify non-legitime traffic in web sessions. The correct identification is crucial in the marketing domain allowing to quantify the real traffic of a web page. Through this thesis it is research how AI could be applied in the domain allowing to detect non-legitime traffic in an efficient fashion. Specifically, in this thesis it is used Supervised Classification models to classify legitime and non-legitime traffic instances. Different kind of features are used to detect web fraud , from connection to behavioural features. In our case, we focus on the problem of fraudulent web traffic created by bad bots that simulate human behaviour to access web pages resources. The behavioural features that are used in this thesis is mouse movement. In order to develop Machine Learning models, suitable datasets are needed. The availability of datasets is one of the challenges faced in this project. Most of the web mouse movement datasets are private. Furthermore, these datasets use synthetic data for fraudulent mouse movement and ad-hoc made applications such as login pages to collect legitimate mouse movement. This is due to the cost of extracting real data and the difficulty of data labeling. Another challenge faced is the volumetry of the data. Mouse movement data is obtained in huge amounts. However, an extensive pre-processing needs to be carried for this data to be useful.

The significance of this thesis in the research ambit is an extensive review of the web fraud detection literature. In this review it can be found a summary of web fraud, the different methods of fraud detection used in the state of the art and an explanation of the different type of the features that are used to carry the detection. The most important contribution of this project is a public web mouse movement dataset that contains both fraudulent and legitimate samples . This dataset has great significance because, as it has been stated previously, most of the datasets are not public. The key aspect of this project's dataset is the legitimate samples are real people browsing freely the internet, whilst datasets in the literature legitimate samples are created by using a landing web page created specifically for that project.

Originally, this thesis was part of a project collaboration between i2CAT foundation and a third private company from the cybersecurity sector. It was supposed to be developed alongside the main project. This main project consisted on legitimate and non-legitimate web traffic classification using mostly connection features. This third company also had to deliver mouse movement data from legitimate and fraudulent web page traffic. However, it was not delivered on time which caused a delay and a restatement of the objectives. Although their data has not been used, a big scientific and technical background was developed during this collaboration that has been very useful throughout the project.

The remainder of this thesis is structured as follows. Section 2 includes a literature review, including other state of the art fraud detectors. Section 3 includes the methodology followed to create the datasets and an explanation of the models used . Section 4 describes the results of the project including an analysis of the created datasets and the results of the trained models. In Section 5 the budget is presented. The thesis ends in section 6, which are our main conclusion and discussing the future work.

1.1 Statement of purpose

In this section the main goals of the project are stated. The project main goals are:

1. Create a public mouse movement dataset and a fraudulent mouse movement generator. The legitimate data from this dataset is real people browsing freely the internet. The fraudulent data is created by using web bots.
2. Implement two different Machine Learning models that classify legitimate and non-legitimate traffic using mouse movement. The first model is based on the variables statistics whilst the second one uses Deep Learning, specifically Long Short-Term Memory (LSTM) neural networks
3. Compare the Machine Learning models by performing different experiments using the novel dataset generated.

1.2 Requirements and Specifications

In this section the different requirements and specifications are stated. On the one hand, there are no hardware or software specifications as this is a research project with no restrictions. However, all the experiments are developed using commodity hardware and open libraries to allow scalability and ease the use to other researchers. On the other hand, the requirements that must be met are:

- Dataset: The dataset must be public and open access.
- Code: The code created must be in Python and must use public and open source libraries.

1.3 Work Plan

1.3.1 Work structure

In this section the structure of how the project has been carried is detailed.

1. Statement

   1.1 Statement of the thesis scope and objectives

2. Research

2.1 State of the art Research

3. Dataset

    3.1 Dataset Research

    3.2 Dataset Creation

    3.3 Dataset Processing

4. Machine Learning Model

    4.1 ML model creation and training

    4.2 ML model evaluation

5. Documentation

    5.1 Documentation

1.3.2 Work packages

In this section the Work Packages that constitute the project are explained

WP1: Statement of purpose

- Description: The purpose of this work package is to create a working plan for the project. This working plan it must include the scope,the objectives of the thesis and the time plan of the thesis
- Tasks:
  - Internal task T1: Statement of purpose.
- End event: Internal task T1: Statement of purpose complete
- Deliverables: None.

WP2: Research

- Description: The purpose of this work package is to is to review the state of the art of web bot detection. The ideas extracted from the reviewed articles must be used in the following working packages.
- Tasks:
  - Internal task T1: State of the art review.
- End event: Internal task T1: State of the art review complete.
- Deliverables: None.

WP3: Dataset

- Description: The purpose of this work package is to review the different public web mouse movement databases and the creation of a public web mouse movement dataset.

- Tasks:
  - Internal task T1: Dataset research.
  - Internal task T2: Dataset creation.
- End event: Internal task T3: Dataset processing complete.
- Deliverables: Public web muse movement dataset.

WP4: Machine Learning model

- Description: The purpose of this work package is to create two Machine Learning models based on the literature reviwed in WP2 using the dataset created in WP3
- Tasks
  - Internal Task T1: ML model creation.
  - Internal Task T2: ML model validation.
- End event: Internal Task 2: ML model validation complete.
- Deliverables: Experiment results and ML model.

WP5 Documentation:

- Description: The purpose of this work package is to document all the tasks and experiments carried in this project.
- Tasks
  - Internal task T1: Documentation.
- End event. End of project
- Deliverables: Documentation.

1.3.3 Gantt Diagram

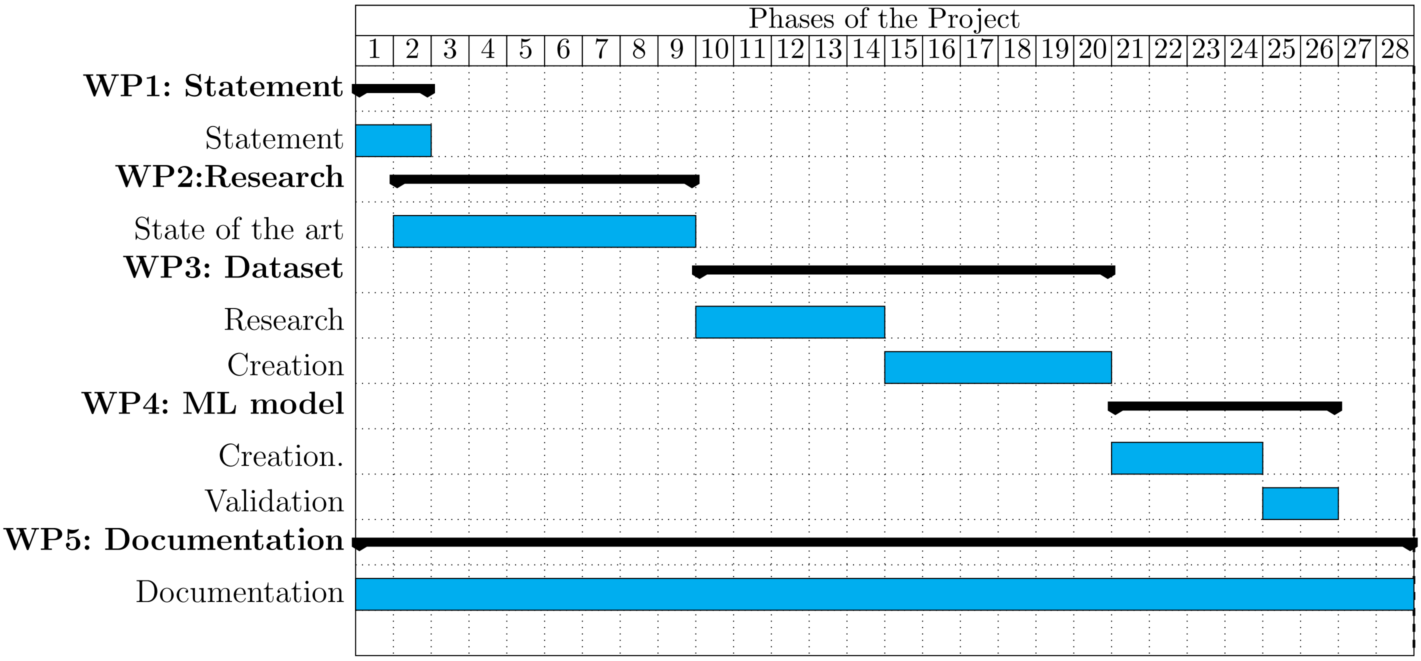


Figure 1: Gantt diagram of the project

1.4 Deviations

In this section the deviations of the project are stated. This thesis has experienced big deviations in WP2. This Work Package is related to dataset research, creation and processing this was due to:

1. Lack of data. The third company data was supposed to arrive on March 2022. However, it did not arrive and a limit date on June 2022 was placed for the arrival of the data. During March 2022 and June 2022 an extensive state of the art research was carried. As the data did not arrive on time, the public databases searched were used to create this project's dataset.
2. Non-legitimate samples creation. As the fraudulent data was not delivered, new synthetic fraudulent data needed to be created. The methods to create this data were researched and applied.
3. Data processing. The legitimate dataset had to be further adjusted as the data did not follow the same schemes proposed by the third company.

2 State of the art of the technology used or applied in this thesis:

In this section it can be found an introduction to the concept of fraudulent traffic and legitimate traffic. Afterwards, a comprehensive review of the different Machine learning techniques found on the literature used to classify legitimate and fraudulent traffic.

Fraudulent traffic or bad bots are software that access web pages to extract or use their resources. One common example is data scraping, where bots try to retrieve accounts and information from the users and the web page. Fraudulent traffic is a big threat that has increased over the years taking up to 27.7% of all the internet traffic. In addition, the complexity of bad bots has drastically increased making them harder to stop. On the other hand, legitimate traffic is composed by humans and good bots. Good bots are used by search engine companies such as Google to map the Internet. These good bots or crawlers index the content of the websites so they can appear when users use the search engine.

In order to detect fraudulent traffic several approaches have been developed being the Turing test system the most known. In this test, a question is asked to the user, if it is answered correctly the user is considered to be a human. If the question is answered incorrectly it is considered a bot. However, this kind of detection systems are being surpassed by the constant increase of the bad bots complexity. In order to detect advanced bots, Machine learning techniques have been implemented in the past decade.

2.1 Introduction to bots: Legitimate and fraudulent traffic

Bots can be defined as softwares created by humans that are used in the Internet and automatically executes tasks. There is a wide range of tasks such as respond messages, retrieve information, execute commands and use social network (C. Harringer2018) [15]. The first bot technology developed was in 1966 a project called ELIZA that was the precursor of the chatbot (A. Godulla et al., 2021)[14]. Harringer distinguishes 13 different types of bot, being social bots and chat bots them most common ones (C. Harringer2018)[15].

Nowadays, bots are very present in the Internet taking up to 42.3% of web traffic [2]. Web Traffic types can be classified as:

1. **Fraudulent traffic**, also known as bad bot.
2. **Legitimate traffic**. This traffic is made up of good bot traffic and human traffic.

Bad bots are programs that do automated task with malicious intentions. They have 21 different uses that are compiled in The Open Web Application Security Project (OWASP) [3]. Some of the most commons threats are:

- **Data scraping** . It consists in illegally acquire data from web pages. It can use compromised accounts to gather this information and gain an insight of the structure of the web page.
- **Data scalping and Data snipping**. Both threats are related to online shopping or

service acquisition. Data scalping is the use of bad bots to automatically purchase the good or service in a way that humans could not undertake manually. Data sniping or auction sniping is related to bid and offer acquisition of services and goods. Bad bots can bid at the last moment denying to the human user another opportunity to bid.

- **Denial of service**. Bad bots use the services of the web page such as memory, CPU, GPU... resembling a human user but they exhaust them. Another type of service denial is Distributed Denial of Service (DDOS) in which big amounts of bad bots connects to a web page overloading the server. The server can no longer give service which denies the usage by human users.
- **Fraud**. It includes Account takeover where the credentials are stolen. Credit card fraud which is the use of fake or stolen credit cards to access to information such as expiration date. Ad fraud where bad bots automatically access to the ads increasing the amount of clicks, this way increasing the revenue.

In 2022 Imperva Bad Bot Report [2] it is reported that in 2021 27.7% of all the traffic is considered fraudulent traffic. Compared with the same study made in 2014, bad bots have increased by 5%. Another type of bad bot classification is by their complexity. Imperva has classified bad bots in 4 different types [2].

- **Simple**: These bots use automated scripts to connect to sites without trying to disguise as a browser.
- **Moderate**: These bots can simulate being a browser when connecting to the site and have the ability to execute JavaScript.
- **Advanced**: These bots copy human behaviour to elude the security. Some human behavioural characteristics are mouse movement, mouse clicks and keybord use. In addition, they use more sophisticated ways to connect to the site such as malware installed in real browsers.
- **Evasive**: These bots are a mix of moderate and advanced bots. Their main characteristics to not be detected by security is change their IP addresses and user agents. In addition, they use what is known as ”low and slow” tactics where they keep a low profile to not be detected by the security. They tend to do a small amount but effective requests in order to not stand out among legitimate connections.

Good bots, also known as crawlers, are beneficial for the user and companies. The World Wide Web is growing constantly. Every day new web pages are created and companies such as Google, Facebook, Bing ... need to keep track of these newly added web pages. To do so, crawlers index the different web pages or, in other words, crawlers create a plan of the internet. This way, the information can be retrieved efficiently and the web pages appear on the search engines. Crawlers when entering a web page collect all the links and recursively continues as it is explained in Chatterjee et al., 2017 [7].

The Imperva bad Bot report attribute to good bots in 2021 the 14.6% of the total web traffic. Compared with the same study made in 2014, good bots have decreased by 21% of the total web traffic. Human traffic has increased, in 2014 was 40.9% and in 2021 was

57.7%. However between 2018 and 20220 human traffic represented more than 62% of the total web traffic. This decrease in 2021 is due to the raise of bad bot traffic [2].

2.2 Bot detection State of the art

In this section it is explained the types of procedures used to classify legitimate traffic from fraudulent traffic found in the literature. Doran D et al,. 2016 summed up the different types of bot detectors in the literature [4]:

- **Syntactical log analysis** It consists on text processing and analyzing server logs that are created by the web traffic. From these logs different parameters such IP addresses or user-agents can be extracted. These two features can be sought in fraudulent IP and user-agent lists. The downside of this technique is that it only detects basic bots. However, the data can be easily extracted.
- **Traffic pattern analysis**. Statistically compare the characteristics and the behaviour of fraudulent traffic in a session against human traffic. One possible comparison is how bots and humans move through the links of a web page. For example, some bots use breath-first or depth-first algorithms when moving through the web page. The downside of this technique is that bots are constantly changing making the patterns obsolete with the need of fast updates.
- **Turing test systems**. A test is added to the web page and the user needs to take it. If the user passes the test it is considered human, if not it is considered bot. The downside of this technique is the cost of implementation and the deterioration the user experience. These tests are thoroughly explained later on this section.
- **Analytical learning techniques.** This technique uses the characteristics of the sessions as features to trains Machine Learning models, being each session a sample. One of the drawbacks is the cost of creating a training dataset. Behaviour data is defined as the interaction of the users with a web page that is being used. Two different types of behavioural features have been found when using analytical learning techinques.

  1. HTTPs and connection based features.
  2. Biometric features.

  This project is focused on creating a public dataset based on mouse movement, which is biometric behavioural feaure. Afterwards Machine Learning models will be trained using this dataset to classify between legitimate and fraudulent traffic.

The key aspects that are considered for each reviewed article are:

1. **Transparency**: The users do not realise of the existence of a program collecting data. For example, one of the most common procedures that is not transparent and it is used as a protection for malicious bots is CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart). As it is explained in V. Prakash Singh et al., 2014 [8] CAPTCHA is a bot detection based on Turing test. In this case the program acts like the judge and delivers the user with a test. If

the user pass this test it means that it is human otherwise it is a bot. Some of the most typical CAPTCHAS are text and image identification, puzzle solving and audio comprehension. However, advances in Machine Learning , computer vision and NLP (Natural Language Proccessing) made this kind of tests surpassable by advanced bots.

One of the first successful text based CAPTCHAs solver is E.Bursztein et al., 2014 [9]. This article presents a Machine Learning algorithm for text based CAPTCHAs that segmentated and identified the letters. Recent works such as Wang J et al., 2019 used deep Convolutional Neural Networks (CNN) to solve this problem obtaining better results.

The first image based CAPTCHAS were based on identifying an object amongst a group of pictures. It was solved by using reverse search of the images and getting their semantic description as in S.Sivakorn et al., 2016 [11]. Recent image based CAPTCHAs are based on object detection. In I.Hossen et al., 2019 [12] use an advanced Deep Learning detection architecture called YOLOv3 to solve them.

One possible solution to avoid CAPTCHAs being solved by bots is by making them harder by adding noise. This is not an optimal solution as humans may not be able to solve them. In addition, CAPTCHAs can be hard to solve for some users as they are not accessible for everybody. For example, text or image identification for visually impaired users can be hard. The same happens with audio based CAPTCHAs with hearing impaired users. Finally, CAPTCHAs can be time consuming and give a bad use experience on the web. In the view of the foregoing, a transparent and accessible way of bot detection can be useful.

2. **Data**: The nature of the data used in the study. It can be real, synthetic or created ad hoc for the project. Amongst all the reviewed literature, few cases used real data extracted from web pages. Most of the articles uses ad-hoc data for the project. This means that a web page is created with the purpose of collecting data. A group of people enter that web page for a period of time to create data labeled as legitimate traffic. Data labeled as fraudulent traffic is created by using bots on that web page. In addition, most of the data is private and cannot be accessed. In this project the dataset created by Kiliç et al,.2021 called Bogazici mouse dynamics dataset is used. This dataset contains real user mouse movement when using a browser [16]. The preprocessing of the dataset is later explained.

2.2.1 HTTPS based features

HTTP based traffic classification uses HTTP data to classify the traffic. This kind of data is acquired when the user connects to the platform. These data is obtained in a transparent way for the user. It is a key aspect as the user does not have to interact with the page.

In order to exemplify HTTP features, previous definitions needs to be done. First HTTP (Hypertex Transfer Protocol) is a protocol used in the transmission of documents. An HTTP request is an action made by the client to a host in a server to access resources of, for example, a web page. Finally, as it is defined in Suchacka et al.,2021 [5]a session is a sequence of HTTP requests with the same IP address and same user-agent. The time span between two requests should be less than 30 minutes.

Some examples of HTTP features used in the literature are: Total HTTP requests done in a session, total amount of HTTP GET/POST/HEAD requests and session time amongst others in Iliou et al.,2019 [17].

Amongst the literature two different main types o machine learning have been found. Supervised and unsupervised learning methods have been reviewed.

1. **Supervised Learning**: In Doran et al., 2016 [4] data is extracted in sessions. For each request several variables are extracted. In order to classify the sessions a DTMC (Discrete Time Markov Chain) is used. DTMC are formed by several states, in this case the states correspond to the different requests made. They sustain that bots and humans do not follow the same request pattern.

   In Suchacka et al., 2021 [5] detection the data is also gathered using the concept of sessions. In addition, new variables extracted from HTTP requests are extracted. However, sessions are classified using MLP (MultiLayer Perceptron). The features extracted from each request are used as input of the MLP and in order to exploit the relation between the different requests in a session a probabilistic sequential analysis is used.

   Both aforementioned articles are ”on the fly”, which means that they can be used in real time and there is no need to process the whole session in order to classify it. In both articles the data is gathered in an e-commerce web, thus making the data real. In order to label the data external data is used. There are several lists of user-agents and IP adresses that are already labeled by the community as bots ,such as abuseIPDB [25] and BotsVSBrowsers [26]. This data can be difficult and costly to label. Finally, both articles are completely transparent for the user.

2. **Unsupervised Learning**: Labeling the data is expensive and time consuming. There are approaches such as Rovetta et al,. 2020 [6] that uses unsupervised learning. This way labeling the data is avoided making the process less expensive and more efficient. Similar results are obtained as supervised models are but it cannot be used ”on the fly”. the data is also collected from a real e-commerce web.

2.2.2 Biometric based traffic classification

As it is defined in V.Matyas et al,. 2003 ”Biometrics are automated methods of authentication based on measurable human physiological or behavioral characteristics” [18]. Physiological biometrics are measures extracted directly from the human body. Some examples are fingerprint scan, iris scan and facial scan [20]. On the other hand, behavioural biometrics are measurements based on human actions. Some examples found in Yampolskiyetal et al ., 2008 are gaig, voice, key stroke and, in this project’s case it is used mouse dynamics [19].

The use of biometrics is widely spread for person identification and person authentication.

For example in border controls or even to unlock your phone. Mouse dynamics can also be used as intruder detector, comparing the mouse features of the owner of the computer with the person using it. In this project mouse dynamics are used to detect weather the user in a web page is a bot or not. Below, the articles reviewed for this projects are explained.

In Ang Wei et al,. 2019 a database is created ad-hoc for the project. It consists in a login landing page that records the mouse movements. This login page has three different fields to fill. Human samples are collected by volunteers that log in in this web page. Bot data is collected using 4 types of bots that also log in in this page. These types of bots follow 4 different types of movements: linear, curve, polyine and semi-straigh line. The mouse features are collected in sequences that are made of consecutive mouse events. These sequences have the following form: $[(x_1, y_1, t_1), ..., (x_n, y_n, t_n)]$, being x and y the pixel coordinate and t the timestamps of the mouse event. The data is further processed creating an image of the mouse sequence. Finally a CNN based model is used to classify the sequences [22]. In H Niu et al,. 2021 the same database is used. However, instead of creating an image out of the movement, the velocity is calculated creating a sequence such as $[(dx_1, dy_1, dx_1/dt, dy_1/dt), ..., (dx_n, dy_n, dx_n/dt_n, dy_n/dt_n)]$. These sequences are then passed through LSTM based models [21]. Out of both articles reviewed, the later obtains better results and is more efficient as the data processing is less costly. Finally, A.Acien et al,. 2021 [23] created a new database based on Shen et al., 2014 [24]. The legitimate samples consists on volunteers repeating a image CAPTCHA type test. When doing this test the mouse movement is captured. The fraudulent samples are created as in Ang Wei [22] and also using GANs (Generative Adversial Netwroks) to create synthetic samples.

Amongst the reviewed mouse movement articles all the data is collected in a non transparent way. Retrieving the data is transparent because users do not realise that the data is being collected. However, users do have to solve CAPTCHAs or login landing pages making the data collection non-transparent. In Table 1 there is a summary with the characterstics of each article.

Article	Data	Features	Transparency	ML Model
Doran [4]	Real	HTTP	Yes	DCTM
Suchacka [5]	Real	HTTP	Yes	MLP+Seq analysis
Rovetta [6]	Real	HTTP	Yes	K-means
Wei [22]	Real ad-hoc	Mouse	No	CNN
Rovetta [21]	Real ad-hoc	Mouse	No	LSTM
A.Acien [23]	Real ad-hoc+GAN	Mouse	No	LSTM
Our contribution	Real	Mouse	Yes	LSTM and statistical model

Table 1: Summary table of the reviewd articles

To conclude, after reviewing the literature, HTTP features seem to be more convenient when detecting bots as they are always transparent. In addition, real fraudulent data can be extracted as it can be labeled by using IP and user agents blacklists, whereas fraudulent mouse movement data in the literature is created by using bots in ad-hoc landing pages. Real legitimate HTTP is also used and can be labeled whilst legitimate mouse movement data in the literature is always created in ad-hoc landing pages. Finally, the objective of this thesis is to create a dataset in which the legitimate data are moving the mouse freely while browsing. This is of great significance as all the previous datasets do not contain this ind of legitimate data.

3 Methodology

In this section is it explained the methodology used throughout the project. In Fig.2 it can be found the diagram of the complete methodology framework. The main parts are:

- **Dataset Creation**. It is explained how the legitimates sequences are extracted and how the non-legitimate sequences are created. The analysis and explanation of the dataset is explained in the Results section.
- **Features Engineering**. It is explained the different features extracted. In total two different set of features are extracted. The first one are scalar valued features, used to train a Decission Tree based model. The second set of features are vector like features and are used to train a Deep Learning model.
- **Model**. It is explained the two different models that have been used. The first model is based on a Decission Tree algorithm called LightGBM and the second one is based Deep Learning, specifically on Long short-term memory LSTM layers.
- **Validation**. It is explained the different metrics used to evaluate the models. This metrics are True Positive Rate (TPR), True Negative Rate (TNR), and Accuracy.

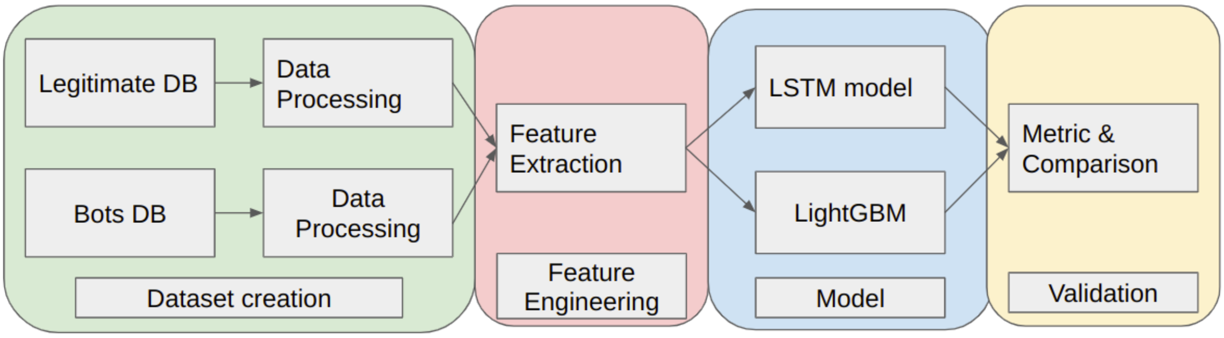


Figure 2: Methodology framework.

3.1 Dataset creation

One of the key aspect of this project is to create a free mouse movement public dataset that contains both legitimate and non-legitimate samples. Reviewed datasets are either private or the mouse movement is not free. This project's datasets is created in two steps:

- **Legitimate mouse movement**: This part of the dataset was created by using Bogazici mouse dynamics created by AA Kiliç et. al[16]. It is a public dataset that contains real free human mouse movement.
- **Non-legitimate mouse movement**: This part of the dataset was created by using bots. This bots were adapted from two different GitHub repositories: bezmouse by vicentbavitz [27] and self-driving-desktop by hofstadter-io [28]

3.1.1 Legitimate mouse movement

The legitimate sequences of the dataset are extracted from the public database Bogazici mouse dynamics [16]. It contains free mouse movements of 43 users recorded during 10 days. This database was meant to be used to train models for intrusion detection. It labels each sample depending on what the user is doing. For example, if the user is coding in an IDE it will be labeled as *Development*; if the user is browsing the internet it is labeled as *Browsing*. In addition, this database also contains the use of the muse buttons and their actions. This database has been chosen because it has been developed in a laboratory environment, created with real people moving the mouse freely.

In this project only the *Browsing* category is considered as this project is focused in web usage of the mouse. *Browsing* samples take up to 51% of the whole database. In addition, only the mouse movement data is considered whilst the buttons data is not. One inconvenient found in this database was that the size of the monitors and the amount of monitors used by the users is not indicated. The size of monitors from users with a single monitor can be discovered by looking a the maximum and minimum coordinate values. However, if more than one monitor was used negative values appeared and the sizes could not found. To solve this problem only data coming from one monitor and a size of $1920x1080$ is used.

In order to explain the data processing some definitions concerning the database are needed:

- **Movement** Movement is defined as all the samples between two mouse buttons clicks.
- **Sequence** Consecutive samples with a duration of 6 seconds. Movements can be made of 1 or more sequences.
- **Sample** A sample is defined as a pair of coordinates $[x, y]$ in a given time t.

In the public database, each sample is taken each $1ms$. It is a very high sampling rate that leads to a very high amount of data. In order to reduce the amount of data a resampling is made each $100ms$. Finally, the article by Ang Wei [22] the length of each sequences is set to 60 samples.

Another problem faced is that human samples tend to be empty, this means that most sequences did not have any movement. To solve this problem first we have to define the position differentials $d_x[n]$ and $d_y[n]$ Eqn. (7). Only sequences with an AoM higher than 50% are used.

$$\begin{aligned} d_x[n] &= x[n] - x[n-1] \\ d_y[n] &= y[n] - y[n-1] \end{aligned} \tag{1}$$

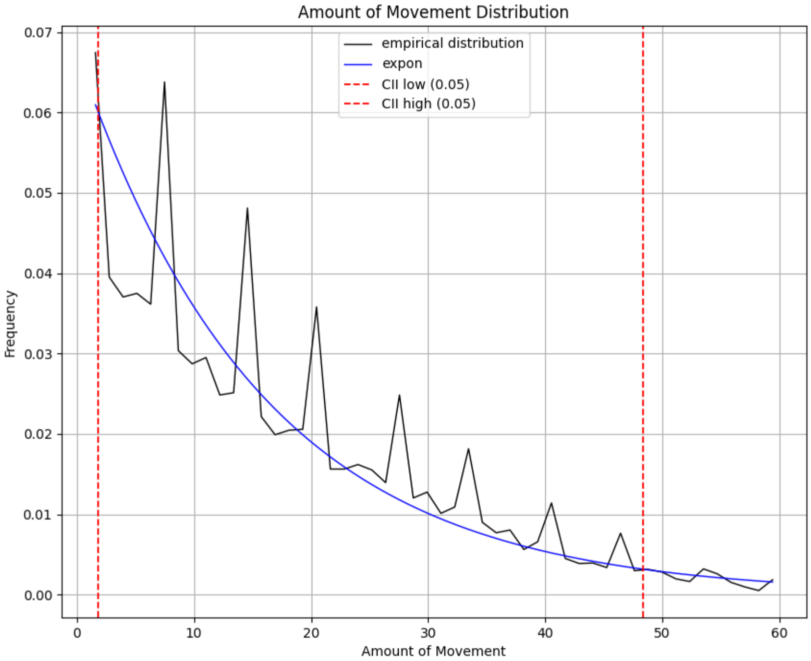
We define *AoM (amount of Movement)* as the percentage of $d_x[n[$ and $d_y[n]$ different to zero Eqn. (2).

$$\%samples \begin{cases} d_x[n] \neq 0 \\ d_y[n] \neq 0 \end{cases} \quad \forall 0 < n < N \tag{2}$$

Finally, the probability density function that AoM follows is estimated using Residual Sum of Squares (RSS). AoM follows an exponential function such as Eqn. (3) and it can be observed in Fig. 3 .

$$f(x) = 15.4 * e^{-15.4x} \tag{3}$$

Figure 3: Probability distribution function of AoM.



3.1.2 Non-Legitimate mouse movement

The non-legitimate part of the dataset is created adapting github bot repositories. The repositories used are

- Self-driving-desktop by hofstadter-io [28]. This repository follows text input orders to create mouse movement. It was adapted so it returns sequences of samples of the desired length. This repository can create sequences that follow different types of movements such as linear and quadratic movments.
- Bezmouse by vicentbavitz [27]. This repository was barely adapted as it already returns a sequence of samples following Beziere curves. Beziere curves are parametric curves that given a set of discrete control points it creates a continuous curve.

In Wei et al[22] bot mouse movements can be of 4 different types: that are Straight lines, semi-straight lines, regular curves and irregular curves. In our case, to mimic [22] we use 4 different type of movements:

- **Linear**. It creates an straight line between two points. The bot moves with constant velocity.
- **Quadratic**. It follows a quadratic curve between two points. Two types are used: Quadratic with higher acceleration at the beginning and quadratic with higher acceleration at the end.
- **Exponential**. It follows an exponential curve between two points. Two types are used: Exponential with higher acceleration at the begining and Exponential with higher acceleration at the end.
- **Bezier**. It creates a Beziere curve between two points given a set of control points. The bot moves with constant velocity.

In order to create this sequences the initial and the final coordinate are created randomly following a uniform distribution. Where x coordinate can have values $0 < x < 1919$ and the y coordinate can have values between $0 < y < 1019$. The length of the sequences is the same as the legitimate ones, which is 60. The AoM of the non-legitimate sequences is always 100%, this means that there is always movement. This does not follow the AoM distribution showed in Eqn.2 and Fig. 3. In order to create similar sequences, stops are added following the AoM distribution of the legitimate samples between $50\% < AoM < 100\%$. This stops add more complexity to the dataset as it will be show in the Results.

3.2 Features Engineering

In this section the different features extracted for each sequence are explained. Two different sets of features are created. The first set of features are used to train Deep Learning models, specifically Recurrent Neural Networks (RNN). The second set of features are meant to train Decission Tree based Mahine Learning models. Decission Tree features are scalar value whilst Deep learning features are vectors of fixed length.

3.2.1 Deep Learning features

Deep Learning features are extracted for each sequences and they are similar to [22]. As it has been explained, a sequence consists on 60 pair of coordiantes such as $[(x_1, y_1), ..., (x_{60}, y_{60})$. In order to train the LSTM we calculate the differential of movement for each sample creating $[(0, 0), (dx_2, dy_2, dx_2), ..., (dx_{60}, dy_{60}]$. As it can be observed the first differential is (0,0) because it cannot be calculated as there is no previous sample. Thus making the first differential not usable. This creates a sequence such as $[(dx_2, dy_2,), ..., (dx_{60}, dy_{60})]$ with length 59. In Wei et al [22], this features are a bit different $[(dx_1, dy_1, dx_1/dt, dy_1/dt), ...,$ $(dx_n, dy_n, dx_n/dt_n, dy_n/dt_n)]$. As it can be observed each sample contains both differentials and the velocities. In our case calculating the features d_x/d_t does not make sense as our time differential dt is constant (100ms). This would only add a constant multiplicative factor to the differentials.

3.2.2 Decission Tree features

Decission Tree features are also calculated for each sequence. However, this features are a scalar value and not a vector as it was in Deep Learning features. In this case 12 different features are calculated per each sequence of longitud N.

- $V_x mean$ and $V_y mean$. They are defined as the mean of the differentials in each direction.
$$V_j mean = \frac{\sum_{i=1}^{N}(d_j[i])}{N} \quad j = [x, y] \tag{4}$$
- $Vmean$ It is defined as the mean of module of d_x and d_y
$$\begin{aligned} Vmod[i] &= \sqrt{d_x[i]^2 + d_y[i]^2} \\ Vmean &= \frac{\sum_{i=1}^{N} Vmod[i]}{N} \end{aligned} \tag{5}$$
- $Vmax$, $V_x max$ and $V_y max$ They are defined as the maximum value of each velocity.
- $A_x mean$ and $A_y mean$. They are defined as the mean differentials of each velocity.
$$A_j mean = \frac{\sum_{i=1}^{N}(d_j[i] - d_j[i-1])}{N} \quad j = [x, y] \tag{6}$$
- $Amean$ It is defined as the mean of module of the differentials of d_x and d_y
$$\begin{aligned} Amod[i] &= \sqrt{(d_x[i] - d_x[i-1])^2 + (d_y[i] - d_y[i-1])^2} \\ Amean &= \frac{\sum_{i=1}^{N} Amod[i]}{N} \end{aligned} \tag{7}$$
- $Amax$, $A_x max$ and $A_y max$ They are defined as the maximum value of each acceleration.

3.3 Models

In this section the two different models used are explained and how they have been trained and tested. The first model is based on Gradient Boosting Decission Trees called LightGBM. The second one is a Deep Learning model that uses RNN layeres, specifically LSTM layeres. Both models will be later compared in the Result section.

3.3.1 LightGBM

GBDT (Gradient Boosting Decision Tree) is a machine learning algorithm with some successful implementations such as XGBOOST by T Chen et al [30]. GBDT are very efficient, accurate and with a high interpretability. However, GBDT algorithms tend to be inefficient when the feature dimension and the data size is high. LightGBM by Guolin Ke et al [29] optimizes the GBDT algorithm and solves the problems mentioned obtaining in most cases the same accuracy. In our case, there is not much difference between using XGBOOST and LightGBM as the feature dimension and dataset size are small. LightGBM is chosen because is a more recent implementation and optimized version of the GBDT algorithm. In order to train the LightGBM model the dataset is divided in train and test dataset. The train dataset is the 75% of the dataset whilst the test dataset is the 15% remaining data. In addition the training is carried using a cross-validation technique called Stratified KFold . Stratified KFold divides the train dataset in Kfolds that preserves the percentage of samples for each class. In this case the number of folds K is 5. In order to search the best hyperparameters a Grid Search is used. Grid Search creates all the possible hyperparameters combinations given a set of hyperparameters. In our casethe hyper parameters tuned are:

- Learning Rate. It is the boosting learning rate
- Max depth. It is the maximimum depth of base learners. When negative there is no limit.
- Min data in leaf. Minimum amount of data in a leaf. If the amount of data is lower than the minimum that leaf will stop bifurcating.

3.3.2 LSTM

Long short-term memory (LSTM) is an artificial neural network with feedback connections. This feedback connections allows to exploit the time characteristics of the inputs. For example, LSTM are widely used in speech and video, which are types of data based on time-series. In our case, mouse movement is based on time series which makes it suitable to use LSTM in this project. Time based characteristics cannot be exploited using normal feed forward networks. In Fig 4 it is shown the architecture of the model used. The main characteristics are:

- LSTM layer: N is the number of LSTM layers with hidden size 50 is the Number of the hidden size.
- Rectified Linear Unit (ReLu) activation Layer. It is a widely used activation layer in the training of Deep neual Netwroks.
- Forward fully connected Layer (FC): It has a hidden size of. It is used to reduce the dimesionality to 1 in order to take a decision.
- Sigmoid Activation Layer. This layer is used in used in binary classification.

In order to train the model the dataset partition has the same train and test dataset as in the LightGBM model. To validate the model the train dataset is also divided into train and validation. The validation dataset is the 15% of the train dataset. Finally, Binary Cross Entropy Loss Eqn.8 and an ADAM optimizer are used.

$$BCE(\boldsymbol{y}, \hat{\boldsymbol{y}}) = -\sum_{i=0}^{1} y_i \log(\hat{y}_i) \tag{8}$$

Where $\boldsymbol{y}$ is the real class and $\hat{\boldsymbol{y}}$ is the estimated class.

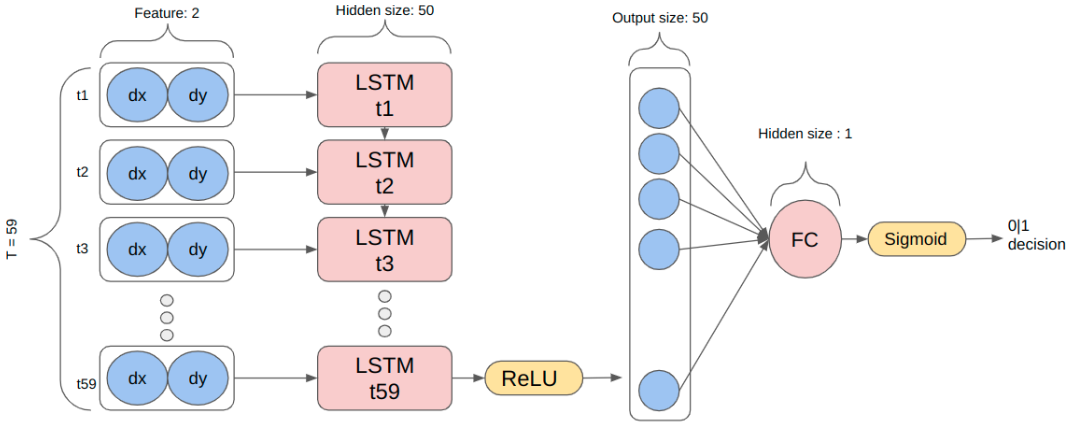


Figure 4: LSTM model.

3.4 Validation

The metric used in this project are TPR Eqn.9 (True positive Rate) and TNR Eqn.10 (True negative rate) and Accuracy Eqn.11. Both models compute the same metrics and are used to compare the performance between models. In addition, TPR and TNR are also used in [22] and are also used to compare our project with theirs.

- **TPR** is defined as the proportion of correct predictions in predictions of positive class. In our case, positive class is non-legitimate traffic Eqn.9.

$$TPR = \frac{TP}{TP + FN} \tag{9}$$

- **TNR** is defined as the proportion of correct predictions in predictions of negative class. In our case Negative classes are legitimate traffic Eqn.9.

$$TNR = \frac{TN}{TN + FP} \tag{10}$$

- **Accuracy** is defined as the proportion correct predictions and the total amount of predictions . In our case it is used to validate the LSTM model and compare the LightGBM and LSTM model Eqn.11.

$$Accuracy = \frac{\#CorrectPredictions}{\#predictions} = \frac{TP + TN}{TN + TP + FP + FN} \tag{11}$$

4 Results

In this section the results are explained. This section is divided in two parts.

- **Dataset analysis**: It is explained the characteristics of the dataset. In this project two different datasets have been created.
  - **Base dataset:** In this dataset the non-legitimate sequences do not contain stops.
  - **Stop dataset:** In this dataset the non-legitimate sequences are the same as in the base dataset but they contain stops.

  Both datasets contain the same legitimate samples. The main idea of creating the Stop dataset was to recreate a more human behaviour in the non-legitimate samples. In Table.2 there is a summary of the datasets.

- **Model results and comparison**: It is explained and compared the results obtained with the LSTM and LightGBM models. These two models are trainned and tested with the aforementioned datasets.

4.1 Dataset analysis and creation

In this thesis two datasets have been created. The first dataset created is called Base dataset. The legitimate sequences are extracted from the Bogazici mouse dynamics dataset [16] as it have been explained in the Methodology. In total there are 2385 legitimate labeled sequences. The non-legitimate sequences have been created using bots extracted from GitHub repositories. These bots follow 4 different types of movements which are: Linear, quadratic, exponential and bezier. For each type of movement it is created 397 samples which makes a total of 2382 non-legitimate sequences. In total the dataset has 4767 sequences with an almost 50-50 balance between legitimate and fraudulent labels.

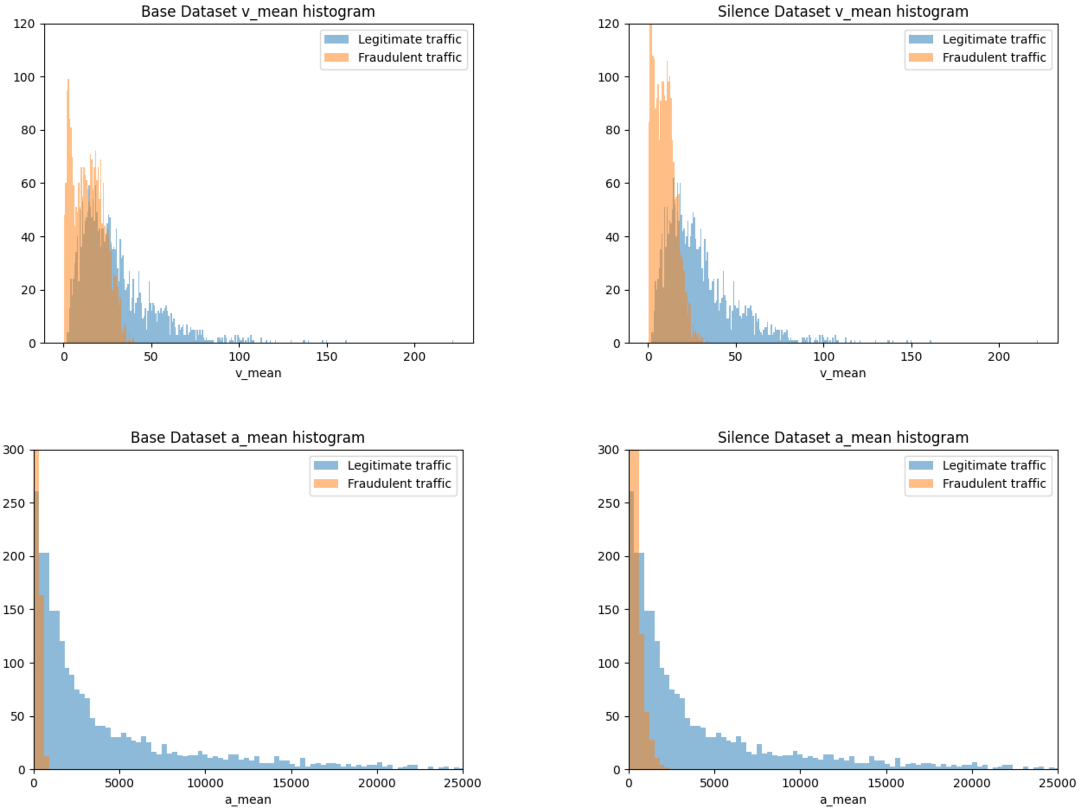
The second dataset is called Stop dataset. This dataset is created to increase the complexity of the bots. This dataset has exactly the same legitimate samples as Base dataset. The non-legitimate samples are the same as in the Base dataset but stops in the movement are added. This makes this dataset to have the exact same size as the Base dataset.

As it can be observed in Table.2, the mean velocity is lower in the Fraudulent sequences and is even lower in the Fraudulent sequences with stops. This is because the initial and end coordiantes of the movements are created randomly. This can lead to create short and slow mouse movements. When stops are added in the fraudulent sequences it creates even slower movements. The mean acceleration is way bigger in the Legitimate sequences than in the Fraudulent sequences. This is because Linear and Bezier movements do not have acceleration as they have constant velocity. When stops are added the mean acceleration fraudulent sequences increases gap closes a little bit. We can further analyse the mean velocity and mean acceleration by looking at the histograms in Table. 3. It can be observed that when adding silences to the fraudulent sequences the the mean velocity decreases whilst he mean fraudulent acceleration increases.

Table 2: Dataset characteristics

Sequences	**# Sequences**	$\overline{AoM}$	$\overline{Vmean}$	$\overline{Amean}$	$\overline{Vmax}$	$\overline{Amax}$
Legitimate	2385	66%	29.64	4813	232.7	67793
Fraudulent	2382	100%	14.13	68.40	48.79	751.2
Fraudulent with stops	2382	65%	9.36	220.2	46.61	2869
Dataset	4767					

Table 3: Acceleration and velocity distributions



4.2 Models results and comparison

In this thesis two types of models have been used LSTM and LightGBM. Several models have been trained using the different datasets. In Table.4 there is a resume of the LightGBM based models and in Table.5 there is the results of the LSTM models based.

First let's start with the LightGBM models and the meaning of each column in the Table. 4. The first column indicates the name of the model. The second column indicates the dataset used in the training. The third column indicates the features used. When Features is velocity, only the features related with the velocity are used. When Features is All, both acceleration and velocity features are used. Finally, the fourth and fifth columns are the results when testing the model with the Base dataset and the Silence dataset.

We start comparing LightGBM (1) and LightGBM (2). Both models obtain very high results on the Base test being LightGBM (2) the one with better results. This is because (1) only uses velocity related features that are not as discriminant as the acceleration in the Base dataset. However, (1) generalizes better than (2) when being tested in the Silence dataset. This is because acceleration features are very discriminant in (2). Comparing LightGBM (3) and LightGBM (4) we can observe that both obtain similar results in the silence dataset test being (4) slightly better. However, (4) generalizes better than (3) as it obtains better results when tested with the base dataset.

The model that obtains the best classification in the Base test is (2) closely followed by (4). This is remarkable as (4) is not trainned with the base dataset which indicates a good generalization. In addition (4) obtains the best results in the Silence dataset test. It can also be observed that the results obtained in (3) and (4) with the Stop test dataset are lower than the results obtained by (1) and (2) in the base dataset. This indicates that the Stop dataset is more complex and human like than the base dataset.

We continue with the LSTM models in Table. 5. The table has similar columns than 4 detailed previously. The main differences are that it includes a #Layers and #Epochs that indicates the number of LSTM layers and epochs used in each model. It can be observed that all the LSTM models obtain very high results on the Base dataset test which is very similar to the LightGBM models (1), (2),(4). However, (5) LSTM obtains better results than (1) and (2) LightGBM when tested on base dataset. This means that LSTM in this case generalizes better than the other two models. Finally, the best results obtained over all the models in the silenced dataset is obtained by (7),

To conclude, it can be said that the Silence dataset is more human like than the base dataset as it is harder to classify by all the models. LSTM based models obtain better results than the LightGBM based models. This can be because the features extracted for the LightGBM are not good enough or because LSTM is more suitable for this job and can find hidden patterns and features.

Table 4: LightGBM model results

Model	Dataset train	Features	Base test	Silence test
(1) LightGBM	Base dataset	Velocity	**TRP: 0.980** **TNR: 0.953** **Acc: 0.966**	TPR: 0.661 TNR: 0.986 Acc: 0.787
(2) LightGBM	Base dataset	All	TPR:0.997 TNR: 1.000 Acc: 0.998	TPR: 0.278 TNR: 1.000 Acc: 0.435
(3) LightGBM	Silence dataset	Velocity	TPR: 0.830 TNR: 0.968 Acc: 0.8914	TPR: 0.921 TNR:0.926 Acc: 0.9238
(4) LightGBM	Silence dataset	All	TRP: 0.997 TNR: 0.969 Acc: 0.983	**TPR: 0.947** **TNR: 0.927** **Acc: 0.937**

Table 5: LSTM model results

Model	#Layers	#Epochs	Dataset train	Base test	Silence test
(5) LSTM	1	2	Base dataset	**TRP: 1.000** **TNR: 0.981** **Acc: 0.990**	TPR: 0.661 TNR: 0.981 Acc: 0.819
(6)LSTM	1	3	Silence dataset	TPR: 1.000 TNR: 0.961 Acc: 0.981	TPR:0.991 TNR: 0.961 Acc: 0.976
(7) LSTM	3	7	Silence dataset	TPR: 1.000 TNR: 0.973 Acc: 0.986	**TPR: 1.000** **TNR: 0.973** **Acc: 0.986**

5 Budget

The Budget used in this Thesis is devided in material cost and salaries.

- Material: A laptop provided by i2CAT, an MSI Pulse GL66. All the project was developed in this computer. The budget of the computer is shown in Table.6

Table 6: Computer budget

Concept	**Budget**
i2CAT PC	1450€

- Salary: I did this thesis as junior project engineer. This thesis consists of 12 ECTS, each ECT is 25 hours of work Table. The salary is shown in Table.7

Table 7: Salary budget

Concept	ECTS	Hours	Price	Total
Salary	12	300	45€	13500€

The final total estimated budget is showning Table.8:

Table 8: Final budget

Concept	**Budget**
Computer	1450€
Salary	13500€
Total	14950€

6 Conclusions and future development:

This thesis has achieved the creation of an open access mouse movement dataset and the delivery of an easy to use tool to create non-legitimate sequences. In addition, the complexity of the dataset has evolved throughout the thesis by adding stops in the sequences. The dataset can be used as a reference benchmark for novel legitime/non-legitime detection algorithms. However, this dataset can be easily improved. For example, real web fraudulent mouse movement should be added to the dataset. The complexity of the fraudulent data can be increased by implementing Generative Adversary Networks(GANs) to generate synthetic sequences from legitimate data.

Regarding the models created, two type of models have been successfully created capable of classify non-legitimate from legitimate mouse movement. The LSTM based models had the best results. Future work should include the study of more advanced Deep Learning techniques such as Transformers. Finally, a further study of new extracted features from the dataset should be done to increase the results of the LightGBM based model. Future work should include the study of more advanced Deep Learning techniques such as Transformers.

In conclusion, I think that the code and dataset produced is a good starting point for future research in i2CAT Foundation and for the research of applied AI in cybersecurity and fraud detection environment.

References

[1] https://www.unit21.ai/blog/fraud-detection-prevention-for-financial-organizations.

[2] Imperva. 2022 imperva bad bot report evasive bots drive online fraud. `https://www.imperva.com/resources/reports/2022-Imperva-Bad-Bot-Report.pdf`.

[3] OWASP. Owasp automated threats to web applications. `https://owasp.org/www-project-automated-threats-to-web-applications/`.

[4] Derek Doran and Swapna S. Gokhale. An integrated method for real time and offline web robot detection. *Expert systems*, 33(6):592–606, 2016.

[5] Grażyna Suchacka, Alberto Cabri, Stefano Rovetta, and Francesco Masulli. Efficient on-the-fly web bot detection. *Knowledge-Based Systems*, 223, 7 2021.

[6] Stefano Rovetta, Grażyna Suchacka, and Francesco Masulli. Bot recognition in a web store: An approach based on unsupervised learning. *Journal of Network and Computer Applications*, 157, 5 2020.

[7] Soumick Chatterjee, Asoke Nath, and M Sc Student. Auto-explore the web-web crawler article in international journal of innovative research in computer and communication engineering. 2017.

[8] Ved Prakash Singh and Preet Pal. Survey of different types of captcha. *International Journal of computer science and information technologies*, 5(2):2242–2245, 2014.

[9] Elie Bursztein, Jonathan Aigrain, Angelika Moscicki, and John C Mitchell. The end is nigh: Generic solving of text-based {CAPTCHAs}. In *8th USENIX Workshop on Offensive Technologies (WOOT 14)*, 2014.

[10] Jing Wang, Jiaohua Qin, Xuyu Xiang, Yun Tan, and Nan Pan. Captcha recognition based on deep convolutional neural network. *Mathematical Biosciences and Engineering*, 16:5851–5861, 2019.

[11] Suphannee Sivakorn, Iasonas Polakis, and Angelos D Keromytis. I am robot: (deep) learning to break semantic image captchas; i am robot: (deep) learning to break semantic image captchas. *2016 IEEE European Symposium on Security and Privacy (EuroSP)*, 2016.

[12] Imran Hossen, Yazhou Tu, Fazle Rabby, Nazmul Islam, Hui Cao, and Xiali Hei. Bots work better than human beings: An online system to break google's image-based recaptcha v2.

[13] Jan Dennis Gumz und Resa Mohabbat Kar. Social bots. `https://www.oeffentliche-it.de/-/social-bots`.

[14] Alexander Godulla, Melanie Bauer, Julia Dietlmeier, Annika Lück, Maike Matzen, and Fiona Vaaßen. Good bot vs. bad bot: Opportunities and consequences of using automated software in corporate communications. 2021.

[15] Claus Harringer. „good bot, bad bot“?: Zur problematik von bot-ontologien. *Information - Wissenschaft Praxis*, 69(5-6):257–264, 2018.

[16] Arjen Aykan Kılıç, Metehan Yıldırım, and Emin Anarım. Bogazici mouse dynamics dataset. *Data in Brief*, 36:107094, 2021.

[17] Christos Iliou, Theodoros Kostoulas, Theodora Tsikrika, Vasilis Katos, Stefanos Vrochidis, and Yiannis Kompatsiaris. Towards a framework for detecting advanced web bots. In *Proceedings of the 14th International Conference on Availability, Reliability and Security*, pages 1–10, 2019.

[18] V Matyas and Z Riha. Toward reliable user authentication through biometrics. *IEEE security privacy*, 1(3):45–49, 2003.

[19] Roman V Yampolskiy and Venu Govindaraju. Behavioural biometrics: a survey and classification. *International Journal of Biometrics*, 1(1):81–113, 2008.

[20] Shaveta Dargan and Munish Kumar. A comprehensive survey on the biometric recognition systems based on physiological and behavioral modalities. *Expert Systems with Applications*, 143:113114, 2020.

[21] Hongfeng Niu, Jiading Chen, Zhaozhe Zhang, and Zhongmin Cai. Mouse dynamics based bot detection using sequence learning. In *Chinese Conference on Biometric Recognition*, pages 49–56. Springer, 2021.

[22] Ang Wei, Yuxuan Zhao, and Zhongmin Cai. A deep learning approach to web bot detection using mouse behavioral biometrics. In *Chinese Conference on Biometric Recognition*, pages 388–395. Springer, 2019.

[23] Alejandro Acien, Aythami Morales, Julian Fierrez, and Ruben Vera-Rodriguez. Becaptcha-mouse: Synthetic mouse trajectories and improved bot detection. *Pattern Recognition*, 127:108643, 2022.

[24] Chao Shen, Zhongmin Cai, Xiaohong Guan, and Roy Maxion. Performance evaluation of anomaly-detection algorithms for mouse dynamics. *Computers & security*, 45:156–171, 2014.

[25] https://www.abuseipdb.com/.

[26] https://botsvsbrowsers.org/.

[27] https://github.com/vincentbavitz/bezmouse.

[28] https://github.com/hofstadter-io/self-driving-desktop.

[29] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.

[30] Tianqi Chen, Tong He, Michael Benesty, Vadim Khotilovich, Yuan Tang, Hyunsu Cho, Kailong Chen, et al. Xgboost: extreme gradient boosting. *R package version 0.4-2*, 1(4):1–4, 2015.