University of Trento

Department of Mathematics

Master's Thesis in Mathematics

Marked Temporal Point Processes for simulating and capturing coordinated behaviour campaigns

A model for enhancing disinformation detection

Supervisors:

Agostinelli Claudio
Iannucci Letizia
Kivelä Mikko
Gallotti Riccardo

Candidate:

Muratore Elisa

Academic Year 2023/2024

To Claudia and Marco,
for always believing in me,
supporting me unconditionally,
and being by my side every step of the way.

Introduction

In recent years, social media has become increasingly prevalent in our lives. Daily, we have access to tons and tons of information, but the sources we encounter are not always authentic. In fact, the diffusion of social media is in conjunction with the growth of *disinformation*, malicious activities performed to influence and manipulate people's opinions [Bradshaw and Howard, 2019]. Unlike misinformation, which could be caused by accidental errors, disinformation campaigns are orchestrated to distort information to deceive and mislead the target intentionally [CUP, 2024]. With the proliferation of social media and the ease of spreading online content, disinformation has been disseminated unprecedentedly. This widespread diffusion has reached a level we can consider a threat to society: in the second European External Action Service Report on Foreign Information Manipulation, the high representative of the European Commission, Josep Borrell Fontelles, has stated how Europe is in danger due to the intentional and *coordinated* attempts to manipulate information to create confusion and spread division, fear and hatred [EEAS, 2024]. The report revealed that coordination of online activities with respect to content and time is a key element in these online campaigns, as coordinated behaviour of malicious users enhances the spread of information and its manipulative strength.

In the context of detecting these coordinated inauthentic behaviours, it turned out that unveiling the temporal dynamic is a key element [Pacheco et al., 2021]. These accounts try to amplify the dissemination of their posts via systematic and synergistic actions, which appear anomalous with respect to authentic users' temporal patterns. These temporal patterns can be modelled as temporal point processes, stochastic processes that allow modelling events occurring at random points in time. Temporal point processes not only keep the focus on the arrival times of the events but also manage information regarding the event types and their interactions. Therefore, this process is an optimal tool for modelling social media activities, where users perform actions at a certain timestamp. In the literature, Sharma et al. implement a method leveraging temporal point processes for coordinated detection. In their work, they exploit the recent and famous masked self-attention technique [Vaswani, 2017] to summarise the history and extract the temporal dynamics information, which is subsequently used to divide the groups of accounts depending on their behaviours and interactions. Finally, coordinated users are identified, assuming that their group's features stand out from those of authentic groups. In this thesis, we review this coordination detection method by presenting its structure and analysing its potential and its limitations. Eventually, we investigate two directions in which the model could be improved. On one side, we explore the implementation of different clustering techniques to leverage better the potential evidence in the data of the number of clusters, which must be selected beforehand

in Sharma et al.'s method. In the other direction, we propose a further approach that exploits self-attention weights differently to capture better the influences that authentic and inauthentic accounts have on each other.

Regarding the evaluation of coordinated detection methods, the limited availability of ground truth data represents a major drawback as it makes it complex to perform a systematic and detailed analysis of these methods. To tackle this issue, we test the three model versions in two modalities. First, we consider a dataset collected on Twitter (now X) during the 2023 Finnish parliamentary elections. To these online activities, we add a synthetic coordinated attack in which inauthentic users target specific authentic accounts within a chosen time window. Secondly, we implement a framework that allows us to systematically evaluate the performance of every detection model within different scenarios. In particular, we simulate social media interactions and coordinated attacks leveraging mutually self-exciting Hawkes processes, a type of marked temporal point process that models the occurrences of events triggering each other. This proposed suite allows researchers to assess the quality of their detection in various environments, spanning from groups of inauthentic users specialised in amplifying each other's contents to coordinated accounts that highly interact with the users they are targeting. Finally, via this framework, we evaluate the three versions of the considered detection method in a variety of different scenarios, and we compare the results with Pacheco et al. [2021], Keller et al. [2020], Ng and Carley [2023], Schoch et al. [2022] and Weber and Neumann [2020], state-of-the-art methods founding their identifications on network-based approaches.

Therefore, this thesis will be structured as follows. In Chapter 1, we introduce the concept of disinformation and coordinated behaviour campaigns and review the latest methodologies used in coordination detection. Chapter 2 focuses on marked temporal point processes, defining key mathematical concepts and outlining the most common methods for generating simulations. In Chapter 3, after introducing the idea of masked self-attention and providing the reader with a picture of how it operates, we present Sharma et al.'s model alongside with our extended versions. Chapter 4 then introduces our framework for simulating coordinated behaviour activities. Lastly, Chapter 5 details case studies where the three versions of the model and the aforementioned state-of-the-art methods are tested using a real dataset and several scenarios generated with our framework.

Contents

Chapter 1

Disinformation and coordinated behaviours

The recent rise of social media has completely changed our society and the way individuals approach and look for information. On the positive side, these online platforms enable people from diverse backgrounds to connect globally, breaking down cultural and geographical barriers while facilitating rapid exchange of ideas. However, they also facilitate the dissemination of disinformation, which is defined as fake information intentionally directed to deceive and manipulate public opinion [CUP, 2024]. Disinformation has emerged as one of the most pressing challenges that modern society needs to face [Sharma et al., 2021]. The ease with which this false information can spread rapidly, reaching millions of people in a short time, undermines trust in reliable information sources [Bradshaw and Howard, 2019]. When groups of social media users strategically orchestrate this diffusion of false information with the intent of misleading others, it is referred to as a coordinated disinformation campaign [Facebook, 2018]. As highlighted by the European Commission in its report EEAS [2024], these campaigns pose a significant societal threat by spreading confusion, division, and fear. In fact, by exploiting the attention algorithms of social media and manipulation techniques, malicious groups of users may create divisions in society, shift the focus away from critical topics, and provoke panic or distrust in crucial issues. For example, in 2019, Giglietto et al. reported that 54% of Italians used Meta as their primary source for collecting news, providing fertile ground for online disinformation campaigns, which were subsequently proven to be indeed taking place [Giglietto et al., 2020].

This situation underscores the necessity of an in-depth study of coordinated disinformation campaigns on social media, both to understand their mechanisms and impacts better and to develop effective detection methods. Therefore, in the following sections, we will explore various examples of real-world disinformation campaigns and review the latest methodologies designed to improve their detection in the fight against disinformation.

1.1 Coordinated behaviour campaigns

With growing concerns about the manipulation of online platforms, social media companies have become increasingly involved in the fight against disinformation. Recognising the role their

platforms play in the rapid dissemination of misleading campaigns, these companies began implementing various measures to detect and mitigate disinformation campaigns [Mannocci et al., 2024]. A fundamental contribution to the literature was made by Facebook (now Meta) in 2018 when it introduced the concept of Coordinated Inauthentic Behaviour (CIB). This definition characterises CIB as "those groups of individuals working together to mislead other users about who they are and what they are doing" [Facebook, 2018]. The individuals involved in these campaigns can take on various assets or crafted identities. For instance, they may be social bots, which are automated software applications [Ferrara et al., 2016], sockpuppets, which are accounts constructed to fool users into believing they are authentic individuals [Kumar et al., 2017], or real people coordinating multiple profiles simultaneously to amplify their influence and spread disinformation [Dawson and Innes, 2019].

These entities coordinate themselves with the primary goal of deceiving others. Driven by economic motivations, for instance, inauthentic accounts could orchestrate coordinated campaigns to convince people to click on malicious links, thereby compromising their security or redirecting them to fraudulent sites [Shao et al., 2017]. Another common tactic of coordinated behaviour campaigns is astroturfing [Keller et al., 2020], where these accounts artificially generate support or amplify certain messages to create the illusion of the presence of a grassroots movement or a common feeling around specific topics. By manipulating the visibility of their messages, these coordinated campaigns can mislead the public into believing that a significant number of individuals endorse a particular viewpoint and provoke the herd effect.

Additionally, coordinated campaigns could also be orchestrated to interfere with authentic users' geopolitical visions, distorting public opinion. Between December 2022 and November 2023, the European External Action Service (EEAS) detected more than 750 cases of foreign information manipulation and interference incidents. It was found that the target was global, including countries, official representatives, organisations and groups of individuals [EEAS, 2024]. Considering the cases concerning elections, coordinated campaigns have been organised to control the information flow, set the agenda on specific topics, target citizens' ability to vote, push for abstention and promote specific political ideas or candidates. For instance, Keller et al. [2020] found that this political interference was ongoing, highlighting a case dating back to the 2012 South Korean elections. One of the most documented cases regarded the Internet Research Agency, a St.Petersburg-based company. It was discovered that this agency ran many information-influencing operations targeting the US and Europe [Dawson and Innes, 2019]. These coordinated campaigns were crafted to maximise the reach of targeted users and manipulate them by disseminating harmful messages crafted by the agency's operators. Other cases in which evidence has been found of coordinated behaviour concerned the US [CISA et al., 2024], the UK [CREST, 2024], France and Italy [EEAS, 2024]. Further notorious cases of inauthentic coordination include the coronavirus health crisis, the Hong Kong protest movement, and the Capital Riots: the reader is referred to Iannucci [2023] for a description of the observed strategies inauthentic accounts followed in such episodes.

1.2 Coordination detection state-of-the-art methods

In the previous paragraph, we observed that Coordinated Inauthentic Behaviour is not necessarily linked to the accuracy or truthfulness of the content being spread. Instead, it relates to the methodologies and the objectives of the users involved. Consequently, coordinated users could be identified based on their coordination level and the interaction they have with the online community [Mannocci et al., 2024]. Specifically, their highly synchronised or systematically arranged activities stand out from the authentic interactions for their anomalous features. These distinctive features could be leveraged to detect coordination automatically. Following the 2016 US election and the Internet Research Agency scandal [Dawson and Innes, 2019], industrial stakeholders and researchers devoted increasing effort to the field, resulting in the development of numerous detection methods. The recent survey Mannocci et al. [2024] collects and categorises the literature on this topic, offering a comprehensive review of existing techniques. This survey, alongside with Iannucci [2023], has been used to structure the current section, which provides a comprehensive overview of the state-of-the-art detection methods.

The main objective of coordination detection methods is to identify groups of anomalously coordinated users based on the activities they performed in their history. Mathematically, we could define $U := \{u_1, u_2, \dots\}$ the user set and H the entire history. Depending on the detection methods, the information contained in the activities' history could be different, but in general, the author, the type of action and the timestamp are always defined. For instance, some methods could only require those features, like Sharma et al. [2021], and others require and exploit the content of the activities, such as posted text, as in Chomel et al. [2022], or shared images, as in Yu [2022]. Based on the gathered information, detection methods conduct their calculations and generate an output that can be structured in three possible formats. In the simplest scenario, the results consist of labels being assigned to users. An account is labelled as "coordinated" if the detection method finds evidence of coordinated behaviour. Otherwise, the account is associated with the "non-coordination" label. These detection methods are typically linked with classifier algorithms, as in Mariconti et al. [2019] and Ruchansky et al. [2017]. The second possible output format is structured as a set of clusters, where users are grouped into N distinct clusters. These methods are generally linked to clustering techniques, as in Sharma et al. [2021] and Assenmacher et al. [2020]. Unlike the previous label-based approach, which immediately clarifies the set of users labelled as coordinated, this method requires additional analysis to identify the coordinated clusters. However, it provides a richer understanding of user behaviour by offering more insight into the clustering of users, revealing their organisation and potential organised subgroups within the entire user set. Finally, the third possible output format involves identifying user communities. These methods are typically associated with networks-based approaches, where nodes are user accounts, while edges represent their interactions. The output of these approaches relies on sub-networks representation of highly coordinated communities. These strategies offer the most comprehensive insight into account behaviour, as they not only identify coordinated users but also reveal the structure of their interactions and the overall network dynamic. Examples of this implementation are Pacheco et al. [2021], Keller et al. [2020], Ng and Carley [2023], Schoch et al. [2022] and Weber and Neumann [2020].

Every detection method takes the user set and the history as input, and it elaborates the information to return the output in one of the three possible formats. Therefore, we could identify detection methods algorithms as functions

$$U, H \to f(U, H) =: Y,$$

where Y could be a list of labels, clusters or communities. According to Mannocci et al. [2024], we could classify these coordination methods into two distinct categories: network-based or machine-learning-based. The former methods leverage the activity patterns to build a network of users, where weighted edges indicate the level of coordination between each pair of accounts. These methods use community detection to group the users. In contrast, the latter category bases its detection on machine learning: it collects information from the input data, reformulates it to retrieve the features associated with each group, and finally exploits it to retrieve the labels or clusters reporting evidence of coordination. In this process, machine learning optimisation procedures are necessary to learn the model parameters based on the training dataset. These machine-learning techniques can, in turn, be divided into two distinct classes: supervised and unsupervised techniques.

1.2.1 Detection methods based on coordination networks

The most common approach for detecting coordinated behaviour relies on network analysis. In this framework, we have already seen that nodes represent users, and weighted edges capture their interactions. The key idea is that these weights are inferred based on specific features of user activities that each detection method emphasises. Afterwards, the obtained edges are filtered to remove those resulting from random interactions, and community detection algorithms are applied to identify coordinated groups of users within the refined network. In this paragraph, we will provide a detailed explanation of each step, outlining the various choices available in the literature for constructing these models.

Starting from the user set U, the first phase involves the construction of the network. The initial step is determining whether to maintain all users or to filter down to a smaller subset $U' \subset U$. This decision can depend on various factors, such as the specific goal of the analysis, the volume of the data, or the computational resources available. If a comprehensive analysis is desired and the number of activities of each account is enough, every user could be retained. Otherwise, if some accounts are rarely active on social media or if the objective is to focus on a particular group of users, a smaller set could be identified depending on the activity level and the interaction frequency. For example, this choice was taken by the authors of Pacheco et al. [2021] to speed up the computational time and expenses and return much more accurate results.

Subsequently, it is necessary to select the metric for inferring the edge weights from the activities. The main point is to find a proper and efficient way to encode the coordination between each pair of nodes. To do so, scholars typically leverage co-action cascades, lists of users' activities performing the same action, such as posting, mentioning or replying, on the same target. Since the systematic participation of users in the same cascades may be evidence of coordination,

these networks are often referred to as *latent coordination networks* [Weber and Neumann, 2021, see]. The metric with which the edge weights are inferred is commonly obtained via a similarity function. The most straightforward case is counting the number of cascades in which the user pairs appear together. This measure is called *cardinality* and is the most popular metric in this context. For instance, this method is implemented in Keller et al. [2020], Ng and Carley [2023] and Schoch et al. [2022]. Another popular similarity measure is based on the *cosine similarity* of vectors representing the activities of the considered users. These vectors, for example, could consist of frequency vectors or binary representations [Neha et al., 2024, Tardelli et al., 2024]. Let us consider any pair of social media users $u_i, u_j \in U$ represented respectively by the vectors E_{u_i} and E_{u_j}. Supposing we are evaluating the weight of their edge (u_i, u_j), the cosine similarity is computed as

$$\text{Cosine similarity}(u_i, u_j) = \frac{E_{u_i} \cdot E_{u_j}}{\|E_{u_i}\| \|E_{u_j}\|}.$$

Another similarity measure that could be employed is the Jaccard index, which quantifies the overlap between two sets. These sets could represent the users who have interacted with u_i and u_j, respectively, or they could be a collection of posts they have engaged with [Pacheco et al., 2021, see]. For instance, consider the same pair of users u_i and u_j, associated with two sets A_{u_i} and A_{u_j}. The Jaccard similarity measure can then be calculated as

$$\text{Jaccard similarity}(u_i, u_j) = \frac{|A_{u_i} \cap A_{u_j}|}{|A_{u_i} \cup A_{u_j}|}.$$

This metric is used, for example, in Pacheco et al. [2021]. Independently from how edge weights are computed, the detection methods may construct their networks considering only a specific type of co-action, or they could consider more of them simultaneously. In these cases, some scholars opted for building a multiplex network, where each layer reports the edges of a singular co-action type, such as Emeric and Victor [2023]. In contrast, others, as the authors of Ng and Carley [2023], opted for flattening the network afterwards.

Scholars implementing approaches based on coordination networks must manage their methods' computational costs carefully to prevent overflows and ensure meaningful encoding of information. To do so, several authors, as Schoch et al., Linhares et al., opted to filter the edges of the constructed network depending on some thresholds. This could be done by removing every weight below a chosen constant, like in Schoch et al. [2022], or it could be done statistically, keeping only the weights within the first quantiles, as in Linhares et al. [2022]. Another widely used filtering technique relies on the activities' temporal dynamics of user interactions. When two users perform the same actions on the same target, their coordination is more indicative when the time difference between their actions is small. In fact, synchronised activities represent strong evidence of coordination [Mannocci et al., 2024]. This concept gave rise to the idea of segmenting activities cascades into different time windows, enhancing the detection of closely timed interactions. Practically, let us consider a specific time window of size $Z = t_{\text{end}} - t_{\text{start}}$. We scan each activity trace and retain only the actions occurring in this time frame. Subsequently, we construct edges based solely on these truncated traces. This approach saves computational resources by limiting the amount of data processed at each step and ensures that the activity

being analysed is relatively close in time. Subsequently, the networks obtained with each temporal window are typically merged together. These could be done in different ways, like by simply adding up the edge weights associated with the same user pair, as in Weber and Neumann [2020], or constructing a multiplex network where each layer is associated with a different time window, as in Tardelli et al. [2024]. In implementing this approach, a fundamental choice is the selection of the time windows. The most straightforward method is dividing the timeline into adjacent intervals of the same size [Weber and Neumann, 2020, see]. In this way, actions performed close in time but across a boundary of two adjacent time windows are not considered as evidence of coordination. To avoid this, overlapping time windows may be chosen. If intervals are constructed by shifting the boundaries of a constant step $\delta < Z$, each time window overlaps with its adjacent windows by $O := Z - \delta$. This procedure is referred to as *Evenly Distributed Overlapping*, and it is implemented, for instance, in Tardelli et al. [2024]. A further approach involves the construction of new time windows as soon as a relevant event occurs. This last procedure is called *Action-Driven Overlapping*, and it was implemented, for example, in Pacheco et al. [2020]. Nevertheless, regardless of the methodologies implemented for the construction of time windows, these methods are very sensible in terms of the choice of the parameters. For instance, selecting the correct window size is delicate as scholars risk missing important co-actions or giving too much importance to interactions that happen by chance.

Finally, the last step leverages community detection techniques to identify the coordinated communities. The most employed methods are Louvain, as in Neha et al. [2024], and Leiden, as in Traag et al. [2019]. If the constructed method is a multiplex, it could be flattened to reduce the complexity, or community detection methods suitable for multi-layer networks, such as the multiplex version of the Louvain algorithm [Suzuki and Tsugawa, 2023], could be used.

1.2.2 Detection methods based on supervised machine learning

If a machine learning detection method exploits background knowledge of user coordination in the training data, it falls into the supervised category. These methods require datasets consisting of users already labelled as coordinated or non-coordinated. This represents a major limitation since such datasets are very scarce. Nevertheless, scholars have created several coordination detection methods based on supervised techniques. Let us introduce some of them.

An example of a supervised detection method is Mariconti et al. [2019]. In this method, Mariconti et al. apply an ensemble classifier to YouTube videos to predict whether they will be victims of coordinated behaviour attacks. To create a ground truth dataset, the authors considered a list of YouTube videos whose links were posted on 4chan, a controversial website where fringe communities often organise raids against YouTube videos promoting ideas the community does not share. Considering the number of hate comments per second and the synchronisation between them, the list has been divided into subjects of coordinated campaigns or not. Using metadata, thumbnails, and audio processing, the features of each video were extracted and fed into a proactive detection tool, returning the likelihood of being targeted by coordinated campaigns.

Another example of coordinated detection based on supervised machine learning techniques is

the *Capture Score Integrate* method proposed in Ruchansky et al. [2017]. In this article, Ruchansky et al. exploited the text of articles, the response that is provoked in users reading it, and the URL source of an article. Their main objective was to classify a list of articles as fake news or not and to identify groups of suspicious users, the objective we are interested in. To do so, the authors constructed a model consisting of three modules. In the first two components, a recurrent neural network extracts the temporal representation of user activities and a fully connected layer is applied to users' features to produce a score. The concatenation of these outputs is then used in the third module for the classification of articles and malicious group identification.

A further article that proposed a supervised approach is Luceri et al. [2020]. Luceri et al. proposed an Inverse Reinforcement Learning method relying on users' activity traces in conjunction with one supervised classification algorithm chosen from state-of-the-art approaches. In particular, they experimented the employment of AdaBoost, Multi-Layer Perceptron, Decision Tree, Naive Bayes and Random Forest. Subsequently, the model has been tested over the malicious account tied to Russia's Internet Research Agency, the coordinated campaign factory we have mentioned in Section 1.1.

1.2.3 Detection methods based on unsupervised machine learning

Up to now, we have discussed machine learning approaches based on supervised learning. However, we have already highlighted the challenge posed by the requirement of label datasets, which are rarely available. When there is no background knowledge about users' involvement in coordination campaigns, the detection method must be based on unsupervised learning techniques. These approaches typically involve extracting users' features from their online activities or metadata and exploiting them to cluster accounts into distinct groups. Below, we will introduce some of the most common models employed in this framework.

Many recently published articles have focused on the activities content of users clustering posts or replies via the stream clustering TextClust [Carnein et al., 2017, see]. This algorithm monitors the text in the activities' stream by grouping them according to their topic and associating each cluster with a weight indicating the number of new elements recently added to itself. These clusters and their weights could be directly exploited to identify activities associated with coordinated campaigns. For instance, Assenmacher et al. analysed micro-cluster evolution. Basing their method on the idea that coordinated campaigns were associated with recently added topics with abnormal lifespans, the authors selected micro-clusters depending on the unusual merging of recent texts into these categories.

A different way of identifying clusters of accounts leverages the users' activity patterns. For example, the Inverse Reinforcement Learning presented above could be associated with an unsupervised clustering technique such as Gaussian Mixture Models or KMEANS. Then, the clusters representing coordinated accounts are individuated according to features resulting anomalous with respect to the other groups, such as the number of accounts [Luceri et al., 2020, see].

Another approach based the coordination detection on learning latent user's representation vec-

tors modelling their activities on temporal point processes. A state-of-the-art method in this domain is the Attentive Mixture Density Network with Hidden Account Group Estimation, as presented in Sharma et al. [2021]. This method leverages temporal data to identify coordinated behaviour patterns by modelling user actions and their group affiliations. Due to its strong connection with temporal point processes and its demonstrated effectiveness in the detection of coordinated inauthentic behaviour, this work will be at the centre of this thesis and is presented in detail in Chapter 3. In fact, we believe that capturing and modelling the temporal dynamics is crucial for improving coordination detection results in real-world scenarios.

Chapter 2

Temporal Point Processes

Temporal point processes are a crucial mathematical concept from the field of probability theory and stochastic processes, with important applications in areas like finance, seismology and queuing systems. In this chapter, we will examine the principal notions, introducing definitions and propositions, as well as examples related to the scope of this thesis. The key reference texts that we have been used are Daley and Vere-Jones [2006], Hawkes [1971] and Laub et al. [2015], to which the reader is referred for further details.

2.1 Stochastic processes

In real-world phenomena, it is widespread to encounter objects that are evolving in a way that is not deterministic. To study their evolution over time or some other dimension, it is necessary to introduce a mathematical concept that is fundamental in the field of probability.

Definition 2.1 (Stochastic process)**.** *Let us consider* $(\Omega, \mathcal{F}, P)$ *a probability space and* (E, Σ) *a measurable space and* T *a set of indices, then a stochastic process is a family of random variables* $\{X_t : \Omega \to E | t \in T\}$.

Stochastic processes can be classified into different groups depending on their nature. We could be dealing with discrete or continuous state space, like $E = \mathbb{R}^n$, and with a discrete or continuous set of indices, as $T = \mathbb{Z}, \mathbb{N}, \mathbb{R}, \ldots$. The latter dimension is commonly referred to as *time*.

Let us introduce some examples of stochastic processes.

Example 2.1 (Rademacher's process)**.** *The collection of independent Rademacher random variables* $\{X_t | t \in \mathbb{N}\}$*, i.e. discrete random variables respecting* $\mathbb{P}(X_t = -1) = \mathbb{P}(X_t = +1) = \frac{1}{2}$*, is a discrete stochastic process.*

Example 2.2 (Random Walk)**.** *Setting* $S_0 := 0$ *and* $S_t := \sum_{k=1}^{t} X_k$ *where* X_k *are independent Rademacher random variables, the family* $\{S_t | t \in \mathbb{N}\}$ *is a discrete stochastic process.*

Example 2.3 (Gaussian White Noise)**.** *A Gaussian White Noise process is a continuous stochastic process defined as the collection of independent and identically distributed random variables such that* $X_t \sim N(0, \sigma^2)$ *for* $t \in T := \mathbb{Z}$ *and* $\sigma^2 \in \mathbb{R}_+$.

If an element of the probability space is fixed, $\omega \in \Omega$, then for every random variable a single possible output is obtained. Thus, as the index varies, the collection $\{X_t(w)\}_{t\in T}$ contains a *realisation of the stochastic process* itself also called *trajectory*.

To describe the probability of different possible realisations, another essential concept must be introduced.

Definition 2.2 (Finite-dimensional distribution)**.** *Considering a stochastic process, its finite-dimensional distribution functions are defined as*

$$F_{(t_1,\dots,t_n)}(x_1,\dots,x_n) := \mathbb{P}(X_{t_1} \leq x_1,\dots,X_{t_n} \leq x_n)$$

for $(x_1,\dots,x_n)^{\mathcal{T}} \in \mathbb{R}^n$*, for* $n = 1,2,\dots$ *and for* $(t_1,\dots,t_n) \in T^n$ *such that* $t_1 \leq t_2 \leq \cdots \leq t_n$*.*

One of the most important results about finite-dimensional distributions asserts that they completely characterise a stochastic process as reported in Kolmogorov's theorem [Brockwell and Davis, 1991, see].

2.2 Simple temporal point processes

To describe events that occur at random points in time, the following type of stochastic processes could be used.

Definition 2.3 (Temporal Point Process)**.** *A temporal point process is* $\{T_i : \Omega \to \mathbb{R} \mid i \in \mathbb{N}\}$ *a stochastic process whose realisation consists of a sequence of discrete events in continuous time as* $\{t_1, t_2, ..., t_N\}$ *for* $t_i \in \mathbb{R}$*.*

Thus, the focus of the process is on the time at which each event happens and not on the specific event itself. Moreover, the term *point process* comes from the fact that each occurrence is instant in time and, in consequence, it can be represented as a point on a timeline. In this thesis, time will be considered only in its strictly related natural *evolutionary character*, meaning that the occurrence of an event may depend on what happened in the past but not on what will happen in the future. The property is also referred to as *causality*. In this way, it is fundamental to be able to capture the past in the concept of history $H_t := \{t_i | t_i < t\}$, which saves every event taking place strictly before the given time.

Going into the details, in this thesis we will be working with *simple* temporal point processes: events' occurrences cannot be simultaneous i.e. $t_i < t_j \quad \forall i < j$. Moreover, we will assume that events happen in finite numbers when considering a finite time interval. The latter condition is often referred to as *no explosion* assumption and it ensures that the process can be properly modelled and analysed. Thanks to these assumptions, it is possible to characterise the process in two further ways: one could only consider the inter-arrival time between the occurrences of the events or it could count the number of cases that happened before a given time. In the former case, we would deal with $\{\tau_i | i = 1, ...\}$ where $\tau_i := t_i - t_{i-1}$ and t_0 is the starting time. This formulation usually simplifies the calculation and allows for the normalisation of data focusing primarily on the duration of the interval of no-events. Regarding the latter case, a new definition must be introduced.

Definition 2.4 (Counting process)**.** *A stochastic process* $\{N_t | t > 0\}$ *that takes non-negative integer values with* $N_s \leq N_t \ \forall s, t$ such that $s < t$ *it is called a counting process.*

Additionally, if the focus is on a certain time interval like $[s, t)$, it is possible to use the notation $N_{[s,t)} := N_t - N_s$, which will become useful while dealing with the interpretation of next mathematical concept which is central in this study. Nevertheless, starting from a given temporal point process, it is sufficient to define $N_t := \sum_i \mathbb{1}_{t_i<t}$ to obtain a counting process that focuses on the number of occurred events in a specific time. Likewise, it is possible to retrieve the temporal point process only looking at the jump of the counting process itself. Therefore for every temporal point process, there is one and only one counting process associated with it.

For a given temporal point process T_i, the causality property implies that we could describe the process simply by explaining how t_{n+1}, the occurrence of the next event, behaves with respect to what happened before. More precisely, it is sufficient to define the conditional density function $f(t_{n+1}|H_{t_n})$ for every event t_{n+1}. In this way, the distribution of all the events can be easily obtained by exploiting the *conditional probability rules*:

$$f(t_1, t_2, ...) = \prod_n f\left(t_{n+1}|t_1, ..., t_n\right) = \prod_n f(t_{n+1}|H_{t_n}).$$

This formulation comes particularly handful when the primary interest is directly describing the time of the next event, the inter-arrival times or, in general, when one deals with the calculation of probabilities or expectations on time intervals. The function $f(t_{n+1}|H_{t_n})$ needs to be modelled for any specific application, but the fact that it has to be a proper conditional probability (it must be non negative and it must integrate to 1) often makes this a difficult task. For facilitating the modelling task, another concept should be introduced.

Definition 2.5 (Conditional intensity function)**.** *Taking into consideration a simple temporal point process with* $f(t|H_t)$ *its* conditional density function *and* $F(t|H_t)$ *its* conditional distribution function*, the conditional intensity function is defined as*

$$\lambda(t|H_t) := \frac{f(t|H_t)}{1 - F(t|H_t)}. \tag{2.1}$$

Moreover, the integral of the conditional intensity function is called the compensator*:*

$$\Lambda(t \mid H_t) = \int_0^t \lambda(u \mid H_u) du.$$

Notice that the conditional intensity function is positive by definition but its integral is not constrained to be equal to 1.

To interpret this definition, let us take a look at the following proposition.

Proposition 2.1. Let $\lambda(t|H_t)$ be the conditional intensity function of a simple temporal point process, N_t its associated counting process, t_n the last event in H_t, T_{n+1} the next event and dt an infinitesimal time interval. The following holds:

$$\lambda(t|H_t)dt = \mathbb{P}\left(T_{n+1} \in [t, t + dt) \,|H_t\right) = \mathbb{E}\left[N_{[t,t+dt)}|H_t\right] = \mathbb{E}\left[dN_t|H_t\right]. \tag{2.2}$$

Proof. Starting from the definition of the conditional intensity function and exploiting the definition of the conditional density function, we get

$$\begin{aligned}\lambda(t|H_{t_n})dt &= \frac{f(t|H_{t_n})dt}{1-F(t|H_{t_n})} = \frac{\mathbb{P}\left(T_{n+1}\in[t,t+dt)\,|H_{t_n}\right)}{1-\mathbb{P}\left(T_{n+1}\in[t_n,t)\,|H_{t_n}\right)} \\ &= \frac{\mathbb{P}\left(T_{n+1}\in[t,t+dt)\,,T_{n+1}\notin[t_n,t)\,|H_{t_n}\right)}{\mathbb{P}\left(T_{n+1}\notin[t_n,t)\,|H_{t_n}\right)} \\ &= \mathbb{P}\left(T_{n+1}\in[t,t+dt)\,|T_{n+1}\notin[t_n,t)\,,H_{t_n}\right) = \mathbb{P}\left(T_{n+1}\in[t,t+dt)\,|H_t\right).\end{aligned}$$

Additionally, if we condition on H_t the entire history until time t in place of H_{t_n} the result does not change. Indeed, we would obtain

$$\lambda(t|H_t)dt = \frac{f(t|H_t)dt}{1-F(t|H_t)} = \frac{\mathbb{P}\left(T_{n+1}\in[t,t+dt)\,|H_t\right)}{1-\mathbb{P}\left(T_{n+1}\in[t_n,t)\,|H_t\right)} = \mathbb{P}\left(T_{n+1}\in[t,t+dt)\,|H_t\right),$$

where the last equality comes from the fact that $\mathbb{P}\left(T_{n+1}\in[t_n,t)\,|H_t\right)=0$ since we already know that no further event happened in H_t. Moreover, the "simple" feature of the *simple temporal point process* guarantees that there is at most one occurrence in the infinitesimal interval, thus

$$\begin{aligned}\mathbb{E}\left[N_{[t,t+dt)}|H_t\right] &= \mathbb{E}\left[N_{t+dt}-N_t|H_t\right] = \mathbb{E}\left[dN_t|H_t\right] \\ &= \mathbb{P}\left(N_{t+dt}-N_t=1|H_t\right)\cdot 1 + \mathbb{P}\left(N_{t+dt}-N_t=0|H_t\right)\cdot 0 \\ &= \mathbb{P}\left(T_{n+1}\in[t,t+dt]\,|H_t\right)+0.\end{aligned}$$

□

In other words, the conditional intensity function represents the expected increment in the number of occurrences in an instant of time given that all the previous arrivals are known. Indeed, describing a temporal point process with this function is particularly useful when one wants to directly control the instantaneous rate of events conditioned on the history. Moreover, it is mathematically more convenient since it requires to be non-negative only and it is more intuitive when modelling a process. For example, the intensity function permits easy incorporation of various independent temporal point processes.

Proposition 2.2. Let us take into consideration two *independent* temporal point processes with respectively $\lambda_a(t|H_t^a)$ and $\lambda_b(t|H_t^b)$. The conditional intensity function of the joint process [1] is:

$$\lambda(t|H_t^a\cup H_t^b) = \lambda_a(t|H_t^a)+\lambda_b(t|H_t^b).$$

Proof. Considering an infinitesimal interval dt and using Proposition 2.1,

$$\begin{aligned}\lambda(t|H_t^a\cup H_t^b)dt &= \mathbb{E}\left[N_{[t,t+dt)}|H_t^a\cup H_t^b\right] \\ &= \mathbb{E}\left[N^a_{[t,t+dt)}+N^b_{[t,t+dt)}|H_t^a\cup H_t^b\right] \\ &\overset{\perp}{=} \mathbb{E}\left[N^a_{[t,t+dt)}|H_t^a\right]+\mathbb{E}\left[N^b_{[t,t+dt)}|H_t^b\right]. \\ &= \lambda_a(t|H_t^a)+\lambda_b(t|H_t^b).\end{aligned}$$

□

[1] Recall that the joint temporal process describes the occurrences of event of type a or type b.

Once the function $\lambda(t|H_t)$ is specified, it is immediate to retrieve the conditional density function and to evaluate the distribution of all the events, as shown below.

Proposition 2.3. A conditional intensity function $\lambda(t|H_t)$, which must be non-negative and integrable in any interval, uniquely defines a temporal point process if for any realisation and any $t > t_n$

$$\lim_{t\to+\infty} \int_{t_n}^{t} \lambda(u|H_u)du = +\infty. \tag{2.3}$$

Moreover, knowing that t_n is the last event in the history H_t, the conditional density function can be retrieved via

$$f(t|H_t) = \lambda(t|H_t) \cdot e^{-\int_{t_n}^{t} \lambda(u|H_u)du}.$$

Proof. Assuming that the densities of the inter-arrival times are well-defined - thus that the temporal point process has a well-defined distribution - and exploiting the definitions of $\lambda(t|H_t)$, $f(t|H_t)$ and $F(t|H_t)$, it follows that

$$\lambda(t|H_t) = \frac{f(t|H_t)}{1 - F(t|H_t)} = \frac{\frac{dF}{dt}(t|H_t)}{1 - F(t|H_t)} = -\frac{d\log\left(1 - F(t|H_t)\right)}{dt}.$$

Integrating on $[t_n, t)$,

$$\log\left(1 - F(t|H_t)\right) = \log\left(1 - F(t|H_t)\right) - \log(1) = -\int_{t_n}^{t} \lambda(u|H_u)du.$$

Therefore we would get a unique possible solution:

$$F(t|H_t) = 1 - e^{-\int_{t_n}^{t} \lambda(t|H_t)dt},$$
$$f(t|H_t) = \lambda(t|H_t)e^{-\int_{t_n}^{t} \lambda(t|H_t)dt}.$$

Finally, we only need to check if they are admissible functions. Focusing on the former, from the hypothesis and the formulation found above, it immediately follows that $0 \leq F(t|H_t) \leq 1$, it is a non-decreasing function and that

$$\lim_{t\to+\infty} F(t|H_t) = \left[1 - e^{-\infty}\right] = 1.$$

Consequently, the cumulative distribution function is unique and well-defined, as well as the density function. If we drop the hypothesis 2.3, everything in the proof still holds except that

$$p := \lim_{t\to+\infty} F(t|H_t) = \lim_{t\to+\infty} 1 - e^{-\int_{t_n}^{t} \lambda(t|H_t)dt} < 1.$$

Thus, in this case, we will not be dealing with a proper cumulative function and there is a positive probability that the process will terminate with no more events. □

Thanks to this proposition, given a realisation $\{t_1, \ldots, t_N\}$ of the process in the interval

$[0, T]$, the joint density distribution is

$$f(t_1, t_2, \ldots, t_n) = \prod_n f(t_{n+1}|H_{t_n}) \cdot \Big(1 - F(T|H_{t_n})\Big) \tag{2.4}$$

$$= \prod_n \lambda(t_{n+1}|H_{t_n}) \cdot e^{-\int_{t_n}^{t_{n+1}} \lambda(u|H_u)du} \cdot e^{-\int_{t_N}^{T} \lambda(u|H_u)du}, \tag{2.5}$$

$$\log f(t_1, t_2, ...) = \sum_{n=1}^{N} \log \lambda(t_{n+1}|H_{t_n}) - \sum_{n=1}^{N} \int_{t_n}^{t_{n+1}} \lambda(u|H_u)du - \int_{t_N}^{T} \lambda(u|H_u)du \tag{2.6}$$

$$= \sum_{n=1}^{N} \log \lambda(t_{n+1}|H_{t_n}) - \int_0^T \lambda(u|H_u)du, \tag{2.7}$$

where the term $1 - F(T|H_{t_n})$ appears to encapsulate the fact that no events happened in (t_N, T).

In constructing a model where the conditional intensity function is parameterised as $\lambda_\theta(u|H_u)$, the maximum likelihood estimate can be evaluated by exploiting the integral equation above. Unfortunately, the cases in which the solution can be found analytically are rare and very often numerical approximations are required.

Summing up, all the basic concepts to deal with temporal point processes have been introduced, thus let us now move to some examples to see how they can be applied in real-world situations and how to do inference on them.

2.2.1 Homogeneous Poisson Process

A process where the probability of a new event depends only on the last arrival time rather than on the entire history is called a *Markov process*. A Markov process whose distribution follows an exponential law is called a *Homogeneous Poisson Process*.

More precisely, we are considering a temporal point process $\{T_i : \Omega \to \mathbb{R}^+ \mid i \in \mathbb{N}\}$ whose conditional density function is equal to

$$f(t|H_t) = f(t|t_n) = \lambda^* e^{-\lambda^*(t-t_n)},$$

where t_n is the last event in the history and λ^* is a fixed parameter. Thus, the inter-event times $\{\tau_i\}_i$ are independent and identically distributed random variables with expected values equal to

$$\mathbb{E}\left[\tau_{i+1}|H_{t_i}\right] = \mathbb{E}\left[T_{i+1} - t_i|H_{t_i}\right] = \int_{\mathbb{R}_+} u \cdot \lambda^* e^{-\lambda^*(u)du} = \frac{1}{\lambda^*}.$$

Moreover, it is possible to retrieve its intensity function by exploiting equation 2.1:

$$\lambda(t|H_t) = \frac{\lambda^* e^{-\lambda^*(t-t_n)}}{1 - \left[1 - e^{-\lambda^*(t-t_n)}\right]}$$

$$= \lambda^* \frac{e^{-\lambda^*(t-t_n)}}{e^{-\lambda^*(t-t_n)}} = \lambda^*.$$

Furthermore, it can be proved that the associated counting process of an Homogeneous Poisson process follows a Poisson distribution

$$N_t \sim Poisson\big(\lambda^* t\big),$$

where the expected number of events before t is simply

$$\mathbb{E}\big[N_t\big] = \lambda^* t.$$

Homogeneous Poisson processes represent a special case in which it is possible to find the maximum likelihood estimator analytically. From equation 2.7, it holds that

$$\begin{aligned} l\big(\lambda^*; t_1, \dots, t_N\big) := \log f(t_1, t_2, ... \mid \lambda^*) &= \sum_{n=1}^{N} \log \lambda^* - \int_0^T \lambda^* du \\ &= N \cdot \log \lambda^* - T\lambda^*. \end{aligned}$$

Consequently, given that

$$\begin{aligned} \frac{dl}{d\lambda^*}\Big(\lambda^*; t_1, \dots, t_N\Big) &= \frac{N}{\lambda^*} - T, \\ \frac{d^2 l}{d{\lambda^*}^2}\Big(\lambda^*; t_1, \dots, t_N\Big) &= -\frac{N}{{\lambda^*}^2} < 0, \end{aligned}$$

then the maximum can be easily found imposing that the first-order derivative is equal to 0:

$$\widehat{\lambda^*} = \frac{N}{T},$$

which is what we intuitively expect.

Another important characteristic that must be checked is if the no explosion condition is satisfied. As the inter-event times are independent and identically distributed, $\tau_i \sim \exp(\lambda^*)$, it follows that the arrival time of the n^{th} event $T_n = \tau_1 + \cdots + \tau_n$ is gamma distributed. The mean of this distribution turns out to be equal to $\mathbb{E}\left[T_n\right] = \frac{n}{\lambda^*}$. Considering any finite interval $\big[s, t\big)$ with $s < t$ and knowing that both N_t and N_s are Poisson distributed, it is possible to evaluate the number of expected arrivals in that window of time:

$$\mathbb{E}\big[N_{[s,t)}\big] = \mathbb{E}\big[N_t - N_s\big] = (t - s)\,\lambda^* < +\infty.$$

Therefore, it implies that the number of arrivals in any finite interval is almost surely finite:

$$\mathbb{P}\left(N_{[s,t)} = +\infty\right) = 0,$$

thus the occurrence of an infinite number of events in a finite time is prevented, so explosions cannot occur.

In conclusion, for the Homogeneous Poisson process, the formulation of the conditional intensity function is the simplest possible with the instantaneous rate being constant in time and it can model various real-life phenomena, from the arrival of cars in a fuel station to the number of new posts with different topics published in a social network.

2.2.2 Hawkes Process

Another well-known process is the Hawkes Process. This temporal point process is defined via its conditional intensity function as follows:

$$\lambda(t|H_t) = \mu + \alpha \sum_{t_i < t} e^{-\beta(t - t_i)},$$

where $\mu \geq 0$, $\alpha > 0$ and $\beta > 0$ are constant in time. The arrival time of a new event depends on all the history in the past: starting from a shared baseline intensity, as soon as an event appears, it enhances the instantaneous rate of α, but as time passes, its influence decreases exponentially. Regarding this, several variations of the Hawkes process exist: they modify how past events keep triggering new occurrences in time. Thus, a more general formulation can be introduced:

$$\lambda(t|H_t) = \mu + \sum_{t_i<t} \phi\left(t - t_i; \beta\right) = \mu + \sum_{t_i \in H_t} \phi\left(t - t_i; \beta\right), \tag{2.8}$$

where $\phi(u) : \mathbb{R} \to \mathbb{R}_+$ may depend on some parameter $\beta \in \mathbb{R}$. Moreover, equation 2.7 leads to a complicated equation for the maximum likelihood estimator which is not analytically solvable:

$$l\big(\mu, \alpha, \theta; t_1, \ldots, t_N\big) = \sum_{n=1}^{N} \log\Big(\mu + \sum_{t_i<t} \phi\left(t - t_i; \theta\right)\Big) - \mu T - \int_0^T \sum_{t_i<u} \phi\left(u - t_i; \theta\right) du.$$

Figure 2.1 represents the conditional intensity function associated to a realisation of the Hawkes process with $\alpha = 0.8$, $\beta = 0.7$ and $\mu = 0.5$.

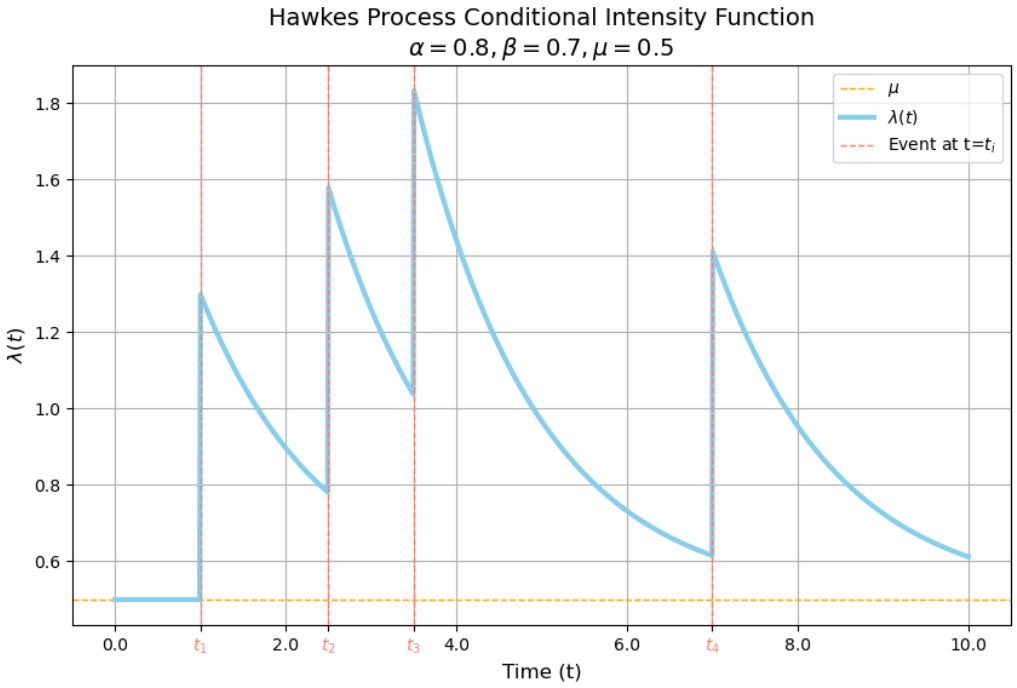

Figure 2.1: Hawkes process conditional intensity function

Regarding the possibility of the process explosion, some hypotheses on the values of the parameters must be added. To understand this problem, we should look at the process as an immigration-birth problem: each new event could be generated either by immigration, arriving with the constant rate μ, or by birth from previous verified events, which is directly related to the self-exciting term of the instantaneous rate. The latter case is the one that could cause explosions, so we will now focus on this.

Considering how many new arrivals an event generates on average, the rate at which a new case $t_i \in \mathbb{R}_+$ creates new offspring is $\phi(t - t_i; \beta)$ for time $t > t_i$. Borrowing from branching theory terminology, let the *branching ratio*

$$n := \int_0^{+\infty} \phi(u; \alpha, \beta) du$$

represent the expected number of offspring generated by the event t_i and Z_1 be the random variable representing the actual number of offspring in the first generation. Then the counting process of arrivals directly produced by the considered event follows a Poisson distribution: $Z_1 \sim Poisson(n)$. Additionally, the number of children in the i^{th} generation can be described by introducing Z_i random variables with $\mathbb{E}[Z_i] = n^i$, where the $i = 0$ case is the original event itself, thus $Z_0 = 1$. To retrieve the expected number of descendants Z_{TOT} for one event (including itself), we can proceed as follows:

$$\mathbb{E}[Z_{\text{TOT}}] = \mathbb{E}\left[\sum_{i=0}^{\infty} Z_i\right] = \sum_{i=0}^{\infty} n^i = \begin{cases} \frac{1}{1-n}, & n < 1 \\ +\infty, & n \geq 1 \end{cases},$$

where swapping the infinite sum and the expectation is allowed since we are dealing with a countable series. Hence, in the case of $n \geq 1$, the average number of offspring generated by each arrival is infinite. On the other hand, if $n < 1$ the number of descendants of one process is controlled and, more importantly, it is finite. How is this feature related to the possibility of explosions? Let's assume that we are dealing with $n < 1$ and let's distinguish the events as being generated due to the immigration (baseline) or due to another arrival triggering them. We will describe the counting process describing the former as M_t, a Homogeneous Poisson process with parameter μ, and the latter as G_t, a simple temporal point process. Considering a finite interval $[0, t)$ and conditioning on M_t, the average number of arrivals happening because of progeny can be controlled by

$$\mathbb{E}[G_t \mid M_t] \leq \mathbb{E}\left[M_t\big(Z_{\text{TOT}} - Z_0\big) \mid M_t\right] \leq M_t \frac{n}{1-n}.$$

Indeed, this expectation of G_t cannot be greater than the total number of immigrants M_t times the size of any event's total progeny.

Exploiting the last result and the properties of the expectation and assuming $n < 1$, we get

$$\begin{aligned} \mathbb{E}[N_t] &= \mathbb{E}\Big[\mathbb{E}[N_t \mid M_t]\Big] \\ &= \mathbb{E}\Big[\mathbb{E}[M_t + G_t \mid M_t]\Big] \\ &\leq \mathbb{E}\Big[M_t + M_t \frac{n}{1-n}\Big] \\ &\leq \frac{1}{1-n}\mu t \\ &< +\infty. \end{aligned}$$

So, the expected number of arrivals in any finite interval $[s, t)$ with $s < t$, is finite

$$0 \leq \mathbb{E}\left[N_{[s,t)}\right] = \mathbb{E}[N_t] - \mathbb{E}[N_s] \leq \mathbb{E}[N_t] < +\infty,$$

which ensures that the counting process itself is almost surely finite and that there is no possibility of getting an infinite number of events in a finite window of time. For further details and a comprehensive discussion of this matter, the reader is referred to the relevant literature on this topic, such as Laub et al. [2015].

In the special case of the Hawkes process with the exponential decay, the branching ratio can be evaluated exactly:

$$n = \int_0^{+\infty} \alpha e^{-\beta \cdot t} dt = \frac{\alpha}{\beta} \int_0^{+\infty} \beta e^{-\beta \cdot t} dt = \frac{\alpha}{\beta}. \tag{2.9}$$

Therefore, to avoid explosion, it is sufficient to respect the $\alpha < \beta$ hypothesis. Hence, if this criterion is met then it is assured that the process cannot explode. Finally, it must be checked that such a process is well-defined. If we consider $\mu > 0$, we get that

$$\lim_{t \to +\infty} \int_{t_n}^{t} \lambda(t \mid H_t) \geq \lim_{t \to +\infty} (t - t_n)\mu = +\infty,$$

thus proposition 2.3 ensures that the conditional intensity function is valid and specifies a unique temporal point process.

Nonetheless, Hawkes processes push events to create clusters in the timeline and they self-excite themselves. Thanks to this, many real-world phenomena naturally fit this model. For example, referring again to modelling a social media platform, a new post on a specific topic may increase users' attention on that same matter triggering new activities, thus perfectly matching the process presented in this section.

2.3 Marked temporal point processes

Until now, we have dealt with processes generating events which are not distinguishable in type and which behave in the same way, but what would happen if different arrivals interact and influence each other in different ways? A new type of stochastic process must be introduced.

Definition 2.6 (Marked Temporal Point Process)**.** *Considering* $(\Omega, \mathcal{F}, \mathbb{P})$ *a probability space and* $\mathcal{K} = \{1, \dots, K\}$ *a categorical space, a marked temporal point process is a random process* $\left\{T_i : \Omega \to \mathbb{R}, M_i : \Omega \to \mathcal{K} \mid i \in \mathbb{N}\right\}$ *whose realisation consists of a sequence of discrete events which are characterised by their arrival time and mark.*

This mathematical concept allows us to keep focusing on the arrival times but at the same time it adds more information on the kind of events happening and their interaction. Thus, in this case, the definition of history is extended to contain both the arrival times and the marks that happened before $t \in \mathbb{R}$: $H_t = \{(t_i, m_i) \mid t_i < t\}$. Moreover, in the same way as in the case of the temporal point process, a counting process N_t could be associated to the events but this time they are divided per type. Similarly as in definition 2.1 and equation 2.2, a *marked intensity function* could be introduced as $\lambda_k(t \mid H_t)$ the instantaneous rate of k-marked events

conditioned on the history such that:

$$\begin{aligned}
\lambda_k(t \mid H_t)dt &= \mathbb{P}\big(T_{n+1} \in [t, t+dt)\,, M_{n+1} = k \mid H_t\big) \\
&= \mathbb{P}\big(T_{n+1} \in [t, t+dt) \mid H_t\big) \cdot \mathbb{P}\big(M_{n+1} = k \mid T_{n+1} \in [t, t+dt)\,, H_t\big)\big) \\
&= \mathbb{P}\big(T_{n+1} \in [t, t+dt) \mid T_{n+1} \notin [t_n, t)\,, H_{t_n}\big) \cdot \mathbb{P}\big(M_{n+1} = k \mid T_{n+1} \in [t, t+dt)\,, H_{t_n}\big) \\
&= \frac{\mathbb{P}\big(T_{n+1} \in [t, t+dt)\,, T_{n+1} \notin [t_n, t) \mid H_{t_n}\big)}{\mathbb{P}\big(T_{n+1} \notin [t_n, t) \mid H_{t_n}\big)} \cdot \mathbb{P}\big(M_{n+1} = k \mid T_{n+1} \in [t, t+dt)\,, H_{t_n}\big) \\
&= \frac{\mathbb{P}\big(T_{n+1} \in [t, t+dt)\,, T_{n+1} \notin [t_n, t) \mid H_{t_n}\big)}{1 - \mathbb{P}\big(T_{n+1} < t \mid H_{t_n}\big)} \cdot \mathbb{P}\big(M_{n+1} = k \mid T_{n+1} \in [t, t+dt)\,, H_{t_n}\big)
\end{aligned}$$

where dt is an infinitesimal interval and t_n is the last event in the history. Summing the intensities of all the possible types of event, the *ground intensity* is obtained and, since it coincides with the definition of intensity in the case with no marks, that same notation is used: $\lambda(t \mid H_t)$. Hence, it holds that

$$\lambda_k(t \mid H_t) = \frac{f_{T_{n+1}}\big(t \mid H_{t_n}\big)}{1 - F_{T_{n+1}}\big(t \mid H_{t_n}\big)} \cdot p_{M_{n+1}}\big(k \mid T_{n+1} = t, H_{t_n}\big)\big).$$

Furthermore, the conditional probability density function of the inter-arrival time of k-type events can be evaluated as

$$f_k(t|H_t) = \lambda_k(t|H_t) e^{-\int_{t_n}^{t} \lambda_k(u|H_u)du},$$

where t_n is the last event in the history.

Finally, considering $D := \{(t_1, m_1), \dots (t_N, m_N)\}$ a realisation of the process in the interval $[0, T]$, the joint conditional density function can be retrieved by applying the chain rule:

$$f(D|H_t) = \prod_{i=1}^{N} f(t_i, m_i \mid H_{t_i}).$$

There are many ways to deal with the designing of $f(t_i, m_i \mid H_{t_i})$, but the simplest and most relevant approach for this thesis is assuming conditional independence of the occurrences and the marks obtaining

$$f(t_i, m_i \mid H_{t_i}) = f(t_i \mid H_{t_i}) \cdot p(m_i \mid H_{t_i}),$$

where $p(m_i \mid H_{t_i})$ is the probability mass function of the categorical distribution and $f(t_i \mid H_{t_i})$ is the conditional density function which can be modelled as in the no marks case using the ground intensity function.

2.3.1 Mutually exciting Hawkes Process

A particular case of marked temporal point processes is the mutually exciting Hawkes process. Considering the exponential decay and K counting processes, then $N_k(t)$ is set such that its conditional intensity function is as follows

$$\lambda_k(t \mid H_t) := \mu_k + \sum_{m=1}^{K} \sum_{t_{i,m} < t} \phi_{k,m}(t - t_{i,m}; \alpha_{k,m}, \beta_{k,m}) \tag{2.10}$$

$$\overset{\text{exponential decay}}{=} \mu_k + \sum_{m=1}^{K} \sum_{t_{i,m} < t} \alpha_{k,m} e^{-\beta_{k,m}(t - t_{i,m})}, \tag{2.11}$$

where $\alpha_{k,m}$, $\beta_{k,m}$ and μ_k are non-negative constants and $t_{i,m}$ is the i^{th} arrival time of m-type event. Intuitively, the parameters μ_k represent the baseline of the rate, thus the part of the intensity function which is the initial rate in a simulation and does not depend on the history. On the contrary, the summation depends on the history and takes into account the influence of every activity that happened before time t. Specifically, $\alpha_{k,m}$ and $\beta_{k,m}$ denote the intensity and the decay rate of the effect of m-type event on the occurrence of m-type mark, which will be instantly and positively trigger by every previous action but their influence will decrease with time passing by. In real-life examples, the type of arrivals influences the intensity of how new cases are triggered but rarely affects how and when past events are forgotten. This is especially true in this thesis, thus we will consider an independent decay rate with respect to the mark in the history: $\beta_{k,m} = \beta_k$.

Example 2.4 (Mutually exciting Hawkes Process with $K = 3$)**.** *In this example, we consider a mutually exciting Hawkes Process characterised by $K = 3$ event types. The baseline intensities are represented by the vector*

$$\mu = \begin{pmatrix} 0.05 \\ 0.02 \\ 0.05 \end{pmatrix},$$

the excitation rates are described by the matrix

$$\alpha := (\alpha_{i,j})_{i,j} = \begin{pmatrix} 0.1 & 0.4 & 0.1 \\ 0.4 & 0.1 & 0.1 \\ 0.05 & 0.05 & 0.2 \end{pmatrix},$$

and the decay rate is set to

$$\beta = 1.0$$

Figure 2.2 shows a realisation of this mutually exciting Hawkes process. In each subplot, the solid lines depict the conditional intensity functions, which start from the baseline value and are influenced by events of any type. In each plot, the circular markers represent event occurrences, while vertical dashed lines report these occurrences in each intensity function plot. The colour of each line corresponds to the event type that occurred. These vertical lines highlight how an event of one type induces a jump in the conditional intensity function of another kind. The plot reveals that events of type 1 *and* 2 *influence each other significantly. On the other hand, the intensity function of type* 3 *events is less affected by other occurrences and causes small jumps to the other events' conditional intensity function.*

Regarding explosion, the calculations are more complicated than in the previous cases. Given the purpose of this chapter, only a non-detailed and more intuitive analysis is provided. For an in-depth discussion, the reader is referred to Hawkes [1971]. Nonetheless, let us adopt again branching theory terminology and define $\phi_k^{i,m}(t) := \alpha_{k,m} e^{-\beta k,m(t-t_{i,m})}$, thus representing the instantaneous ratio of k-type occurrence directly cased by the event $t_{i,m}$. To control the process and avoid explosions, it is necessary to contain the average number of offspring generated by a single event: let $Z_{k,m}^1$ be the random variable counting the number of k-type arrivals directly

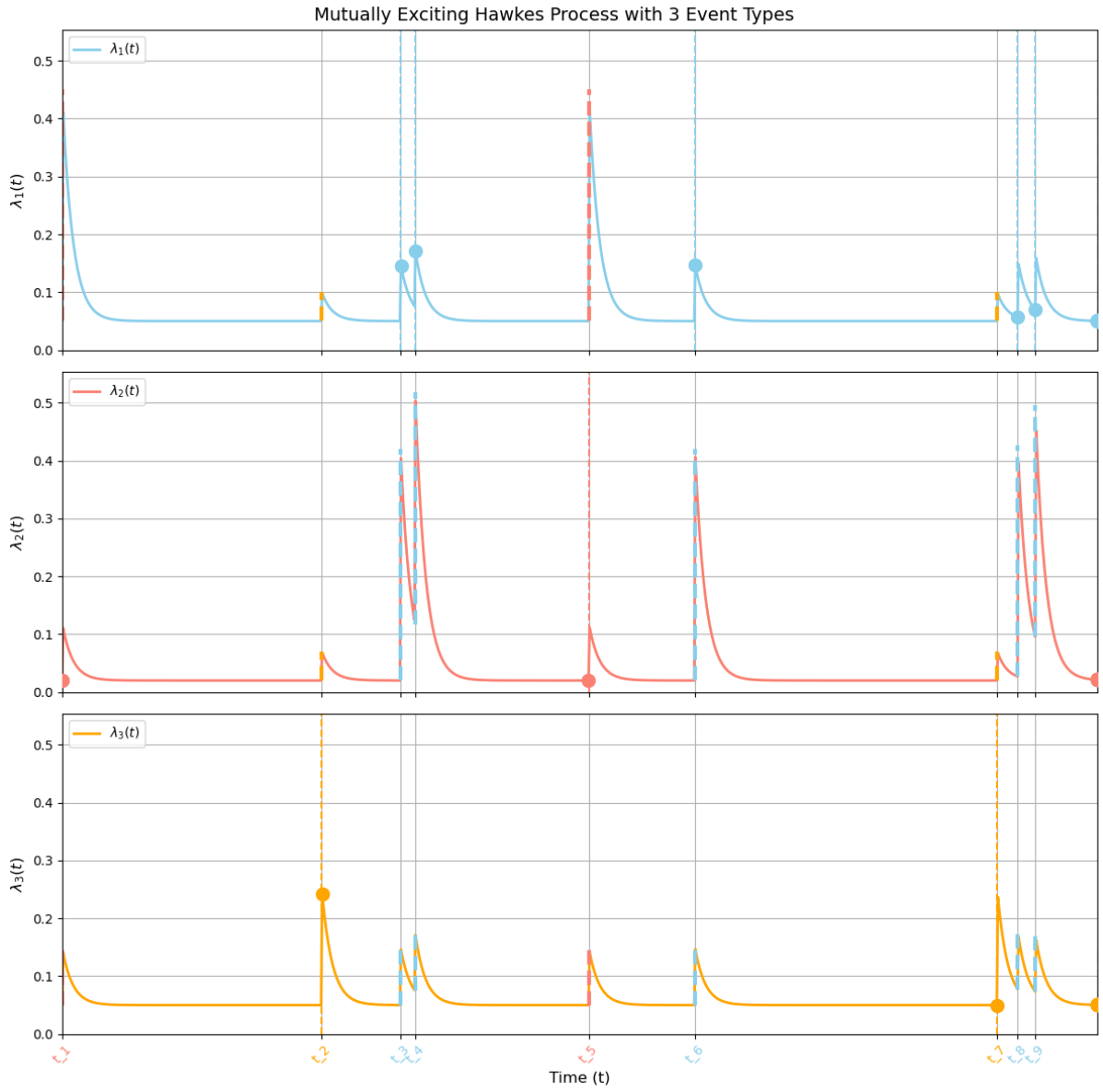

Figure 2.2: Mutually exciting Hawkes process' conditional intensity of Example 2.4

generated by event at time $t_{i,m}$ and let

$$n^1_{k,m} := \int_0^{+\infty} \alpha_{k,m} \cdot e^{-\beta_{k,m} t} dt = \frac{\alpha_{k,m}}{\beta_{k,m}} \int_0^{+\infty} \beta_{k,m} \cdot e^{-\beta_{k,m} t} dt = \frac{\alpha_{k,m}}{\beta_{k,m}} \tag{2.12}$$

be its average. Considering $\mathbf{Z}^l_m = \left(Z_{i,j}\right)_i$ the random vector representing the number of events in the l^{th} generation per type, we get that

$$\mathbf{n}^l_m = \left(n_{k,m}\right)_k := \mathbb{E}\left[\mathbf{Z}^l_m\right] = A \cdot \mathbb{E}\left[\mathbf{Z}^{l-1}_m\right] = A^l \cdot \mathbb{E}\left[\mathbf{Z}^0_m\right] = A^l \mathbf{e}_m,$$

where $A := \left(n^1_{k,m}\right)_{k,m}$ captures the interaction of each type of event on every other type and $\mathbf{Z}^0_m = \mathbf{e}_m$ is the m^{th} vector of the $\mathbb{R}^K$-Euclidean basis. Then the expected total number of descendants of the event $t_{i,m}$, stored in a vector divided per type, is equal to

$$\mathbf{n}^{\text{TOT}}_m = \sum_{l=1}^{+\infty} \mathbf{n}^l_m = \sum_{l=1}^{+\infty} A^l \mathbf{n}^0_m = \sum_{l=1}^{+\infty} A^l \mathbf{e}_m. \tag{2.13}$$

Therefore, it is possible to control the average number of total arrivals in a finite interval by exploiting the fact that for every kind of event we have an Homogeneous Poisson process and limiting the spectral radius [2] of the matrix A:

$$\rho(A) = \rho\left(\left[\frac{\alpha_{ij}}{\beta_{ij}}\right]_{i,j}\right) < 1. \tag{2.14}$$

Indeed, the last condition implies that the series converges for every m, thus it ensures that an arrival does not trigger infinite progeny. From this point and following the same approach as in Section 2.2.2, it follows that

$$\begin{aligned} ||\mathbb{E}\left[\mathbf{N}_t\right]||_1 &= \left|\left|\mathbb{E}\left[\left(N_{k,t}\right)_k\right]\right|\right|_1 \\ &\leq \left|\left|\sum_{l=0}^{+\infty} A^l\left(t\mu_k\right)_k\right|\right|_1 < +\infty, \end{aligned}$$

which implies that

$$\mathbb{P}(\mathbf{N}_t < +\infty) = 1.$$

Hence, the convergence of the series in the right term confirms that every counting process has a finite average thus the total number of events in $[0, t)$ is almost surely finite, which in turn ensures that no explosion could happen.

Summing up, the mutually self-exciting Hawkes processes are the appropriate stochastic processes when it comes to modelling events of various types triggering each other differently. Not only they inherit the power of the simple Hawkes process, but also they allow to store more information on the arrivals and to better model the influences between different marks. Moving to the theme of social media, an example of their use could be the arrival of posts written by specific accounts. Thus, an event consists of a given arrival time and a mark, which represents the user writing it. Given that different people influence each other in different ways, the marked temporal point process presented in this section perfectly models this situation.

2.4 Temporal point processes sampling

Until now, we have encountered some simple cases of temporal point process and we have focused especially on their definition, how to perform inference on them, and how to prevent explosion. However, we still need to discuss what happens when it comes to data simulation. This section will be devoted to that, reporting three of the most popular techniques: the *transformation method*, the *thinning method* and *Ogata's modified thinning method*. Finally, a re-adaptation of the last technique to generate marked temporal point process is presented in the last subsection.

2.4.1 Transformation method

The idea behind this method is based on the simplicity of simulating a simple Homogeneous Poisson process.

[2] Recall that the spectral radius of a matrix is the largest absolute value of the matrix's eigenvalues.

Algorithm 2.1 (Simulating Homogeneous Poisson process)**.** Starting from a realisation $\{u_1, u_2, \dots\}$ of uniform random variables $U_i \sim Uniform(0,1)$, it is possible to retrieve a sample from the exponential distribution by inverting the distribution function: from $F(s \mid H_s) = 1 - e^{-\lambda^* s} = u$, it is obtained that $s = -\frac{log(1-u)}{\lambda^*}$. Since

$$F_S(s) = \mathbb{P}(S \leq s) = \mathbb{P}(-\frac{log(1-U)}{\lambda^*} \leq s) = \mathbb{P}(U \leq 1 - e^{-\lambda^* s}) = 1 - e^{-\lambda^* s},$$

the new set $\{s_1, s_2, \dots\}$ is a realisation of independent and identically distributed exponential random variables, thus $S_i \sim exp(\lambda^*)$, as required for the inter-arrival times. To obtain the actual arrival times, it is enough to add up the time intervals: $t_i = \sum_{j=1}^{i} s_i$.

Thus, if it is possible to modify the data generated by a Homogeneous Poisson in such a way that the probability function assumes the expected conditional density function, it would be possible to generate a temporal point process with any conditional intensity function. The following proposition is exactly what is needed.

Proposition 2.4. Considering $\{S_i\}_i$ a unit rate Poisson process and assuming that all the following quantities exist, then the temporal point process defined as $T_i := \Lambda^{*-1}(S_i)$ has the conditional intensity function equal to $\lambda(t_i \mid H_t) = \frac{d\Lambda^*}{dt}(t_i)$.

Proof. Assume that the proposition holds for the first n points, let's check what happens to the next arrival:

$$\begin{aligned} F(t \mid H_{t_n}) &= \mathbb{P}(T_{n+1} \leq t \mid H_{t_n}) \\ &= \mathbb{P}(\Lambda^{*-1}(S_i) \leq t \mid H_{t_n}) \\ &= \mathbb{P}(S_i \leq \Lambda^*(t) \mid H_{t_n}) \\ &= 1 - e^{-1 \cdot [\Lambda^*(t) - s_n]} \\ &= 1 - e^{-[\Lambda^*(t) - \Lambda^*(t_n)]} \\ &= 1 - e^{-\int_{t_n}^{t} \lambda^*(u) du}. \end{aligned}$$

Hence, the random variable T_{n+1} has the right distribution function, thus its conditional intensity function corresponds to what is declared in the proposition. Furthermore, given that the case $n = 0$ is trivial, the thesis is proven by induction. □

Therefore, starting from a realisation of a Homogeneous Poisson with rate $\lambda^* = 1$, it is possible to retrieve a sample from any temporal point process whose compensator is invertible. A simulation in any given interval $[0, T]$ could be obtained via the following algorithm.

Algorithm 2.2 (Simulation by transformation)**.** Setting $t = 0$ and $n = 0$, repeat the next steps until $t > T$:

1. Generate s from an exponential distribution with rate $\lambda^* = 1$,
2. Evaluate $t = \Lambda^{*-1}(s)$,
3. If $t < T$, update $n = n + 1$ and let $t_n = t$ an arrival of the process.

The obtained output is $\{t_1, ..., t_n\}$.

This algorithm allows the generation of a sample from a temporal point process, but it requires the additional complexity of inverting the conditional intensity function. As a matter of fact, the compensator could be in general non-invertible and, even when it is invertible, it is not always possible to evaluate the inverse function exactly, creating the necessity to appeal to numerical methods.

2.4.2 Thinning method

Another way to simulate a temporal point process is to start from a Homogeneous Poisson process with a faster rate M. With faster rate, it is meant a constant rate that is always greater than the conditional intensity function of the temporal point process. In particular, the conditional intensity function needs to be a limited function for this method to be applicable, and this is not always the case. Every generated arrival is then accepted or rejected depending on a specific probability related to the value of the actual conditional intensity function.

The tricky point of this method is finding an appropriate rate for the initial sampling. Indeed, it is required to find the global maximum of the conditional intensity function, but this does not always exist. Thus, it is necessary to restrict the class of temporal point process to the ones independent from the history (otherwise the maximum is not uniquely defined) and with an existing $M := \max_t \lambda(t)$. If these conditions are met, then it is possible to generate a sample from the temporal point process in an interval $[0, T)$ by exploiting the following algorithm.

Algorithm 2.3 (Simulation by thinning)**.** Setting $t = 0$ and $n = 0$, repeat the next step until $t > T$:

1. Generate s from an exponential distribution with rate $\lambda^* = 1$,
2. Update $t = t + s$ and generate u from a $Uniform(0, M)$,
3. If $t < T$ and $u \leq \lambda(t)$, update $n = n + 1$ and let $t_n = t$ a new arrival time.

The obtained output is $\{t_1, ..., t_n\}$.

Intuitively, this procedure rescales the simulation conditional intensity function to $\frac{\lambda(t)}{M}$, thus generating a sample with the needed rate. For the detailed proof, the reader is referred to Chen [2016].

2.4.3 Ogata's modified thinning method

Is there a way to modify the aforementioned thinning method to be available for a larger class of temporal point processes? The answer is yes. Ogata modified the previous method so it could be extended to every temporal point process with a non-increasing conditional intensity function in periods with no arrivals.

The idea is to adjust the classical thinning method simply by updating M at every step of the algorithm.

Algorithm 2.4 (Simulation by Ogata's modified thinning)**.** Setting $t = 0$, $n = 0$ and $\epsilon << 1$, repeat the next step until $t > T$:

1. Find $M = \lambda^*(t + \epsilon)$,
2. Generate s from an exponential distribution with rate $\lambda^* = M$,
3. Update $t = t + s$ and generate u from a $Uniform(0, M)$,
4. If $t < T$ and $u \leq \lambda^*(t)$, update $n = n + 1$ and let $t_n = t$ a new arrival time.

The obtained output is $\{t_1, ..., t_n\}$.

The reader is referred to Ogata [1981] for the detailed discussion on this method.

2.4.4 Ogata's method for marked temporal point process

Until now, we have gone through methods only regarding the simulation of simple temporal point processes, but how could we generate data of marked temporal point processes? The most natural way of dealing with this problem is to extend Ogata's modified method.

However, this method inherits the necessary condition to apply Ogata's method: the intensity function must not increase its values when no events arrive: a condition respected by mutually self-exciting Hawkes processes.

Let's consider a process composed of $\{N_m \mid m = 1, \ldots, K\}$, K possible counting processes, each associated with a specific mark. The basic idea of this method is to simulate events from a faster Homogeneous Poisson process, thin out some cases and choose the mark depending on the intensity function of each process.

Algorithm 2.5 (Simulation of marked temporal point processes)**.** Setting $t = 0$, $n = 0$ and $\epsilon << 1$, repeat the next step until $t > T$:

1. Find $M = \sum_{k=1}^{K} \lambda_k^*(t + \epsilon)$,
2. Generate s from an exponential distribution with rate $\lambda^* = M$,
3. Update $t = t + s$ and generate u from a $Uniform(0, M)$,
4. If $t < T$ and $u \leq \sum_{k=1}^{K} \lambda_k^*(t)$:
   (a) Update $n = n + 1$ and let $t_n = t$ a new arrival time,
   (b) Let m_n be the smallest integer m such that $u \leq \sum_{k=1}^{m} \lambda_k^*(t)$.

The obtained output is $\{(t_1, m_1), ..., (t_n, m_n)\}$.

Chapter 3

Capturing coordinated behaviour

Up to this point, we have presented a general discussion of disinformation detection and some theoretical aspects of temporal point process. In this chapter, we will merge these two fields by exploring how it is possible to leverage marked temporal point process for detecting coordinated behaviour. Capturing the temporal patterns of online activities is fundamental for revealing coordination campaigns. In fact, models that relies on this stochastic processes can detect anomalous group of users depending on their timing and interaction dynamics. In particular, they could identify subtle temporal patterns of coordination that might otherwise go undetected. Therefore, these models represent a valuable weapon in the fight against disinformation.

The key idea behind this approach is to leverage the vast amount of activity data on social media to extract user features related to the timing of their interactions. These features are then utilise for detecting inhomogeneous patterns indicative of coordinated behaviour. However, the huge amount of social activities and their disorganise structure represent a challenge. Not only it is necessary to find a way of efficiently summarise the historical activities of each account, but it is fundamental to do so in a systematic and structural way. In Section 3.1, we address this challenge by introducing masked self-attention, an extremely powerful tool that allows us to reduce users' interaction histories into simple vectors of real numbers. In this way, it is possible to summarise the available extensive information.

Next, we introduce the Attentive Mixture Density Network with Hidden Account Group Estimation (AMDN-HAGE) in Section 3.2. This model is a state-of-the-art detection method that is based on temporal patterns and masked self attention. Moreover, in Section 3.3, we discuss some possible extensions of the aforementioned model which have been designed to overcome its limitations and enhance its ability to detect coordination more accurately. Specifically, these extensions aim to improve detection by refining how users are grouped together. We believe that these implementations result in more coherent clusters, ultimately improving the overall performance of the detection model.

3.1 Self-Attention for summarising the history of MTPP

The introduction of the attention mechanism has deeply changed deep learning history, revolutionising the entire field. Transformers, which are compelling attention-based tools, have revolutionised the whole field. For instance, Generative Pre-trained Transformers (GPTs) have outperformed precedent Large Language Models, starting a new era of generative artificial intelligence and text translations. Then, an immediate question arises: are the prominent attention-based models valuable in the context of temporal point process and coordination detection? To be able to answer this question, we need to introduce the concept of attention. In particular, the methodologies we will discuss are based on the foundational work presented in Vaswani [2017], an accessible interpretation by Sanderson [2024] and further advancements that adapt attention mechanisms to temporal point process as applied in Xu et al. [2019] and Sharma et al. [2021].

When reading long texts or translating sentences, humans are capable of understanding complex structures and deep meanings of words depending on the context they are immersed in. Generally, we keep our focus on some parts of the sentence and update its meaning depending on other words interacting with them. For instance, let us consider the sentence "Temporal point processes are valuable tools in mitigating disinformation campaigns". There are many ways of how a person reads it and catches its meaning but one possible way of doing so is focusing first on the nouns. We read, for example, the word "processes" and then we could update its meaning with the adjectives "point" and "temporal". Similarly, we could update the meaning of "tools" with "valuable" and of "campaigns" with "disinformation". Subsequently, the word "are" associates the updated meaning of "processes" and "tools", and the verb "mitigate" associates the subject and the object of the action. Hence, by reading a word in a sentence, we naturally understand where its focus goes and where its attention is directed, and we use this to update the meaning of its surrounding terms. Attention mechanism tries to mimic this behaviour.

At first, the inputs, which could be texts, sequences of events or other types of data, are broken down into smaller pieces, which are referred to as *tokens*. Each of these pieces is then associated with a fixed-dimension vector referred to as *embedding*. These vectors describe in some ways the features of the tokens, reporting their meanings. The key point is that the vectors whose tokens have similar meanings are close in the embedding vector space. At this point, the goal is to evaluate the influence of a token over another token. If we understand how to do so, we can successfully update their meaning depending on the influence they have on each other. To estimate this influence, specific vector transformations and scalar products can be used. These operations constitute a module called *attention heads*, we will elaborate this topic in the next subsections.

As mentioned above, these computed influences can be used to direct the attention of a token towards the tokens that exert the most influence on it. The model then updates the token's embedding accordingly. Repeating the procedure for some iterations will return vectors encoding every token's interactions and impacts. If the required outputs are embeddings, these representations could be directly used. Otherwise, they can eventually be modified by different operations

to retrieve the output tokens. This ultimate technique is referred to as *unembedding*. Therefore, the final result of the entire mechanism could lead to a form of pure embeddings, a new text, a vector summarising the sequence history of a temporal point process, among numerous other possibilities. If the output are tokens living in the same space of input tokens, the process is called *self-attention*. We will focus on this case. In particular, in the following sections we will discuss every part of this process and how it could be effectively applied to data originated from marked temporal point processes.

3.1.1 Embedding matrix

So far, we have discussed how each token should be associated with a specific embedding. Assuming that their fixed length is equal to m, the embedding space is the real vector space $\mathbb{R}^m$. We refer to these vectors with the following notation[1]:

$$X_{\text{token}} \in \mathbb{R}^m.$$

In such a space, the dot product between two embeddings describes how well two tokens resemble each other, a fundamental characteristic for understanding the token's influence. When two tokens have similar meanings, their vectors should be roughly aligned thus their dot product should be positive. On the other hand, if their meanings are not related to each other, their dot product should be equal to 0 or negative.

Regarding the association of the vector space and the token features, it was mentioned that the embeddings should encode the features of their tokens and they should be updated depending on the surrounding context. To clarify this, let us consider the case in which inputs are sentences and each word is a token. The directions in the embedding vector space should somehow carry a semantic meaning. For instance, once updated by the adjectives "temporal" and "point", we would expect the vector X_{process} to move closer to the words from the mathematical fields. On the other hand, if the adjective associated with "process" was "sale", then its embedding vector would have moved towards embeddings of tokens coming from the economic field. Similarly, there could be a direction that in some ways resembles the plural of words. Then, the difference $X_{\text{processes}} - X_{\text{process}}$ should be very similar to the vector $X_{\text{campaigns}} - X_{\text{campaign}}$, thus their dot product should be positive. Of course, it would be impractical to control everything by hand and construct a direction for every possible meaning or concept. Machine learning comes to our rescue, making these embeddings learnable quantities.

3.1.2 Queries and keys

Once every token is translated into an embedding, it is time to evaluate how each token changes meaning depending on the context, which is constituted by the surrounding tokens. To do so, the concepts of queries, keys and values should be introduced.

[1] In order to standardise the notation with Sharma et al. [2021], vectors will henceforth be considered row vectors. Moreover, notations such as $\left(\square^i_j\right)_j$ indicate a row vector wherein the j index varies across each vector element, while the i index remains constant throughout. In addition, $\left(\square^i_j\right)_{i,j}$ denotes a matrix where i and j indices vary along the row and column dimensions, respectively.

We could imagine a *query* as a representation of a question that a token asks. It should regard something related to the context the token is immersed in and it should try to understand if and how the token interacts with its surroundings. In fact, its objective is adjusting the embedding depending on how other tokens reply. These answers are referred to as *keys*. Mathematically, this question-and-answer formulation can be encoded into matrices multiplication. Specifically, the weight matrices W_q and W_k transform embeddings E_i and E_j into the query Q_i and the key K_j, which are two vectors of length d_k within the same dimensional space $\mathbb{R}^{d_k}$:

$$Q_i = E_i W_q,$$
$$K_j = E_j W_k.$$

The dot product between K_j and Q_i could be used to understand how well the former answers to the latter. If the two vectors align, then the dot product is positive and we will say that the key *attends to* the query. Otherwise, if the dot product is negative then the key is not significant for the query. The alignment should then be calculated for each existing token, which will formulate an answer to the query Q_i, obtaining

$$\left(w_j^i\right)_j := \left(Q_i \cdot K_j\right)_j \qquad (i \text{ fixed}),$$

a list of values indicating the influence they have on the token who formulate the question.

Subsequently, a softmax layer can be used to transform the list into discrete probability values. In this way, every weight w_j^i is converted into a non negative value a_j^i. Additionally, summing over i, these numbers sum up to 1. In accomplishing this operation, care must be taken when dealing with negative weights. Since these keys do not have any influence on the token in the query, the most natural way is to set every negative value to $-\infty$. This automatically drives the softmax to output those a_j^i as equal to 0. This procedure is known as *masking*. Moreover, another operation that is run on each weight w_j^i is dividing it by $\sqrt{d_k}$ before the softmax layer. It was shown that this procedure improves numerical stability. Summing up, for every query Q_i, we calculate the dot product with each key K_j. We thus apply a transformation such that the resulting quantities represent the influences associated with each key attending to the query. These final quantities are called *attention weights*. Mathematically, fixing i, we obtain a_j^i for every j as

$$a_j^i := \text{softmax}\left(\frac{Q_i \cdot K_j}{\sqrt{d_k}}\right). \tag{3.1}$$

Since the same query is repeated for every token, we can reconstruct a unique matrix holding all the weights. This matrix is called the *attention matrix* and it can be computed as

$$A = \left(a_j^i\right)_{i,j} := \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right), \tag{3.2}$$

where the softmax is intended by row. Again, preparing questions and formulating them as matrix multiplication would be extremely complex. Luckily, the best questions to ask and their translations into matrices W_q and W_k are automatically learnt by the model from data without any manual operation needed.

To move onto a practical case, let us consider again the case of semantic inputs. A question that could be asked is if there are any adjectives associated to a token. To understand if that is the case, we take a token, for instance the word "process", and transform it into its query by multiplying it by W_q. Then, for every possible key, we evaluate the associated attention weight. These final values would be 0 everywhere except for "temporal" and "point" which attend to the query. This result show that, with respect to the query of looking for adjectives, the two tokens with positive attention weights highly influence the considered token. Hence, they should be used to update its meaning.

3.1.3 Values and attention heads

Considering a query Q_i and a key K_j attending to it, we want the embedding vector of the token in the former to be updated depending on the influence it receives from the latter. To do so, W_v, another weight matrix, is introduced. Assuming that there is a perfect attendance to it, meaning $a_j^i = 1$, this matrix transforms the key embeddings to the actual vector we want to update the query embeddings with:

$$V_j := E_j \cdot W_v. \tag{3.3}$$

Notice that the resulting vector should be part of $\mathbb{R}^m$, the same vector space as the embeddings. Practically, to reduce the number of parameters, the matrix W_v is often constrained to be a low-rank matrix.

Many keys could attend to a query with different weights, influencing the considered token in different amounts. Therefore, before applying the changes to the embedding of a token, each value vector V_j is multiplied by the attention weights a_j^i. In this way, we will give more importance to the values associated with highly influencing keys. Hence, we assign more weight to the influential updates with respect to less effective values, which anyway maintain a low but still positive influence value. Each embedding vector E_i is then updated as

$$E_i' = E_i + \sum_j a_j^i V_j.$$

This resulting vector should better resemble the meaning of its token and the surrounding context. For instance, let us go back to the example of the previous paragraph. The values V_{temporal} and V_{point} are the vectors that would update the embedding $E_{\text{processes}}$. Therefore, we would obtain the new vector

$$E'_{\text{processes}} := E_{\text{processes}} + a^{\text{processes}}_{\text{temporal}} V_{\text{temporal}} + a^{\text{processes}}_{\text{point}} V_{\text{point}}.$$

Repeating this procedure for every token in the considered sentence would allow to update every noun respect to a new embedding encoding also the meaning of its adjective, thus improving the representation quality.

Concisely, it is possible to write every computation as a unique operation:

$$H_{attn} := AV = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) V, \tag{3.4}$$

where the matrix V concatenates every value vector. The resulting matrix H_{attn} is referred to as *attention head*. Therefore, it turned out that a very complex procedure as the attention could be described by simple transformations in a vector space. Moreover, the matrices appearing and the embeddings description as well could be learnt via gradient-based algorithms, for example the ADAM method [Kingma, 2014, see].

We have now considered only one possible query, key and value at a time. In reality, in many models, there are numerous ways in which tokens can influence each other. For this reason, multiple queries, keys and values are often considered, creating *multi-head attention* models running various attentive layers in parallel. The resulting embeddings should be updated considering each one of these layers. However, these other layers augmented the number of parameters and the computational time. Therefore, we have decided to utilise a single attention head for our models.

This attention head is usually utilised as input for other layers. For instance, self-attention heads and 2-layer feed-forward networks are iteratively repeated in transformers. Hence, the embeddings are continuously updated depending on the newly discovered features of tokens. The output of the final iteration could be simply the vector of the last token, imagining it has collected all the input information. Alternatively, a further recurrent network layer may be used to summarise the results from the embeddings of every token.

3.1.4 Adaptation for marked temporal point process

To adapt the concept of self-attention to marked temporal point processes, we will consider a self-attention head mechanism. In this context, the input tokens are events formed by couples of marks and inter-arrival times (m_i, t_i). Moreover, we will feed the attention head results H_{attn} into a normalisation layer, a dropout layer, and a feed-forward layer. Recall that a normalisation layer receives batches of values and it returns them normalised [Ba, 2016], a dropout layer randomly sets to zero some elements of the input to avoid overfitting [Hinton, 2012], and a feed-forward neural network is a type of neural network where the information flows in one direction only, from the input layer towards the output, with no loops admissible. Therefore, the final output is a fixed-dimensional vector that encodes the entire history.

However, it is necessary to modify the standard attention mechanism. We have already seen in Chapter 2 that marked temporal point processes are causal. Therefore, it would not make any sense for previous tokens to be influenced by future tokens. Hence, a new mask should be introduced. Assuming that tokens are ordered depending on their arrival times, the attention weight matrix should be modified to set the upper triangular part to 0 everywhere. This could be represented as

$$H_{attn} := \left(\text{softmax} \left(\frac{QK^\top}{\sqrt{d_k}} \right) + M \right) V, \tag{3.5}$$

where M is the *mask* matrix.

Another delicate task is to take care of the embeddings. While the codification of marks can be handled exactly as in the semantic context, dealing with temporal information operates on

a different level. In fact, time introduces a continuous dimension that requires specialised techniques. Thankfully, the adaptation proposed in Xu et al. [2019] can be directly used. Specifically, we will encode inter-arrival times with the *translation-invariant temporal kernel functions* $\phi(\tau_i)$. Additionally, we will enforce the importance of the ordering by applying trigonometric integral functions for positional encoding $PE_{pos=i}$, as used in Vaswani [2017]. The reader is referred to these two articles for a complete discussion. Nevertheless, the embedding vector of the token (m_i, t_i) is the concatenation of the three row vectors:

$$X_i = [E_{u_i}, PE_{pos=i}, \phi(\tau_i)]\,. \tag{3.6}$$

The dimension of this layer is m, which must coincide with the sum of the inner vectors' dimensions.

Summing up, every embedding vector of the input tokens goes into the self-attention mechanism, and its value is updated by every previous event that attends to it. Subsequently, the resulting vectors pass through some layers for normalisation and over-fitting avoidance. Finally, the embedding passes through a feed-forward network. This final representation is an m-dimensional vector containing the information of the entire history H_t, and it will be referred to as *context* vector.

3.2 AMDN-HAGE model

The Attentive Mixture Density Network with Hidden Account Group Estimation (AMDN-HAGE) is a generative model that integrates Temporal Point Processes with Gaussian Mixture Models to detect inauthentic accounts. This approach targets coordinated users by identifying those accounts that collectively have anomalous features. Sharma et al., the developers of this method and authors of the related paper Sharma et al. [2021], expected that such groups of coordinated accounts would stand out due to their unconventional behaviour patterns, making them distinguishable from authentic users.

To understand how this model is structured, it is essential to introduce the concept of activity traces first. An activity trace, also known as *cascade*, refers to a sequence of events users have performed on a social network. In the context of this model, these events can be represented as a list of ordered pairs containing relevant information about each activity. In particular, each pair is composed by the arrival time of the event t_i and the user u_i who performed the action. Two events that are related to each other are part of the same trace. More precisely, cascades could be used to represent users sharing the same hashtag (co-hashtag), replying to the same post (co-reply), re-sharing the same tweet/post (co-retweet), participating in the same conversation (co-conv) or sharing the same website link (co-URL). If systematic, these co-actions may reveal coordinated accounts. Nevertheless, within the context of this model, the construction of cascades is flexible, and the only essential inputs are the timestamps and the author's identifier of every action. This flexibility allows the model to be applied across different types of social media and based on different interactions. Moreover, in this framework, activity traces can be viewed as realisations of marked temporal point processes, where each user serves as the

mark and each timestamp as the arrival times of events. Consequently, independently from the considered co-action, a cascade C_s can be represented as

$$C_s = [(u_1, t_1), (u_2, t_2), \ldots, (u_n, t_n)]\,. \tag{3.7}$$

This formulation enables the model to capture the temporal dynamics and interactions within the network, regardless of how the cascades are specifically defined, making it a versatile tool for analysing user behaviour and detecting coordinated activity.

Using their observed activity traces, the goal is to describe each user's behaviour effectively. To achieve this, users' embedding should be introduced. The model assumes that every active account u_i is associated with a latent vector $E_{u_i} \in \mathbb{R}^m$, where m is the fixed dimension of the vectors. These vectors encapsulate the features of users based on their participation in activity traces in the same way as we explained in Section 3.1. In particular, users with similar behaviours are expected to have embeddings that are close in the vector space, whereas accounts involved in significantly different activity traces are envisioned farther apart from each other. To simplify notation, we define the total embedding matrix E, which contains the embeddings of all users in the system, as follows:

$$E := \begin{bmatrix} \frac{E_{u_1}^\top}{} \\ \frac{E_{u_2}^\top}{} \\ \frac{\cdots}{} \\ \frac{E_{u_{|U|}}^\top}{} \end{bmatrix}.$$

Here, each row corresponds to the transpose of an individual user's embedding vector, $E_{u_i}^\top$, and the matrix E provides a compact representation of the behavioural features for all users in the network. This embedding matrix serves as the foundation for further analysis and clustering within the model.

Considering the activity traces of accounts in a social network and the corresponding users' embeddings, the primary objective of the model is to construct a membership function that associates each account with a specific cluster. Assuming the total number of clusters is predetermined and denoted with N, the goal is to define the function

$$M : U \to \{1, \ldots, N\}, \tag{3.8}$$

which assigns each user to one of the N clusters. In the previous equation, U represents the set of all users in the social network. Once a clustering is obtained, the next step is identifying groups that deviate from the others in size or heterogeneity. These anomalous groups are likely composed of coordinated accounts, as their behaviour differs significantly from the other clusters. Additionally, the actual embeddings and the underlying activity traces model can be leveraged to further analyse and interpret the characteristics of these clusters, enhancing the detection.

How do we structure a model capable of achieving everything mentioned so far? Sharma et al. opted to divide their coordination detection method into two key components: the Attentive Mixture Density Network (AMDN), which models observed activity traces, and the Hidden Account Group Estimation (HAGE), which handles the clustering of accounts. In the following

sections, we will describe the implementation of each component, explain how they are integrated to efficiently learn the account embeddings, the parameters of the activity trace model and the membership function, and illustrate how the model operates as a cohesive system to achieve its coordination detection goals.

3.2.1 AMDN component

As mentioned above, the first component of Sharma et al.'s method focuses on modelling the activity traces by treating them as realisations of stochastic processes. Specifically, conditioning on the set of users involved in a cascade and their embeddings E, the probability density of an observed activity trace C_s can be formulated as the distribution of a marked temporal point process, see Section 2.3. In particular, by assuming conditional independence[2] between the timestamp variable and the user variable, the log-likelihood of the observed traces can be decomposed into two components for each event:

$$\log p(C_s \mid U, E, \theta_a) = \sum_{i=1}^{|C_s|} \left[\log p_{E,\theta_a}(t_i \mid H_{t_i}) + \log p_{E,\theta_a}(m_i \mid H_{t_i})\right]. \tag{3.9}$$

This factorisation allows us to separately model the timestamp component and the user selection component.

The first step is to find a proper way to handle the history, which is a significant problem when dealing with vast amounts of data and inputs of different lengths. An adaptation of the masked self-attention technique described in Section 3.1.4 has been used to summarise the entire history. In particular, the embedding vector E_u has been defined as in Equation 3.6 and the attention head as in Equation 3.5. Therefore, starting from H_t, the encoding provides the attention outputs, which in turn are transformed into the final output $C \in \mathbb{R}^{|C_s| \times m}$. The resulting matrix contains the fixed-dimensional vector

$$C_i := (C_{i,j})_j \,,$$

which summarises H_{t_i}, the history of every occurrence happening up to time t_i. Since these vectors collect the event embedding and previous events' influence on it, they are referred to as *context vectors*.

Subsequently, it is possible to model the temporal component by exploiting context vectors. It is well known that, with an appropriate choice of parameters, mixture distributions can be exploited for approximating any target distribution with real support to any arbitrary level of accuracy [O. et al., 2019, see]. Due to this property and the non-negative domain of inter-arrival times, the mixture of log-normal distribution has been selected to model $\tau_i := t_i - t_{i-1}$. As a result, the density function of the inter-arrival times, defined over the domain $\mathbb{R}_+$, have been expressed as

$$\log p_{E,\theta_a}(\tau_i \mid C_i) := \sum_{k=1}^{K} \omega_i^k \frac{1}{s_i^k \sqrt{2\pi}} \exp\left(-\frac{\left(\log \tau_i - \mu_i^k\right)^2}{2\left(s_i^k\right)}\right). \tag{3.10}$$

[2]This independence is conditioned on the users embeddings and the entire history.

The parameters of the previous equation have been chosen as transformations of the context vector. Specifically, they have been obtained as

$$\begin{aligned}\omega_i &= \text{softmax}\left(V_\omega C_i + b_\omega\right),\\ s_i &= \exp\left(V_s C_i + b_s\right),\\ \mu_i &= V_\mu C_i + b_\mu,\end{aligned}$$

where $V_\square$ and $b_\square$ are learnable parameters and the non-linear transformations were added to make sure that ω_i and s_i are always positive. This mixture allows the system to adaptively fit the distribution of inter-arrival times depending on the context provided by the history. Moreover, the flexibility of the log-normal mixture ensures that the model can accurately represent the complex temporal pattern arising in the cascades. Practically, the number of components to use is set before the training process depending on the desired level of preciseness.

To model the user component, it is crucial to select a model that appropriately captures the finite and discrete nature of the user domain while also considering the influence of the history. Sharma et al. address this requirement by employing a Multi-Layer Perceptron (MLP)[3] on the context vector with a softmax layer. Mathematically, the distribution has been defined as

$$\log p_{E,\theta_a}(u_i \mid C_i) := \text{softmax}\left(MLP\left(C_i\right)\right)_{u_i}. \tag{3.11}$$

This formulation allows this model to capture the dependencies between users and their activities within each cascade. Moreover, based on the previous activity, it finds which user is likely to act next. We refer to all the parameters of this encoder-decoder architecture as θ_a.

3.2.2 HAGE component

The second component of the model focuses on the subdivision of accounts U into latent groups. In particular, the focus is on properly modelling the probability distribution of the accounts participating into one specific cascade given the users' embeddings E. Once it was assumed that the number of groups is fixed to N and that users are independent conditioning on their embedding vectors, Sharma et al. opted to model users' latent group by employing Gaussian Mixture Models[4] (GMM) on the users' embedding vector space.

Hence, in this setup, a hidden group is composed of embedding vectors which are supposed to be drawn from a multivariate Normal distribution. Specifically, the probability of the user u_j being active in a specific cascade depends on the likelihood of the vector $E_{u_j} \in \mathbb{R}^m$ to be drawn from the mixture. Mathematically, the probability that u_j is active is equal to

$$p_{\theta_b}\left(u_j \mid E\right) = \sum_{i=1}^{N} p_{\theta_b}\left(u_j, i \mid E\right) = \sum_{i=1}^{N} w_i \cdot p\left(E_{u_j} \mid \mu_i, \Sigma_i\right),$$

where θ_b contains every parameter related to the HAGE component. Among these parameters, it also includes μ_i and Σ_i, which are the mean and covariance matrix of the i^{th} component, and w_i,

[3] Recall that a Multi-Layer Perceptron is a feed-forward neural network with three or more layers

[4] Recall that a Gaussian Mixture Model is a probabilistic model assuming that a mixture of a finite number of Normal distributions generates the data.

which is the prior probability of being in the i^{th} group. Moreover, with $p_{\theta_b}(u_j, i \mid E)$ we denote the probability that the embedding vector E_{u_j} is drawn specifically from the i^{th} component of the mixture.

Therefore, it is possible to evaluate the log-likelihood of the entire set of active accounts by computing

$$\log p(U \mid E, \theta_b) = \sum_{j=1}^{|U|} \log \left[\sum_{i=1}^{N} w_i \cdot p\left(E_{u_j} \mid \mu_i, \Sigma\right) \right]. \tag{3.12}$$

For simplicity, each component has been assumed to share the same covariance matrix Σ, which is further assumed to be diagonal.

The hidden groups can be exposed only when the user embeddings and the HAGE parameters are learnt. When this is the case, the posterior probability of users being drawn from a specific component must be obtained. Exploiting conditional probabilities laws, this quantity turns out to be equal to

$$p_{\theta_b}(z_{i,j} = 1 \mid E) = \frac{p\left(E_{u_j} \mid \mu_i, \Sigma_i\right)}{\sum_{k=1}^{N} p\left(E_{u_j} \mid \mu_k, \Sigma_k\right)},$$

where $z_{i,j} = 1$ represents the event that the user u_j is part of the latent group i. This is a well-established result within the framework of mixture models and the expectation maximisation algorithm.

3.2.3 Integration of AMDN and HAGE components

In the previous sections the two components of the AMDN-HAGE have been modelled, let us now describe how they have been merged together and how the learning process can be performed.

Considering a cascade C_s, which is represented as in Equation 3.7, and its set of active users U, it is necessary to evaluate its likelihood. Exploiting Equation 3.9 and 3.12, it can be obtained that the log-likelihood of a given cascade C_s and a set of active users U is equal to

$$\begin{aligned}
\log p_{\theta_a,\theta_b}(C_s, U \mid E) &= \log p_{\theta_a}(C_s \mid U, E) + \log p_{\theta_b}(U \mid E) \\
&= \sum_{i=1}^{|C_s|} \log p_{E,\theta_a}(t_i \mid H_{t_i}) + \log p_{E,\theta_a}(m_i \mid H_{t_i}) + \sum_{j=1}^{|U|} \log \sum_{i=1}^{N} w_i \cdot p_{\mu_i,\Sigma_i}\left(E_{u_j}\right).
\end{aligned}$$

To retrieve the parameters, this likelihood must be maximised. However, a trivial solution for the HAGE component would be to simply assign each user to its own cluster with $\mu_i = E_{u_j}$ and $\det(\Sigma) = 0$, which is an infinite asymptotic solution. To solve this problem, Sharma et al. restricted the determinant of the covariance matrix to be greater than a small positive constant ϵ:

$$\det(\Sigma) \geq \epsilon > 0.$$

Further constraints should be added to obtain valid estimations of the parameters for the Gaussian Mixture Model. Not only should the covariance matrix be positive definite, but the mixture

weights must be positive, and their sum must be equal to 1.

Solving this complex constrained optimisation is possible by applying a joint-learning technique. The idea is first to fix the parameter of the Gaussian Mixture model and maximise the likelihood via optimising the AMDN parameters and the user embeddings. These could be done by Adam optimiser. Subsequently, we fix these last two quantities to find the constrained maximum for the GMM via the Expectation Maximisation algorithm. These two steps are repeated:

$$\hat{\theta_b} = \text{argmax}_{\theta_b} \log p_{\hat{E},\theta_b}(U),$$
$$\hat{\theta_a}, \hat{E} = \text{argmax}_{\theta_a,E} \log p_{E,\theta_a}(C_s \mid U) + \log p_{E,\hat{\theta_b}}(U).$$

Proceeding iteratively ensures that this joint learning reaches a saddle point or a local maximum. The reader is referred to Sharma et al. [2021] for the convergence proof. In addition, Sharma et al. implemented early stopping to decrease the computational time. After dividing the cascades into 75% training, 15% validation and 10% test splits, the training procedure is carried on until the validation likelihood of sequences reaches a tolerance level. If this happens *patience* times in a row, the training will be stopped. Otherwise, the training procedure ends after a fixed maximum number of iterations.

Moreover, for model training, various parameters must be set. In particular, the learning rate of the Adam optimiser, the patience level of the training and the number of components of the Gaussian Mixture Model have represented the most tricky choices. This can be solved by fine-tuning these parameters based on some qualitative metrics. For instance, if the dataset ground truth is unknown, then the quality of the model performance could be assessed via the silhouette score. Otherwise, when the true subdivision of accounts is known, we could use metrics that compare the true clustering with the predicted clustering.

3.2.4 AMDN-HAGE clustering and self-attention matrix recovery

Considering any dataset consisting of cascades in the form of Equation 3.7, we have seen how the model could be setup and trained. In particular, after splitting the data into the train, evaluation and test dataset, it is possible to fine-tune the hyperparameters and learn the user embedding and the attention weights. Let us now focus on retrieving the users' clustering and the self-attention matrix.

Sharma et al. has proposed two possible methods for recovering the clusters from the learnt embedding vectors and the parameters of the HAGE component.
The first approach exploits the likelihood function, assigning each user to the mixture component by evaluating the likelihood under each of them. This methodology is referred to as the Classification Expectation Maximisation algorithm. Each user u_j is clustered into the group that reports the highest probability that its embedding vector has been drawn from that specific component of the GMM. Therefore, we need to find the i which reports the highest $p_{\theta_b}\left(z_{i,j} \mid E_{u_j}\right)$. Mathematically, every user u_j is associated with the cluster

$$\hat{i} := \text{argmax}_i \;\; p_{\theta_b}\left(z_{i,j} \mid E_{u_j}\right).$$

We will refer to this clustering as *the model*. The second proposed way for grouping users is aggregating them depending on their embedding but with a different clustering technique. Since coordinated users behaving similarly are expected to have close embedding vectors living in the real vector space $\mathbb{R}^m$, Sharma et al. suggested using the KMEANS method, which is a wide-spread clustering technique that divides a dataset into k groups minimising the inter-group variances.

Regarding the attentive component, it is possible to recover the user-to-user self-attention weights $A_{\text{TOT}} \in \mathbb{R}^{|U|\times|U|}$. In this matrix, each row corresponds to a user who performs an action, and each column represents the influence he receives from other users. The higher the value $A_{\text{TOT}}[i,j]$, the stronger the influence of user u_j previous action on triggering user u_i new action. To evaluate this self-attention matrix, the validation dataset has been divided into batches of length b, and each batch has been employed to compute its attention weights. In particular, considering a batch X, the input vector of every event is evaluated using the learnt parameters θ_b. These vectors are then used to compute the self-attention matrix A_X. Recall that, as seen in Section 3.1, this matrix collects the influence weights that each past event has had on triggering a new event. In particular, the value $A_X[k,h]$ refers to the influence that (m_k, t_k), the k^{th} event in the sequence, has received by (m_h, t_h), the h^{th} event. Therefore, these values should be added to the element in the matrix A_{TOT} associated with the corresponding influenced user (fixing the row) and its influencer (fixing the column). In detail, the value $A_{\text{TOT}}[i,j]$ is updated with every self-attention weight $A_X[k,h]$ related to the influenced user u_i and the influencer user u_j:

$$A'_{\text{TOT}}[i,j] \; = A_{\text{TOT}}[i,j] + \sum_{\{k|m_k=u_i\}} \sum_{\{h|m_h=u_j\}} A_X[k,h]. \tag{3.13}$$

Repeating these computations for every batch in the validation set ensures that the final matrix A_{TOT} properly reports the user-to-user self-attention weights. Finally, it is possible to evaluate the *influence matrix* I, which is the normalisation of the user-to-user self-attention matrix over the number of times each element is updated with a positive weight.

Example 3.1 (Updating the user-to-user self-attention weights)**.** *Let us consider the set of users* $U = \{u_0, u_1, u_2\}$*, the matrix* $A_{TOT} \in \mathbb{R}^{3\times3}$*, and a new batch containing the trace*

$$C_s = [(m_0 := u_0, t_0), (m_1 := u_2, t_1), (m_2 := u_2, t_3)]\,.$$

Firstly, the input X *and the dot product* A_X *associated with this batch are evaluated. Subsequently, the matrix* A_{TOT} *should be updated depending on the new evidence. For instance, by exploiting Equation 3.13, the new diagonal values become*

$$\begin{aligned}
A'_{TOT}[0,0] &= A_{TOT}[0,0] + A_X[0,0],\\
A'_{TOT}[1,1] &= A_{TOT}[1,1] + 0,\\
A'_{TOT}[2,2] &= A_{TOT}[2,2] + A_X[1,1] + A_X[1,2] + A_X[2,1] + A_X[2,2].
\end{aligned}$$

In a similar way, the new off-diagonal values can be obtained with analogous computations. Concisely, the elements of the matrix A_{TOT} *are updated one by one as*

$$A_{TOT}[(u_0,u_2,u_2),(u_0,u_2,u_2)] := \begin{pmatrix} A_{TOT}[0,0] & A_{TOT}[0,2] & A_{TOT}[0,2] \\ A_{TOT}[2,0] & A_{TOT}[2,2] & A_{TOT}[2,2] \\ A_{TOT}[2,0] & A_{TOT}[2,2] & A_{TOT}[2,2] \end{pmatrix} += A_X \in \mathbb{R}^{3\times3},$$

where each column and each row is associated with every mark appearing in the sequence.

3.3 Extensions of AMDN-HAGE

Sharma et al.'s detection method models user interaction based on their behaviour in the arrival times and the marks influence, assuming a latent clustering of the accounts exists. This model was evaluated using different datasets in Sharma et al. [2021], and its result showed great potential. However, we have tested the AMDN-HAGE model in the case studies presented in Section 5.1 and its performance has not achieved the same results. In particular, the model has reported low precision measurements in clustering the coordinated accounts altogether. Therefore, we propose two possible approaches to extend the model and improve its performance.

The first extension of the models we have considered has been driven by trying to soften some requirements of the AMDN-HAGE model. Specifically, we believe setting the number of clusters in advance is a significant limitation that may seriously affect the clustering performance. In fact, this constraint on the Gaussian Mixture Model may cause the merging of different clusters into a unique group, even when the dataset shows evidence of a different number of clusters. Hence, we propose the clustering recovery to be executed with clustering techniques that do not require the number of clusters as input. Specifically, we suggest the usage of Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) and Ordering Points To Identify the Clustering Structure (OPTICS). Recall that the former is a well-spread clustering technique that bases its density-based implementation on finding the cluster with the best stability [Campello et al., 2015, see], and the latter is another clustering technique that builds a density-based augmented ordering of the dataset from which it is possible to retrieve the clustering [Ankerst et al., 1999, see]. Both of these techniques do not require any specification in the number of clusters. In this way, we allow the model to find the number of groups depending on the evidence in the data. We believe that this new cluster recovery allows for the creation of groups reaching a higher level of precision in detecting coordinated accounts.

Regarding the second approach, we have decided to leverage the information contained in the interactions of users in the cascades. In the results obtained in the case studies in Section 5.1, we have noticed that a quick look at the user-to-user self-attention matrix would return a perfect clustering of the users. However, these could not be found in the AMDN-HAGE model. This suggests that the self-attention component has been able to retrieve coordinated and authentic account features, but it is not able to exploit them in the clustering retrieval. Therefore, we have proposed grouping the users based on how each user is influenced by the other accounts. Specifically, we propose learning the membership function of users based on the columns of the user-to-user self-attention weights in place of the user embeddings. This approach will be referred as *self-attention*. Moreover, for the same reasons mentioned above, we test this approach with three additional clustering techniques: *self-attention + KMEANS*, *self-attention + HDBSCAN* and *self-attention + OPTICS*. These models should better exploit the influence weights, which have shown to be capable of perfectly capturing the real subdivision of users. The results of these methods are presented in Chapter 5.

Chapter 4

Simulating coordinated behaviour

Implementing a method for coordinated behaviour detection is a difficult task, as seen in Chapter 1, but even the evaluation of the techniques may be very complicated. As a matter of fact, the limited availability of data, the loss of a shared simulation framework in the literature and the wide number of coordinated strategies represent major drawbacks.

In response to this, we propose the development of a framework that can be used to assess the quality of the detection in a variety of environments. The primary idea is to construct a basic suite that allows researchers to systematically test their detection method, to assess the detection quality in different scenarios and to understand the boundaries of detectability.

In a social media environment, we can identify two different groups of accounts concerning their coordinated activities: authentic users and inauthentic users. The former kind is composed of *normal people*'s accounts, a very heterogeneous group with notable interdependent activities, while the latter class is made of a unique type of coordinated users that follow the same strategy.

In the next sections, we will present how the accounts in these categories can be modelled to construct datasets spanning a wide range of situations in terms of differences in the accounts' behaviours and in their interactions.

4.1 Modelling non-interacting authentic users

As we have seen in Section 2.3.1, the mutually exciting Hawkes process represents a great mathematical concept to model users interacting with each other and to capture the mutual self-excitement within events.

Considering this setup and the intensity function defined in Equation 2.10, the key point is to decide how to fix the parameters to represent the heterogeneity of the accounts. Given that we are dealing with positive parameters, independent Gamma distributions have been chosen to

sample their values:

$$\mu_i \sim \Gamma(mean = \mu, var = \sigma_\mu^2),$$
$$\alpha_{i,j} \sim \Gamma(mean = \alpha, var = \sigma_\alpha^2),$$
$$\beta_i \sim \Gamma(mean = \beta, var = \sigma_\beta^2),$$

where the value of each hyper-parameter must be selected depending on the social media and the action that is going to be represented. For example, a framework for generating new posts interacting with each other may have a smaller baseline average than a framework for comments, which are in general faster to write.

In the choice of these constants, researchers should always keep in mind that the process must not explode, which is ensured if the sample quantities make Equation 2.14 hold. Other than that, it is sufficient to select positive values of the hyper-parameters to generate valid datasets.

4.2 Modelling non-interacting inauthentic users

Inauthentic users could behave in very different ways depending on the coordination strategy they are following, see Chapter 1. Given their flexibility, mutually self-exciting Hawkes processes represent once again a great mathematical tool that can be used to model their activity. We just need to understand how to choose their parameters in order to catch their differences in behaviors with respect to the authentic users. In order to show how this can be done, let us consider a unique scenario - thus a unique type of coordinated activity per time.

First of all, we can assume that, within each scenario, the group of coordinated accounts are homogeneous. Indeed, it could be the case in which behind them there is a unique entity that created them in the exact same way. They could be accounts that follow the exact same commands or, more generally, they are expected to identically interact and get influenced. Due to this, the parameters controlling the baseline, the excitation and the decay are fixed and do not change between different users.

Secondly, their behaviour may differ from authentic users depending on the intensity, on the entity of their triggering events and on their reaction to authentic and inauthentic activities. To control these features, we have introduced the parameters c_μ, c_α and c_β that respectively control the scale of the intensity, the excitation and the decay with respect to the mean of the authentic hyper-parameters. In other words, these parameters control how much the coordinated accounts distinguish themselves from an average authentic user: the farther from 1 the more distinguishable interactions.

Finally, coordinated accounts may be active only in specific time of the day, after the occurrence of a specific event or following the order of a latent user who organise their activity. The variable Z has been introduced to model this feature: if the value is equal to 0 then all the inauthentic accounts remain silent, but as soon as its value is changed, the inauthentic users start interacting and influencing each other.

For example, setting $Z(t) := 1$ will cause each user to be always active, while $Z(t) := \mathbb{1}_{sin(\omega t) > c_{threshold}}$ would introduce a period activation and deactivation depending on the parameters ω and $c_{threshold}$.

Therefore, we have modelled coordinated accounts as mutually self-exciting Hawkes process with parameters:

$$\begin{aligned} \mu_i &= c_\mu \mu Z, \\ \alpha_{i,j} &= c_\alpha \alpha Z, \\ \beta_i &= c_\beta \beta Z. \end{aligned}$$

4.3 Modelling interacting users

The two processes outlined above assume that authentic and inauthentic users do not interact with one another, which is a substantial simplification of real-world dynamics. These models are suitable for scenarios where inauthentic users operate in isolation from the broader social network. However, when inauthentic users attempt to disguise their deceptive behaviour, cascades are more likely to be mixed. For example, sockpuppets aim to present themselves as genuine disguising their inauthentic activities by interacting with authentic users to some degree, thus the cascades in which they participate in are likely mixed interactions. To simulate this behaviour, a third mutually self-exciting Hawkes process is introduced. This process takes into account these interactions in modelling authentic and inauthentic users behaviour.

The intensity function's baseline and decay rate have been fixed as in the aforementioned cases according to the user type and independently from the users' authenticity it is interacting with. Specifically, the baseline vector will be the concatenation of the authentic and inauthentic users' baseline vectors:

$$(\mu_i)_i = \begin{pmatrix} \mu_i \\ \vdots \\ \mu_i \\ \mu \cdot c_\mu Z \\ \vdots \\ \mu \cdot c_\mu Z \end{pmatrix}. \tag{4.1}$$

Analogously, the vector controlling the decay rate for every user will naturally be:

$$(\beta_{i,.})_i = \begin{pmatrix} \beta_i \\ \vdots \\ \beta_i \\ \beta \cdot c_\beta Z \\ \vdots \\ \beta \cdot c_\beta Z \end{pmatrix}. \tag{4.2}$$

Concerning the extension of the excitation rates, it is necessary to set how users of different types interact with one another. The matrix could be divided into four blocks depending on

which type of users is getting influenced by which type of users' actions. The first (upper-left) and the fourth (lower-right) blocks could be kept as set above in respectively the non-interacting authentic simulations and inauthentic simulations.

Regarding the lower-left block, coordinated users who are attacking authentic users may be highly influenced by their target's actions. They could be asked to repeat right away someone's post or to push a specific hashtag that appeared in their target's post. Therefore, to control the amount of this influence with respect to an inauthentic-to-inauthentic interaction, a new parameter I is introduced.

Finally, in the upper-right block, inauthentic influence on authentic accounts may vary significantly depending on the simulation we want to run. Assuming that authentic users can detect *inauthenticity* to some extent or anyway that authentic-to-inauthentic coordination activity is very low with respect to the other types of interactions, the second block could be set to 0 everywhere. Opting for this scenario, the complete excitation matrix for modelling interacting users would be

$$(\alpha_{i,j})_{i,j} = \left(\begin{array}{cc|ccc|ccc} & & \multicolumn{3}{c|}{\text{authentic}} & \multicolumn{3}{c}{\text{inauthentic}} \\ \multirow{3}{*}{\rotatebox{90}{authentic}} & & \alpha_{i,j} & \cdots & \alpha_{i,j} & 0 & \cdots & 0 \\ & & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ & & \alpha_{i,j} & \cdots & \alpha_{i,j} & 0 & \cdots & 0 \\ \hline \multirow{3}{*}{\rotatebox{90}{inauthentic}} & & c_\alpha\alpha IZ & \cdots & c_\alpha\alpha IZ & c_\alpha\alpha Z & \cdots & c_\alpha\alpha Z \\ & & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ & & c_\alpha\alpha IZ & \cdots & c_\alpha\alpha IZ & c_\alpha\alpha Z & \cdots & c_\alpha\alpha Z \end{array}\right). \tag{4.3}$$

Nevertheless, it could also be assumed that authentic accounts are not able to distinguish the origin of the content they have been exposed to at all, so there would be no reason to modify α value along one row and a generic user u_i would be triggered by an action of u_j independently if the latter account is authentic or inauthentic. Therefore, the excitation matrix would become

$$(\alpha_{i,j})_{i,j} = \left(\begin{array}{cc|ccc|ccc} & & \multicolumn{3}{c|}{\text{authentic}} & \multicolumn{3}{c}{\text{inauthentic}} \\ \multirow{3}{*}{\rotatebox{90}{authentic}} & & \alpha_{i,j} & \cdots & \alpha_{i,j} & \alpha & \cdots & \alpha \\ & & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ & & \alpha_{i,j} & \cdots & \alpha_{i,j} & \alpha & \cdots & \alpha \\ \hline \multirow{3}{*}{\rotatebox{90}{inauthentic}} & & c_\alpha\alpha IZ & \cdots & c_\alpha\alpha IZ & c_\alpha\alpha Z & \cdots & c_\alpha\alpha Z \\ & & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ & & c_\alpha\alpha IZ & \cdots & c_\alpha\alpha IZ & c_\alpha\alpha Z & \cdots & c_\alpha\alpha Z \end{array}\right). \tag{4.4}$$

4.4 Simulating a dataset

In the above sections, we have presented how it is possible to model each type of interaction between two possible types of users in a social network. Let us now focus on how to put everything together to properly generate a dataset.

Fixing n_{aut} authentic users, n_{ina} inauthentic users, $n_{cascades}$ the number of cascades to be generated and the Hawkes process parameters presented above, we only need to choose how the social interactions should be simulated. Let's introduce the parameter p to control the percentage of mixed interactions, and the parameter q to set the percentage of cascade with only inauthentic users over the cascade with all the non-mixed interactions. Finally, the data can be simulated via Ogata's modified thinning algorithm for marked temporal point processes, which is described in Section 2.4.4.

Algorithm 4.1 (Simulation of social media activities)**.** Fix the parameters regarding the set up (n_{aut}, n_{ina}, $n_{cascades}$, p, q) and the parameters regarding the Hawkes processes (c_μ, c_α, c_β, I, Z) and sample authentic users parameters checking they do not cause explosion.
Then start the actual simulations generating each cascade via Algorithm 2.5 with the following setup:

- For the first p percentage of cascades, set the parameters as defined in Section 4.3.
- For the remaining cascades, take the first q percentage and set the parameters as defined in Section 4.2.
- For the rest, set the parameters as defined in Section 4.1.

4.5 Generating the framework

Up to now, we have learnt how to model users' behaviour and how to generate a dataset from a specific environment. Thus, we have all the necessary elements to design a framework that systematically covers a variety of different scenarios allowing the exploration and evaluation of coordination detection methods.

The key idea behind the framework is to create datasets varying only specific simulation parameters while keeping others constant. This approach allows for a comprehensive creation of different scenarios covering all the effects caused by the variation of each parameter. Hence, this structured methodology ensures the creation of datasets where coordinated users become more and more similar to authentic users for their intensity baseline, decay time or excitation rate.

In our proposed framework, the simulations were conducted considering only non-interacting cascades, thus with p set to 0, with the parameters c_μ, c_α and c_β generally fixed at a value of 1. However, in each scenario, one of these parameters was varied while the others remained constant. Specifically:

- c_μ was varied across the values $\{0.1, 1, 10\}$;
- c_α was varied across the values $\{0.5, 1, 2, 4\}$;
- c_β was varied across the values $\{1, 10, 100\}$.

In addition to these scenarios, other simulations were conducted with the parameters $c_\mu = 0.1$, $c_\alpha = 4$, and $c_\beta = 10$ fixed. The reason behind this setup is to try to emulate the expected

behaviour of inauthentic users: it is reasonable to assume that coordinated users have a higher degree of excitation within one another, that their background activity is low unless they are attacking and that they forget other users activities in a faster than authentic users. In these cases, the parameter p was varied across the values $\{0, 0.3, 0.7, 1\}$ to enable a detailed analysis of how a detection method performs when the cascades are mixed between the two different types of users.

Summing up, Table 4.1 reports the choices of parameters we have proposed to generate a framework allowing for an in-depth exploration of multiple scenarios.

Simulation name	c_μ	c_α	c_β	p
$c_\mu = 0.1$	0.1	1	1	0
$c_\mu = 1$	1	1	1	0
$c_\mu = 10$	10	1	1	0
$c_\alpha = 0.5$	1	0.5	1	0
$c_\alpha = 2$	1	2	1	0
$c_\alpha = 4$	1	4	1	0
$c_\beta = 10$	1	1	10	0
$c_\beta = 100$	1	1	100	0
$p = 0$	0.1	4	100	0
$p = 0.3$	0.1	4	100	0.3
$p = 0.7$	0.1	4	100	0.7
$p = 1$	0.1	4	100	1

Table 4.1: Variation of parameters in the proposed framework.

While the current framework provides a robust method for exploring the influence of varying individual parameters, several extensions could further enhance the evaluation of detection models. Thanks to the flexibility of the proposed framework, we believe that it would not be complicated to design scenarios resembling the activities of other possible types of coordinated campaigns. For example, inauthentic users' attacks targeting only specific authentic users could be introduced simply by manually selecting the group of authentic users being part of the mixed cascades. These extensions would broaden the applicability of the framework, allowing it to address a wider range of problems.

Chapter 5

Case Studies

In the previous chapters of this thesis, we have presented some coordination detection methods and a way to generate a framework for their evaluation. Now, it's time to merge everything and understand whether and to what extent coordinated campaigns can be detected. Specifically, we will test two coordination detection methods: Sharma et al.'s AMDN-HAGE model (referred to as "the model") and our extended version (referred to as "self-attention"). Moreover, we will compare the results with Pacheco et al. [2021], Keller et al. [2020], Ng and Carley [2023], Schoch et al. [2022] and Weber and Neumann [2020], state-of-the-art detection methods based on sliding windows.

These analyses will be executed on datasets coming from two different backgrounds. At first, we will quickly investigate the detection methods' performances over a hybrid dataset, coming from real Twitter data for authentic activities plus synthetic data simulating coordinated behaviour campaigns. Subsequentely, we will direct our attention to the main case study: how will the methods perform in the scenarios created via the synthetic framework we introduced in Chapter 4. These explorations will allow an in-depth evaluation of the methods' performances as inauthentic users become more and more similar to authentic accounts.

For every scenario, we will run the code for each model, fine-tune the parameters and assess the quality of their finding via different metrics to capture different features of the resulting clusters. In particular, we will report results' quality by comparing the clusters found by the detection model against the true division of users into authentic or inauthentic labels. To this aim, we will focus on two widely used methods for evaluating the distance between two clustering:

- Hausdorff distance measure, which is based on one of the most used distances between sets, it creates robust and comprehensive evaluation of the distance between two sets of points and it captures the maximum extent of separation between them [Chacón, 2015, see];

- maxF1 score, which centres its attention on the precision and the recall of the cluster best matching the ground truth category (inauthentic accounts in this case). Practically, it is evaluated by taking the maximum F1 score over the possible clusters.

Additionally, further results obtained with other metrics will be reported in Appendix.

5.1 Hybrid datasets

As mentioned above, the first assessment of the performance of the models has been realised via a hybrid dataset. In this section, we will describe this case study and report its results.

5.1.1 Case study initial datasets

First of all, let us introduce the dataset. Using Twitter scraping functions, the dataset was collected during the 2023 Finnish parliamentary election via a list of Finnish keywords related to parliamentary elections, candidates, parties, and political topics. Examples of such keywords include terms like "vaalit" (elections) and "eduskunta" (parliament). From the 1^{st} September 2022 to the 13^{th} April 2023, $5.2M$ tweets were retrieved saving the author id, the content, the timestamp and whether it was a retweet or not. Figure 5.1, shows how tweets were spread along the months before the elections.

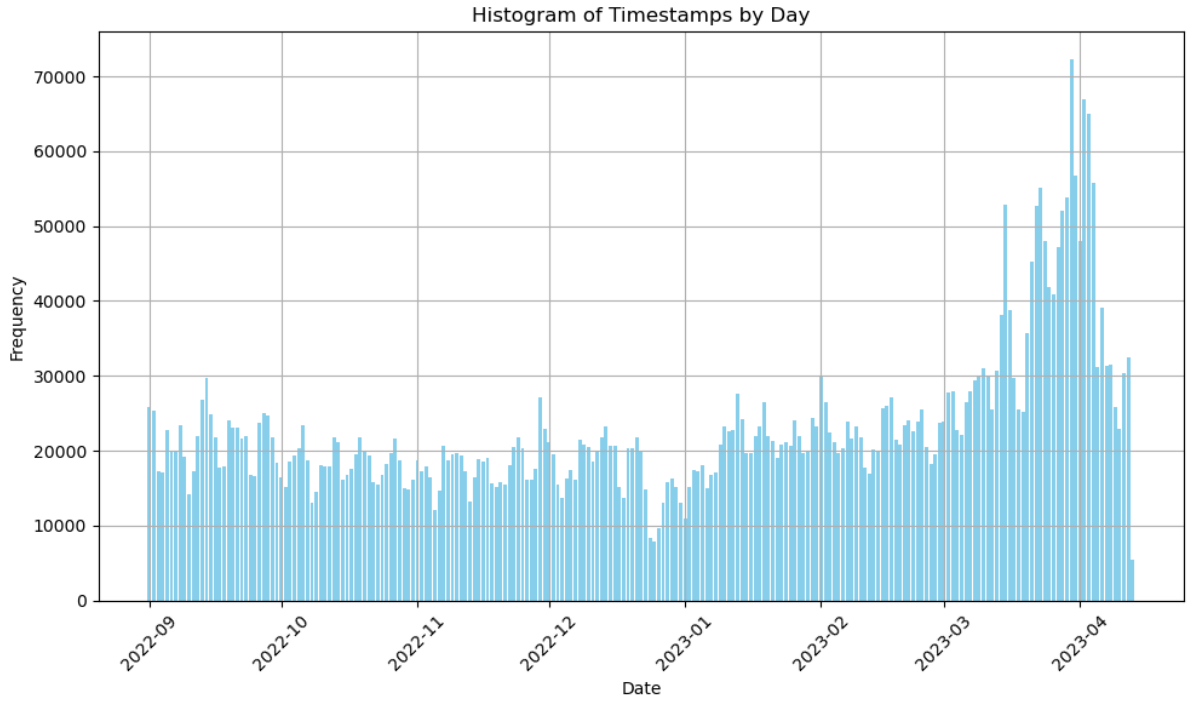

Figure 5.1: Histograms of time stamp by day.

In total, the number of active accounts were $159,085$, of which only 762 created more than 1000 tweets. Due to the limited computational capability and time, we have focused on the latter group, see Figure 5.2. Moreover, since the lack of ground truth would have not allowed a quantitative assessment of the results, we have opted to consider all of these real accounts as authentic and to manually generate 100 coordinated users. To do so, we have proceed in two ways.

In the first case the coordinated campaign concerns only inauthentic users, they write post and retweets only inauthentic users content following a uniform distribution and creating a sample of

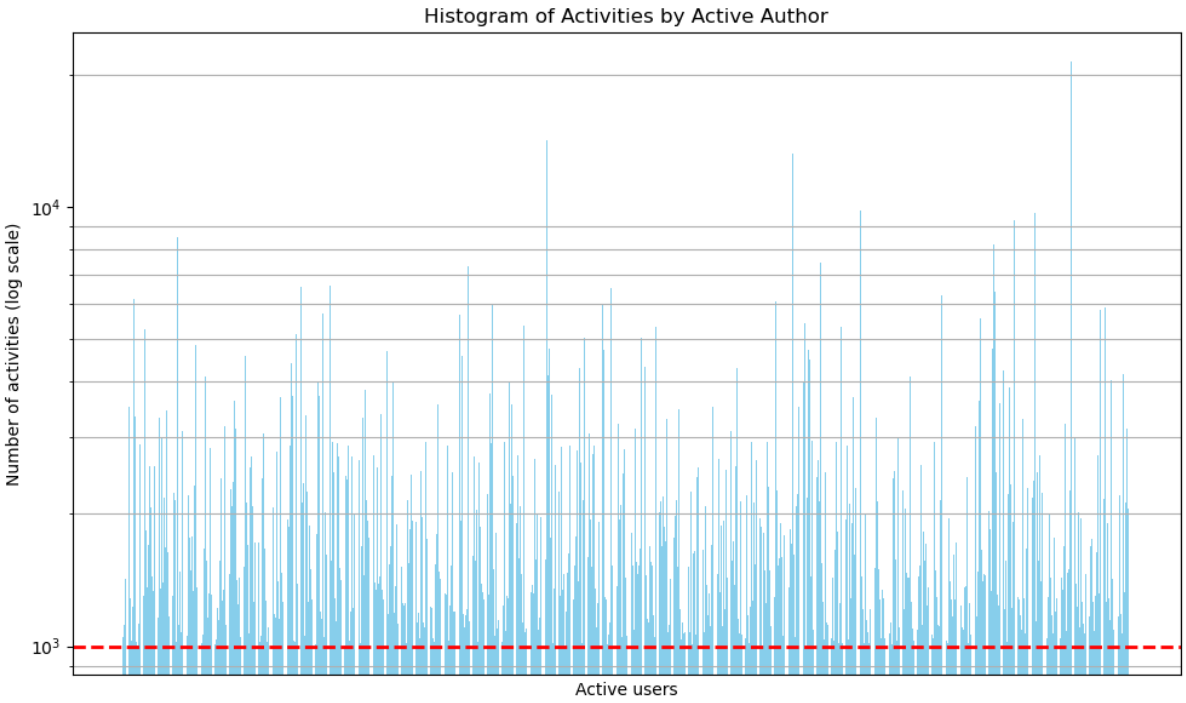

Figure 5.2: Histograms of activities by active author with at least 1000 tweets or retweets, where the dashed red line represents this threshold.

260,000 activities, corresponding to about 5% of the real dataset activities. This percentage has been chosen in line with Guess and Lyons [2020], which estimated that 5% of political content originated from disinformation sources. We expect these inauthentic accounts to be identified quite easily since they do not interact with authentic users, thus their behaviour should clearly stand out. In Figure 5.3, the combined activities have been shown. Specifically, the authentic dataset has been divided depending on the day posts were published and its bars are coloured in blue, while the activities of coordinated campaigns are coloured in red.

Moreover, we decided to cover an attack of a greater extent where inauthentic accounts retweet authentic users' post during all days. This second dataset resemble the first dataset in terms of number of activities, but this time there is an intense interaction between the two types of accounts. We believe that this feature will make detection more difficult with respect to the first scenario.

5.1.2 Case study preprocessing and model implementation

From the scrapped activities and the coordinated attacks, it is necessary to construct each retweet cascade. For every new post, we have constructed a list containing the author index and the timestamp and each time that specific text is reposted the retweet author and the timestamp has been appended to the list. Finally, we have deleted all the cascades shorter than 2, which are not relevant for the understanding of each user's behaviour. The total number of the remaining cascades is of the order of a hundred thousand, way smaller than the initial datasets, and the longest cascade reports around ten thousand retweets for both the two datasets.

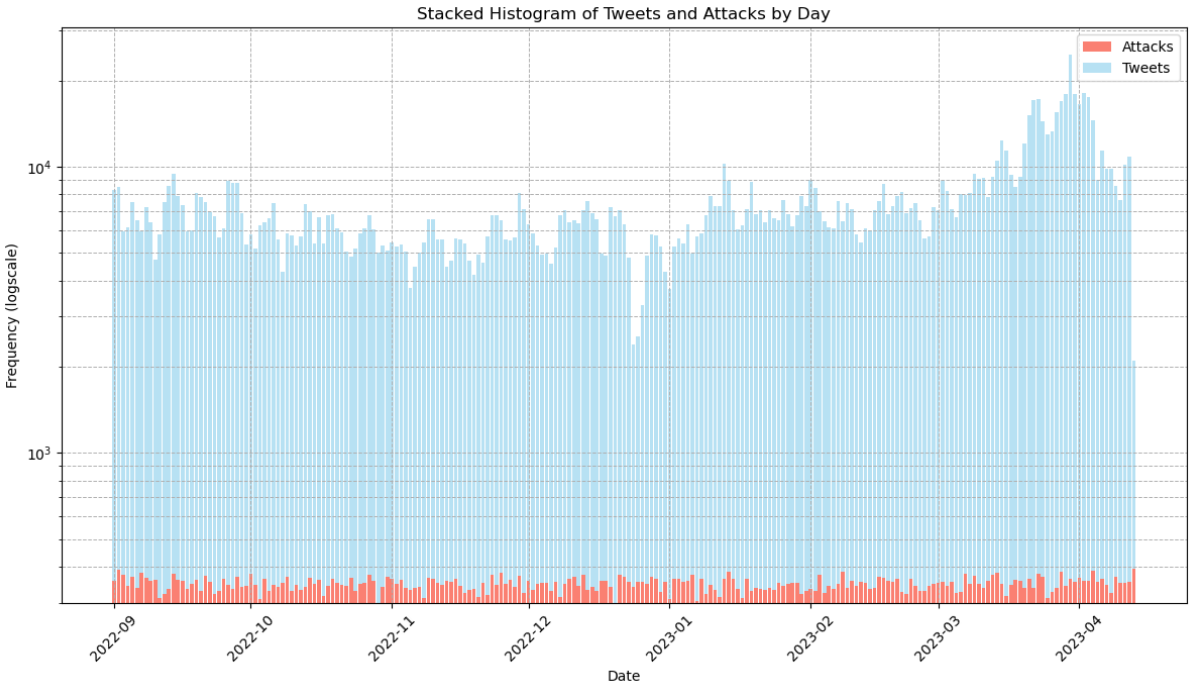

Figure 5.3: Stacked histograms of timestamps by Day with non-interacting coordinated users.

After setting the number of clusters to 3 and fine-tuning the parameters of the learning rate and the patience by qualitatively looking at the loss function plots, everything is ready for the models' evaluation.

5.1.3 Case study results

The clustering obtained by running the model over the interacting dataset is reported in Figure 5.4. Each heatmap displays the confusion matrix corresponding to the clustering created by the detection model under consideration. On the y-axis, the ground truth labels are presented, with 1 indicating inauthentic users and 0 representing authentic accounts. Meanwhile, the x-axis shows the identified clusters, which vary according to the number of clusters the model uncovers. The colour intensity of each cell is scaled in proportion to its value, allowing for an immediate visual assessment of the clustering quality. Hence, the image shows how authentic (inauthenticity = 0) and inauthentic users (inauthenticity = 1) are grouped via the detection method simple version or by the detection model plus KMEANS, HDBSCAN or OPTICS techniques. Besides the actual clusters, the group of accounts labelled "cluster = 0" has been introduced to collect the accounts whose clustering techniques could not classify them into the existing groups, a problem that HBSCAN and OPTICS may encounter. Nevertheless, in this scenario, a perfect detection method would return 3 clusters, each made only of a unique kind of user. In particular, the best results for our coordination detection aim would gather the 100 inauthentic accounts in a unique class (the method's recall), and this class would contain only those accounts (the method's precision). The level of recall in the clustering obtained via the model is very close to 1, indicating that the model effectively identifies most inauthentic users. However, the precision

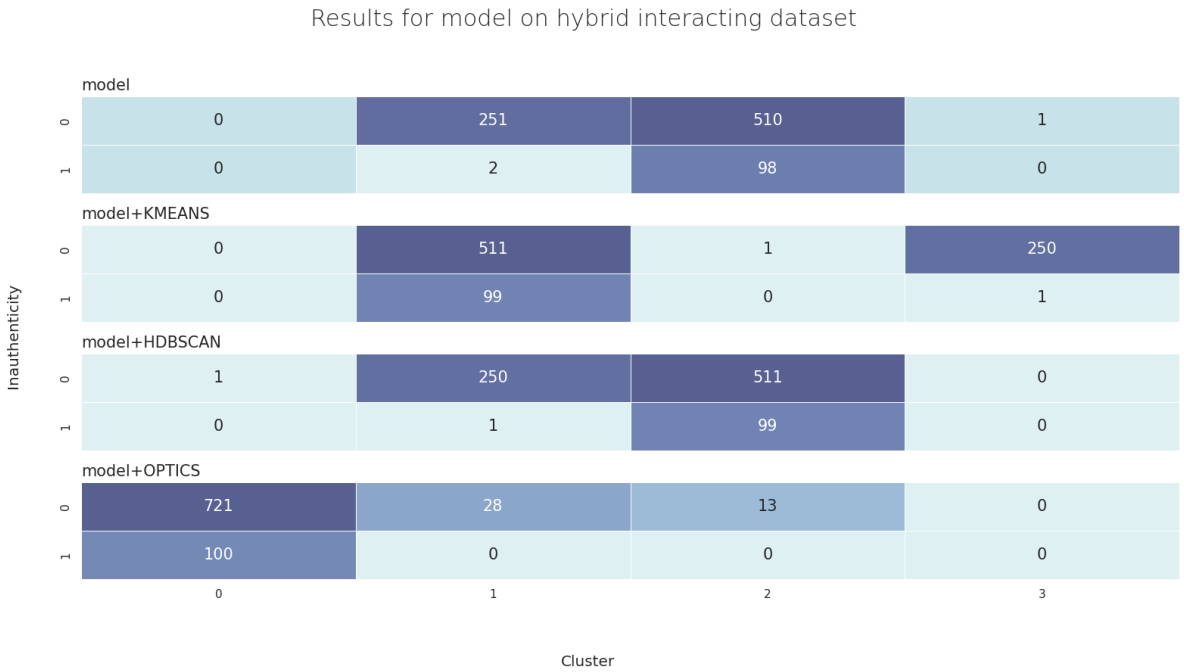

Figure 5.4: Results for model on the interacting dataset.

level is significantly low, remaining below 0.17 across all clustering techniques, suggesting a high number of false positives in the detection process.

To investigate the reasons behind this low precision level, we have analysed the learnt user-to-user self-attention weights, as these are expected to encode the influence accounts have on one another. Suppose a distinct pattern that separates the two types of users emerges. In that case, the AMDN component of the model succeeds in encoding the history information, but the clustering component seems to fail in fully exploiting it, resulting in its inability to identify clusters with high precision. Upon examining the self-attention weights learnt by the model, see Figure 5.5, we realised that this is the case. In fact, the matrix representing these self-attention weights forms a block matrix, perfectly dividing authentic and inauthentic users. However, the clustering process does not leverage this clear separation, leading to low precision in inauthentic cluster identification. To improve the detection quality and leverage the information embedded in the user interactions, we have proposed the self-attention model, a modified version of AMDN-HAGE (see Chapter 3). In testing it with the hybrid dataset, we have realised that the results vary depending on the clustering techniques implemented. As Figure 5.6 shows, the best performance comes from KMEANS, which has been able to cluster the 100 inauthentic users within a group of 359 accounts, thus improving the precision to about 0.28.

Switching to the non-interacting dataset, we would expect the performance of the detection methods to improve. However, this is not consistently observed. The clustering results of the model, as shown in Figure 5.7, indicate that inauthentic users are not entirely grouped together. This may be due to the model identifying the formation of multiple sub-clusters for inauthentic activities, as these are processed in isolation without the interaction with authentic activities.

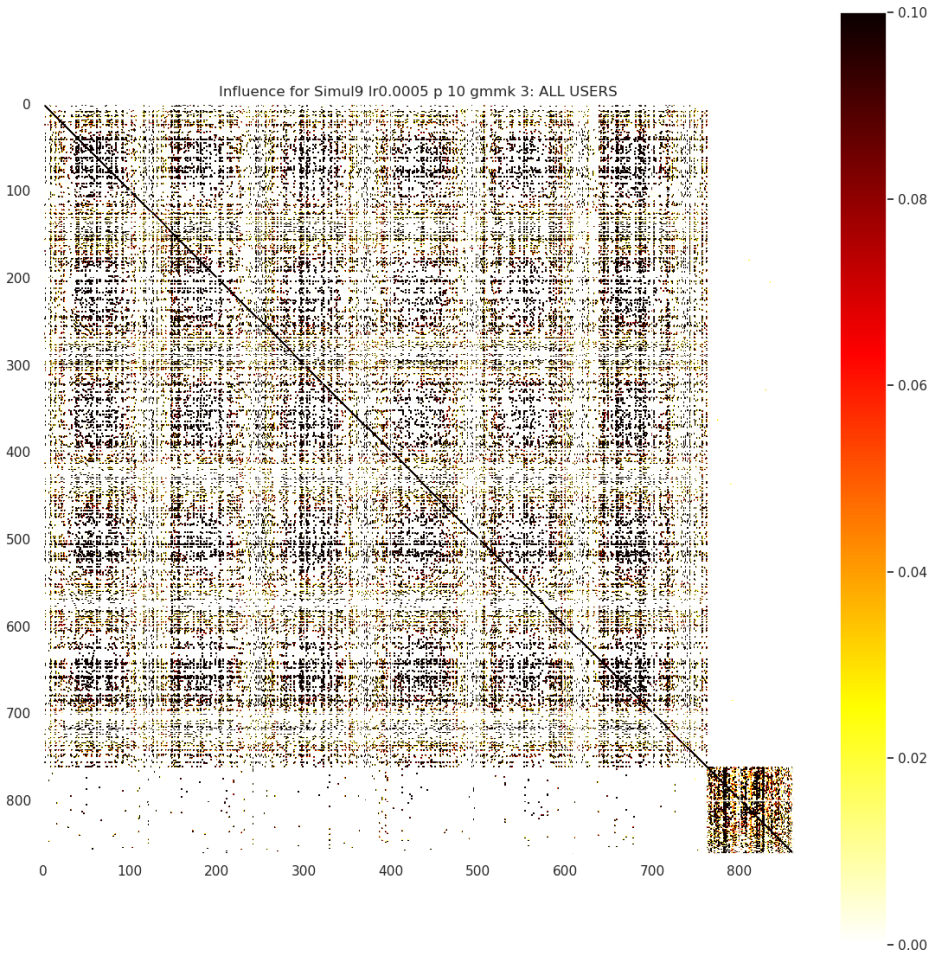

Figure 5.5: Self-attention matrix resulting from the interacting dataset.

Despite this, the precision level shows an improvement with respect to the result obtained in the interacting case. Considering the self-attention methods instead, Figure 5.8 reports an almost perfect results for the version with the HDBSCAN. In this scenario, all 99 coordinated users are grouped into a single distinct class, reflecting highly accurate clustering. In fact, the level of recall is nearly maximal, while the precision reaches the optimal value of 1, indicating perfect precision.

5.2 Synthetic datasets

In this second case study we have opted to test the detection models on the synthetic framework presented in Chapter 4. In this section, we will report how the parameters for the simulations

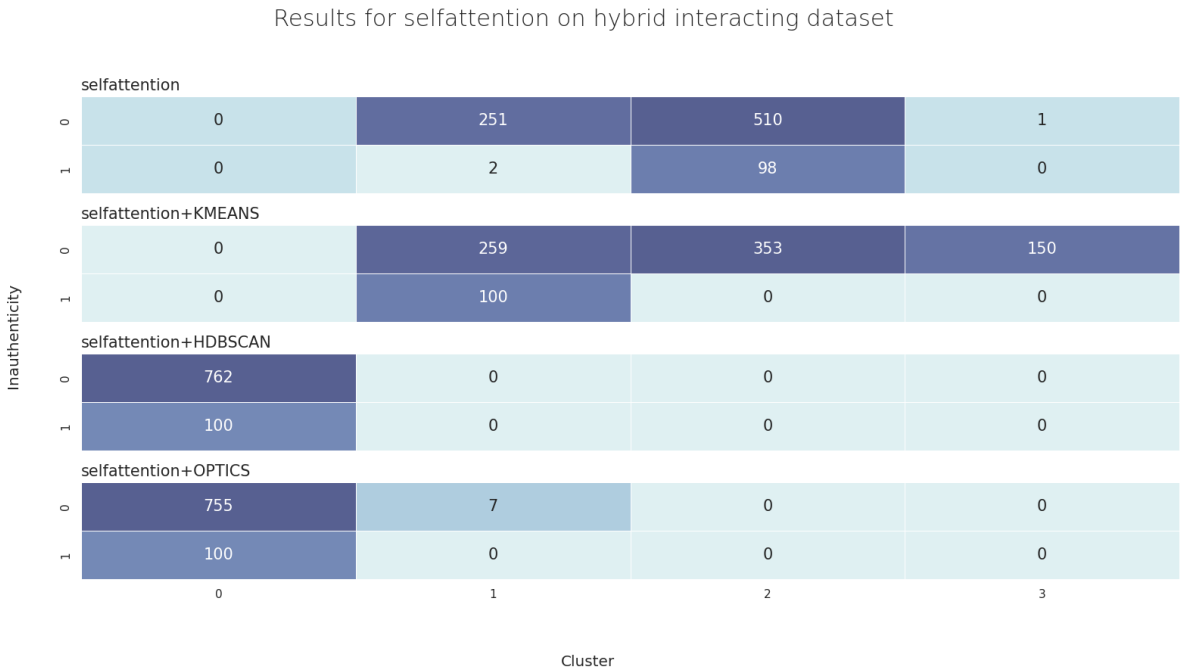

Figure 5.6: Results for self-attention on the interacting dataset.

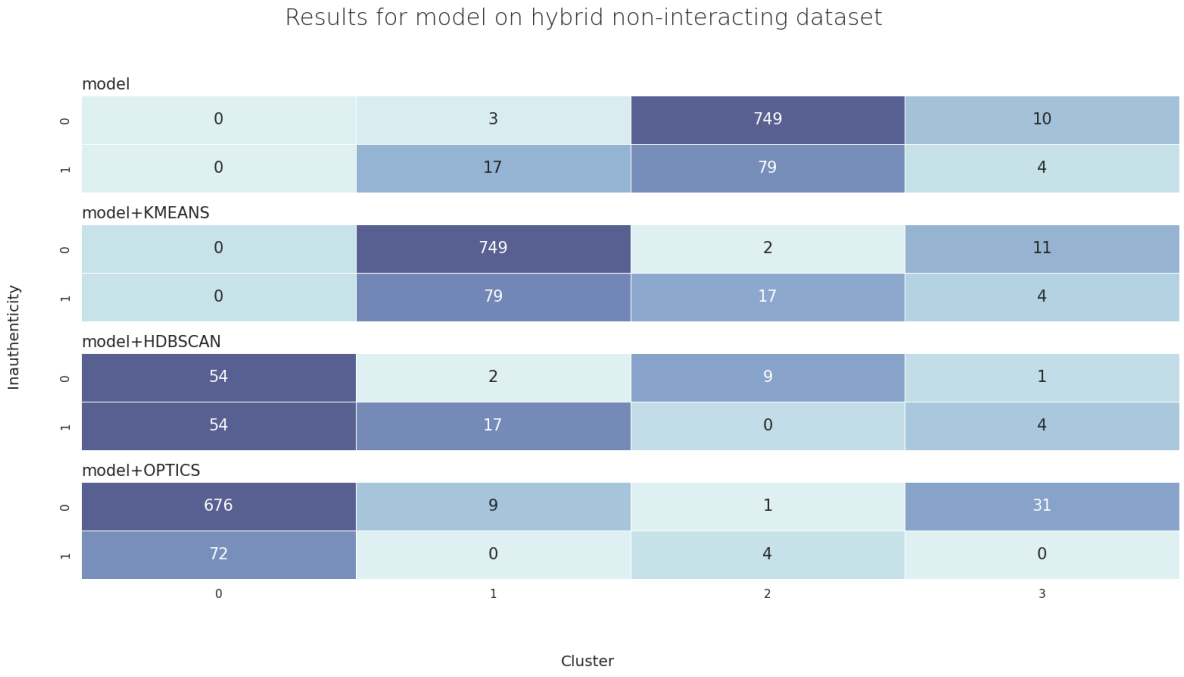

Figure 5.7: Results for model on the non-interacting dataset.

have been chosen, how the methods have been fine-tuned and how they perform in the different scenarios.

5.2.1 Case study initial datasets

A critical aspect of data generation involves determining the parameters that govern the scope of the simulations and devising a method to model authentic user behaviour consistently with

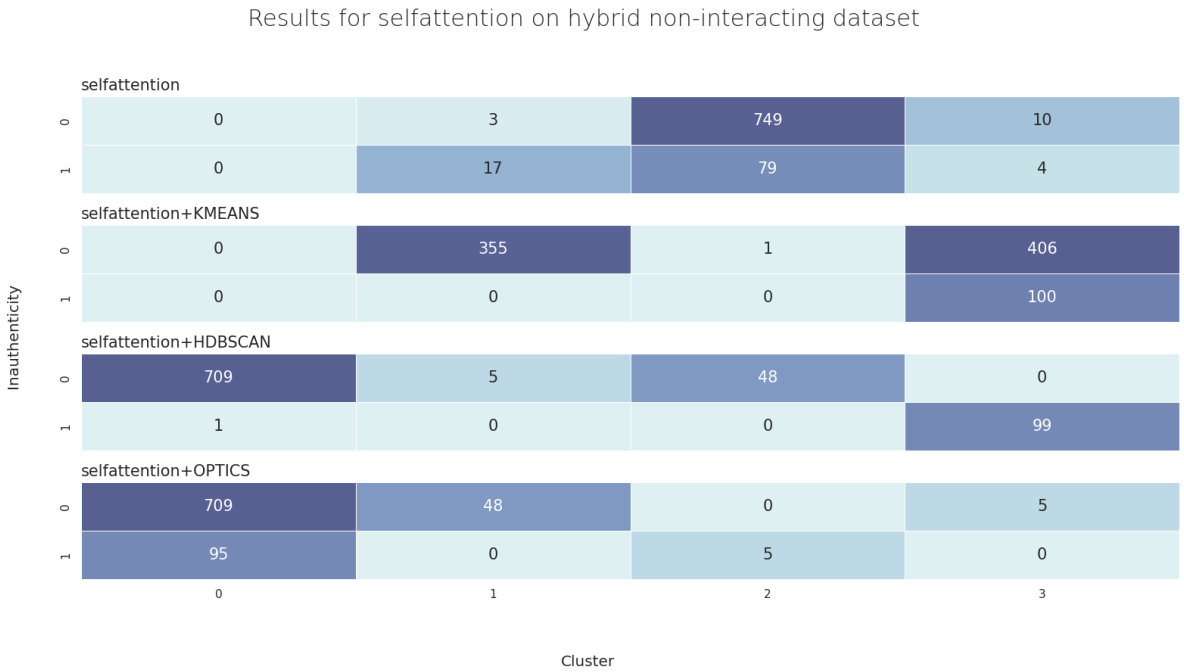

Figure 5.8: Results for self-attention on the non-interacting dataset.

real-world data.

In order to keep each dataset small enough to have manageable computational time and simultaneously big enough to develop the users' behaviour, we have decided to set the parameters as follows:

- $n_{aut} = 100$;
- $n_{ina} = 10$;
- $n_{cascades} = 1000$.

Moreover, the percentage of inauthentic behaviour has been fixed to 5%, resembling the expected extension of disinformation spreaders as mentioned in Guess and Lyons [2020] and Twitter [2014], and the length of each cascade has been set to 128, the maximum dimension manageable for the model. Indeed, longer sequences would be split into smaller cascades.

Concerning the parameters for the Hawkes processes, we have opted for setting I and Z to 1, the value making the simulation the simplest possible, while c_μ, c_α, c_β and p have been varied as described in Section 4.5.

Conversely, choosing proper hyperparameters for the authentic cascade, see Section 4.1, turns out to be a non-trivial task. To maintain the highest level of realism, we have decided to estimate these values via Stan [Stan Development Team, 2024] following a Bayesian hierarchical structure. For the priors selections, we have opted for Gamma distributions due to their support, which coincides with the support of Hawkes parameters, and for their flexibility. Specifically, we have

modelled the shapes ($h_a^{\square}$) and the rates ($h_b^{\square}$) distributions as follows:

$$\begin{aligned} h_a^{\mu} &\sim \Gamma(0.5, 1), \\ h_a^{\alpha} &\sim \Gamma(0.5, 1), \\ h_a^{\beta} &\sim \Gamma(0.5, 1), \\ h_b^{\mu} &\sim \Gamma(1, 1), \\ h_b^{\alpha} &\sim \Gamma(1, 1), \\ h_b^{\beta} &\sim \Gamma(1, 1). \end{aligned}$$

These are hierarchically connected with the actual Hawkes process parameters as

$$\begin{aligned} \mu_i &\sim \Gamma(h_a^{\mu}, h_b^{\mu}), \\ \alpha_{i,j} &\sim \Gamma(h_a^{\alpha}, h_b^{\alpha}), \\ \beta_i &\sim \Gamma(h_a^{\beta}, h_b^{\beta}). \end{aligned}$$

Furthermore, the likelihood of the Hawkes process has been simplified to make the computation of the process more tractable, especially in our scenarios where the data involves large event sequences. Specifically, for every event (m_n, t_n) in the cascade, the intensity function is evaluated at the time t_n as in Equation 2.10 and it is assumed to be constant in the previous inter-arrival time:

$$\lambda(t \mid H_{t_n}) \simeq \lambda_n := \lambda\left(t_n \mid H_{t_n}\right) \quad for \;\; t \quad in \quad (t_{n-1}, t_n]\,. \tag{5.1}$$

Therefore, the evaluation of the log-likelihood for the cascade D can be approximated as

$$\log\left(D\right) \simeq \sum_{n=2}^{len(D)} \log(\lambda_n) - \lambda_n \cdot (t_n - t_{n-1})\,. \tag{5.2}$$

Considering the longest cascade in the Finnish dataset of the previous section, we have used Stan to simulate 4 Markov chains for 1000 iteration with the level of warmup set to 500. Finally, for robustness, the estimations of the hyperparameters have been carried out via the median of the obtained chains. For the summary of the simulation, the reader is referred to the Appendix 5.2.4.

In conclusion, once the hyper-parameter values have been set, the datasets can be created following Algorithm 4.1. Figure 5.9 shows an example of the resulting activities for the datasets created varying p.

5.2.2 Case study implementation

The datasets obtained in the previous section can be directly used to train the detection models we have been studying in Chapter 3.

The last problem that needs to be addressed is choosing the learning rate and the patience properly. To do so, we have selected some possible values of the two parameters: $\{1, 0.8, 0.5, 0.3, 0.1\} \times 10^{-3}$ for the learning rate and $\{10, 20, 50\}$ for the patience. For each possible couple of values, we

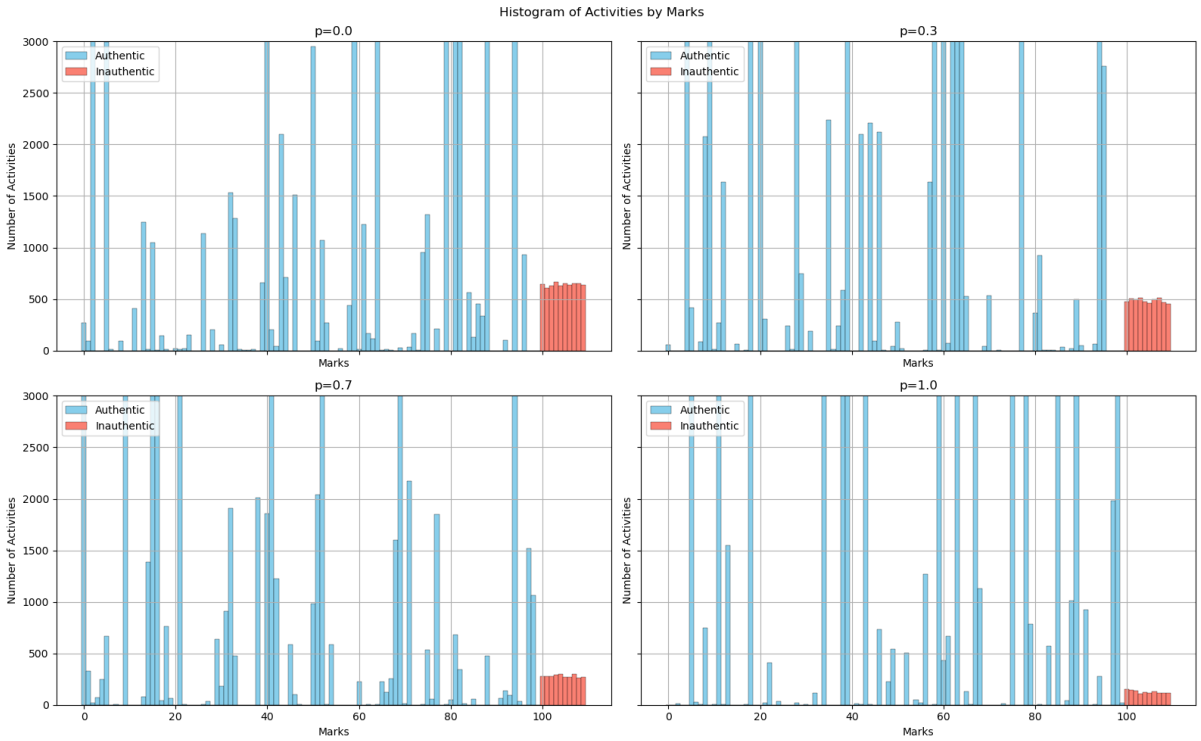

Figure 5.9: Histograms of activities for the datasets created varying p.

have run the detection model and evaluated its performance via the Hausdorff distance. Finally, the learning rate and patience returning the best clustering have been chosen and their results saved.

5.2.3 Case study results

For every dataset in the framework and every detection model, we generated 8 user clustering: one for each technique (GMM, KMEANS, HDBSCAN or OPTICS) and per method (the model-based or the self-attention-based). In this section, we will analyse the results for every scenario, evaluating both model versions.

Starting from the case where $p = 0.0$, representing the scenario in which authentic users and coordinated users are the most distinct, Figure 5.10 and 5.11 display the results obtained via the model-based and the self-attention-based methods respectively. In the former scenario, HDBSCAN and OPTICS appear to be the best-performing methods. They successfully divide the user into five groups, with only one cluster containing mixed types of accounts and one cluster consisting solely of inauthentic users, resulting in high precision. However, we would have expected inauthentic users to be more aggregated due to their identical Hawkes process parameters. In contrast, the self-attention-based detection method performed better regarding recall, though it resulted in more authentic accounts being clustered with inauthentic users.

Hence, in the scenario where authentic and inauthentic users do not interact, there is no clear prevalence between Sharma et al.-based and self-attention-based models. Thus, the choice of

the best detection methods depends on the specific metric of interest and the similarity level between authentic and inauthentic users. However, in the scenario where coordinated users do interact with authentic users, the situation changes. Here, the detection models based on the self-attention matrix are more effective in grouping inauthentic accounts while maintaining a small number of users within the cluster, as shown in Figure 5.12.

Nonetheless, to facilitate a more efficient and systematic comparison of the outcomes as the values of c_α, c_β, c_μ and p vary, we have aggregated the results as illustrated in Figure 5.13, 5.14, 5.15 and 5.16 respectively. Every image has a subplot for each measure and model type, with four lines representing the score for the different clustering techniques. Additionally, a black dotted line has been added to highlight the parameter's value in which authentic and inauthentic accounts differ significantly in their behaviour. We expect the worst results to occur along this line.

Let us now proceed to analyse the first case. Focusing on the detection methods based on Sharma et al.'s model (depicted in the left subplots), the smallest Hausdorff distance is achieved when c_α has the highest value between the four considered constants. This outcome aligns with our expectations, indicating that this scenario represents the most straightforward case for detecting coordinated behaviour. As the values of the parameters approach 1, the distance from the true clustering increases. No single clustering technique consistently prevails over the others across all three scenarios. Among the techniques based on Sharma et al.'s model, KMEANS achieves the best performance regarding Hausdorff distance. However, when considering the maxF1 metric, all the techniques yield similar scores, with HDBSCAN slightly outperforming the others. Fur-

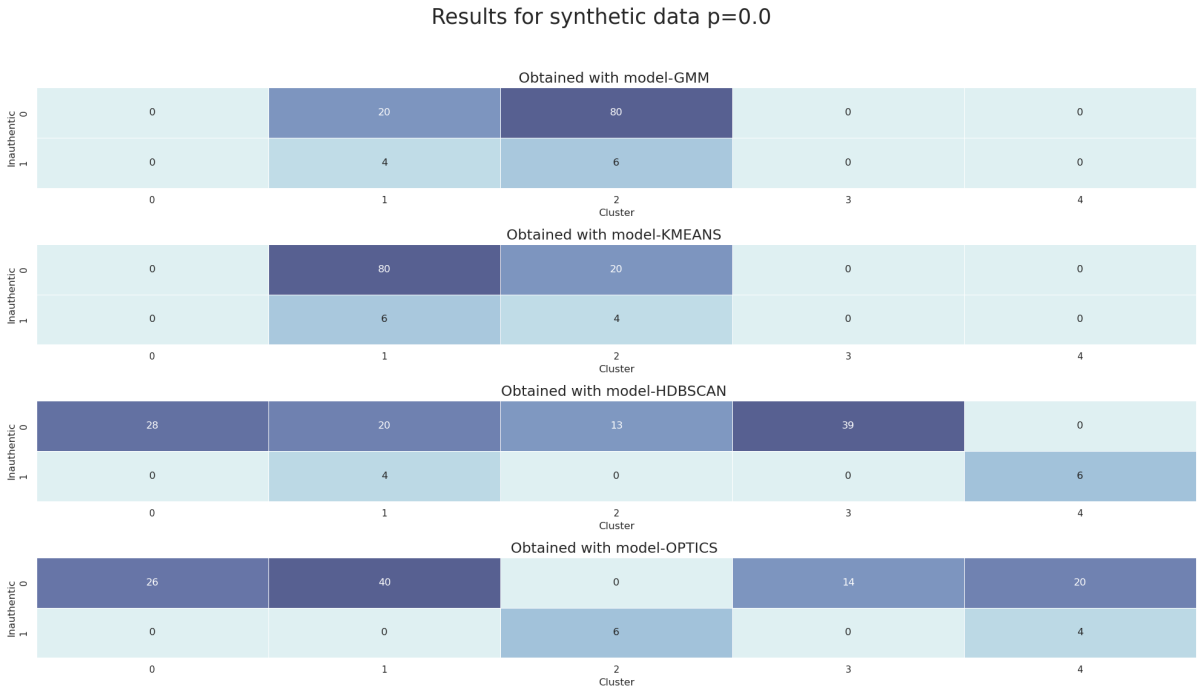

Figure 5.10: Results based on Sharma et al.'s model for $p = 0$ dataset.

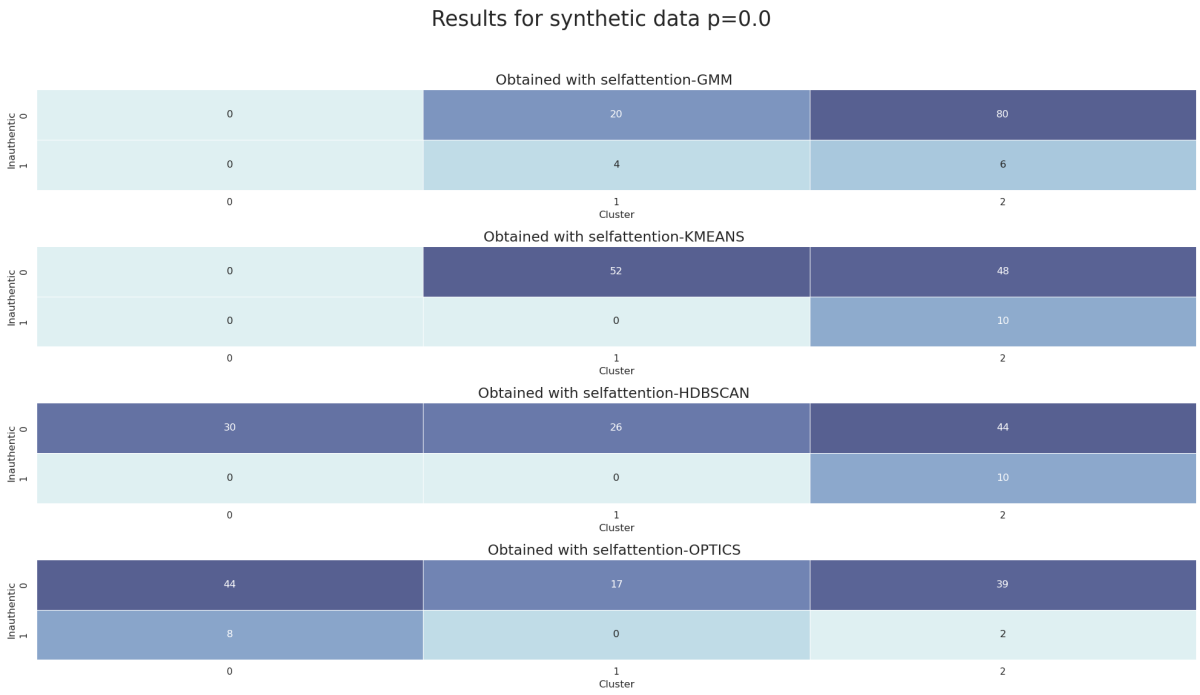

Figure 5.11: Results based on self-attention matrix for $p = 0$ dataset.

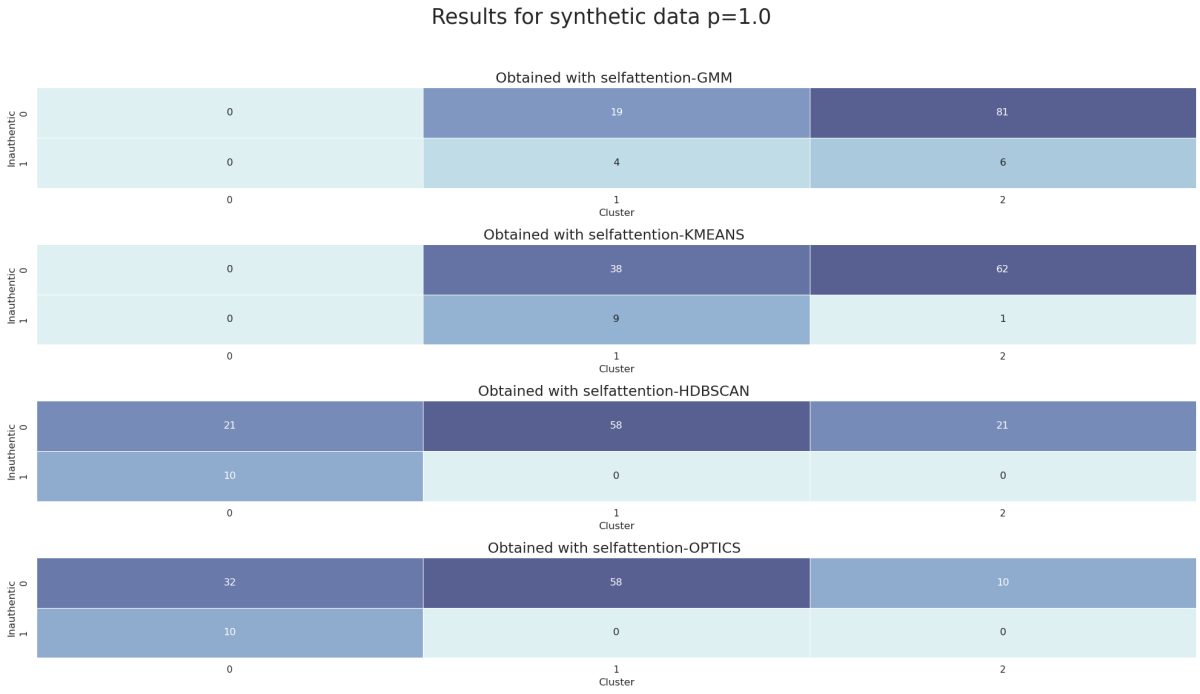

Figure 5.12: Results based on self-attention matrix for $p = 1$ dataset.

thermore, it is possible to notice that OPTICS improves its results with the self-attention-based model. Conversely, this based-on model increases the quality of the KMEANS's clustering only when inauthentic accounts are less influenced by other users, specifically when $c_\alpha \leq 1$.

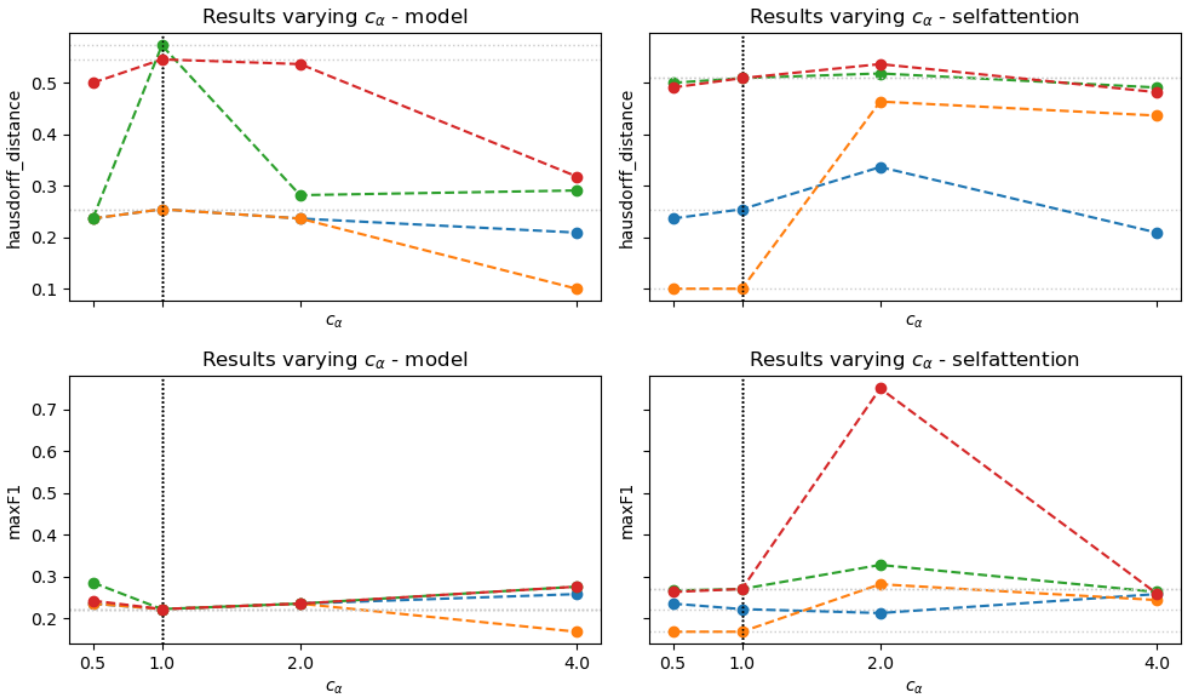

Figure 5.13: Results obtained via the clustering techniques GMM (blue line), KMEANS (orange), HDBSCAN (green) and OPTICS (red) as the value of c_α varies.

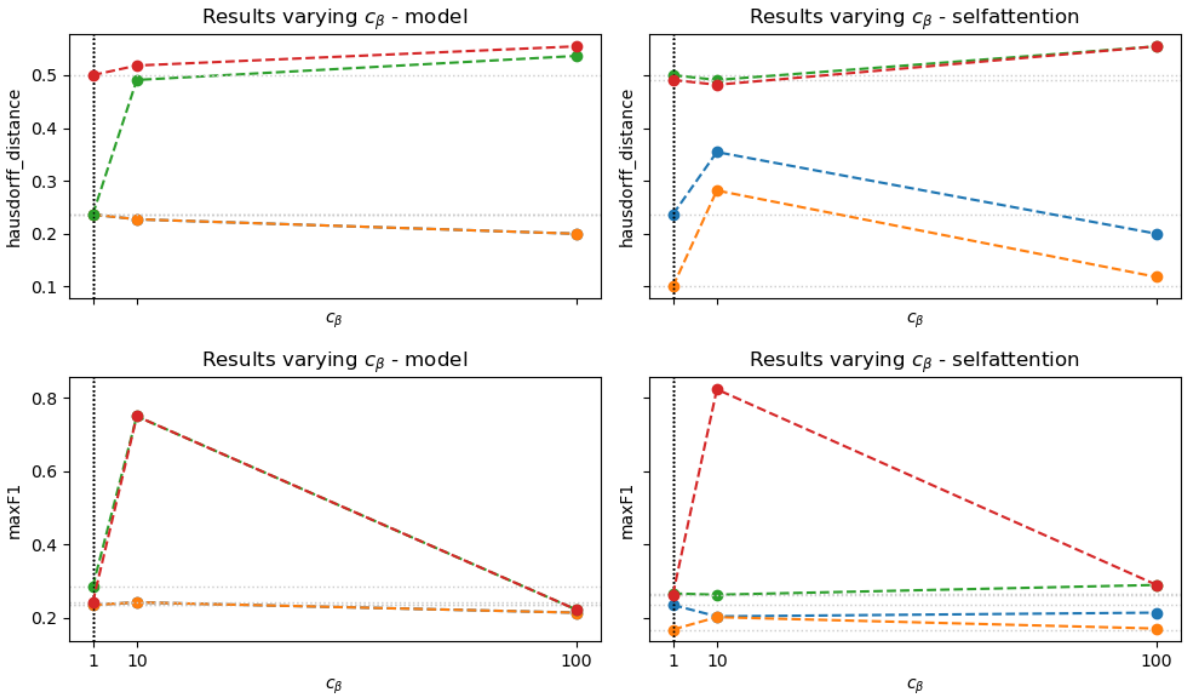

Figure 5.14: Results obtained via the clustering techniques GMM (blue line), KMEANS (orange), HDBSCAN (green) and OPTICS (red) as the value of c_β varies.

Regarding the scenario in which c_β varies, the results do not entirely align with our initial expectations. We hypothesised that the cases where $c_\beta = 10$ and $c_\beta = 100$ would yield the most

favourable results overall. However, the subplots do not consistently support this hypothesis across all detection methods. Specifically, Hausdorff distance indicates a decline in the clustering performance when the value of c_β is equal to 10, while the maxF1 metric shows improvements for the models based on Sharma et al.'s detection. Interestingly, the maxF1 measure indicates that the clustering results for the $c_\beta = 1$ and $c_\beta = 100$ scenarios are not easily distinguishable. This suggests that even with a decay rate a hundred times greater than the decay rate of authentic accounts, coordinated behaviour remains challenging to detect unless the other parameter values differ remarkably. Despite these findings, it is still possible to compare the clustering techniques. In terms of the Hausdorff measure, KMEANS consistently outperforms the other methods across all values of c_β and shows improvement within the self-attention-based model. On the other hand, OPTICS yields the best performance when considering the maxF1 metric.

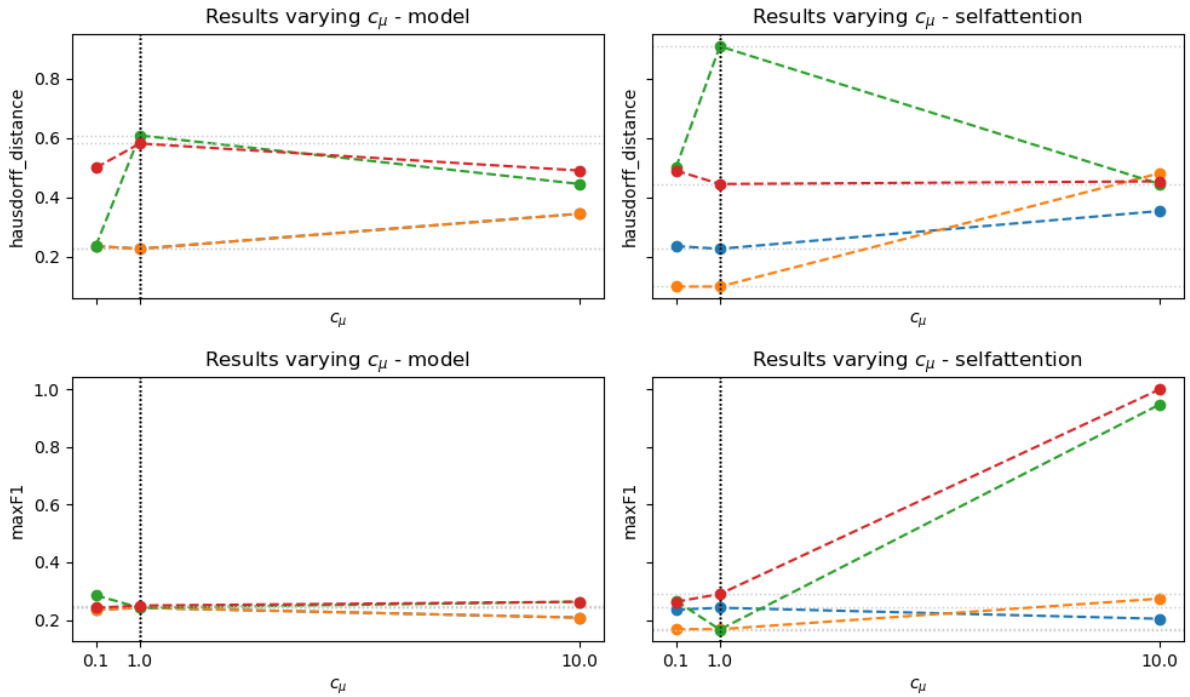

Figure 5.15: Results obtained via the clustering techniques GMM (blue line), KMEANS (orange), HDBSCAN (green) and OPTICS (red) as the value of c_μ varies.

The results obtained by varying the parameter c_μ align with our initial expectations. The most challenging scenario for detecting coordination occurs when the baseline value equals 1. In contrast, coordinated behaviour becomes easier to expose when the baseline is ten times greater than the average baseline of authentic accounts. In terms of the Hausdorff measure, KMEANS performs the best among the methods based on Sharma et al.'s model, and its performance improves when applied to the self-attention-based model, particularly when dealing with more similar users, where $c_\mu \leq 1$. Additionally, the highest maxF1 score is achieved by OPTICS in the self-attention version.

Finally, the results obtained in the scenario where p is varied reveal complex and highly inter-

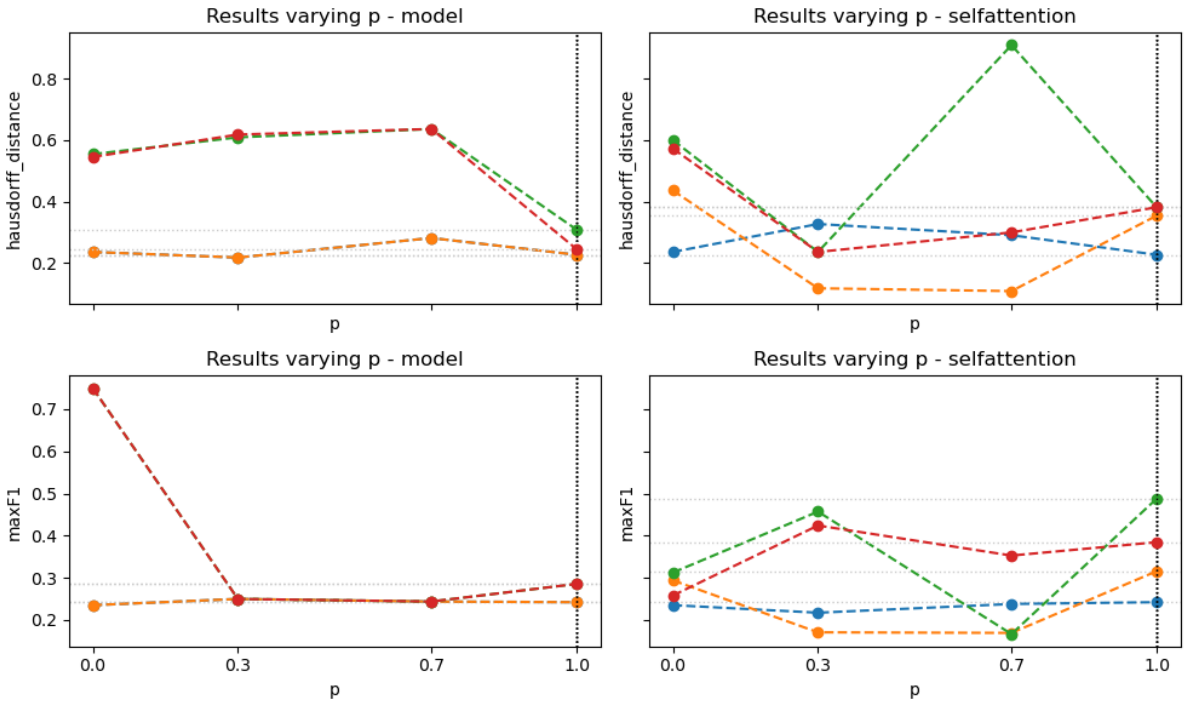

Figure 5.16: Results obtained via the clustering techniques GMM (blue line), KMEANS (orange), HDBSCAN (green) and OPTICS (red) as the value of p varies.

twined line patterns across the different clustering methods. The level of interaction between users does not seem to significantly impact the results achieved by KMEANS, as its results remain relatively stable regardless of changes in p. This suggests that KMEANS may be less sensitive to variations in user interaction levels in this context. In contrast, the level of interaction appears to have a more pronounced effect on OPTICS's performance. Specifically, this method shows its best detection performance at the extremes of the interaction spectrum, indicating that it is more effective in scenarios where the interaction level is either very low or very high. Contrary to our initial expectations, the most challenging scenario for detecting coordinated behaviour occurs when user interactions are unclear and fall somewhere in the middle of the spectrum. In this situation, the lack of distinct interaction patterns makes it difficult for clustering methods to accurately separate authentic and inauthentic users. This finding suggests that detection models struggle the most when user interactions are neither minimal nor extreme but rather ambiguous and harder to characterise.

In conclusion, while no single method consistently outperforms others across all scenarios, certain clustering techniques demonstrate particular strength depending on the evaluation metric and the parameter settings. For example, KMEANS tends to excel in terms of the Hausdorff measure, especially when dealing with scenarios when the differences between user groups are more pronounced. Moreover, its performance is most improved by choosing the self-attention-based method, especially when the inauthentic Hawkes parameter values are more similar to those of the not-coordinated accounts. Conversely, OPTICS often achieves superior performance when evaluated using the maxF1 metric, particularly in cases where interaction levels are at the extremes.

These findings suggest that the effectiveness of a detection method is highly context-dependent, and the choice of the best approach should be guided by the dataset's specific characteristics and the analysis's primary objectives. Consequently, a combination of clustering techniques may be necessary to achieve the most robust detection of coordinated behaviour across diverse scenarios.

5.2.4 Case study comparison with network science methods

After analysing the results obtained with the method and our extensions, we have used our framework to test other state-of-the-art models. In particular, we employed a single-layer user network approach with Evenly Distributed Overlapping [Pacheco et al., 2021, Keller et al., 2020, Ng and Carley, 2023, Schoch et al., 2022] or Adjacent [Weber and Neumann, 2020] sliding windows filtering in conjunction with the Louvain community detection method. To ensure meaningful encoding of activities and improve computational times, we applied a further layer filtering the user network edges based on either the Jaccard [Pacheco et al., 2021] or the cardinality similarity measures [Keller et al., 2020, Ng and Carley, 2023, Schoch et al., 2022, Weber and Neumann, 2020]. The reader is referred to Section 1.2.1 for a detailed explanation of the rationale behind the use of these approaches and their underlying assumptions.

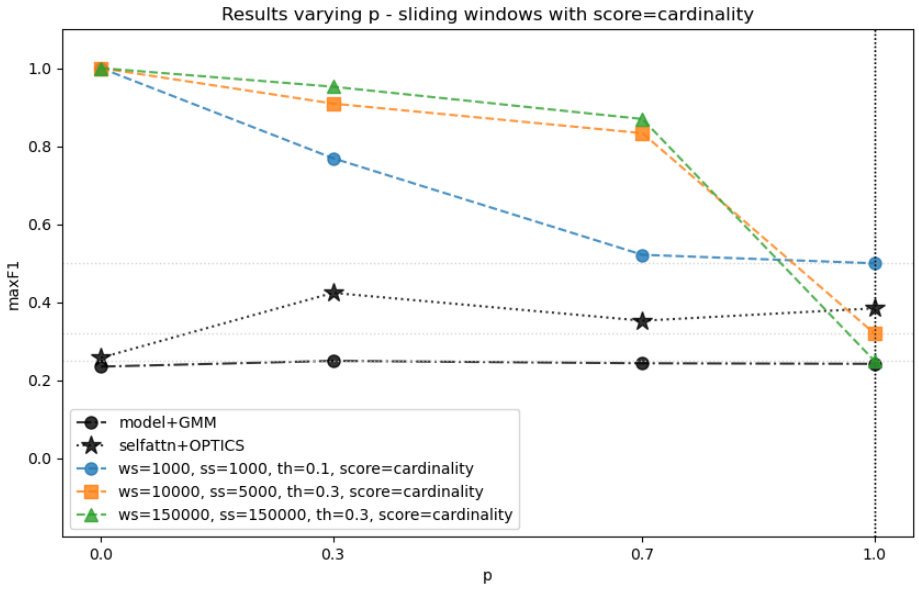

Figure 5.17: Results obtained via sliding windows (coloured lines) as the value of p varies.

We applied the aforementioned methods, focusing on scenarios in which the percentage of interaction among authentic and coordinated users varies. We will identify each method with its window size ws (in minutes), step size ss (in minutes), similarity measure $score$ (scalar) and threshold th (scalar). The values of these parameters have been selected manually to further minimise computational time while ensuring a comprehensive analysis. Figure 5.17 and 5.18 report these results compared to the detection obtained with Sharma+GMM and the self-attention+OPTICS model. It is immediately evident that when there is a portion of cascades involving only inauthentic interactions ($p < 1.0$), the sliding windows methods outperform the

models based on temporal point processes in detecting coordinated users. Nevertheless, these implementations are sensitive to the parameter settings, as larger window size consistently lead to higher detection scores. On the other hand, when every activity trace is mixed ($p = 1.0$), the sliding window performance decreases significantly. In this situation, our model extension achieves a higher maxF1 score compared to all large-window implementations. However, the cardinality-score method with the smallest considered window size, $ws = 1000$, achieves the best result overall.

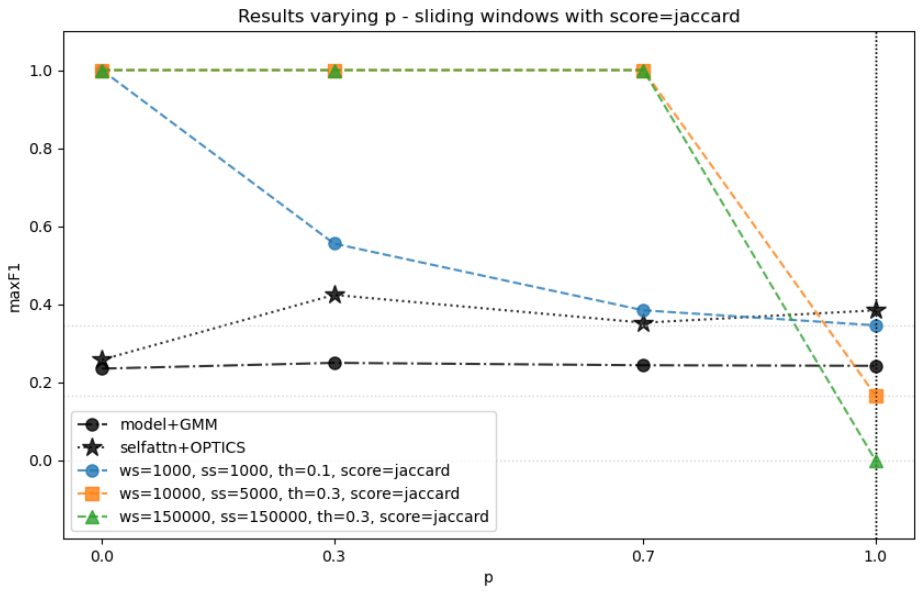

Figure 5.18: Results obtained via sliding windows (coloured lines) as the value of p varies.

In conclusion, our model extension demonstrates significant potential in scenarios where coordination detection becomes particularly challenging, such as when all cascades are mixed ($p = 1.0$). On the other hand, the sliding window approach consistently excels in coordination detection, especially when some cascades consist of isolating coordinated interactions. Nonetheless, their efficacy can significantly diminish if their window size is not appropriately configured in completely mixed cascades. The real-world implications of struggling to detect coordination translate into the challenges associated with identifying specific types of coordinated users. For example, consider spammers: their interactions are likely part of isolated cascades due to the repetitive nature of content dissemination on social media without attempts to obscure intent. Our results indicate that the sliding window approaches are most effective in detecting such inauthentic users. Conversely, when users seek to conceal their identities, as observed with sockpuppets, their activities tend to be embedded within fully mixed cascades. Sockpuppets deliberately blend their interactions with those of authentic accounts, complicating the detection of their coordinated actions. In these scenarios, the proposed self-attention extension demonstrates considerable potential for uncovering this mix of authentic and deceptive behaviour. Furthermore, our findings underscore the importance of a thorough understanding of each method's performance across diverse scenarios, emphasising the necessity of analysing these methods within a comprehensive

framework, such as the one we have introduced.

Conclusions and future perspective

In this thesis, we have explored the complexities of coordination detection and how marked temporal point processes could be leveraged in this field. Starting with an introduction to coordinated behaviour campaigns and a literature review, we have proceeded with the presentation of temporal point processes covering the main concept and the existing methods for generating simulations. We then have moved on to understanding how these mathematical concepts could be used in coordination detection. Since summarising the history was the first issue that had to be faced, we have introduced masked self-attention. In particular, we have presented its concept, providing a method for visualising it, allowing the reader to deeply comprehend this extremely famous technique. Subsequently, we have presented Sharma et al.'s state-of-the-art model, which tackles coordination detection via Gaussian mixture models. In studying and testing this method, we have realised that there could be two directions in which it could be extended. On the one hand, we have proposed using HDBSCAN and OPTICS clustering techniques, which do not require knowing the number of clusters in advance. Moreover, we have developed an additional version of the model that based the clustering recovery on the user-to-user self-attention weights. In such a way, we could better capture the influences coordinated inauthentic users have with each other and the rest of the social media users.

Driven by the necessity of datasets with background knowledge for a systematic comparison of detection methods, we have implemented a novel framework for simulating social media activities. The proposed suite facilitates researchers in assessing the effectiveness of their coordination detection mechanisms across a variety of settings, including clusters of inauthentic users engineered to amplify one another's content, as well as synchronised accounts that intensely interact with their designated targets. Ultimately, through this framework, we have assessed the three versions of the detection method in various scenarios and have compared the findings with cutting-edge methods based on network-based approaches. The results of our case studies have shown that no individual clustering method consistently outperforms others across all scenarios. KMEANS demonstrate efficacy when using the Hausdorff measure, particularly under conditions where significant differences exist between user groups. Furthermore, the OPTICS-based self-attention mechanism demonstrates better performance in the maxF1 metric, particularly in scenarios characterised by minimal interactions or entirely mixed cascades. Regarding the comparison with detection methods based on the network approach, the findings indicate that sliding window techniques surpass temporal point process models in identifying isolated inauthentic interactions. However, the efficacy of sliding window methods is highly dependent on window size: larger windows produce superior detection metrics but require more computational

expenses as more activities are processed together. In contrast, within fully mixed interaction contexts, the efficiency of sliding windows is reduced. Although the method with the smallest window size has proven to be the most effective in the case of mixed interactions, our extended model registered high detection performance. In particular, the maxF1 score it has obtained in the scenario characterised by fully mixed cascades was the second highest overall, surpassed only by the cardinality sliding window with the smallest window size.

In conclusion, our research has highlighted the importance of deeply comprehending each method's performance across varying scenarios. In fact, the effectiveness of the detection methods we analysed is dependent on the specific characteristics of the dataset in question. Consequently, we propose that future research in coordination detection should focus on applying our framework to conduct a complete and systematic evaluation of these methods. Moreover, our findings indicate that additional studies should explore integrating clustering recovery of user-to-user self-attention weights with the network approaches. We believe that this research direction holds significant potential for yielding improved results.

Appendix

Further analysis for the synthetic case study

In Table 1, we report the results of the STAN implementation. Among the various columns, the column reporting the value of *Rhat* is the most interesting for validating the results. Indeed, the convergence of the chains is ensured when its value is close to 1, which is the case for every parameter.

Parameter	Mean	SE Mean	SD	5%	95%	n_eff	Rhat
hyp_baseline_a	1.2151e-01	1.6260e-03	6.0138e-02	4.3249e-02	2.3511e-01	1367.9225	1.0003
hyp_alpha_a	7.2923e-01	2.0078e-02	3.3445e-01	3.0829e-01	1.3467e+00	277.4576	1.0206
hyp_beta_a	5.3195e-01	1.0191e-02	2.8748e-01	1.7924e-01	1.0765e+00	795.6765	1.0051
hyp_baseline_b	1.4723e+00	2.6949e-02	1.2522e+00	1.5594e-01	3.9285e+00	2159.2386	0.9989
hyp_alpha_b	2.4795e+00	6.0542e-02	1.4990e+00	6.2093e-01	5.2726e+00	613.0761	1.0079
hyp_beta_b	2.0034e+00	3.4913e-02	1.3827e+00	3.6650e-01	4.7576e+00	1568.5394	0.9984
baseline[1,1]	2.9218e-04	5.0893e-06	1.4069e-04	1.0847e-04	5.5570e-04	764.1918	1.0021
baseline[2,1]	5.7014e-04	2.4021e-05	5.8967e-04	1.2526e-06	1.7900e-03	602.5973	1.0070
baseline[3,1]	1.0748e-04	8.5860e-07	3.4234e-05	5.8396e-05	1.7127e-04	1589.7721	1.0018
baseline[4,1]	7.4420e-06	8.2706e-08	2.6602e-06	3.7714e-06	1.2094e-05	1034.5709	1.0023
adjacency[1,1]	1.8732e-01	2.3870e-03	6.8027e-02	1.0391e-01	3.1976e-01	812.1818	1.0045
adjacency[1,2]	3.8301e-01	1.9420e-02	7.3254e-01	1.6893e-03	1.3584e+00	1422.8056	0.9993
adjacency[1,3]	5.6417e-01	1.3743e-02	5.2287e-01	5.8362e-02	1.5339e+00	1447.4752	1.0020
adjacency[1,4]	3.7810e-01	2.3023e-02	1.0335e+00	1.1803e-03	1.4182e+00	2015.0303	0.9994
adjacency[2,1]	3.7201e-01	1.7378e-02	6.2079e-01	2.1746e-03	1.4192e+00	1276.1046	1.0030
adjacency[2,2]	3.6444e-01	1.3439e-02	5.6931e-01	1.2722e-03	1.2907e+00	1794.4985	0.9986
adjacency[2,3]	3.6646e-01	1.7974e-02	7.1803e-01	1.1976e-03	1.2165e+00	1595.9012	1.0011
adjacency[2,4]	3.9693e-01	1.8687e-02	6.0862e-01	2.5772e-03	1.5411e+00	1060.7548	1.0000
adjacency[3,1]	2.5105e-01	5.6945e-03	1.9467e-01	3.6196e-02	6.4308e-01	1168.6684	1.0057
adjacency[3,2]	3.7041e-01	1.6837e-02	6.2328e-01	1.2076e-03	1.2857e+00	1370.4304	1.0041
adjacency[3,3]	1.3567e-01	8.3839e-04	2.3838e-02	9.7806e-02	1.7635e-01	808.4192	1.0121
adjacency[3,4]	1.2313e-01	2.4462e-03	9.9249e-02	1.7818e-02	3.1006e-01	1646.0784	1.0011
adjacency[4,1]	4.0178e-01	1.8741e-02	7.1040e-01	1.9142e-03	1.5013e+00	1436.9280	1.0039
adjacency[4,2]	3.7891e-01	2.2780e-02	9.5627e-01	1.6414e-03	1.3425e+00	1762.1520	1.0005
adjacency[4,3]	2.3344e-01	4.1833e-03	1.3133e-01	5.3473e-02	4.7330e-01	985.5097	1.0025
adjacency[4,4]	7.1661e-02	3.9121e-04	1.2664e-02	5.5888e-02	9.6024e-02	1047.8381	1.0059
decay[1,1]	1.3046e-01	1.2615e-03	3.5810e-02	8.1975e-02	1.9806e-01	805.7411	1.0042
decay[2,1]	4.3853e-01	5.4562e-02	1.4301e+00	4.5433e-03	1.6044e+00	686.9719	1.0090
decay[3,1]	9.1935e-02	5.0457e-04	1.2816e-02	7.0412e-02	1.1314e-01	645.1880	1.01418
decay[4,1]	5.7006e-02	2.9234e-04	9.4102e-03	4.5268e-02	7.5020e-02	1036.1354	1.00538

Table 1: Simulation summary for Stan results.

Moreover, Figure 19 and 20 report the trace plots for the simulated hyperparameters and their histograms. Finally, Figure 21, 22, 23 and 24 reports the results for several other metrics that could be considered for the clustering evaluations.

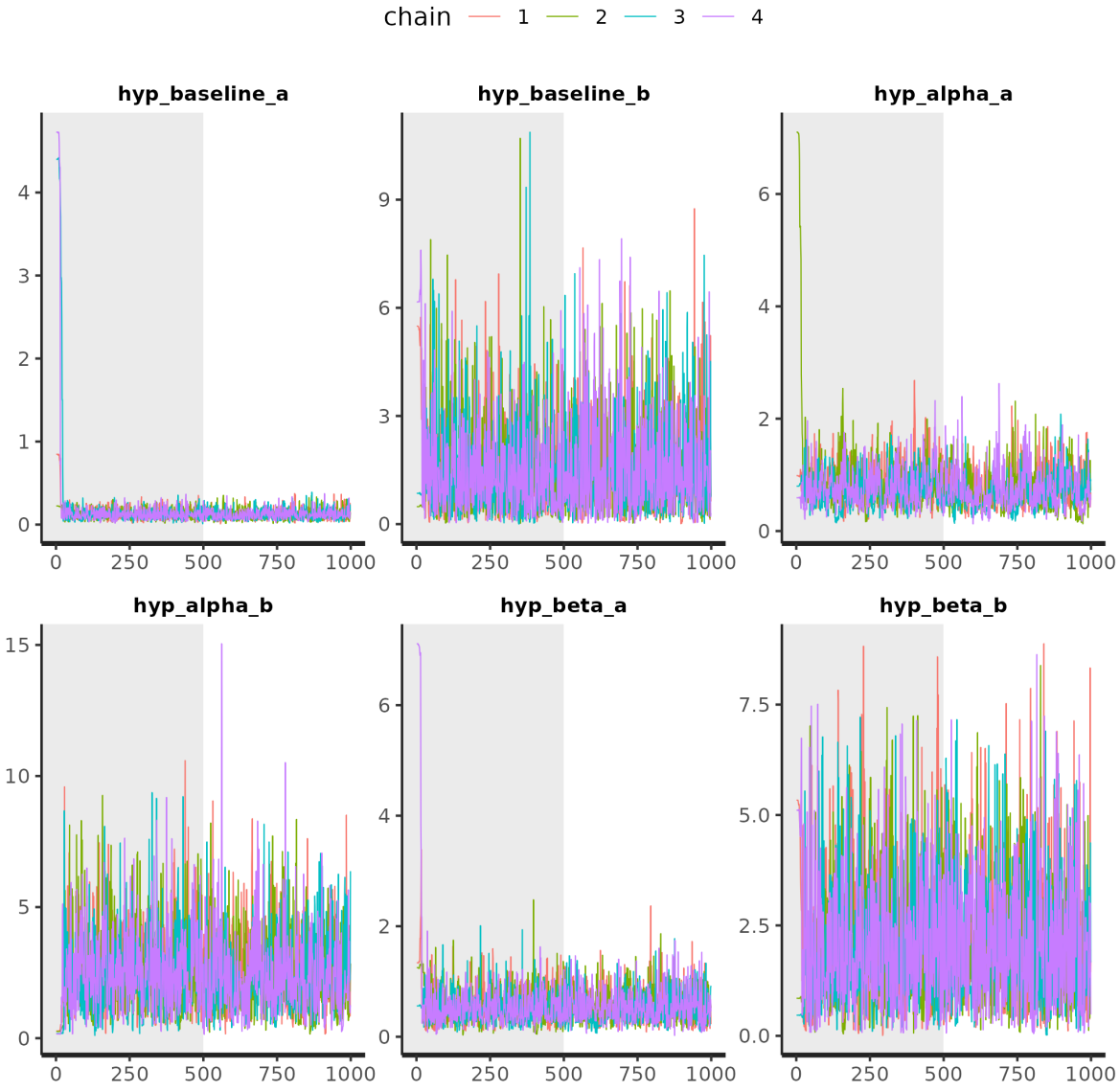

Figure 19: Stan trace plot for the simulated hyperparameters.

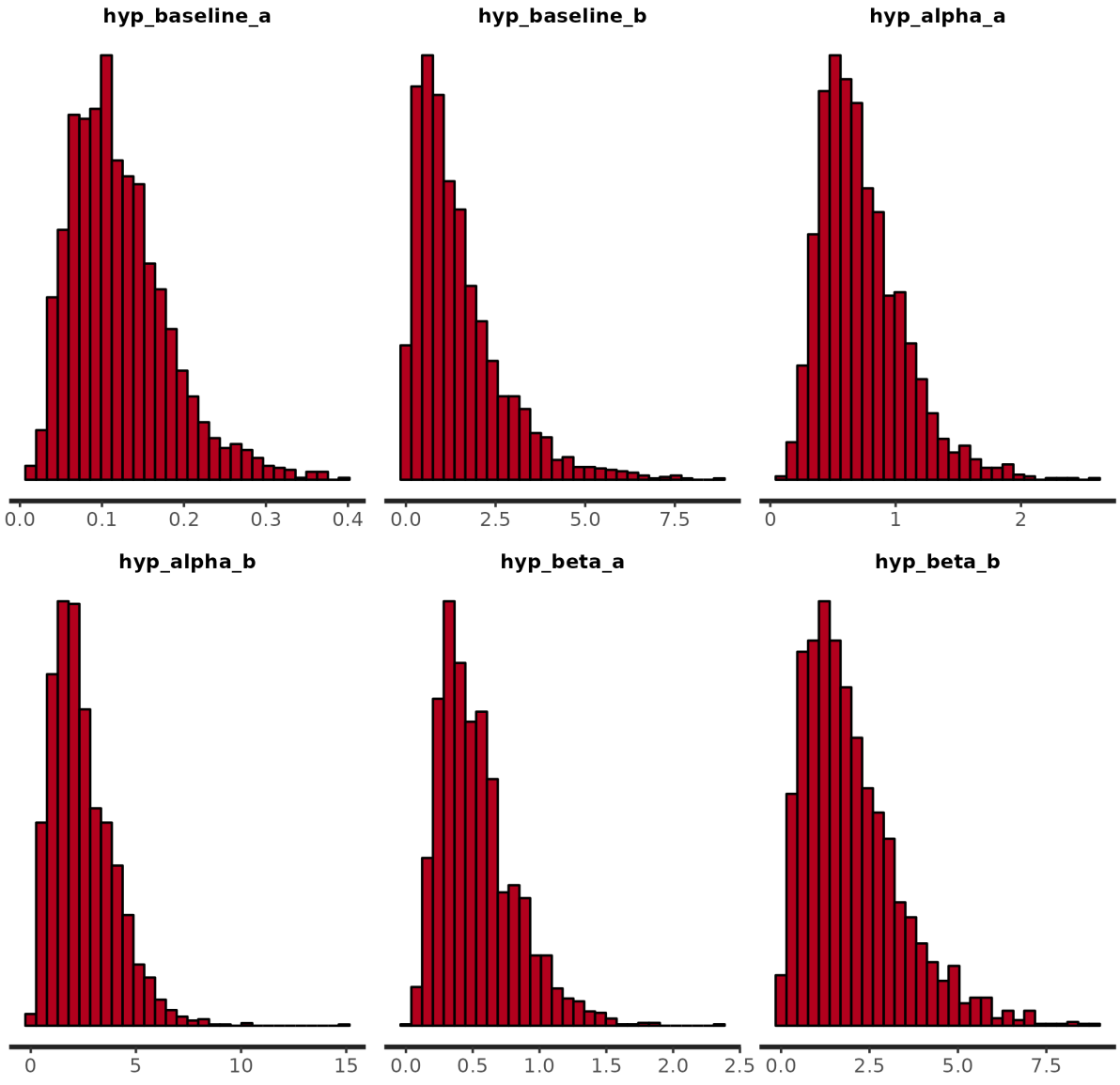

Figure 20: Stan histogram for the estimated hyperparameters.

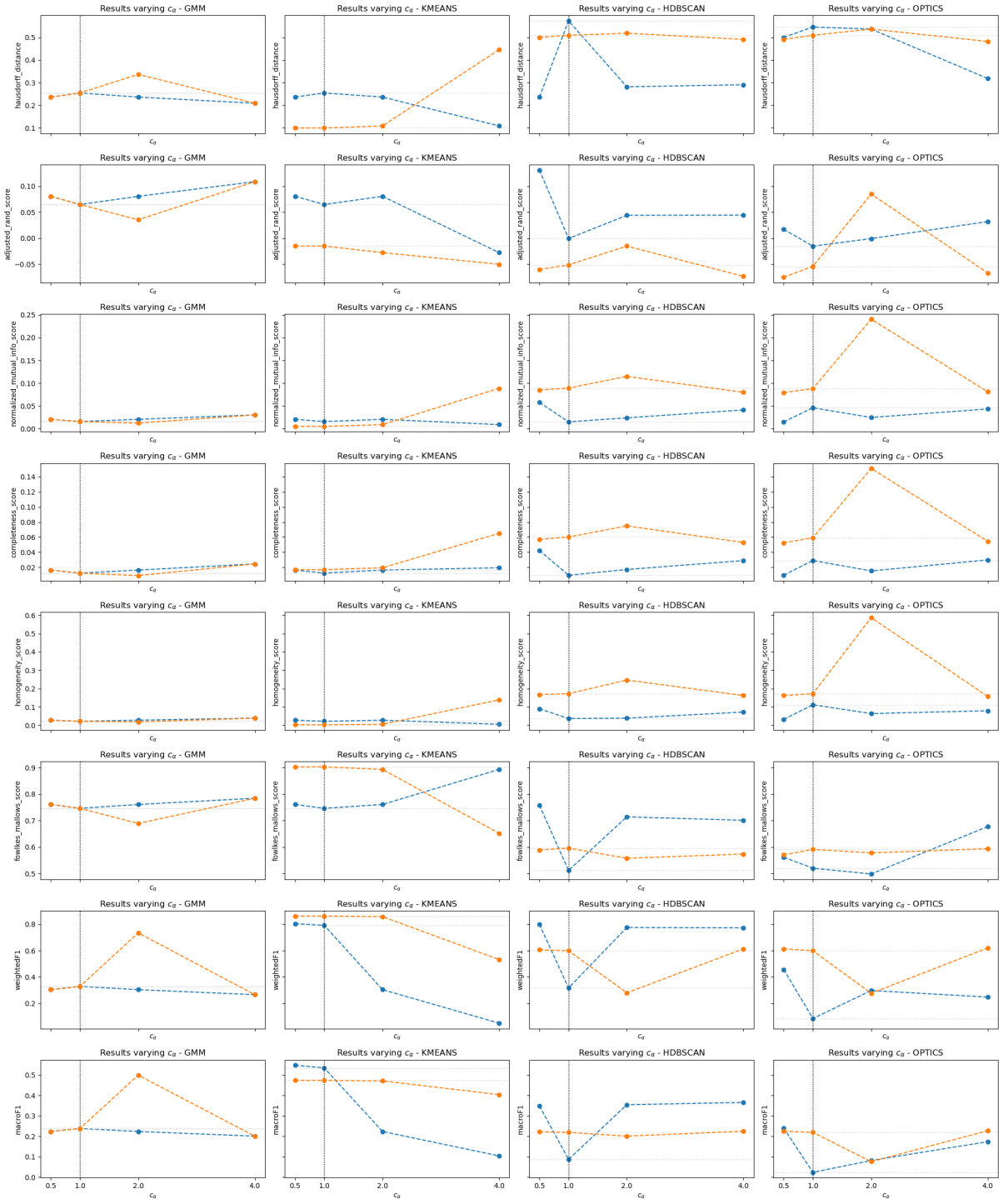

Figure 21: Results for different metrics as c_α varies, where the yellow line reports the values of the self-attention-based methods, while the blue line represents the models based on Sharma et al.'s method.

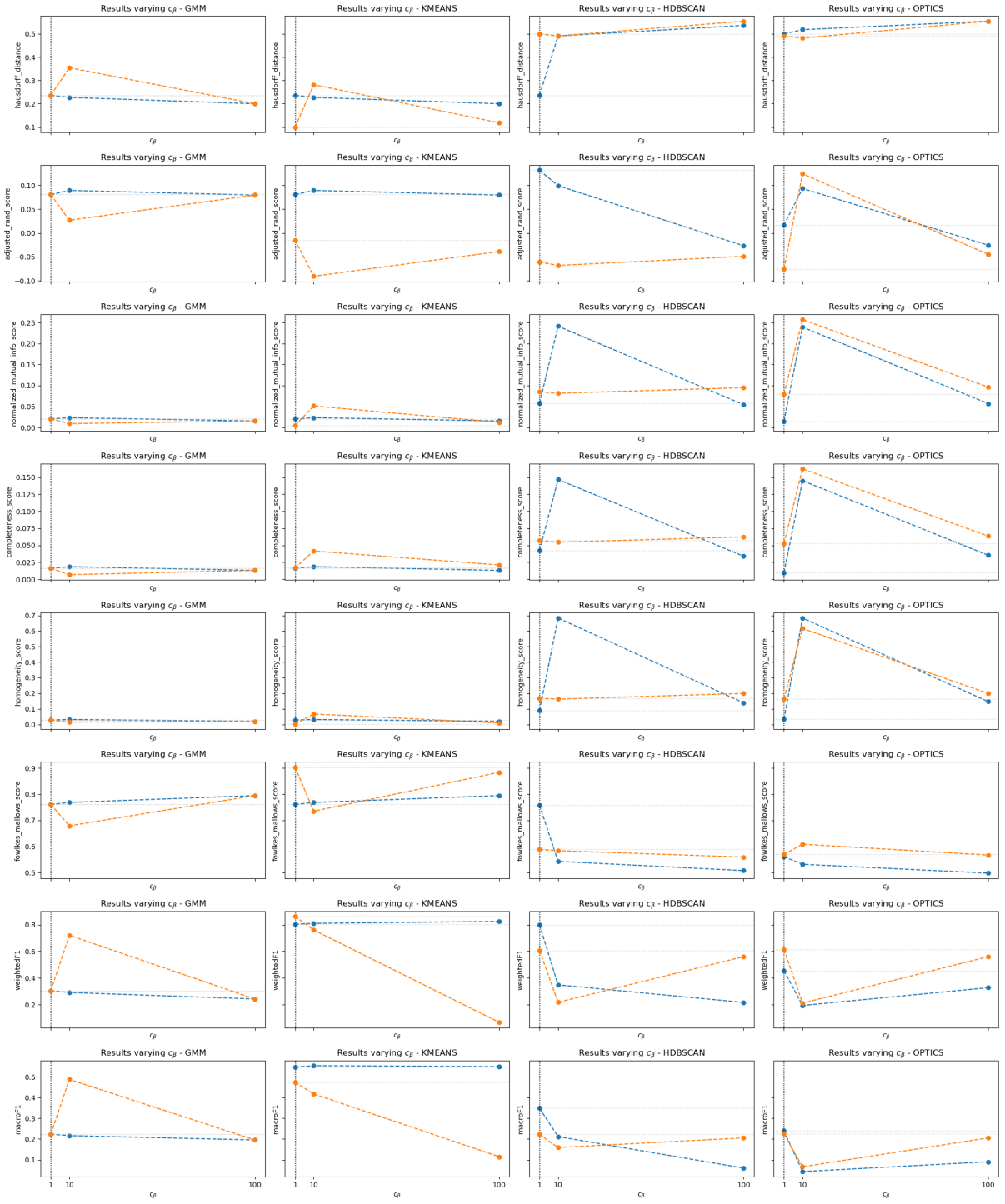

Figure 22: Results for different metrics as c_β varies, where the yellow line reports the values of the self-attention-based methods, while the blue line represents the models based on Sharma et al.'s method.

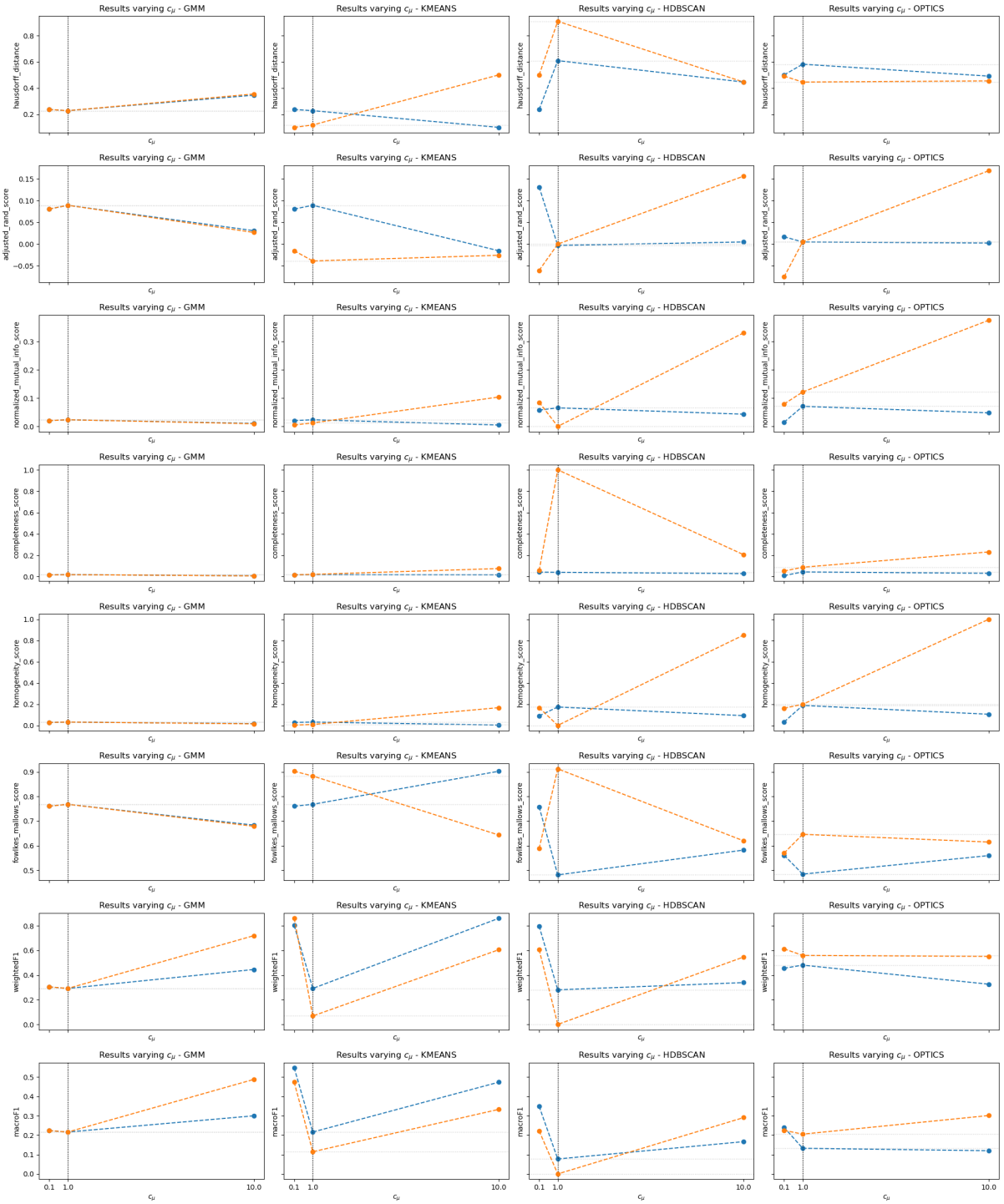

Figure 23: Results for different metrics as c_μ varies, where the yellow line reports the values of the self-attention-based methods, while the blue line represents the models based on Sharma et al.'s method.

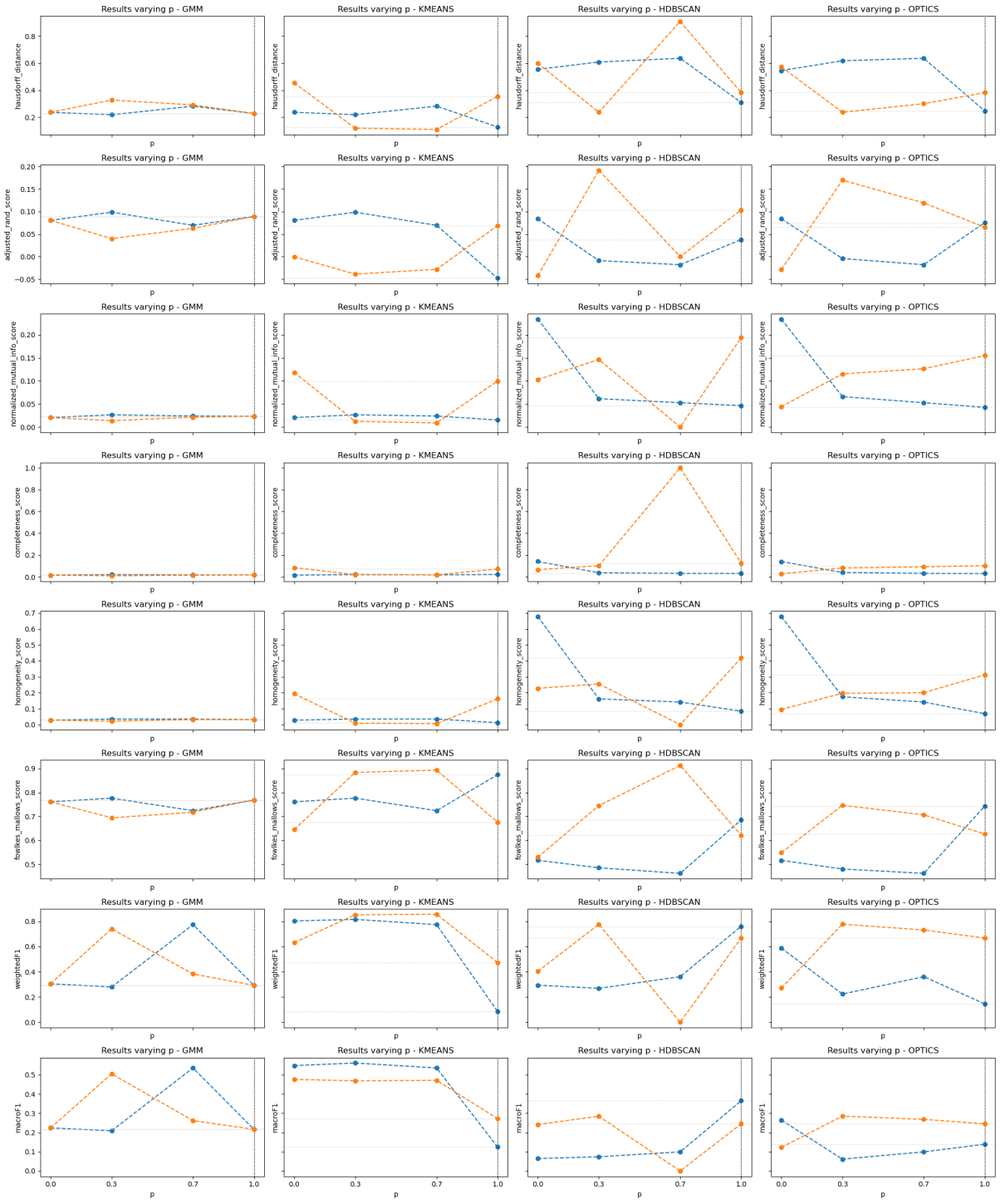

Figure 24: Results for different metrics as p varies, where the yellow line reports the values of the self-attention-based methods, while the blue line represents the models based on Sharma et al.'s method.

Bibliography

Mihael Ankerst, Markus M Breunig, Hans-Peter Kriegel, and Jörg Sander. Optics: Ordering points to identify the clustering structure. *ACM Sigmod record*, 1999.

Dennis Assenmacher, Lena Adam, Heike Trautmann, and Christian Grimme. Towards real-time and unsupervised campaign detection in social media. In *33rd International FLAIRS Conference, FLAIRS 2020*, pages 303–306. AAAI, 2020.

JL Ba. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Samantha Bradshaw and Philip N. Howard. The global disinformation order: 2019 global inventory of organised social media manipulation, 2019. URL `https://shorturl.at/yrYmy`. Accessed: 2024-09-28.

Peter J Brockwell and Richard A Davis. *Time series: theory and methods*. Springer science & business media, 1991.

Ricardo JGB Campello, Davoud Moulavi, Arthur Zimek, and Jörg Sander. Hierarchical density estimates for data clustering, visualization, and outlier detection. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 2015.

Matthias Carnein, Dennis Assenmacher, and Heike Trautmann. Stream clustering of chat messages with applications to twitch streams. In *Advances in Conceptual Modeling: ER 2017 Workshops AHA, MoBiD, MREBA, OntoCom, and QMMQ, Valencia, Spain, November 6–9, 2017, Proceedings 36*. Springer, 2017.

José E Chacón. A population background for nonparametric density-based clustering. *Statistical Science*, 2015.

Yuanda Chen. Thinning algorithms for simulating point processes. *Florida State University, Tallahassee, FL*, 2016.

Victor Chomel, Maziyar Panahi, and David Chavalarias. Manipulation during the french presidential campaign: coordinated inauthentic behaviors and astroturfing analysis on text and images. In *International Conference on Complex Networks and Their Applications*. Springer, 2022.

Cybersecurity and Infrastructure Security Agency CISA, Office of the Director of National Intelligence ODNI, and the Federal Bureau of Investigation FBI. Securing election infrastructure against the tactics of foreign malign influence operations, 2024. URL `https://shorturl.at/zBSgB`. Accessed: 2024-09-20.

The Centre for Research and Evidence on Security Threats CREST. Russian influence and interference measures following the 2017 uk terrorist attacks, 2024. URL https://crestresearch.ac.uk/resources/russian-influence-uk-terrorist-attacks/. Accessed: 2024-09-20.

Cambridge University Press CUP. Meaning of disinformation, 2024. URL https://dictionary.cambridge.org/dictionary/english/disinformation. Accessed: 2024-09-28.

Daryl J Daley and David Vere-Jones. *An introduction to the theory of point processes: volume I: elementary theory and methods*. Springer Science & Business Media, 2006.

Andrew Dawson and Martin Innes. How russia's internet research agency built its disinformation campaign. *The Political Quarterly*, 2019.

European External Action Service EEAS. 2nd eeas report on foreign information manipulation and interference (fimi) threats. Technical report, 2024. URL https://www.eeas.europa.eu/eeas/2nd-eeas-report-foreign-information-manipulation-and-interference-threats_en. Accessed: 2024-09-20.

Auriant Emeric and Chomel Victor. Interpretable cross-platform coordination detection on social networks. In *International Conference on Complex Networks and Their Applications*, pages 143–155. Springer, 2023.

Facebook. Coordinated inauthentic behavior explained, 2018. URL https://about.fb.com/news/2018/12/inside-feed-coordinated-inauthentic-behavior/. Accessed: 2024-09-20.

Emilio Ferrara, Onur Varol, Clayton Davis, Filippo Menczer, and Alessandro Flammini. The rise of social bots. *Communications of the ACM*, 2016.

Fabio Giglietto, Nicola Righetti, Luca Rossi, and Giada Marino. It takes a village to manipulate the media: coordinated link sharing behavior during 2018 and 2019 italian elections. *Information, Communication & Society*, 2020.

Andrew M. Guess and Benjamin A. Lyons. *Misinformation, Disinformation, and Online Propaganda*. Cambridge University Press, 2020.

Alan G Hawkes. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 1971.

GE Hinton. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

Letizia Iannucci. Detecting coordinated online behaviour—a multiplex network approach. 2023.

Franziska B Keller, David Schoch, Sebastian Stier, and JungHwan Yang. Political astroturfing on twitter: How to coordinate a disinformation campaign. *Political communication*, 2020.

Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Srijan Kumar, Justin Cheng, Jure Leskovec, and VS Subrahmanian. An army of me: Sockpuppets in online discussion communities. In *Proceedings of the 26th international conference on world wide web*, 2017.

Patrick J Laub, Thomas Taimre, and Philip K Pollett. Hawkes processes. *arXiv preprint arXiv:1507.02822*, 2015.

Renan S Linhares, José M Rosa, Carlos HG Ferreira, Fabricio Murai, Gabriel Nobre, and Jussara Almeida. Uncovering coordinated communities on twitter during the 2020 us election. In *2022 IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM)*. IEEE, 2022.

Luca Luceri, Silvia Giordano, and Emilio Ferrara. Detecting troll behavior via inverse reinforcement learning: A case study of russian trolls in the 2016 us election. In *Proceedings of the international AAAI conference on web and social media*, volume 14, pages 417–427, 2020.

Lorenzo Mannocci, Michele Mazza, Anna Monreale, Maurizio Tesconi, and Stefano Cresci. Detection and characterization of coordinated online behavior: A survey. *arXiv preprint arXiv:2408.01257*, 2024.

Enrico Mariconti, Guillermo Suarez-Tangil, Jeremy Blackburn, Emiliano De Cristofaro, Nicolas Kourtellis, Ilias Leontiadis, Jordi Luque Serrano, and Gianluca Stringhini. "you know what to do" proactive detection of youtube videos targeted by coordinated hate attacks. *Proceedings of the ACM on Human-Computer Interaction*, 2019.

Kumari Neha, Vibhu Agrawal, Saurav Chhatani, Rajesh Sharma, Arun Balaji Buduru, and Ponnurangam Kumaraguru. Understanding coordinated communities through the lens of protest-centric narratives: A case study on# caa protest. In *Proceedings of the International AAAI Conference on Web and Social Media*, 2024.

Lynnette Hui Xian Ng and Kathleen M Carley. Do you hear the people sing? comparison of synchronized url and narrative themes in 2020 and 2023 french protests. *Frontiers in big Data*, 2023.

Shchur O., Biloš M., and Günnemann S. Intensity-free learning of temporal point processes. *arXiv preprint arXiv:1909.12127*, 2019.

Yosihiko Ogata. On lewis' simulation method for point processes. *IEEE transactions on information theory*, 1981.

Diogo Pacheco, Alessandro Flammini, and Filippo Menczer. Unveiling coordinated groups behind white helmets disinformation. In *Companion proceedings of the web conference 2020*, 2020.

Diogo Pacheco, Pik-Mai Hui, Christopher Torres-Lugo, Bao Tran Truong, Alessandro Flammini, and Filippo Menczer. Uncovering coordinated networks on social media: methods and case studies. In *Proceedings of the international AAAI conference on web and social media*, 2021.

Natali Ruchansky, Sungyong Seo, and Yan Liu. Csi: A hybrid deep model for fake news detection. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2017.

Grant Sanderson. Visualizing attention, a transformer's heart, 2024. URL https://www.3blue1brown.com/lessons/attention#title.

David Schoch, Franziska B Keller, Sebastian Stier, and JungHwan Yang. Coordination patterns reveal online political astroturfing across the world. *Scientific reports*, 2022.

Chengcheng Shao, Giovanni Luca Ciampaglia, Onur Varol, Alessandro Flammini, and Filippo Menczer. The spread of fake news by social bots. *arXiv preprint arXiv:1707.07592*, 2017.

Karishma Sharma, Yizhou Zhang, Emilio Ferrara, and Yan Liu. Identifying coordinated accounts on social media through hidden influence and group behaviours. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021.

Stan Development Team. RStan: the R interface to Stan, 2024. URL https://mc-stan.org/. R package version 2.32.6.

Daiki Suzuki and Sho Tsugawa. Evaluating community detection algorithms for multilayer networks: Effectiveness of link weights and link direction. *Complex Systems*, 2023.

Serena Tardelli, Leonardo Nizzoli, Maurizio Tesconi, Mauro Conti, Preslav Nakov, Giovanni Da San Martino, and Stefano Cresci. Temporal dynamics of coordinated online behavior: Stability, archetypes, and influence. *Proceedings of the National Academy of Sciences*, 2024.

Vincent A Traag, Ludo Waltman, and Nees Jan Van Eck. From louvain to leiden: guaranteeing well-connected communities. *Scientific reports*, 2019.

Inc. Twitter. Form 10-q quarterly report of the securities exchange act of 1934, 2014. URL https://www.sec.gov/Archives/edgar/data/1418091/000156459014003474/twtr-10q_20140630.htm. Accessed: 2024-09-28.

A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

Derek Weber and Frank Neumann. Who's in the gang? revealing coordinating communities in social media. In *2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. IEEE, 2020.

Derek Weber and Frank Neumann. Amplifying influence through coordinated behaviour in social networks. *Social Network Analysis and Mining*, 2021.

Da Xu, Chuanwei Ruan, Evren Korpeoglu, Sushant Kumar, and Kannan Achan. Self-attention with functional time representation learning. *Advances in neural information processing systems*, 2019.

William Emmanuel S Yu. A framework for studying coordinated behaviour applied to the 2019 philippine midterm elections. In *Proceedings of Sixth International Congress on Information and Communication Technology: ICICT 2021, London, Volume 2*. Springer, 2022.

Acknowledgments

I would like to express my heartfelt gratitude to my supervisors, Agostinelli Claudio, Iannucci Letizia, Kivelä Mikko and Gallotti Riccardo, for their invaluable guidance and mentorship throughout this research, which has significantly shaped my academic journey and personal growth.

I would also like to thank Trento University, Clesio College, Aalto University, and the Bruno Kessler Foundation for their support and the opportunities they provided, which enabled me to broaden my horizons and deepen my knowledge. Additionally, I am grateful to my colleagues for their collaboration and assistance, which made this experience both enjoyable and enriching.

Finally, I extend my deepest appreciation to my family and friends for their unwavering availability, patience, and encouragement during the course of this journey.